Sequential Decision Making For Choice Functions On Gambles

HUNTLEY, NATHAN

Sequential Decision Making For Choice Functions On Gambles

Nathan Huntley

A Thesis presented for the degree of
Doctor of Philosophy

Statistics and Probability Group
Department of Mathematical Sciences
Durham University
UK

June 2011

Sequential Decision Making For Choice Functions On Gambles

Nathan Huntley

Submitted for the degree of Doctor of Philosophy
March 2011

Abstract

Choice functions on gambles (uncertain rewards) provide a framework for studying diverse preference and uncertainty models. For single decisions, applying a choice function is straightforward. In sequential problems, where the subject has multiple decision points, it is less easy. One possibility, called a normal form solution, is to list all available strategies (specifications of acts to take in all eventualities). This reduces the problem to a single choice between gambles.

We primarily investigate three appealing behaviours of these solutions. The first, subtree perfectness, requires that the solution of a sequential problem, when restricted to a sub-problem, yields the solution to that sub-problem. The second, backward induction, requires that the solution of the problem can be found by working backwards from the final stage of the problem, removing everything judged non-optimal at any stage. The third, locality, applies only to special problems such as Markov decision processes, and requires that the optimal choice at each stage (considered separately from the rest of the problem) forms an optimal strategy.

For these behaviours, we find necessary and sufficient conditions on the choice function. Showing that these hold is much easier than proving the behaviour from first principles. It also leads to answers to related questions, such as the relationship between the normal form and another popular form of solution, the extensive form. To demonstrate how these properties can be checked for particular choice functions, and how the theory can be easily extended to special cases, we investigate common choice functions from the theory of coherent lower previsions.

Declaration

The work in this thesis is based on research carried out at the Statistical and Probability Group, the Department of Mathematical Sciences, Durham University, UK. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text. In particular, Chapter 4 and Section 6.5.1 are based on joint work with Ricardo Shirota Filho, adapted from [76].

Dedicated to

Tony Kiddle.

Acknowledgements

I am indebted to the support of my parents, as well as my supervisor Matthias Troffaes, my team-mates at Durham and Middlesbrough Chess Clubs, and my other friends, enemies and traitors; in particular Rebecca Baker, Ally Bletsas, Ric Crossman, Jonathan Cumming, David Goodman, Ruth Hilton, Joey Oliver, Rebecca O'Neil, John Rhodes, Luke Stanbra, James Thompson, Stevie Wells, Danny Williamson, and most especially my long-suffering office-mate Rachel Oxlade (a clever hobbit indeed).

Contents

List of Figures

Chapter 1

Introduction

1.1 Outline

Decision making under uncertainty is a common problem in many fields. Applications abound in, for example, medicine [39, 48], artificial intelligence [15, 21], economics [70, 59], environmental science [20, 37], operations research [19, 83], and engineering [2, 33]. Sequential decision making (where the subject needs to make decisions in sequence, possibly with other events or observations occurring between decisions) are of particular theoretical and philosophical interest, because they raise two related questions: how should we choose between uncertain outcomes, and, given a means of doing this, how should we make plans for sequential problems?

Classically, the answer to the first question is to assign probabilities to events and utilities to outcomes, and maximize expected utility [78, 61]. The properties of expected utility lead to convenient backward induction solutions [46, 7, 56]. The behaviour of this solution is well known and exhibits many attractive properties, but is not universally accepted. Some argue that the *independence axiom* (also known as the *sure-thing principle* [61] or *substitutability*), a fundamental assumption underlying expected utility theory, does not correctly mirror human behaviour [1, 18, 34] and disallows apparently reasonable alternative criteria [49, 50]. Others claim that, when confronted by severe uncertainty, it is unreasonable to expect exact probabilities to be available [44, 8], and it may therefore be unreasonable to state a single action as "optimal" [63, 79].

For these reasons, many alternatives to expected utility have been proposed, including prospect theory [34], weighted utility [29], subjectively weighted utility [35], Dempster-Shafer belief functions [16, 68], imprecise probability [81, 79], game-theoretic probability [69], optimism/pessimism [28], info-gap theory [8], and anticipated utility [54]. With such theories, different plausible approaches to solutions of sequential problems give different answers, as observed by, among others, LaValle and Wapman [42], LaValle and Fishburn [41], and Machina [50]. Often, such observations in the literature take the form of counter-examples for a particular theory, but, inspired in particular by Hammond [24] and McClennen [51], we follow a different route, seeking necessary and sufficient conditions for a theory of choice to satisfy a particular behaviour.

With this in mind, we need to find a fairly general language for choice under uncertainty. A natural tool for this is the concept of *choice functions*. These have their roots in social choice theory [55, 5, 66] and are, in their simplest form, functions that map any sets of options to non-empty subsets (the non-empty restriction is called *minimal intelligible choice* by McClennen [51]). The interpretation is that, given a set of options, a subject (or group of subjects) should select an option from the set returned by the choice function.

Of particular use in single-agent decision making under uncertainty are choice functions on *gambles*, which are uncertain rewards—these can be seen as generalizations of random variables. Savage [61] considers acts and gambles to be equivalent: since a particular act gives a particular gamble, choice is simply a matter of choosing between gambles. Similarly, Hammond [24] proposes that acts should be valued by their gambles, and therefore a choice function between acts should be equivalent to a choice function on gambles (a key property that he calls *consequentialism*). Using choice functions on gambles rather than acts is intuitive (since it is the outcome of acts that the subject is interested in) and is a natural extension of maximizing the expected utility of random variables. Further, many theories of choice can easily be represented in the framework of gambles, and so throughout this thesis we will consider, for the most part, uncertainty and preference models based on choice functions on gambles.

Choice functions on gambles give a method for decision making in “static” situations when the subject must choose one act from a set. In the sequential setting, it is less clear how to apply the choice function. One natural way, commonly called the *normal form* solution and the method we investigate the most, is to list all possible strategies the subject can take, and then apply the choice function to this set. This essentially turns a sequential problem into a static one. An alternative approach is to determine the decision that will be taken at the final layer of nodes, eliminate all others, move back to the next layer of decision nodes, and so on. This ensures that at every layer the problem is again a “static” choice between acts, and so the choice function can be applied. This solution method is usually called the *extensive form*. Details of these methods of solution can be found in many books on decision theory, e.g. [46, 56, 45, 11].

Unfortunately, the standard extensive form method is impossible to apply if the choice function does not return a single optimal gamble, or set of equivalent gambles, at every set. Therefore, following Hammond [24], in Section 1.2.4 we introduce a wider class of extensive form solutions that can handle more complicated choice functions. We also consider more general normal form solutions: instead of considering the set of all strategies and then applying the choice function, we simply allow a normal form solution to be a subset of strategies. Although, as mentioned, the normal form solution using a choice function on gambles is of principal interest to us, whenever a result can be proved in wider generality we shall take the opportunity.

Our main goal, then, is to examine the normal form solution induced by a choice function, identifying certain behaviours and finding simple conditions on the choice function for these behaviours to hold. There are two main motivations for doing so. The first is that, should a particular behaviour be judged necessary for rationality, identifying exactly the set of choice functions implying that behaviour allows us either to dismiss all other choice functions, or to dismiss the idea of normal form solutions via choice functions (see for instance [41, 24] for accounts with this goal). The second reason is that it is much more convenient to check a few simple properties on a choice function than to determine whether the behaviours hold for a particular choice function by first principles (since the core proofs are often long-winded and somewhat tedious). Since there are many proposed choice functions whose sequential properties are not known, and presumably there are even more plausible choice functions not yet proposed, having quick methods to determine their behaviour is very useful.

The first behaviour we investigate, in Chapter 2, is *subtree perfectness*. This is related to Hammond's consistency property [24], and named after Selten's concept of subgame perfectness in extensive form games [65]. Subtree perfectness means that a solution of a sequential problem, when restricted to a smaller part of the problem, should agree with the solution of that smaller problem. Simply put, the solution's implications in a sub-problem can be obtained by "snipping off" the rest of the problem and solving the sub-problem separately. This is a very useful property to have, since lacking it could suggest that the solution has some sort of *dynamic inconsistency* [63] and should be regarded with suspicion. However, some authors have defended lack of subtree perfectness as perfectly rational [50, 51]. Nonetheless, a choice function with this property is likely to be more palatable than one without. Unfortunately, we find that the normal form solution is only subtree perfect for a very restricted set of choice functions, mirroring Hammond's results for extensive form solutions [24]. This suggests that there is a strong link between subtree perfectness and the equivalence of the two forms of solution; we investigate this in Section 2.5.

Perhaps because of this link, it has often been suggested that lack of subtree perfectness implies failure of backward induction (or "rolling back" decision trees) [42, 50]. Given the typical extensive form interpretation of backward induction, this suggestion is understandable. Kikuti et al. [38], however, suggested a backward induction algorithm that finds *normal form* solutions. Further, this normal form backward induction can be applied to any choice function on gambles, not just weak orders. In Chapter 3 we formalize this idea and find necessary and sufficient conditions on the choice function for backward induction to work (that is, to give the same solution as applying the choice function directly). Consideration of whether satisfying backward induction but not subtree perfectness is sensible behaviour is given in Section 3.3. This chapter also contains some ideas of generalizations of extensive form backward induction.

Chapters 4 and 5 consider special types of sequential decision problems in which rewards are received after every stage rather than all at the end of the process, such as Markov decision processes [60] or dynamic programming for discrete-time systems [7, 47, 13]. This allows more realistic modelling of problems and more convenient methods of solution. The chapters demonstrate how the general ideas of the previous two chapters can be adapted easily to deal with other models.

Finally, in Chapter 6, we apply the results of the previous chapters to the popular choice functions of a model of uncertainty called *coherent lower previsions* [81, 79]. This uncertainty model is applicable when the subject does not feel able to express exact expectations for all relevant gambles, but can express bounds for them, or, equivalently, can provide a convex set of probability distributions representing her knowledge. As an example, consider the extreme case of complete ignorance about what might happen. If a particular gamble gives a minimum reward of 1 and a maximum reward of 2, then we can say that the value of the gamble is bounded by 1 and 2. The theory can be cast in terms of buying and selling prices for gambles, in the same vein as De Finetti's theory of previsions [14]. Having expressed bounds for the values of whichever gambles she feels comfortable, the subject can use the theory of coherent lower previsions to extend these to find the intervals for the values of other gambles, including conditional values.

Chapter 6 is useful in three ways. First, it provides what is to our knowledge the most comprehensive investigation of sequential decision making using coherent lower previsions. Second, it demonstrates how the theory of the previous chapters works in practice. Finally, Sections 6.4 and 6.5, show that, even when a choice function for an uncertainty model does not in general show a particular behaviour, the standard approaches from the previous chapters can be easily adapted to find the minimal restrictions that must be places on the uncertainty model or decision problem for the required behaviour to be exhibited. Since subtree perfectness in particular is a very difficult property to satisfy in general, it is useful to know under what circumstances it will hold for one's preferred uncertainty model and choice function. For coherent lower previsions it turns out that, while no choice function is subtree perfect in all circumstances, in some relevant special cases subtree perfectness can hold.

For the rest of this chapter, we introduce the necessary definitions and notation for working with the normal and extensive form solutions of decision problems that can be represented as *decision trees* [56, 45, 11].

1.2 Definitions and Notation

This section introduces the concepts and notation required for working with decision trees, choice functions, and gambles. These are used consistently throughout Chapters 2 and 3, and most of Chapter 6 (Sections 6.1 to 6.3, and Section 6.4). Chapter 4 uses some different notation because the decision processes involved use different graphical representations from standard decision trees; the specific notation required for these processes are contained in the relevant sections of that chapter. Chapter 5 makes a few alterations to the standard notation.

1.2.1 Gambles

The decision processes investigated in this thesis involve the subject's making several choices, sequentially, between uncertain outcomes. It is instructive to first consider the simplest case of choice: the subject must choose one action from a set of actions, each of which gives a reward, where the reward received depends on the state of the world and the option chosen. Formally, let Ω be the *possibility space*: the set of all possible states of the world. Elements of the possibility space are called *outcomes*. Let $\mathcal{R}$ be a set of rewards. Rewards represent the possible results that the subject can receive; for example, a free hamburger, or death (despite the name, rewards do not have to be desirable). Although in practice rewards are often expressed in units of either money or utility, we place no restriction on what the elements of $\mathcal{R}$ can be.

In our simple problem, the reward received after having chosen a particular action depends on the state of nature. We define a *gamble* to be a function $X : \Omega \to \mathcal{R}$. The interpretation is that, should $\omega \in \Omega$ be the actual state of the world, the gamble X will give the subject the reward $X(\omega)$. Then the simple decision problem becomes the following: if we must choose one gamble from a set of gambles, which gamble should be chosen?

Some further notation is required. Elements of Ω are usually denoted by ω. Subsets of Ω are called *events* and are usually denoted by A, B, or E. The complement of an event is denoted by $\overline{A}$ and so on. Gambles are usually denoted by X, Y, or Z, and sets of gambles by $\mathcal{X}$ and so on. Rewards are usually denoted by r. Unless otherwise specified, any set or event considered will always be finite and non-empty.

Now consider the following situation: if event A occurs, the subject receives gamble X, otherwise she receives gamble Y. Clearly this is still a gamble, since the reward acquired is determined by the true ω: the reward is $X(\omega)$ if $\omega \in A$ and $Y(\omega)$ otherwise. We use the following notation:

$$AX \oplus \overline{A}Y = \begin{cases} X(\omega) & \text{for } \omega \in A \\ Y(\omega) & \text{for } \omega \in \overline{A} \end{cases}$$

In other words, $\oplus$ is an operator that combines partial maps defined on disjoint domains (where for example AX is $X|_A$, the restriction of X to A). This idea can be extended to any partition $A_1, \ldots, A_n$ and any gambles $X_1, \ldots, X_n$ by

$$\bigoplus_{i=1}^{n} A_i X_i = \begin{cases} X_1(\omega) & \text{for } \omega \in A_1 \\ \quad\vdots & \qquad\vdots \\ X_n(\omega) & \text{for } \omega \in A_n \end{cases}$$

It proves necessary to extend this notation to combinations of sets of gambles. For any partition $A_1, \ldots, A_n$ and sets of gambles $\mathcal{X}_1, \ldots, \mathcal{X}_n$, define

$$\bigoplus_{i=1}^{n} A_i \mathcal{X}_i = \left\{ \bigoplus_{i=1}^{n} A_i X_i \colon X_i \in \mathcal{X}_i \right\}.$$

Finally, the following extension of the $\oplus$ notation is convenient: for any gamble Z, event $A \neq \Omega$, and set of gambles $\mathcal{X}$, define

$$A\mathcal{X} \oplus \overline{A}Z = \{AX \oplus \overline{A}Z \colon X \in \mathcal{X}\}.$$

1.2.2 Choice Functions

Suppose one is given a set of options from which one must be chosen. Ideally, one would be able to determine from any set a uniquely optimal action. In practice, this will be impossible in some situations. Even if one follows classical decision theory

[78, 46, 56], specifying probabilities for all events and utilities for all rewards, at best one only arrives at a weak order (two different actions can have the same expected utility). So even here a uniquely optimal action will not always be available.

It is more reasonable to assume that, for any set of options, one can identify a subset that can be eliminated from consideration. Of course, if one has limited information about the consequences of each action then this subset may be empty. The actions that have not been eliminated can then be reported as *optimal*: any action that remains is a plausible candidate to be picked. So, for any set of options, we can assume that the subject is able to report a subset of unacceptable options, or equivalently a non-empty subset of acceptable options. We could work with either of these subsets, and it proves most useful to use the latter, which corresponds to the concept of a *choice function* from the social choice literature [5, 66]: the subject has a function, opt, that maps sets of options to non-empty subsets.[1]

This concept has been introduced for sets of options of any type, but we are only interested in sets of gambles. So, we suppose that, for any set of gambles, the subject can identify an acceptable subset; that is, she has a *choice function on gambles*. For brevity, and since there are no other types of choice function in this thesis, we simply use *choice function* to refer to these.

Since we deal with sequential decision making, there will be occasions where choices have to be made after some event has been observed. Therefore we consider *conditional choice functions*.

Definition 1.1. *A conditional choice function* opt *is a function that, for any non-empty event A, maps each non-empty finite set $\mathcal{X}$ of gambles to a non-empty subset of that set:*

$$\emptyset \neq \operatorname{opt}(\mathcal{X}|A) \subseteq \mathcal{X}.$$

Note that most uses of choice functions in the literature do not refer to conditioning. All choice functions in this thesis are in fact conditional choice functions, so for

[1] The usual notation in the literature is to represent a choice function by C; we prefer opt to link with work by De Cooman and Troffaes [13], which is very similar to our backward induction schemes.

brevity we omit the word 'conditional', and use the shorthand $\mathrm{opt}(\mathcal{X}) = \mathrm{opt}(\mathcal{X}|\Omega)$.

If the subject provides a choice function, then the simple decision problem of the previous section can be solved: if the subject must choose from $\mathcal{X}$, then the solution is $\mathrm{opt}(\mathcal{X})$. Of course, since this may not return a singleton, the subject must find some way of picking which gamble in $\mathrm{opt}(\mathcal{X})$ she will take. The role of the choice function is to contain all the information her knowledge and preferences can give her about what to do, and if that is insufficient to choose a single option then something more arbitrary is required. One possibility is to use another choice function that always returns a singleton, and apply this to the result of the first choice function. This second choice function is called a *security criterion* by Levi [43, § VII]. Other methods exist (for instance, randomization), but this question of how the subject actually picks what to do in a particular instance is not the focus of our investigation.

So far we have not restricted the choice function in any way, although it is clear that there are many properties that one may wish to enforce. For example, if $\mathrm{opt}(\mathrm{opt}(\mathcal{X})) \neq \mathrm{opt}(\mathcal{X})$, the choice function does not seem to have a sensible interpretation at all. Studying simple one-stage decision problems as introduced in the previous section can yield many plausible restrictions on choice functions. An illuminating investigation of some of these and the arguments behind them can be found in Luce and Raiffa [46, §13.3].

We pursue a similar goal for sequential problems: there are many characteristics that one may want for solutions to a class of sequential decision problems. If the solutions are in some way induced by a choice function, then what properties must the choice function satisfy to display these characteristics? In doing so, we shall not make any general restrictions upon the choice function, no matter how obviously rational they may seem, in order to find the minimal conditions required for a particular type of behaviour. Of course, we are still assuming existence of the choice function (and therefore implicitly the possibility space and the reward set).

To achieve this, we first need some way of solving a sequential decision problem using choice functions. It is not as easy to see how to represent a problem with multiple decision points using gambles as it is when only one decision needs to be made. There are, however, several possibilities for each class of problem. In Chapters 2 and 3 we examine sequential decision problems that are modelled by *decision trees*, which we will now introduce and consider how they may be transformed into sets of gambles.

A Note on Act-State Dependence

This use of a choice function on gambles assumes that taking an act can be considered equivalent to receiving a gamble. This is a sensible assumption only in the case of *act-state independence*, that is, when the action taken does nothing to influence the underlying state of nature. In the main examples we use throughout, this assumption is clearly reasonable, but there are many situations where it would be dubious. For instance, suppose the subject is investing in security for her house. The relevant state of nature would naturally be a successful burglary, and clearly the subject's belief about this event will be influenced by the chosen level of security.

Should the subject be able to specify a probability of successful burglary for each possible level of security, then the optimal solution is easily found: one can find the expected utility of each choice. How this generalizes to arbitrary choice functions (or even particular ones: it is not clear how the generalization of expected utility maximization given in Chapter 6 should be adapted to deal with act-state dependence) is not obvious. It is conceivable that a general approach for arbitrary choice functions and act-state dependence is unachievable. For simplicity, we assume act-state independence throughout, commenting again on this issue only in Chapter 7 (in particular, there we argue that most problems can be reformulated as act-state independent ones, though at the cost of exponentially increasing Ω).

Properties of Choice Functions

In Chapters 2–5 we investigate different types of behaviour in various sequential decision problems. Our principal goal throughout is to identify necessary and sufficient conditions on choice functions for these behaviours to exist, and express these conditions as collections of simple properties. Since there are many such properties in our work, for convenience we have reproduced all of them in Appendix A on page 197.

	rain E_1	no rain E_2
waterproof d_1	10	15
no waterproof d_2	5	20

Table 1.1: Payoff table for the rain and waterproof problem.

1.2.3 Decision Trees

Decision trees are a graphical representation of a type of decision problem, in which the subject makes decisions and observes information sequentially until at some stage she receives a reward and the process ends. A formal definition is not particularly illuminating, so we begin with an example, which is both simple enough to admit easy study, and complex enough to demonstrate all concepts involved in decision trees.

Tomorrow, a subject is going for a walk in the lake district. Tomorrow, it may rain (E_1), or not (E_2). The subject can either take a waterproof (d_1), or not (d_2). But the subject may also choose to buy today's newspaper to learn about tomorrow's weather forecast (d_S), or not ($d_{\overline{S}}$), before leaving for the lake district. For the sake of simplicity, we assume that the forecast can have either two outcomes: predicting rain (S_1), or not (S_2). The possibility space for this problem therefore has four elements: "the newspaper predicts no rain, and it rains" and so on.

The reward space has eight elements, for instance "it rains, the subject does not have the waterproof, and she paid for the newspaper". To simplify this example, let the rewards be in utiles. The utility of each combination, if the subject does not buy the newspaper, is summarized in Table 1.1. If the subject buys the newspaper, then c utiles are subtracted from the utilities.

The decision tree corresponding to this example is shown in Figure 1.1. Decision nodes are depicted by squares, and chance nodes are depicted by circles. From each node, a number of branches emerge. For decision nodes, each branch corresponds to a decision; for chance nodes, each branch corresponds to an event. For each chance node, the events which emerge form a partition of the possibility space: at least one

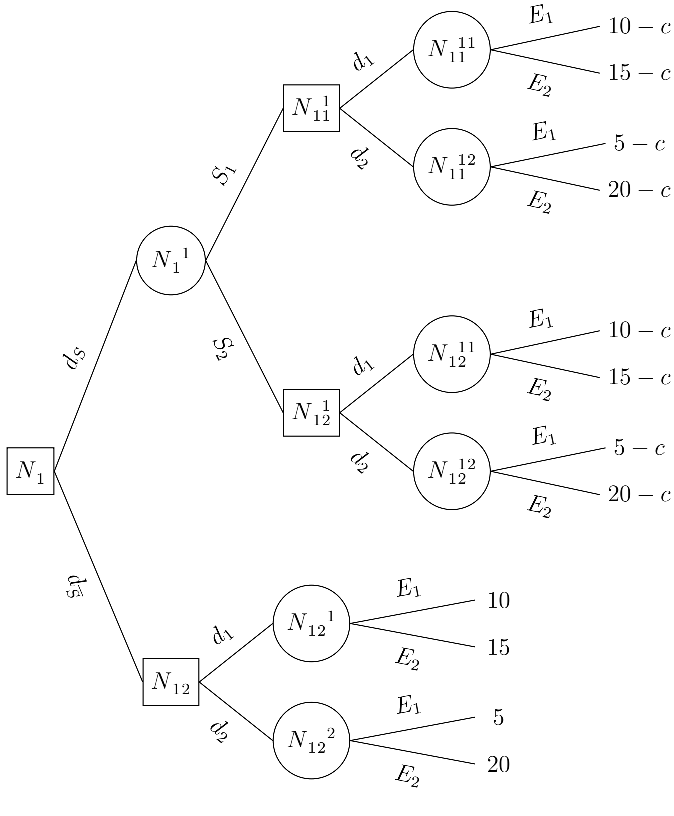

Figure 1.1: A decision tree for walking in the lake district.

of the events must obtain, and no two can obtain simultaneously.

In the lake district problem, the subject is first confronted with the decision to buy the newspaper or not, hence the tree starts off with a decision node. If the subject buys the newspaper (d_S), then it can inform him about tomorrow's weather forecast. Thus, the chance node following the subject's decision d_S has two branches, forecasting rain (S_1), or no rain (S_2). Next, when the subject leaves for the lake district, she can take either her waterproof with her (d_1) or not (d_2), hence the decision node following S_1. Finally, during the walk, it can either rain (E_1) or not (E_2), which is depicted by a chance node for each possible combination of events and decisions preceding the actual walk.

So, each path in a decision tree corresponds to a particular sequence of decisions and events. The payoffs resulting from each such sequence is put at the right end of the tree.

Precise definitions of decision trees vary slightly in the literature. Of particular interest to us in Chapter 2 is the work of Hammond, who gives a formal definition of a decision tree with some important differences [24, p. 31]. Events at Hammond's chance nodes form a partition of the set of all states of nature still possible upon reaching the chance node; in other words, they partition the intersection of all events leading up to the chance node. A further difference to note is that our chance nodes are called *natural nodes* by Hammond. Hammond's chance nodes involve probabilities and do not appear in our decision trees.

We do not find a formal definition of our decision trees useful for our purposes, but readers who disagree or who want to compare with Hammond's definition can find one in Appendix B.1. We do, however, find that the following notation is very useful for complicated definitions and proofs.

Decision trees can be seen as combinations of smaller decision trees: for instance, in the lake district example, one could draw the subtree corresponding to buying the newspaper, and also draw the subtree corresponding to making an immediate decision. The decision tree for the full problem is then formed by joining these two subtrees at a decision node.

Hence, we can represent a decision tree by its subtrees and the type of its root node. Let $T_1, \ldots, T_n$ be decision trees and $E_1, \ldots, E_n$ be a partition of the possibility space. If T is rooted at a decision node, we write

$$T = \bigsqcup_{i=1}^{n} T_i.$$

If T is formed by combining the trees at a chance node, with subtree T_i being connected by event E_i, we write

$$T = \bigodot_{i=1}^{n} E_i T_i.$$

For instance, for the tree of Fig. 1.1, we write

$$(S_1(T_1 \sqcup T_2) \odot S_2(T_1 \sqcup T_2)) \sqcup (U_1 \sqcup U_2)$$

with

$$T_1 = E_1(10 - c) \odot E_2(15 - c) \qquad U_1 = E_1 10 \odot E_2 15$$

$$T_2 = E_1(5 - c) \odot E_2(20 - c) \qquad U_2 = E_1 5 \odot E_2 20$$

Admittedly, this notation is not really useful here, particularly when compared with the tree itself, but it becomes very convenient when considering recursive definitions and proofs, and when dealing with general trees rather than specific ones we can draw.

For some proofs we need to consider decision trees formed by adding a decision node in front of a tree T, that is, a decision tree whose root is a decision node where there is only one option. This is denoted by $\sqcup T$.

The decision tree notation extends easily to sets of trees. Consider all possible ways that sets of decisions trees $\mathcal{T}_1, \ldots, \mathcal{T}_n$ can be combined. Combining at a decision node is represented by

$$\bigsqcup_{i=1}^{n} \mathcal{T}_i = \left\{ \bigsqcup_{i=1}^{n} T_i \colon T_i \in \mathcal{T}_i \right\}.$$

For any partition $E_1, \ldots, E_n$, combination at a chance node is represented by

$$\bigodot_{i=1}^{n} E_i \mathcal{T}_i = \left\{ \bigodot_{i=1}^{n} E_i T_i \colon T_i \in \mathcal{T}_i \right\}.$$

We often consider particular types of subtrees of larger trees, obtained by "snipping off" everything before a certain node in the full tree.

Definition 1.2. *A subtree of a tree T obtained by removal of all non-descendants of a particular node N, but retaining N, is called* the subtree of T at N *and is denoted by* $\mathrm{st}_N(T)$.[2]

These subtrees are called *continuation trees* by Hammond [24]. The definition extends to sets of trees.

Definition 1.3. *If $\mathcal{T}$ is a set of decision trees and N a node, then*

$$\mathrm{st}_N(\mathcal{T}) = \{\mathrm{st}_N(T) \colon T \in \mathcal{T} \text{ and } N \text{ in } T\}.$$

For subtrees, it is important to know the events that were observed in the past. Two subtrees with the same configuration of nodes and arcs may have different preceding events, and should be treated differently. Therefore we associate with every decision tree T an event $\mathrm{ev}(T)$ representing the intersection of all the events on chance arcs that have preceded T. Hammond [24, p. 27] denotes these events by $S(n)$.

Since the only restriction on the events at each chance node is that they must form a partition of the possibility space, there is no guarantee yet that all paths through the tree are possible. It could be that due to a modelling oversight a tree is drawn such that a chance arc is impossible given the preceding events. This will cause problems when we attempt to use conditional choice functions at this point, since it would involve conditioning on the empty set, and so we choose to work only with trees that avoid this issue. If one's decision tree has an impossible arc, it can be amended by removing the arc and changing the partition at this chance node. For example, if event A_1 has been observed, and the partition at a following chance node is A_1, A_2, A_3, one could change the partition to $A_1 \cup A_2, A_3$. Of course this is an arbitrary choice, but it seems sensible that one's method of solution should not be influenced by which arcs the impossible outcomes are redistributed to. Therefore,

[2]Formally, if $\mathcal{N}$ is the set of nodes comprising N and all its descendents, then $\mathrm{st}_N(T)$ is the subgraph of T induced by $\mathcal{N}$ (the subgraph of T induced by vertex set $\mathcal{N}$ is the graph U whose vertex set is $\mathcal{N}$ and where, for each pair of vertices $N_1, N_2 \in \mathcal{N}$, N_1N_2 is an arc of U if and only if it is an arc of T).

if the arbitrary choice were to matter, the method of solution could be considered questionable.

We say that a decision tree without impossible branches is *consistent*.

Definition 1.4. *A decision tree T is called consistent if, for every node N of T,*

$$\mathrm{ev}(\mathrm{st}_N(T)) \neq \emptyset.$$

Clearly, if a decision tree T is consistent, then for any node N in T, $\mathrm{st}_N(T)$ is also consistent. One might consider the above definition slightly too restrictive: suppose that an arc leading to a reward node from a chance node causes inconsistency in a tree, but the rest of the tree is consistent. We would never in practice need to apply a choice function at the inconsistent point (since there is no choice once a reward node is reached), and so we would never have to worry about conditioning on the empty set in this tree. It seems that this special type of inconsistent tree would be reasonable to consider. This is true, but since correcting an inconsistent tree should not be a problem anyway, particularly in this extreme situation (eliminate the inconsistent reward node!), it does not seem worth making the exception.

In many proofs we work with the children (the immediate successors) of the root node of a tree. We refer to this set of nodes as $\mathrm{ch}(T)$.

1.2.4 Solving Decision Trees: Extensive and Normal Forms

The traditional methods of solving decision trees are to use either the classical extensive form or the classical normal form. These rely on having probabilities for all events and utilities for all rewards. Let us consider how decision trees can be solved in this simple situation.

The classical extensive form solution relies on backward induction, calculating expected utilities in subtrees and eliminating branches by maximization. Let us use the decision tree in Fig. 1.1 as an example. Suppose that $p(S_1) = 0.6$, $p(E_1|S_1) = 0.7$, and $p(E_1|S_2) = 0.2$, so $p(E_1) = 0.5$. We first calculate the expected utility of the final chance nodes. For example, the expected utility at N_{11}^{11} is

$$0.7(10 - c) + 0.3(15 - c) = 11.5 - c,$$

and the expected utility at ${N_{11}}^{12}$ is $9.5 - c$.

We now see that at ${N_{11}}^{1}$ it is better to choose decision d_1. We then *replace* ${N_{11}}^{1}$ and its subtree with the expected utility of ${N_{11}}^{11}$: $11.5 - c$. Also follow this procedure for ${N_{12}}^{1}$ and N_{12}, and the tree has been reduced by a stage. We find that d_2 is optimal at ${N_{12}}^{1}$ with value $17 - c$, and at N_{12} both decisions are optimal with value 12.5.

Next, take expected utility at ${N_1}^1$, which is $0.6(11.5-c)+0.4(17-c) = 13.7-c$. At N_1, we therefore take decision d_S if $c \leq 1.2$ and $d_{\overline{S}}$ if $c \geq 1.2$. So, if $c < 1.2$ then choose to buy the newspaper, if the newspaper predicts rain then take the waterproof and if the newspaper does not predict rain then do not take the waterproof. If $c > 1.2$, then do not buy the newspaper, and then arbitrarily decide whether or not to take the waterproof. If $c = 1.2$, then arbitrarily decide whether or not to buy the newspaper.

The alternative method of solution is the classical normal form. This involves listing all possible strategies, such as "buy the newspaper, and take the waterproof only if the newspaper does not predict rain". There are six such strategies in this decision tree. The expected utility of each strategy can be calculated, since the outcome of a strategy depends only on the true state of nature.

Computing these expectations is tedious and left as an exercise, but the result is unsurprising. If $c < 1.2$, the optimal strategy is to buy the newspaper and take the waterproof only if rain is predicted. If $c > 1.2$, there are two optimal strategies: do not buy the newspaper and take the waterproof; do not buy the newspaper and do not take the waterproof. If $c = 1.2$, all three previous strategies are optimal. In situations where more than one strategy is optimal, the subject then just picks one to carry out.

We see in this example that it does not really matter which method is used. When $c > 1.2$, the normal form solution and the extensive form solution are identical. When $c \leq 1.2$, they differ only in the point where an arbitrary choice has to be made. For instance, for $c = 1.2$, using the normal form the subject has to pick today whether to buy the newspaper and follow its advice, to reject the newspaper and take the waterproof tomorrow, or to reject to newspaper and go without the waterproof tomorrow. In the extensive form, the subject needs only pick whether to take the waterproof or not when tomorrow actually comes, and only needs to do this if she rejected the newspaper today. But the actions that are judged optimal are all the same for both methods. This property holds for maximizing expected utility in all decision trees [56], when probabilities are non-zero [24, p. 44].

For other choice functions, there is no reason to assume that the two methods will give the same optimal acts, and indeed it is not even obvious how the extensive form method should be generalized to choice functions that do not correspond to a total preorder. Further, they represent two possible ways of solving decision trees, which make good sense for expected utility, but plenty of other possibilities exist. Therefore we wish to extend the concepts of normal and extensive form, so that they can be used to describe solutions for general choice functions and even solutions that do not require choice functions, while retaining what we consider to be their core feature: the point at which decisions are made.

For extensive form solutions, decisions are made when decision nodes are reached. They involve selecting some arcs at decision nodes that are unacceptable and removing them, but not stating which remaining branch will be chosen until that node is actually reached. An extensive form solution can be represented by a decision tree. For normal form solutions, all decisions are made initially; however, we do not require the solution to include only one strategy, rather it can be many strategies from which the subject can pick a single one to enact. Normal form solutions can then be represented by a set of decision trees.

Definition 1.5. *An* extensive form solution *of a decision tree T is a decision tree obtained by removing, at each decision node of T, some (possibly none, but not all) of the decision arcs. All nodes following deleted arcs are also removed.*

For example, in the lake district example, the expected utility extensive form solution for $c = 1.2$ that we calculated earlier is represented in Fig. 1.2. The interpretation of the extensive form solution is as follows: upon reaching a decision node in T, the subject picks one of the decision arcs in the extensive form solution and follows it. The subject only needs to decide which arc to follow at a decision node upon actually reaching that node.

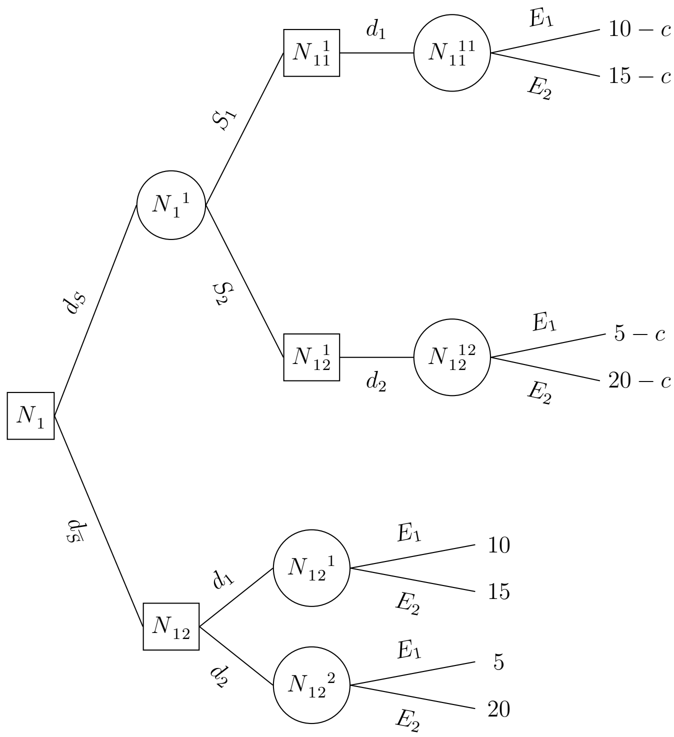

Figure 1.2: Extensive form solution to Fig. 1.1 with $c = 1.2$.

Definition 1.6. *A* normal form decision *is an extensive form solution with precisely one arc at each decision node.*

Normal form decisions have also been called *strategies*, *pure strategies* [46, p. 51], *plans* [51, §6.3], and *policies* [30].

Definition 1.7. *The set of all normal form decisions of a tree T is denoted by* $\mathrm{nfd}(T)$*. The set of all normal form decisions of a set of trees $\mathcal{T}$ is given by*

$$\mathrm{nfd}(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} \mathrm{nfd}(T).$$

With a normal form decision, the subject's actions are uniquely determined in every eventuality. If a subject could specify for any decision tree a unique normal form decision that they considered optimal (or equivalently they specified that their extensive form solution was a normal form decision) this would be ideal, and the distinction between the normal form and the extensive form would be immaterial. But, as with choosing from sets of gambles, this will in general be impossible (as we

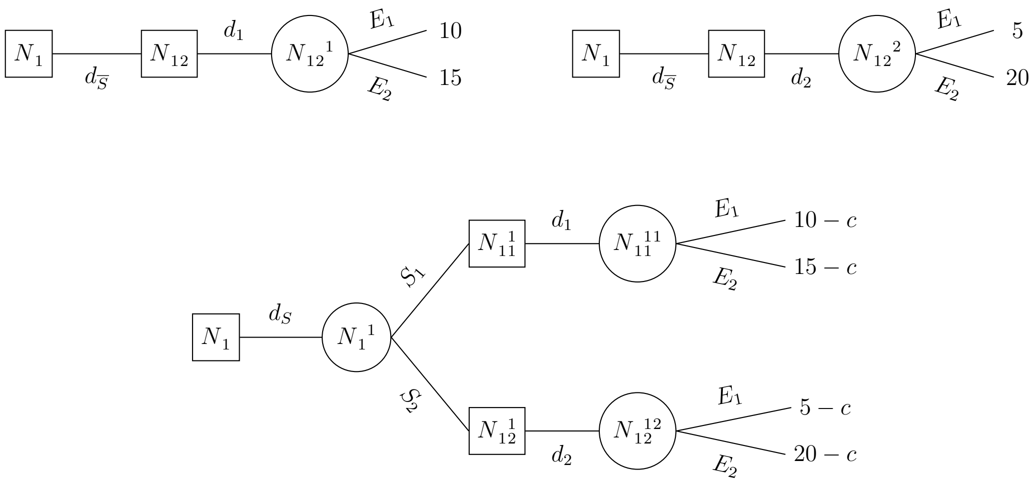

Figure 1.3: Normal form solution to Fig. 1.1 with $c = 1.2$.

saw even in the case of maximizing expected utility). Therefore we introduce the concept of normal form solutions.

Definition 1.8. *A* normal form solution *of a decision tree T is a non-empty subset of* $\mathrm{nfd}(T)$.

For example in the lake district example, for $c = 1.2$, the expected utility normal form solution that we calculated earlier is represented in Fig. 1.3. The interpretation of the normal form solution is that the subject picks one of the normal form decisions and follows the specified actions. As with choice functions, the idea is that specifying the normal form solution uses all the subject's knowledge and preferences, and the picking of the normal form decision must be arbitrary.

Note that in these definitions we have abandoned the popular link between backward induction and the extensive form. This is because there are methods of solution that have extensive form properties yet have nothing to do with backward induction, and that, as we shall see in Section 3.1, backward induction can often be employed to find normal form solutions.

Whenever normal form and extensive form solutions are introduced, it is common to consider whether they are equivalent. In the simplest case when the normal form solution is a singleton this is of course trivial to check. In general, it is not even clear what equivalence of these forms even means. We defer consideration of this issue to Section 2.5.

These two forms of solutions are by no means the only possibilities. For example, McClennen [51] considers the following form of solution (the name of the form is ours).

Definition 1.9. *A* dynamic normal form *solution of a decision tree T is a collection of subsets of* $\mathrm{nfd}(\mathrm{st}_N(T))$ *for each decision node N of T (including the root).*

So, a dynamic normal form solution specifies a set of normal form decisions *at every node in the tree.* McClennen is not clear about the interpretation of this type of solution. Apparently, the subject chooses a normal form decision each time she reaches a decision node. But why choose a normal form decision if the next time she reaches a decision node she can choose a different one? Note that our normal form solutions could be seen as special cases of dynamic normal form solutions, in which the acceptable plans at a particular node are simply the restriction, to the node in question, of the acceptable plans at the root node. McClennen calls this a *dynamically consistent* solution and seems to suggest [51, p. 116] that restricting attention to dynamically consistent solutions is sensible, so the confusion about the interpretation of solutions failing this property is not so important.

Extensive form and normal form solutions as defined here represent solutions to a particular decision tree. Usually we consider methods that can be applied to any decision tree with a suitable uncertaintly model, such as maximizing expected utility when probabilities are available. We describe these by means of extensive form and normal form operators.

Definition 1.10. *An* extensive form operator ext *is a function mapping decision trees to extensive form solutions. A* normal form operator norm *is a function mapping decision trees to normal form solutions.*

Clearly the two classical methods of maximizing expected utility are extensive form and normal form operators respectively, which is reassuring. Hammond [24, p. 28] calls extensive form solutions *behaviour norms.*

1.2.5 Normal Form Operator Induced by a Choice Function

As mentioned previously, there are many possible ways of defining extensive form and normal form operators using choice functions, some of which are not particularly obvious. In this section we introduce possibly the simplest approach, which is popular in the literature. Much of our investigation in following chapters is based upon the behaviour of this operator, or very similar alternatives. It generalizes the classical normal form operator based on expected utility. The basic idea is simple: reduce a decision tree to a set of gambles, apply a choice function to find the optimal subset, and then transform each optimal gamble to a normal form decision, thereby obtaining an optimal subset of normal form decisions, hence, a normal form solution.

Recall that a normal form decision prescribes the subject's actions, so, once one has been chosen, the reward the subject receives is determined entirely by the events that obtain. In other words, a normal form decision has a corresponding gamble, which we call its *normal form gamble*. We denote the set of all normal form gambles associated with a decision tree T by $\operatorname{gamb}(T)$. So, gamb applied to a normal form decision returns a singleton containing its normal form gamble, and for any other tree

$$\operatorname{gamb}(T) = \bigcup_{U \in \operatorname{nfd}(T)} \operatorname{gamb}(U). \tag{1.1}$$

Let us explain how to find the gamble corresponding to a normal form decision, using Fig. 1.1 as an example, with $c = 1$. Instead of looking at the full tree, for simplicity let us first consider the subtree with root at N_{11}^{1}. The only two normal form decisions in this subtree are simply d_1 and d_2. The former gives reward 9 utiles if $\omega \in E_1$ and 14 utiles if $\omega \in E_2$, which corresponds to a gamble

$$E_1 9 \oplus E_2 14. \tag{1.2}$$

In the above expression, the $\oplus$ operator combines partial maps defined on disjoint domains (i.e. the constant partial map $E_1 9$ defined on E_1, and the constant partial map $E_2 14$ defined on E_2).

Now consider the subtree with root at $N_1{}^1$, and in particular the normal form decision 'd_1 if S_1 and d_2 if S_2'. This gives reward 9 if $\omega \in S_1 \cap E_1$, reward 14 if

	ω_1	ω_2	ω_3	ω_4
$E_1 9 \oplus E_2 14$	9	9	14	14
$S_1 (E_1 9 \oplus E_2 14) \oplus S_2 (E_1 4 \oplus E_2 19)$	9	4	14	19

Table 1.2: Example of normal form gambles.

$\omega \in S_1 \cap E_2$, and so on. The corresponding gamble is

$$(S_1 \cap E_1)9 \oplus (S_1 \cap E_2)14 \oplus (S_2 \cap E_1)4 \oplus (S_2 \cap E_2)19,$$

or briefly, if we omit '$\cap$' and employ distributivity,

$$S_1 (E_1 9 \oplus E_2 14) \oplus S_2 (E_1 4 \oplus E_2 19), \tag{1.3}$$

where multiplication with an event is now understood to correspond to restriction, i.e., 9 is a constant map on Ω, $E_1 9$ is a constant map restricted to E_1, and $S_1(E_1 9)$ is obtained from $E_1 9$ by further restriction to $E_1 \cap S_1$. For illustration, we tabulate the values of some normal form gambles in Table 1.2, where $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$, $E_1 = \{\omega_1, \omega_2\}$, and $S_1 = \{\omega_1, \omega_3\}$.

Observe that the gamble in Eq. 1.3 incorporates the gamble in Eq. (1.2) from N_{11}^{1}. Relationships between sets of normal form gambles for different subtrees allow a very convenient recursive definition of the gamb operator, given next. First, we extend $\oplus$ to sets of gambles.

Definition 1.11. *For any events E_1, ..., E_n which form a partition, and any finite family of sets of gambles $\mathcal{X}_1$, ..., $\mathcal{X}_n$, we define the following set of gambles:*

$$\bigoplus_{i=1}^{n} E_i \mathcal{X}_i = \left\{ \bigoplus_{i=1}^{n} E_i X_i \colon X_i \in \mathcal{X}_i \right\}. \tag{1.4}$$

An equivalent form of Eq. (1.4) that is employed in some proofs is

$$\bigoplus_{i=1}^{n} E_i \mathcal{X}_i = \bigcup_{X_1 \in \mathcal{X}_1} \cdots \bigcup_{X_n \in \mathcal{X}_n} \left\{ \bigoplus_{i=1}^{n} E_i X_i \right\}. \tag{1.5}$$

With these tools at hand, we can easily define the set of all normal form gambles associated with a decision tree:

Definition 1.12. *With any decision tree T, we associate a set of gambles* $\operatorname{gamb}(T)$*, recursively defined through:*

- *If a tree T consists of only a leaf with reward $r \in \mathcal{R}$, then*

$$\operatorname{gamb}(T) = \{r\}. \tag{1.6a}$$

- *If a tree T has a chance node as root, that is, $T = \bigodot_{i=1}^{n} E_i T_i$, then*

$$\operatorname{gamb}\left(\bigodot_{i=1}^{n} E_i T_i\right) = \bigoplus_{i=1}^{n} E_i \operatorname{gamb}(T_i). \tag{1.6b}$$

- *If a tree T has a decision node as root, that is, if $T = \bigsqcup_{i=1}^{n} T_i$, then*

$$\operatorname{gamb}\left(\bigsqcup_{i=1}^{n} T_i\right) = \bigcup_{i=1}^{n} \operatorname{gamb}(T_i). \tag{1.6c}$$

In Hammond's notation, $\operatorname{gamb}(\operatorname{st}_N(T))$ is denoted by $F(T, n)$. McClennen writes $G(T)$ for $\operatorname{gamb}(T)$. Extension of gamb to apply to sets of trees is natural and easy.

Definition 1.13. *For any set $\mathcal{T}$ of decision trees,*

$$\operatorname{gamb}(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} \operatorname{gamb}(T)$$

It should be obvious from Definition 1.12 that the recursive definition gives the correct gamble for a normal form decision. Lemma B.2 provides the less obvious result that it satisfies Eq. (1.1). The recursive definition works for sets as well:

$$\operatorname{gamb}\left(\bigodot_{i=1}^{n} E_i \mathcal{T}_i\right) = \bigoplus_{i=1}^{n} E_i \operatorname{gamb}(\mathcal{T}_i), \tag{1.7}$$

and

$$\operatorname{gamb}\left(\bigsqcup_{i=1}^{n} \mathcal{T}_i\right) = \bigcup_{i=1}^{n} \operatorname{gamb}(\mathcal{T}_i). \tag{1.8}$$

Most decision problems can be represented by several decision trees. This suggests the following definition (see for instance [41, 50]):

Definition 1.14. *Two decision trees T_1 and T_2 are called* strategically equivalent *if* $\operatorname{gamb}(T_1) = \operatorname{gamb}(T_2)$.

Strategically equivalent trees are called *consequentially equivalent* trees by Hammond [24, p. 38]. One could consider a sensible addition to this definition, namely to require that $\mathrm{ev}(T_1) = \mathrm{ev}(T_2)$, since only then would it model the same problem. This is widely ignored in the literature, so we avoid this addition.

The function gamb allows us to transform a decision tree to a set of gambles. A choice function opt then allows us to map this set of gambles to an optimal subset. Mapping back to normal form decisions is easy: the optimal normal form decisions are all elements of $\mathrm{nfd}(T)$ whose normal form gamble is an element of the optimal set of gambles.

Definition 1.15. *For a choice function* opt*, the* normal form operator induced by opt *is defined for any decision trees T with* $\mathrm{ev}(T) \neq \emptyset$ *by*

$$\mathrm{norm}_{\mathrm{opt}}(T) = \{U \in \mathrm{nfd}(T)\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))\} \tag{1.9}$$

Of course, since U is always a normal form decision, $\mathrm{gamb}(U)$ is always a singleton in this definition. In particular, the following equality holds,

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)), \tag{1.10}$$

which will be used extensively further on.

Note that, although $\mathrm{norm}_{\mathrm{opt}}$ is applied to trees, it really depends only on the set of normal form gambles associated with the tree. Hence, regardless of the properties of opt, the operator $\mathrm{norm}_{\mathrm{opt}}$ will respect strategic equivalence:

Theorem 1.16. *If T_1 and T_2 are strategically equivalent, then*

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_2))$$

whenever $\mathrm{ev}(T_1) = \mathrm{ev}(T_2) \neq \emptyset$*.*

This is of course an attractive property, as a decision tree is a graphical representation of a problem, and there may be several strategically equivalent trees that are plausible representations of the same problem. The above theorem guarantees that our solution is independent of the particular representation we use.

In the next two chapters we investigate how the properties of opt influence the behaviour of $\mathrm{norm}_{\mathrm{opt}}$. To do this, we need to know when a set of gambles $\mathcal{X}$ can be represented by consistent decision trees conditional on an event A. If there is no consistent decision tree with $\mathrm{ev}(T) = A$ and $\mathrm{gamb}(T) = \mathcal{X}$ then the properties of $\mathrm{opt}(\mathcal{X}|A)$ are irrelevant.

Definition 1.17. *A set of gambles $\mathcal{X}$ is called A-*consistent *if there exists a consistent decision tree T with $\mathrm{ev}(T) = A$ and $\mathrm{gamb}(T) = \mathcal{X}$.*

Of course, it is always possible to find a decision tree with $\mathcal{X}$ as its normal form gambles, so it is only the consistency of the tree that needs to be checked. We now show how to deduce that $\mathcal{X}$ is A-consistent without needing reference to trees.

Theorem 1.18. *For a non-empty event A, a set of gambles $\mathcal{X}$ is A-consistent if and only if, for every $r \in \mathcal{R}$ and every $X \in \mathcal{X}$ such that $X^{-1}(r) \neq \emptyset$, it holds that $X^{-1}(r) \cap A \neq \emptyset$. In other words, the image of X equals the image of AX: $X[A] = X[\Omega]$.*

Proof. "only if". Let T be a decision tree with $\mathrm{gamb}(T) = \mathcal{X}$ and $\mathrm{ev}(T) = A$, and suppose that there is $X \in \mathcal{X}$ such that $X[A] \neq X[\Omega]$, so there exists $r \in \mathcal{R}$ such that $X^{-1}(r) \cap A = \emptyset$. We know that there is a reward node N in T corresponding to r. By definition, $\mathrm{ev}(\mathrm{st}_N(T)) \subseteq A \cap X^{-1}(r)$: if $\mathrm{ev}(T) = \Omega$, then $\mathrm{ev}(\mathrm{st}_N(T)) \subseteq X^{-1}(r)$. Since $A \cap X^{-1}(r) = \emptyset$, T is not consistent. Therefore if there is a consistent decision tree T with $\mathrm{gamb}(T) = \mathcal{X}$ and $\mathrm{ev}(T) = A$, then $X[A] = X[\Omega]$.

"if". Suppose that for every $r \in \mathcal{R}$ and every $X \in \mathcal{X}$ such that $X^{-1}(r) \neq \emptyset$, it holds that $X^{-1}(r) \cap A \neq \emptyset$. Consider the decision tree

$$T = \bigsqcup_{X \in \mathcal{X}} \bigodot_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r) r \tag{1.11}$$

with $\mathrm{ev}(T) = A$. The notation may mask the simplicity of the tree: for each $X \in \mathcal{X}$ it is obvious how to draw a trivial corresponding decision tree with a chance node as the root and all other nodes being reward leaves. Then, combine all these trees at a decision node. This constructs precisely T.

Let $N(X)$ denote the chance node of T associated with X, and let $N(X, r)$ denote the reward node of T associated with X and r (of course $N(X, r)$ only exists for $X^{-1}(r) \neq \emptyset$, by definition of T).

Clearly, T is consistent, because $\text{ev}(\text{st}_{N(X)}(T)) = \text{ev}(T) = A \neq \emptyset$ and

$$\text{ev}(\text{st}_{N(X,r)}(T)) = \text{ev}(\text{st}_{N(X)}(T)) \cap X^{-1}(r) = A \cap X^{-1}(r) \neq \emptyset$$

by assumption, and

$$\begin{aligned} \text{gamb}(T) &= \bigcup_{X \in \mathcal{X}} \text{gamb}\left(\bigodot_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r) r \right) \\ &= \bigcup_{X \in \mathcal{X}} \left\{ \bigoplus_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r) r \right\} = \bigcup_{X \in \mathcal{X}} \{X\} = \mathcal{X} \end{aligned}$$

which establishes that Definition 1.17 is satisfied. □

Chapter 2

Subtree Perfectness For Normal Form Operators

In this chapter we briefly outline Selten's concept of subgame perfectness for multi-agent sequential games [65], and extend the concept to decision trees, in particular to normal form operators induced by choice functions. It turns out that subtree perfectness for normal form operators is closely linked to Hammond's (extensive form) consequentialist theory [24]. The question of normal form and extensive form equivalence is also easily answered when subtree perfectness is satisfied.

2.1 Definition of Subtree Perfectness

Selten [65] introduced a concept called *subgame perfectness* for multi-agent sequential games. Since game theory is not our focus here, we do not go into detail, mentioning it only for motivation of our term and concept. An equilibrium in a sequential game is subgame perfect if it is a Nash equilibrium for every subgame of the full game. In other words, if the subject were playing any smaller game that is embedded in the larger game, then there would be an equilibrium strategy exactly given by the restriction of the full subgame perfect equilibrium strategy to this smaller game.

In the idea of a single-agent sequential decision problem, we call the corresponding concept *subtree* perfectness. Subtree perfectness (or rather, the lack of it) is best illustrated by an example. Although our focus is on normal form operators for the most part, subtree perfectness is also easier to understand for extensive form operators. Suppose we are applying an extensive form operator to the tree T in Fig. 1.1. This operator will delete some (possibly none) of the decision arcs at $N = N_{1}{}_{1}^{1}$. Let us suppose that the arc d_1 to $N_{1}{}_{1}^{12}$ is deleted. Now consider a second decision tree T' that is identical to T, except that $N_{1}{}_{2}{}^{2}$ is not present (if the subject does not buy the newspaper, then for some reason she is forced to take the waterproof). N appears in this tree as well. Suppose that the extensive form operator does not remove any of the arcs at N for this new tree. So, we have a situation where $\mathrm{st}_N(T) = \mathrm{st}_N(T')$ but $\mathrm{st}_N(\mathrm{ext}(T)) \neq \mathrm{st}_N(\mathrm{ext}(T'))$. In other words, behaviour in a subtree depends on the larger tree in which the subtree is embedded. We say that ext lacks subtree perfectness. Hammond [24, p. 34] has an identical property to subtree perfectness for behaviour norms (extensive form operators) only, that he calls *consistency*.

It may not be immediately clear that this concept is analagous to subgame perfectness as introduced by Selten. Recall that subgame perfectness states that an equilibrium point (of the full game) restricted to a subgame is an equilibrium point of that subgame. The corresponding idea for an extensive form operator would be that the extensive form solution, when restricted to a subtree, is the extensive form solution of that subtree. This is how we choose to define subtree perfectness; a brief examination of the following definition should convince one that this approach is equivalent to the informal definition in terms of embedding that was given above.

Definition 2.1. *An extensive form operator* ext *is called* subtree perfect *if for every consistent decision tree T and every node N such that N is in* $\mathrm{ext}(T)$*,*

$$\mathrm{st}_N(\mathrm{ext}(T)) = \mathrm{ext}(\mathrm{st}_N(T)).$$

An normal form operator norm *is called* subtree perfect *if for every consistent decision tree T and every node N such that N is in at least one element of* $\mathrm{norm}(T)$

$$\mathrm{st}_N(\mathrm{norm}(T)) = \mathrm{norm}(\mathrm{st}_N(T)).$$

Observe that $\mathrm{norm}(T)$ *is a set of trees so it is Definition 1.3 being applied on the left and Definition 1.2 being applied on the right.*

$$
\begin{tikzcd}
T \arrow[r, "\text{restrict}"] \arrow[dd, "\text{optimize}"'] & \mathrm{st}_N(T) \arrow[d, "\text{optimize}"] \\
 & \mathrm{ext}(\mathrm{st}_N(T)) \arrow[d, equal, "\text{if } N \text{ in } \mathrm{ext}(T)"] \\
\mathrm{ext}(T) \arrow[r, "\text{restrict}"'] & \mathrm{st}_N(\mathrm{ext}(T))
\end{tikzcd}
$$

Figure 2.1: For a subtree perfect extensive form operator, optimisation and restriction commute.

In other words, for a subtree perfect operator, it does not matter whether we first restrict our attention to a subtree at a particular node N and then optimize this subtree, or first optimize, and only then look at the resulting subtree at a particular node N: roughly speaking, subtree perfectness means that optimization and restriction commute, as in Fig. 2.1 for an extensive form operator. If the operator lacks subtree perfectness, $\mathrm{st}_N(\mathrm{ext}(T))$ can differ from $\mathrm{ext}(\mathrm{st}_N(T))$ for some decision trees T and nodes N in $\mathrm{ext}(T)$. The corresponding diagram for normal form operators is similar.

Note that the definition for subtree perfectness requires the node N to appear somewhere in the solution. In other words, we do not care how the operator would behave in subtrees that the subject would never choose to reach.

So, we have two different interpretations of subtree perfectness, each with logical motivation. The interpretation from Definition 2.1 states that the global solution is built from local solutions of smaller problems. This is inuitively attractive on both computational and philosophical grounds. The interpretation of the example, that behaviour in a subtree does not depend on the tree in which it is embedded, is appealing because one never has to worry about what larger system a problem is part of. Suppose for instance one is carrying out a policy in a decision tree, and upon reaching a decision node some new option, not in the original tree, presents itself. The natural response is just to solve the local problem. If subtree perfectness holds

	A	$\overline{A}$
X	-1	-1
Y	-2	2
Z	0	0

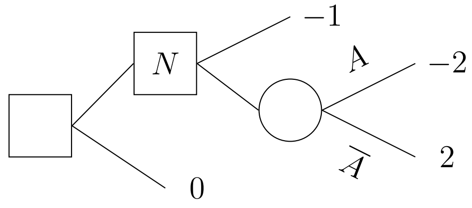

Figure 2.2: Decision tree for Example 2.2.

then this is unambiguously the correct approach, but without subtree perfectness, it may be necessary to consider the whole tree, including the consequences of events that are now impossible. It is hard to criticise someone who finds this unpalatable.

The extensive form operator corresponding to the usual backward induction using expected utility (as briefly described in Section 1.2.3) is well known to be subtree perfect, provided probabilities at chance nodes are non-zero. Also, the usual normal form operator corresponding to maximizing expected utility over all normal form decisions is subtree perfect, because it is equivalent to the extensive form operator.

Before we examine subtree perfectness in more detail, we give an example of a choice function that lacks the property: pointwise dominance.

Example 2.2. *Let T be the decision tree in Fig. 2.2, where X, Y, and Z are its normal form gambles. Under pointwise dominance, X and Y are incomparable, as are Y and Z. Hence,* $\mathrm{norm}(\mathrm{st}_N(T))$ *is* $\{U_X, U_Y\}$*, where U_X is the normal form decision with corresponding gamble X, and similarly for U_Y. But, since Z clearly dominates X, we have* $\mathrm{opt}(\{X, Y, Z\}) = \{Y, Z\}$*, therefore* $\mathrm{norm}_{\mathrm{opt}}(T) = \{U_Y, U_Z\}$*. Restricting this solution to* $\mathrm{st}_N(T)$ *gives the normal form solution* $\{U_Y\}$*. Concluding,*

$$\{U_X, U_Y\} = \mathrm{norm}(\mathrm{st}_N(T)) \neq \mathrm{st}_N(\mathrm{norm}(T)) = \{U_Y\}$$

and therefore the normal form operator induced by pointwise dominance lacks subtree perfectness.

We wish to determine for which choice functions the induced normal form operator satisfies subtree perfectness. Since the behaviour of such an operator is entirely determined by the choice function, this amounts to finding a collection of properties for choice functions that are necessary and sufficient to imply subtree perfectness.

2.2 Subtree Perfectness Properties

It turns out that only three very simple properties are required for subtree perfectness to hold for normal form operators induced by choice functions. We first present them, before discussing their meaning and links to other properties in the theory of choice functions, and consider some further properties that can be derived from combinations of the three.

Property 1 (Conditioning Property). *Let A be a non-empty event, and let $\mathcal{X}$ be a non-empty finite A-consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$. If $X \in \operatorname{opt}(\mathcal{X}|A)$, then $Y \in \operatorname{opt}(\mathcal{X}|A)$.*

Property 2 (Intersection property). *For any event $A \neq \emptyset$ and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$ and $\operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$,*

$$\operatorname{opt}(\mathcal{Y}|A) = \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

Property 3 (Mixture property). *For any events A and B such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, any $\overline{A} \cap B$-consistent gamble Z, and any non-empty finite $A \cap B$-consistent set of gambles $\mathcal{X}$,*

$$\operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B) = A \operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

Conditioning places a fairly weak and very sensible restriction on a choice function. It states that if two gambles are equal on A, then either both are optimal in $\mathcal{X}$ given A or neither is. This is weaker than the quite reasonable condition that removing either Y or X from $\mathcal{X}$ effectively changes nothing, so Conditioning ought not to arouse any controversy. Failure of the property implies a rather strange form of conditional preference that would cause more concern than lack of subtree perfectness anyway.

Intersection states that, if we have a set of gambles $\mathcal{Y}$ and add elements to reach $\mathcal{X}$, either everything from $\mathcal{Y}$ becomes non-optimal, or the optimality of a particular $Y \in \mathcal{Y}$ is unchanged. It turns out that this property is equivalent to several ordering properties commonly discussed in the choice function literature, including Arrow [5], Sen [66], and Luce and Raiffa [46], the last-mentioned giving the following illuminating summary of the property's meaning:

> The addition of new acts to a decision problem under uncertainy never changes old, originally non-optimal acts into optimal ones and, in addition, either
>
> (i) All the old originally optimal acts remain optimal, or,
>
> (ii) None of the old originally optimal acts remain optimal.

We now give formal definitions of these other properties and prove their equivalence.

Property 4 (Strong path independence)**.** *For any non-empty event A and any non-empty finite A-consistent sets of gambles $\mathcal{X}_1$, ...,$\mathcal{X}_n$, there is a non-empty $\mathcal{I} \subseteq \{1, \dots, n\}$ such that*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \bigcup_{i\in\mathcal{I}} \operatorname{opt}(\mathcal{X}_i|A).$$

This property states that, if a set of gambles $\mathcal{X}$ is partitioned, then each element $\mathcal{X}_i$ of the partition either contributes nothing to $\operatorname{opt}(\mathcal{X}|A)$ or contributes all of $\operatorname{opt}(\mathcal{X}_i|A)$.

Property 5 (Very strong path independence)**.** *For any non-empty event A and any non-empty finite A-consistent sets of gambles $\mathcal{X}_1$, ...,$\mathcal{X}_n$,*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \bigcup_{\substack{i=1 \\ \mathcal{X}_i \cap \operatorname{opt}(\cup_{i=1}^{n} \mathcal{X}_i|A) \neq \emptyset}}^{n} \operatorname{opt}(\mathcal{X}_i|A)$$

This property is obviously implied by Strong Path Independence, differing only in that the set $\mathcal{I}$ is explicitly defined.

Property 6 (Total preorder)**.** *For every event $A \neq \emptyset$, there is a total preorder $\succeq_A$ on A-consistent gambles such that for every non-empty finite set of A-consistent gambles $\mathcal{X}$,*

$$\operatorname{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(X \succeq_A Y)\}$$

A total preorder, or weak order, is frequently assumed as a minimum requirement for rational choice, although, as Luce and Raiffa argue [46, pp. 289–290], it is difficult to see why without invoking elaborate arguments such as subtree perfectness.

Lemma 2.3. *Intersection, Strong Path Independence, Very Strong Path Independence, and Total Preordering are equivalent.*

Proof. In this proof, A is a non-empty event and all gambles are A-consistent.

Intersection $\implies$ Very Strong Path Independence. Let $\mathcal{X}_1, \dots, \mathcal{X}_n$ be non-empty finite sets of gambles, and let $\mathcal{X} = \bigcup_{i=1}^n \mathcal{X}_i$. If $\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{X}_k \neq \emptyset$, then

$$\mathrm{opt}(\mathcal{X}_k|A) = \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{X}_k.$$

Hence,

$$\begin{aligned}\mathrm{opt}(\mathcal{X}|A) = \bigcup_{k=1}^n \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{X}_k &= \bigcup_{\substack{k=1\\ \mathcal{X}_k \cap \mathrm{opt}(\mathcal{X}|A) \neq \emptyset}}^n \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{X}_k \\ &= \bigcup_{\substack{k=1\\ \mathcal{X}_k \cap \mathrm{opt}(\mathcal{X}|A) \neq \emptyset}}^n \mathrm{opt}(\mathcal{X}_k|A)\end{aligned}$$

Very Strong Path Independence $\implies$ Strong Path Independence. Immediate.

Strong Path Independence $\implies$ Total Preordering. Define $X \succeq_A Y$ if $X \in \mathrm{opt}(\{X, Y\}|A)$. First, we prove that $\succeq_A$ is a total preorder (i.e. total, reflexive, and transitive). Clearly, $\succeq_A$ is total since $X \in \mathrm{opt}(\{X, Y\}|A)$ or $Y \in \mathrm{opt}(\{X, Y\}|A)$, hence $X \succeq_A Y$ or $Y \succeq_A X$, for all gambles X and Y. Obviously, $\succeq_A$ is reflexive. Is $\succeq_A$ transitive? Suppose $X \succeq_A Y$ and $Y \succeq_A Z$.

By Strong Path Independence, using the partition $\{X\}$ and $\{Y, Z\}$,

$$\mathrm{opt}(\{X, Y, Z\}|A) = \begin{cases} \mathrm{opt}(\{Y, Z\}|A), & \text{or} \\ \{X\}, & \text{or} \\ \mathrm{opt}(\{Y, Z\}|A) \cup \{X\}. \end{cases}$$

Since, $Y \succeq_A Z$, it follows that $\mathrm{opt}(\{X, Y, Z\}|A) \neq \{Z\}$.

Again, by Strong Path Independence and the partition $\{X,Y\}$ and $\{Z\}$,

$$\operatorname{opt}(\{X,Y,Z\}|A) = \begin{cases} \operatorname{opt}(\{X,Y\}|A), & \text{or} \\ \{Z\}, & \text{or} \\ \operatorname{opt}(\{X,Y\}|A) \cup \{Z\}. \end{cases}$$

As we just showed, the middle case is impossible. Hence,

$$X \in \operatorname{opt}(\{X,Y\}|A) \implies X \in \operatorname{opt}(\{X,Y,Z\}|A).$$

Once more by Strong Path Independence and the partition $\{X,Z\}$ and $\{Y\}$,

$$\operatorname{opt}(\{X,Y,Z\}|A) = \begin{cases} \operatorname{opt}(\{X,Z\}|A), & \text{or} \\ \{Y\}, & \text{or} \\ \operatorname{opt}(\{X,Z\}|A) \cup \{Y\}. \end{cases}$$

We just showed that $X \in \operatorname{opt}(\{X,Y,Z\}|A)$, hence the second case cannot occur and it can only be that also $X \in \operatorname{opt}(\{X,Z\}|A)$, establishing $X \succeq_A Z$.

Finally, we prove that

$$\operatorname{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(X \succeq_A Y)\},$$

or equivalently, we prove for any $X \in \mathcal{X}$ that $X \in \operatorname{opt}(\mathcal{X}|A)$ if and only if $X \in \operatorname{opt}(\{X,Y\}|A)$ for all $Y \in \mathcal{X}$.

Indeed, by Strong Path Independence, for any X and Y in $\mathcal{X}$, it holds that

$$\operatorname{opt}(\mathcal{X}|A) = \begin{cases} \operatorname{opt}(\{X,Y\}|A), & \text{or} \\ \operatorname{opt}(\mathcal{X} \setminus \{X,Y\}|A), & \text{or} \\ \operatorname{opt}(\{X,Y\}|A) \cup \operatorname{opt}(\mathcal{X} \setminus \{X,Y\}|A). \end{cases}$$

and hence, if $X \in \operatorname{opt}(\mathcal{X}|A)$ then the second option is impossible and therefore $X \in \operatorname{opt}(\{X,Y\}|A)$ for all $Y \in \mathcal{Y}$.

Conversely, again by Strong Path Independence

$$\operatorname{opt}(\mathcal{X}|A) = \operatorname{opt}\left(\bigcup_{Y \in \mathcal{X}} \{X,Y\} \middle| A\right) = \bigcup_{Y \in \mathcal{Y}} \operatorname{opt}(\{X,Y\}|A)$$

for some subset $\mathcal{Y}$ of $\mathcal{X}$, and hence, if $X \in \operatorname{opt}(\{X, Y\}|A)$ for all $Y \in \mathcal{X}$, then $X \in \operatorname{opt}(\mathcal{X}|A)$.

Total Preordering $\implies$ Intersection. Assume that $\operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. This means that there must be a $Y^* \in \mathcal{Y}$ such that $Y^* \succeq_A X$ for all $X \in \mathcal{X}$. Clearly, $Y^* \in \operatorname{opt}(\mathcal{Y}|A)$. But, for all $Y \in \operatorname{opt}(\mathcal{Y}|A)$ it must also hold that $Y \succeq_A Y^*$, and hence $Y \succeq_A X$ for all $X \in \mathcal{X}$ as $\succeq_A$ is transitive. We conclude:

$$\operatorname{opt}(\mathcal{Y}|A) = \{Y \in \mathcal{Y} \colon (\forall X \in \mathcal{X})(Y \succeq_A X)\} = \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}. \qquad \square$$

Mixture is closely related to the famous Independence Axiom (see e.g. [49]). This principle has many forms, depending on context (McClennen considers many: see properties CIND, CIND-E, CIND-S, IND, and ISO [51, pp. xi–xii]). In its simplest form, the principle states that the subject prefers reward r_1 to r_2 if and only if, for all rewards r_3 and any non-zero probability p, she prefers $pr_1 \oplus (1-p)r_3$ to $pr_2 \oplus (1-p)r_3$. Our property is a natural generalization of this to include gambles instead of rewards, and sets instead of pairwise comparisons. In relation to the various properties in the literature, it seems in essence to be McClennen's CIND [51, pp. 57–58] (roughly, this is Mixture but with probabilities and unconditional preference) adapted to involve Arrow's concept of conditional preference [4].

The Independence Axiom and its variations have attracted more widespread criticism and questioning than the weak order property. The paradoxes of Allais [1] and Ellsberg [18] demonstrate experimental violations of the axiom, and accounts including Kahneman and Tversky [34], Machina [50], and McClennen [51] have argued that violations may be philosophically acceptable.

The issue is one of "context sensitivity": if one considers the relative worth of rewards to depend on the gambles in which they are embedded, then violations are reasonable. Machina's example [50, § 6.6] is illuminating. Suppose that, in the formulation of the Independence Axiom above, r_1 and r_3 are rewards of a similar type, but r_3 is vastly superior (Machina's example has r_3 as a romantic week with a beautiful movie star, and r_1 as watching one of their movies), and r_2 is something rather different (a gourmet hamburger). It is reasonable to suppose that r_1 is preferable to r_2, but the gamble $0.97r_3 \oplus 0.03r_2$ is preferable to $0.97r_3 \oplus 0.03r_1$, since in the latter case the disappointment of failing to obtain r_3 would make watching the movie very upsetting. We return to this discussion later, in Section 3.3.

If the arguments in favour of Mixture seem plausible, then the following stronger property should also be appealing.

Property 7 (Multiple Mixture Property). *For any event B and partition $A_1, \dots, A_n$ such that $A_i \cap B \neq \emptyset$ for all i, and sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$ such that $\mathcal{X}_i$ is $A_i \cap B$-consistent,*

$$\operatorname{opt}\left(\bigoplus_{i=1}^{n} A_i \mathcal{X}_i \middle| B\right) = \bigoplus_{i=1}^{n} A_i \operatorname{opt}(\mathcal{X}_i | A_i \cap B). \tag{2.1}$$

This is simply an extension of Mixture to consider more complicated mixtures of gambles. There is a link between it and Intersection and Mixture. Note that the opposite implication in this lemma does *not* hold.

Lemma 2.4. *Intersection and Mixture together imply Multiple Mixture.*

Proof. The statement is trivial if $n = 1$ (because, in that case, $A_1 = \Omega$). Let us prove the statement also in case $n \geq 2$. Our proof relies on the technical Lemma B.3 from the Appendix.

Let $\mathcal{X} = \bigoplus_{i=1}^{n} A_i \mathcal{X}_i$. The idea of the proof is, for each i, to partition $\mathcal{X}$ into sets of the form $A_i \mathcal{X}_i \oplus \overline{A}_i Z$, and then apply Mixture. Consider any $k \in \{1, \dots, n\}$ and let

$$\mathcal{Z}_k = \bigoplus_{j \neq k} A'_j \mathcal{X}_j$$

where $(A'_j)_{j \neq k}$ forms an arbitrary partition of Ω such that $\overline{A}_k \cap A'_j = A_j$ for all $j \neq k$. Clearly, $\mathcal{Z}_k$ is $\overline{A}_k \cap B$-consistent because we can trivially find a consistent decision tree T with $\operatorname{ev}(T) = \overline{A}_k \cap B$ and $\operatorname{gamb}(T) = \mathcal{Z}_k$, using the $A_j \cap B$-consistency (and hence, $\overline{A}_k \cap A'_j \cap B$-consistency) of each $\mathcal{X}_j$ for $j \neq k$.

Now, observe that by construction of $\mathcal{Z}_k$,

$$\mathcal{X} = A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k = \bigcup_{Z_k \in \mathcal{Z}_k} (A_k \mathcal{X}_k \oplus \overline{A}_k Z_k),$$

where the latter equality follows from using a combination of the two representations for $\oplus$: Eq. (1.4) and Eq. (1.5). Note that $\mathcal{X}$ is B-consistent (indeed, because each $\mathcal{X}_i$

is $A_i \cap B$-consistent, we can trivially find a consistent decision tree T with $\mathrm{ev}(T) = B$ and $\mathrm{gamb}(T) = \mathcal{X}$).

Since Intersection holds, Strong Path Independence holds as well by Lemma 2.3. So, if we apply $\mathrm{opt}(\cdot|B)$ on both sides of the above equality, then it follows from Strong Path Independence that

$$\mathrm{opt}(\mathcal{X}|B) = \bigcup_{Z_k \in \mathcal{Z}_k^*} \mathrm{opt}(A_k \mathcal{X}_k \oplus \overline{A}_k Z_k | B),$$

for some $\mathcal{Z}_k^* \subseteq \mathcal{Z}_k$. By Mixture,

$$= \bigcup_{Z_k \in \mathcal{Z}_k^*} (A_k \,\mathrm{opt}(\mathcal{X}_k | A_k \cap B) \oplus \overline{A}_k Z)$$

$$= A_k \,\mathrm{opt}(\mathcal{X}_k | A_k \cap B) \oplus \overline{A}_k \mathcal{Z}_k^* \tag{2.2}$$

Since this holds for each $k \in \{1, \ldots, n\}$, we arrive at Eq. (2.1), by Lemma B.3. □

In the next section we show that Conditioning, Intersection, and Mixture are necessary and sufficient for subtree perfectness of $\mathrm{norm}_{\mathrm{opt}}$. Before doing so, we demonstrate that it is possible to satisfy any two properties without satisfying the other, so they are indeed distinct.

Example 2.5. *Assume real-valued rewards in this example.*

- *Conditioning and Intersection but not Mixture. Let* opt *be the maximin choice function:* $\mathrm{opt}(\mathcal{X}|A) = \arg\max_{X \in \mathcal{X}} \min_{\omega \in A} X(\omega)$*. It is clear by definition that the two required properties are satisfied. Mixture fails trivially: suppose that the minimum reward for all* $A\mathcal{X} \oplus \overline{A}Z$ *comes only from* $\overline{A}Z$*.*

- *Conditioning and Mixture but not Intersection. Let* opt *be the choice function corresponding to the pointwise dominance strict partial order* $>_A$*:* $X >_A Y$ *if* $X(\omega) \geq Y(\omega)$ *for all* $\omega \in A$ *and* $X \neq Y$*. This is not a total preorder, therefore Intersection fails, but the other two properties follow immediately from the definition.*

- *Intersection and Mixture but not Conditioning. Suppose that* opt *is a choice function corresponding to maximizing expected utility under some mass function* p *for which all events have positive probability. Suppose that there are a* *particular X, Y such that $AX = AY$ but $X \neq Y$, and no other gamble with the same expected value to X and Y conditional on A. Let* opt^* *be the choice function obtained by using* opt *but modifying the ordering so that X is preferred to Y given A, changing nothing else. This construction violates Conditioning, but still satisfies the other two.*

2.3 Subtree Perfectness Theorem

Theorem 2.6 (Subtree perfectness theorem)**.** *A normal form operator* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect if and only if* opt *satisfies Conditioning, Intersection, and Mixture.*

The proof relies on several technical lemmas, which give some insight into the various roles of the three properties. The first lemma checks that, for decision trees whose root is at a decision node, the initial decisions that are in optimal normal form decisions are determined exactly by the $\mathcal{I}$ of Very Strong Path Independence.

Lemma 2.7. *Consider a consistent decision tree T whose root is a decision node, so $T = \bigsqcup_{i=1}^{n} T_i$, and any choice function* opt*. For each tree T_i, let N_i be its root. Then, N_i is in at least one element of* $\mathrm{norm}_{\mathrm{opt}}(T)$ *if and only if*

$$\mathrm{gamb}(T_i) \cap \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)) \neq \emptyset. \tag{2.3}$$

Proof. By Eq. (1.1), Eq. (2.3) holds if and only if

$$\mathrm{gamb}(\mathrm{nfd}(T_i)) \cap \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)) \neq \emptyset,$$

or equivalently, if and only if there is a normal form decision $U \in \mathrm{nfd}(T_i)$ such that $\mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))$ (remember that $\mathrm{gamb}(U)$ is a singleton).

But, by definition of the gamb operator, it also holds that $\mathrm{gamb}(U) = \mathrm{gamb}(\sqcup U)$. Hence, Eq. (2.3) holds if and only if there is a normal form decision $U \in \mathrm{nfd}(T_i)$ such that $\mathrm{gamb}(\sqcup U) \subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))$. Since U is a normal form decision of T_i, and T is formed by combination at a decision node, $\sqcup U$ is a normal form decision of T. By definition of $\mathrm{norm}_{\mathrm{opt}}$, this is equivalent to stating that Eq. (2.3) holds if and only if N_i is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$. □

As mentioned earlier, gambles are much easier to work with than normal form decisions, so we use them wherever possible. The following two lemmas allow us to switch from optimal gambles to optimal normal form decisions.

Lemma 2.8. *If T is a consistent decision tree with a chance node as the root, so $T = \bigodot_{i=1}^n E_i T_i$, and* opt *is a choice function satisfying Conditioning, then*

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)) \tag{2.4}$$

implies

$$\mathrm{norm}_{\mathrm{opt}}(T) = \bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i).$$

Proof. Assume that Eq. (2.4) holds.

First, consider a normal form decision $U \in \bigodot_{i=1}^n E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i)$. Obviously,

$$\mathrm{gamb}(U) \subseteq \mathrm{gamb}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i)\right)$$

so, by the definition of gamb, Eq. (1.7) in particular,

$$= \bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i))$$

and hence, by Eq. (2.4),

$$= \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)).$$

So, there is a $V \in \mathrm{norm}_{\mathrm{opt}}(T)$ such that $\mathrm{gamb}(V) = \mathrm{gamb}(U)$. Since $U \in \mathrm{nfd}(T)$, by definition of $\mathrm{norm}_{\mathrm{opt}}$ we have $U \in \mathrm{norm}_{\mathrm{opt}}(T)$. So we have shown that

$$\mathrm{norm}_{\mathrm{opt}}(T) \supseteq \bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i).$$

Next, consider a normal form decision $U \in \mathrm{norm}_{\mathrm{opt}}(T)$. We know by Eq. (2.4) that

$$\mathrm{gamb}(U) \subseteq \bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)).$$

We can write $U = \bigodot_{i=1}^n E_i U_i$, where $U_i \in \mathrm{nfd}(T_i)$, so

$$\bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(U_i) \subseteq \bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)).$$

Consider any k and any normal form decision $V_k \in \operatorname{norm}_{\mathrm{opt}}(T_k)$. By the above equation and Eq. (1.10), we can choose each V_k such that $E_k \operatorname{gamb}(V_k) = E_k \operatorname{gamb}(U_k)$. Of course, because $V_k \in \operatorname{norm}_{\mathrm{opt}}(T_k)$,

$$\operatorname{gamb}(V_k) \subseteq \operatorname{opt}(\operatorname{gamb}(T_k)|\operatorname{ev}(T_k)).$$

We wish to establish that also $\operatorname{gamb}(U_k) \subseteq \operatorname{opt}(\operatorname{gamb}(T_k)|\operatorname{ev}(T_k))$.

This will follow from Conditioning if we can check that all the conditions hold. Observe that both singletons $\operatorname{gamb}(U_k)$ and $\operatorname{gamb}(V_k)$ are subsets of $\operatorname{gamb}(T_k)$, $E_k \operatorname{gamb}(U_k) = E_k \operatorname{gamb}(V_k)$, and $\operatorname{gamb}(V_k) \subseteq \operatorname{opt}(\operatorname{gamb}(T_k)|\operatorname{ev}(T_k))$. Consistency of T confirms that $\operatorname{gamb}(T_k)$ is $\operatorname{ev}(T_k)$-consistent. Hence, Conditioning applies, and

$$\operatorname{gamb}(U_k) \subseteq \operatorname{opt}(\operatorname{gamb}(T_k)|\operatorname{ev}(T_k)).$$

Therefore, $U_k \in \operatorname{norm}_{\mathrm{opt}}(T_k)$ by definition of $\operatorname{norm}_{\mathrm{opt}}(T_i)$. Since this holds for any k, we conclude that $U \in \bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i)$. So, we have shown that also

$$\operatorname{norm}_{\mathrm{opt}}(T) \subseteq \bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i).$$

$\square$

Lemma 2.9. *If T is a consistent decision tree whose root is a decision node, so $T = \bigsqcup_{i=1}^{n} T_i$ and* opt *is a choice function satisfying Intersection, then*

$$\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)) = \bigcup_{i \in \mathcal{I}} \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T_i)) \tag{2.5}$$

implies

$$\operatorname{norm}_{\mathrm{opt}}(T) = \operatorname{nfd}\left(\bigsqcup_{i \in \mathcal{I}} \operatorname{norm}_{\mathrm{opt}}(T_i)\right),$$

where $\mathcal{I} = \{i \in \{1, \dots, n\} \colon \operatorname{gamb}(T_i) \cap \operatorname{opt}(\operatorname{gamb}(T)|\operatorname{ev}(T)) \neq \emptyset\}$.

Proof. Assume that Eq. (2.5) holds.

Consider any normal form decision $V \in \operatorname{nfd}\left(\bigsqcup_{i \in \mathcal{I}} \operatorname{norm}_{\mathrm{opt}}(T_i)\right)$. By definition of gamb,

$$\operatorname{gamb}(V) \subseteq \bigcup_{i \in \mathcal{I}} \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T_i))$$

and, by Eq. (2.5),

$$= \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)).$$

Hence, by definition of $\mathrm{norm}_{\mathrm{opt}}$, and the obvious fact that $V \in \mathrm{nfd}(T)$, it follows that $V \in \mathrm{norm}_{\mathrm{opt}}(T)$. So we have shown that

$$\mathrm{norm}_{\mathrm{opt}}(T) \supseteq \mathrm{nfd}\left(\bigsqcup_{i\in\mathcal{I}} \mathrm{norm}_{\mathrm{opt}}(T_i)\right).$$

Conversely, let $V \in \mathrm{norm}_{\mathrm{opt}}(T)$. Then, again by Eq. (2.5),

$$\mathrm{gamb}(V) \subseteq \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \bigcup_{i\in\mathcal{I}} \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)).$$

Now $V = \sqcup U$ where $U \in \mathrm{nfd}(T_i)$ for some $i \in \mathcal{I}$. We want to show that $U \in \mathrm{norm}_{\mathrm{opt}}(T_i)$.

Indeed, let X be the gamble corresponding to V, and also U,

$$\mathrm{gamb}(V) = \mathrm{gamb}(U) = \{X\}.$$

Because $V \in \mathrm{norm}_{\mathrm{opt}}(T)$, we know that $X \in \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))$. It is established that $U \in \mathrm{norm}_{\mathrm{opt}}(T_i)$ if we can show that $X \in \mathrm{opt}(\mathrm{gamb}(T_i)|\mathrm{ev}(T_i))$. But this follows from Intersection, because $i \in \mathcal{I}$ (recall the definition of $\mathcal{I}$), $\mathrm{gamb}(T_i) \subseteq \mathrm{gamb}(T)$, $\mathrm{ev}(T) = \mathrm{ev}(T_i)$, $X \in \mathrm{gamb}(T_i)$, and all sets of gambles are consistent with respect to the relevant events:

$$\mathrm{opt}(\mathrm{gamb}(T_i)|\mathrm{ev}(T_i)) = \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)) \cap \mathrm{gamb}(T_i).$$

Concluding, also

$$\mathrm{norm}_{\mathrm{opt}}(T) \subseteq \mathrm{nfd}\left(\bigsqcup_{i\in\mathcal{I}} \mathrm{norm}_{\mathrm{opt}}(T_i)\right).$$

$\square$

The next lemma concerns necessity of Conditioning, Intersection, and Mixture for subtree perfectness—to establish this result, interestingly, it suffices to consider only the two decision trees in Figure 2.3.

Lemma 2.10. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect, then* opt *satisfies Conditioning, Intersection, and Mixture.*

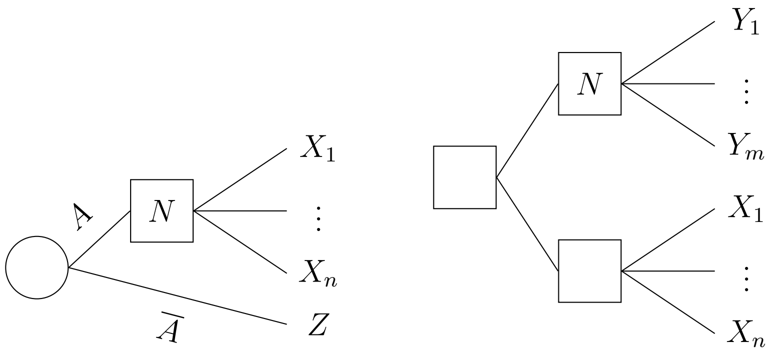

Figure 2.3: Decision trees for Lemma 2.10.

Proof. Assume that $\mathrm{norm}_{\mathrm{opt}}$ is subtree perfect.

We first establish Conditioning. Let A be a non-empty event, and let $\mathcal{X}$ be a non-empty finite set of A-consistent gambles such that $\{X, Y\} \subseteq \mathcal{X}$ with $AX = AY$ and $X \in \mathrm{opt}(\mathcal{X}|A)$. We show that $Y \in \mathrm{opt}(\mathcal{X}|A)$. If $A = \Omega$ the result is trivial, so assume $A \subset \Omega$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\mathrm{gamb}(T_1) = \mathcal{X}$, $\mathrm{ev}(T) = \Omega$, and T_2 is a normal form decision with $\mathrm{gamb}(T_2) = \{Z\}$, an $\overline{A}$-consistent gamble (see the left tree in Fig. 2.3). We know by consistency of the gambles that there is such a T (see Definition 1.17).

Consider $U \in \mathrm{nfd}(T_1)$ with $\mathrm{gamb}(U) = \{X\}$ and $V \in \mathrm{nfd}(T_1)$ with $\mathrm{gamb}(V) = \{Y\}$. By definition of $\mathrm{norm}_{\mathrm{opt}}$, we have $U \in \mathrm{norm}_{\mathrm{opt}}(T_i)$. Therefore by subtree perfectness, $AU \odot \overline{A}T_2 \in \mathrm{norm}_{\mathrm{opt}}(T)$ and of course $AV \odot \overline{A}T_2 \in \mathrm{norm}_{\mathrm{opt}}(T)$. Again by subtree perfectness, $V \in \mathrm{norm}_{\mathrm{opt}}(T_i)$, whence $Y \in \mathrm{opt}(\mathcal{X}|A)$.

Next, we establish Intersection. Let A be a non-empty event, and let $\mathcal{Y} \subseteq \mathcal{X}$ be non-empty finite A-consistent sets of gambles such that $\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. We show that $\mathrm{opt}(\mathcal{Y}|A) = \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y}$.

Let $T = T_1 \sqcup T_2$ be a consistent decision tree with $\mathrm{ev}(T) = A$, $\mathrm{gamb}(T_1) = \mathcal{Y}$ and $\mathrm{gamb}(T_2) = \mathcal{X}$ (see the right tree in Fig. 2.3). We know by consistency of the gambles that there is such a T.

Let N be the decision node at the root of T_1. By subtree perfectness, we have

$$\mathrm{gamb}(\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T))) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_N(T))).$$

The right-hand side is equal to $\operatorname{opt}(\mathcal{Y}|A)$. Also,

$$\begin{aligned}\operatorname{gamb}(\operatorname{st}_N(\operatorname{norm}_{\operatorname{opt}}(T))) &= \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T)) \cap \operatorname{gamb}(\operatorname{st}_N(T)) \\ &= \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}\end{aligned}$$

as required.

Finally, we establish Mixture. Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let $\mathcal{X}$ be a non-empty finite $A \cap B$-consistent set of gambles, and let Z be a $\overline{A} \cap B$-consistent gamble.

Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree such that $\operatorname{ev}(T) = B$, $\operatorname{gamb}(T_1) = \mathcal{X}$, and $\operatorname{gamb}(T_2) = \{Z\}$ (see the left tree in Fig. 2.3).

By subtree perfectness, we have (letting N be the root node of T_1)

$$\operatorname{gamb}(\operatorname{st}_N(\operatorname{norm}_{\operatorname{opt}}(T))) = \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(\operatorname{st}_N(T))).$$

Here, the right-hand side is $\operatorname{opt}(\mathcal{X}|A \cap B)$, and

$$\operatorname{gamb}(\operatorname{st}_N(\operatorname{norm}_{\operatorname{opt}}(T))) = \{X \in \mathcal{X} \colon AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B)\},$$

whence Mixture follows. □

The method of proof we use for Theorem 2.6 is structural induction: assume that subtree perfectness holds for all subtrees of a tree T, and show it then holds for T. The following lemma shows that it is only the immediate successors of the root node of T that need to be examined, making the ultimate proof more straightforward. Recall that $\operatorname{ch}(T)$ is the set of immediate successor nodes of the root node of T.

Lemma 2.11. *Let* norm *be any normal form operator. Let T be a consistent decision tree. If,*

(i) for all nodes $K \in \operatorname{ch}(T)$ such that K is in at least one element of $\operatorname{norm}(T)$*,*

$$\operatorname{st}_K(\operatorname{norm}(T)) = \operatorname{norm}(\operatorname{st}_K(T)),$$

(ii) and, for all nodes $K \in \operatorname{ch}(T)$, and all nodes $L \in \operatorname{st}_K(T)$ such that L is in at least one element of $\operatorname{norm}(\operatorname{st}_K(T))$*,*

$$\operatorname{st}_L(\operatorname{norm}(\operatorname{st}_K(T))) = \operatorname{norm}(\operatorname{st}_L(\operatorname{st}_K(T))),$$

then, for all nodes N in T such that N is in at least one element of $\mathrm{norm}(T)$*,*

$$\mathrm{st}_N(\mathrm{norm}(T)) = \mathrm{norm}(\mathrm{st}_N(T)).$$

Proof. If N is the root of T, then the statement is trivial. If $N \in \mathrm{ch}(T)$, then the statement follows from (i). Otherwise, N must belong to $\mathrm{st}_K(T)$ for one $K \in \mathrm{ch}(T)$.

First, note that K is a node of at least one element of $\mathrm{norm}(T)$. Indeed, it is given that N is a node of at least one element, say U, of $\mathrm{norm}(T)$. Then, obviously, K must also be a node of U, simply because any node on the unique path within T between the root of T and N must be a node of U, and one of those nodes is K. So, K is a node of an element of $\mathrm{norm}(T)$.

Secondly, note that N is also a node of at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$. Indeed, N is a node of an element of $\mathrm{norm}(T)$, and hence, in particular also of $\mathrm{st}_K(\mathrm{norm}(T))$. But, by (i), and the fact that K is a node of at least one element of $\mathrm{norm}(T)$ (as just proven), it follows that $\mathrm{st}_K(\mathrm{norm}(T)) = \mathrm{norm}(\mathrm{st}_K(T))$. Hence, N is also a node of at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$.

Combining everything, it follows that

$$\mathrm{norm}(\mathrm{st}_N(T)) = \mathrm{norm}(\mathrm{st}_N(\mathrm{st}_K(T)))$$

so, by (ii), and because N is in at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$,

$$= \mathrm{st}_N(\mathrm{norm}(\mathrm{st}_K(T)))$$

hence, by (i), and since K is in at least one element of $\mathrm{norm}(T)$,

$$= \mathrm{st}_N(\mathrm{st}_K(\mathrm{norm}(T))) = \mathrm{st}_N(\mathrm{norm}(T)).$$

□

These are all the necessary ingredients to prove the subtree perfectness theorem.

Proof of Theorem 2.6. "only if". See Lemma 2.10.

"if". We proceed by structural induction on all possible arguments of $\mathrm{norm}_{\mathrm{opt}}$, that is, on all consistent decision trees. In the base step, we prove the implication for trees consisting of only a single node. In the induction step, we prove that if the implication holds for the subtrees at every child of the root node, then the implication also holds for the whole tree.

First, if the decision tree T has only a single node, and hence, a reward at the root and no further children, then subtree perfectness is trivially satisfied. No properties of opt are required at this stage.

Next, suppose that the consistent decision tree T has multiple nodes. Let $\{N_1, \ldots, N_n\} = \mathrm{ch}(T)$, and let $T_i = \mathrm{st}_{N_i}(T)$. The induction hypothesis says that subtree perfectness is satisfied for all subtrees at every child of the root node, that is, for all T_i. More precisely, for all $i \in \{1, \ldots, n\}$, and all nodes $L \in T_i$ such that L is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T_i)$,

$$\mathrm{st}_L(\mathrm{norm}_{\mathrm{opt}}(T_i)) = \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_L(T_i)).$$

We must show that

$$\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_N(T))$$

for all nodes N in T such that N is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$. By Lemma 2.11, and the induction hypothesis, it suffices to prove the above equality only for $N \in \mathrm{ch}(T)$, that is, it suffices to show that

$$\mathrm{st}_{N_i}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{norm}_{\mathrm{opt}}(T_i) \tag{2.6}$$

for each $i \in \{1, \ldots, n\}$ such that N_i is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$.

If T has a chance node as its root, that is, $T = \bigodot_{i=1}^n E_i T_i$, then all N_i are actually in every element of $\mathrm{norm}_{\mathrm{opt}}(T)$, so we must simply establish Eq. (2.6) for all $i \in \{1, \ldots, n\}$. Observe that, if we can establish

$$\mathrm{norm}_{\mathrm{opt}}(T) = \bigodot_{i=1}^{n} E_i \ \mathrm{norm}_{\mathrm{opt}}(T_i), \tag{2.7}$$

then Eq. (2.6) follows immediately. Eq. (2.7) is most easily derived using Lemma 2.8. Indeed, by Eq. (1.10),

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))$$

and by the definition of the gamb operator, Eq. (1.6b) in particular,

$$= \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right)$$

and so by Lemma 2.4,

$$= \bigoplus_{i=1}^{n} E_i \operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T)\cap E_i)$$

so, since $\operatorname{ev}(T)\cap E_i = \operatorname{ev}(T_i)$, and again by Eq. (1.10),

$$= \bigoplus_{i=1}^{n} E_i \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T_i)),$$

whence Eq. (2.7) follows by Lemma 2.8.

Finally, assume that T has a decision node as its root, that is, $T = \bigsqcup_{i=1}^{n} T_i$. Let $\mathcal{I}$ be the subset of $\{1,\dots,n\}$ such that $i \in \mathcal{I}$ if and only if N_i is in at least one element of $\operatorname{norm}_{\operatorname{opt}}(T)$. We must establish Eq. (2.6) for all $i \in \mathcal{I}$. Equivalently, we must show that

$$\operatorname{norm}_{\operatorname{opt}}(T) = \operatorname{nfd}\left(\bigsqcup_{i\in\mathcal{I}} \operatorname{norm}_{\operatorname{opt}}(T_i)\right). \tag{2.8}$$

Indeed, by Eq. (1.10),

$$\operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T)) = \operatorname{opt}(\operatorname{gamb}(T)|\operatorname{ev}(T))$$

and by the definition of the gamb operator, Eq. (1.6c) in particular,

$$= \operatorname{opt}\left(\bigcup_{i=1}^{n} \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right)$$

and so by Very Strong Path Independence,

$$= \bigcup_{i\in\mathcal{I}^*} \operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T)),$$

where $\mathcal{I}^* = \{i \in \{1,\dots,n\}\colon \operatorname{gamb}(T_i)\cap \operatorname{opt}(\operatorname{gamb}(T)|\operatorname{ev}(T)) \neq \emptyset\}$, and so because $\operatorname{ev}(T) = \operatorname{ev}(T_i)$, and again by Eq. (1.10),

$$= \bigcup_{i\in\mathcal{I}^*} \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T_i)).$$

Hence, the conditions of Lemma 2.9 are satisfied, and $\mathcal{I}^* = \mathcal{I}$ by Lemma 2.7, so Eq. (2.8) is established. $\square$

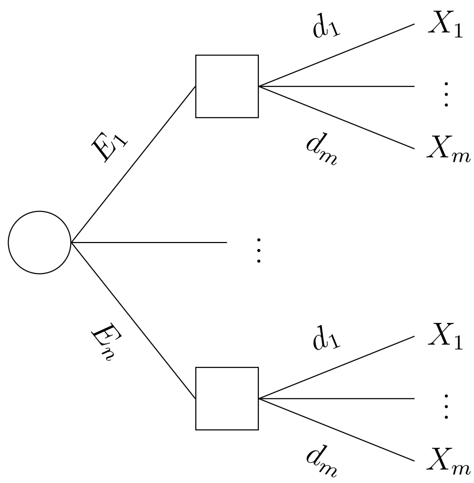

Figure 2.4: Simple sequential decision tree with one decision node per branch

2.4 Special Case: Subtree Perfectness In Statistical Decision Theory

The conditions required for subtree perfectness are highly restrictive, and only very few choice functions will satisfy them. For other choice functions, we can at least investigate particular classes of decision tree for which subtree perfectness holds. We consider a well-known simple sequential decision problem. A subject is going to observe the outcome of some experiment, and then must choose one action from a set $\mathcal{D}$. Hypothesis testing and parameter estimation are examples of this type of problem. Assuming that there are finite possible values for the experiment, and finite actions in $\mathcal{D}$, this problem can be represented on a decision tree, as in Fig. 2.4.

It is straightforward to see that subtree perfectness for such a tree would follow from Conditioning and Multiple Mixture, so Intersection is not necessary. In fact, only a weaker version of Multiple Mixture is necessary for subtree perfectness of Fig. 2.4. This is because Multiple Mixture requires all combinations of optimal subgambles to be optimal in a mixture, whereas subtree perfectness requires only that any optimal subgamble is part of *at least one* optimal mixture.

Property 8 (Weak Multiple Mixture Property)**.** *For any non-empty event B and partition $A_1, \ldots, A_n$ such that $A_i \cap B \neq \emptyset$, and any non-empty finite sets of gambles*

$\mathcal{X}_i, \ldots, \mathcal{X}_n$ *such that* $\mathcal{X}_i$ *is* $A_i \cap B$*-consistent,*

- *if* $\bigoplus A_i X_i \in \operatorname{opt}(\bigoplus A_i \mathcal{X}_i | B)$*, then* $X_i \in \operatorname{opt}(\mathcal{X}_i | A_i \cap B)$*, and,*
- *if* $X_k \in \operatorname{opt}(\mathcal{X}_k | A_k \cap B)$*, then for each* $j \neq k$*, there is a* $X_j \in \mathcal{X}_j$ *such that* $\bigoplus A_i X_i \in \operatorname{opt}(\bigoplus A_i \mathcal{X}_i | B)$*.*

Theorem 2.12. *Let* $T = \bigodot E_i T_i$ *be a consistent decision tree where each of the* T_i *has a decision node as the root, and there is only one decision node in every branch. If* opt *satisfies Conditioning and Weak Multiple Mixture, then* $\operatorname{norm}_{\operatorname{opt}}$ *is subtree perfect for* T*.*

Proof. By the special structure of T, we just need to show that, for each $N \in \operatorname{ch}(T)$,

$$\operatorname{norm}_{\operatorname{opt}}(\operatorname{st}_N(T)) = \operatorname{st}_N(\operatorname{norm}_{\operatorname{opt}}(T)).$$

Suppose $U \in \operatorname{norm}_{\operatorname{opt}}(T_k)$ for some k, with $\operatorname{gamb}(U) = \{X\}$. We show that there is a $\bigodot_{i=1}^n E_i U_i \in \operatorname{norm}_{\operatorname{opt}}(T)$ with $U_k = U$. We know that

$$X \in \operatorname{opt}(\operatorname{gamb}(T_k) | E_k \cap \operatorname{ev}(T)),$$

and so by Weak Multiple Mixture there is a gamble

$$\bigoplus_{i=1}^n E_i X_i \in \operatorname{opt}(\operatorname{gamb}(T) | \operatorname{ev}(T))$$

with $X_k = X$. By definition of $\operatorname{norm}_{\operatorname{opt}}$, there is a $\bigodot_{i=1}^n E_i U_i \in \operatorname{norm}_{\operatorname{opt}}(T)$ with $U_k = U$.

Next, suppose there is a $V = \bigodot_{i=1}^n E_i U_i \in \operatorname{norm}_{\operatorname{opt}}(T)$. If $X_i \in \operatorname{gamb}(U_i)$ for each i, then $X_i \in \operatorname{opt}(\operatorname{gamb}(T_i) | E_i \cap \operatorname{ev}(T))$ for each i, by Weak Multiple Mixture. By Conditioning, $U_i \in \operatorname{norm}_{\operatorname{opt}}(T_i)$ for each i. □

It is natural to wonder whether there are larger trees that are subtree perfect under Weak Multiple Mixture rather than Mixture, for instance by taking trees that are subtree perfect by Theorem 2.12 and joining them together. The next theorem demonstrates subtree perfectness of any tree that has a chance node for a root, and all subtrees satisfy subtree perfectness. Note that this results requires not only Conditioning and Weak Multiple Mixture but also Path Independence (see Section 3.1.2).

Theorem 2.13. *If T has a root at a chance node, so $T = \bigodot E_i T_i$,* opt *satisfies Conditioning, Path Independence, and Weak Multiple Mixture, and* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for all subtrees of T, then* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for T.*

Proof. It suffices to show that subtree perfectness holds between T and any of its children T_i. If this is established, then subtree perfectness of each T_i can be invoked to arrive at the result.

Observe that for each i, we can find T_i' such that T_i' has only one decision node, at the root, and T_i' and T_i are strategically equivalent. So, if $U_i \in \mathrm{norm}_{\mathrm{opt}}(T_i)$, there is a $V_i \in \mathrm{norm}_{\mathrm{opt}}(T_i')$ with $\mathrm{gamb}(U_i) = \mathrm{gamb}(V_i)$.

If $U_1 \in \mathrm{norm}_{\mathrm{opt}}(T_1)$ then there is a $V_1 \in \mathrm{norm}_{\mathrm{opt}}(T_1')$ with $\mathrm{gamb}(U_1) = \mathrm{gamb}(V_1)$. By Theorem 2.12, there is a $V \in \mathrm{norm}_{\mathrm{opt}}(\bigodot E_i T_i')$ with $\mathrm{st}_{T_1'}(V) = V_1$. By strategic equivalence, there is a $U \in \mathrm{norm}_{\mathrm{opt}}(T)$ with $\mathrm{st}_{T_1}(U) = U_1$. This extends to any i.

So we have shown that if U_i is optimal in T_i it forms part of an optimal normal form decision in T. The proves one direction of subtree perfectness. Backward induction provides the other direction. □

2.5 Equivalence of Normal Form and Extensive Form

2.5.1 Equivalence for General Operators

A normal form solution specifies the subject's actions from the beginning. It can be argued that, if the problem has been posed as a sequential one (that is, as a decision tree with a decision node somewhere other than the root) then the solution should be of extensive form, since the subject in reality needs only to make decisions at decision nodes, not in advance. When a normal form operator admits a unique normal form decision, this is not a problem, since the normal form solution is then an extensive form solution anyway. Difficulties arise for indeterminate normal form solutions, because for most decision trees there are more possible normal form solutions than extensive form ones.

Consider the decision tree in Fig. 2.5. Suppose that the normal form solution

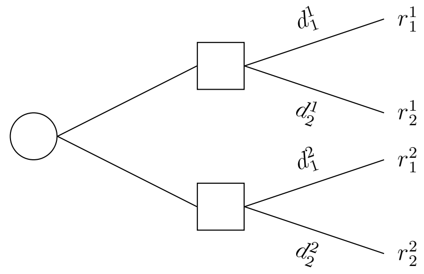

Figure 2.5: An example for non-correspondence of extensive and normal form solutions.

contains only two normal form decisions: namely $d_1^1 d_2^2$ and $d_2^1 d_1^2$. If we wanted to transform this solution into an extensive form solution, the only possibility would be to include all four decision arcs, thus returning the original tree. But suppose instead the normal form solution were that all four normal form decisions are optimal. Clearly transforming this solution to an extensive form solution would again return the whole tree.

A one-to-one correspondence between extensive form and normal form solutions therefore does not exist. We have seen that any sensible map from normal form solutions to extensive form solutions will not be injective. Is there a sensible injective map from extensive form solutions to normal form solutions? We would claim that there is. The interpretation of an extensive form solution was that the subject, upon reaching a decision node, picked an arc to follow without considering what she would pick at any different decision node. Therefore, the normal form solution corresponding to an extensive form solution should be *the set of normal form decisions of the extensive form solution.*

Definition 2.14. *If* $\mathrm{ext}(\cdot)$ *is an extensive form operator then its normal form correspondent is* $\mathrm{nfd}(\mathrm{ext}(\cdot))$*.*

Proposition 2.15. *If* ext_1 *and* ext_2 *are extensive form operators, and* T *is a decision tree such that* $\mathrm{nfd}(\mathrm{ext}_1(T)) = \mathrm{nfd}(\mathrm{ext}_2(T))$*, then* $\mathrm{ext}_1(T) = \mathrm{ext}_2(T)$*.*

Proof. We use proof by contradiction. Two extensive form solutions of the same decision tree can only differ if one, say $\mathrm{ext}_1(T)$ includes a decision arc that the other does not. In such a case, $\mathrm{nfd}(\mathrm{ext}_1(T))$ includes at least one normal form decision featuring this decision arc, whereas $\mathrm{nfd}(\mathrm{ext}_2(T))$ does not. Therefore, if the normal form correspondents are the same then the extensive form solutions must be the same. $\square$

So, we have a sensible way to move from an extensive form solution to a normal form solution, and this map is injective. What is the most sensible way to define the extensive form correspondent of a normal form solution? If a decision arc is in one of the normal form decisions then it should be in the extensive form solution, and if it is not in any of the normal form decisions then it should be absent from the extensive form solution.

Definition 2.16. *Let $\mathcal{T}$ be a normal form solution. Its extensive form correspondent is constructed as follows: a node N is in the extensive form correspondent if it is in at least one element of $\mathcal{T}$. If* norm *is a normal form operator then its extensive form correspondent is the extensive form operator that maps each T to the extensive form correspondent of* $\mathrm{norm}(T)$*.*

We are interested in the situation where operators norm and ext are equivalent. We define this as follows.

Definition 2.17. *A normal form operator* norm *and an extensive form operator* ext *are* equivalent *when each is the correspondent of the other.*

It is worth checking whether this condition of equivalence can ever be satisfied. In fact, by construction, moving from extensive form to normal form and then back will always return the original solution. So, in particular, an extensive form operator always has an equivalent normal form operator.

Lemma 2.18. *Let* ext *be an extensive form solution, let* norm *be its normal form correspondent, and let* ext^* *be the extensive form correspondent of* norm*. Then* $\mathrm{ext} = \mathrm{ext}^*$*.*

Proof. For any T, if N is a node in $\mathrm{ext}(T)$ then it is a node in an element of $\mathrm{nfd}(\mathrm{ext}(T)) = \mathrm{norm}(T)$ and so by definition is in $\mathrm{ext}^*(T)$. Similarly if N is not in $\mathrm{ext}(T)$ then it is not in $\mathrm{ext}^*(T)$. $\square$

It turns out that equivalence of operators is closely related to subtree perfectness.

Lemma 2.19. *Suppose that* ext *and* norm *are equivalent. Then,* norm *is subtree perfect if and only if* ext *is subtree perfect.*

Proof. "if". Suppose ext is subtree perfect and a node N is in $\mathrm{ext}(T)$, then N is in at least one element of $\mathrm{norm}(T)$. Because nfd and st_N commute, we have

$$\begin{aligned}\mathrm{norm}(\mathrm{st}_N(T)) &= \mathrm{nfd}(\mathrm{ext}(\mathrm{st}_N(T))) = \mathrm{nfd}(\mathrm{st}_N(\mathrm{ext}(T))) \\ &= \mathrm{st}_N(\mathrm{nfd}(\mathrm{ext}(T))) = \mathrm{st}_N(\mathrm{norm}(T)).\end{aligned}$$

This demonstrates subtree perfectness of norm.

"only if". By Lemma 2.18, for a particular norm there can be no more than one equivalent ext. We show that this ext is subtree perfect. A node N is in this ext if and only if it is in at least one element of $\mathrm{norm}(T)$. Similarly, a node M is in $\mathrm{st}_N(\mathrm{ext}(T))$ if and only if M is in at least one element of $\mathrm{st}_N(\mathrm{norm}(T))$. By subtree perfectness of norm, the latter is satisfied if and only if M is in at least one element of $\mathrm{norm}(\mathrm{st}_N(T))$. But, again by definition of ext, the latter is satisfied if and only if M is in $\mathrm{ext}(\mathrm{st}_N(T))$. This establishes subtree perfectness of ext. □

We know that a normal form operator may not have an equivalent extensive form representation. One might hope that if norm is subtree perfect then there is guarenteed to be an extensive form representation. Unfortunately this is not true. Consider again the tree in Fig. 2.5, and the first normal form solution we considered. Suppose that our normal form operator, when applied to the upper subtree only, returns both decisions as optimal, and does the same when applied to the lower subtree. Then by Definition 2.1, we have subtree perfectness. But of course there is no equivalent extensive form operator. This may suggest that subtree perfectness is not the strongest condition we could wish to impose.

Definition 2.20. *Consider a consistent decision tree T, a normal form operator* norm*, a normal form decision $U \in \mathrm{norm}(T)$, a node N such that N is in U, a normal form decision $V \in \mathrm{norm}(\mathrm{st}_N(T))$, and the normal form decision U^* such that $\mathrm{st}_N(U^*) = V$ and U^* coincides with U everywhere else. Then,* norm *is* strongly subtree perfect *if it is subtree perfect and, for any such* T*,* U*,* N*,* V*, we have* $U^* \in \mathrm{norm}(T)$*.*

Strong subtree perfectness says that any optimal strategy containing N can be amended by replacing the substrategy in $\mathrm{st}_N(T)$ with *any* optimal substrategy in $\mathrm{st}_N(T)$. It should not be too surprising that this condition leads to extensive form equivalence. For our purposes, however, the strengthening is not so interesting because, when restricting attention to normal form operators induced by choice functions, any subtree perfect operator has an extensive form equivalent; that is, strong subtree perfectness is implied by subtree perfectness for $\mathrm{norm}_{\mathrm{opt}}$.

2.5.2 Equivalence For Choice Functions

In this section we show that any subtree perfect normal form operator induced by a choice function has an equivalent extensive form solution. Choice functions that induce normal form operators that lack subtree perfectness may or may not have equivalent extensive form solutions; finding conditions for this remains an open problem.

Theorem 2.21. *If a normal form operator* $\mathrm{norm}_{\mathrm{opt}}$ *induced by a choice function* opt *is subtree perfect, then there exists an equivalent subtree perfect extensive form operator* ext*.*

Proof. By Lemma 2.19, if an equivalent ext exists then it is subtree perfect. Let ext be the extensive form correspondent of $\mathrm{norm}_{\mathrm{opt}}$ (see Definition 2.16). We must show that ext satisfies

$$\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{nfd}(\mathrm{ext}(T))$$

for all consistent decision trees T.

We now proceed by structural induction. The base step, that $\mathrm{nfd}(\mathrm{ext}(T)) = \mathrm{norm}_{\mathrm{opt}}(T)$ for any decision tree comprising only a single node, is as usual satisfied trivially.

Let us proceed with the induction step. The induction hypothesis states that, for any node K in $\mathrm{ch}(T)$, $\mathrm{nfd}(\mathrm{ext}(\mathrm{st}_K(T))) = \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_K(T))$. We must show that $\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{nfd}(\mathrm{ext}(T))$.

It is useful to show first that, if $\mathcal{K}$ is the set of all $K \in \mathrm{ch}(T)$ that appear in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$ (or equivalently, that appear in $\mathrm{ext}(T)$), then for any $K \in \mathcal{K}$,

$$\mathrm{st}_K(\mathrm{nfd}(\mathrm{ext}(T))) = \mathrm{st}_K(\mathrm{norm}_{\mathrm{opt}}(T)). \tag{2.9}$$

Consider any node K in $\mathrm{ch}(T)$ that appears in $\mathrm{ext}(T)$. Clearly,

$$\mathrm{st}_K(\mathrm{nfd}(\mathrm{ext}(T))) = \mathrm{nfd}(\mathrm{st}_K(\mathrm{ext}(T)))$$

and since we just proved that ext is subtree perfect,

$$= \mathrm{nfd}(\mathrm{ext}(\mathrm{st}_K(T)))$$

and by the induction hypothesis,

$$= \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_K(T))$$

but, by definition of ext, the node K also appears in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$, so by the subtree perfectness of $\mathrm{norm}_{\mathrm{opt}}$,

$$= \mathrm{st}_K(\mathrm{norm}_{\mathrm{opt}}(T)).$$

This establishes Eq. (2.9).

Now, suppose that the root of T is a decision node. Observe that

$$\mathrm{nfd}(\mathrm{ext}(T)) = \bigsqcup_{K \in \mathcal{K}} \mathrm{nfd}(\mathrm{st}_K(\mathrm{ext}(T))),$$

and since $\mathrm{st}(\cdot)$ and $\mathrm{nfd}(\cdot)$ commute,

$$= \bigsqcup_{K \in \mathcal{K}} \mathrm{st}_K(\mathrm{nfd}(\mathrm{ext}(T))).$$

Also, because opt satisfies Conditioning and Intersection, we have (as seen in the proof of Theorem 2.6)

$$\mathrm{norm}_{\mathrm{opt}}(T) = \bigsqcup_{K \in \mathcal{K}} \mathrm{st}_K(\mathrm{norm}_{\mathrm{opt}}(T)),$$

whence by Eq. (2.9),

$$\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{nfd}(\mathrm{ext}(T)).$$

Finally, suppose that the root of T is a chance node. Here, $\mathcal{K}$ is always simply $\operatorname{ch}(T) = \{K_1, \dots, K_n\}$. Similarly to before, we have

$$\begin{aligned}\operatorname{nfd}(\operatorname{ext}(T)) &= \bigodot_{i=1}^{n} E_i \operatorname{nfd}(\operatorname{st}_{K_i}(\operatorname{ext}(T))) \\ &= \bigodot_{i=1}^{n} E_i \operatorname{st}_{K_i}(\operatorname{nfd}(\operatorname{ext}(T))).\end{aligned}$$

Since opt satisfies Conditioning, Intersection, and Mixture, we have (as seen in the proof of Theorem 2.6),

$$\begin{aligned}\operatorname{norm}_{\text{opt}}(T) &= \bigodot_{i=1}^{n} E_i \operatorname{norm}_{\text{opt}}(\operatorname{st}_{K_i}(T)) \\ &= \bigodot_{i=1}^{n} E_i \operatorname{st}_{K_i}(\operatorname{norm}_{\text{opt}}(T)),\end{aligned}$$

whence by Eq. (2.9), we have

$$\operatorname{nfd}(\operatorname{ext}(T)) = \operatorname{norm}_{\text{opt}}(T).$$

□

One may wonder why this proof cannot be extended to normal form operators not induced by choice functions. It is because the proof relies on a consequence of Lemmas 2.4 and 2.8:

$$\operatorname{norm}_{\text{opt}}\left(\bigodot_{i=1}^{n} E_i T_i\right) = \bigodot_{i=1}^{n} E_i \operatorname{norm}_{\text{opt}}(T_i).$$

Subtree perfectness for $\operatorname{norm}_{\text{opt}}$ implies equality here, but for general operators equality need not hold.

So, given a subtree perfect choice function, there is an extensive form equivalent. Since the normal form solution induced by a choice function is usually inconvenient to find, perhaps the extensive form equivalent is more efficient. Recall the backward induction scheme for expected utility from Section 1.2.4. In this, at every stage we only need to retain from each decision node the maximum expected utility (and of course the decision arcs leading to this maximum). For an arbitrary choice function we cannot simply replace nodes with a number, but one would hope we could replace nodes with a single gamble (any one of the optimal normal form gambles at this stage), and that Intersection and Mixture ensure that no matter which of the gambles we choose, we arrive at the extensive form equivalent. It turns out that this approach always works. The investigation of this is delayed until Section 3.2, because it relies on various ideas of backward induction that have not yet been introduced.

2.6 Links to Other Work

As mentioned earlier, our results have strong links with the work of Hammond [24], Machina [50], and McClennen [51].

Machina [50] assumes probabilities at chance nodes and choice functions that correspond to total preorders. Under these assumptions, a necessary condition for subtree perfectness of $\mathrm{norm}_{\mathrm{opt}}$ is that opt satisfies *separability over mutually exclusive events*. Consider a partition with n events with probabilities $p_1, \ldots, p_n$, a gamble giving reward $r_i \in \mathcal{R}$ if event i obtains, and a second gamble that gives reward r_i if event i occurs for $i > 1$ and $r_* \in \mathcal{R}$ for $i = 1$. In other words, the two gambles differ on only one event. Machina's condition of separability says that the second gamble is preferred to the first if and only if r_* is preferred to r_1. Several conditions called separability exist, so we shall refer to this one as Machina-separability.

There are various ways to adapt separability for our more general setting. For example, consider a non-trivial event A and gambles X and Y such that $\overline{A}X = \overline{A}Y$, $AX = Ar_1$, and $AY = Ar_2$ for some rewards r_1 and $r_2 \in \mathcal{R}$. Machina-separability could be: $\mathrm{opt}(\{X, Y\}) = \{X\}$ if and only if $\mathrm{opt}(\{r_1, r_2\}) = \{r_1\}$. Note that Mixture implies this. Indeed, Mixture can be seen as a strong form of Machina-separability, for it implies every reasonable generalization of Machina-separability. Further, Multiple Mixture would be the strongest generalization abandoning a total preorder.

As already noted, Hammond's [24] results are slightly different from ours because of the definition of decision trees, and whether gambles involving probabilities are admitted. If such details are dealt with, then his results become very similar to ours. Using our terminology and notation, Hammond's first defines:

Definition 2.22. *An extensive form operator* ext *is* consistent *if*

$$\mathrm{ext}(\mathrm{st}_N(T)) = \mathrm{st}_N(\mathrm{ext}(T))$$

for any N in $\mathrm{ext}(T)$*.*

So, consistency is just another term for subtree perfectness.

Definition 2.23. *An extensive form operator* ext *is* consequentialist *if, for any decision trees T_1 and T_2 such that* $\mathrm{gamb}(T_1) = \mathrm{gamb}(T_2)$ *and* $\mathrm{ev}(T_1) = \mathrm{ev}(T_2)$*,*

$$\mathrm{gamb}(\mathrm{ext}(T_1)) = \mathrm{gamb}(\mathrm{ext}(T_2)).$$

In other words, consequentialism means respecting strategic equivalence. It is not entirely clear whether this is precisely what Hammond means by consequentialism, though. He writes that, following Anscombe [3, p. 9], consequentialism means that

> acts are valued by their consequences.

He then states that Definition 2.23 follows from this. Then, throughout the paper, he uses only Definition 2.23. But, while the definition is certainly a corollary of the above quotation, it is not equivalent.

Consider as an example Fig. 2.6, where U and V represent normal form decisions that form the continuation of the trees at these points. A possible extensive form solution removes the upper decision arc at the initial node, and removes nothing at the second node. It is easy to make this subtree perfect, and therefore Hammond-consistent, and it also satisfies Definition 2.23. But it does not "value acts by their consequences", since one branch corresponding to U is deleted yet the other is not.

This distinction is, admittedly, somewhat artificial, yet has important consequences for equivalence results. Clearly, the extensive form solution in this example cannot be equivalent to any normal form solution induced by a choice function, and so Definition 2.23 is insufficient to reproduce the results of Section 2.5. However, "acts are valued by their consequences" can be more carefully formulated into the condition we call *strong consequentialism*.

Definition 2.24. *An extensive form operator* ext *is* strongly consequentialist *if it is consequentialist and, for any $U \in \mathrm{nfd}(T)$ such that $\mathrm{gamb}(U) \subseteq \mathrm{gamb}(\mathrm{ext}(T))$, $U \in \mathrm{nfd}(\mathrm{ext}(T))$.*

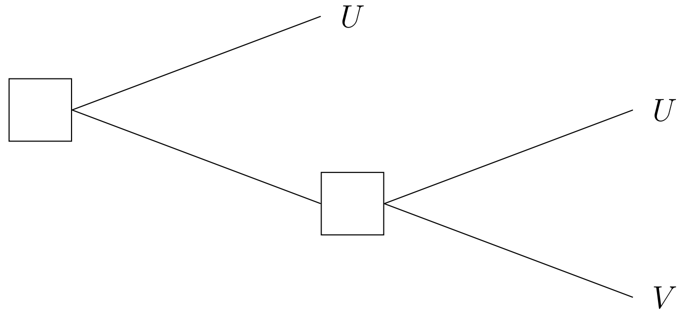

Figure 2.6: Example for the difference between Definition 2.23 and 2.24

Hammond argues that consistent and (not necessarily strongly) consequentialist extensive form operators induce a choice function on gambles as follows.

Definition 2.25. *For a consistent and consequentialist extensive form operator* ext*, define its corresponding choice function* $\mathrm{opt}_{\mathrm{ext}}$ *by*

$$\mathrm{opt}_{\mathrm{ext}}(\mathcal{X}|A) = \mathrm{gamb}(\mathrm{ext}(T)),$$

where T *is any consistent decision tree with* $\mathrm{gamb}(T) = \mathcal{X}$ *and* $\mathrm{ev}(T) = A$*. Because* ext *is consistent and consequentialist, this choice function exists and does not depend on the choice of* T*.*

With these definitions, we can prove a slightly stronger version of Hammond's results [24, Theorem 5.4, Theorem 6, Theorem 7, and Theorem 8].

Theorem 2.26. *A choice function* opt *satisfies Conditioning, Intersection, and Mixture if and only if there is a consistent and strongly consequentialist extensive form operator* ext *such that* $\mathrm{opt}_{\mathrm{ext}} = \mathrm{opt}$*.*

Proof. "if". Follow the approach of Hammond [24, Theorem 5.4 and Theorem 7]. Note that these proofs require only consequentialism.

"only if". Suppose opt satisfies Conditioning, Intersection, and Mixture. By Theorems 2.6 and 2.21, $\mathrm{norm}_{\mathrm{opt}}$ is subtree perfect and has an equivalent subtree perfect extensive form operator ext. By definition of $\mathrm{norm}_{\mathrm{opt}}$, for any strategically equivalent trees T_1 and T_2, $\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_2))$, and so the same holds for ext. Hence, ext is consistent and consequentialist. By construction, ext is also strongly consequentialist, and obviously also $\mathrm{opt}_{\mathrm{ext}} = \mathrm{opt}$. ∎

It is easily seen that, for a particular choice function opt, there is exactly one consistent and strongly consequentialist extensive form operator that induces opt. Therefore, there is an equivalence between the consistent and *strongly* consequentialist extensive form operator inducing opt and the subtree perfect normal form operator induced by opt. This equivalence is *not* present in Hammond's account, since multiple consistent and (*not strongly*) consequentialist extensive form operators can induce the same choice function.

As with Hammond, McClennen's [51] decision trees differ in that some chance nodes can have probabilities for events. Also, there seems to be no concept of conditioning in McClennen's account. As noted in Section 1.2.4, McClennen uses *dynamic normal form solutions*, which are generalizations of normal form solutions. Some of his results [51, Theorems 8.1 and 8.2] are similar to ours, and are based on three restrictions placed on his solutions.

Definition 2.27 (McClennen, [51, p. 120])**.** *A dynamic normal form solution satisfies* dynamic consistency *if, for every node N in the tree, the restriction of the optimal set at the root node to N is exactly the optimal set at N.*

There is a one-to-one correspondence between dynamic normal form solutions satisfying dynamic consistency, and our normal form solutions. Therefore $\mathrm{norm}_{\mathrm{opt}}$ implicitly satisfies dynamic consistency.

Definition 2.28 (McClennen,[51, p. 114])**.** *A dynamic normal form solution satisfies* plan reduction *if, for every normal form decision in T that induces the same gamble, either all or none of them are optimal.*

By definition, $\mathrm{norm}_{\mathrm{opt}}$ satisfies plan reduction.

Definition 2.29 (McClennen,[51, p. 122])**.** *A dynamic normal form solution satisfies* separability *if, for any tree T and any node N in T, the set of optimal plans at N is the same as the set of optimal plans of the separate tree $\mathrm{st}_N(T)$.*

On its own, separability is not exactly subtree perfectness, but if dynamic consistency holds then the two properties are equivalent. McClennen's two theorems can then be adapted into our setting as:

Theorem 2.30. *If a dynamic normal form solution satisfies plan reduction, dynamic consistency, and separability, then then it coincides with* $\mathrm{norm}_{\mathrm{opt}}$ *for a choice function* opt *satisfying Conditioning, Intersection, and Mixture.*

Proof. The proof is essentially identical to that of Lemma 2.10. □

Chapter 3

Backward Induction

Typically, one considers solution of decision trees by some method of backward induction. An example can be seen in Section 1.2.4. There are two principal reasons for such an approach. The first is computational efficiency: the total number of gambles in a decision tree increases exponentially with the tree depth, and so a method of eliminating some decisions at the right of the tree will lead to elimination of many decisions further to the left. This is especially useful if dealing with choice functions whose computational cost is worse than linear in the size of their argument. Then, many applications of the choice function to small sets will be more efficient than a single application to a very large set.

The second reason is more philosophical. When considering what to do in the present, one is likely to think of future decision points and think what one would expect to do at these. Anything one thinks will be rejected in the future should not be part of one's present plan: why intend to take an action that one will want to reject upon reaching it? It seems that backward induction is a natural way to think about sequential decision problems.

In most of the literature [78, 46, 56, 7, 41, 64] (but notably not Kikuti et al. [38], whose ideas we later follow), backward induction schemes are essentially presented as subtree perfect extensive form solutions. We address these for general choice functions in Section 3.2, but first we examine backward induction as a means of finding (potentially subtree imperfect) normal form solutions.

3.1 Normal Form Backward Induction

3.1.1 Definition

Although the operator $\mathrm{norm}_{\mathrm{opt}}$ is a simple and popular way of defining a normal form solution, it has a major practical difficulty. The set of normal form decisions associated with T grows at least exponentially with the size of the tree, and so $\mathrm{gamb}(T)$ may have many elements. For example, if a tree T has at least n decision nodes in every path from the root to any leaf, and each decision node has at least two children, then there will be at least 2^n normal form decisions associated with T (and often a lot more). Moreover, some choice functions are computationally expensive when applied to very large sets. For instance, maximality with respect to a lower prevision (discussed further in Chapter 6) requires us to solve $\frac{k(k-1)}{2}$ linear programs for a set of k decisions. Hence, direct calculation of $\mathrm{norm}_{\mathrm{opt}}$ will usually be impractical, if not impossible. For this reason, we suggest the following backward induction method. First, for notational convenience we extend $\mathrm{norm}_{\mathrm{opt}}$ to sets of trees.

Definition 3.1. *Given a choice function* opt *and any set* $\mathcal{T}$ *of consistent decision trees, where* $\mathrm{ev}(T) = A$ *for all* $T \in \mathcal{T}$*,*

$$\mathrm{norm}_{\mathrm{opt}}(\mathcal{T}) = \{U \in \mathrm{nfd}(\mathcal{T})\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A)\}.$$

Care should be taken over this definition. It can be understood in the following way: if there is a set of decision trees $\mathcal{T}$ and we must choose one tree and one strategy in that tree, then $\mathrm{norm}_{\mathrm{opt}}(\mathcal{T})$ should be our optimal set. With this extension, there is a corresponding result to Eq. (1.10) for sets of trees: if $\mathcal{T}$ is a set of consistent decisions trees where $\mathrm{ev}(T) = A$ for all $T \in \mathcal{T}$, then

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(\mathcal{T})) = \mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A). \tag{3.1}$$

We can now define our backward induction algorithm through a normal form operator $\mathrm{back}_{\mathrm{opt}}$. This is a formalization of an algorithm by Kikuti et al. [38].

Definition 3.2. *Let* $\mathrm{back}_{\mathrm{opt}}$ *be a normal form operator defined for any consistent decision tree* T *through the following identities:*

- *If a tree T consists of only a leaf with reward $r \in \mathcal{R}$, then*

$$\mathrm{back}_{\mathrm{opt}}(T) = \{T\}. \tag{3.2a}$$

- *If a tree T has a chance node as root, that is, $T = \bigodot_{i=1}^{n} E_i T_i$, then*

$$\mathrm{back}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i T_i\right) = \mathrm{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}}\,(T_i)\right) \tag{3.2b}$$

- *If a tree T has a decision node as root, that is, if $T = \bigsqcup_{i=1}^{n} T_i$, then*

$$\mathrm{back}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} T_i\right) = \mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{back}_{\mathrm{opt}}(T_i)\right). \tag{3.2c}$$

It is instructive to compare the definition of the gamb operator (Definition 1.12) with the above definition of $\mathrm{back}_{\mathrm{opt}}$. The main difference is that $\mathrm{back}_{\mathrm{opt}}$ inserts $\mathrm{norm}_{\mathrm{opt}}$ at every stage of the recursion, to eliminate as many normal form decisions as possible, early on.

If $\mathrm{back}_{\mathrm{opt}}$ always yields the same normal form solution as $\mathrm{norm}_{\mathrm{opt}}$, we can use the former as an efficient way of calculating the latter. Of course, such a procedure only works if a normal form gamble that is non-optimal in a subtree at a node cannot be part of an optimal gamble in the full tree. It is well known that choice functions exist for which this property does not hold: for examples, see LaValle and Wapman [42], Jaffray [32], and Seidenfeld [64]. In such cases, there exist trees such that $\mathrm{back}_{\mathrm{opt}}(T) \neq \mathrm{norm}_{\mathrm{opt}}(T)$, and our approach cannot be used.

We now investigate necessary and sufficient properties of choice functions for $\mathrm{back}_{\mathrm{opt}}(T)$ to coincide with $\mathrm{norm}_{\mathrm{opt}}(T)$ for any consistent decision tree T. Readers wishing first to see an example of $\mathrm{back}_{\mathrm{opt}}$ in action before continuing should proceed to Chapter 6, in particular Sections 6.1 and 6.3.

3.1.2 Backward Induction Properties

As with subtree perfectness, the behaviour of $\mathrm{back}_{\mathrm{opt}}$ can be understood by several conditions on choice functions. Perhaps unsurprisingly, they turn out to be closely related to but weaker than the subtree perfectness properties.

Property 9 (Backward conditioning property)**.** *Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, and let $\mathcal{X}$ be a non-empty finite $A \cap B$-consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$ and $X \in \operatorname{opt}(\mathcal{X}|A \cap B)$. If there is an $\overline{A} \cap B$-consistent gamble Z such that*

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B),$$

then $Y \in \operatorname{opt}(\mathcal{X}|A \cap B)$.

Property 10 (Insensitivity of optimality to the omission of non-optimal elements)**.** *For any event $A \neq \emptyset$, and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$,*

$$\operatorname{opt}(\mathcal{X}|A) \subseteq \mathcal{Y} \subseteq \mathcal{X} \Rightarrow \operatorname{opt}(\mathcal{Y}|A) = \operatorname{opt}(\mathcal{X}|A).$$

Property 11 (Preservation of non-optimality under the addition of elements)**.** *For any event $A \neq \emptyset$, and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$,*

$$\mathcal{Y} \subseteq \mathcal{X} \Rightarrow \operatorname{opt}(\mathcal{Y}|A) \supseteq \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

Property 12 (Backward mixture property)**.** *For any events A and B such that $B \cap A \neq \emptyset$ and $B \cap \overline{A} \neq \emptyset$, any $B \cap \overline{A}$-consistent gamble Z, and any non-empty finite $B \cap A$-consistent set of gambles $\mathcal{X}$,*

$$\operatorname{opt}\left(A\mathcal{X} \oplus \overline{A}Z|B\right) \subseteq A \operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

If opt satisfies Insensitivity To Omission, then removing non-optimal elements from a set does not affect whether or not each of the remaining elements is optimal. The property is called 'insensitivity to the omission of non-optimal elements' by De Cooman and Troffaes [13], and 'property ϵ' by Sen [66] who attributes this designation to Douglas Blair. It seems a very natural condition to impose on a choice function, but is not seen so often in the literature, perhaps because it is a consequence of the more popular 'property β' [66, p. 65].

Preservation Under Addition is called 'property α' by Sen [66, p. 64], Axiom 7 by Luce and Raiffa [46, p. 288], and 'independence of irrelevant alternatives' by

Radner and Marschak [55].[1] It states that any gamble that is non-optimal in a set of gambles $\mathcal{Y}$ is non-optimal in any set of gambles containing $\mathcal{Y}$; in other words, adding new gambles cannot make previously non-optimal gables become optimal.

Insensitivity To Omission and Preservation Under Addition are together equivalent to another property, called path independence. Compare this property with Strong Path Independence and Very Strong Path Independence from Section 2.2.

Property 13 (Path independence)**.** *For any non-empty event A, and for any finite family of non-empty finite A-consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$,*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n}\mathcal{X}_i\middle|A\right)=\operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{opt}(\mathcal{X}_i|A)\middle|A\right).$$

Lemma 3.3 (Sen [66, Proposition 17])**.** *A choice function* opt *satisfies Preservation Under Addition if and only if, for any non-empty event A and any finite family of non-empty finite A-consistent sets of gambles $\mathcal{X}_1, \dots \mathcal{X}_n$,*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n}\mathcal{X}_i\middle|A\right)\subseteq\operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{opt}(\mathcal{X}_i|A)\middle|A\right)\subseteq\bigcup_{i=1}^{n}\operatorname{opt}(\mathcal{X}_i|A).$$

Lemma 3.4 (Sen [66, Proposition 19])**.** *A choice function* opt *satisfies Insensitivity To Omission and Preservation Under Addition if and only if* opt *satisfies Path Independence.*

Insensitivity To Omission, Preservation Under Addition, and Path Independence are expressed slightly differently here than in Sen [66], who does not use the concepts of conditioning and consistency. Despite this, the proofs of Lemmas 3.3 and 3.4 proceed identically to the corresponding propositions by Sen. Also, Sen defines path independence only for pairs of subsets ($n = 2$), but Plott [53, Theorem 1, p. 1082] shows that path independence for $n = 2$ is indeed equivalent to Path Independence.

Path independence appears frequently in the social choice literature. Plott [53] gives a detailed investigation of path independence and its possible justifications. Path independence perhaps is most easily understood by reference to Axiom $7'$ of Luce and Raiffa [46, p. 289]:

[1]This is different from several other properties bearing the same name, such as that of Arrow's Impossibility Theorem. For further discussion, see Ray [58].

> The addition of new acts does not transform an old, originally non-optimal act into an optimal one, and it can change an old, originally optimal act into a non-optimal one only if at least one of the new acts is optimal.

We can formulate this axiom mathematically as follows.

Property 14 (Luce and Raiffa's Axiom $7'$)**.** *For any non-empty event A and any non-empty finite sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$,*

$$\operatorname{opt}(\mathcal{Y}|A) \supseteq \mathcal{Y} \cap \operatorname{opt}(\mathcal{X}|A)$$

and

$$\operatorname{opt}(\mathcal{X}|A) \not\subset \operatorname{opt}(\mathcal{Y}|A).$$

Proposition 3.5. *Path Independence and Axiom $7'$ are equivalent.*

Proof. First, note that the first condition is equivalent to Preservation Under Addition. Now, suppose that opt satisfies Insensitivity To Omission. We have that $\operatorname{opt}(\mathcal{X}|A) = \operatorname{opt}(\mathcal{Y}|A)$ and so the second condition follows.

Now suppose that opt satisfies Axiom $7'$, and suppose that $\operatorname{opt}(\mathcal{X}|A) \subseteq \mathcal{Y} \subseteq \mathcal{X}$. By the first condition we know that $\operatorname{opt}(\mathcal{Y}|A)$ contains $\operatorname{opt}(\mathcal{X}|A)$. But by the second condition we know that $\operatorname{opt}(\mathcal{X}|A)$ is not a strict subset of $\operatorname{opt}(\mathcal{Y}|A)$. The only possibility remaining is $\operatorname{opt}(\mathcal{X}|A) = \operatorname{opt}(\mathcal{Y}|A)$, so Insensitivity To Omission holds.

Finally, Lemma 3.4 completes the proof. □

Backward Conditioning only differs from Conditioning by Z. The mysterious appearance of this gamble is explained in Lemma 3.17. In the presence of Preservation Under Addition, Backward Conditioning can be strengthened to the following property, which proves more useful in the proofs.

Property 15 (Strong backward conditioning property)**.** *For any events A and B such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$ and any non-empty finite set $A \cap B$-consistent set of gambles $\mathcal{X}$ with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$, $X \in \operatorname{opt}(\mathcal{X}|A \cap B)$ implies $Y \in \operatorname{opt}(\mathcal{X}|A \cap B)$ whenever there is a non-empty finite $\overline{A} \cap B$-consistent set of gambles $\mathcal{Z}$ such that, for at least one $Z \in \mathcal{Z}$,*

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B).$$

Lemma 3.6. *Suppose* opt *satisfies Preservation Under Addition. Then* opt *satisfies Backward Conditioning if and only if* opt *satisfies Strong Backward Conditioning.*

Proof. "only if". If a Z exists, take $\mathcal{Z} = \{Z\}$, and the result is immediate. Otherwise, consider any non-empty finite set of $\overline{A} \cap B$-consistent gambles $\mathcal{Z}$. By Preservation Under Addition,

$$\begin{aligned}\operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B) &= \operatorname{opt}\left(\bigcup_{Z\in\mathcal{Z}} A\mathcal{X} \oplus \overline{A}Z \middle| B\right) \\ &\subseteq \operatorname{opt}\left(\bigcup_{Z\in\mathcal{Z}} \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B) \middle| B\right)\end{aligned}$$

Since there is no Z such that $AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$, there is no $\mathcal{Z}$ and $Z \in \mathcal{Z}$ such that $AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B)$. Therefore Strong Backward Conditioning holds.

"if". Suppose $\mathcal{Z}$ is a set of the required form in Strong Backward Conditioning, and X and Y are the gambles in question. It is clear that if a required $\mathcal{Z}$ does not exist then Backward Conditioning follows trivially. Let $Z \in \mathcal{Z}$ be a gamble such that

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B). \tag{3.3}$$

We know $A\mathcal{X} \oplus \overline{A}Z \subseteq A\mathcal{X} \oplus \overline{A}\mathcal{Z}$, and so by Preservation Under Addition,

$$\operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B) \cap (A\mathcal{X} \oplus \overline{A}Z) \subseteq \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B).$$

By Eq. (3.3) it follows that

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B).$$

□

Lemma 3.7. *If* opt *satisfies Strong Backward Conditioning and Mixture, then* opt *satisfies Conditioning.*

Proof. First, observe that if $AX = AY$ then $(A \cap B)X = (A \cap B)Y$, and so Strong Backward Conditioning implies Conditioning whenever a suitable $\mathcal{Z}$ exists. We show that Mixture guarantees the existence of such a $\mathcal{Z}$.

By Mixture, for any $\overline{A} \cap B$-consistent gamble Z,

$$\operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B) = A\operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z,$$

and so $AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. Take $\mathcal{Z} = \{Z\}$. □

Backward Mixture is just Mixture but with inclusion rather than equality. We do not need the equality because backward induction moves from right to left in the tree. Equivalently, the property is a form of the Independence Axiom in which only one direction of implication is required. We have not encountered such a property elsewhere in the literature, perhaps because it seems to have no justification other than ensuring that backward induction works.

As with Mixture, there is an extension to arbitrary partitions of Ω. And again, this extension is implied by the combination of several of the properties already introduced.

Property 16 (Multiple Backward Mixture Property)**.** *For any partition of Ω A_1, …, A_n, any non-empty event B such that $A_i \cap B \neq \emptyset$ for all A_i, and for any non-empty finite sets of gambles $\mathcal{X}_1$, …, $\mathcal{X}_n$ where each $\mathcal{X}_i$ is $A_i \cap B$-consistent,*

$$\operatorname{opt}\left(\bigoplus_{i=1}^{n} A_i\mathcal{X}_i \middle| B\right) \subseteq \bigoplus_{i=1}^{n} A_i \operatorname{opt}(\mathcal{X}_i|A_i \cap B). \tag{3.4}$$

Lemma 3.8. *Let A_1, …, A_n be a finite partition of Ω. Let B be any event such that $A_i \cap B \neq \emptyset$ for all A_i. Let $\mathcal{X}_1$, …, $\mathcal{X}_n$ be a finite family of non-empty finite sets of gambles where each $\mathcal{X}_i$ is $A_i \cap B$-consistent. If a choice function* opt *satisfies Preservation Under Addition and Backward Mixture, then* opt *satisfies Multiple Backward Mixture.*

Proof. Let $\mathcal{X} = \bigoplus_{i=1}^{n} A_i\mathcal{X}_i$. Consider any $k \in \{1, \dots, n\}$ and let

$$\mathcal{Z}_k = \bigoplus_{j \neq k} A'_j\mathcal{X}_j$$

where $(A'_j)_{j\neq k}$ forms an arbitrary partition of Ω such that $\overline{A}_k \cap A'_j = A_j$ for all $j \neq k$. Clearly, $\mathcal{Z}_k$ is $\overline{A}_k \cap B$-consistent because we can trivially find a consistent decision tree T with $\operatorname{ev}(T) = \overline{A}_k \cap B$ and $\operatorname{gamb}(T) = \mathcal{Z}_k$, using the $A_j \cap B$-consistency (and hence, $\overline{A}_k \cap A'_j \cap B$-consistency) of each $\mathcal{X}_j$ for $j \neq k$.

Now, observe that by construction of $\mathcal{Z}_k$,

$$\mathcal{X} = A_k\mathcal{X}_k \oplus \overline{A}_k\mathcal{Z}_k = \bigcup_{Z_k\in\mathcal{Z}_k} (A_k\mathcal{X}_k \oplus \overline{A}_k Z_k).$$

Note that $\mathcal{X}$ is B-consistent (indeed, because each $\mathcal{X}_i$ is $A_i \cap B$-consistent, we can trivially find a consistent decision tree T with $\mathrm{ev}(T) = B$ and $\mathrm{gamb}(T) = \mathcal{X}$).

If we apply $\mathrm{opt}(\cdot|B)$ on both sides of the above equality, then it follows from Lemma 3.3 that

$$\mathrm{opt}(\mathcal{X}|B) \subseteq \bigcup_{Z_k\in\mathcal{Z}_k} \mathrm{opt}\left(A_k\mathcal{X}_k \oplus \overline{A}_k Z_k|B\right)$$

and by Backward Mixture (once noted that $\mathcal{X}_k$ is $A_k \cap B$-consistent by assumption, and Z_k is $\overline{A}_k \cap B$-consistent by construction),

$$\begin{aligned} &\subseteq \bigcup_{Z_k\in\mathcal{Z}_k} (A_k \,\mathrm{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k Z_k) \\ &= A_k \,\mathrm{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k \mathcal{Z}_k \end{aligned}$$

whence by Lemma B.4,

$$\mathrm{opt}(\mathcal{X}|B) \subseteq \bigoplus_{i=1}^{n} A_i \,\mathrm{opt}(\mathcal{X}_i|A_i \cap B).$$

□

Lemma 3.9. *If a choice function* opt *satisfies Insensitivity To Omission, Preservation Under Addition, and Backward Mixture, then*

$$\mathrm{opt}\left(\bigoplus_{i=1}^{n} A_i\mathcal{X}_i \middle| B\right) = \bigoplus_{i=1}^{n} A_i \,\mathrm{opt}(\mathcal{X}_i|A_i \cap B). \tag{3.5}$$

Proof. By Lemma 3.8 and the definition of opt,

$$\mathrm{opt}\left(\bigoplus_{i=1}^{n} A_i\mathcal{X}_i \middle| B\right) \subseteq \bigoplus_{i=1}^{n} A_i \,\mathrm{opt}(\mathcal{X}_i|A_i \cap B) \subseteq \bigoplus_{i=1}^{n} A_i\mathcal{X}_i,$$

whence Eq. (3.5) follows by Insensitivity To Omission. □

As in Section 2.2, we can ask whether the properties are indeed distinct. From the work in that section it ought to be clear that the only question worth considering is whether Insensitivity To Omission and Preservation Under Addition are distinct. The simplest situation where they are distinct is when there are only three gambles available.

Example 3.10. *Suppose there are three possible gambles, X, Y, Z. Suppose we have*

$$\begin{array}{c|cccc} & X,Y & X,Z & Y,Z & X,Y,Z \\ \text{opt} & X & X,Z & Y,Z & X,Y,Z \end{array}$$

This satisfies Insensitivity To Omission. But set $\mathcal{Y} = \{X, Y\}$ and $\mathcal{X} = \{X, Y, Z\}$ and Preservation Under Addition fails.

Example 3.11. *Suppose there are three possible gambles, X, Y, Z. Suppose we have*

$$\begin{array}{c|cccc} & X,Y & X,Z & Y,Z & X,Y,Z \\ \text{opt} & X,Y & X,Z & Y,Z & X \end{array}$$

This satisfies Preservation Under Addition. But set $\mathcal{Y} = \{X, Y\}$ and $\mathcal{X} = \{X, Y, Z\}$ and Insensitivity To Omission fails.

We now turn to our main result, and characterize precisely when $\mathrm{back}_{\mathrm{opt}}$ gives $\mathrm{norm}_{\mathrm{opt}}$ for all consistent decision trees.

3.1.3 Backward Induction Theorem

Theorem 3.12 (Backward induction theorem)**.** *Let* opt *be any choice function. The following conditions are equivalent.*

(A) *For any consistent decision tree T, it holds that* $\mathrm{back}_{\mathrm{opt}}(T) = \mathrm{norm}_{\mathrm{opt}}(T)$.

(B) opt *satisfies Backward Conditioning, Insensitivity To Omission, Preservation Under Addition, and Backward Mixture.*

The proof of this theorem, while long, is very similar to that for Theorem 2.6. The required lemmas are also very similar, and so their proofs have been moved to Appendix B.3.

Lemma 3.13. *For any consistent decision tree $T = \bigodot_{i=1}^n E_i T_i$ and any choice function* opt *satisfying Backward Conditioning,*

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{gamb}\left(\mathrm{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i\,\mathrm{norm}_{\mathrm{opt}}(T_i)\right)\right) \tag{3.6}$$

implies

$$\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i\,\mathrm{norm}_{\mathrm{opt}}(T_i)\right).$$

Lemma 3.14. *For any consistent decision tree $T = \bigsqcup_{i=1}^n T_i$, and any choice function* opt *satisfying Preservation Under Addition,*

$$\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)) = \operatorname{gamb}\left(\operatorname{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^n \operatorname{norm}_{\mathrm{opt}}(T_i)\right)\right) \tag{3.7}$$

implies

$$\operatorname{norm}_{\mathrm{opt}}(T) = \operatorname{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^n \operatorname{norm}_{\mathrm{opt}}(T_i)\right).$$

Lemma 3.15. *If* $\operatorname{back}_{\mathrm{opt}}(T) = \operatorname{norm}_{\mathrm{opt}}(T)$ *for any consistent decision tree T, then* opt *satisfies Backward Conditioning.*

Lemma 3.16. *If* $\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T)) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T))$ *for any consistent decision tree T, then* opt *satisfies Path Independence.*

Lemma 3.17. *If* $\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T)) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T))$ *for any consistent decision tree T, then* opt *satisfies Backward Mixture.*

Proof of Theorem 3.12. (A) $\Longrightarrow$ (B). By Lemmas 3.15, 3.16, and 3.17, we see that (A) implies Backward Conditioning, Backward Mixture, and Path Independence. Lemma 3.4 completes the proof.

(B) $\Longrightarrow$ (A). We prove this part by structural induction on the tree. In the base step, we prove that the implication holds for consistent decision trees that consist of only a single node. In the induction step, we prove that if the implication holds for the subtrees at every child of the root node, then the implication also holds for the whole tree.

First, if the decision tree T has only a single node, and hence, a reward at the root and no further children, then by definition (Eq. (1.6a) in particular) we have $\operatorname{back}_{\mathrm{opt}}(T) = \operatorname{norm}_{\mathrm{opt}}(T)$.

Next, suppose T is consistent and has a chance node as its root: $T = \bigodot_{i=1}^n E_i T_i$. By the induction hypothesis, we know that for every T_i,

$$\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_i)) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T_i)). \tag{3.8}$$

We show that $\operatorname{back}_{\mathrm{opt}}(T) = \operatorname{norm}_{\mathrm{opt}}(T)$. By Lemma 3.13, it therefore suffices to show that $\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)) = \operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T))$. By Eq. (1.10) and the definition of gamb,

$$\begin{aligned}
\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)) &= \operatorname{opt}(\operatorname{gamb}(T)|\operatorname{ev}(T)) \\
&= \operatorname{opt}\left(\operatorname{gamb}\left(\bigodot_{i=1}^{n} E_i T_i\right)\middle|\operatorname{ev}(T)\right) \\
&= \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right),
\end{aligned}$$

and by Eq. (3.8), Eq. (1.10), and the definition of gamb,

$$\begin{aligned}
\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T)) &= \operatorname{gamb}\left(\operatorname{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \operatorname{back}_{\mathrm{opt}}(T_i)\right)\right) \\
&= \operatorname{opt}\left(\operatorname{gamb}\left(\bigodot_{i=1}^{n} E_i \operatorname{back}_{\mathrm{opt}}(T_i)\right)\middle|\operatorname{ev}(T)\right) \\
&= \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_i))\middle|\operatorname{ev}(T)\right) \\
&= \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T_i))\middle|\operatorname{ev}(T)\right) \\
&= \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T)\cap E_i)\middle|\operatorname{ev}(T)\right),
\end{aligned}$$

whence equality follows from Backward Mixture and Preservation Under Addition, and Lemma 3.9.

Finally, suppose that the root of the consistent tree T is a decision node, that is $T = \bigsqcup_{i=1}^{n} T_i$. We show that $\operatorname{back}_{\mathrm{opt}}(T) = \operatorname{norm}_{\mathrm{opt}}(T)$. By Lemma 3.14, it suffices to show that $\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T)) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T))$. Indeed,

$$\begin{aligned}
\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)) &= \operatorname{opt}(\operatorname{gamb}(T)|\operatorname{ev}(T)) \\
&= \operatorname{opt}\left(\operatorname{gamb}\left(\bigsqcup_{i=1}^{n} T_i\right)\middle|\operatorname{ev}(T)\right) \\
&= \operatorname{opt}\left(\bigcup_{i=1}^{n} \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right),
\end{aligned}$$

and,

$$\begin{aligned}
\operatorname{gamb}\left(\operatorname{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n}\operatorname{back}_{\mathrm{opt}}(T_i)\right)\right) &= \operatorname{opt}\left(\operatorname{gamb}\left(\bigsqcup_{i=1}^{n}\operatorname{back}_{\mathrm{opt}}(T_i)\right)\middle|\operatorname{ev}(T)\right)\\
&= \operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_i))\middle|\operatorname{ev}(T)\right)\\
&= \operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T_i))\middle|\operatorname{ev}(T)\right)\\
&= \operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T_i))\middle|\operatorname{ev}(T)\right)\\
&= \operatorname{opt}\left(\bigcup_{i=1}^{n}\operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T))\middle|\operatorname{ev}(T)\right),
\end{aligned}$$

whence equality follows from Insensitivity To Omission and Preservation Under Addition, and Lemma 3.4.

Concluding, we have shown that the implication holds for consistent decision trees consisting of a single nodes, and that if the implication holds for all children of the root node then it also holds for the whole tree. By induction, the implication holds for any consistent decision tree. □

Theorems 1.16 and 3.12 together imply the following corollary.

Corollary 3.18. *If* opt *satisfies Backward Conditioning, Insensitivity To Omission, Preservation Under Addition, and Backward Mixture, then for any strategically equivalent consistent decision trees* T_1 *and* T_2 *with* $\operatorname{ev}(T_1) = \operatorname{ev}(T_2)$*, it holds that* $\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_1)) = \operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_2))$*.*

It should be noted that even if $\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_1)) = \operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T_2))$ for all strategically equivalent trees, Backward Conditioning may not hold. It does imply the other three properties, however, as the following weakening of Theorem 3.12 shows.

Theorem 3.19. *Let* opt *be any choice function. The following conditions are equivalent.*

(A) For any consistent decision tree T*, it holds that*

$$\operatorname{gamb}(\operatorname{back}_{\mathrm{opt}}(T)) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)).$$

(B) opt *satisfies Insensitivity To Omission, Preservation Under Addition, and Backward Mixture.*

(C) For any strategically equivalent consistent decision trees T_1 and T_2 such that $\mathrm{ev}(T_1) = \mathrm{ev}(T_2)$, *it holds that*

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)).$$

Proof. (A) $\implies$ (B). This follows from Lemmas 3.16 and 3.17.

(B) $\implies$ (A). Follow the same method as in the corresponding part of the proof of Theorem 3.12, but using Eq. (3.8) as the induction hypothesis rather than $\mathrm{back}_{\mathrm{opt}}(T_i) = \mathrm{norm}_{\mathrm{opt}}(T_i)$. In the proof of Theorem 3.12, we arrive at

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T)) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T))$$

using only Eq. (3.8) and Insensitivity To Omission, Preservation Under Addition, and Backward Mixture.

(A) $\implies$ (C). This follows immediately from Theorem 1.16.

(C) $\implies$ (A). We show that for any consistent T_1, we can find a strategically equivalent T_2 with $\mathrm{ev}(T_1) = \mathrm{ev}(T_2)$ and $\mathrm{back}_{\mathrm{opt}}(T_2) = \mathrm{norm}_{\mathrm{opt}}(T_2)$. Then the result will follow from Theorem 1.16. Indeed, let T_2 be a consistent decision tree with $\mathrm{ev}(T_2) = \mathrm{ev}(T_1)$, $\mathrm{gamb}(T_2) = \mathrm{gamb}(T_1)$, and only one decision node (at the root). Theorem 1.18 assures us of the existence of T_2 (see the tree defined in Eq. (1.11), with $\mathcal{X} = \mathrm{gamb}(T_1)$). By definition of $\mathrm{back}_{\mathrm{opt}}$, it follows that $\mathrm{back}_{\mathrm{opt}}(T_2) = \mathrm{norm}_{\mathrm{opt}}(T_2)$.

Now, by definition, $\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_2))$. Recall that we have $\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2))$ by assumption. Therefore

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1))$$

as required. $\square$

This weakening could be useful, because the equality $\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{back}_{\mathrm{opt}}(T)$ may be considered unnecessarily restrictive in practice. In Theorem 3.19, a normal form decision $U \in \mathrm{norm}_{\mathrm{opt}}(T)$ may not be present in $\mathrm{back}_{\mathrm{opt}}(T)$, but there is a $V \in \mathrm{back}_{\mathrm{opt}}(T)$ with $\mathrm{gamb}(V) = \mathrm{gamb}(U)$. Since choosing V would give exactly the same reward as U for all outcomes, from a practical perspective it may not matter that U has been eliminated. Further, when using normal form backward induction in practice, one would presumably carry back the optimal gambles rather than the optimal normal form decisions from each stage, and only upon reaching the root transform back to normal form decisions. With such a method, Theorem 3.19 and Definition 1.15 assure us that $\mathrm{norm}_{\mathrm{opt}}$ will be obtained.

3.1.4 Relationship with Subtree Perfectness

It is unsurprising, given the similarities between the properties, that backward induction is closely related to subtree perfectness. Obviously, Conditioning implies Backward Conditioning, and Mixture implies Backward Mixture. Also, it is easily shown that:

Lemma 3.20. *Intersection implies Insensitivity To Omission and Preservation Under Addition.*

Hence, from Theorems 2.6 and 3.12, we can immediately conclude that subtree perfectness is sufficient for backward induction.

Corollary 3.21. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect, then* $\mathrm{norm}_{\mathrm{opt}} = \mathrm{back}_{\mathrm{opt}}$.

Subtree perfectness is, however, not necessary for backward induction. For example, it is easy to see that point-wise dominance satisfies Conditioning, Insensitivity To Omission, Preservation Under Addition, and Backward Mixture, but as we saw in Example 2.2, it lacks subtree perfectness.

Backward induction does, however, imply a weaker form of subtree perfectness. Suppose $\mathrm{back}_{\mathrm{opt}}(T) = \mathrm{norm}_{\mathrm{opt}}(T)$ for all consistent decision trees. Then, by definition of $\mathrm{back}_{\mathrm{opt}}$, for any node N such that N is in at least one element of $\mathrm{back}_{\mathrm{opt}}(T)$ we have

$$\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{st}_N(\mathrm{back}_{\mathrm{opt}}(T)) \subseteq \mathrm{back}_{\mathrm{opt}}(\mathrm{st}_N(T)) = \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_N(T)).$$

Why can this be seen as a type of subtree perfectness? Recall that a subgame perfect equilibrium point is one that induces an equilibrium point in all subgames.

According to our definition earlier, a subtree perfect normal form operator is one that induces its normal form solution in all subtrees. If the conditions of the backward induction theorem are satisfied, then every optimal normal form *decision* induces an optimal normal form decision in any subtree. That is, although the restriction of the solution to a subtree is not necessarily the solution to the restriction, the restriction of every element of the solution is an element of the solution of the restriction. So while we do not have subtree perfectness of *solutions* we do have subtree perfectness of *decisions*.

Definition 3.22. *A normal form operator* norm *is* subtree perfect for normal form decisions *if for every N in T such that N appears in at least one element of* $\mathrm{norm}(T)$,

$$\mathrm{st}_N(\mathrm{norm}(T)) \subseteq \mathrm{norm}(\mathrm{st}_N(T)).$$

It is clear that backward induction implies subtree perfectness for normal form decisions. It turns out that the opposite implication does not hold: Insensitivity To Omission is not necessary for subtree perfectness for normal form decisions. We can prove that subtree perfectness for normal form decisions of $\mathrm{norm}_{\mathrm{opt}}$ is equivalent to Backward Conditioning, Preservation Under Addition, and Backward Mixture.

Theorem 3.23. *For any choice function* opt, $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for normal form decisions if and only if* opt *satisfies Backward Conditioning, Preservation Under Addition, and Backward Mixture.*

The proof will again rely on structural induction. The following lemma helps with the inductive step by requiring us only to check subtree perfectness for normal form decisions at the immediate successors of the root of a tree.

Lemma 3.24. *Let* norm *be any normal form operator. Let T be a consistent decision tree. If,*

(i) for all nodes $K \in \mathrm{ch}(T)$ such that K is in at least one element of $\mathrm{norm}(T)$,

$$\mathrm{st}_K(\mathrm{norm}(T)) \subseteq \mathrm{norm}(\mathrm{st}_K(T)),$$

(ii) and, for all nodes $K \in \mathrm{ch}(T)$, and all nodes $L \in \mathrm{st}_K(T)$ such that L is in at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$,

$$\mathrm{st}_L(\mathrm{norm}(\mathrm{st}_K(T))) \subseteq \mathrm{norm}(\mathrm{st}_L(\mathrm{st}_K(T))),$$

then, for all nodes N in T such that N is in at least one element of $\mathrm{norm}(T)$,

$$\mathrm{st}_N(\mathrm{norm}(T)) \subseteq \mathrm{norm}(\mathrm{st}_N(T)).$$

Proof. If N is the root of T, then the result is immediate. If $N \in \mathrm{ch}(T)$, then the result follows from (i). Otherwise, N must be in $\mathrm{st}_K(T)$ for one $K \in \mathrm{ch}(T)$.

By assumption, there is a $U \in \mathrm{norm}(T)$ that contains N (and of course also K). Therefore, $U \in \mathrm{st}_K(\mathrm{norm}(T))$, and by (i), $\mathrm{st}_K(U) \in \mathrm{norm}(\mathrm{st}_K(T))$, and so N is also in at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$.

We use the fact that, if $\mathcal{U}$ and $\mathcal{V}$ are sets of normal form decisions such that $\mathcal{U} \subseteq \mathcal{V}$, then for any node N, $\mathrm{st}_N(\mathcal{U}) \subseteq \mathrm{st}_N(\mathcal{V})$. Combining everything, by (i),

$$\mathrm{st}_N(\mathrm{st}_K(\mathrm{norm}(T))) \subseteq \mathrm{st}_N(\mathrm{norm}(\mathrm{st}_K(T)))$$

hence, since N is in at least one element of $\mathrm{norm}(\mathrm{st}_K(T))$, by (ii) we have

$$\subseteq \mathrm{norm}(\mathrm{st}_N(\mathrm{st}_K(T))),$$

whence the desired result follows, since $\mathrm{st}_N(\mathrm{st}_K(T)) = \mathrm{st}_N(T)$. □

Unsurprisingly, the proofs of necessity are very similar to those for the backward induction theorem. They can be found in Appendix B.4.

Lemma 3.25. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for normal form decisions, then* opt *satisfies Backward Conditioning.*

Lemma 3.26. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for normal form decisions, then* opt *satisfies Preservation Under Addition.*

Lemma 3.27. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for normal form decisions, then* opt *satisfies Backward Mixture.*

Proof of Theorem 3.23. "only if". Follows from Lemmas 3.25, 3.26, and 3.27.

"if". We proceed as usual by structural induction. The base step is trivial as usual. Let $\mathrm{ch}(T) = \{K_1, \dots, K_n\}$ and let $T_i = \mathrm{st}_{K_i}(T)$. The induction hypothesis says that $\mathrm{norm}_{\mathrm{opt}}$ is subtree perfect for normal form decisions on all T_i. More precisely, for each T_i, and for every L that is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T_i)$,

$$\mathrm{st}_L(\mathrm{norm}_{\mathrm{opt}}(T_i) \subseteq \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_L(T)).$$

We must show that, for any N in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$,

$$\mathrm{st}_N(\mathrm{norm}(T)) \subseteq \mathrm{norm}(\mathrm{st}_N(T)).$$

By the induction hypothesis and Lemma 3.24, it is enough to show this only for $N \in \mathrm{ch}(T)$, that is, to show that

$$\mathrm{st}_{K_i}(\mathrm{norm}_{\mathrm{opt}}(T)) \subseteq \mathrm{norm}_{\mathrm{opt}}(T_i) \tag{3.9}$$

for each i such that K_i is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$.

Suppose the root of T is a decision node, so $T = \bigsqcup_{i=1}^n T_i$. Let U be an element of $\mathrm{norm}_{\mathrm{opt}}(T)$. There is a j such that K_j is in U; let U_j denote $\mathrm{st}_{K_j}(U)$. To establish Eq. (3.9) we must show that $U_j \in \mathrm{norm}_{\mathrm{opt}}(T_j)$.

Note that $\mathrm{gamb}(U_j) = \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T))$ since $U \in \mathrm{norm}_{\mathrm{opt}}(T)$. Obviously, also $\mathrm{gamb}(U_j) \in \mathrm{gamb}(T_j)$ by definition of gamb. Hence, it must hold that

$$\mathrm{gamb}(U_j) \subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)) \cap \mathrm{gamb}(T_j),$$

but, also, because $\mathrm{gamb}(T_j) \subseteq \mathrm{gamb}(T)$, and once noted that $\mathrm{ev}(T) = \mathrm{ev}(T_j)$, it follows from Preservation Under Addition that

$$\subseteq \mathrm{opt}(\mathrm{gamb}(T_j)|\mathrm{ev}(T_j))$$

Putting everything together, we confirm that $U_j \in \mathrm{norm}_{\mathrm{opt}}(T_j)$. This proves the induction step for decision nodes.

Now suppose that the root of T is a chance node, so $T = \bigodot_{i=1}^n E_i T_i$. Again, let $U = \bigodot_{i=1}^n E_i U_i \in \mathrm{norm}_{\mathrm{opt}}(T)$. To establish Eq. (3.9) we must show that $U_i \in \mathrm{norm}_{\mathrm{opt}}(T_i)$ for all i.

Indeed, since $U \in \operatorname{norm}_{\mathrm{opt}}(T)$,

$$\operatorname{gamb}(U) \in \operatorname{opt}(\operatorname{gamb}(T)) = \operatorname{opt}\left(\bigoplus E_i \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right)$$

so by Lemma 3.8,

$$\subseteq \bigoplus E_i \operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T) \cap E_i).$$

So, for each T_i, there is a normal form decision $V_i \in \operatorname{norm}_{\mathrm{opt}}(T_i)$ such that

$$E_i \operatorname{gamb}(V_i) = E_i \operatorname{gamb}(U_i).$$

Can we apply Backward Conditioning?

Obviously, $\{\operatorname{gamb}(V_i), \operatorname{gamb}(U_i)\} \subseteq \operatorname{gamb}(T_i)$, and

$$\begin{aligned} E_i \operatorname{gamb}(V_i) \oplus \overline{E_i} Z = \operatorname{gamb}(U) &\subseteq \operatorname{opt}\left(\bigoplus E_i \operatorname{gamb}(T_i)\middle|\operatorname{ev}(T)\right) \\ &= \operatorname{opt}(E_i \operatorname{gamb}(T_i) \oplus \overline{E_i}\mathcal{Z}|\operatorname{ev}(T)) \end{aligned}$$

for suitable choices for $\mathcal{Z}$ and a $Z \in \mathcal{Z}$. Therefore we can apply Backward Conditioning and Lemma 3.6 to conclude that U_i is in $\operatorname{norm}_{\mathrm{opt}}(T_i)$ for each i. This proves the induction step for chance nodes. □

3.1.5 Computation of $\operatorname{back}_{\mathrm{opt}}$ Using Other Choice Functions

If the conditions of Theorem 3.12 are met, we can use $\operatorname{back}_{\mathrm{opt}}$ to find $\operatorname{norm}_{\mathrm{opt}}(T)$ without having to compare every normal form decision, potentially saving time for large trees. If opt is very computationally expensive, however, repeated applications, even on smaller sets, may be too difficult. Instead of performing backward induction using opt, one might wish to use a more conservative choice function if it is more tractable, and apply opt only at the end of the process. The following theorem establishes the validity of this method. In fact, it is a corollary of our more general and powerful Theorem 3.32 presented later in this section, which is the theorem one would want to apply in practice, but to understand the process it is more useful to state and prove the weaker theorem first.

Theorem 3.28. *Let* opt_1 *and* opt_2 *be choice functions such that* opt_1 *satisfies Backward Conditioning, Insensitivity To Omission, Preservation Under Addition, and Backward Mixture, and for any non-empty event* A *and any non-empty finite set of* A*-consistent gambles* $\mathcal{X}$*,*

$$\mathrm{opt}_1(\mathcal{X}|A) \subseteq \mathrm{opt}_2(\mathcal{X}|A).$$

Then, for any consistent decision tree T*,*

$$\mathrm{norm}_{\mathrm{opt}_1}(T) = \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T)). \tag{3.10}$$

Lemma 3.29. *Let* A *be any non-empty event, and let* opt_1 *and* opt_2 *be choice functions such that* opt_1 *satisfies Insensitivity To Omission, and such that for any non-empty finite set of* A*-consistent gambles* $\mathcal{X}$*,*

$$\mathrm{opt}_1(\mathcal{X}|A) \subseteq \mathrm{opt}_2(\mathcal{X}|A). \tag{3.11}$$

Then, for any set $\mathcal{T}$ *of consistent decision trees, where* $\mathrm{ev}(T) = A$ *for all* $T \in \mathcal{T}$*,* $\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})) = \mathrm{norm}_{\mathrm{opt}_1}(\mathcal{T})$*.*

Proof. For brevity of notation, we assume $A = \Omega$ and $\mathrm{ev}(T) = \Omega$ for all $T \in \mathcal{T}$; the argument for a more general A is identical.

We have, by the definition of $\mathrm{norm}_{\mathrm{opt}_1}$,

$$\begin{aligned}
&\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})) \\
&\qquad = \{U \in \mathrm{nfd}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T}))\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})))\} \\
&\qquad = \{U \in \mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})))\}
\end{aligned}$$

and now, by Eq. (1.10) for $\mathrm{norm}_{\mathrm{opt}_2}$,

$$= \{U \in \mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{opt}_2(\mathrm{gamb}(\mathcal{T})))\}$$

and now, by Eq. (3.11) and Insensitivity To Omission of opt_1 (using $\mathcal{X} = \mathrm{gamb}(\mathcal{T})$ and $\mathcal{Y} = \mathrm{opt}_2(\mathcal{X})$),

$$= \{U \in \mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{gamb}(\mathcal{T}))\}$$

and now, by definition of $\mathrm{norm}_{\mathrm{opt}_2}$,

$$
\begin{aligned}
= \{U \in \mathrm{nfd}(\mathcal{T})\colon\ & \mathrm{gamb}(U) \subseteq \mathrm{opt}_2(\mathrm{gamb}(\mathcal{T})) \text{ and} \\
& \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{gamb}(\mathcal{T}))\}
\end{aligned}
$$

and finally, again by Eq. (3.11),

$$
\begin{aligned}
&= \{U \in \mathrm{nfd}(\mathcal{T})\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}_1(\mathrm{gamb}(\mathcal{T}))\} \\
&= \mathrm{norm}_{\mathrm{opt}_1}(\mathcal{T}).
\end{aligned}
$$

□

Lemma 3.30. *If $\mathcal{T} \subseteq \mathcal{U} \subseteq \mathcal{V}$ are sets of consistent decision trees, with* $\mathrm{ev}(T) = A$ *for all* $T \in \mathcal{V}$, opt *satisfies Insensitivity To Omission, and* $\mathrm{norm}_{\mathrm{opt}}(\mathcal{T}) = \mathrm{norm}_{\mathrm{opt}}(\mathcal{V})$, *then* $\mathrm{norm}_{\mathrm{opt}}(\mathcal{U}) = \mathrm{norm}_{\mathrm{opt}}(\mathcal{V})$.

Proof. By assumption, we have that

$$\mathrm{opt}(\mathrm{gamb}(\mathcal{V})|A) = \mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A) \subseteq \mathrm{gamb}(\mathcal{T}) \subseteq \mathrm{gamb}(\mathcal{U}) \subseteq \mathrm{gamb}(\mathcal{V}),$$

Hence, by Insensitivity To Omission,

$$\mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A) = \mathrm{opt}(\mathrm{gamb}(\mathcal{U})|A) = \mathrm{opt}(\mathrm{gamb}(\mathcal{V})|A)$$

So,

$$
\begin{aligned}
\mathrm{norm}_{\mathrm{opt}}(\mathcal{U}) &= \{U \in \mathcal{U}\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A)\} \\
&\supseteq \{U \in \mathcal{T}\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(\mathcal{T})|A)\} = \mathrm{norm}_{\mathrm{opt}}(\mathcal{T})
\end{aligned}
$$

because $\mathcal{U} \supseteq \mathcal{T}$, and

$$
\begin{aligned}
\mathrm{norm}_{\mathrm{opt}}(\mathcal{U}) &= \{U \in \mathcal{U}\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(\mathcal{V})|A)\} \\
&\subseteq \{U \in \mathcal{V}\colon\ \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(\mathcal{V})|A)\} = \mathrm{norm}_{\mathrm{opt}}(\mathcal{V})
\end{aligned}
$$

because $\mathcal{U} \subseteq \mathcal{V}$. We conclude that

$$\mathrm{norm}_{\mathrm{opt}}(\mathcal{T}) \subseteq \mathrm{norm}_{\mathrm{opt}}(\mathcal{U}) \subseteq \mathrm{norm}_{\mathrm{opt}}(\mathcal{V}).$$

Now use $\mathrm{norm}_{\mathrm{opt}}(\mathcal{T}) = \mathrm{norm}_{\mathrm{opt}}(\mathcal{V})$. □

Proof of Theorem 3.28. We prove this by structural induction. Let T be any consistent decision tree. The base step, that the result holds for decision trees comprising only one node, is trivial. The induction hypothesis is that, when $T = \bigsqcup_{i=1}^{n} T_i$ or $T = \bigodot_{i=1}^{n} E_i T_i$, we have

$$\mathrm{norm}_{\mathrm{opt}_1}(T_i) = \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T_i)).$$

Suppose $T = \bigodot_{i=1}^{n} E_i T_i$. We know by Theorem 3.12 and the definition of $\mathrm{back}_{\mathrm{opt}_1}$ that

$$\begin{aligned}\mathrm{norm}_{\mathrm{opt}_1}(T) &= \mathrm{back}_{\mathrm{opt}_1}(T) \\ &= \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}_1}(T_i)\right) \\ &= \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}_1}(T_i)\right),\end{aligned}$$

whence, using the induction hypothesis,

$$= \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T_i))\right). \tag{3.12}$$

On the other hand, by definition of $\mathrm{back}_{\mathrm{opt}_2}$,

$$\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T)) = \mathrm{norm}_{\mathrm{opt}_1}\left(\mathrm{norm}_{\mathrm{opt}_2}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}_2}(T_i)\right)\right).$$

and from $\mathrm{opt}_1(\mathcal{X}|A) \subseteq \mathrm{opt}_2(\mathcal{X}|A)$, and Insensitivity To Omission, it follows quickly from the definition of $\mathrm{norm}_{\mathrm{opt}}$ that $\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{norm}_{\mathrm{opt}_2}(\mathcal{T})) = \mathrm{norm}_{\mathrm{opt}_1}(\mathcal{T})$, and so we have

$$= \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}_2}(T_i)\right). \tag{3.13}$$

It is fairly easy to see that

$$\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T_i)) \subseteq \bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}_2}(T_i) \subseteq \bigodot_{i=1}^{n} E_i \,\mathrm{nfd}(T_i).$$

But, by Eq. (3.12), together with $\mathrm{nfd}(T) = \bigodot_{i=1}^{n} E_i \,\mathrm{nfd}(T_i)$ and $\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{nfd}(T)) = \mathrm{norm}_{\mathrm{opt}_1}(T)$, we can apply Lemma 3.30, and hence

$$\mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{back}_{\mathrm{opt}_2}(T_i)\right) = \mathrm{norm}_{\mathrm{opt}_1}(T)$$

Now using Eq. (3.13), we find indeed that

$$\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T)) = \mathrm{norm}_{\mathrm{opt}_1}(T).$$

Suppose finally that $T = \bigsqcup_{i=1}^{n} T_i$. The proof proceeds in the same way as for chance nodes: by Theorem 3.12,

$$\mathrm{norm}_{\mathrm{opt}_1}(T) = \mathrm{norm}_{\mathrm{opt}_1}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T_i))\right),$$

and by Insensitivity To Omission and $\mathrm{opt}_1(\mathcal{X}|A) \subseteq \mathrm{opt}_2(\mathcal{X}|A)$,

$$\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T)) = \mathrm{norm}_{\mathrm{opt}_1}\left(\bigsqcup_{i=1}^{n} \mathrm{back}_{\mathrm{opt}_2}(T_i)\right).$$

Again, it is easy to see that

$$\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T_i)) \subseteq \bigsqcup_{i=1}^{n} \mathrm{back}_{\mathrm{opt}_2}(T_i) \subseteq \bigsqcup_{i=1}^{n} \mathrm{nfd}(T_i),$$

whence we find, as before,

$$\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{back}_{\mathrm{opt}_2}(T)) = \mathrm{norm}_{\mathrm{opt}_1}(T).$$

□

In fact, we can generalize this theorem to an even more useful one. At some steps of the backward induction we may wish to use opt_2. At other steps we may wish to use opt_1. Or we may wish to use opt_2 followed by opt_1 (although Insensitivity To Omission tells us this is the same result as using opt_1 only, so we ignore this possibility). Or perhaps another $\mathrm{opt}_3 \supseteq \mathrm{opt}_1$ (for instance the identity choice function: don't try to eliminate anything at this stage). One might want such a complicated scheme because one has a good idea of what stages nothing is worth applying, what stages opt_2 will do well, and what stages opt_1 is easy enough to apply.

We can define a normal form operator to deal with such a scheme. Suppose that $\mathrm{opt}_1, \dots, \mathrm{opt}_m$ are a set of choice functions such that opt_1 satisfies Backward Conditioning, Insensitivity To Omission, Preservation Under Addition, and Backward Mixture, and of every i, $\mathrm{opt}_1 \subseteq \mathrm{opt}_i$. Let mix be a normal form operator defined as follows.

Definition 3.31. *For each decision tree T, identify* $\mathrm{opt}_T \in \{\mathrm{opt}_1, \dots, \mathrm{opt}_m\}$*, the choice function to use for that tree.*

- *If a tree T consists of only a leaf with reward $r \in \mathcal{R}$, then*

$$\mathrm{mix}(T) = \{T\}.$$

- *If a tree T has a chance node as root, that is, $T = \bigodot_{i=1}^{n} E_i T_i$,*

$$\mathrm{mix}\left(\bigodot_{i=1}^{n} E_i T_i\right) = \mathrm{norm}_{\mathrm{opt}_T}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{mix}\,(T_i)\right)$$

- *If a tree T has a decision node as root, that is, if $T = \bigsqcup_{i=1}^{n} T_i$, then*

$$\mathrm{mix}\left(\bigsqcup_{i=1}^{n} T_i\right) = \mathrm{norm}_{\mathrm{opt}_T}\left(\bigsqcup_{i=1}^{n} \mathrm{mix}(T_i)\right).$$

Theorem 3.32. *For any consistent decision tree T,*

$$\mathrm{norm}_{\mathrm{opt}_1}(T) = \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{mix}(T)).$$

Proof. We proceed by structural induction. The base step is, as usual, trivial. The induction hypothesis is that, for $T = \bigodot_{i=1}^{n} E_i T_i$ or $T = \bigsqcup_{i=1}^{n} T_i$, we have for all i that

$$\mathrm{norm}_{\mathrm{opt}_1}(T_i) = \mathrm{norm}_{\mathrm{opt}_1}(\mathrm{mix}(T_i)).$$

Suppose that T has root at a chance node, so $T = \bigodot_{i=1}^{n} E_i T_i$. By insensitivity to the omission of non-optimal elements, Lemma 3.29, and $\mathrm{opt}_1 \subseteq \mathrm{opt}_k$ for all k, we have

$$\begin{aligned}\mathrm{norm}_{\mathrm{opt}_1}(\mathrm{mix}(T)) &= \mathrm{norm}_{\mathrm{opt}_1}\left(\mathrm{norm}_T\left(\bigodot_{i=1}^{n} E_i \,\mathrm{mix}(T_i)\right)\right)\\ &= \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{mix}(T_i)\right),\end{aligned}$$

since $\mathrm{norm}_{\mathrm{opt}_1} \circ \mathrm{norm}_{\mathrm{opt}_k} = \mathrm{norm}_{\mathrm{opt}_1}$ for any opt_k, in particular for opt_T . Now, by Theorem 3.12,

$$\mathrm{norm}_{\mathrm{opt}_1}(T) = \mathrm{norm}_{\mathrm{opt}_1}\left(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}_1}(T_i)\right)$$

and by the induction hypothesis

$$= \operatorname{norm}_{\operatorname{opt}_1}\left(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\operatorname{opt}_1}(\operatorname{mix}(T_i))\right).$$

Now, clearly $\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\operatorname{opt}_1}(\operatorname{mix}(T_i)) \subseteq \bigodot_{i=1}^{n} E_i \operatorname{mix}(T_i) \subseteq \operatorname{nfd}(T)$, and we have shown that applying $\operatorname{norm}_{\operatorname{opt}_1}$ to the first set is the same as applying $\operatorname{norm}_{\operatorname{opt}_1}$ to the third set, so we can apply Lemma 3.30 to find

$$\operatorname{norm}_{\operatorname{opt}_1}(\operatorname{mix}(T)) = \operatorname{norm}_{\operatorname{opt}_1}\left(\bigodot_{i=1}^{n} E_i \operatorname{mix}(T_i)\right) = \operatorname{norm}_{\operatorname{opt}_1}(T)$$

as required. If T has its root at a decision node there is absolutely no difference in the proof. □

3.2 Extensive Form Backward Induction

Backward induction seems to represent the intuition that, in sequential decision making, the subject can delete options that are locally non-optimal because, upon reaching that decision point, she would never choose them. This reasoning is of a decidedly extensive form nature, and so it makes sense to investigate extensive form solutions obtained by backward induction. For general choice functions there are several plausible ways of doing this. These can appear quite complicated but we now have all the tools and ideas necessary to investigate them and their relationship to $\operatorname{norm}_{\operatorname{opt}}$.

If opt violates the conditions of Theorem 3.12, then any attempt to use backward induction is likely to lead to problems for some decision trees. It is still possible to use any of this section's extensive form solutions in such cases, but the results may not be appealing. When comparing these backward induction methods, it will cause confusion if problems appear because opt is badly behaved rather than because the method is badly behaved, so for this section we assume that opt satisfies Backward Mixture, Insensitivity To Omission, Preservation Under Addition, and Backward Conditioning.

The first possibility is very simple: use the extensive form equivalent to $\operatorname{back}_{\operatorname{opt}}$ (and, by assumption, $\operatorname{norm}_{\operatorname{opt}}$). Since any decision arc in this solution is part of an optimal normal form decision, the subject will not choose any obviously foolish arcs. However, it lacks subtree perfectness. It is also not an extensive form solution in spirit, being based entirely on normal form logic. Finally, its normal form equivalent may not be the original normal form solution (recall Fig. 2.5).

We can refine this approach to enforce subtree perfectness and a true extensive form interpretation. At each node N in T, delete all decision arcs that do not appear in at least one element of $\mathrm{back}_{\mathrm{opt}}(\mathrm{st}_N(T))$. In other words, at a particular node, remove any arc for which, out of all remaining strategies for this subtree, none of the optimal ones at this stage contain said arc. When $\mathrm{back}_{\mathrm{opt}} = \mathrm{norm}_{\mathrm{opt}}$, the interpretation is even clearer: remove any arc that has no optimal strategies associated with it in this subtree. Call this extensive form operator $\mathrm{ext}_{\mathrm{back}_{\mathrm{opt}}}$ (this inelegant notation shall shortly be made obsolete by Proposition 3.34)

We can provide an alternative definition of this approach without explicit reference to $\mathrm{back}_{\mathrm{opt}}$.

Definition 3.33. *Define the extensive form operator* $\mathrm{ext}_{\mathrm{opt}}$*, for a particular choice function* opt*, recursively as follows. For reward nodes,* $\mathrm{ext}_{\mathrm{opt}}(T) = T$*. At chance nodes,*

$$\mathrm{ext}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i T_i\right) = \bigodot_{i=1}^{n} E_i \,\mathrm{ext}_{\mathrm{opt}}(T_1),$$

so $\mathrm{ext}_{\mathrm{opt}}$ *eliminates nothing that was not eliminated in subtrees. At decision nodes, where* $T = \bigsqcup_{i=1}^{n} T_i$*, let* d_i *represent the arc from the root to* T_i*. Find all normal form decisions* U *such that*

$$\begin{aligned} U &\in \mathrm{nfd}\left(\bigsqcup_{i=1}^{n} \mathrm{ext}_{\mathrm{opt}}(T_i)\right), \qquad \textit{and} \\ \mathrm{gamb}(U) &\subseteq \mathrm{opt}\left(\mathrm{gamb}\left(\bigsqcup_{i=1}^{n} \mathrm{ext}_{\mathrm{opt}}(T_i)\right)\middle|\,\mathrm{ev}(T)\right). \end{aligned} \tag{3.14}$$

Let $\mathcal{I}$ *be the set of* i *such that* d_i *is in at least one such* U*. Then*

$$\mathrm{ext}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} T_i\right) = \bigsqcup_{i\in\mathcal{I}} \mathrm{ext}_{\mathrm{opt}}(T_i).$$

Proposition 3.34. *For any choice function* opt*,* $\mathrm{ext}_{\mathrm{back}_{\mathrm{opt}}} = \mathrm{ext}_{\mathrm{opt}}$*.*

Proof. The structural induction for this proof is very simple. The base step, that the two operators coincide at reward nodes, is trivial. The induction hypothesis is that the two operators coincide for all immediate successors of the root of T.

When T is a chance node, then both operators simply mix the extensive form solutions of their immediate successors at a chance node, and so the induction hypothesis provides the result immediately.

When $T = \bigsqcup_{i=1}^{n} T_i$, then the set of normal form decisions satisfying Eq. (3.14) becomes, by the induction hypothesis, the set of U such that

$$U \in \mathrm{nfd}\left(\bigsqcup_{i=1}^{n} \mathrm{ext}_{\mathrm{back}_{\mathrm{opt}}}(T_i)\right), \qquad \text{and}$$
$$\mathrm{gamb}(U) \subseteq \mathrm{opt}\left(\mathrm{gamb}\left(\bigsqcup_{i=1}^{n} \mathrm{ext}_{\mathrm{back}_{\mathrm{opt}}}(T_i)\right)\middle| \mathrm{ev}(T)\right).$$

Observe that this is equivalent to the set of U such that

$$U \in \mathrm{nfd}\left(\bigsqcup_{i=1}^{n} \mathrm{back}_{\mathrm{opt}}(T_i)\right), \qquad \text{and}$$
$$\mathrm{gamb}(U) \subseteq \mathrm{opt}\left(\mathrm{gamb}\left(\bigsqcup_{i=1}^{n} \mathrm{back}_{\mathrm{opt}}(T_i)\right)\middle| \mathrm{ev}(T)\right),$$

that is, this set is exactly $\mathrm{back}_{\mathrm{opt}}(T)$. So, $\mathrm{ext}_{\mathrm{opt}}$ deletes any decision arc from the root of T that is not in any strategy in $\mathrm{back}_{\mathrm{opt}}(T)$ and so both $\mathrm{ext}_{\mathrm{opt}}$ and $\mathrm{ext}_{\mathrm{back}_{\mathrm{opt}}}$ delete the same decision arcs at the root of T. □

It is clear from the definition that $\mathrm{ext}_{\mathrm{opt}}$ is subtree perfect. If $\mathrm{back}_{\mathrm{opt}} = \mathrm{norm}_{\mathrm{opt}}$, then also the subject will never follow an arc that is only included in non-optimal normal form decisions *in the particular subtree in question.* This does not prevent the subject from following an arc that is not part of an optimal normal form decision in the full tree (clearly, else this method would be identical to the first proposal). This can be seen most simply in Fig. 3.1 and pointwise dominance. If X and Y are gambles such that neither pointwise dominates the other, then there is a c such that Y does not dominate $X - c$ and X does not dominate $Y - c$. So, applying $\mathrm{ext}_{\mathrm{opt}}$ to the tree in the figure will simply return the whole tree. But $X - c$ and $Y - c$ are not part of any optimal strategy at the root node.

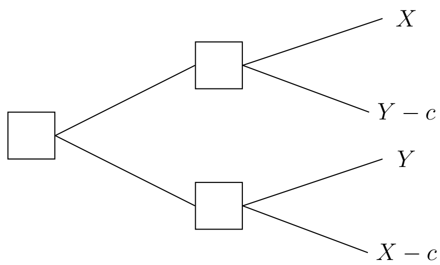

Figure 3.1: A tree for which $\mathrm{ext}_{\mathrm{opt}}$ can choose globally non-optimal arcs.

We should not be surprised by this: if $\mathrm{norm}_{\mathrm{opt}}$ is not subtree perfect then we cannot expect to find a subtree perfect extensive form solution that mimics its behaviour. Should $\mathrm{norm}_{\mathrm{opt}}$ be subtree perfect, then $\mathrm{ext}_{\mathrm{opt}}$ and $\mathrm{norm}_{\mathrm{opt}}$ will be equivalent.

Theorem 3.35. $\mathrm{norm}_{\mathrm{opt}}$ *and* $\mathrm{ext}_{\mathrm{opt}}$ *are equivalent if and only if* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect.*

Proof. “only if”. Observe that $\mathrm{ext}_{\mathrm{opt}}$ is subtree perfect. By Lemma 2.19, $\mathrm{norm}_{\mathrm{opt}}$ is subtree perfect.

“if”. Use the definition of $\mathrm{ext}_{\mathrm{opt}}$ that invoked $\mathrm{back}_{\mathrm{opt}}$. By subtree perfectness and backward induction, for any consistent decision tree T, $\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{back}_{\mathrm{opt}}(\mathrm{st}_N(T))$ for all N that are in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$ (equivalently, all N in $\mathrm{ext}_{\mathrm{opt}}(T)$). Therefore by definition, $\mathrm{ext}_{\mathrm{opt}}$ is the extensive form correspondent of $\mathrm{norm}_{\mathrm{opt}}$. By Theorem 2.21, $\mathrm{norm}_{\mathrm{opt}}$ has an extensive form equivalent. This must be $\mathrm{ext}_{\mathrm{opt}}$. □

Using $\mathrm{ext}_{\mathrm{opt}}$ rather than the extensive form correspondent to $\mathrm{back}_{\mathrm{opt}}$ is appealing because of subtree perfectness and a truer extensive form nature, but there is still the remains of normal form reasoning: arcs are deleted based on optimal gambles, and these gambles are obtained via normal form decisions. It may seem as though any extensive form operator based on a choice function must have this feature. Choice functions act on gambles, and how else can gambles appear if not through normal form decisions? Seidenfeld [63] provides an intriguing alternative.

His method relies on having a choice function capable of encoding the concept of *complete ignorance* about which event in a particular partition will obtain. In the particular application Seidenfeld is considering, the subject represents her beliefs as a set of probability mass functions, and considers any gamble optimal if it maximizes expected utility for at least one element of this set (this is called E-admissibility; see Section 6.1 for more details). In such a setting, complete ignorance about a partition is easy to represent. For other choice functions, representation may be difficult or impossible. For instance, if one wishes to use a single probability mass function, then the uniform distribution over the partition is the only sensible possibility, but this can easily be criticised too [79, §5.5.1].

So, in what follows, assume that the subject's belief model can express complete ignorance. Seidenfeld's algorithm then runs as follows. At ultimate decision nodes, opt can be applied to the gambles. Delete any arc corresponding to a non-optimal gamble. Then, to move backwards, transform the decision node into a chance node, expressing *complete ignorance* about what will be chosen. The interpretation is that the subject's choice in the future is an event, about which she knows nothing at present. Thus, the final layer of decision nodes turns into a layer of chance nodes, and the penultimate decision nodes become ultimate. The process is repeated until all decision nodes have been dealt with.

As an example (which will also serve as a demonstration of some inconvenient behaviour of this method), consider Fig. 3.2, again using pointwise dominance and where neither X nor Y dominate the other. Applying the choice function to the upper decision node eliminates the $X - c$ branch. Nothing is deleted from the lower branch. Replacement by chance nodes leads to Fig. 3.3.

Now the process moves to the root node. The possibility space has become larger, with each element ω splitting into four (E_1 and E_2 obtain, E_1 and $\overline{E}_2$ obtain, $\overline{E}_1$ and E_2 obtain, $\overline{E}_1$ and $\overline{E}_2$ obtain). Is either decision arc non-optimal? Clearly the upper arc must be optimal. For the lower arc, consider an ω in the initial Ω where $X(\omega) - c > Y(\omega)$. In the new possibility space, there is an outcome ω^* where $E_1 X(\omega^*) \oplus \overline{E}_1 Y(\omega^*) = Y(\omega)$ and $E_2(X(\omega^*) - c) \oplus \overline{E}_2(Y(\omega^*) - c) = X(\omega) - c > Y(\omega)$. So, the lower arc is not pointwise dominated by the upper, and the extensive form solution is given in Fig. 3.4.

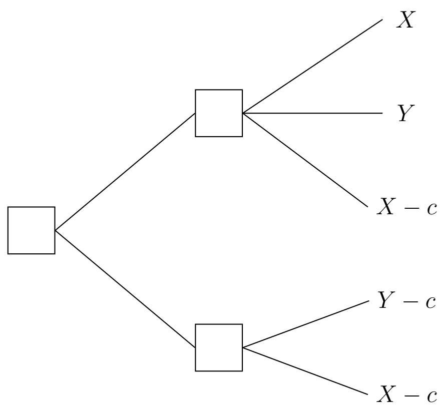

Figure 3.2: An example for Seidenfeld's backward induction method.

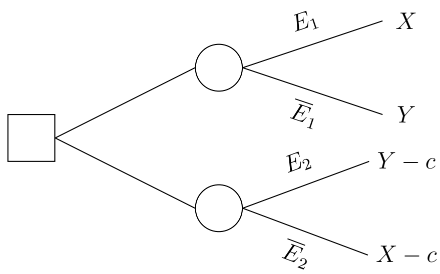

Figure 3.3: Seidenfeld's method: stage two.

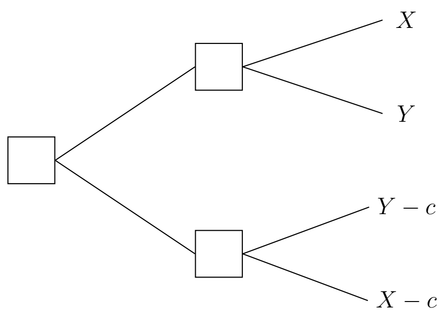

Figure 3.4: Seidenfeld's method: solution.

Compare this with $\mathrm{ext}_{\mathrm{opt}}$. At the first stage, $\mathrm{ext}_{\mathrm{opt}}$ eliminates the $X - c$ arc in the upper branch identically (the first stage of $\mathrm{ext}_{\mathrm{opt}}$ is always the first stage of Seidenfeld's method). At the root, $\mathrm{ext}_{\mathrm{opt}}$ notes that $Y - c$ is dominated by Y and $X - c$ is dominated by X, and so removes the lower arc. This seems intuitively reasonable: suppose one is offered either to choose between two options or to pay to choose between the same options. Would anyone choose to pay? On the other hand, Seidenfeld's method can be justified by arguing that, since one does not know what one will do in either case, one cannot exclude the possibility that one would choose X in the upper branch, $Y - c$ in the lower branch, and end up doing better out of $Y - c$. We find this reasoning difficult to dismiss, but are uncomfortable with the solutions it induces.

The problem for this particular example can be eliminated by forcing $E_1 = E_2$, so that the subject assumes she will make the same choice in the future whether she is confronted with $\{X, Y\}$ or $\{X - c, Y - c\}$. Unfortunately, generalizing this to more complicated trees does not seem to lead anywhere useful. Which of the two solutions to use seems to depend on whether or not one finds the solution in Fig. 3.4 sensible.

Finally, we consider a special extensive form backward induction method that is only available in certain circumstances. Recall that the traditional approach for backward induction involved replacing decision nodes with more convenient objects, namely expected utilities. With $\mathrm{back}_{\mathrm{opt}}$, we looked at a generalization that works with any choice function, effectively replacing decision nodes with sets of gambles. A simpler approach would be to replace a decision node with a single gamble: one of the optimal gambles in this subtree. Of course, for most choice functions this procedure depends on the arbitrary choice of a single gamble to retain. We investigate for what choice functions the procedure yields a well-defined extensive form operator.

Definition 3.36. *Let $\mathrm{ext}^*_{\mathrm{opt}}$ be an extensive form operator defined as follows. If the root of T is an ultimate decision node, $\mathrm{ext}^*_{\mathrm{opt}}(T) = \mathrm{ext}_{\mathrm{opt}}(T)$. If $T = \bigodot_{i=1}^n E_i T_i$, then*

$$\mathrm{ext}^*_{\mathrm{opt}}(T) = \bigodot_{i=1}^{n} E_i \, \mathrm{ext}^*_{\mathrm{opt}}(T_i).$$

If $T = \bigsqcup_{i=1}^n T_i$ and its root is a non-ultimate decision node, then for each i choose *a single gamble* $X_i \in \mathrm{norm}_{\mathrm{opt}}(\mathrm{ext}^*_{\mathrm{opt}}(T_i))$*, and find* $\mathrm{opt}(\{X_1, \dots, X_n\}|\mathrm{ev}(T))$*. Delete arc* d_i *if its gamble does not appear in this set.*

Lemma 3.37. *Suppose* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect,* $T = \bigsqcup_{i=1}^n T_i$*, and* $\mathrm{ext}^*_{\mathrm{opt}}(T_i)$ *does not depend on any of the gamble choices for all* i*, with* $\mathrm{ext}^*_{\mathrm{opt}}(T_i) = \mathrm{ext}_{\mathrm{opt}}(T_i)$*. Then* $\mathrm{ext}^*_{\mathrm{opt}}(T)$ *does not depend on the choice of gambles, and* $\mathrm{ext}^*_{\mathrm{opt}}(T) = \mathrm{ext}_{\mathrm{opt}}(T)$*.*

Proof. Since $\mathrm{ext}^*_{\mathrm{opt}}(T_i) = \mathrm{ext}_{\mathrm{opt}}(T_i)$, and

$$\begin{aligned}
\mathrm{gamb}(\mathrm{ext}_{\mathrm{opt}}(T_i)) &= \mathrm{gamb}(\mathrm{nfd}(\mathrm{ext}_{\mathrm{opt}}(T))) \\
&= \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) \\
&= \mathrm{opt}(\mathrm{gamb}(T_i)|\mathrm{ev}(T)),
\end{aligned}$$

whichever gamble X_i we pick to use from $\mathrm{ext}^*_{\mathrm{opt}}(T_i)$ will be in the same position p_i in the total preorder (where we write $p_i > p_j$ if p_i is higher in the order than p_j). Then, $\mathrm{opt}(\{X_1, \dots, X_n\}|\mathrm{ev}(T)) = \{X_i \colon (\forall j)(p_i \geq p_j)\}$, so the set $\mathcal{D}$ of arcs deleted by $\mathrm{ext}^*_{\mathrm{opt}}$ is

$$\begin{aligned}
\mathcal{D} &= \{d_i \colon X_i \in \mathrm{opt}(\{X_1, \dots, X_n\}|\mathrm{ev}(T))\} \\
&= \{d_i \colon (\forall j)(p_i \geq p_j)\},
\end{aligned}$$

and this is independent of the choices of $X_1, \dots, X_n$. Therefore $\mathrm{ext}^*_{\mathrm{opt}}(T)$ is well-defined. Further, again by total preordering, applying opt to $\{X_1, \dots, X_n\}$ will delete the same *arcs* as applying opt to $\bigcup_{i=1}^n \mathrm{gamb}(\mathrm{ext}^*_{\mathrm{opt}}(T_1))$. So, $\mathrm{ext}^*_{\mathrm{opt}}(T) = \mathrm{ext}_{\mathrm{opt}}(T)$. □

Theorem 3.38. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect, then* $\mathrm{ext}^*_{\mathrm{opt}}$ *does not depend on any of the choices of gambles, and* $\mathrm{ext}^*_{\mathrm{opt}} \equiv \mathrm{ext}_{\mathrm{opt}}$*.*

Proof. We proceed as usual by structural induction. The base step is satisfied trivially. The induction hypothesis is that, for any consistent tree $T = \bigodot_{i=1}^n E_i T_i$ or $T = \bigsqcup_{i=1}^n T_i$, the $\mathrm{ext}^*_{\mathrm{opt}}(T_i)$ are well-defined and $\mathrm{ext}^*_{\mathrm{opt}}(T_i) = \mathrm{ext}_{\mathrm{opt}}(T_i)$.

If $T = \bigsqcup_{i=1}^n T_i$ then, by Lemma 3.37, $\mathrm{ext}^*_{\mathrm{opt}}(T)$ is well-defined and $\mathrm{ext}^*_{\mathrm{opt}}(T) =$

$\mathrm{ext}_{\mathrm{opt}}(T)$. If $T = \bigodot_{i=1}^{n} E_i T_i$, then by definition and the induction hypothesis,

$$\begin{aligned}
\mathrm{ext}^*_{\mathrm{opt}}(T) &= \bigodot_{i=1}^{n} E_i\, \mathrm{ext}^*_{\mathrm{opt}}(T_i) \\
&= \bigodot_{i=1}^{n} E_i\, \mathrm{ext}_{\mathrm{opt}}(T_i) \\
&= \mathrm{ext}_{\mathrm{opt}}(T).
\end{aligned}$$

□

3.3 Backward Induction or Subtree Perfectness?

In this section we consider various arguments for or against backward induction and subtree perfectness. When we say "against backward induction" we mean an argument that a choice function opt with $\mathrm{back}_{\mathrm{opt}} \neq \mathrm{norm}_{\mathrm{opt}}$ can be acceptable. Similarly, an argument against subtree perfectness is an argument in favour of using choice functions for which $\mathrm{norm}_{\mathrm{opt}}$ lacks subtree perfectness. Given the relationship between backward induction and subtree perfectness, there are three possible theses:

- Subtree perfectness is necessary;
- Backward induction is necessary but subtree perfectness is not;
- Backward induction is not necessary.

We review some common criticisms and defences of these positions. Often, the distinction between subtree perfectness and backward induction is rarely made clearly, so on occasion an argument apparently against one is in truth an argument against the other. Usually this confusion is caused by implicit assumption of Total Preordering, under which the two concepts become very similar. Indeed, we are not aware of any notable choice function satisfying Total Preordering that satisfies Backward Mixture but not Mixture.

Criticisms of failures of subtree perfectness and backward induction can be broadly split into three categories. The first is that the subject should not choose a strategy that involves a substrategy that is locally non-optimal. This is an appeal to the logic of backward induction. Authors may go on from here to argue that, following this logic, subtree perfectness must hold, but this only follows from other assumptions they have made. Without appeal to other principles, for instance Total Preordering, this is an argument that only backward induction is necessary.

A second argument is that the subject should be free to make a local decision in a subtree based only on the current subtree. This was our motivation for introducing subtree perfectness. Hammond [24] in particular strongly argues for this principle. Consider the decision tree in Fig. 3.5. Suppose that $\mathrm{opt}(\{X, Y, Z\}) = \{Z\}$, $\mathrm{opt}(\{Y, Z\})]\{Y\}$, and $\mathrm{opt}(\{X, Y\}) = \{X\}$. The motivation for this example is the *potential addict* in [23, § 3]: X corresponds to avoiding an addictive drug, Z to trying it once but never again, and Y to becoming an addict. Should local decisions, and hence subtree perfectness, be required, then the solution must either be X, Y, or both. Thus, $\mathrm{norm}_{\mathrm{opt}}$ is not an acceptable solution. If the solution chosen is Y, then the subject is called *myopic*: she travels to the second decision node because it has the preferred Z, knowing well that when she reaches it she will choose Y. This behaviour has been widely criticized.

The alternative strategy, X, is called *sophisticated*, defended for instance by Hazen [27]. The subject realizes that at the second decision node she will choose Y, so eliminates Z from her considerations. This solution corresponds to $\mathrm{back}_{\mathrm{opt}}$, and, even though backward induction does not "work" here in the sense we mean it, this solution is much more acceptable than myopic behaviour. Hammond [24] argues against this solution, however, invoking his second and more celebrated principle of consequentialism: a subject's decision at a particular node should depend only on all of the available gambles. Calling for consequentialism not only rejects $\mathrm{norm}_{\mathrm{opt}}$ as unacceptable (because of the violation of local decision making), but also rejects other sensible solutions based on opt (any solution based on opt that is not $\mathrm{norm}_{\mathrm{opt}}$ must violate consequentialism).

Finally, it may be argued that the properties required for subtree perfectness or backward induction are rational, for other reasons (in particular, static rather than sequential). For instance, the motivation for the simplest form of the Independence Axiom seems quite natural: if one will receive r_1 if a coin lands heads,

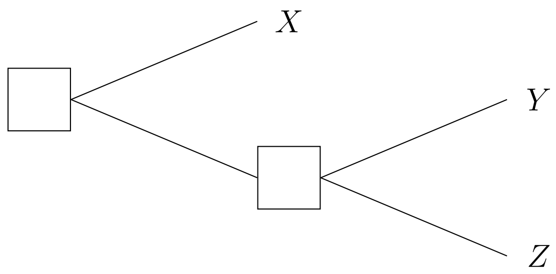

Figure 3.5: Consistency does not imply subtree perfectness.

and can choose to receive either r_2 or r_3 if the coin lands tails (choosing before the coin is tossed), one would probably decide whether r_2 or r_3 is preferable without reference to r_1. More complicated formulations, up to Mixture can be justified similarly. Similarly, Insensitivity To Omission and Preservation Under Addition are quite straightforward to justify (see for instance [46]), although one must be careful to ensure that adding or removing options from a set gives the subject no further information about the options.[2] Further, we believe that it is very difficult to imagine wanting to violate Conditioning.

This leaves only Total Preordering, a very popular property indeed, and yet difficult to justify independently. Indeed, Luce and Raiffa suggest a possible rationalization of the property and then immediately conclude it is "apparently not suitable" when combined with other, much more reasonable, properties [46, p. 289]. It is also instructive to observe that Hammond, a proponent of Total Preordering, immediately uses sequential arguments to criticize Seidenfeld's abandoning of Total Preordering [63].

Let us now consider how these criticisms can be answered by those who wish to fail backward induction or subtree perfectness. The third criticism can be answered in two ways. The first, following for instance Allais [1], Ellsberg [18], and

[2]Luce and Raiffa give the example of a diner choosing from a menu, and then finding extra options that he doesn't like, yet changing his order anyway. Their idea is that simply seeing that certain options are available can inform about the quality of restaurant, hence potentially switching a decision from a meal that is "safe" at any restaurant to a meal that would only be pleasant at a fine restaurant.

Kahnemann and Tversky [34], is to observe that one can experimentally verify that people violate forms of mixture properties, and apparently are sometimes content with doing so even when the violation is pointed out to them. One can argue from this that the principle is not compelling, since people do not want to follow it. We do not consider this argument particularly convincing: when investigating normative decision making, it is not sufficient for an individual to hold a particular opinion if it cannot be justified. Still, if one has a choice function violating the mixture properties but otherwise providing appealing results, it would be comforting to know its behaviour may not be abhorrent to many people. And certainly when trying to model the way *other people* make decisions, the importance of these results cannot be dismissed.

In a related argument, McClennen [51, §14.7] gives a robust defence of indeterminate choice functions. He observes that in the literature of normative decision theory, the call for determinism is louder than is perhaps justified, and postulates that a principal reason for this may be linked to descriptive theory. In a descriptive theory, he argues, a determinite answer is certainly preferable, and suggests that this preference may be transferred to normative theory without justification. This underlines the earlier point that arguments in favour of Total Preordering tend not to have as strong a normative justification as most of the other properties.

The other possible counter-argument, following Machina [50] among others, is that violations of mixture properties may be much more convenient for modelling decision making. The argument is given only in terms of mixture properties, but could presumably be extended to allow violations of the others. A small example is illuminating.

Suppose that one has a choice between gambles $pZ+(1-p)X$, and $pZ+(1-p)Y$, where p is a probability near one, independent of all of the gambles. By Mixture, the choice will depend only on the choice between X and Y. Suppose that X is preferred to Y. But suppose we are in a situation where the gambles Z and X are gambles about the "same sort of thing", but with Z much better than X, whereas Y is something of a quite different order. Machina's example has X to be watching a movie featuring a very attractive star, and Z to be a romantic week with them.

The argument then is that, having had a high probability to attain Z but failing, watching the movie X would involve merely several hours of miserable regret. Better, perhaps, to choose Y.

Is this argument convincing? As Machina points out, a defender of subtree perfectness can reason as follows. Gamble X is not simply the gamble "watch a movie with the beautiful star" but in fact "watch a movie with the beautiful star having failed to achieve an almost-certain week with said star". Therefore $\mathrm{opt}(\{X, Y\}) = \{X\}$ and Mixture is satisfied. In short, some of the rewards are not just the physical reward but also the regret of not having received the best reward. Machina argues that this is an inelegant solution, and that encoding the regret in a context-sensitive preference model is more artful. He questions whether there is any point in choosing the more complicated subtree perfect model. Our answer would be that the advantages of subtree perfectness and hence backward induction will be vital in large problems, and so it is well worth the effort. Further, is one supposed to, when solving a decision problem, remember the full tree in which one is embedded? Machina's response, to deal with this by "assuming (or hoping)" that the effects of distant decisions on the current decision should become negligible, is perhaps tenuous. One can certainly imagine that failing to win a romantic week with a beautiful movie star would be remembered for a long time.

Also, in our setting, the failure to acquire Z is an *event*, say A, and so we are not considering $\mathrm{opt}(\{X, Y\})$ but in fact $\mathrm{opt}(\{X, Y\}|A)$, and there is no reason why we should have those two sets equal. Indeed, even if it were a choice between rewards rather than gambles, there is nothing in our framework to prevent choice between basic rewards from being different conditional on events (a form of state-dependent preferences [36, 62]). We believe that these are sufficient reasons to be suspicious of this particular criticism.

Now, consider the first criticism of failing backward induction. The argument is that we should not do anything that is non-optimal in a subtree. This is really a weaker version of the second criticism (that we should be able to make free choice in subtrees), so it makes sense to consider both criticisms at once. These are quite strong arguments that are difficult to answer. In particular, if one cannot make local decisions, how does one react to changes in the tree? We are not aware of a satisfactory solution to this problem (see [23] for more information).

We turn instead to the following argument: if the normal form solution leads to a subtree in which the subject's local preferences overrule the solution, then the solution is flawed because the subject would just choose whatever they want in this region. In the setting of Hammond's potential addict, the subject should not say they will try something addictive but not more than once, because having tried it they will want to try it again, and will do so.

Following McClennen [51] and others, we consider the concept of present and future selves. At the root node, the subject is the present self, and has to take into account not only what she wants to happen in the future but also what her future selves will want to happen at the point at which they make decisions. The future selves, one for each future decision node, have preferences determined locally (only depending on their subtree). The potential addict's present self may wish her future self to take a certain decision, but in reality, the future self will ignore her.

Violations of subtree perfectness require some form of co-operation between the selves at different times, or an acceptance by the future selves of the authority of the past selves. McClennen [51] considers the concept of *resolute choice*, where the future selves agree to make some sacrifices in order to be resolute with the present self's decision. Consider Fig. 3.5 again. Suppose that the present self judges $X > Z > Y$ and the future self judges $Y > X > Z$. McClennen suggests that, should the present self choose to move to the future self's decision node (intending to move to X), the future self ought to choose X, co-operating with the present self's rejection of Z.

This form of co-operation may make some sense if X and Y are considered significantly better than Z for the future self. Imagine it is not present and future selves but two friends. If your friend had given you the choice between X and Y instead of the very unpleasant Z (that the friend wouldn't have really minded), but had indicated that she would like you to choose X, it seems churlish to choose Y. Jaffray [32] tried to quantify some sort of co-operation parameter, but, as he notes, this seems rather fictitious and arbitrary. Further, why would a future self even follow it? After all, the past self is no longer in a position to fight back! When considering the analogy of two friends, part of the reason the friend would choose X would be to retain his friend. The future self has neither the ability nor the motivation to retain the past self.

When subtree perfectness fails wildly, so the future self wants to do only actions that the past self definitely did not want, this resolute choice approach seems very tenuous. If a future self has a decision to make, why would it reject what it wants to do in favour of some past self? No satisfactory answer seems to exist. When subtree perfectness fails but backward induction works, a situation that apparently has not been considered by the proponents of this idea, it perhaps makes more sense.

Imagine the problem again as a decision between two friends. The first friend has $\mathrm{opt}(\{X, Y, Z\}) = \{X\}$. The second friend has $\mathrm{opt}(\{X, Y\}) = \{X, Y\}$. The first friend chooses to move towards X, whereupon the second friend has to option to go along with X or overrule the first friend. In the spirit of friendship, the second friend elects to resolve his indecision according to friendship, by accepting the first friend's wish, to move to X. We can assume that present and future selves should be friends, and so this behaviour seems much more acceptable. Assuming backward induction holds, then the future selves will always be satisfied with the present self's decision.[3]

This is not quite how things work with normal form solutions, since the present self picks all the choices for the future selves. But on the other hand there is now no reason for the future self to deviate: it might have choices that are not worse than the chosen decision, but none that is better, so why not go along with the initial plan?

Astute readers will observe that this argument could be framed in the opposite direction. Suppose we do not have subtree perfectness for normal form decisions, but instead the opposite inclusion in Definition 3.22. In this situation, the first self may have plans available that her future selves would disagree with. Her future selves this time have no optimal plans that she disagrees with. This is actually

[3]In fact, this does not even require backward induction, just subtree perfectness for normal form decisions (Section 3.1.4)

even more appealing than the backward induction argument, since no agreement is even required: the first self can choose any plan she likes, and the future selves can overrule it if they find it non-optimal. If the first self must choose a normal form decision, she may as well choose one that her future selves are happy with, because she knows that otherwise she will be overruled.

Without going into details (the proofs are effectively the same as those for subtree perfectness for normal form decisions), the required properties for this behaviour are modified versions of Preservation Under Addition and Backward Mixture, changed so that the direction of inclusion is changed in each. Interestingly, no type of conditioning property is needed here (they protect from problems when moving backwards, whereas this is a form of "forward induction"). Finding popular choice functions that obey these two modified properties is not easy, but $\mathrm{opt}_{\mathrm{maximin}}(\mathcal{X}|A) = \arg\max_{\omega\in A} X(\omega)$ is one, as is the corresponding maximax choice function, but interestingly none of the intermediate Hurwicz criteria (maximizing a convex combination of maximum and minimum reward) is.

Another advantage of this opposite form of subtree perfectness is that there is no difficulty if the decision tree changes mid-problem. The self at the time of the change can just pick a locally optimal strategy, knowing that the past selves would all be satisfied by it. A disadvantage would be computation: no backward induction would be available to find the overall solution (although it is likely that some form of backward induction could find some optimal strategies).

Chapter 4

Locality

This chapter is based on joint work with Ricardo Shirota Filho, adapted from [76].

4.1 Introduction

So far we have investigated a particular representation of sequential decision problems, namely decision trees. Although most problems that can be solved using choice functions on gambles should be representable as a decision tree, often it is more convenient and intuitive to describe them in different ways. Suppose, for example, that rewards are actually gained not at the end of a branch but at certain points along it. It seems natural to try to use this structure to simplify the problem and the solution. In this chapter and the next, we examine some particular structures of decision problem that allow such simplifications. It is noticeable that the underlying results and proofs follow similar lines to those we have seen earlier. This indicates that the approach of the previous two chapters will be useful in many circumstances.

The particular types of problem investigated are those in which rewards are not received at the end of the process, but rather some reward is generated after every stage. Thus, the overall reward is the sum of all these rewards. Such problems could be represented as a standard decision tree with the sum of all relevant rewards present on the reward leaves as usual. But instead, when working in a particular subtree, it may be possible to ignore all the rewards that have already been generated.

Consider the following simple example. A coin is going to be tossed a fixed number of times. At each stage, before the coin is tossed, the subject must choose one of two acts: predict the toss for this stage will be heads, or predict tails. A successful prediction is worth 1 utile, and an unsuccessful prediction is worth -1. Given a particular choice function, how can the subject determine an optimal policy? Clearly one could represent this problem as a standard decision tree and use the approach of the previous chapters. But this problem has particular structure: at each stage, a part of the total reward is received. Were we to want to apply backward induction, why not ignore all the rewards that have already been received? In terms of the decision tree, this essentially means that rewards are associated with chance arcs rather than terminal nodes. If representing the problem with an influence diagram [67], this would involve multiple value nodes (see for instance [73]).

This suggests the following method of solution. Each stage can be viewed as a separate decision problem: choose between the two bets. For a particular stage, we can find the optimal gambles conditional on every possible sequence of events that can lead up to that stage. All these local solutions can then be combined in all possible ways to form a set of normal form decisions for the global problem.

If this method agrees with $\mathrm{norm}_{\mathrm{opt}}$, we say that opt satisfies *locality*. Locality essentially represents a very strong form of subtree perfectness for particular decision problems: locality means that the global sequential problem can be solved by solving a local problem at each stage, for each stage taking only into account rewards incurred locally at that stage, and any events observed from previous stages. An important special case of a sequential decision problem which ought to satisfy locality is a sequence of unrelated (in the sense of rewards and information) decision problems. The latter example is philosophically particularly important, because it means that the intuitively logical answer—solving each decision problem separately—coincides with the answer where one models the sequential problem in full and solves them jointly in the normal form. Locality implies this not only for unrelated decision problems, but also for slightly more general situations where information and rewards at each stage depend on previous stages.

Since the coin-tossing example is not particularly interesting or useful, consider the ways it could be generalized.

- The possibility space could change from stage to stage. For example, introduce a second coin at stage 4.
- The possible actions could change from stage to stage. For example, the subject could be allowed to decline to bet in the first two stages, but then must bet from stage 3 onwards.
- The rewards could depend on the history. For example, successfully betting on heads could give a higher reward if relatively few heads have been observed.
- The possible actions could depend on the sequence of observed events. More generally, the possible actions could depend on the path taken through the problem, which corresponds to the sequence of both observed events and previously selected decisions.
- The possibility space at a particular stage could depend on the sequence of observed events.
- Act-state independence could be violated. For example, the subject could have the option to tamper with the coin. This would influence the coin's future behaviour.

In our investigation, we allow generalizations of the first three types. The fourth type is not allowed, but can be partially addressed as we shall see later. The last two types are not allowed. In other words,

- rewards can depend on the full state history,
- state spaces and decision spaces can depend on the stage.

Before we continue, it is useful to consider what locality means for a Markov decision process. Consider a classical Markov decision process with discount rate $0 \leq \gamma < 1$, and transition probabilities p^d_{st} and rewards r^d_{st} (real-valued, i.e. utilities) for going from state s to t under decision d. For a one-stage Markov decision process starting in state s_1, the optimal expected utility is

$$V_1(s_1) = \max_{d_1} \sum_{s_2} p^{d_1}_{s_1 s_2} r^{d_1}_{s_1 s_2} \tag{4.1}$$

For a two-stage process starting in state s_1, the optimal expected utility is

$$\begin{aligned} V_2(s_1) &= \max_{d_1, d_2(\cdot)} \sum_{s_2 s_3} p^{d_1}_{s_1 s_2} (r^{d_1}_{s_1 s_2} + \gamma p^{d_2(s_2)}_{s_2 s_3} r^{d_2(s_2)}_{s_2 s_3}) && (4.2) \\ &= \max_{d_1} \sum_{s_2} p^{d_1}_{s_1 s_2} (r^{d_1}_{s_1 s_2} + \gamma \max_{d_2} \sum_{s_3} p^{d_2}_{s_2 s_3} r^{d_2}_{s_2 s_3}) && (4.3) \\ &= \max_{d_1} \sum_{s_2} p^{d_1}_{s_1 s_2} (r^{d_1}_{s_1 s_2} + \gamma V_1(s_2)) && (4.4) \end{aligned}$$

In general,

$$V_0(s) = 0 \qquad\qquad V_{n+1}(s) = \max_{d} \sum_{t} p^{d}_{st} (r^{d}_{st} + \gamma V_n(t)) \tag{4.5}$$

This corresponds to the usual *value iteration* algorithm [72, Sec. 4.4] for finding optimal policies in infinite horizon Markov decision processes, with precisely one policy evaluation step and one policy improvement step at each stage.

Under act-state independence, transition probabilities do not depend on the decisions, and the solution turns out to be extremely simple:

$$V_{n+1}(s) = \left(\max_{d} \sum_{t} p_{st} r^{d}_{st} \right) + \gamma \sum_{t} p_{st} V_n(t) \tag{4.6}$$

In other words, at every stage, the optimal decision d^* can be obtained by solving a simple one-stage problem, and the sequential decision problem reduces to a sequence of static decision problems. In essence, this property is what we will call locality. Of course, under act-state dependence, locality is clearly no longer a reasonable requirement, since in such case Markov decision processes cannot be solved just locally, and backward induction techniques are required. As stated before, we will not be concerned with the act-state dependent case.

Throughout the section we clarify the concepts presented by using the coin-tossing example.

4.2 Problem Specification

The notation for this section is similar to that for decision trees, but requires some refinements. As we will be concerned with sequential decision processes, we will consider for Ω a Cartesian product of *state spaces* S_0, S_1, S_2, ..., S_n:

$$\Omega = S_0 \times S_1 \times \cdots \times S_n$$

where S_0 is the set of possible states of the system at time 0, and so on. Particular elements of these spaces are denoted by s_0, s_1, ..., s_n. We identify any such element s_k also with an event $E_{s_k} = \{(s'_0, \dots, s'_n) \colon s'_k = s_k\}$. For brevity, we will sometimes write s_k instead of E_{s_k} when no confusion is possible, for instance when conditioning.

The states are observed sequentially, and after each observation, we can take a decision from some set, and receive a reward from a set $\mathcal{R}$. Let h_k be the history of *events* up to stage k, so $h_k = (s_0, \dots, s_k)$. We can describe the process as:

- observe $s_0 \in S_0$,
- choose $d_1 \in D_1$, observe $s_1 \in S_1$, receive $r_1(s_0 d_1 s_1) = r_1(h_0 d_1 s_1)$,
- choose $d_2 \in D_2$, observe $s_2 \in S_2$, receive $r_2(s_0 s_1 d_2 s_2) = r_2(h_1 d_2 s_2)$,
- ...
- choose $d_n \in D_n$, observe $s_n \in S_n$, receive $r_n(h_{n-1} d_n s_n)$.

The total reward resulting from the above process is again assumed to be an element of $\mathcal{R}$. More precisely, we now need to assume an operator $+$ on $\mathcal{R}$ which maps every two elements r and r' of $\mathcal{R}$ to another element $r + r'$ of $\mathcal{R}$. To avoid some technical details in results, we may assume that $+$ has an identity element 0; no further properties of $+$ are assumed. To avoid many brackets in our notation, we always let $+$ evaluate from right to left. The total reward is assumed to be:

$$r_1(h_0 d_1 s_1) + r_2(h_1 d_2 s_2) + \cdots + r_n(h_{n-1} d_n s_n)$$

Finally, we assume that our preferences over any non-empty finite set $\mathcal{X}$ of gambles on $F_{k+1} = S_{k+1} \times \cdots \times S_n$ (where $0 \leq k < n$), given any event $h_k = s_0 \dots s_k$, can be represented by a choice function $\mathrm{opt}(\mathcal{X}|h_k)$. At this point it becomes clear that the apparent restriction that D_k must not depend on h_{k-1} is irrelevant, because the *rewards* can depend on h_{k-1}. So, if one wanted a decision to become available only for certain h_{k-1}, then one could simply make that decision's gamble have terrible rewards for other possible histories. This ensures that the choice function will only choose that decision for histories where it is allowed. In short, making a decision unavailable and making a decision unacceptable are practically equivalent.

The process described above is a special case of a sequential decision problem (or, decision tree), in the sense that:

- chance and decision nodes follow each other consecutively, hence the problem consists of clearly defined *stages*,
- a variable is observed regardless of the history (for instance, you cannot decide to observe a different variable),
- at each stage, a decision incurs a reward, which may depend on the state history, the current decision, and the next state, but not on anything else,
- the final reward is obtained through combining local rewards.

In our toy example (sequential coin tossing), at each stage, the agent must bet on the outcome—hence, there are also two possible decisions: heads or tails—and the agent loses one utile if wrong, but gains one utile if right. So,

$$S_1 = S_2 = \cdots = S_n = \{H, T\},$$
$$D_1 = D_2 = \cdots = D_n = \{d_H, d_T\},$$

and

$$r_k(h_{k-1} d_k s_k) = \begin{cases} 1 & \text{if } d_k \text{ matches } s_k, \\ -1 & \text{otherwise.} \end{cases}$$

Note that the initial state s_0 is not of relevance in this problem. For mathematical convenience, we can simply let S_0 be a singleton.

4.3 Normal Form Solution

4.3.1 Normal Form Decisions

Normal form decisions can be constructed as usual. Consider our decision process after a particular sequence of decisions and states $s_0 d_1 s_1 \dots d_{k-1} s_{k-1}$ has already occurred. A normal form decision for our sequential decision process at this stage consists of a specification of a decision $d_k \in D_k$ and decision functions $d_{k+1}(\cdot)\colon S_k \to D_{k+1}$, ..., and $d_n(\cdot)\colon S_k \times \dots \times S_{n-1} \to D_n$:

- $d_k \in D_k$,
- $d_{k+1}(s_k) \in D_{k+1}$,
- $\dots$
- $d_n(s_k \dots s_{n-1}) \in D_n$,

The set of all normal form decisions is denoted by Π_k^n:

$$\Pi_k^n = D_k \times D_{k+1}^{S_k} \times \dots \times D_n^{S_k \times \dots \times S_{n-1}}$$

In our toy example, normal form decisions include 'always bet tails', 'always bet heads', 'bet tails on odd stages, and heads on even stages', and 'bet tails if we have observed more tails than heads in the past, otherwise bet heads'. Of course there are many others.

4.3.2 Gambles

As usual, we need to find the gambles associated with normal form decisions. Each state history $s_0 s_1 \dots s_{k-1} = h_{k-1}$ in H_{k-1} and each normal form decision $\pi_k^n = (d_k, d_{k+1}(\cdot), \dots, d_n(\cdot))$ in Π_k^n incurs a gamble $X_k^n(h_{k-1}, \pi_k^n)$, that is, a mapping from $S_k \times \dots \times S_n$ to $\mathcal{R}$:

$$\bigoplus_{s_k} E_{s_k} \bigoplus_{s_{k+1}} E_{s_{k+1}} \dots \bigoplus_{s_n} E_{s_n} \Big(r_k(h_{k-1} d_k s_k) + r_{k+1}\big(h_k d_{k+1}(h_k) s_{k+1}\big) + \dots$$
$$\dots + r_n\big(h_{n-1} d_n(h_{n-1}) s_n\big)\Big)$$

which we can also write as

$$\bigoplus_{s_k} E_{s_k}\Bigg(r_k(h_{k-1}d_k s_k) + \bigoplus_{s_{k+1}} E_{s_{k+1}}\Bigg(r_{k+1}\big(h_k d_{k+1}(h_k) s_{k+1}\big) + \dots$$
$$\dots + \bigoplus_{s_n} E_{s_n} r_n\big(h_{n-1} d_n(h_{n-1}) s_n\big)\Bigg)\dots\Bigg)$$

For the sake of brevity, we have slightly abused notation: $d_\ell(h_{\ell-1})$ denotes of course simply $d_\ell(s_k \dots s_{\ell-1})$.

The gamble $X_k^n(h_{k-1}, \pi_k^n)$ describes the reward $X_k^n(h_{k-1}, \pi_k^n)(f_k)$ that we receive for each possible sequence of future states $f_k = (s_k, \dots, s_n)$ when we follow normal form decision π_k^n after having observed state history h_{k-1}.

The gamble $X_k^n(h_{k-1}, \pi_k^n)$ generalizes the classical *value function* $V(h_{k-1}, \pi_k^n)$ *of a policy at a given state history* to the case where probabilities are not given and rewards are not assumed to be expressed in utiles. Indeed, if we were given a probability $p(\cdot|h_{k-1})$ on $F_k = S_k \times \dots \times S_n$, and rewards were expressed in utiles, then the expected value of a given normal form decision π_k^n at state history h_{k-1} would be precisely

$$V(h_{k-1}, \pi_k^n) = \sum_{f_k} p(f_k|h_{k-1}) X_k^n(h_{k-1}, \pi_k^n)(f_k)$$

Again, considering our toy example, the gamble corresponding to the normal form decision 'always bet tails' would be:

$$X_k^n(h_{k-1}, \pi_k^n)(f_k) = n_T - n_H$$

where n_T is the number of tails in f_k, and n_H the number of heads in f_k, because under the given normal form decision 'always bet tails', we gain one utile for each tail in f_k, and lose one utile for each head in f_k. As an other example, in a single stage problem, the normal form decision 'bet heads' has gamble:

$$X_k^k(h_{k-1}, d_H)(s_k) = \begin{cases} 1 & \text{if } s_k = H \\ -1 & \text{if } s_k = T \end{cases}. \tag{4.7}$$

More elaborate normal form decisions become quickly complicated, even in this simple case.

4.3.3 Normal Form Solution

A normal form decision π_k^n is optimal for a given state history h_{k-1} if

$$X_k^n(h_{k-1}, \pi_k^n) \in \mathrm{opt}(\mathcal{X}_k^n(h_{k-1})|h_{k-1})$$

where $\mathcal{X}_k^n(h_{k-1})$ is the set of gambles incurred by all normal form decisions when we start from state history h_{k-1}. The set of all optimal normal form decisions, for a given state history h_{k-1}, is the normal form solution and is denoted by $\Pi_k^n(h_{k-1})$. So,

$$\begin{aligned}\Pi_k^n(h_{k-1}) &= \arg\operatorname*{opt}_{\pi_k^n \in \Pi_k^n}(X_k^n(h_{k-1}, \pi_k^n)|h_{k-1}) \\ &= \{\pi_k^n \in \Pi_k^n \colon X_k^n(h_{k-1}, \pi_k^n) \in \mathrm{opt}(\mathcal{X}_k^n(h_{k-1})|h_{k-1})\}.\end{aligned}$$

If we instead drew the decision tree corresponding to the problem, then this would be just the same as applying $\mathrm{norm}_{\mathrm{opt}}$ to the tree.

The set of optimal gambles $\mathrm{opt}(\mathcal{X}_k^n(h_{k-1})|h_{k-1})$ generalises of course the classical *(optimal) value function $V(h_{k-1})$ at a given state history* to the case where probabilities are not given and rewards are not assumed to be expressed in utiles. Indeed, if we were given a joint probability p on $F_k = S_k \times \cdots \times S_n$, and rewards were expressed in utiles, then the (optimal) value function at h_{k-1} would be

$$V(h_{k-1}) = \max_{\pi_k^n \in \Pi_k^n} V(h_{k-1}, \pi_k^n)$$

which is, for this case, the expectation of (any element of) $\mathrm{opt}(\mathcal{X}_k^n(h_{k-1})|h_{k-1})$.

4.4 Locality

As seen previously, under act-state independence, we can solve an n-stage Markov decision process simply by solving n one-stage ones. We now generalize this idea.

To express locality conveniently, we first introduce some further notation. Let $\Pi_k^k(\cdot)$ denote all locally optimal decision functions:

$$\Pi_k^k(\cdot) = \{d_k(\cdot) \in (D_k)^{H_{k-1}} \colon d_k(h_{k-1}) \in \Pi_k^k(h_{k-1})\}$$

(It may be useful at this point to recall that $\Pi_k^k = D_k$.) More generally,

$$\Pi_k^n(\cdot) = \{(d_k(\cdot), d_{k+1}(\cdot), \dots, d_n(\cdot)) \in (\Pi_k^n)^{H_{k-1}} : \\ \big(d_k(h_{k-1}), d_{k+1}(h_{k-1}\cdot), \dots, d_n(h_{k-1}\cdot)\big) \in \Pi_k^n(h_{k-1})\}$$

where we used the identity

$$\begin{aligned}(\Pi_k^n)^{H_{k-1}} &= \Big(D_k \times D_{k+1}^{S_k} \times \dots \times D_n^{S_k \times \dots \times S_{n-1}}\Big)^{H_{k-1}} \\ &= D_k^{H_{k-1}} \times D_{k+1}^{H_{k-1} \times S_k} \times \dots \times D_n^{H_{k-1} \times S_k \times \dots \times S_{n-1}} \\ &= D_k^{H_{k-1}} \times D_{k+1}^{H_k} \times \dots \times D_n^{H_{n-1}}\end{aligned}$$

With any opt on S_0, …, S_n, we can associate the following property:

Property 17 (Locality). *A choice function* opt *satisfies* locality *on S_0, …, S_n whenever, for each sequential decision process on S_0, …, S_n and each $1 \le k < n$,*

$$\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^{k+1}(\cdot) \times \dots \times \Pi_n^n(\cdot).$$

For instance, in the coin-tossing example, we can work out at each stage and for every different history whether it is optimal to bet on heads, tails, or either. If locality holds, then the optimal strategies are all combinations of these local decisions.

Perhaps surprisingly, locality on two stage problems is (necessary and) sufficient for locality on problems of any number of stages:

Lemma 4.1. *For every $1 \le k < n$,*

$$\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^n(\cdot) \tag{4.8}$$

if and only if, for every $1 \le k < n$,

$$\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^{k+1}(\cdot) \times \dots \times \Pi_n^n(\cdot).$$

Proof. Apply the given equality recursively:

$$\begin{aligned}\Pi_k^n(\cdot) &= \Pi_k^k(\cdot) \times \Pi_{k+1}^n(\cdot) \\ &= \Pi_k^k(\cdot) \times \Pi_{k+1}^{k+1}(\cdot) \times \Pi_{k+2}^n(\cdot) \\ &= \dots \\ &= \Pi_k^k(\cdot) \times \Pi_{k+1}^{k+1}(\cdot) \times \dots \times \Pi_n^n(\cdot)\end{aligned}$$

□

4.4.1 Sequential Distributivity

Following our usual approach, we now show that one simple condition on opt is equivalent to locality. Recall $H_k = S_0 \times \cdots \times S_k$, and $F_k = S_k \times \cdots \times S_n$. With any opt on $S_0, \ldots, S_n$, we can associate the following property:

Property 18 (Sequential Distributivity)**.** *For any $1 \leq k < n$, any value h_{k-1} of H_{k-1}, all finite sets of gambles $\mathcal{X}$ on S_k, all finite sets of gambles $\mathcal{Y}(s_k)$ on F_{k+1} (one such set for each $s_k \in S_k$), and all $X \in \mathcal{X}$ and $Y(s_k) \in \mathcal{Y}(s_k)$:*

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle| h_{k-1}\right)$$
$$\iff X \in \operatorname{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

We shall show that the above property is necessary and sufficient for our normal form solution induced by opt to reduce to a sequence of n single-stage normal form solutions.

Theorem 4.2. *A choice function* opt *satisfies locality if and only if* opt *satisfies sequential distributivity.*

Proof. Before we start the proof, note that we can write Π_k^n as

$$\begin{aligned}\Pi_k^n &= D_k \times D_{k+1}^{S_k} \times \cdots \times D_n^{S_k \times \cdots \times S_{n-1}} \\ &= D_k \times \left(D_{k+1} \times D_{k+2}^{S_{k+1}} \times \cdots \times D_n^{S_{k+1} \times \cdots \times S_{n-1}}\right)^{S_k} \\ &= D_k \times (\Pi_{k+1}^n)^{S_k}\end{aligned}$$

In other words, any normal form decision π_k^n can be interpreted as a 2-stage normal form decision, where we first choose $d_k \in D_k$, and then $\pi_{k+1}^n(s_k) \in \Pi_{k+1}^n$ depending on the event s_k that occurred. Throughout the proof, we will make use of this correspondence, and typically write π_k^n as $(d_k, \pi_{k+1}^n(\cdot))$ instead of the more usual $(d_k, d_{k+1}(\cdot), \ldots, d_n(\cdot))$.

"if". We must show that opt satisfies locality, given that it satisfies sequential distributivity. By Lemma 4.1, it suffices to show that $\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^n(\cdot)$ for all $1 \leq k < n$.

Consider any $1 \leq k < n$. For every h_{k-1}, we must show that $\pi_k^{n*} = (d_k^*, \pi_{k+1}^{n*}(\cdot)) \in \Pi_k^n(h_{k-1})$ if and only if $d_k^* \in \Pi_k^k(h_{k-1})$ and $\pi_{k+1}^{n*}(s_k) \in \Pi_{k+1}^n(h_{k-1}s_k)$ for every s_k. Equivalently, we show that, regardless of h_{k-1},

$$X_k^n(h_{k-1}, \pi_k^{n*}) \in \operatorname{opt}(\mathcal{X}_k^n(h_{k-1})|h_{k-1}) \tag{4.9}$$

if and only if

$$X_k^k(h_{k-1}, d_k^*) \in \operatorname{opt}(\mathcal{X}_k^k(h_{k-1})|h_{k-1}) \text{ and} \tag{4.10}$$

$$X_{k+1}^n(h_{k-1}s_k, \pi_{k+1}^{n*}(s_k)) \in \operatorname{opt}(\mathcal{X}_{k+1}^n(h_{k-1}s_k)|h_{k-1}s_k) \text{ for all } s_k. \tag{4.11}$$

First, note that, for any $\pi_k^n = (d_k, \pi_{k+1}^n(\cdot)) \in \Pi_k^n$,

$$X_k^n(h_{k-1}, \pi_k^n) = X_k^k(h_{k-1}, d_k) + \bigoplus_{s_k} E_{s_k} X_{k+1}^n(h_{k-1}s_k, \pi_{k+1}^n(s_k)) \tag{4.12}$$

simply by the definitions of $X_k^n(h_{k-1}, \pi_k^n)$, $X_k^k(h_{k-1}, d_k)$, and $X_{k+1}^n(h_{k-1}s_k, \pi_{k+1}^n(s_k))$. Taking the union over all $\pi_k^n \in \Pi_k^n$, it follows that also

$$\mathcal{X}_k^n(h_{k-1}) = \mathcal{X}_k^k(h_{k-1}) + \bigoplus_{s_k} E_{s_k} \mathcal{X}_{k+1}^n(h_{k-1}s_k). \tag{4.13}$$

Now, simply apply sequential distributivity, and use Eqs. (4.12) and (4.13) to see that Eq. (4.9) holds if and only if Eqs. (4.10) and (4.11) hold.

"only if". Let $1 \leq k < n$. By the locality assumption, we know that $\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^n(\cdot)$. We show that sequential distributivity holds.

Indeed, consider any value h_{k-1} of H_{k-1}, any set of gambles $\mathcal{X} = \{X_1, \dots, X_p\}$ on S_k, any finite sets of gambles $\mathcal{Y}(s_k) = \{Y_1(s_k), \dots, Y_q(s_k)\}$ on F_{k+1} (one such set for each $s_k \in S_k$)—for simplicity of notation, for each s_k we use an index set for $\mathcal{Y}(s_k)$ of the same size q: this goes without loss of generality as we can always allow some of the $Y_j(s_k)$ to be equal to one another—and any $X_i \in \mathcal{X}$ and $Y_{j(s_k)}(s_k) \in \mathcal{Y}(s_k)$—so $j(\cdot)$ is a mapping from S_k to $\{1, \dots, q\}$.

We must show that

$$X_i + \bigoplus_{s_k} E_{s_k} Y_{j(s_k)}(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k) \middle| h_{k-1}\right)$$
$$\iff X_i \in \operatorname{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y_{j(s_k)}(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k. \tag{4.14}$$

To this end, consider any n-stage process where

$$\begin{aligned} D_k &= \{x_1, \ldots, x_p\} \\ D_{k+1} = \cdots = D_{n-1} &= \{d\} \\ D_n &= \{y_1, \ldots, y_q\} \end{aligned}$$

and

$$\begin{aligned} r_k(h_{k-1}x_i s_k) &= X_i(s_k) \\ r_{k+1}(h_k d s_{k+1}) = \cdots = r_{n-1}(h_{n-2} d s_{n-1}) &= 0 \\ r_n(h_{n-1} y_j s_n) &= Y_{j(s_k)}(s_k)(f_{k+1}) \end{aligned}$$

where 0 is the identity element in $\mathcal{R}$ with respect to $+$. Observe that, for this process,

$$\begin{aligned} \mathcal{X}_k^k(h_{k-1}) &= \mathcal{X}, \\ \mathcal{X}_{k+1}^n(h_{k-1}s_k) &= \mathcal{Y}(s_k), \end{aligned}$$

and, as shown before,

$$\mathcal{X}_k^n(h_{k-1}) = \mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k).$$

But, note that $\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^n(\cdot)$ means that, for any $\pi_k^n = (x_i, y_{j(\cdot)}) \in \Pi_k^n$,

$$\begin{aligned} &(x_i, y_{j(\cdot)}) \in \Pi_k^n(h_{k-1}) \\ &\qquad \iff x_i \in \Pi_k^k(h_{k-1}) \text{ and } y_{j(s_k)} \in \Pi_{k+1}^n(h_{k-1}s_k) \text{ for all } s_k. \end{aligned}$$

By definition of $\Pi_k^n(h_{k-1})$, $\Pi_k^k(h_{k-1})$, and $\Pi_{k+1}^n(h_{k-1}s_k)$, and because of what we have just shown about $\mathcal{X}_k^n(h_{k-1})$, $\mathcal{X}_k^k(h_{k-1})$, and $\mathcal{X}_{k+1}^n(h_{k-1}s_k)$, this is exactly equivalent to Eq. (4.14). □

Chapter 5

Optimal Control and Deterministic Discrete-Time Systems

5.1 Problem Specification

In this chapter we formalize and extend the results of de Cooman and Troffaes [13] for deterministic discrete-time systems with uncertain gains. This is a branch of *control theory* [7, 47], which more generally covers the behaviour and control of a multitude of dynamic systems. The particular class of systems we investigate is best illustrated by example. Consider Fig. 5.1. This depicts a system that starts at N_1, and can reach N_4 by multiple paths. The subject, who controls the system, can choose the path the system will take. Travelling down a particular arc gives the subject an associated reward. For instance, choosing the arc from N_1 to N_2 will give the subject X. The subject's task is to find the optimal path for the system to take.

This is an example of a *deterministic discrete-time system*. If V, W, X, Y, Z are certain rewards, so the subject knows exactly what she will receive when choosing a particular route, this is a system with *certain gains*. This would be a relatively trivial type of problem to solve: simply find the path with the highest total reward. We instead consider systems with uncertain gains, allowing V, W, X, Y, Z to be gambles. The overall reward for a particular path is then determined by the sum of the gambles

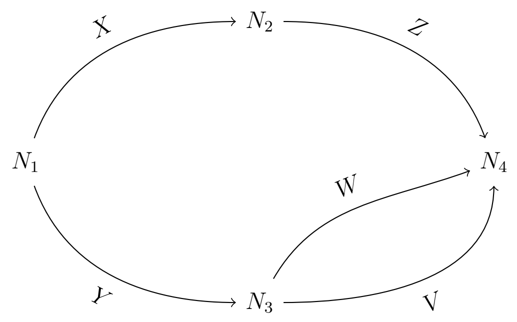

Figure 5.1: A simple deterministic system.

for all arcs in the path. The sum is performed using the $+$ operator introduced in Section 4.1; we also later need to assume that $r_1 + r_2 = r_1 + r_3$ implies $r_2 = r_3$; this holds for instance if every reward $r \in \mathcal{R}$ has a left inverse $-r \in \mathcal{R}$, so $(-r) + r$ equals the left identity element 0.

With uncertain rewards, there are two possible concepts of a normal form decision. If the subject receives the reward from a gamble as soon as that arc is chosen, and is able to react to it, then she can use this information to choose her next arc. For example, an informal strategy for Fig. 5.1 could be "choose Y, and then choose W if Y has given a large reward, V otherwise". Alternatively, the subject may not be allowed to do this, and so could have no strategy more complicated than, say "choose Y, then W". This could be because she does not learn of the outcome of Y until later, or because she has no power to change her strategy while the system operates. The latter set-up is followed by de Cooman and Troffaes, and so we follow it too. Normal form decisions are thus very simple, and no concept of conditioning is required either. Also, since in this case everything is completely deterministic until the final decision has been made, it seems natural to use normal form solutions for such a model.

These problems can be represented as decision trees, such as Fig. 5.2 (truncated at the chance nodes, since we have not defined what comes after). We could use the backward induction method from Section 3.1, but this would not be particularly helpful or interesting. Instead, we can exploit the special structure of the problem and employ a type of backward induction developed by Bellman [7], usually called

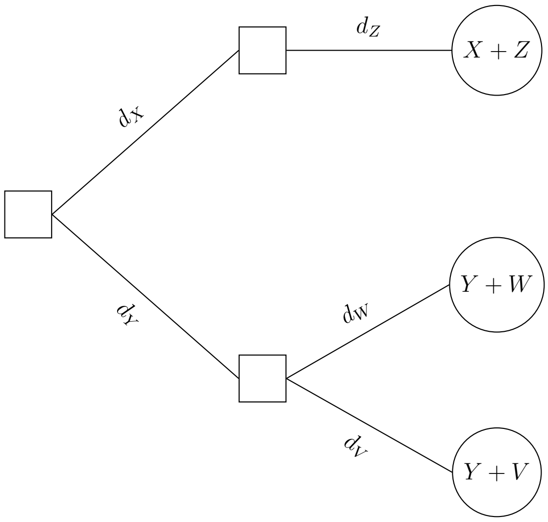

Figure 5.2: The decision tree for Fig. 5.1.

dynamic programming. First, we would find opt($\{W, V\}$). Suppose this is $\{W\}$. Then, we would find opt($\{X + W, Y + Z\}$) and this would determine the normal form solution. If opt($\{W, V\}$) $= \{W, V\}$, we would instead calculate opt($\{X + W, X + V, Y + Z\}$), and so on.

To formalize this idea, it is clearer and more convenient to represent the systems as a different sort of tree, in which the chance node and reward nodes are removed entirely and the gambles are placed on arcs. A *deterministic system tree* is a rooted tree of nodes of two types: decision nodes and terminal nodes. All branches end with a terminal node, and all terminal nodes appear at the end of branches. Every arc corresponds to a decision, and each arc has an associated gamble. The deterministic system tree for Fig. 5.1 is shown in Fig. 5.3 (where the terminal nodes lie at the end of each branch, but are invisible). It must be emphasised that, although gambles are acquired upon choosing a decision arc, their value is not discovered until the terminal node is reached. Therefore there is no learning or conditioning involved in this model.

We are interested in determining when this dynamic programming method gives the same answer as norm_{opt}. Bellman [7] introduced his famous *principle of optimality* to help answer this question.

An optimal policy has the property that, whatever the initial state

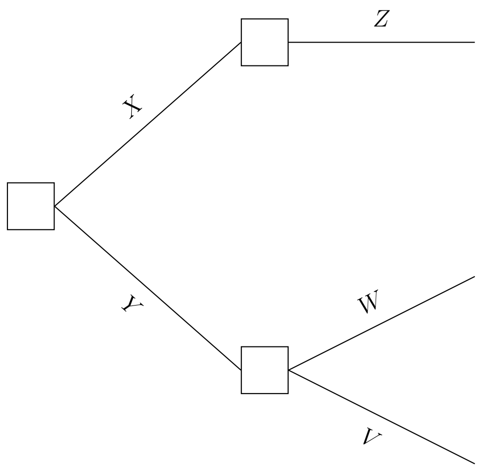

Figure 5.3: The deterministic system tree for Fig. 5.1.

> and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

We recognize this as subtree perfectness for normal form decisions (see p. 77). Although Bellman states the principle in terms of the first decision only, it is clear that it implies that the restriction of an optimal policy to any subtree must be optimal. We formalize the principle into the following property.

Property 19 (Bellman's principle of optimality)**.** *A normal form operator* norm *satisfies the principle of optimality if, for any deterministic system tree T, and any node N in at least one element of* $\mathrm{norm}(T)$*,*

$$\mathrm{st}_N(\mathrm{norm}(T)) \subseteq \mathrm{norm}(\mathrm{st}_N(T)).$$

Equivalently, for any normal form decision $U \in \mathrm{norm}(T)$ and any node N in U,

$$\mathrm{st}_N(U) \in \mathrm{norm}(\mathrm{st}_N(T)).$$

As is the case in Chapter 3, and as shown by de Cooman and Troffaes, Bellman's Principle of Optimality is not sufficient for backward induction to work: it again needs to be augmented by Insensitivity To Omission. Our contribution is to show that Bellman's Principle of Optimality can be reduced to two simple conditions on opt, then to formulate de Cooman and Troffaes's backward induction theorem into the langauge of deterministic system trees, and finally to extend the theory to cover subtree perfectness.

The proofs proceed on similar lines to previous chapters, showing that our approach is useful for several types of model, and requires surprisingly few notational changes. First, any deterministic system trees $T_1, \dots, T_n$ joined at a decision node, with X_i the gamble on the arc to T_i, can be written as

$$\bigsqcup_{i=1}^{n} X_i T_i.$$

Normal form decisions and solutions are defined as usual. In particular, a normal form decision in a deterministic system tree is a path from the root node to a terminal node. The gamble associated with a normal form decision is simply the sum of the gambles for all the arcs in the path. The gamb operator is as usual defined as

$$\operatorname{gamb}(T) = \bigcup_{U \in \operatorname{nfd}(T)} \operatorname{gamb}(U).$$

The recursive definition from Definition 1.12 now becomes

$$\operatorname{gamb}\left(\bigsqcup_{i=1}^{n} X_i T_i\right) = \bigcup_{i=1}^{n} (X_i + \operatorname{gamb}(T_i)),$$

where we use the notation

$$X + \mathcal{Y} = \{X + Y : Y \in \mathcal{Y}\}.$$

The operator $\operatorname{norm}_{\mathrm{opt}}$ can be defined as usual, and interestingly so can $\operatorname{back}_{\mathrm{opt}}$:

$$\operatorname{back}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} X_i T_i\right) = \operatorname{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} X_i \operatorname{back}_{\mathrm{opt}}(T_i)\right).$$

5.2 Backward Induction Theorem

In this section we show that Bellman's Principle of Optimality and Insensitivity To Omission are necessary and sufficient for backward induction to work. We also show that Bellman's Principle of Optimality is equivalent to Preservation Under Addition and a new property, Backward Addition. Note that all properties in this section are the unconditional versions.

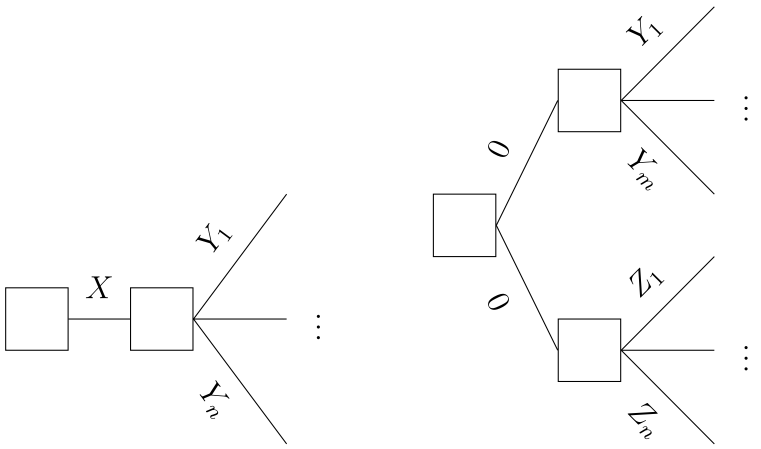

Figure 5.4: Decision trees for Theorem 5.1.

Property 20 (Backward Addition Property)**.** *For any gamble X and any non-empty finite set of gambles $\mathcal{Y}$,*

$$\mathrm{opt}(X + \mathcal{Y}) \subseteq X + \mathrm{opt}(\mathcal{Y}).$$

Theorem 5.1. $\mathrm{norm}_{\mathrm{opt}}$ *satisfies Bellman's Principle of Optimality if and only if* opt *satisfies Preservation Under Addition and Backward Addition.*

Proof. "only if". Let X be a gamble and $\mathcal{Y} = \{Y_1, \dots, Y_n\}$ be a set of gambles. Consider the upper tree in Fig. 5.4. If $X + Y_k \in \mathrm{opt}(X + \mathcal{Y})$, then by Bellman's Principle of Optimality it follows that $Y \in \mathrm{opt}(\mathcal{Y})$, hence Backward Addition holds. Next, consider the lower tree. Let $\mathcal{Y} = \{Y_1 \dots, Y_m\}$, $\mathcal{Z} = \{Z_1, \dots, Z_n\}$ and suppose $\mathcal{Y} \cap \mathcal{Z} = \emptyset$. Now let $\mathcal{X} = \mathcal{Y} \cup \mathcal{Z}$. By Bellman's Principle of Optimality we know that if $Y \in \mathcal{Y} \cap \mathrm{opt}(\mathcal{X})$, then $Y \in \mathrm{opt}(\mathcal{Y})$, hence Preservation Under Addition holds.

"if". We proceed by structural induction. Let T be a deterministic system tree. The base step, to show the result when T consists of a terminal node only, is trivial. The inductive step is to suppose that Bellman's Principle of Optimality holds for every $\mathrm{st}_K(T)$ where $K \in \mathrm{ch}(T)$, and then show that Bellman's Principle of Optimality holds for T. By Lemma 3.24, we need only show that for every $K \in \mathrm{ch}(T)$ that is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$,

$$\mathrm{st}_K(\mathrm{norm}_{\mathrm{opt}}(T)) \subseteq \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_K(T)).$$

So, the proof is established if we can show that, for every $U \in \mathrm{norm}_{\mathrm{opt}}(T)$ passing through $K \in \mathrm{ch}(T)$,

$$\mathrm{st}_K(U) \in \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_K(T)). \tag{5.1}$$

We now express this in terms of gambles—but first we introduce some notation.

Let $\mathrm{ch}(T) = \{K_1, \dots, K_n\}$, and $K = K_k$. Let $\mathrm{gamb}(\mathrm{st}_{K_i}(T)) = \mathcal{Y}_i$, and let X_i be the gamble corresponding to the arc to K_i. That is,

$$T = \bigsqcup_{i=1}^{n} X_i \,\mathrm{st}_{K_i}(T).$$

Recall, U contains the node K_k, so $\mathrm{gamb}(U) = X_k + Y_k$ for some $Y_k \in \mathcal{Y}_k$.

Now, because $U \in \mathrm{norm}_{\mathrm{opt}}(T)$, we know that

$$X_k + Y_k \in \mathrm{opt}(\mathrm{gamb}(T)) = \mathrm{opt}\left(\bigcup_{i=1}^{n}(X_i + \mathcal{Y}_i)\right). \tag{5.2}$$

To establish Eq. (5.1), we must simply show that $Y_k \in \mathrm{opt}(\mathcal{Y}_k)$.

Indeed. Obviously,

$$X_k + \mathcal{Y}_k \subseteq \bigcup_{i=1}^{n}(X_i + \mathcal{Y}_i).$$

Applying Preservation Under Addition,

$$\mathrm{opt}(X_k + \mathcal{Y}_k) \supseteq \mathrm{opt}\left(\bigcup_{i=1}^{n}(X_i + \mathcal{Y}_i)\right) \cap (X_k + \mathcal{Y}_k).$$

However, by Eq. (5.2), $X_k + Y_k$ belongs to the right hand side, whence, it must also belong to the left hand side. Now, apply Backward Addition, to see that indeed $Y_k \in \mathrm{opt}(\mathcal{Y}_k)$. This completes the inductive step. □

Theorem 5.2. *Let* opt *be any choice function. The following conditions are equivalent.*

(A) For any deterministic system tree T, it holds that $\mathrm{back}_{\mathrm{opt}}(T) = \mathrm{norm}_{\mathrm{opt}}(T)$.

(B) For any strategically equivalent deterministic system trees, T_1 and T_2, it holds that

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)).$$

(C) opt *satisfies Insensitivity To Omission and Bellman's Principle of Optimality.*

Lemma 5.3. *If, for all strategically equivalent deterministic system trees T_1 and T_2, it holds that*

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)),$$

then opt *satisfies Insensitivity To Omission.*

Proof. Let $\mathcal{X}$ and $\mathcal{Y} = \{Y_1, \dots, Y_n\}$ be sets of gambles such that $\mathrm{opt}(\mathcal{X}) \subseteq \mathcal{Y} \subseteq \mathcal{X}$. Let T_1 be a deterministic system tree with just one decision node and $\mathrm{gamb}(T_1) = \mathcal{X}$. Let T_2 be a deterministic system tree constructed as follows: there is one decision arc with gamble 0 that leads to T_1, and n other decision arcs, each leading immediately to a terminal node, with gambles Y_1 to Y_n. Clearly, $\mathrm{gamb}(T_2) = \mathcal{X}$. We have

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)) = \mathrm{opt}(\mathrm{opt}(\mathcal{X}) \cup \mathcal{Y}) = \mathrm{opt}(\mathcal{Y}).$$

because $\mathrm{opt}(\mathcal{X}) \subseteq \mathcal{Y}$. Since $\mathrm{back}_{\mathrm{opt}}$ is assumed to preserve strategic equivalence, and T_1 and T_2 are strategically equivalent by construction, it follows that $\mathrm{opt}(\mathcal{Y}) = \mathrm{opt}(\mathcal{X})$, as required. □

Lemma 5.4. *If, for all strategically equivalent deterministic system trees T_1 and T_2, it holds that*

$$\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)),$$

then $\mathrm{norm}_{\mathrm{opt}}$ *satisfies Bellman's Principle of Optimality.*

Proof. We show that opt must satisfy Preservation Under Addition and Backward Addition and invoke Theorem 5.1. We can again use the two trees from Fig. 5.4. Let the upper tree be called T_1, and let T_2 be a tree with only one decision node and $\mathrm{gamb}(T_2) = X + \mathcal{Y}$. Then,

$$\begin{aligned}\mathrm{opt}(X + \mathcal{Y}) &= \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_2)) \\ &= \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) \\ &= \mathrm{opt}(X + \mathrm{opt}(\mathcal{Y})) \subseteq X + \mathrm{opt}(\mathcal{Y}),\end{aligned}$$

so Backward Addition holds.

Let T_1 be the lower tree in Fig. 5.4, with $\{\mathcal{Y}, \mathcal{Z}\}$ a partition of $\mathcal{X}$. Let T_2 have one decision node and $\mathrm{gamb}(T_2) = \mathcal{X}$. As assumed, $\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) =$

$\mathrm{opt}(\mathrm{opt}(\mathcal{Y}) \cup \mathrm{opt}(\mathcal{Z})) = \mathrm{opt}(\mathcal{X})$. So,

$$\begin{aligned}\mathrm{opt}(\mathcal{X}) \cap \mathcal{Y} &= \mathrm{opt}(\mathrm{opt}(\mathcal{Y}) \cup \mathrm{opt}(\mathcal{Z})) \cap \mathcal{Y} \\ &\subseteq (\mathrm{opt}(\mathcal{Y}) \cup \mathrm{opt}(\mathcal{Z})) \cap \mathcal{Y} \\ &= \mathrm{opt}(\mathcal{Y}) \cap \mathcal{Y} = \mathrm{opt}(\mathcal{Y}),\end{aligned}$$

so Preservation Under Addition holds. □

Proof of Theorem 5.2. (A) $\Longrightarrow$ (B). Immediate, since for strategically equivalent trees, $\mathrm{norm}_{\mathrm{opt}}(T_1) = \mathrm{norm}_{\mathrm{opt}}(T_2)$ by definition.

(B) $\Longrightarrow$ (C). See Lemmas 5.3 and 5.4.

(C) $\Longrightarrow$ (A). We proceed by structural induction. The base step is trivial. The induction hypothesis is that, for a $T = \bigsqcup_{i=1}^{n} X_i T_i$, we have $\mathrm{norm}_{\mathrm{opt}}(T_i) = \mathrm{back}_{\mathrm{opt}}(T_i)$ for all i. The induction step is to show that this implies $\mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{back}_{\mathrm{opt}}(T)$.

Let K_i be the root node of T_i. For any i such that K_i is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$, we know from Bellman's Principle of Optimality that

$$\mathrm{st}_{K_i}(\mathrm{norm}_{\mathrm{opt}}(T)) \subseteq \mathrm{norm}_{\mathrm{opt}}(T_i) = \mathrm{back}_{\mathrm{opt}}(T_i).$$

If instead K_i is not in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$, then nothing from $\mathrm{back}_{\mathrm{opt}}(T_i)$ is involved in $\mathrm{norm}_{\mathrm{opt}}(T)$. Therefore,

$$\mathrm{norm}_{\mathrm{opt}}(T) \subseteq \bigsqcup_{i=1}^{n} X_i \, \mathrm{back}_{\mathrm{opt}}(T_i) \subseteq \mathrm{nfd}(T).$$

Since $\mathrm{norm}_{\mathrm{opt}}(\mathrm{nfd}(T)) = \mathrm{norm}_{\mathrm{opt}}(T)$ and it follows from Insensitivity To Omission that

$$\mathrm{norm}_{\mathrm{opt}}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{norm}_{\mathrm{opt}}(T),$$ [1]

we can use Lemma 3.30 to conclude that

$$\begin{aligned}\mathrm{back}_{\mathrm{opt}}(T) &= \mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} X_i \, \mathrm{back}_{\mathrm{opt}}(T_i)\right) \\ &= \mathrm{norm}_{\mathrm{opt}}(T).\end{aligned}$$

□

[1] Use $\mathcal{Y} = \mathrm{opt}(\mathcal{X})$ in Insensitivity To Omission.

5.3 Subtree Perfectness

De Cooman and Troffaes [13] only investigate backward induction for these decision problems, and do not mention subtree perfectness. Given the similarities between the proofs for subtree perfectness and backward induction with decision trees, one would not be surprised to find identical relationships for deterministic system trees, and indeed this turns out to be the case. We provide the analagous result to Theorem 2.6 for deterministic system trees. We conjecture that the previous results for extensive form equivalence would work in the same way for deterministic system trees, but have not pursued this investigation.

Property 21 (Addition Property)**.** *For any gamble X and any non-empty finite set of gambles $\mathcal{Y}$,*

$$\mathrm{opt}(X + \mathcal{Y}) = X + \mathrm{opt}(\mathcal{Y}).$$

Theorem 5.5. *The normal form operator* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect for deterministic system trees if and only if* opt *satisfies Intersection and Addition.*

Lemma 5.6. *Consider a deterministic system tree $T = \bigsqcup_{i=1}^{n} X_i T_i$, and any choice function* opt*. For each tree T_i, let K_i be its root. Then, K_i is in at least one element of* $\mathrm{norm}_{\mathrm{opt}}(T)$ *if and only if*

$$(X_i + \mathrm{gamb}(T_i)) \cap \mathrm{opt}(\mathrm{gamb}(T)) \neq \emptyset. \tag{5.3}$$

Proof. Eq. (5.3) holds if and only if there is a normal form decision $U \in \mathrm{nfd}(T_i)$ such that $X_i + \mathrm{gamb}(U) \subseteq \mathrm{opt}(\mathrm{gamb}(T))$. This is equivalent to there being a U such that $\mathrm{gamb}(\sqcup X_i U) \subseteq \mathrm{opt}(\mathrm{gamb}(T))$. Clearly, $\sqcup X_i U$ is a normal form decision of T, and so by definition of $\mathrm{norm}_{\mathrm{opt}}$, Eq. (5.3) holds if and only if $\sqcup X_i U$ is in $\mathrm{norm}_{\mathrm{opt}}(T)$, which holds if and only if K_i is in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$. □

Lemma 5.7. *If $T = \bigsqcup_{i=1}^{n} X_i T_i$, and* opt *is a choice function satisfying Intersection and Addition, then*

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \bigcup_{i \in \mathcal{I}} (X_i + \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i))) \tag{5.4}$$

implies

$$\operatorname{norm}_{\operatorname{opt}}(T) = \operatorname{nfd}\left(\bigsqcup_{i\in\mathcal{I}} X_i \operatorname{norm}_{\operatorname{opt}}(T_i)\right),$$

where $\mathcal{I} = \{i \in \{1, \dots, n\}\colon (X_i + \operatorname{gamb}(T_i)) \cap \operatorname{opt}(\operatorname{gamb}(T)) \neq \emptyset\}$.

Proof. We first show that

$$\operatorname{norm}_{\operatorname{opt}}(T) \supseteq \operatorname{nfd}\left(\bigsqcup_{i\in\mathcal{I}} X_i \operatorname{norm}_{\operatorname{opt}}(T_i)\right).$$

Consider a normal form decision $U \in \operatorname{nfd}\left(\bigsqcup_{i\in\mathcal{I}} X_i \operatorname{norm}_{\operatorname{opt}}(T_i)\right)$. To show that $U \in \operatorname{norm}_{\operatorname{opt}}(T)$, we must show that $U \in \operatorname{nfd}(T)$ and $\operatorname{gamb}(U) \subseteq \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T))$. The former is obvious, and the latter is established by Eq. (5.4):

$$\begin{aligned} \operatorname{gamb}(U) &\subseteq \bigcup_{i\in\mathcal{I}} (X_i + \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T_i))) \\ &= \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T)). \end{aligned}$$

Next we show that

$$\operatorname{norm}_{\operatorname{opt}}(T) \subseteq \operatorname{nfd}\left(\bigsqcup_{i\in\mathcal{I}} X_i \operatorname{norm}_{\operatorname{opt}}(T_i)\right).$$

Let $U \in \operatorname{norm}_{\operatorname{opt}}(T)$. Let V be U with the root node removed, that is, $U = \sqcup X_k V$ for some k. Clearly, $V \in \operatorname{nfd}(T_k)$. It suffices to show that $V \in \operatorname{norm}_{\operatorname{opt}}(T_k)$. Let $\{Y\} = \operatorname{gamb}(V)$ and let $\mathcal{Y} = \operatorname{gamb}(T_k)$. We know that $X_k + Y \in \operatorname{gamb}(T)$, and $Y \in \operatorname{gamb}(T_k)$. Also, $X_k + \mathcal{Y} \subseteq \operatorname{gamb}(T)$. By Intersection and Lemma 5.6,

$$\operatorname{opt}(X_k + \mathcal{Y}) = \operatorname{opt}(\operatorname{gamb}(T)) \cap (X_k + \mathcal{Y}).$$

By Addition,

$$X_k + \operatorname{opt}(\mathcal{Y}) = \operatorname{opt}(X_k + \mathcal{Y}),$$

whence

$$X_k + \operatorname{opt}(\mathcal{Y}) = \operatorname{opt}(\operatorname{gamb}(T)) \cap (X_k + \mathcal{Y}).$$

We know $X_k + Y$ is in the right hand side, so $X_k + Y$ is in the left hand side. Therefore $Y \in \operatorname{opt}(\mathcal{Y})$ and $V \in \operatorname{norm}_{\operatorname{opt}}(T_k)$. □

Lemma 5.8. *If* $\operatorname{norm}_{\operatorname{opt}}$ *is subtree perfect then* opt *satisfies Intersection.*

Proof. Let $\mathcal{X}$ and $\mathcal{Y}$ be sets of gambles such that $\mathcal{Y} \subseteq \mathcal{X}$. Let T_1 and T_2 be deterministic system trees with exactly one decision node, and $\mathrm{gamb}(T_1) = \mathcal{X}$, $\mathrm{gamb}(T_2) = \mathcal{Y}$. Let $T = T_1 \sqcup T_2$ (so the arcs to T_1 and T_2 have reward 0), and N be the node at the root of T_2. So, $\mathrm{gamb}(T) = \mathcal{X}$. Now, $\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(\mathcal{X})$, and $\mathrm{gamb}(\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T))) = \mathrm{gamb}(\mathcal{Y}) \cap \mathrm{opt}(\mathcal{X})$. By subtree perfectness, Intersection follows. □

Lemma 5.9. *If* $\mathrm{norm}_{\mathrm{opt}}$ *is subtree perfect, then* opt *satisfies Addition.*

Proof. Let X be a gamble and let $\mathcal{Y}$ be a non-empty finite set of gambles. Let T_1 be a deterministic system tree with exactly one decision node and $\mathrm{gamb}(T_1) = \mathcal{Y}$. Let $T = \sqcup X T_1$, so $\mathrm{gamb}(T) = X + \mathcal{Y}$. Now,

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(X + \mathcal{Y})$$

and

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) = \mathrm{opt}(\mathcal{Y}).$$

By subtree perfectness and the definition of $\mathrm{norm}_{\mathrm{opt}}$, we must have that, first, any gamble $X + Y \in \mathrm{opt}(X + \mathcal{Y})$ must have $Y \in \mathrm{opt}(\mathcal{Y})$ (else there is a $U \in \mathrm{norm}_{\mathrm{opt}}(T)$ that is non-optimal in T_1), and second, any $Y \in \mathrm{opt}(\mathcal{Y})$ must have $X + Y \in \mathrm{opt}(X + \mathcal{Y})$ (else there is a $U \in \mathrm{norm}_{\mathrm{opt}}(T_1)$ with $\sqcup X U$ non-optimal in T). Therefore $\mathrm{opt}(X + \mathcal{Y}) = X + \mathrm{opt}(\mathcal{Y})$. □

Proof of Theorem 5.5. "only if". Follows from Lemmas 5.8 and 5.9.

"if". We proceed by structural induction as usual. The base step is trivial. The induction hypothesis is that, for a $T = \bigsqcup_{i=1}^{n} X_i T_i$, we have subtree perfectness at all T_i. If we can show that

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \bigcup_{i \in \mathcal{I}} (X_i + \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)))$$

for $\mathcal{I} = \{i \in \{1, \dots, n\} \colon (X_i + \mathrm{gamb}(T_i)) \cap \mathrm{opt}(\mathrm{gamb}(T)) \neq \emptyset\}$, then by Lemma 5.7 and Lemma 2.11, subtree perfectness holds for T.

We have

$$\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}\left(\bigcup_{i=1}^{n} (X_i + \mathrm{gamb}(T_i))\right)$$

whence by Very Strong Path Independence

$$= \bigcup_{i \in \mathcal{I}} \operatorname{opt}(X_i + \operatorname{gamb}(T_i))$$

whence by Addition

$$= \bigcup_{i \in \mathcal{I}} (X_i + \operatorname{opt}(\operatorname{gamb}(T_i)))$$
$$= \bigcup_{i \in \mathcal{I}} (X_i + \operatorname{gamb}(\operatorname{norm}_{\operatorname{opt}}(T_i)))$$

as required. $\square$

Chapter 6

Application To Coherent Lower Previsions

In this section we examine the behaviour of several popular choice functions for a generalization of subjective probability theory called *coherent lower previsions* (also often called *imprecise probability*, although this term also sees use for a broader set of theories). Particular attention is paid to normal form backward induction, because this has not been thoroughly investigated in the literature. We can also apply the theory for subtree perfectness, but none of the choice functions satisfies all the conditions. Instead, we can investigate special cases where subtree perfectness does hold. Similarly, locality works only in restricted cases. We do not investigate dynamic programming in detail, since most of the work has already been done by de Cooman and Troffaes [13].

6.1 Coherent Lower Previsions and Credal Sets

First, we outline a straightforward generalization of the theory of probability, allowing the subject to model uncertainty in cases where too little information is available to identify a unique probability distribution (see for instance [10, 57, 14, 71, 81, 82, 79]).

6.1.1 Credal Sets and Coherent Lower Previsions

Suppose that

- rewards are expressed in utiles, so $\mathcal{R} = \mathbb{R}$,
- the subject can express her beliefs by means of a closed convex set $\mathcal{M}$ of probability mass functions P on the power set of Ω ($\mathcal{M}$ is called the *credal set*), and
- each probability mass function $P \in \mathcal{M}$ satisfies $P(\omega) > 0$, for all $\omega \in \Omega$.
- Note that the subject does not express beliefs about the relative plausibility of the distributions, so there is no second-order probability over the set.

Under the above assumptions, each P in $\mathcal{M}$ determines a conditional expectation

$$E_P(X|A) = \frac{\sum_{\omega \in A} X(\omega)P(\omega)}{\sum_{\omega \in A} P(\omega)},$$

and the whole set $\mathcal{M}$ determines a conditional lower and upper expectation

$$\underline{P}(X|A) = \min_{P \in \mathcal{M}} E_P(X|A) \qquad \overline{P}(X|A) = \max_{P \in \mathcal{M}} E_P(X|A), \tag{6.1}$$

and this for every gamble X and every non-empty event A.

In general, the first two assumptions are all that is required to define a coherent lower prevision: the credal set can contain mass functions assigning zero probability to some events. Clearly in such cases, one cannot use (6.1) to define the conditional lower and upper expectations when A has lower probability zero. Rather than deal with the complications of the more general setting, and because all of our results fail when zero probabilities are involved, we prefer to restrict attention to credal sets containing only positive probabilities. It is important to note, however, that the theory can be extended, and indeed occasionally we see zero probability appear in a couple of examples (to illustrate how easily they can crop up). In practice, lower probability zero can be dealt with through small perturbations; see Section 6.5.1 for more details.

Why do our results not work with lower probability zero? Consider for instance the more familiar case of having a single probability mass function, and maximizing expected utility. If a chance arc has probability zero, then that arc will contribute zero to the expected utility for every normal form decision. Therefore Mixture and Backward Mixture will fail (unless the choice function conditional on an event of probability zero is assumed to be the identity, which would violate pointwise dominance). This issue is highlighted by Hammond [24, p. 44]. It turns out that all of our results for coherent lower previsions also do not hold if any events in the decision tree have probability zero for some $P \in \mathcal{M}$ (except for deterministic system trees; the problems arise from decisions following events of lower probability zero, and this never occurs in a deterministic system tree). Rather than taking the trouble to define coherent lower previsions in the more general case, only to show that the generality adds nothing for our application, it is more convenient to ignore such credal sets entirely.

The functional $\underline{P}$ is called a *coherent conditional lower prevision*, and similarly, $\overline{P}$ is called a *coherent conditional upper prevision*. Although here we have defined these by means of a set of probability measures, there are different ways of obtaining and interpreting lower and upper previsions (see for instance Miranda [52] for a survey). Indeed, the meaning of credal set or how to specify it is not clear from our definitions. A more meaningful approach is the interpretation of lower and upper previsions as buying and selling prices for gambles, generalizing De Finetti's definition of expectation as the fair price for a gamble [14].

For a particular gamble X, suppose that the subject can assess a *supremum buying price* for X, $\underline{P}(X)$. By this, the subject is stating that she has a clear preference to buy X for any price $\mu < \underline{P}(X)$ (in other words, choose the gamble $X - \mu$ over the status quo). Suppose the subject also states an *infimum selling price* $\overline{P}(X)$, which is a statement that she has a clear preference to sell X for any price $\mu > \overline{P}(X)$ (i.e. choose the gamble $\mu - X$ over the status quo). Conditional buying and selling prices can be defined in terms of called-off bets: the lower prevision $\underline{P}(X|A)$ is the supremum price the subject is disposed to pay to receive the gamble X if the bet is called off (and the subject's money returned) in the event that A does not obtain.

Clearly, not just any specifications of buying and selling prices will do. For instance, if $\underline{P}(X) > \overline{P}(X)$, then the subject is apparently disposed to buy a gamble and then immediately sell it for less. This behaviour is called incurring sure loss. Also, suppose the subject specifies $\underline{P}(X) = 1$, $\underline{P}(Y) = 1$, $\underline{P}(X + Y) = 1$. She is disposed to pay up to 1 for X and up to 1 for Y, and so should be disposed to pay up to 2 for $X + Y$, conflicting with $\underline{P}(X + Y) = 1$. This behaviour is called incoherence. We assume all specifications a subject makes both avoid sure loss and are coherent. Further, we assume all events of interest have strictly positive lower prevision (for the reasons outlined above).

It is often unreasonable to expect the subject to make such specifications for all gambles, or even all gambles of interest. A limited number of specifications that avoid sure loss is, however, enough to infer lower and upper previsions for any gamble, by a process called *natural extension* [79, §3.1]. This process also shows the link between buying and selling prices and the credal set. If $\mathcal{A}$ is the set of all relevant conditioning events, and $\mathcal{X}_A$ is the set of all gambles X for which the subject has specified $\underline{P}(X|A)$, then we can define a credal set via

$$\mathcal{M} = \{P : (\forall A \in \mathcal{A}, X \in \mathcal{X}_A)(E_P(X|A) \geq \underline{P}(X|A))\}.$$

Then, as shown by Williams [81], if the original specifications were coherent, the equalities in (6.1) are satisfied for all relevant X and A. Further, these equalities can be used to find upper and lower previsions for any other gamble, and this is guaranteed to be coherent. Note also that, if the original specifications were incoherent (but avoided sure loss), then this method can be used to automatically correct the incoherence. For instance, in the example of incoherence in the previous paragraph, natural extension would correct the lower prevision for $X + Y$ to 2.

At this point, a simple example of calculating upper and lower probability for events, including conditioning, may be useful.

Example 6.1. *Suppose that A, B, C are logically independent events with $\underline{P}(A) = 0.4$, $\overline{P}(A) = 0.7$, $\underline{P}(B) = 0.5$, $\overline{P}(B) = 0.8$, $\underline{P}(C) = 0.51$, $\overline{P}(C) = 0.7$. What is $\underline{P}(A \cup B)$, $\overline{P}(A \cup B)$, $\underline{P}(C|A \cup B)$ and $\overline{P}(C|A \cup B)$?*

Let $D = A \cup B$. Observe that we can find some $P \in \mathcal{M}$ such that $P(A) = 0.4$,

$P(B) = 0.5$, and $P(A \cap B) = 0.4$. In this case, $P(D) = 0.5$. This must be the lowest probability for D in $\mathcal{M}$. Therefore we have $\underline{P}(D) = 0.5$. Also, observe we can find some $P \in \mathcal{M}$ with $P(A) = 0.7$, $P(B) = 0.8$, and $P(A \cap B) = 0.5$, so $P(D) = 1$. Therefore $\overline{P}(D) = 1$.

Consider next $\overline{P}(C|D)$. As just shown, there are $P \in \mathcal{M}$ such that $P(D) = 0.7$. We can find one of these that has $P(C) = 0.7$ and $P(C \cap D) = 0.7$, so that $P(C|D) = 1$. Clearly $\overline{P}(C|D) = 1$.

Finally, consider $\underline{P}(C|D)$. If we have a $P \in \mathcal{M}$ with $P(D) = p$, then we would want to minimize $P(C \cap D)$. To do this, we should take $P(C) = 0.51$. We should also assign as much probability as possible to $C \cap \overline{D}$: this is $1 - P(D)$. Then $P(C \cap D) = P(C) - (1 - P(D)) = P(D) - 0.49$. So,

$$P(C|D) = \frac{P(D) - 0.49}{P(D)}.$$

This is minimized by $P(D) = 0.5$ giving $P(C|D) = 0.02$. So $\underline{P}(C|D) = 0.02$. Observe how the conditional for C given D is almost vacuous. This is because the initial specifications provided very little information about the unions and intersections of events.

Observe that, since $\overline{P}(D) = 1$, $\underline{P}(\overline{D}) = 0$. Thus, there are events of lower probability zero in this example! In practice this only causes problems for our results if one of these events appears on a chance arc.

Below are some properties of coherent conditional lower and upper previsions that we require later (see Williams [81, 82] or Walley [79] for proofs). Following De Finetti [14], where it is convenient we denote the indicator gamble $I_A = 1A \oplus 0\overline{A}$ simply by A as well, so we write for instance $\underline{P}(A|B)$ for $\underline{P}(I_A|B)$. We also drop the $\oplus$ notation, simply using $+$.

Proposition 6.2. *For all non-empty events A, B, gambles X, Y, and constants $\lambda > 0$:*

(i) *If $AX = AY$ then $\underline{P}(X|A) = \underline{P}(Y|A)$.*

(ii) $\underline{P}(X|A) + \underline{P}(Y|A) \leq \underline{P}(X + Y|A) \leq \underline{P}(X|A) + \overline{P}(Y|A)$.

(iii) $\underline{P}(\lambda X|A) = \lambda\underline{P}(X|A)$ *and* $\overline{P}(\lambda X|A) = \lambda\overline{P}(X|A)$.

(iv) $\underline{P}(X|A) = -\overline{P}(-X|A)$.

(v) $\underline{P}(A(X - \underline{P}(X|A \cap B))|B) = 0$.

By property (iv), we see that a coherent lower prevision over every gamble completely determines the upper previsions. Thus, in what follows we need only assume the existence of the lower prevision. Property (v) is a generalization of the *generalized Bayes rule* [79, p. §6.4], and shows how conditional lower previsions are related.

If a lower prevision is self-conjugate, so that $\underline{P}(X|A) = \overline{P}(X|A)$ for all X and A, then this coincides with De Finetti's concept of previsions [14]. We call these *linear previsions*. Since upper and lower previsions are always the same for linear previsions, we can simply call them P.

Finally, recall the law of iterated expectation for classical probability. The theorem does not in general hold for lower previsions, becoming an inequality instead. We introduce some notation to express this conveniently. If $\mathcal{A} = \{A_1, \ldots, A_n\}$, then define $\underline{P}(X|\mathcal{A})$ to be the gamble

$$\underline{P}(X|\mathcal{A}) = \sum_{i=1}^{n} A_i\underline{P}(X|A_i).$$

Then for a coherent lower prevision, the following inequalities hold:

$$\underline{P}(X|B) \geq \underline{P}(\underline{P}(X|\mathcal{A} \cap B)|B) \qquad \overline{P}(X|B) \leq \underline{P}(\underline{P}(X|\mathcal{A} \cap B)|B).$$

It is possible to have equality for some partitions. If there is equality for a particular partition, then $\underline{P}$ is said to satisfy *marginal extension* [79, §6.7.2] for this partition. It turns out that satisfying marginal extension over all partitions in a decision tree often yields better-behaved normal form solutions.

Definition 6.3. *Let $\underline{P}$ be a coherent lower prevision, X be a gamble, B be an event, and $\mathcal{A} = \{A_1, \ldots, A_n\}$ be a partition of Ω. $\underline{P}$* satisfies marginal extension *for X with respect to $\mathcal{A}$ and B if*

$$\underline{P}(X|B) = \underline{P}(\underline{P}(X|\mathcal{A} \cap B)|B) = \underline{P}\left(\sum_{i=1}^{n} A_i\underline{P}(X|A_i \cap B)\middle| B\right).$$

6.1.2 Choice Functions and Optimality

We now consider four popular choice functions that have been proposed for choosing between gambles given a coherent lower prevision. Further discussion of the criteria presented here can be found in Troffaes [77].

Maximality

Maximality is based on the following strict partial preference order $>_{\underline{P}|A}$.

Definition 6.4. *Given a coherent lower prevision $\underline{P}$, for any two gambles X and Y we write $X >_{\underline{P}|A} Y$ whenever $\underline{P}(X - Y|A) > 0$.*

If $\underline{P}(X - Y|A) > 0$, then the subject would be willing to pay to swap Y for X. If forced to choose between X and Y, it is then reasonable to prefer X to Y. This gives rise to the choice function *maximality*, proposed by Condorcet [12, pp. lvj–lxix, 4.^{e} Exemple], Sen [66], and Walley [79], among others.[1]

Definition 6.5. *For any non-empty finite set of gambles $\mathcal{X}$ and each event $A \neq \emptyset$,*

$$\mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(Y \not>_{\underline{P}|A} X)\}.$$

Interpreting maximality from a credal set perspective, a gamble X is optimal in $\mathcal{X}$ if, for each $Y \in \mathcal{X}$, there is a $P_Y \in \mathcal{M}$ such that $P_Y(X|A) \geq P_Y(Y|A)$. Observe that the linear prevision used can change with Y, so that there may be no $P \in \mathcal{M}$ for which X maximizes expected utility. This suggests the following refinement.

E-admissibility

Another criterion is E-admissibility, proposed by Levi [44]. Recall that $\underline{P}(\cdot|A)$ is the lower envelope of $\mathcal{M}$. For each $P \in \mathcal{M}$ we can maximize expected utility:

$$\mathrm{opt}_P(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(E_P(Y|A) \leq E_P(X|A))\}.$$

Then the set of E-admissible options is defined by:

[1] Because all probabilities in $\mathcal{M}$ are assumed to be strictly positive, Walley's admissibility condition is implied and hence omitted in Definition 6.5.

Definition 6.6. *For any non-empty finite set of gambles $\mathcal{X}$ and each event $A \neq \emptyset$,*

$$\mathrm{opt}_{\mathcal{M}}(\mathcal{X}|A) = \bigcup_{P \in \mathcal{M}} \mathrm{opt}_P(\mathcal{X}|A).$$

A gamble X is therefore E-admissible when it maximizes expected utility under at least one $P \in \mathcal{M}$. Any E-admissible gamble is maximal [79, p. 162, ll. 26–28].

Interval Dominance

Interval dominance is based on the strict partial preference order $\sqsupset_{\underline{P}|A}$.

Definition 6.7. *Given a coherent lower prevision $\underline{P}$, for any non-empty event A and any two A-consistent gambles X and Y we write $X \sqsupset_{\underline{P}|A} Y$ whenever $\underline{P}(X|A) > \overline{P}(Y|A)$.*

This ordering induces a choice function usually called *interval dominance* [84, 77]:

Definition 6.8. *For any non-empty finite set of gambles $\mathcal{X}$ and each event $A \neq \emptyset$,*

$$\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(Y \not\sqsupset_{\underline{P}|A} X)\}.$$

The above criterion was apparently first introduced by Kyburg [40] and was originally called stochastic dominance. The motivation is quite simple. If the upper prevision of X is lower than the lower prevision of Y, then if X and Y are among our options we should never choose X. The motivation for this can be seen from the behavioural interpretations. Suppose one chooses X. Then one would sell X for $\overline{P}(X) + \epsilon$ if offered. One would then also buy Y for $\underline{P}(Y) - \epsilon$, thus one would have $\overline{P}(X) - \underline{P}(Y) + Y < Y$. It would have been more sensible to have chosen Y in the first place.

It follows from this argument that any choice function for coherent lower previsions should give a subset of interval dominance, but not that interval dominance is the best choice or even a vaguely sensible one. For instance, maximality is a subset of interval dominance, and the extra gambles it removes seem sensible to omit. The credal set interpretation of interval dominance clearly demonstrates its shortcomings. A gamble X is optimal if there is a $P \in \mathcal{P}$ such that, for each gamble Y there is a $Q_Y \in \mathcal{P}$ such that $P(X) \geq Q_Y(Y)$. Note that the Q_Y can be different for different Y. This is hardly compelling reason to call X optimal. After all, this means that interval dominance can rarely choose between X and $X - \epsilon$. Still, interval dominance is easier to calculate than maximality or E-admissibility, and so may be attractive in large problems.

Γ-maximin

Γ-maximin selects gambles that maximizes the minimum expected reward.

Definition 6.9. *For any non-empty finite set of gambles $\mathcal{X}$ and each event $A \neq \emptyset$,*

$$\mathrm{opt}_{\underline{\mathrm{P}}}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(\underline{P}(X|A) \geq \underline{P}(Y|A))\}.$$

Γ-maximin is a total preorder, and so usually selects a single gamble regardless of the degree of uncertainty in $\underline{P}$. Γ-maximin can be criticized for being too conservative (see Walley [79, p. 164]), as it only takes into account the worst possible scenario. That this is a risk-averse choice can also be seen from a credal set interpretation: whatever gamble X we choose, assume that the "true" $P \in \mathcal{P}$ is the worst case one, that is, the one that minimizes $P(X)$. Under this assumption, we choose the gamble for which this minimum is greatest. Γ-maximin has roots in *robust Bayesian statistics*, and is discussed by Berger [9, § 4.7.6]. Another criticism of Γ-maximin is that it fails to take into account the possibility of indecision even when the intervals between upper and lower previsions are large. Of course, being able to always choose one optimal decision can also be seen as a desirable propery.

Two criteria related to Γ-maximin are Γ-maximax (maximizing upper expectation), and Hurwicz (maximizing a combination of upper and lower previsions). These properties of these two choice functions tend to be similar to or worse than Γ-maximin for subtree perfectness and backward induction, and they do not appear so commonly in the literature of coherent lower previsions, so we shall not consider them in detail.

Relationships between the choice functions

The relationships between the choice functions are shown in Fig. 6.1, from [77]. So, for instance, a gamble that is Γ-maximin is always maximal but may not be

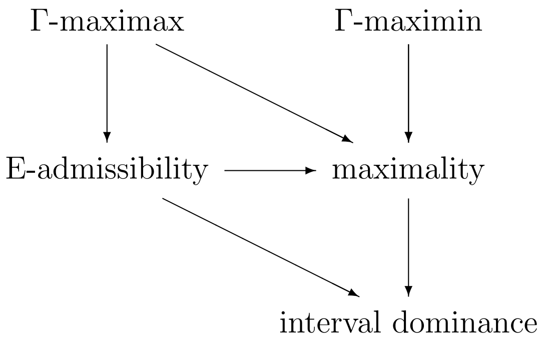

Figure 6.1: Relationships between the choice functions

	ω_1	ω_2
X	0	1
Y	1	0
Z	0.4	0.4

Table 6.1: Gambles for Example 6.10

E-admissible.

The following example (as seen for instance in Seidenfeld [64]) is instructive to see how the differences between E-admissibility, maximality, and Γ-maximin arise.

Example 6.10. *Let $\Omega = \{\omega_1, \omega_2\}$, and let X, Y, and Z be the gambles in Table 6.1. Suppose that we are completely ignorant about which of ω_1, ω_2 is true, and so assign the vacuous lower prevision, $\underline{P}(\omega_1) = \underline{P}(\omega_2) = 0$. Then, $\mathrm{opt}_{\underline{\mathrm{P}}}(\{X, Y, Z\}) = \{Z\}$ since its lower prevision is 0.4 rather than 0. But consider a $P \in \mathcal{M}$. If $P(\omega_1) \geq 0.5$, then $P(Y) \geq 0.5 > P(Z)$. And if $P(\omega_1) \leq 0.5$, then $P(X) \geq 0.5 > P(Z)$. So there is no $P \in \mathcal{M}$ such that Z maximizes expected utility, and so $Z \notin \mathrm{opt}_{\mathcal{M}}(\{X, Y, Z\}) = \{X, Z\}$. But Z is maximal, since there is a $P \in \mathcal{M}$ such that $P(\omega_1) \leq 0.4$ and a $P \in \mathcal{M}$ such that $P(\omega_2) \leq 0.4$.*

These differences can be easily understood by considering the randomized gamble that chooses X with probability $1/2$ and Y with probability $1/2$. For any $P \in \mathcal{M}$, this randomized gamble has expected value 0.5, and so dominates Z everywhere in $\mathcal{M}$. That is, $\underline{P}(1/2X + 1/2Y - Z) = 0.1 > 0$, so in the set of all randomized gambles of $\{X, Y, Z\}$, Z is not maximal, and is therefore not E-admissible in $\{X, Y, Z\}$.

6.2 Backward Induction Properties for Coherent Lower Previsions

In this section we formally investigate which of the choice functions for coherent lower previsions satisfy the conditions of Theorem 3.12(B). It turns out that only maximality and E-admissibility do, although interval dominance does not behave too badly. Some of the proofs are based on the following results for more general choice functions.

6.2.1 Results for General Choice Functions

This section details intermediate results required for the proofs in Section 6.1. Since the results could be applicable for choice functions that are nothing to do with coherent lower previsions, and so may be useful for investigating other uncertainty models, we present them separately.

Proposition 6.11. *For each non-empty event A, let $\succ_A$ be any strict partial order on A-consistent gambles. The choice function induced by these strict partial orders, that is,*

$$\operatorname{opt}_{\succ|\cdot}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(Y \not\succ_A X)\}$$

satisfies Insensitivity To Omission and Preservation Under Addition.

Proof. By Lemma 3.4, it suffices to show that $\operatorname{opt}_{\succ|\cdot}$ is path independent. Let $\mathcal{X}_1$, …, $\mathcal{X}_n$ be non-empty finite sets of A-consistent gambles, and let A be a non-empty event. Let $\mathcal{X} = \bigcup_{i=1}^n \mathcal{X}_i$ and $\mathcal{Z} = \bigcup_{i=1}^n \operatorname{opt}_{\succ|\cdot}(\mathcal{X}_i|A)$. We show must show that

$$\operatorname{opt}_{\succ|\cdot}(\mathcal{X}|A) = \operatorname{opt}_{\succ|\cdot}(\mathcal{Z}|A). \tag{6.2}$$

By definition,

$$\operatorname{opt}_{\succ|\cdot}(\mathcal{Z}|A) = \{Z \in \mathcal{Z} \colon (\forall Y \in \mathcal{Z})(Y \not\succ_A Z)\},$$

and, observe that, if $X \in \mathcal{X}$ but $X \notin \mathcal{Z}$, by transitivity of $\succ_A$ and finiteness of $\mathcal{X}$, there is a $Y \in \mathcal{Z}$ such that $Y \succ_A X$. Therefore again by transitivity of $\succ_A$, for any $Z \in \mathcal{Z}$ such that $X \succ_A Z$, we have $Y \succ_A Z$. So,

$$= \{Z \in \mathcal{Z} \colon (\forall Y \in \mathcal{X})(Y \not\succ_A Z)\},$$

and once again, by definition of $\mathcal{Z}$, if $X \in \mathcal{X}$ but $X \notin \mathcal{Z}$ there is a $Y \in \mathcal{X}$ such that $Y \succ_A X$, so we have

$$= \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(Y \not\succ_A X)\}$$
$$= \mathrm{opt}_{\succ|\cdot}(\mathcal{X}|A).$$

□

Proposition 6.12. *Let* $\{\mathrm{opt}_i \colon i \in \mathcal{I}\}$ *be a family of choice functions. For any non-empty event* A *and any non-empty finite set of* A*-consistent gambles* $\mathcal{X}$*, let*

$$\mathrm{opt}(\mathcal{X}|A) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A).$$

(i) If each opt_i *satisfies Insensitivity To Omission, then so does* opt*.*

(ii) If each opt_i *satisfies Preservation Under Addition, then so does* opt*.*

(iii) If each opt_i *satisfies Backward Mixture, then so does* opt*.*

(iv) If each opt_i *satisfies Backward Conditioning, Preservation Under Addition, and Backward Mixture, then* opt *satisfies Backward Conditioning.*

(v) If each opt_i *satisfies Mixture then so does* opt*.*

(vi) If each opt_i *satisfies Preservation Under Addition, Mixture, and Backward Conditioning, then* opt *satisfies Conditioning.*

Proof. (i). By definition of opt and by assumption, for any finite non-empty sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$ and for any $i \in \mathcal{I}$, $\mathrm{opt}_i(\mathcal{X}|A) \subseteq \mathrm{opt}(\mathcal{X}|A) \subseteq \mathcal{Y}$, and therefore by Insensitivity To Omission, $\mathrm{opt}_i(\mathcal{X}|A) = \mathrm{opt}_i(\mathcal{Y}|A)$. Whence,

$$\mathrm{opt}(\mathcal{Y}|A) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{Y}|A) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A) = \mathrm{opt}(\mathcal{X}|A).$$

(ii). By assumption, for any finite non-empty sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$ and for any $i \in \mathcal{I}$, $\mathrm{opt}_i(\mathcal{Y}|A) \supseteq \mathrm{opt}_i(\mathcal{X}|A) \cap \mathcal{Y}$. Therefore,

$$\mathrm{opt}(\mathcal{Y}|A) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{Y}|A) \supseteq \bigcup_{i \in \mathcal{I}} (\mathrm{opt}_i(\mathcal{X}|A) \cap \mathcal{Y})$$
$$= \mathcal{Y} \cap \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A) = \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

(iii). By assumption, for any non-empty finite set of gambles $\mathcal{X}$, any gamble Z, any events A and B such that $A \cap B \neq \emptyset$, and for any $i \in \mathcal{I}$,

$$\mathrm{opt}_i(A\mathcal{X} \oplus \overline{A}Z|B) \subseteq A\,\mathrm{opt}_i(\mathcal{X}|A \cap B) \oplus \overline{A}Z,$$

whence

$$\begin{aligned}
\mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B) &= \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(A\mathcal{X} \oplus \overline{A}Z|B) \\
&\subseteq \bigcup_{i \in \mathcal{I}} (A\,\mathrm{opt}_i(\mathcal{X}|A \cap B) \oplus \overline{A}Z) \\
&= \overline{A}Z \oplus A \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A \cap B) \\
&= \overline{A}Z \oplus A\,\mathrm{opt}(\mathcal{X}|B).
\end{aligned}$$

(iv). Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, Z be a $\overline{A} \cap B$-consistent, and $\mathcal{X}$ be a non-empty finite set of $A \cap B$-consistent gambles such that there is $\{X, Y\} \subseteq \mathcal{X}$ with $AX = AY$, and $AX + \overline{A}Z \in \mathrm{opt}(A\mathcal{X} + \overline{A}Z|B)$. If this situation does not exist then opt satisfies Backward Addition automatically.

By definition of opt, there is a j such that $AX + \overline{A}Z \in \mathrm{opt}_j(A\mathcal{X} + \overline{A}Z|B)$. We show that both X and Y are in $\mathrm{opt}_j(\mathcal{X}|A \cap B)$, and therefore are both in $\mathrm{opt}(\mathcal{X}|A \cap B)$. It follows from Preservation Under Addition and Backward Mixture that

$$\mathrm{opt}_j(A\mathcal{X} + \overline{A}Z|B) \subseteq A\,\mathrm{opt}_j(\mathcal{X}|A \cap B) + \overline{A}Z.$$

Therefore, there is a $W \in \mathrm{opt}_j(\mathcal{X}|A \cap B)$ with $AW = AX$. Finally, opt_j satisfies Backward Conditioning, and therefore both X and Y must be in $\mathrm{opt}_j(\mathcal{X}|A \cap B)$. This establishes Backward Conditioning for opt.

(v). By assumption, for any non-empty finite set of gambles $\mathcal{X}$, any gamble Z, any events A and B such that $A \cap B \neq \emptyset$, and for any $i \in \mathcal{I}$,

$$\mathrm{opt}_i(A\mathcal{X} \oplus \overline{A}Z|B) = A\,\mathrm{opt}_i(\mathcal{X}|A \cap B) \oplus \overline{A}Z,$$

whence

$$\begin{aligned}\mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B) &= \bigcup_{i\in\mathcal{I}} \mathrm{opt}_i(A\mathcal{X} \oplus \overline{A}Z|B)\\ &= \bigcup_{i\in\mathcal{I}} (A\,\mathrm{opt}_i(\mathcal{X}|A\cap B) \oplus \overline{A}Z)\\ &= \overline{A}Z \oplus A\bigcup_{i\in\mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A\cap B)\\ &= \overline{A}Z \oplus A\,\mathrm{opt}(\mathcal{X}|B).\end{aligned}$$

(vi). By the above results, opt satisifes Preservation Under Addition, Mixture, and Backward Conditioning. By Lemma 3.6, opt satisfies Strong Backward Conditioning. By Lemma 3.7, opt satisfies Conditioning. □

6.2.2 Maximality

Maximality is a strict partial order, so Insensitivity To Omission and Preservation Under Addition hold, by Proposition 6.11.

Proposition 6.13. *Maximality satisfies Conditioning.*

Proof. We show that maximality satisfies Conditioning. Let A be a non-empty event, $\mathcal{X}$ be a non-empty finite set of A-consistent gambles, and $\{X,Y\} \subseteq \mathcal{X}$ with $AX = AY$. We show that, for any event B such that $A\cap B \neq \emptyset$, $X \in \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A\cap B)$ implies $Y \in \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A\cap B)$.

If $X \in \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A\cap B)$, then for every $Z \in \mathcal{X}$, $\underline{P}(Z - X|A\cap B) \leq 0$. But $AX = AY$ implies $(A\cap B)X = (A\cap B)Y$, and, by Proposition 6.2(i), $\underline{P}(Z-X|A\cap B) = \underline{P}(Z-Y|A\cap B)$, and so it immediately follows that $Y \in \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A\cap B)$. □

Proposition 6.14. *Maximality satisfies Backward Mixture.*

Proof. Consider events A and B such that $A\cap B \neq \emptyset$ and $\overline{A}\cap B \neq \emptyset$, a non-empty finite set of $A\cap B$-consistent gambles $\mathcal{X}$, and an $\overline{A}\cap B$-consistent gamble Z. To establish Backward Mixture, it suffices to show that for any $Y \in \mathcal{X}$,

$$Y \notin \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A\cap B) \implies AY + \overline{A}Z \notin \mathrm{opt}_{>_{\underline{P}|\cdot}}(A\mathcal{X} \oplus \overline{A}Z|B).$$

If $Y \notin \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A \cap B)$ then there is an $X \in \mathcal{X}$ with $\underline{P}(X-Y|A \cap B) > 0$. The result follows if we show that $\underline{P}(AX+\overline{A}Z-(AY+\overline{A}Z)|B) > 0$. By Proposition 6.2(ii)–(v),

$$\begin{aligned} 0 &= \underline{P}(A(X-Y-\underline{P}(X-Y|A \cap B))|B) \\ &\le \underline{P}(A(X-Y)|B) + \overline{P}(-A\underline{P}(X-Y|A \cap B)|B) \\ &= \underline{P}(A(X-Y)|B) - \underline{P}(A\underline{P}(X-Y|A \cap B)|B) \\ &= \underline{P}(A(X-Y)|B) - \underline{P}(A|B)\underline{P}(X-Y|A \cap B) \end{aligned}$$

where we relied on $\underline{P}(X-Y|A \cap B) > 0$ in the last step. So, indeed

$$\underline{P}(A(X-Y)|B) \ge \underline{P}(A|B)\underline{P}(X-Y|A \cap B) > 0.$$

□

Corollary 6.15. *For any consistent decision tree T, it holds that*

$$\mathrm{back}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T) = \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T).$$

Proof. Immediate, from Propositions 6.11, 6.13, and 6.14, and Theorem 3.12. □

6.2.3 E-admissibility

Since E-admissibility is a union of maximality choice functions we have:

Corollary 6.16. *For any consistent decision tree T, it holds that*

$$\mathrm{back}_{\mathrm{opt}_{\mathcal{M}}}(T) = \mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}(T).$$

Proof. Immediate, from Proposition 6.12, Corollary 6.15, and Theorem 3.12. □

Further, from Theorem 3.28, we have:

Corollary 6.17. *For any consistent decision tree T,*

$$\mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}(T) = \mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}(\mathrm{back}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T)).$$

	A	$\overline{A}$
X	1	1
Y	1.5	3.5
Z	0	4

	$\underline{P}(\cdot\|B)$	$\overline{P}(\cdot\|B)$
X	1	1
Y	2	3
Z	1	3

	$\underline{P}$	$\overline{P}$
$BX+\overline{B}Z$	1	2
$BY+\overline{B}Z$	1.5	3

Table 6.2: Gambles and their lower and upper previsions for Example 6.18.

6.2.4 Interval Dominance

By Proposition 6.11, interval dominance satisfies Insensitivity To Omission and Preservation Under Addition, and it satisfies Backward Conditioning because $AX = AY$ implies $\underline{P}(X|A) = \underline{P}(Y|A)$ and $\overline{P}(X|A) = \overline{P}(Y|A)$. We now show that interval dominance fails Backward Mixture.

Example 6.18. *Suppose A and B are events, and X, Y, and Z are the gambles given in Table 6.2. Let $\mathcal{M}$ contain all mass functions P such that A and B are independent, $1/4 \le P(A) \le 3/4$, and $P(B) = 1/2$. Let $\underline{P}$ be the lower envelope of $\mathcal{M}$.*

Lower and upper previsions of relevant gambles are given in Table 6.2; for example,

$$\begin{aligned}\overline{P}(BY+\overline{B}Z) &= \max_{P\in\mathcal{M}} P(BY+\overline{B}Z) = \max_{P\in\mathcal{M}} P(B)P(Y|B) + P(\overline{B})P(Z|\overline{B}) \\ &= \frac{1}{2}\max_{P\in\mathcal{M}}(P(Y)+P(Z)) = \frac{1}{2}\max_{p\in[\frac{1}{4},\frac{3}{4}]}(1.5(1-p)+3.5p+4p) = 3\end{aligned}$$

and similar for all other gambles. Clearly, Y interval dominates X conditional on B, however, $BY+\overline{B}Z$ does not interval dominate $BX+\overline{B}Z$, violating Backward Mixture.

Even though interval dominance violates Backward Mixture, it can still be of use in backward induction. It is easily shown that (see for instance Troffaes [77])

$$\mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}|A) \subseteq \mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}(\mathcal{X}|A).$$

By Theorem 3.28, we therefore have:

Corollary 6.19. *For any consistent decision tree T,*

$$\mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T) = \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T))$$
$$\mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}(T) = \mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}(\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T))$$

It can also be shown that $\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T) \subseteq \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$ for all T, so all strategies found by backward induction will be optimal with respect to $\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}$.

Lemma 6.20. *For any non-empty event A and any coherent lower prevision $\underline{P}$, if $Z >_{\underline{P}|A} Y$ and $Z \not\sqsupset_{\underline{P}|A} X$, then $Y \not\sqsupset_{\underline{P}|A} X$.*

Proof. Let $\underline{P}(Y|A) + \epsilon = \overline{P}(X|A)$. Then

$$\underline{P}(Y|A) + \epsilon \geq \underline{P}(Z|A).$$

So,

$$\begin{aligned}0 &\leq \underline{P}(Y|A) - \underline{P}(Z|A) + \epsilon \\ &= \underline{P}(Y|A) + \overline{P}(-Z|A) + \epsilon \\ &\leq \overline{P}(Y - Z|A) + \epsilon \\ &\leq \epsilon.\end{aligned}$$

Therefore, $\epsilon > 0$ and $Y \not\sqsupset_{\underline{P}|A} X$. □

Theorem 6.21. *For any lower prevision $\underline{P}$ and any consistent decision tree T,*

$$\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T) \subseteq \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$$

Proof. We must show that there are no elements of $\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$ that are not in $\mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$. Since $\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}$ corresponds to a partial order, this can only happen if there is a $V \in \mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$ and a $U \in \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T) \setminus \mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$ such that $\mathrm{gamb}(U) \sqsupset_{\underline{P}|\mathrm{ev}(T)} \mathrm{gamb}(V)$.

For any $U \in \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T) \setminus \mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$, there is an N in U such that $\mathrm{st}_N(U) \notin \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(\mathrm{st}_N(T))$ (for instance, let N be a node at the first level of T where $\mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}$ and $\mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}$ differ; note that this cannot be the root of T). Because interval dominance contains all maximal gambles, $\mathrm{st}_N(U) \notin \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(\mathrm{st}_N(T))$.

By Theorem 3.12, $U \notin \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T)$. So, there is at least one $W \in \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}(T)$ such that $\mathrm{gamb}(W) >_{\underline{P}|\mathrm{ev}(T)} \mathrm{gamb}(U)$. By Theorem 3.28, $W \in \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$.

For any $V \in \mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$, we know

$$\mathrm{gamb}(W) \not\sqsupset_{\underline{P}|\mathrm{ev}(T)} \mathrm{gamb}(V)$$

. By Lemma 6.20, $\mathrm{gamb}(U) \not\sqsupset_{\underline{P}|\mathrm{ev}(T)} \mathrm{gamb}(V)$ for any $U \in \mathrm{norm}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T) \setminus \mathrm{back}_{\mathrm{opt}_{\sqsupset_{\underline{P}|\cdot}}}(T)$. □

6.2.5 Γ-maximin

It has been shown that Γ-maximin fails Theorem 3.12(A). To demonstrate this, we give a variation on a counter-example by Seidenfeld [64, Sequential Example 1, pp. 75–77].

Example 6.22. *Consider two coins. One is known to be fair (with probability* 1/2 *of landing heads, and* 1/2 *of landing tails). Nothing is known about the other, so it has lower probability* 0 *of landing heads, and lower probability* 0 *of landing tails (we use* 0 *here for simplicity: although this violates the assumption that lower probabilities are positive, we can replace* 0 *with some small* $\epsilon > 0$ *and nothing would change). It is known that the result of tossing one coin does not influence the other. Consider the following gamble,* X*: the subject receives* 1 *if both coins land heads,* 1 *if both coins land tails, and* 0 *otherwise. It turns out there is no imprecision in this gamble: there is a* 1/2 *chance that the fair coin agrees with the mysterious coin, so* $\underline{P}(X) = \overline{P}(X) = 0.5$*,*

Now suppose the subject is offered the sequential problem in Figure 6.2. She has two choices initially: to pay 0.4 *for the randomization* X*, or to be given* 0.05 *to observe the fair coin. After observing the fair coin, she again must decide whether or not to buy* X *for* 0.4*. Of course, at this point* X *is no longer a randomization because the fair coin's result is known. Having observed* H*,* X *is equivalent to a gamble giving* 1 *if the mysterious coin lands heads and* 0 *otherwise, and similarly for observing* T*. Since nothing is known about the mysterious coin, these two gambles have lower prevision* 0*.*

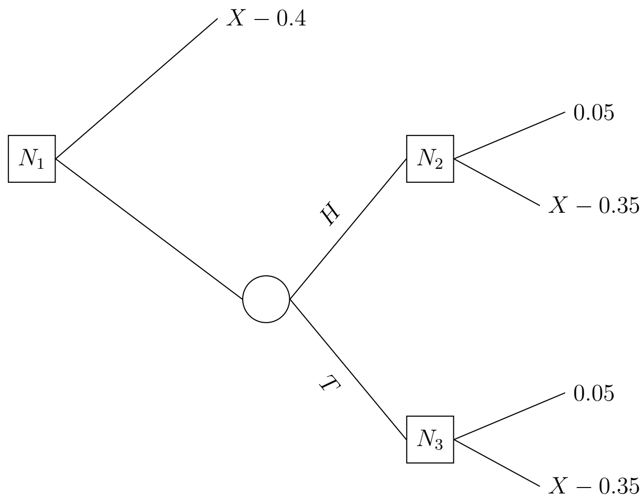

Figure 6.2: An example for backward induction using Γ-maximin

So, applying Γ*-maximin at* N_2 *and* N_3 *will eliminate the* $X - 0.35$ *branches. Backward induction therefore chooses the normal form gamble* $X - 0.4$*, since this has higher prevision than* $1/2 \cdot 0.05 + 1/2 \cdot 0.5$*. But observe that there is a normal form gamble* $H(X - 0.35) \oplus T(X - 0.35) = X - 0.35$*. Therefore* $X - 0.4$ *cannot be an optimal normal form gamble.*

Since Γ-maximin is induced by an ordering, it satisfies Insensitivity To Omission and Preservation Under Addition by Proposition 6.11. As for interval dominance, Γ-maximin satisfies Backward Conditioning. Hence, Γ-maximin must fail Backward Mixture. Indeed, backward induction can fail in a particularly serious way: it can select a single gamble that is inferior to another normal form gamble. Hence, backward induction may not find any Γ-maximin gambles.

Later we see that if $\underline{P}$ satisfies marginal extension for all partitions in a decision tree, $\mathrm{norm}_{\mathrm{opt}_{\underline{P}}}$ is subtree perfectness, and therefore in such situations backward induction does work.

6.3 Backward Induction Examples

In this section we give two examples of the use of $\mathrm{back}_{\mathrm{opt}}$ to solves simple decision trees.

6.3.1 Lake District Problem

Consider again the lake district problem depicted in Fig. 1.1, but now suppose that the subject has specified a coherent lower prevision, instead of a singe probability measure. For this example, we consider an ϵ-contamination model: with probability $1-\epsilon$, observations follow a given probability mass function P, and with probability ϵ, observations follow an unknown arbitrary distribution. One can easily check that, under this model, the lower expectation for a gamble X is

$$\underline{P}(X) = (1-\epsilon)E_P(X) + \epsilon \inf X$$

The conditional lower expectation is [79, p. 309]

$$\underline{P}(X|A) = \frac{(1-\epsilon)E_P(AX) + \epsilon \inf_{\omega \in A} X(\omega)}{(1-\epsilon)P(A) + \epsilon}$$

As before, let $P(S_1) = 0.6$, $P(E_1|S_1) = 0.7$, and $P(E_1|S_2) = 0.2$, so $P(E_1) = 0.5$. Let $\epsilon = 0.1$.

Naively, she could solve the problem using $\mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}$: she lists all possible strategies, finds the corresponding gambles, and applies maximality.

Table 6.3 lists all strategies and their gambles. Each strategy gives a reward determined entirely by ω, and hence has a corresponding gamble. For example, the gamble for the last strategy is

$$(5-c)S_1E_1 + (20-c)S_1E_2 + (10-c)S_2E_1 + (15-c)S_2E_2 = S_1Y + S_2X - c,$$

with $X = 10E_1 + 15E_2$ and $Y = 5E_1 + 20E_2$. Here, and in the following, we denote the indicator function $I_A(\omega) = \left\{\begin{smallmatrix} 0 \text{ if } \omega \in A \\ 1 \text{ if } \omega \notin A \end{smallmatrix}\right.$ of an event A also simply by A: for instance, $(5-c)S_1E_1$ is just a shorthand notation for $(5-c)I_{S_1}I_{E_1}$.

Maximality can then be applied to find the optimal gambles: this requires comparison of all six gambles at once. Skipping the details of this calculation, for instance with $c = 0.5$, we find that we should buy the newspaper and follow its advice.

More efficient is to use $\mathrm{back}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}$, as illustrated in Fig. 6.3.

Denote subtrees at a particular node N_*^* by $T_*^* = \mathrm{st}_{N_*^*}(T)$.

(i) First, write down the gambles at the final chance nodes. For example, at N_{11}^{11} the gamble is $(10-c)E_1 + (15-c)E_2 = X - c$, and similarly for all others.

strategy	gamble
$d_{\overline{S}}$, then d_1	X
$d_{\overline{S}}$, then d_2	Y
d_S, then d_1 if S_1 and d_1 if S_2	$X - c$
d_S, then d_2 if S_1 and d_2 if S_2	$Y - c$
d_S, then d_1 if S_1 and d_2 if S_2	$S_1X + S_2Y - c$
d_S, then d_2 if S_1 and d_1 if S_2	$S_1Y + S_2X - c$

Table 6.3: Strategies and gambles for the lake district problem.

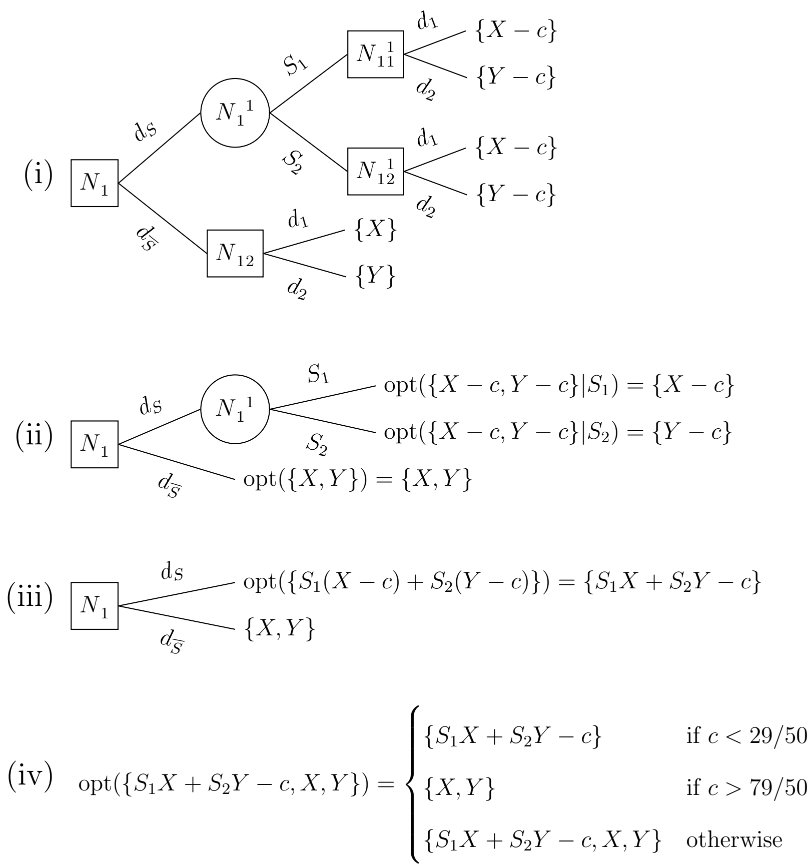

Figure 6.3: Solving the lake district example by backward induction.

(ii) Let us first deal with the branch corresponding to refusing the newspaper. At the decision node N_{12}, we have a choice between two strategies that correspond to the gambles X and Y. We also have $\mathrm{ev}(T_{12}) = \Omega$. So to determine the optimal strategies in this subtree, we must compare these two gambles unconditionally:

$$\underline{P}(X - Y) = \underline{P}(Y - X) = -5\epsilon = -1/2,$$

so at T_{12} the strategies d_1 and d_2 are both optimal.

Now we move to the branch corresponding to buying the newspaper. At N_{11}^{1}, we need to compare $X - c$ and $Y - c$. We have that $\mathrm{ev}(T_{11}^{1}) = S_1$, and

$$\underline{P}((X - c) - (Y - c)|S_1) = \tfrac{6-31\epsilon}{3+2\epsilon} > 0,$$

so $X - c >_{\underline{P}|S_1} Y - c$ and the uniquely optimal strategy is d_1. Next, considering N_{12}^{1}, we see that $\mathrm{ev}(T_{12}^{1}) = S_2$, and

$$\underline{P}((Y - c) - (X - c)|S_2) = \tfrac{6-31\epsilon}{2+3\epsilon} > 0,$$

so the optimal strategy here is d_2.

(iii) Moving to $N_1{}^1$, we see that only one of the original four strategies remains: "d_1 if S_1 and d_2 if S_2", corresponding to the gamble $S_1X + S_2Y - c$.

(iv) Finally, considering the entire tree T, three strategies are left: "d_1 if S_1 and d_2 if S_2"; "$d_{\overline{S}}$, then d_1"; "$d_{\overline{S}}$, then d_2". Therefore we need to find

$$\mathrm{opt}_{>_{\underline{P}|\cdot}}(\{S_1X + S_2Y - c, X, Y\}).$$

We have

$$\begin{aligned}
\underline{P}\left(X - (S_1X + S_2Y - c)\right) &= c - (6 + 19\epsilon)/5 = c - 79/50 \\
\underline{P}\left((S_1X + S_2Y - c) - X\right) &= (6 - 31\epsilon)/5 - c = 29/50 - c \\
\underline{P}\left(Y - (S_1X + S_2Y - c)\right) &= c - (6 + 19\epsilon)/5 = c - 79/50 \\
\underline{P}\left((S_1X + S_2Y - c) - Y\right) &= (6 - 31\epsilon)/5 - c = 29/50 - c
\end{aligned}$$

Concluding (see Fig. 6.3(iv)):

- if the newspaper costs less than 29/50, we should buy and follow its advice.
- if it costs more than 79/50, we do not buy, but have insufficient information to decide whether to take the waterproof or not.
- if the newspaper costs between 29/50 and 79/50, we can take any of the three remaining options.

Comparing this with the solution calculated in Section 1.2.4, we observe that the imprecision has created a range of c for which it is unclear whether buying the newspaper is better than not, rather than the single value for c in the precise case. Despite this, should the subject decide to buy the newspaper, she will follow the same policy in both cases: take the waterproof only if the newspaper predicts rain. Finally it should be noted that, although in both cases both $d_{\overline{S}}d_1$ and $d_{\overline{S}}d_2$ are involved in optimal normal form decisions for some values of c, in the precise case this is because they are *equivalent* and in the imprecise case they are *incomparable.* A tiny increase in value for, say, not taking the waterproof and no rain, would make $d_{\overline{S}}d_1$ always non-optimal under E_P but still optimal under $\underline{P}$ for $c \geq 29/50$.

6.3.2 The Oil Wildcatter

The oil wildcatter is a classic introductory example of a decision tree. We solve the version used by Kikuti et al. [38, Fig. 2]. Fig. 6.4 depicts the decision tree, with utiles in units of \$10000. The subject must decide whether to drill for oil (d_2) or not (d_1). Drilling costs 7 and provides a return of 0, 12, or 27 depending on the richness of the site. The events S_1 to S_3 represent the different yields, with S_1 being the least profitable and S_3 the most. The subject may pay 1 to test the site before deciding whether to drill; this gives one of three results T_1 to T_3, where T_1 is the most pessimistic and T_3 the most optimistic.

Lower and upper probabilities are given for each T_i (Table 6.4), and for each S_i conditional on T_i (Table 6.5). (Some intervals are tighter than those in Kikuti et al., since their values are incoherent—we corrected these by natural extension [79, §3.1].)

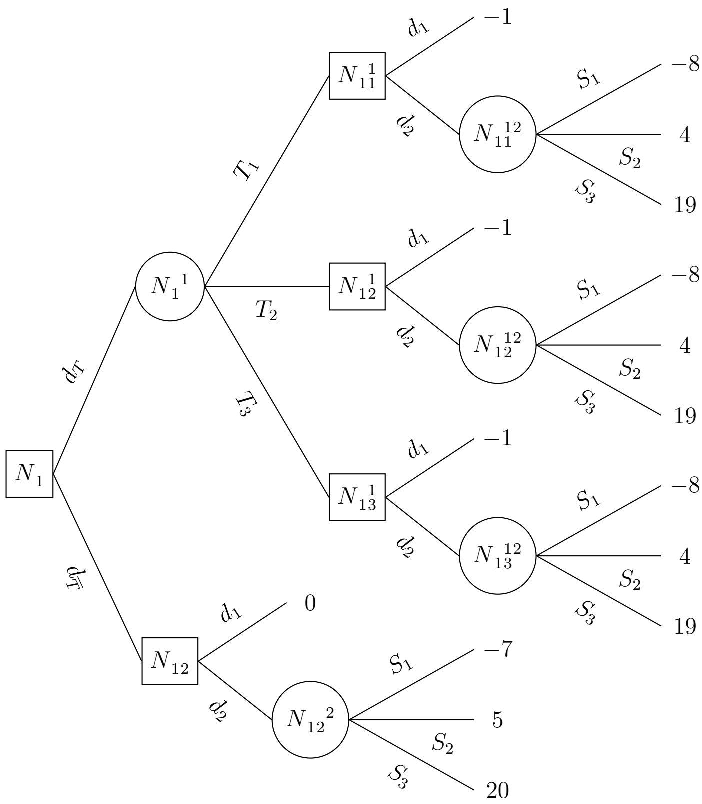

Figure 6.4: Decision tree for the oil wildcatter.

T_1	T_2	T_3
0.183, 0.222	0.333, 0.363	0.444, 0.454

Table 6.4: Unconditional lower and upper probabilities $\underline{P}(T_i)$ and $\overline{P}(T_i)$ for oil example.

	T_1	T_2	T_3
S_1	0.547, 0.653	0.222, 0.333	0.111, 0.166
S_2	0.222, 0.272	0.363, 0.444	0.333, 0.363
S_3	0.125, 0.181	0.250, 0.363	0.471, 0.556

Table 6.5: Conditional lower and upper probabilities $\underline{P}(S_i|T_i)$ and $\overline{P}(S_i|T_i)$ for oil example.

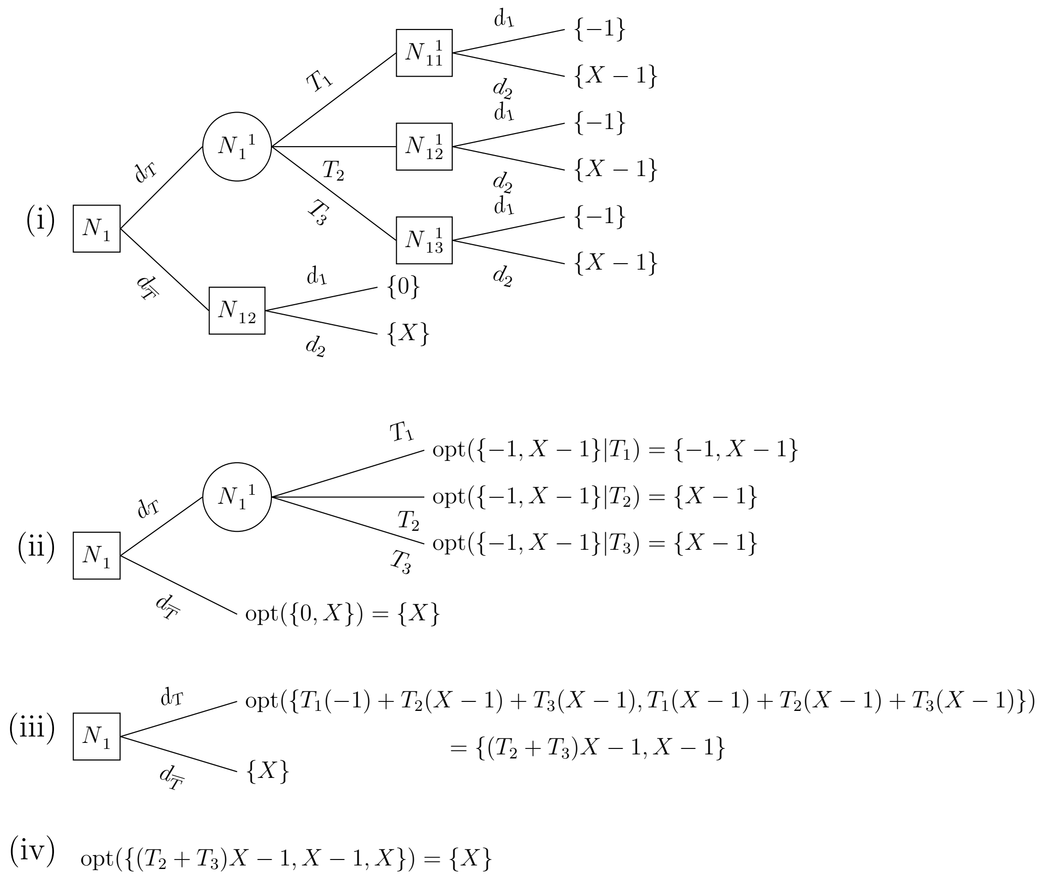

Figure 6.5: Solving the oil wildcatter example by normal form backward induction.

By *marginal extension* [79, §6.7.2], the lower prevision of a gamble Z is then

$$\underline{P}(Z) = \underline{P}(T_1\underline{P}(Z|T_1) + T_2\underline{P}(Z|T_2) + T_3\underline{P}(Z|T_3)).$$

Let $X = -7S_1 + 5S_2 + 20S_3$, and again let $T_*^* = \mathrm{st}_{N_*^*}(T)$. Since we will only be concerned with maximality, and normal form decisions in this problem are uniquely identified by their gambles, we can conveniently work with gambles in this example. Therefore, we use the following notation:

$$\mathrm{opt} = \mathrm{opt}_{>_{\underline{P}|\cdot}} \qquad \mathrm{back} = \mathrm{gamb} \circ \mathrm{back}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}} \qquad \mathrm{norm} = \mathrm{gamb} \circ \mathrm{norm}_{\mathrm{opt}_{>_{\underline{P}|\cdot}}}$$

(i) Of course, $\mathrm{back}(\cdot)$ at the final chance nodes simply reports the gamble:

$$\mathrm{back}(T_{11}{}^{12}) = \mathrm{back}(T_{11}{}^{22}) = \mathrm{back}(T_{11}{}^{32}) = \{X - 1\},$$

and $\mathrm{back}(T_{12}{}^{2}) = \{X\}$.

(ii) For T_{11}^1, we must find $\underline{P}((X-1)-(-1)|T_1)$ and $\underline{P}(-1-(X-1)|T_1)$. These lower previsions can be computed using Table 6.5 as follows: X will have lowest expected value when the worst outcome S_1 is most likely (probability 0.653) and the best outcome S_3 is least likely (probability 0.125), and so the probability of S_2 is 0.222. So, $\underline{P}(X|T_1) = -7 \times 0.653 + 5 \times 0.222 + 20 \times 0.125 = -0.961$. Similarly, $\underline{P}(-X|T_1) = -1.151$. Neither of these is positive, so $\text{back}(T_{11}^1) = \{X-1, -1\}$.

For T_{12}^1, $\underline{P}((X-1)-(-1)|T_2) = 4.754$, and therefore d_2 dominates d_1, so $\text{back}(T_{12}^1) = \{X-1\}$. Similarly, $\underline{P}((X-1)-(-1)|T_3) = 10.073$, so $\text{back}(T_{13}^1) = \{X-1\}$.

For T_{12}, we need to find $\underline{P}(X-0)$. By marginal extension we have

$$\begin{aligned}\underline{P}(X) &= \underline{P}(T_1\underline{P}(X|T_1) + T_2\underline{P}(X|T_2) + T_3\underline{P}(X|T_3))\\ &= \underline{P}(-0.961T_1 + 4.754T_2 + 10.073T_3)\\ &= 0.222 \times -0.961 + 0.334 \times 4.754 + 0.444 \times 10.073 = 5.846906.\end{aligned}$$

This is greater than zero, so $\text{back}(T_{12}) = \{X\}$.

(iii) At ${T_1}^1$ there are two potentially optimal gambles: $T_1(X-1) + T_2(X-1) + T_3(X-1) = X-1$ and $T_1(-1)+T_2(X-1)+T_3(X-1) = (T_2+T_3)X-1$. We must find $\underline{P}((X-1)-((T_2+T_3)X-1)) = \underline{P}(T_1X)$ and $\underline{P}(((T_2+T_3)X-1)-(X-1)) = \underline{P}(-T_1X)$. Using marginal extension,

$$\underline{P}(T_1X) = \underline{P}(T_1\underline{P}(X|T_1)) = \underline{P}(-0.961T_1) = 0.222 \times -0.961 = -0.213342 < 0,$$

$$\underline{P}(-XT_1) = \underline{P}(T_1\underline{P}(-X|T_1)) = \underline{P}(-1.151T_1) = 0.222 \times -1.151 = -0.255522 < 0,$$

so $\text{back}({T_1}^1) = \{X-1, (T_2+T_3)X-1\}$.

(iv) Finally, for T, we must consider $\{X, X-1, (T_2+T_3)X-1\}$. It is clear that $\underline{P}(X-(X-1)) = 1 > 0$, so $X-1$ can be eliminated. It is also clear that if a gamble does not dominate $X-1$ then is also does not dominate X, so by our calculation at ${T_1}^1$ we know that X is maximal. We finally have

$$\underline{P}(X-((T_2+T_3)X-1)) = \underline{P}(T_1X+1) = \underline{P}(T_1X) + 1 = -0.213342 + 1 > 0,$$

so $\text{back}(T) = \{X\}$. So, the optimal strategy is: do not test and just drill.

We found a single maximal strategy. By Corollary 6.17, it is also the unique E-admissible strategy. (Our solution differs from Kikuti et al. [38]; since they do not detail their calculations, we could not identify why.) Of course, if the imprecision was larger, we would have found more, but it does show that non-trivial sequential problems can give unique solutions even when probabilities are imprecise.

In this example, the usual normal form method requires comparing 10 gambles at once. By normal form backward induction, we only had to compare 2 gambles at once at each stage (except at the end, where we had 3), leading us much more quickly to the solution: the computational benefit of normal form backward induction is obvious.

6.4 Subtree Perfectness

Since Γ-maximin and interval dominance fail to satisfy the conditions of Theorem 3.12, they lack subtree perfectness (Corollary 3.21). Since maximality and E-admissibility are not total preorders, they lack subtree perfectness because of Intersection.

All is not lost, however. For particular restrictions on $\underline{P}$ or special types of decision tree, some of the choice functions have subtree perfectness. Further, these restrictions and special cases are practically useful. It is only interval dominance for which such interesting cases could not be found (any effort to do so seems to give a trivial tree or a linear prevision).

Γ-maximin

Although Γ-maximin can be criticised strongly for failing to model the indecision inherent to an imprecise model, it is the most fruitful for subtree perfectness. This is because the indecision is precisely the feature that causes Intersection to fail, and so any choice function that accurately models it must fail subtree perfectness. For Γ-maximin, all we need to do is to force $\underline{P}$ to satisfy Mixture. It turns out that all we need for this is marginal extension on all relevant partitions.

Definition 6.23. *Let $\underline{P}$ be a coherent lower prevision, T be a consistent decision tree, and N be a chance node in T. Let $\mathcal{E}$ be the partition of Ω at N, and $B = \operatorname{ev}(\operatorname{st}_N(T))$. $\underline{P}$ satisfies marginal extension at N if, for all $X \in \operatorname{gamb}(\operatorname{st}_N(T))$, $\underline{P}$ satisfies marginal extension for X with respect to $\mathcal{E}$ and B:*

$$\underline{P}(X|B) = \underline{P}(\underline{P}(X|\mathcal{E} \cap B)|B).$$

Since Γ-maximin satisfies Conditioning and Intersection, we only need to show that it satisfies Mixture for all relevant events and gambles. It is perhaps more instructive to show that it satisifes Multiple Mixture instead. First we need a technical detail about lower previsions.

Lemma 6.24. *For any lower prevision $\underline{P}$, any non-empty events A and B with $A \cap B \neq \emptyset$, any real numbers λ, μ with $\lambda > \mu$, and any gamble X,*

$$\underline{P}(\lambda A + X|B) > \underline{P}(\mu A + X|B).$$

Proof. For any $P \in \mathcal{M}$,

$$P(\lambda A + X|B) = \lambda P(A|B) + P(X|B) > \mu P(A|B) + P(X|B) = P(\mu A + X|B).$$

Therefore,

$$\min_{P\in\mathcal{M}} P(\lambda A + X|B) > \min_{P\in\mathcal{M}} P(\mu A + X|B).$$

□

Lemma 6.25. *Let $\underline{P}$ be a coherent lower prevision, B be a non-empty event, $\mathcal{A} = \{A_1 \dots, A_n\}$ be a partition of Ω such that $A_i \cap B \neq \emptyset$ for all i, $\mathcal{X}_1, \dots, \mathcal{X}_n$ be non-empty finite sets of gambles where $\mathcal{X}_i$ is $A_i \cap B$-consistent for all i. If $\underline{P}$ satisfies marginal extension for each $\sum_{i=1}^n A_i X_i \in \sum_{i=1}^n A_i \mathcal{X}_i$ (with respect to $\mathcal{A}$ and B), then*

$$\operatorname{opt}_{\underline{\mathrm{P}}}\left(\sum_{i=1}^n A_i \mathcal{X}_i \middle| B\right) = \sum_{i=1}^n A_i \operatorname{opt}_{\underline{\mathrm{P}}}(\mathcal{X}_i|A_i \cap B).$$

Proof. Let $\sum_{i=1}^n A_i X_i \in \sum_{i=1}^n A_i \mathcal{X}_i$. By marginal extension,

$$\underline{P}\left(\sum_{i=1}^n A_i X_i \middle| B\right) = \underline{P}\left(\sum_{i=1}^n A_i \underline{P}(X_i|A_i \cap B) \middle| B\right).$$

Suppose there is a k such that $X_k \notin \mathrm{opt}_{\underline{P}}(\mathcal{X}_k|A_k \cap B)$. Then there is a $Y_k \in \mathcal{X}_k$ such that $\underline{P}(Y_k|A_k \cap B) > \underline{P}(X_k|A_k \cap B)$. Since $\underline{P}(A_k|B) > 0$, replacing X_k by Y_k in $\sum_{i=1}^n A_i X_i$ will increase the lower prevision of this gamble by Lemma 6.24. Therefore $\sum_{i=1}^n A_i X_i$ is not optimal with respect to Γ-maximin.

Suppose instead that $X_i \in \mathrm{opt}_{\underline{P}}(\mathcal{X}_i|A_i \cap B)$ for all i. We know that, for any i and any $Y_i \in \mathcal{X}_i$, $\underline{P}(X_i|A_i \cap B) \geq \underline{P}(Y_i|A_i \cap B)$. Therefore,

$$\underline{P}\left(\sum_{i=1}^n A_i \underline{P}(X_i|A_i \cap B)\middle| B\right) \geq \underline{P}\left(\sum_{i=1}^n A_i \underline{P}(Y_i|A_i \cap B)\middle| B\right) = \underline{P}\left(\sum_{i=1}^n A_i Y_i \middle| B\right)$$

for any $Y_1 \in \mathcal{X}_1, \ldots, Y_n \in \mathcal{X}_n$. Therefore $\sum_{i=1}^n A_i X_i$ is optimal with respect to Γ-maximin. This establishes Multiple Mixture. $\square$

Theorem 6.26. *Let $\underline{P}$ be a coherent lower prevision. For any consistent decision tree T such that $\underline{P}$ satisfies marginal extension for all chance nodes in T,* $\mathrm{norm}_{\mathrm{opt}_{\underline{P}}}$ *is subtree perfect in T.*

Proof. We follow the same approach as in the proof of Theorem 2.6, omitting details for brevity. The induction hypothesis is that subtree perfectness holds in each subtree at the immediate successors of the root node. The inductive step is to show that for all $K \in \mathrm{ch}(T)$,

$$\mathrm{norm}_{\mathrm{opt}_{\underline{P}}}(\mathrm{st}_K(T)) = \mathrm{st}_T(\mathrm{norm}_{\mathrm{opt}_{\underline{P}}}(T)).$$

If T has its root at a decision node then this follows immediately from Intersection as we have seen before. If T has its root at a chance node, then this follows immediately because the conditions of Lemma 6.25 are satisfied, and $\mathrm{opt}_{\underline{P}}$ satisfies Conditioning. Therefore subtree perfectness follows from Lemma 2.11. $\square$

It appears that marginal extension is enough to ensure subtree perfectness (and hence also backward induction) for Γ-maximin. Unfortunately, although marginal extension may be satisfied for $\underline{P}$ for all relevant partitions of a particular decision tree, for any $\underline{P}$ that is not linear we can always find a decision tree where marginal extension is *not* satisfied for the relevant partitions. Thus, marginal extension can provide subtree perfectness for some problems, but it does not solve everything. Interestingly, we see in Proposition 6.34 that marginal extension is not sufficient for locality.

E-admissibility

E-admissibility satisfies Conditioning and Mixture but not Intersection. As can be seen in the proof of Lemma 2.10, the failure of Intersection causes problems in trees with multiple decision nodes. These problems would not manifest in trees where E-admissibility selects only one option at every decision arc. But in such trees, using E-admissibility would then be equivalent to just picking any one $P \in \mathcal{M}$ and maximizing expected utility under that; the choice of P is irrelevant. It is not particularly interesting that E-admissibility is subtree perfect for problems where using any consistent P would be identical.

We can, however, use the results of Section 2.4 to demonstrate subtree perfectness for trees with at most one decision node per branch. Here, failure of Intersection is not a problem.

Lemma 6.27. *Let* $\{\mathrm{opt}_i \colon i \in \mathcal{I}\}$ *be a family of choice functions. For any non-empty event A and any non-empty finite set of A-consistent gambles $\mathcal{X}$, let*

$$\mathrm{opt}(\mathcal{X}|A) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i(\mathcal{X}|A).$$

If all the opt_i *satisfy Weak Multiple Mixture then so does* opt.

Proof. We have

$$\mathrm{opt}\left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B\right) = \bigcup_{i \in \mathcal{I}} \mathrm{opt}_i\left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B\right).$$

If

$$\bigoplus_{j=1}^{n} A_j X_j \in \mathrm{opt}\left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B\right),$$

then there is at least one i such that

$$\bigoplus_{j=1}^{n} A_j X_j \in \mathrm{opt}_i\left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B\right).$$

By assumption, for such an i,

$$X_j \in \mathrm{opt}_i(\mathcal{X}_j | A_j \cap B),$$

and so by definition of opt,

$$X_j \in \mathrm{opt}(\mathcal{X}_j | A_j \cap B).$$

This proves the first part of the property.

If $X_k \in \mathrm{opt}(\mathcal{X}_k|A_k \cap B)$, then there is at least one $i \in \mathcal{I}$ such that there are $X_j \in \mathcal{X}_j$ for each $j \neq k$ such that

$$\bigoplus_{j=1}^{n} A_j X_j \in \mathrm{opt}_i \left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B \right),$$

whence

$$\bigoplus_{j=1}^{n} A_j X_j \in \mathrm{opt} \left(\bigoplus_{j=1}^{n} A_j \mathcal{X}_j \middle| B \right)$$

as required. □

Corollary 6.28. *Let $T = \bigodot E_i T_i$ be a consistent decision tree where each of the T_i has a decision node as the root, and there is only one decision node in every branch. Then* $\mathrm{norm}_{\mathrm{opt}_{\mathcal{M}}}$ *is subtree perfect in this tree.*

Proof. Follows from Lemmas 6.12 and 6.27 and Theorem 2.12. □

Maximality

Weak Multiple Mixture has so far resisted proof or counter-example for maximality. There are, however, some special cases where it is easy to demonstrate.

Lemma 6.29. *Maximality satisfies Weak Multiple Mixture when, for all i, $|\mathcal{X}_i| \leq 2$.*

Proof. The first part of Weak Multiple Mixture is just Backward Mixture, which holds for maximality. For the second part, let $X_k \in \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}_k|A_k \cap B)$ for some k. It follows immediately from the definitions of E-admissibility and maximality that $\mathrm{opt}_{\mathcal{M}}(\mathcal{Y}|A) = \mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{Y}|A)$ if $|\mathcal{Y}| \leq 2$. Therefore, $X_k \in \mathrm{opt}_{\mathcal{M}}(\mathcal{X}_k|A_k \cap B)$. E-admissibility satisfies Weak Multiple Mixture, so for each $j \neq k$ there is an $X_j \in \mathcal{X}_k$ such that $\sum_{i=1}^{n} A_i X_i \in \mathrm{opt}\left(\sum_{i=1}^{n} A_i \mathcal{X}_i | B\right)$. Now, any E-admissible gamble is maximal [77], so the second part of Weak Multiple Mixture holds for maximality. □

Unsurprisingly, Weak Multiple Mixture holds whenever $\underline{P}$ satisfies marginal extension on the relevant gambles and events. In fact, Multiple Mixture holds (this is also true for E-admissibility, by a similar proof).

Lemma 6.30. *Suppose $\underline{P}$ satisfies marginal extension for $\sum_{i=1}^{n} A_i \mathcal{X}_i$ with respect to $\mathcal{A}$ and B. Then, maximality satisfies Multiple Mixture for these $\mathcal{X}_i$, A_i, B.*

Proof. Let $X_1, \dots, X_n$ be gambles such that $X_i \in \mathrm{opt}(X_k|A_k \cap B)$. Let $\sum_{i=1}^n A_i Y_i$ be a gamble in $\sum_{i=1}^n A_i \mathcal{X}_i$. Then,

$$\begin{aligned}\underline{P}\left(\sum_{i=1}^n A_i(Y_i - X_i)\middle| B\right) &= \underline{P}\left(\sum_{i=1}^n A_i \underline{P}\left(\sum_{j=1}^n A_j(Y_j - X_j)\middle| A_i \cap B\right)\middle| B\right) \\ &= \underline{P}\left(\sum_{i=1}^n A_i \underline{P}(Y_i - X_i|A_i \cap B)\middle| B\right) \\ &\leq 0\end{aligned}$$

by conditional maximality of each of the X_i. Therefore

$$\mathrm{opt}_{>_{\underline{P}|\cdot}}\left(\sum_{i=1}^n A_i \mathcal{X}_i\middle| B\right) \supseteq \sum_{i=1}^n A_i \,\mathrm{opt}_{>_{\underline{P}|\cdot}}(\mathcal{X}_i|A_i \cap B).$$

The opposite inclusion follows from backward induction. □

6.5 Cumulative Decision Processes

In this section we apply the theory for locality and deterministic system trees to imprecise probability.

6.5.1 Locality

This section is based on joint work with Ricardo Shirota Filho, adapted from [76].

It is interesting that the property of locality does not in general hold for any of the choice functions. We shall instead investigate minimal restrictions on the form of $\underline{P}$ for each of the choice functions to work. As with subtree perfectness, we find that marginal extension is crucial. We need to express marginal extension in a form consistent with our notation for this subject.

Definition 6.31. *Let the possibility space be $\Omega = S_0 \times \cdots \times S_n$. A coherent lower prevision $\underline{P}$ is then said to satisfy marginal extension with respect to S_0, ..., S_n whenever, for all $1 \leq k < n$, all gambles Z on F_k, and all $h_{k-1} \in H_{k-1}$,*

$$\underline{P}(Z|h_{k-1}) = \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$$

In the above definition, $\underline{P}(Z|h_{k-1}S_k)$ denotes the gamble

$$\underline{P}(Z|h_{k-1}S_k)\colon s_k \mapsto \underline{P}(Z|h_{k-1}s_k)$$

Note that the order of the state spaces is relevant for marginal extension. For instance, satisfying marginal extension with respect to S_0, S_1, S_2, is not equivalent to satisfying marginal extension with respect to S_0, S_2, S_1.

Finally, note that for conditional previsions E_P, marginal extension corresponds to disintegrability [17, p. 90, Eq. (3)], and hence is always satisfied in our case (since we are concerned with finite state spaces only).

Maximality

Assume that we are given a coherent lower prevision $\underline{P}(\cdot|\cdot)$—by natural (or regular) extension, we may assume without loss of generality that $\underline{P}(\cdot|\cdot)$ is defined on all gambles on $S_0 \times \cdots \times S_n$, and conditional on all non-empty events in $S_0 \times \cdots \times S_n$. Then, a policy $\pi_k^{n*} \in \Pi_k^n$ is optimal, in the sense of maximality, whenever

$$\overline{P}(X_k^n(h_{k-1}, \pi_k^{n*}) - X_k^n(h_{k-1}, \pi_k^n)|h_{k-1}) \geq 0 \tag{6.3}$$

for all policies $\pi_k^n \in \Pi_k^n$.

Proposition 6.32. *Maximality with respect to a coherent lower prevision satisfies locality on S_0, ..., S_n, if and only if $\underline{P}$ satisfies marginal extension with respect to S_0, ..., S_n.*

Proof. By Theorem 4.2, it suffices to show that maximality satisfies sequential distributivity if and only if the given condition holds.

"if". Consider any $1 \leq k < n$, any value h_{k-1} of H_{k-1}, any finite set of gambles $\mathcal{X}$ on S_k, any finite sets of gambles $\mathcal{Y}(s_k)$ on F_{k+1} (one such set for each $s_k \in S_k$), and any $X \in \mathcal{X}$ and $Y(s_k) \in \mathcal{Y}(s_k)$. We must show that

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle| h_{k-1}\right)$$
$$\iff X \in \operatorname{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

or, by the definition of maximality, that

$$\forall X' \in \mathcal{X},\ s_k \in S_k,\ Y'(s_k) \in \mathcal{Y}(s_k)\colon$$
$$\overline{P}\left(X + \bigoplus_{s_k} E_{s_k} Y(s_k) - X' - \bigoplus_{s_k} E_{s_k} Y'(s_k) \middle| h_{k-1}\right) \geq 0 \quad (6.4)$$

is equivalent to

$$\forall X' \in \mathcal{X}\colon \overline{P}\,(X - X' | h_{k-1}) \geq 0, \text{ and} \qquad (6.5)$$
$$\forall s'_k \in S_k,\ Y''(s'_k) \in \mathcal{Y}(s'_k)\colon \overline{P}\,(Y(s'_k) - Y''(s'_k) | h_{k-1} s'_k) \geq 0 \qquad (6.6)$$

Obviously, Eq. (6.4) implies Eq. (6.5): simply consider the case in Eq. (6.4) where $Y'(s_k) = Y(s_k)$ for all s_k.

Next, we show that Eq. (6.4) implies Eq. (6.6). Consider any s'_k and $Y''(s'_k)$. In Eq. (6.4), take $X' = X$, and $Y'(s_k) = Y(s_k)$ for all s_k except s'_k, for which $Y'(s'_k) = Y''(s'_k)$:

$$\begin{aligned} 0 &\leq \overline{P}\left(\bigoplus_{s_k} E_{s_k} Y(s_k) - \bigoplus_{s_k} E_{s_k} Y'(s_k) \middle| h_{k-1}\right) \\ &= \overline{P}\left(E_{s'_k}(Y(s'_k) - Y''(s'_k)) | h_{k-1}\right) \end{aligned}$$

and by coherence [79, 6.3.5(5), p. 296],

$$\leq \overline{P}\left(\overline{P}\left(E_{s'_k}(Y(s'_k) - Y''(s'_k)) \middle| h_{k-1} S_k\right) \middle| h_{k-1}\right)$$

and by separate coherence,

$$= \overline{P}\left(E_{s'_k} \overline{P}\,(Y(s'_k) - Y''(s'_k) | h_{k-1} s'_k) \middle| h_{k-1}\right)$$

But, because $\underline{P}(E_{s'_k} | h_{k-1}) > 0$, the above inequality can only hold if

$$\overline{P}\,(Y(s'_k) - Y''(s'_k) | h_{k-1} s'_k) \geq 0$$

which establishes the desired implication.

To complete the proof we show the opposite implication: if Eqs. (6.5) and (6.6) hold, will Eq. (6.4) hold too?

Consider any $X' \in \mathcal{X}$, $s_k \in S_k$, and $Y'(s_k) \in \mathcal{Y}(s_k)$. Then,

$$\overline{P}\left(X + \bigoplus_{s_k} E_{s_k} Y(s_k) - X' - \bigoplus_{s_k} E_{s_k} Y'(s_k) \middle| h_{k-1}\right)$$

$$= \overline{P}\left(X - X' + \bigoplus_{s_k} E_{s_k}(Y(s_k) - Y'(s_k)) \middle| h_{k-1}\right)$$

and by assumption on $\underline{P}$,

$$= \overline{P}\left(\overline{P}\left(X - X' + \bigoplus_{s_k} E_{s_k}(Y(s_k) - Y'(s_k)) \middle| h_{k-1}S_k\right) \middle| h_{k-1}\right)$$

and by separate coherence,

$$= \overline{P}\left(X - X' + \bigoplus_{s_k} E_{s_k} \overline{P}\left(Y(s_k) - Y'(s_k) | h_{k-1}s_k\right) \middle| h_{k-1}\right)$$

and by monotonicity of $\overline{P}$, and Eq. (6.6),

$$\geq \overline{P}\left(X - X' | h_{k-1}\right)$$

and by Eq. (6.5),

$$\geq 0.$$

"only if". We prove this part by contradiction. Assume that

$$\underline{P}(Z|h_{k-1}) > \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}).$$

for some gamble Z and h_{k-1} (indeed, if equality does not hold, we must have a strict inequality as given, by [79, 6.3.5(5), p. 296]).

Consider the following sets of gambles:

$$\mathcal{X} = \{\underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}), \underline{P}(Z|h_{k-1}S_k)\}$$
$$\mathcal{Y}(s_k) = \{Z(s_k, \cdot), \underline{P}(Z|h_{k-1}s_k)\} \text{ for all } s_k$$

Let $X = \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$ and $Y(s_k) = \underline{P}(Z|h_{k-1}s_k)$. We can easily show that $X \in \mathrm{opt}(\mathcal{X}|h_{k-1})$:

$$\overline{P}(\underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) - \underline{P}(Z|h_{k-1}S_k)|h_{k-1})$$
$$= \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) + \overline{P}(-\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$$
$$= \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) - \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) = 0.$$

If we can also show that

$$Y(s_k) \in \mathrm{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k \tag{6.7}$$

but

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \notin \mathrm{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle| h_{k-1}\right) \tag{6.8}$$

then it is established that Sequential Distributivity cannot hold.

Indeed, for any $Y''(s_k) \in \mathcal{Y}(s_k)$,

$$\begin{aligned}\overline{P}\left(Y(s_k) - Y''(s_k)|h_{k-1}s_k\right) &= \overline{P}\left(\underline{P}(Z|h_{k-1}s_k) - Y''(s_k)|h_{k-1}s_k\right)\\ &= \underline{P}(Z|h_{k-1}s_k) + \overline{P}\left(-Y''(s_k)|h_{k-1}s_k\right)\\ &= \underline{P}(Z|h_{k-1}s_k) - \underline{P}\left(Y''(s_k)|h_{k-1}s_k\right)\end{aligned}$$

but, $Y''(s_k)$ is either $\underline{P}(Z|h_{k-1}s_k)$ or $Z(s_k, \cdot)$, and in either case,

$$= 0.$$

which shows that Eq. (6.7) is satisfied.

However, for $X' = \underline{P}(Z|h_{k-1}S_k)$ and $Y'(s_k) = Z(s_k, \cdot)$, we have that

$$\begin{aligned}&\underline{P}\left(X' + \bigoplus_{s_k} E_{s_k} Y'(s_k) - X - \bigoplus_{s_k} E_{s_k} Y(s_k)\middle| h_{k-1}\right)\\ &= \underline{P}\left(\underline{P}(Z|h_{k-1}S_k) + Z - \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) - \underline{P}(Z|h_{k-1}S_k)|h_{k-1}\right)\\ &= \underline{P}(Z|h_{k-1}) - \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) > 0\end{aligned}$$

which means that Eq. (6.8) is satisfied. □

A first observation is that locality provides a behavioural argument for marginal extension: violating marginal extension with respect to some sequence of states S_0, …, S_n, implies violating locality for some act-state independent sequential decision problems on S_0, …, S_n. Although marginal extension is a convenient assumption to make, for instance due to computational reasons [79, §6.7.5, p. 316], we are not aware of any other behavioural motivation.

How does locality work in practice for maximality? Let us return to our coin-tossing example. First, our agent must assess a coherent lower prevision $\underline{P}(\cdot|\cdot)$ reflecting his beliefs about the coin. For instance, he could use the *imprecise Dirichlet* *model* (IDM) [80], which states that, for any gamble X on $S_k = \{H, T\}$:

$$\underline{P}(X|h_{k-1}) = \frac{n_H X(H) + n_T X(T) + s \min\{X(H), X(T)\}}{n_H + n_T + s}$$

where n_H is the number of heads observed in h_{k-1}, n_T is the number of tails in h_{k-1}, and s is a hyper-parameter of the model, usually taken to be 1 or 2. Eq. (6.9) is called the *predictive lower prevision.* Similarly, the *predictive upper prevision* is

$$\overline{P}(X|h_{k-1}) = \frac{n_H X(H) + n_T X(T) + s \max\{X(H), X(T)\}}{n_H + n_T + s}$$

The IDM models a completely vacuous state of knowledge if $n_H = n_T = 0$, and converges to the empirical expectation as $n_H + n_T$ grows, hence this seems a reasonable model. It does, however, involve events of lower probability zero, so this does not fall within our restricted class of coherent lower previsions. It is well known, however, that a small perturbation of a coherent lower prevision does not affect optimality much [75]. So, for small ϵ, we can use the alternative specifications

$$\underline{P}(X|h_{k-1}) = \frac{n_H X(H) + n_T X(T) + s \min\{X(H), X(T)\}}{n_H + n_T + s} \cdot (1 - \epsilon) + \epsilon Q(X|h_{k-1}) \tag{6.9}$$

and

$$\overline{P}(X|h_{k-1}) = \frac{n_H X(H) + n_T X(T) + s \max\{X(H), X(T)\}}{n_H + n_T + s} \cdot (1 - \epsilon) + \epsilon Q(X|h_{k-1}), \tag{6.10}$$

where Q is a linear prevision (i.e. expectation) that has all probabilities strictly positive—for example, the conditional expectation with respect to the uniform mass function. This ensures that all events have positive lower probability. We can then combine all these marginal predictive lower previsions into a joint model satisfying marginal extension, and apply the theorem.

Applying maximality to our example is now straightforward. By definition of maximality (Eq. (6.3)), betting on heads is locally maximal if

$$\overline{P}(X_k^k(h_{k-1}, d_H) - X_k^k(h_{k-1}, d_T)|h_{k-1}) \geq 0.$$

By $X_k^k(h_{k-1}, d_H) = -X_k^k(h_{k-1}, d_T)$, and Eqs. (4.7) and (6.10), we conclude that, for ϵ sufficiently small, betting on heads is locally optimal whenever $n_H \geq n_T - s$, and similarly, betting on tails is optimal whenever $n_T \geq n_H - s$. By construction, $\underline{P}(\cdot|\cdot)$ satisfies marginal extension, so applying Proposition 6.32, we conclude that this is also the global solution.

E-admissibility

Perhaps surprisingly, E-admissibility satisfies locality if and only if maximality does:

Proposition 6.33. *E-admissibility satisfies locality on $S_0, \ldots, S_n$ if and only if $\underline{P}$ satisfies marginal extension with respect to $S_0, \ldots, S_n$.*

Proof. "if". If

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \mathrm{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k) \middle| h_{k-1}\right),$$

then, by definition of E-admissibility, there is a $Q \in \mathcal{M}$ such that

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \mathrm{opt}_Q\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k) \middle| h_{k-1}\right).$$

Since we assumed that $\underline{P}(E_{s_k}|h_{k-1}) > 0$, we know that $Q(E_{s_k}|h_{k-1}) > 0$. Given this, and Bayes theorem for linear previsions, it follows easily from Proposition 6.32 that Sequential Distributivity holds for maximality with respect to Q. Therefore,

$$X \in \mathrm{opt}_Q(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \mathrm{opt}_Q(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

and so, again by definition of E-admissibility,

$$X \in \mathrm{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \mathrm{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

We have proved one direction of Sequential Distributivity.

Let us now prove the other direction. If

$$X \in \mathrm{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \mathrm{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k,$$

then for all s_k there are linear previsions $Q(\cdot|h_{k-1}) \in \mathcal{M}(\cdot|h_{k-1})$ and $R(\cdot|h_{k-1}s_k) \in \mathcal{M}(\cdot|h_{k-1}s_k)$ such that

$$X \in \mathrm{opt}_Q(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \mathrm{opt}_R(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

By the lower envelope theorem for marginal extension [79, 6.7.4], the linear prevision Q, where

$$Q(\cdot|h_{k-1}) = Q(R(\cdot|h_{k-1}S_k)|h_{k-1}),$$

belongs to $\mathcal{M}(\cdot|h_{k-1})$. But, using the linearity of Q, Bayes theorem, and the above decomposition of Q,

$$\begin{aligned}
&\max_{\substack{X'\in\mathcal{X}\\ Y'(s_k)\in\mathcal{Y}(s_k)}} Q\left(X' + \bigoplus_{s_k} E_{s_k} Y'(s_k)\middle| h_{k-1}\right)\\
&= \max_{X'\in\mathcal{X}} Q(X'|h_{k-1}) + \sum_{s_k} Q(E_{s_k}|h_{k-1}) \max_{Y'(s_k)\in\mathcal{Y}(s_k)} Q(Y'(s_k)|h_{k-1}s_k)\\
&= \max_{X'\in\mathcal{X}} Q(X'|h_{k-1}) + \sum_{s_k} Q(E_{s_k}|h_{k-1}) \max_{Y'(s_k)\in\mathcal{Y}(s_k)} R(Y'(s_k)|h_{k-1}s_k)\\
&= Q(X|h_{k-1}) + \sum_{s_k} Q(E_{s_k}|h_{k-1}) R(Y(s_k)|h_{k-1}s_k)\\
&= Q(X|h_{k-1}) + \sum_{s_k} Q(E_{s_k}|h_{k-1}) Q(Y(s_k)|h_{k-1}s_k)\\
&= Q\left(X + \bigoplus_{s_k} E_{s_k} Y(s_k)\middle| h_{k-1}\right)
\end{aligned}$$

so,

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \mathrm{opt}_Q\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k)\middle| h_{k-1}\right),$$

whence

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \mathrm{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k)\middle| h_{k-1}\right).$$

"only if". Assume that

$$\underline{P}(Z|h_{k-1}) > \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}).$$

for some gamble Z and h_{k-1} (indeed, if equality does not hold, we must have a strict inequality as given, by [79, 6.3.5(5), p. 296]).

Consider the sets of gambles used in the maximality proof:

$$\begin{aligned}
\mathcal{X} &= \{\underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}), \underline{P}(Z|h_{k-1}S_k)\}\\
\mathcal{Y}(s_k) &= \{Z(s_k,\cdot), \underline{P}(Z|h_{k-1}s_k)\} \text{ for all } s_k
\end{aligned}$$

Let $X = \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$ and $Y(s_k) = \underline{P}(Z|h_{k-1}s_k)$. When a set of gambles has only two elements, E-admissibility and maximality coincide. So we know from the previous proof that, for E-admissibility,

$$X \in \mathrm{opt}(\mathcal{X}|h_{k-1})$$
$$Y(s_k) \in \mathrm{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

The set of E-admissible gambles is always a subset of the maximal gambles, so if a gamble is non-maximal in a set, it is not E-admissible either. In the proof for maximality, we showed that

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k)$$

is not maximal in

$$\mathcal{X} + \bigoplus_{s_k} E_{s_k} \mathcal{Y}(s_k),$$

and so it is not E-admissible. Hence, again, Sequential Distributivity does not hold. □

In order to apply E-admissibility to the example, simply note that E-admissibility is equivalent to maximality in case of binary choice: indeed, a gamble X is maximal in $\{X, Y\}$ if and only if

$$\overline{P}(X - Y) \geq 0$$

which is, by the lower envelope theorem [79, p. 134, §3.3.3], equivalent to

$$\exists Q \in \mathcal{M} \colon Q(X - Y) \geq 0,$$

which means exactly that X is E-admissible in $\{X, Y\}$.

Hence, locally, E-admissibility and maximality coincide and the agent always selects the same action in both cases. So, by Proposition 6.33, whose conditions we already verified earlier, E-admissibility and maximality also coincide globally.

Γ-maximin

Γ-maximin satisfies locality only in very restricted cases.

Proposition 6.34. *Γ-maximin with respect to a coherent lower prevision $\underline{P}$ satisfies locality on S_0, …, S_n if and only if the following conditions hold:*

(i) $\underline{P}$ *satisfies marginal extension with respect to* $S_0, \dots, S_n$*, and*

(ii) $\underline{P}$ *is locally linear in the sense that, for all* $1 \leq k < n$ *and all gambles* X *and* Y *on* S_k*,*

$$\underline{P}(X + Y|h_{k-1}) = \underline{P}(X|h_{k-1}) + \underline{P}(Y|h_{k-1}),$$

Proof. "if". Consider any $1 \leq k < n$, any value h_{k-1} of H_{k-1}, any finite set of gambles $\mathcal{X}$ on S_k, any finite sets of gambles $\mathcal{Y}(s_k)$ on F_{k+1} (one such set for each $s_k \in S_k$), and any $X \in \mathcal{X}$ and $Y(s_k) \in \mathcal{Y}(s_k)$. We must show that

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle|h_{k-1}\right)$$
$$\iff X \in \operatorname{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k. \tag{6.11}$$

We have, by marginal extension,

$$\underline{P}\left(X + \bigoplus_{s_k} E_{s_k} Y(s_k)\middle|h_{k-1}\right)$$
$$= \underline{P}\left(X + \bigoplus_{s_k} E_{s_k} \underline{P}(Y(s_k)|h_{k-1}s_k)\middle|h_{k-1}\right)$$

whence, by local linearity

$$= \underline{P}(X|h_{k-1}) + \underline{P}\left(\bigoplus_{s_k} E_{s_k} \underline{P}(Y(s_k)|h_{k-1}s_k)\middle|h_{k-1}\right). \tag{6.12}$$

This expression can clearly be maximised by choosing X to maximise $\underline{P}(X|h_{k-1})$ and choosing each $Y(s_k)$ to maximise $\underline{P}(Y(s_k)|h_{k-1}s_k)$. That is, if $X \in \operatorname{opt}(\mathcal{X}|h_{k-1})$ and $Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k)$ for all s_k, then Eq. (6.12) is maximal. This establishes the left implication in Eq. (6.11).

Next, suppose that

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle|h_{k-1}\right),$$

so there is no other gamble in $\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)$ with a higher lower prevision. Again considering Eq. (6.12), it is clear that $X \in \operatorname{opt}(\mathcal{X}|h_{k-1})$, since otherwise we could increase the lower prevision by instead using an optimal element of $\mathcal{X}$. Also, because $\underline{P}(E_{s_k}|h_{k-1}) > 0$, we see that $Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k)$, for otherwise we could increase the lower prevision by using an optimal element of $\mathcal{Y}(s_k)$. This establishes the right implication in Eq. (6.11).

"only if". Suppose that $\underline{P}(\cdot|h_{k-1})$ is not locally linear for some $h_{k-1} \in H_{k-1}$. Then there must be gambles X and Y on S_k, and an $\epsilon > 0$, such that

$$\underline{P}(X|h_{k-1}) + \underline{P}(Y|h_{k-1}) + \epsilon < \underline{P}(X+Y|h_{k-1}).$$

Let $\mathcal{X} = \{X, \underline{P}(X|h_{k-1}) + \epsilon\}$ and $\mathcal{Y}(s_k) = \{Y(s_k)\}$. Obviously, $\operatorname{opt}(\mathcal{X}|h_{k-1}) = \{\underline{P}(X|h_{k-1}) + \epsilon\}$, and $\operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}) = \{Y(s_k)\}$. However,

$$\begin{aligned}\operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k) \middle| h_{k-1}\right) &= \operatorname{opt}(\{X+Y, \underline{P}(X|h_{k-1}) + \epsilon + Y\}|h_{k-1}) \\ &= \{X+Y\}\end{aligned}$$

because

$$\begin{aligned}\underline{P}(\underline{P}(X|h_{k-1}) + \epsilon + Y|h_{k-1}) &= \underline{P}(X|h_{k-1}) + \epsilon + \underline{P}(Y|h_{k-1}) \\ &< \underline{P}(X+Y|h_{k-1}).\end{aligned}$$

So, locality fails whenever local linearity fails.

Next, we show that marginal extension must hold. Suppose it does not, and hence that

$$\underline{P}(Z|h_{k-1}) > \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}).$$

for some gamble Z and h_{k-1}.

Consider the following sets of gambles:

$$\begin{aligned}\mathcal{X} &= \{\underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}), \underline{P}(Z|h_{k-1}S_k)\} \\ \mathcal{Y}(s_k) &= \{Z(s_k,\cdot), \underline{P}(Z|h_{k-1}s_k)\} \text{ for all } s_k\end{aligned}$$

Let $X = \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$ and $Y(s_k) = \underline{P}(Z|h_{k-1}s_k)$. It is immediate that $\operatorname{opt}(\mathcal{X}|h_{k-1}) = \mathcal{X}$ and $\operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) = \mathcal{Y}(s_k)$. But as we saw in the proof of Proposition 6.32, the gamble $X + \bigoplus_{s_k} E_{s_k}Y(s_k)$ is not maximal. A Γ-maximin gamble is always maximal [79, 3.9.7], and therefore $X + \bigoplus_{s_k} E_{s_k}Y(s_k)$ is not Γ-maximin either, and so locality fails.

□

The locality conditions for Γ-maximin imply full linearity on all gambles on $S_0 \times \cdots \times S_{n-1}$, i.e. linearity on every gamble that does not involve the final state S_n. Of course, in cases where such strong form of linearity is satisfied, usually full linearity will actually be satisfied. In other words, one cannot really endorse locality for Γ-maximin and at the same time use imprecise probabilities.

Corollary 6.35. *If* Γ*-maximin satisfies locality, then for any* $1 \leq k < n$*, any gamble* Z *on* $S_k \times \dots S_{n-1}$*, and any* $h_{k-1} \in H_{k-1}$*,*

$$\underline{P}(Z|h_{k-1}) = \overline{P}(Z|h_{k-1}).$$

Proof. Simply apply local linearity and marginal extension repeatedly. Indeed, the case $k = n-1$ follows from local linearity. Suppose that we have already established the result for $k = m+1$, let us show that it also holds for $k = m$. For any gamble Z on $S_m \times \cdots \times S_{n-1}$, we have, by marginal extension

$$\underline{P}(Z|h_{m-1}) = \underline{P}(\underline{P}(Z|h_{m-1}S_m)|h_{m-1})$$

but, $\underline{P}(Z|h_{m-1}S_m)$ is a gamble on S_m, and hence, by local linearity,

$$= \overline{P}(\underline{P}(Z|h_{m-1}S_m)|h_{m-1})$$

but, by the induction hypothesis, $\underline{P}(\cdot|h_{m-1}s_m)$ is linear for all $s_m \in S_m$, and hence,

$$= \overline{P}(\overline{P}(Z|h_{m-1}S_m)|h_{m-1})$$

and hence, again by marginal extension,

$$= \overline{P}(Z|h_{m-1})$$

which establishes the desired result. □

However, a locally Γ-maximin policy is always locally maximal, and so if marginal extension holds it is globally maximal, by Proposition 6.32. So, using a locally Γ-maximin policy may be a reasonable choice, even though it is not always globally Γ-maximin.

In our example, betting on heads is optimal under local Γ-maximin whenever

$$\underline{P}(X_k^k(h_{k-1}, d_H)|h_{k-1}) \geq \underline{P}(X_k^k(h_{k-1}, d_T)|h_{k-1}).$$

By Eqs. (4.7) and (6.9), we obtain (ignoring Q)

$$n_H \quad \geq \quad n_T.$$

So, betting on heads is approximately locally optimal if $n_H \geq n_T$ and similarly on tails when $n_T \geq n_H$. However, by Proposition 6.34, for this also to be guaranteed to be a global Γ-maximin policy, $\underline{P}$ would be required to be a linear prevision (except on the last stage). Because the imprecise Dirichlet model starts with a vacuous lower prevision as prior, $\underline{P}$ does *not* satisfy the linearity condition, and hence, the policy we just found is not necessarily globally Γ-maximin. Nevertheless, the local Γ-maximin policy is still an interesting alternative for the reasons we highlighted earlier.

Interval Dominance

Interval dominance with respect to a coherent lower prevision $\underline{P}$ is:

$$\operatorname{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(\overline{P}(X|A) \geq \underline{P}(Y|A))\}.$$

Proposition 6.36. *Interval dominance with respect to a coherent lower prevision $\underline{P}$ satisfies locality on S_0, ..., S_n if and only if*

(i) $\underline{P}$ satisfies marginal extension with respect to S_0, ..., S_n, and

(ii) $\underline{P}(\cdot|h_{k-1})$ is locally linear, in the sense that for all $1 \leq k \leq n$, $h_{k-1} \in H_{k-1}$, and gambles X on S_k,

$$\underline{P}(X|h_{k-1}) = \overline{P}(X|h_{k-1})$$

Proof. "if". If $\underline{P}$ satisfies properties (i) and (ii), then it is easy to show that $\underline{P}(\cdot|h_{k-1})$ is linear for any $1 \leq k \leq n$ and all gambles on $S_k \times \cdots \times S_n$ (see the proof of Corollary 6.37). This implies that interval dominance and maximality coincide at every stage, and so by Proposition 6.32, interval dominance satisfies locality.

"only if". Suppose first that $\underline{P}(\cdot|h_{k-1})$ is not locally linear for some $1 \leq k < n$ and $h_{k-1} \in H_{k-1}$. Then there must be a gamble X on S_k and an $\epsilon > 0$ such that

$$\underline{P}(X|h_k) + \epsilon < \overline{P}(X|h_k).$$

Let $\mathcal{X} = \{0\}$ and $\mathcal{Y}(s_k) = \{X(s_k), X(s_k) + \epsilon\}$. Obviously, $\mathrm{opt}(\mathcal{X}|h_{k-1}) = \{0\}$, and $\mathrm{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) = \{X(s_k) + \epsilon\}$. However,

$$\begin{aligned}\mathrm{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle| h_{k-1}\right) &= \mathrm{opt}\left(\{X + \epsilon I_E \colon E \subseteq S_k\}\middle| h_{k-1}\right) \\ &= \{X + \epsilon I_E \colon E \subseteq S_k\}\end{aligned}$$

because

$$\underline{P}(X + \epsilon I_E|h_k) \leq \underline{P}(X + \epsilon|h_k) < \overline{P}(X|h_k)$$

for all $E \subseteq S_k$. We have shown that locality fails, or, in other words, local linearity for $1 \leq k < n$ is necessary for locality to hold for interval dominance (we will establish local linearity for $k = n$ further).

Next, suppose that local linearity holds for $1 \leq k < n$, but

$$\underline{P}(Z|h_{k-1}) > \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}).$$

for some gamble Z on $S_k \times \cdots \times S_n$ and h_{k-1} (indeed, if equality does not hold, we must have a strict inequality as given, by [79, 6.3.5(5), p. 296]).

Consider the following sets of gambles:

$$\begin{aligned}\mathcal{X} &= \{0\} \\ \mathcal{Y}(s_k) &= \{Z(s_k, \cdot), \underline{P}(Z|h_{k-1}s_k)\} \text{ for all } s_k\end{aligned}$$

Let $X = 0$ and $Y(s_k) = \underline{P}(Z|h_{k-1}s_k)$. These gambles are maximal in their respective sets so they must also be optimal with respect to interval dominance. We have

$$\overline{P}\left(X + \bigoplus_{s_k} E_{s_k}Y(s_k)\middle| h_{k-1}\right) = \overline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$$

and by local linearity

$$= \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1})$$

Now consider the gamble

$$X + \bigoplus_{s_k} E_k Z(s_k, \cdot) = Z.$$

We have

$$\begin{aligned}\underline{P}(Z|h_{k-1}) &> \underline{P}(\underline{P}(Z|h_{k-1}S_k)|h_{k-1}) \\ &= \overline{P}\left(X + \bigoplus_{s_k} E_{s_k} Y(s_k)|h_{k-1}\right),\end{aligned}$$

so we have found a gamble in $\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)$ that dominates $X + \bigoplus_{s_k} E_{s_k} Y(s_k)$ with respect to interval dominance, and so locality fails. In other words, if local linearity holds for $1 \leq k < n$ then marginal extension is necessary for locality to hold for interval dominance. From our result earlier in the proof, this implies that local linearity for $1 \leq k < n$ and marginal extension are necessary.

Next, suppose that local linearity for $1 \leq k < n$ and marginal extension hold, but local linearty for $k = n$ fails. Then, there must be an $h_{n-1} = h_{n-2}s'_{n-1} \in H_{n-1}$, a gamble Z on S_n, and an $\epsilon > 0$, such that $\underline{P}(Z|h_{n-1}) + \epsilon < \overline{P}(Z|h_{n-1})$. Let $\mathcal{Y}(s_{n-1}) = \{0\}$ for all s_{n-1} except $\mathcal{Y}(s'_{n-1}) = \{Z\}$. Let $\mathcal{X} = \{0, \epsilon P(E_{s'_{n-1}}|h_{n-2})\}$.

Then by marginal extension and local linearity for stage $n-1$,

$$\begin{aligned}\overline{P}\left(0 + E_{s'_{n-1}} Z|h_{n-2}\right) &= P(E_{s'_{n-1}}\overline{P}(Z|h_{n-1})|h_{n-2}) \\ &= P(E_{s'_{n-1}}|h_{n-2})\overline{P}(Z|h_{n-1}) \\ &\geq P(E_{s'_{n-1}}|h_{n-2})(\epsilon + \underline{P}(Z|h_{n-1})),\end{aligned}$$

but

$$\begin{aligned}\underline{P}\left(\epsilon P(E_{s'_{n-1}}|h_{n-2}) + E_{s'_{n-1}} Z|h_{n-2}\right) &= P(\epsilon P(E_{s'_{n-1}}|h_{n-2}) + E_{s'_{n-1}}\underline{P}(Z|h_{n-1})|h_{n-2}) \\ &= P(E_{s'_{n-1}}|h_{n-2})(\epsilon + \underline{P}(Z|h_{n-1})).\end{aligned}$$

Therefore the gamble $0 + E_{s'_{n-1}} Z$ is optimal with respect to interval dominance. But clearly, $\mathrm{opt}(\mathcal{X}|h_{n-2}) = \{\epsilon P(E_{s'_{n-1}}|h_{n-2})\}$ and so locality fails. In other words, if local linearity holds for $1 \leq k < n$ and marginal extension holds, then local linearity for $k = n$ is necessary for locality to hold for interval dominance.

So, marginal extension and local linearity for $1 \leq k \leq n$ are necessary for locality. The proof of Corollary 6.37 shows that these conditions imply full linearity of $\underline{P}$. □

Interval dominance requires even stronger conditions than Γ-maximin. Indeed, we have the following corollary.

Corollary 6.37. *If interval dominance satisfies locality on $S_0, \dots, S_n$, then, for all $1 \leq k \leq n$, all gambles X on F_k, and all $h_{k-1} \in H_{k-1}$,*

$$\underline{P}(X|h_{k-1}) = \overline{P}(X|h_{k-1}).$$

Proof. Almost identical to the proof of Corollary 6.35. □

So, for interval dominance to satisfy locality, $\underline{P}$ must essentially correspond to a coherent prevision E_P for some full conditional probability P: you cannot be imprecise, and, at the same time, endorse locality for interval dominance.

6.5.2 Dynamic Programming

De Cooman and Troffaes [13, §3.2–3.5] investigate whether dynamic programming works for maximality, E-admissibility, Γ-maximin, and interval dominance. The first two satisfy all properties, and the latter two fail Backward Addition. Γ-maximin and interval dominance fail because of the non-additivity of a coherent lower prevision.

For subtree perfectness, none of the choice functions satisfies all the necessary properties. Intersection requires a total preorder, and, of the four, only Γ-maximin is. Since Γ-maximin fails Backward Addition, it automatically fails Addition. These results mirror those for standard decision trees [31]: only maximality and E-admissibility allow backward induction, and nothing is subtree perfect.

As mentioned by de Cooman and Troffaes, Γ-maximin could satisfy Addition for certain lower previsions. Suppose that Ω is a product of possibility spaces $\Omega_1, \dots, \Omega_m$, and the gambles on the ith decision arc in any path is a gamble on Ω_i. If the overall lower prevision $\underline{P}$ is a suitable independent product of lower previsions $\underline{P}_i$ on the Ω_i, then for any gambles X_1 on Ω_1, X_2 on Ω_2, and so on, it will hold that

$$\underline{P}\left(\sum_{i=1}^{m} X_I\right) = \sum_{i=1}^{m} \underline{P}_i(X_I).$$

We refer to [13, §3.4] for more details and references. Note that de Cooman and Troffaes mention only that, under such circumstances, dynamic programming will work

	Property											
	1	2	3	7	8	9	10	11	12	18	20	21
E-admissibility	✓		✓		✓	✓	✓	✓	✓		✓	✓
Maximality	✓		✓		?	✓	✓	✓	✓		✓	✓
Γ-maximin	✓	✓				✓	✓	✓				
Interval Dominance	✓					✓	✓	✓				

Table 6.6: Properties of various choice functions.

	Property									
	1	2	3	7	8	9	10	11	12	18
E-admissibility	✓		✓	✓	✓	✓	✓	✓	✓	✓
Maximality	✓		✓	✓	✓	✓	✓	✓	✓	✓
Γ-maximin	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Interval Dominance	✓					✓	✓	✓		

Table 6.7: Properties of various choice functions under marginal extension.

for both Γ-maximin and interval dominance. We can add that subtree perfectness will also hold for Γ-maximin (but not interval dominance).

6.6 Summary

We have shown that backward induction works only for E-admissibility and maximality, that subtree perfectness fails for every choice function but is satisfied under marginal extension by Γ-maximin, that locality fails for every choice function but is satisfied under marginal extension for maximality and E-admissibility, and that dynamic programming works for maximality and E-admissibility. Summaries of the important properties satisfied by each choice function are given in Tables 6.6 (the case without marginal extensions) and 6.7 (the case with marginal extension).

Chapter 7

Conclusion

7.1 Overview

Our goal has been to identify interesting properties of particular forms of solution to decision trees, and to find simple necessary and sufficient conditions for these properties to hold, while making as few assumptions as possible about the criteria the subject use to make their decision. It is much easier to check these conditions than check whether the original properties hold, making it more convenient to determine whether a newly-proposed solution would satisfy the properties.

We have primarily investigated normal form solutions [56, § 1.3] to decision trees under act-state independence (meaning that the true state of nature does not depend on the actions the subject takes). Normal form solutions involve making a decision for all eventualities immediately, and then implementing those decisions. In simple cases, one may be able to identify a uniquely optimal decision for every decision point, thus leading to a single strategy to follow. In more complicated situations, this may not be feasible, and so we defined a normal form solution to be a set of such strategies. The subject may then pick any of these strategies and implement it.

Upon choosing and implementing a strategy, the subject no longer has any choices to make, and the result of the strategy is completely determined by nature. Therefore each strategy corresponds to an uncertain reward (called a *gamble* [14]), and identifying a normal form solution corresponds to choosing a set of acceptable gambles. The language of *choice functions* [66] was used to described such choice: a choice function maps a set of options to a non-empty (optimal) subset. Many popular uncertainty and preference models involve choice functions on gambles, so this is a useful way of representing the problem.

Given a choice function on gambles, a natural normal form solution presents itself. One can find the set of all strategies associated with a decision tree, and hence the set of all associated gambles. Next, the choice function can be invoked to find the optimal subset of these gambles. Finally, the set of all strategies associated with these gambles can be found. This provides an optimal set of strategies, from which the subject can pick one to follow.

We principally considered two possible properties of this solution. The first, subtree perfectness (inspired by Selten's concept of subgame perfectness for multi-agent games [65]), links the solution of a decision tree to the solutions of any subtree. Informally, a solution satisfies subtree perfectness if, for every subtree of the decision tree, the solution of the decision tree restricted to that subtree is exactly the solution of that subtree. We showed that subtree perfectness only holds in general for very few choice functions: it requires that the choice function corresponds to a total preorder, and also that it satisfies a condition very closely related to the famous Independence Axiom [78]. Thus, it is not possible to stray far from expected utility without violating subtree perfectness.

The second property, backward induction (following the method of Kikuti et al. [38]), allows a decision tree to be solved by folding back from right to left, eliminating arcs that are non-optimal in subtrees. If the choice function corresponds to a total preorder, backward induction is quite similar to subtree perfectness. We have shown, however, that a total preorder is not necessary for backward induction to work: it can be relaxed to a condition called path independence (a stronger form of independence of irrelevant alternatives). Path independence is satisfied for a much wider variety of choice functions, such as those corresponding to a partial order. Indeed, a choice function can satisfy path independence without even being an ordering at all.

We then applied these results to the theory of coherent lower previsions [79], a generalization of probability theory for cases where a single probability measure cannot be identified. Four popular choice functions in this theory were examined (maximality, E-admissibility, interval dominance, and Γ-maximin), providing several new and important results. Normal form backward induction works for maximality and E-admissibility, the two most logically motivated of the choice functions. Interval dominance, although rightly criticised for being too conservative, can be used to aid calculation of maximality or E-admissibility when using backward induction, potentially saving time. E-admissibility is subtree perfect for trees containing only one decision node per branch, for example trees representing a decision after observing an experiment (see Augustin [6] for more details of solving such problems with coherent lower previsions). Although such trees are very basic, they are also important for a multitude of statistical applications, and it is reassuring to know that E-admissibility can be used for those without having to worry about whether subtree perfectness is violated or whether such a violation is unacceptable.

Finally, Γ-maximin was shown to be subtree perfect whenever a particular property of coherent lower previsions called *marginal extension* holds over all relevant partitions. Γ-maximin is popular because of its simplicity, its conservative nature, and its tendency to choose a single decision. It is criticised for failing to take into account the imprecision inherent in the model, for sometimes choosing gambles that are not E-admissible, and for being wildly subtree imperfect in sequential decision problems. Our results show that one of these criticisms can be avoided, although it requires specification of beliefs in a particular way.

7.2 Discussion

Normal form solutions involve making all decisions in advance; effectively turning the sequential problem into a static one. The standard normal form solution induced by a choice function defined in Section 1.2.5 can be seen as a transformation of the original decision tree to a tree with a single decision node, suggesting that normal form solutions correspond to "flattening" of a tree. We have instead treated normal form solutions as sets of subtrees. This is a more general representation than using a static decision tree; for instance, it allows modelling of normal form solutions that do not respect strategic equivalence. This definition departs somewhat from the commonly understood use of the term. We believe this departure is more expressive and convenient (for example, being able to define normal form backward induction).

The framework of choice functions proved to be a powerful tool for investigating normal form solutions, being capable of describing a variety of different uncertainty and preference models without having to make too many assumptions. Having said this, there can be problems or models that cannot be generalized to arbitrary choice functions on gambles. In particular, the language of choice functions and gambles is not rich enough to deal with problems of act-state dependence except in special cases (trying to adapt the method to work with influence diagrams, for example, does not seem to lead anywhere useful). This restriction is particularly noticeable in Chapter 4, where the problems we were able to treat without losing act-state independence are very basic and not widely applicable.

Our purpose was to investigate different properties of solutions to decision problems: primarily backward induction and subtree perfectness. One way to do this is to investigate such properties for specific choice functions (see for instance [41, 50, 6, 64]), but this is clearly an inefficient approach. Another possibility, one often followed, is to first argue for some restrictions on the choice function (Total Preordering is a popular choice) and then determine what choice functions behave acceptably under these restrictions. We avoided this route because it risks overlooking some important distinctions, such as only requiring Backward Mixture rather than Mixture for backward induction.

If one wants to argue that certain things are the requirements of "rational behaviour", then the above approach makes sense. If one has decided that a certain property is necessary to be rational, it seems sensible to assume that property in all that follows. There seems little point in showing that Path Independence is necessary for backward induction if Total Preordering is already required. Similarly, it is not unreasonable to argue that normal form and extensive form equivalence should hold; if so, there is no need to ask what happens when equivalence does not hold.

Our goal was different. We wanted to identify particular concepts, such as backward induction, and specify exactly what restrictions on the choice function are equivalent. For the most part, we do not try to argue that a particular concept must hold for the solution to be sensible (although all the concepts are at least intuitively appealing—there has to be some reason to investigate them!), nor do we restrict ourselves to only a subset of possible choice functions. Observe that even the apparently obvious conditioning properties are not initially assumed, nor is the fundamental equality $\mathrm{opt}(\mathrm{opt}(\mathcal{X})) = \mathrm{opt}(\mathcal{X})$.[1]

Another unusual approach we make is the avoidance of probabilities and utilities. This might seem a very important step, but, at least for the case of subtree perfectness, the conclusion effectively agrees with Hammond [24], who has some chance nodes with probabilities and some without. So it is not clear if we really gain anything concrete by this distinction.

Chapter 2 also shows the interesting result that subtree perfectness is not, in general, sufficient for the equivalence of normal form and extensive form, outside of the choice function framework. Intuitively one might expect that "the solution in a subtree does not depend on the full tree in which it is embedded" is exactly the condition that would lead to equivalence, but, as we saw, even with this condition there are still more normal form solutions than extensive form ones. The startling discovery is that, when using the canonical normal form solution, subtree perfectness and equivalence of forms coincide exactly. This is because the necessary properties for choice functions combine to form properties that are sufficient for strong subtree perfectness (p. 53). This may also suggest that strong subtree perfectness is the core property to be required.

Having confirmed that, as expected, very few practical choice functions will satisfy subtree perfectness, in Chapter 3 we turned attention to the less obvious matter of backward induction, proposing a method to find *normal form* solutions via back-

[1] "Fundamental" because, without it, the choice function does not seem to have the interpretation from Section 1.2.2: our interpretation of a choice function is that, confronted by $\mathcal{X}$, the subject would be willing to choose any of $\mathrm{opt}(\mathcal{X})$ but no other gamble. That is, the acceptable subset of $\mathcal{X}$ is $\mathrm{opt}(\mathcal{X})$. If $\mathrm{opt}(\mathrm{opt}(\mathcal{X})) \subset \mathrm{opt}(\mathcal{X})$ then $\mathrm{opt}(\mathcal{X})$ can hardly be said to be the acceptable subset of $\mathcal{X}$.

ward induction (rather than the usual extensive form ones). Presenting backward induction as a normal form method allows an easy definition for an arbitrary choice function, whereas we saw in Section 3.2 that defining backward induction for something more complicated than a total preorder is non-trivial. Although in Section 3.3 we presented a possible argument for backward induction but not subtree perfectness to be considered a requirement for rationality, the main motivation for this chapter was for more efficient computation of normal form solutions.

Although subtree perfectness holds for very few choice functions, backward induction is more fruitful for those wishing to avoid expected utility theory. It will not help those who consider the Independence Axiom as misguided, since Backward Mixture will fail for the typical examples where the Independence Axiom fails (see Machina [50] for a review). It is more appealing for those who consider Total Preordering too restrictive. Path Independence is a much weaker condition (even allowing choice functions that do not correspond to any type of ordering), but is relatively easy to justify. Simple explanations of why Insensitivity To Omission and Preservation Under Addition are appealing can be found in Luce and Raiffa [46, §13.3]. Although arguments abound as to why Total Preordering is unacceptable [46, 44, 51, 79], we are not aware of significant normative objections to Path Independence.

Also in Chapter 3 is Theorem 3.32, allowing calculation of $\mathrm{norm}_{\mathrm{opt}}$ using different choice functions. Although rather distinct from the general theme of the thesis, this is a useful example of the sorts of things that are possible with choice functions that are apparently not so useful. As observed in Chapter 6, the interval dominance choice function for coherent lower previsions only satisfies our main sequential conditions in trivial cases, but according to this theorem it could be useful when trying to solve difficult problems using maximality. We expect that there will be more results like this that allow the use of the lesser choice function to help with calculations.

Chapters 4 and 5 extends $\mathrm{norm}_{\mathrm{opt}}$ to some related decision problems, that feature cumulative rewards. Not only does this provide corresponding results for these special processes, it also demonstrates that, given a new type of decision problem, the same sort of approach can often work well. Once one understands the concepts for one type of problem, extending them to other types can be intuitive.

Chapter 6 demonstrated application of the main theorems to four major choice functions from the theory of coherent lower previsions: maximality, E-admissibility, interval dominance, and Γ-maximin. This not only demonstrates practical value, but also how they can be easily adapted to provide more information about specific uncertainty and preference models. For instance, it was already known (if not stated explicitly) in the literature that all the imprecise probability choice functions failed subtree perfectness, but we saw it was also easy to find special cases in which subtree perfectness can hold for some of the choice functions. The salient point is that, although the theorems deal with all possible consistent decision trees, it is possible to use analogous ideas to deal with restricted sets of decision trees: the theory remains broadly the same in such cases.

7.3 Further Work

7.3.1 Subtree Perfectness

It seems that there is little more work to be done on normal form subtree perfectness in the general case: we could try to determine exactly how small the set of subtree perfect choice functions is, but it is difficult to find anything that does not correspond to expected utility or lexicographic expected utility [26, 22]. More progress may be possible when looking at more restricted cases, such as we did in Section 6.4. For instance simplifying the unwieldy formulation of Weak Multiple Mixture would be useful, as would a more rigorous treatment of some results in that section.

Subtree perfect extensive form solutions have more potential. In Section 3.2 we introduced a few subtree perfect extensive form operators based on choice functions, but only looked in detail at when norm_{opt} is subtree perfect, and all the different operators coincide. Investigating their differences when norm_{opt} is not subtree perfect could be enlightening. In particular, Seidenfeld's extensive form operator looks to yield fairly sensible results for E-admissibility, and is probably quite efficient to compute (when compared with ext_{opt}). There are possibilities that some techniques, based on clever choices of events for future choices, could eliminate the sorts of unappealing behaviour seen in Fig. 3.2.

7.3.2 Backward Induction

For backward induction, we could try to find special tree structures that allow unruly choice functions to work, but this would probably force the decision tree to be too trivial for backward induction to have much computational use. More computational tricks like Theorem 3.32 would be useful; for instance, is there any quick way to deduce which choice function would be worth applying at a particular node in Theorem 3.32? Presumably, this question would be have to be answered on a case-by-case basis rather than with a general method. Also, the way that $\mathrm{back}_{\mathrm{opt}}$ is actually defined is not particularly useful in practice: observe in Section 6.3 that we did not need to compare gambles we had already compared when moving back. A real algorithm would need to have such tricks built in. In particular, should Multiple Mixture hold, then chance nodes become much more convenient to deal with.

7.3.3 Act-State Independence

The work in Chapter 4 is limited to act-state dependent problems, which are very restrictive for the particular type of problem under consideration. To study problems that are much more widely applicable, we need the transitions to depend on the actions taken, that is, we need to introduce act-state dependence. As mentioned in Section 1.2.2, this is not easy. Classically we would just make the decisions affect the transition probabilities, but this relies on the nature of maximizing expected utility: to compare two gambles, one calculates a value for the first, and a value for the second, and then one checks which is higher. In the case of act-state independence, this is equivalent to using a choice function on gambles, but in the act-state dependent case, it is more complicated. Recall that gambles are functions from the possibility space to the reward set. For act-state dependence, we need a type of "generalized" gamble that also contains information about the implication of choosing that gamble on the probabilities of the events. So, a generalized gamble for expected utility would be a list of the form (X, p) where X is a map from Ω to

$\mathcal{R}$ and p is a probability mass function. The expected utility choice function would then order sets of gambles according to $P((X, p)) = P_p(X)$.

So far, our choice functions have encoded both the uncertainty model and the preference model being used. Generalizing the above approach for dealing with act-state dependence would leave the choice function representing only the preference model given the uncertainty model, with the uncertainty model itself being encoded as part of a generalized gamble. So instead of choice functions on gambles, we would be looking at choice functions on generalized gambles. What effects this will have on the theory is unclear.

One clear effect of this generalization would be found when looking at coherent lower previsions. Consider how the maximality partial order compares gambles: by finding the lower prevision of their differences. But if each gamble corresponds to a *different* lower prevision, this approach cannot be used. It would seem that, unless the lower previsions are the same, then in such a case interval dominance would have to be used instead [25, 74]. This seems very restrictive, especially since if X is preferred to Y under $\underline{P}$ according to maximality but not interval dominance, and $\underline{Q}$ is only different from $\underline{P}$ by a tiny degree, one would hope that $(X, \underline{P})$ would be preferred to $(Y, \underline{Q})$, but apparently we cannot say this. Treatment of maximality and E-admissibility under act-state dependence is an open problem in imprecise decision theory.

There is another way of dealing with act-state independence: try to adapt the model so that it disappears. This generally will lead to much larger possibility spaces and will make specifying the uncertainty model more difficult. Consider the simplest non-trivial situation, with $\Omega = \{\omega_1, \omega_2\}$, $\mathcal{X} = \{X, Y\}$. Suppose that, if X is chosen, $P(\omega_1) = 0.4$, and if Y is chosen, $P(\omega_1) = 0.5$, so we have act-state dependence. Now, consider a different possibility space $\Omega^* = \{\omega_{11}, \omega_{12}, \omega_{21}, \omega_{22}\}$, where outcome ω_{xy} represents the outcome "if X is chosen then ω_x obtains (from the original Ω), and if Y is chosen then ω_y obtains". The gambles X and Y are updated according to $X(\omega_{xy}) = X(\omega_x)$ and $Y(\omega_{xy}) = Y(\omega_y)$.

Act-state dependence has been eliminated, but at what cost? Clearly, the size of the possibility space has increased. In general, if $|\Omega| = a$ and $|\mathcal{X}| = b$, then $|\Omega^*| = a^b$.

For large problems this could become a burden. Also, consider the uncertainty model for Ω^* consistent with that for Ω. We know that $P(\{\omega_{11}, \omega_{12}\}) = 0.4$ and $P(\{\omega_{12}, \omega_{22}\}) = 0.5$, but nothing more. This leads to $\underline{P}(\omega_{11}) = 0$, $\overline{P}(\omega_{11}) = 0.4$, and so on; we have a lower prevision on Ω^*. This is not really a problem here, since there is no reason to use the lower prevision when the precise probabilities are available, but it could cause problems in attempts to generalize the method, for instance when the original act-state dependent uncertainty models are themselves lower previsions.

Given a set of gambles with act-state dependent coherent lower previsions, we could move to a single act-state independent coherent lower prevision by extending the method above, but this may not always be a good idea. In the precise example above, we took the most conservative lower prevision consistent with the original probabilities. The properties of the gambles and of maximizing expected utility assure us, however, that we could have taken any consistent lower prevision. If the initial problem contained lower probabilities instead, choosing a less conservative lower prevision for Ω^* could lead to a different optimal set of gambles.

This could be exploited in cases where we actually think we know more about the problem than can be contained in the separate act-state dependent lower previsions. Suppose for instance that ω_1 represents a machine's working correctly in a certain time period, and that the act inducing gamble Y involves performing maintenance on the machine, whereas X involves no maintenance. Assume the probabilities from above, and that maintenance cannot accidentally degrade the machine (a bold assumption, admittedly). Then one might argue there is only one sensible uncertainty model for Ω^*: $P(\omega_{11}) = 0.4$, $P(\omega_{21}) = 0.1$, $P(\omega_{22} = 0.5)$.

7.3.4 Changes To Decision Trees

Returning to normal form decision making in general, problems will arise when circumstances change during implementation of a normal form decision. For instance, new options may have become available, or more information may become known. The originally chosen normal form decision may not be optimal in the new problem. If subtree perfectness holds, there is nothing to worry about: the subject can draw a new decision tree with root node being her current node, solve this, and pick

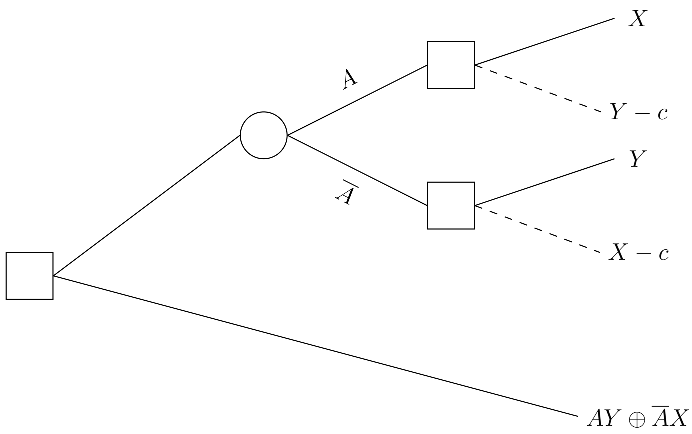

Figure 7.1: The problem of hidden options.

a new optimal strategy. If subtree perfectness does not hold, this might be more questionable, because this could result in choosing a strategy that was not originally optimal.

It could be argued that we should not worry about whether something was originally optimal any more, since things have changed and so the original problem is now irrelevant, but this seems to be part of a more general argument in favour of subtree perfect extensive form solutions. For the purpose of this discussion we assume that originally non-optimal choices should be avoided where possible. The problem becomes more complicated after observing that as well as the original solution of the global problem and the new solution of the (altered) local problem, there is also the new solution of the altered global problem to worry about. It may seem that this third solution need not be considered, since the subject never had the opportunity to pick a strategy from the altered global problem. The following example, adapted from Machina [50, Fig. 12, p. 1657], casts some doubt on this.

Suppose that the subject is confronted by the tree in Fig. 7.1, where the dashed lines represent arcs that are not revealed until their node is reached. Also suppose that $AX \oplus \overline{A}Y$ is preferred to $AY \oplus \overline{A}X$, but $Y - c$ is preferred to X given A and $X - c$ is preferred to Y given $\overline{A}$ (so the choice function fails Multiple Backward Mixture). The subject will initially choose the upper arc. Suppose A occurs, and so $Y - c$ becomes available. If the subject considers only the new local problem (in which $Y - c$ is optimal) and the old global problem (in which $Y - c$ didn't exist) then she would choose Y. Similarly, should $\overline{A}$ occur then $X - c$ would be chosen. This corresponds to choosing the gamble $A(Y - c) \oplus \overline{A}X$, an inferior version of a gamble she initially rejected. Of course, if some options are hidden then solutions may behave poorly, but it is worrying if hidden options can certainly lead the subject to choose the worst possible strategy.

If backward induction holds, there are fewer problems like this. The following procedure should provide reasonable results if the subject encounters something unexpected. If the originally chosen strategy is still locally optimal, continue to follow it. If it is not, search for a strategy that was originally optimal and still locally optimal, and follow that. If no such strategy exists, choose a locally optimal strategy that did not exist before. If no such strategy exists, then choose any locally optimal strategy. Only in the last case will we select a strategy that was initially non-optimal globally. The backward induction conditions make this last case only appear in circumstances where it is reasonable, such as if the conditioning event is not what was expected, or if all initially optimal arcs do not actually exist. Exactly how this procedure performs would require investigation, particularly when many different changes are made at once. At the very least it should usually ensure that the subject avoids following a strategy that is worse than one already rejected, unless there really is no better option (this might not hold if previous-optimal arcs turn out not to exist; in such circumstances it may be justified to solve the initial problem with the non-existent arcs removed).

7.4 Concluding Remarks

In conclusion, we have given a fairly general framework for analysing particular properties of solutions to sequential decision problems, principally using choice functions on gambles and normal form solutions. We have found simple conditions for various behaviours, which are easy to check if one has a new choice function to test. We have shown that the methods can often be extended to different types of problem without much difficulty, and also can be directed at specific cases of a more general problem to find special behaviour in those restricted instances. However, we have found a number of extensions that do not follow easily, such as act-state dependence and changes to the problem.

Bibliography

[1] M. Allais. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école Américaine. *Econometrica*, (21):503–546, 1953.

[2] E. Altman. Applications of Markov decision processes in communication networks: a survey. Technical report, INRIA, 2000.

[3] G. E. M. Anscombe. Modern moral philosophy. *Philosophy*, 33(124):1–19, Jan 1958.

[4] K. Arrow. Exposition of the theory of choice under conditions of uncertainty. In C. McGuire and R. Radner, editors, *Decision and Organization*, pages 19–55.

[5] K. J. Arrow. Rational choice functions and orderings. *Economica*, 26(102):121–127, May 1959.

[6] T. Augustin. On decision making under ambiguous prior and sampling information. In G. de Cooman, T.L. Fine, S. Moral, and T. Seidenfeld, editors, *ISIPTA 01: Proceedings of the Second International Symposium on Imprecise Probabilities*, 2001.

[7] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, 1957.

[8] Y. Ben-Haim. *Info-Gap Decision Theory: Decisions Under Severe Uncertainty.* Academic Press, 2001.

[9] J. O. Berger. *Statistical decision theory and Bayesian analysis.* Springer-Verlag, New York, second edition, 1985.

[10] G. Boole. *An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities.* Walton and Maberly, London, 1854.

[11] R. T. Clemen and T. Reilly. *Making Hard Decisions.* Duxbury, 2001.

[12] Marquis de Condorcet. *Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.* L'Imprimerie Royale, Paris, 1785.

[13] G. De Cooman and M.C.M. Troffaes. Dynamic programming for deterministic discrete-time systems with uncertain gain. *International Journal of Approximate Reasoning*, 39(2-3):257–278, Jun 2005.

[14] B. de Finetti. *Theory of Probability: A Critical Introductory Treatment.* Wiley, New York, 1974–5. Two volumes.

[15] T. Dean, L. P. Kaelbling, J. Kirman, and A. Nicholson. Planning under time constraints in stochastic domains. *Artificial Intelligence*, 76:35–74, 1995.

[16] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Statist.*, 38:325–339, 1967.

[17] L. E. Dubins. Finitely additive conditional probabilities, conglomerability and disintegrations. *The Annals of Probability*, 3(1):89–99, Feb 1975.

[18] D. Ellsberg. Risk, ambiguity, and the savage axioms. *Quarterly Journal of Economics*, 4(75):643–669, 1961.

[19] T. Fabian, J. L. Fisher, M. W. Sasieni, and A. Yardeni. Purchasing raw material on a fluctuating market. *Operations Research*, 7(1):107–122, Jan.-Feb. 1959.

[20] S. Faucheux and G. Froger. Decision-making under environmental uncertainty. *Ecological Economics*, 15(1):29–42, 1995.

[21] D. Ferguson and A. Stentz. Focussed processing of MDPs for path planning. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, 2004.

[22] P. C. Fishburn. A study of lexicographic expected utility. *Management Science*, 17(11):672–678, 1971.

[23] P. Hammond. Changing tastes and coherent dynamic choice. *The Review of Economic Studies*, 43(1):pp. 159–173, 1976.

[24] P. Hammond. Consequentialist foundations for expected utility. *Theory and Decision*, 25(1):25–78, Jul 1988.

[25] D. Harmanec. Generalizing Markov decision processes to imprecise probabilities. *Journal of Statistical Planning and Inference*, 105(1):199–213, Jun 2002.

[26] M. Hausner. Multidimensional utilities. In R. M. Thrall, C. H Coombs, and R. L. Davies, editors, *Decision Processes*, chapter 12, pages 167–180. John Wiley, 1954.

[27] G. Hazen. Does rolling back decision trees really require the independence axiom? *Management Science*, 33(6):807–809, Jun 1987.

[28] J. D. Hey. The economics of optimism and pessimism: A definition and some applications. *Kyklos*, 37(2):181–205, 1984.

[29] Chew Soo Hong. A generalization of the quasilinear mean with applications to the measurement of income inequality and decision theory resolving the allais paradox. *Econometrica*, 51(4):1065–1092, 1983.

[30] N. Huntley and M. C. M. Troffaes. An efficient normal form solution to decision trees with lower previsions. In Didier Dubois, M. Asunción Lubiano, Henri Prade, María Ángeles Gil, Przemyslaw Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, Advances in Soft Computing, pages 419–426. Springer, Sep 2008.

[31] N. Huntley and M. C. M. Troffaes. Characterizing factuality in normal form sequential decision making. In Thomas Augustin, Frank P. A. Coolen, Serafin Moral, and Matthias C. M. Troffaes, editors, *ISIPTA'09: Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications*, pages 239–248, 2009.

[32] J. Jaffray. Rational decision making with imprecise probabilities. In *1st International Symposium on Imprecise Probabilities and Their Applications*, 1999.

[33] A. Jayakumar and S. Asgarpoor. Maintenance optimization of equipment by linear programming. *Probability in the Engineering and Informational Sciences*, 20:186–193, 2006.

[34] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, (47):263–291, 1979.

[35] U. S. Karmarkar. Subjectively weighted utility: A descriptive extension of the expected utility model. *Organizational Behavior and Human Performance*, 21(1):61–72, 1978.

[36] E. Karni. *Decision Making Under Uncertainty—The Case of State Dependent Preferences*. Harvard University Press, 1985.

[37] G. A. Kiker, T. S. Bridges, A. Varghese, T. P. Seager, and I. Linkov. Application of multicriteria decision analysis in environmental decision making. *Integrated Environmental Assessment and Management*, 1(2):95–108, 2005.

[38] D. Kikuti, F. Cozman, and C. P. de Campos. Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities. In R. Brafman and U. Junker, editors, *IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, pages 118–123, 2005.

[39] P. Kolesar. A Markovian model for hospital admission scheduling. *Management Science*, 16(6):B384–B396, February 1970.

[40] H.E. Kyburg. Rational belief. *Behavioral and Brain Sciences*, 8(2), 1983.

[41] I. H. LaValle and P. C. Fishburn. Equivalent decision trees and their associated strategy sets. *Theory and Decision*, (23):37–63, 1987.

[42] I. H. LaValle and K. Wapman. Rolling back decision trees requires the independence axiom! *Management Science*, 32(3):382–385, Mar 1986.

[43] I. Levi. On indeterminate probabilities. *Philosophy of Science*, 71(13):391–418, Jul 1974.

[44] I. Levi. *The Enterprise of Knowledge*. MIT Press, London, 1980.

[45] D. V. Lindley. *Making Decisions*. Wiley, London, 2nd edition, 1985.

[46] R.D. Luce and H. Raiffa. *Games and Decisions: introduction and critical survery*. Wiley, 1957.

[47] D. G. Luenberger. *Introduction to Dynamic Systems*. Wiley, 1979.

[48] L. B. Lusted. Decision-making studies in patient management. *New England Journal of Medicine*, 284(8):416–424, 1971.

[49] M. J. Machina. "Expected utility" analysis without the independence axiom. *Econometrica*, 50(2):277–323, Mar 1982.

[50] M. J. Machina. Dynamic consistency and non-expected utility models of choice under uncertainty. *Journal of Economic Literature*, 27(1622-1688), 1989.

[51] E. F. McClennen. *Rationality and Dynamic Choice: Foundational Explorations*. Cambridge University Press, 1990.

[52] E. Miranda. A survey of the theory of coherent lower previsions. *International Journal of Approximate Reasoning*, 48(2):628–658, 2008.

[53] C.R. Plott. Path independence, rationality, and social choice. *Econometrica*, 41(6):1075–1091, Nov 1973.

[54] J. Quiggin. A theory of anticipated utility. *Journal of Economic Behavior & Organization*, 3(4):323–343, 1982.

[55] R. Radner and J. Marschak. Note on some proposed decision criteria. In R. M. Thrall, C. H Coombs, and R. L. Davies, editors, *Decision Processes*, pages 61–68. John Wiley, 1954.

[56] H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard University Press, 1961.

[57] F. P. Ramsey. Truth and probability. In R. B. Braithwaite, editor, *Foundations of Mathematics and other Logical Essays*, chapter VII, pages 156–198. Routledge and Kegan Paul, London, 1931. Published posthumously.

[58] P. Ray. Independence of irrelevant alternatives. *Econometrica*, 41(5):987–991, Sep 1973.

[59] J. Rust. When is it optimal to kill off the market for used durable goods? *Econometrica*, 54(1):65–86, 1986.

[60] J. K. Satia and R. E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, May-June 1973.

[61] L. J. Savage. *The Foundations of Statistics*. Dover, New York, 1972. Second revised edition.

[62] M. J. Schervish, T. Seidenfeld, and J. B. Kadane. State-dependent utilities. *Journal of the American Statistical Association*, 85(411):840–847, Sep 1990.

[63] T. Seidenfeld. Decision theory without 'independence' or without 'ordering': What is the difference? *Economics and Philosophy*, 4:267–290, 1988.

[64] T. Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: Γ-maximin versus E-admissibility. *Synthese*, 140:69–88, 2004.

[65] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4(1):25–55, Mar 1975.

[66] A. K. Sen. Social choice theory: A re-examination. *Econometrica*, 45(1):53–89, 1977.

[67] R. D Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–822, Nov–Dec 1986.

[68] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[69] G. Shafer and V. Vovk. *Probability and Finance: It's Only a Game!* Wiley, 2001.

[70] H. A. Simon. Theories of decision-making in economics and behavioral science. *The American Economic Review*, 49(3):253–283, 1959.

[71] C. A. B. Smith. Consistency in statistical inference and decision. *Journal of the Royal Statistical Society*, B(23):1–37, 1961.

[72] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998. http://www.cs.ualberta.ca/∼sutton/book/the-book.html.

[73] J. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):365–379, 1990.

[74] M. C. M. Troffaes. Learning and optimal control of imprecise Markov decision processes by dynamic programming using the imprecise Dirichlet model. In M. Lopz-Daz, M. Gil, P. Grzegorzewski, O. Hyrniewicz, and J. Lawry, editors, *Soft Methodology and Random Information Systems*, pages 141–148. Springer, 2004.

[75] M. C. M. Troffaes. Finite approximations to coherent choice. *International Journal of Approximate Reasoning*, 50(4):655–665, April 2009.

[76] M. C. M. Troffaes, N. Huntley, and R. Shirota Filho. Sequential decision processes under act-state independence with arbitrary choice functions. In E. Huellermeier, R. Kruse, and F. Hoffmann, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 98–107. Springer, 2010.

[77] M.C.M. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45:17–29, 2007.

[78] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, Princeton, third edition, 1953.

[79] P. Walley. *Statistical Reasoning with Imprecise Probabilities.* Chapman and Hall, London, 1991.

[80] P. Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):3–57, 1996.

[81] P. M. Williams. Notes on conditional previsions. Technical report, School of Math. and Phys. Sci., Univ. of Sussex, 1975.

[82] P. M. Williams. Notes on conditional previsions. *International Journal of Approximate Reasoning*, 44(3):366–383, 2007.

[83] K. Yin, H. Liu, and N. E. Johnson. Markovian inventory policy with application to the paper industry. *Computers and Chemical Engineering*, 26:1399–1413, 2002.

[84] M. Zaffalon, K. Wesnes, and O. Petrini. Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data. *Artificial Intelligence in Medicine*, 29(1–2):61–79, 2003.

Appendix A

List of Properties

For convenience, we present a list of all properties in order of appearance, with their abbreviated name and page reference.

Property 1 (Conditioning, p. 32)**.** *Let A be a non-empty event, and let $\mathcal{X}$ be a non-empty finite A-consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$. If $X \in \operatorname{opt}(\mathcal{X}|A)$, then $Y \in \operatorname{opt}(\mathcal{X}|A)$.*

Property 2 (Intersection, p. 32)**.** *For any event $A \neq \emptyset$ and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$ and $\operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$,*

$$\operatorname{opt}(\mathcal{Y}|A) = \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

Property 3 (Mixture, p. 32)**.** *For any events A and B such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, any $\overline{A} \cap B$-consistent gamble Z, and any non-empty finite $A \cap B$-consistent set of gambles $\mathcal{X}$,*

$$\operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B) = A \operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

Property 4 (Strong Path Independence, p. 33)**.** *For any non-empty event A and any non-empty finite A-consistent sets of gambles $\mathcal{X}_1, \ldots, \mathcal{X}_n$, there is a non-empty $\mathcal{I} \subseteq \{1, \ldots, n\}$ such that*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \bigcup_{i \in \mathcal{I}} \operatorname{opt}(\mathcal{X}_i|A).$$

Property 5 (Very Strong Path Independence, p. 33). *For any non-empty event A and any non-empty finite A-consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$,*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \bigcup_{\substack{i=1 \\ \mathcal{X}_i \cap \operatorname{opt}(\cup_{i=1}^{n} \mathcal{X}_i | A) \neq \emptyset}}^{n} \operatorname{opt}(\mathcal{X}_i | A)$$

Property 6 (Total Preordering, p. 33). *For every event $A \neq \emptyset$, there is a total preorder $\succeq_A$ on A-consistent gambles such that for every non-empty finite set of A-consistent gambles $\mathcal{X}$,*

$$\operatorname{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} \colon (\forall Y \in \mathcal{X})(X \succeq_A Y)\}$$

Property 7 (Multiple Mixture, p. 37). *For any event B and partition $A_1, \dots, A_n$ such that $A_i \cap B \neq \emptyset$ for all i, and sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$ such that $\mathcal{X}_i$ is $A_i \cap B$-consistent,*

$$\operatorname{opt}\left(\bigoplus_{i=1}^{n} A_i \mathcal{X}_i \middle| B\right) = \bigoplus_{i=1}^{n} A_i \operatorname{opt}(\mathcal{X}_i | A_i \cap B).$$

Property 8 (Backward Conditioning, p. 65). *Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, and let $\mathcal{X}$ be a non-empty finite $A \cap B$-consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$ and $X \in \operatorname{opt}(\mathcal{X}|A \cap B)$. If there is an $\overline{A} \cap B$-consistent gamble Z such that*

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z | B),$$

then $Y \in \operatorname{opt}(\mathcal{X}|A \cap B)$.

Property 9 (Insensitivity To Omission, p. 65). *For any event $A \neq \emptyset$, and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$,*

$$\operatorname{opt}(\mathcal{X}|A) \subseteq \mathcal{Y} \subseteq \mathcal{X} \Rightarrow \operatorname{opt}(\mathcal{Y}|A) = \operatorname{opt}(\mathcal{X}|A).$$

Property 10 (Preservation Under Addition, p. 65). *For any event $A \neq \emptyset$, and any non-empty finite A-consistent sets of gambles $\mathcal{X}$ and $\mathcal{Y}$,*

$$\mathcal{Y} \subseteq \mathcal{X} \Rightarrow \operatorname{opt}(\mathcal{Y}|A) \supseteq \operatorname{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

Property 11 (Backward Mixture, p. 65)**.** *For any events A and B such that $B \cap A \neq \emptyset$ and $B \cap \overline{A} \neq \emptyset$, any $B \cap \overline{A}$-consistent gamble Z, and any non-empty finite $B \cap A$-consistent set of gambles $\mathcal{X}$,*

$$\operatorname{opt}\big(A\mathcal{X} \oplus \overline{A}Z|B\big) \subseteq A \operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

Property 12 (Path Independence, p. 66)**.** *For any non-empty event A, and for any finite family of non-empty finite A-consistent sets of gambles $\mathcal{X}_1$, ..., $\mathcal{X}_n$,*

$$\operatorname{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \operatorname{opt}\left(\bigcup_{i=1}^{n} \operatorname{opt}(\mathcal{X}_i|A) \middle| A\right).$$

Property 13 (Axiom $7'$, p. 67)**.** *For any non-empty event A and any non-empty finite sets of gambles $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{Y} \subseteq \mathcal{X}$,*

$$\operatorname{opt}(\mathcal{Y}|A) \supseteq \mathcal{Y} \cap \operatorname{opt}(\mathcal{X}|A)$$

and

$$\operatorname{opt}(\mathcal{X}|A) \not\subset \operatorname{opt}(\mathcal{Y}|A).$$

Property 14 (Strong Backward Conditioning, p. 67)**.** *For any events A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$ and any non-empty finite set $A \cap B$-consistent set of gambles $\mathcal{X}$ with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$, $X \in \operatorname{opt}(\mathcal{X}|A \cap B)$ implies $Y \in \operatorname{opt}(\mathcal{X}|A \cap B)$ whenever there is a non-empty finite $\overline{A} \cap B$-consistent set of gambles $\mathcal{Z}$ such that, for at least one $Z \in \mathcal{Z}$,*

$$AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B).$$

Property 15 (Multiple Backward Mixture, p. 69)**.** *For any partition of Ω A_1, ..., A_n, any non-empty event B such that $A_i \cap B \neq \emptyset$ for all A_i, and for any non-empty finite sets of gambles $\mathcal{X}_1$, ..., $\mathcal{X}_n$ where each $\mathcal{X}_i$ is $A_i \cap B$-consistent,*

$$\operatorname{opt}\left(\bigoplus_{i=1}^{n} A_i \mathcal{X}_i \middle| B\right) \subseteq \bigoplus_{i=1}^{n} A_i \operatorname{opt}(\mathcal{X}_i|A_i \cap B).$$

Property 16 (Locality, p. 111)**.** *A choice function* opt *satisfies* locality *on S_0, ..., S_n whenever, for each sequential decision process on S_0, ..., S_n and each $1 \leq k < n$,*

$$\Pi_k^n(\cdot) = \Pi_k^k(\cdot) \times \Pi_{k+1}^{k+1}(\cdot) \times \cdots \times \Pi_n^n(\cdot).$$

Property 17 (Sequential Distributivity, p. 112)**.** *For any $1 \leq k < n$, any value h_{k-1} of H_{k-1}, all finite sets of gambles $\mathcal{X}$ on S_k, all finite sets of gambles $\mathcal{Y}(s_k)$ on F_{k+1} (one such set for each $s_k \in S_k$), and all $X \in \mathcal{X}$ and $Y(s_k) \in \mathcal{Y}(s_k)$:*

$$X + \bigoplus_{s_k} E_{s_k} Y(s_k) \in \operatorname{opt}\left(\mathcal{X} + \bigoplus_{s_k} E_{s_k}\mathcal{Y}(s_k)\middle| h_{k-1}\right)$$
$$\iff X \in \operatorname{opt}(\mathcal{X}|h_{k-1}) \text{ and } Y(s_k) \in \operatorname{opt}(\mathcal{Y}(s_k)|h_{k-1}s_k) \text{ for all } s_k.$$

Property 18 (Bellman's Principle of Optimality, p. 118)**.** *A normal form operator* norm *satisfies the principle of optimality if, for any dynamic programming tree T, and any node N in at least one element of* $\operatorname{norm}(T)$,

$$\operatorname{st}_N(\operatorname{norm}(T)) \subseteq \operatorname{norm}(\operatorname{st}_N(T)).$$

Equivalently, for any normal form decision $U \in \operatorname{norm}(T)$ and any node N in U,

$$\operatorname{st}_N(U) \in \operatorname{norm}(\operatorname{st}_N(T)).$$

Property 19 (Backward Addition, p. 120)**.** *For any gamble X and any non-empty finite set of gambles $\mathcal{Y}$,*

$$\operatorname{opt}(X + \mathcal{Y}) \subseteq X + \operatorname{opt}(\mathcal{Y}).$$

Property 20 (Addition, p. 124)**.** *For any gamble X and any non-empty finite set of gambles $\mathcal{Y}$,*

$$\operatorname{opt}(X + \mathcal{Y}) = X + \operatorname{opt}(\mathcal{Y}).$$

Property 21 (Weak Multiple Mixture, p. 48)**.** *For any non-empty event B and partition $A_1, \dots, A_n$ such that $A_i \cap B \neq \emptyset$, and any non-empty finite sets of gambles $\mathcal{X}_i, \dots, \mathcal{X}_n$ such that $\mathcal{X}_i$ is $A_i \cap B$-consistent,*

- *if $\bigoplus A_i X_i \in \operatorname{opt}\left(\bigoplus A_i \mathcal{X}_i | B\right)$, then $X_i \in \operatorname{opt}(\mathcal{X}_i | A_i \cap B)$, and,*
- *if $X_k \in \operatorname{opt}(\mathcal{X}_k | A_k \cap B)$, then for each $j \neq k$, there is a $X_j \in \mathcal{X}_j$ such that $\bigoplus A_i X_i \in \operatorname{opt}\left(\bigoplus A_i \mathcal{X}_i | B\right)$.*

Appendix B

Supplementary Definitions and Proofs

B.1 Definition Of Decision Trees

Hammond [24, pp. 31–32] gives a definition of *consequentialist decision trees.* The decision trees we introduce informally in Section 1.2.3 are very similar; they model essentially the same problems as Hammond's. It is easy to provide a definition of our decision trees that follows the same approach as Hammond. We do not find this adds any use or understanding, but the reader may disagree.

Definition B.1. *A* decision tree *is a collection*

$$T = (\Omega, \mathcal{R}, \mathcal{N}, \mathcal{D}, \mathcal{C}, \mathcal{L}, N_0, \mathrm{succ}(\cdot), \mathrm{S}(\cdot), \mathrm{rew}(\cdot), \mathrm{ev}(T)),$$

where

- Ω *is a possibility space;*
- $\mathcal{R}$ *is a reward set;*
- $\mathcal{N}$ *is a set of nodes partitioned into the three sets* $\mathcal{D}$, $\mathcal{C}$, $\mathcal{L}$ *described below;*
- $\mathcal{D}$ *is a (possibly empty) set of decision nodes;*
- $\mathcal{C}$ *is a (possibly empty) set of chance nodes;*

- $\mathcal{L}$ *is a non-empty set of leaves or reward nodes;*
- N_0 *is the* initial node*;*
- $\text{succ}\colon \mathcal{N} \to \mathcal{P}(\mathcal{N})$ *is the* immediate successor function *satisfying*
  - *For all* $N \in \mathcal{N}$, $N \notin \text{succ}(N)$*;*
  - *For all* $N \in \mathcal{N}$, $\text{succ}(N) = \emptyset \iff N \in \mathcal{L}$*;*
  - *For all* $N, N' \in \mathcal{N}$, $\text{succ}(N) \cap \text{succ}(N') \neq \emptyset \iff N = N'$*;*
  - *For all* $N \in \mathcal{N}$, $N = N_0$ *if and only if, for all* $N' \in \mathcal{N}$, $N \notin \text{succ}(N')$*;*
- S *is a function defined on the set of all nodes whose immediate predecessor is a chance node, and maps each such node to an event, such that, for all* $N \in \mathcal{C}$, $\{S(N')\colon N' \in \text{succ}(N)\}$ *forms a partition of* Ω*;*
- $\text{rew}\colon \mathcal{L} \to \mathcal{R}$ *identifies the reward received for reaching each leaf;*
- $\text{ev}(T)$ *is a non-empty subset of* Ω*, representing all events observed prior to* N_0.

B.2 Results for Sets of Gambles

This section details a few useful results about the behaviour of certain sets of gambles.

Lemma B.2. *For any decision tree* T, $\text{gamb}(T) = \text{gamb}(\text{nfd}(T))$.

Proof. We prove this by structural induction. In the base step, we prove the equality for trees comprising only one node. In the induction step, we prove that if the equality holds for the subtrees at every child of the root node of T, then the equality also holds for T.

If T consists of only a single node, namely a reward node, then $\text{nfd}(T) = \{T\}$ and the result holds trivially. Thus the base step is confirmed.

Suppose T has a chance node at the root, that is, $T = \bigodot_{i=1}^{n} E_i T_i$. Each element of $\text{nfd}(T)$ is of the form $\bigodot_{i=1}^{n} U_i$, where $U_i \in \text{nfd}(T_i)$. In other words, $\text{nfd}(T)$ is the set of all possible mixtures of the elements of $\mathrm{nfd}(T_i)$, that is,

$$\mathrm{nfd}(T) = \bigodot_{i=1}^{n} E_i \, \mathrm{nfd}(T_i).$$

The induction hypothesis is $\mathrm{gamb}(T_i) = \mathrm{gamb}(\mathrm{nfd}(T_i))$ for each i. We have

$$\begin{aligned}
\mathrm{gamb}(\mathrm{nfd}(T)) &= \mathrm{gamb}\left(\bigodot_{i=1}^{n} E_i \, \mathrm{nfd}(T_i)\right) \\
&= \mathrm{gamb}\left(\left\{\bigodot_{i=1}^{n} E_i U_i \colon U_i \in \mathrm{nfd}(T_i)\right\}\right) \\
&= \bigoplus_{i=1}^{n} E_i \, \mathrm{gamb}(\mathrm{nfd}(T_i)) \\
&= \bigoplus_{i=1}^{n} E_i \, \mathrm{gamb}(T_i) \\
&= \mathrm{gamb}(T).
\end{aligned}$$

On the other hand, if T has a decision node as a root, that is, $T = \bigsqcup_{i=1}^{n} T_i$, then

$$\mathrm{nfd}(T) = \left\{ \sqcup\, U \colon U \in \bigcup_{i=1}^{n} \mathrm{nfd}(T_i) \right\},$$

and, since $\mathrm{gamb}(\sqcup U) = \mathrm{gamb}(U)$ for any U,

$$\begin{aligned}
\mathrm{gamb}(\mathrm{nfd}(T)) &= \left\{ \mathrm{gamb}(\sqcup U) \colon U \in \bigcup_{i=1}^{n} \mathrm{nfd}(T_i) \right\} \\
&= \left\{ \mathrm{gamb}(U) \colon U \in \bigcup_{i=1}^{n} \mathrm{nfd}(T_i) \right\} \\
&= \bigcup_{i=1}^{n} \mathrm{gamb}(\mathrm{nfd}(T_i))
\end{aligned}$$

and again, the induction hypothesis says that $\mathrm{gamb}(T_i) = \mathrm{gamb}(\mathrm{nfd}(T_i))$ for each i, so

$$\begin{aligned}
&= \bigcup_{i=1}^{n} \mathrm{gamb}(T_i) \\
&= \mathrm{gamb}(T).
\end{aligned}$$

This completes the induction step. $\square$

Lemma B.3. *Let $A_1, \ldots, A_n$ be a finite partition of Ω, $n \geq 2$. Let $\mathcal{X}$, and $\mathcal{X}_1$, $\mathcal{Z}_1$, $\ldots$, $\mathcal{X}_n$, $\mathcal{Z}_n$ be non-empty finite sets of gambles. If*

$$\mathcal{X} = A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k$$

for all $k \in \{1, \ldots, n\}$, then

$$\mathcal{X} = \bigoplus_{i=1}^{n} A_i \mathcal{X}_i.$$

Proof. Let $X \in \mathcal{X}$. Then $X \in A_k \mathcal{X} \oplus \overline{A}_k \mathcal{Z}_k$ for all k. Therefore, $A_k X_k \in A_k \mathcal{X}_k$ for all k, where $A_k \mathcal{X}_k$ is all the restrictions of gambles of $\mathcal{X}_k$ to A_k. So, $X \in \bigoplus_{i=1}^{n} A_k \mathcal{X}_k$, establishing that $\mathcal{X} \subseteq \bigoplus_{i=1}^{n} A_k \mathcal{X}_k$.

Now we show using induction that $\bigoplus_{i=1}^{n} A_i \mathcal{X}_i \subseteq \mathcal{X}$. The inductive step is to show that, if

$$X = A_1 X_1 \oplus \ldots \oplus A_k X_k \oplus A_{k+1} Z_{k+1} \oplus \ldots \oplus A_n Z_n \in \mathcal{X},$$

for any $X_1 \in \mathcal{X}_1, \ldots, X_k \in \mathcal{X}_k$ and some $Z_{k+1}, \ldots, Z_n$, then for any $X_{k+1} \in \mathcal{X}_{k+1}$,

$$A_1 X_1 \oplus \ldots \oplus A_{k+1} X_{k+1} \oplus A_{k+2} Z_{k+2} \oplus \ldots \oplus A_n Z_n \in \mathcal{X}.$$

The base step, for $k = 1$, clearly holds by definition of $\mathcal{X}$. Suppose $k > 1$. Observe that X can be rewritten as

$$X = A_{k+1} Z_{k+1} \oplus \overline{A}_{k+1} X,$$

that is, $X \in \mathcal{Z}_{k+1}$. Therefore $A_{k+1} X_{k+1} \oplus \overline{A}_{k+1} X \in \mathcal{X}$ for all $X_{k+1} \in \mathcal{X}_{k+1}$. But

$$A_{k+1} X_{k+1} \oplus \overline{A}_{k+1} X = A_1 X_1 \oplus \ldots \oplus A_{k+1} X_{k+1} \oplus A_{k+2} Z_{k+2} \oplus \ldots \oplus A_n Z_n,$$

proving the inductive step. □

Lemma B.4. *Let $A_1, \ldots, A_n$ be a finite partition of Ω, $n \geq 2$. Let $\mathcal{X}$, and $\mathcal{X}_1$, $\mathcal{Z}_1$, $\ldots$, $\mathcal{X}_n$, $\mathcal{Z}_n$ be finite sets of gambles. If*

$$\mathcal{X} \subseteq A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k$$

for all $k \in \{1, \ldots, n\}$, then

$$\mathcal{X} \subseteq \bigoplus_{k=1}^{n} A_k \mathcal{X}_k.$$

Proof. Let $X \in \mathcal{X}$, then $X \in A_k\mathcal{X}_k \oplus \overline{A}_k\mathcal{Z}_k$ for all k. Therefore, for each k, there is an $X_k \in \mathcal{X}_k$ such that $A_kX = A_kX_k$. Whence,

$$X = \bigoplus_{k=1}^{n} A_kX_k \in \bigoplus_{k=1}^{n} A_k\mathcal{X}_k.$$

□

B.3 Lemmas for the Backward Induction Theorem

This section provides proofs for some of the intermediate results required for Theorem 3.12.

Proof of Lemma 3.13. In this proof we require Eq. (1.10), and the following consequence of Eq. (1.7):

$$\operatorname{gamb}\left(\operatorname{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i)\right)\right) \\ = \operatorname{opt}\left(\bigoplus_{i=1}^{n} E_i \operatorname{opt}(\operatorname{gamb}(T_i)|\operatorname{ev}(T) \cap E_i)\middle|\operatorname{ev}(T)\right). \quad \text{(B.3.1)}$$

We first show that

$$\operatorname{norm}_{\mathrm{opt}}(T) \supseteq \operatorname{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i)\right).$$

Consider a normal form decision $U \in \operatorname{norm}_{\mathrm{opt}}(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i))$. We have, by Eq. (3.6),

$$\operatorname{gamb}(U) \subseteq \operatorname{gamb}\left(\operatorname{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i)\right)\right) = \operatorname{gamb}(\operatorname{norm}_{\mathrm{opt}}(T)).$$

So, there exists a normal form decision $V \in \operatorname{norm}_{\mathrm{opt}}(T)$ such that $\operatorname{gamb}(V) = \operatorname{gamb}(U)$. Since $U \in \operatorname{nfd}(T)$, by definition of $\operatorname{norm}_{\mathrm{opt}}$, $U \in \operatorname{norm}_{\mathrm{opt}}(T)$, which establishes the claim.

Next, we show that

$$\operatorname{norm}_{\mathrm{opt}}(T) \subseteq \operatorname{norm}_{\mathrm{opt}}\left(\bigodot_{i=1}^{n} E_i \operatorname{norm}_{\mathrm{opt}}(T_i)\right).$$

Consider a normal form decision $U \in \mathrm{norm}_{\mathrm{opt}}(T)$. We know by Eq. (3.6) and Eq. (B.3.1) that

$$\mathrm{gamb}(U) \subseteq \mathrm{opt}\left(\bigoplus_{i=1}^{n} E_i \,\mathrm{opt}(\mathrm{gamb}(T_i)|\mathrm{ev}(T) \cap E_i)\middle|\mathrm{ev}(T)\right).$$

We can write $U = \bigodot_{i=1}^{n} E_i U_i$, where $U_i \in \mathrm{nfd}(T_i)$, so by Eq. (1.6b),

$$\bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(U_i) \subseteq \mathrm{opt}\left(\bigoplus_{i=1}^{n} E_i \,\mathrm{opt}(\mathrm{gamb}(T_i)|\mathrm{ev}(T) \cap E_i)\middle|\mathrm{ev}(T)\right).$$

Consider normal form decisions $V_i \in \mathrm{norm}_{\mathrm{opt}}(T_i)$. The above equation and Eq. (1.10) tell us that, for each i, we can find V_i such that $E_i \,\mathrm{gamb}(V_i) = E_i \,\mathrm{gamb}(U_i)$. Of course, because $V_i \in \mathrm{norm}_{\mathrm{opt}}(T_i)$,

$$\mathrm{gamb}(V_i) \subseteq \mathrm{opt}(\mathrm{gamb}(T_i)|E_i \cap \mathrm{ev}(T)).$$

We further have, for $V = \bigodot_{i=1}^{n} E_i V_i$, $\mathrm{gamb}(V) = \mathrm{gamb}(U)$ and $V \in \mathrm{nfd}(T)$, and so $V \in \mathrm{norm}_{\mathrm{opt}}(T)$.

If we can establish that

$$\mathrm{gamb}(U_i) \subseteq \mathrm{opt}(\mathrm{gamb}(T_i)|E_i \cap \mathrm{ev}(T)), \tag{B.3.2}$$

then, by definition of $\mathrm{norm}_{\mathrm{opt}}$ and because $U_i \in \mathrm{nfd}(T_i)$, it follows that $U_i \in \mathrm{norm}_{\mathrm{opt}}(T_i)$. So, in that case, there is a $V \in \mathrm{norm}_{\mathrm{opt}}(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i))$ such that $\mathrm{gamb}(V) = \mathrm{gamb}(U)$, and $U \in \mathrm{nfd}(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i))$. Therefore by definition of $\mathrm{norm}_{\mathrm{opt}}$, we will have $U \in \mathrm{norm}_{\mathrm{opt}}(\bigodot_{i=1}^{n} E_i \,\mathrm{norm}_{\mathrm{opt}}(T_i))$, establishing the desired result.

We use Lemma 3.6 to show that Eq. (B.3.2) indeed holds by Backward Conditioning. When $n = 1$ the result is trivial, so assume $n \geq 2$. Observe that both singletons $\mathrm{gamb}(U_i)$ and $\mathrm{gamb}(V_i)$ are subsets of $\mathrm{gamb}(T_i)$, $E_i \,\mathrm{gamb}(U_i) = E_i \,\mathrm{gamb}(V_i)$, and $\mathrm{gamb}(V_i) \subseteq \mathrm{opt}(\mathrm{gamb}(T_i)|E_i \cap \mathrm{ev}(T))$. Further, $\mathrm{gamb}(T_i)$ is $E_i \cap \mathrm{ev}(T)$-consistent. We are almost ready to apply Backward Conditioning We use Lemma 3.6 and Strong Backward Conditioning.

We know that

$$\begin{aligned}\mathrm{gamb}(V) = \bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(V_i) &\subseteq \mathrm{opt}(\mathrm{gamb}(T)|\mathrm{ev}(T)) \\ &= \mathrm{opt}\left(\bigoplus_{i=1}^{n} E_i \,\mathrm{gamb}(T_i)\middle|\mathrm{ev}(T)\right).\end{aligned}$$

Letting

$$Z = (E_1 \cup E_2)\,\mathrm{gamb}(V_2) \oplus E_3\,\mathrm{gamb}(V_3) \oplus \ldots \oplus E_n\,\mathrm{gamb}(V_n)$$

and

$$\mathcal{Z} = (E_1 \cup E_2)\,\mathrm{gamb}(T_2) \oplus E_3\,\mathrm{gamb}(T_3) \oplus \ldots \oplus E_n\,\mathrm{gamb}(T_n),$$

we see that $Z \in \mathcal{Z}$ and $\mathcal{Z}$ is $\overline{E}_1 \cap \mathrm{ev}(T)$-consistent. Further,

$$E_1\,\mathrm{gamb}(V_1) \oplus \overline{E}_1 Z = \bigoplus_{i=1}^{n} E_i\,\mathrm{gamb}(V_i) = \mathrm{gamb}(V)$$

and

$$E_1\,\mathrm{gamb}(T_1) \oplus \overline{E}_1 \mathcal{Z} = \bigoplus_{i=1}^{n} E_i\,\mathrm{gamb}(T_i).$$

We see that

$$E_1\,\mathrm{gamb}(V) \oplus \overline{E}_1 Z \subseteq \mathrm{opt}(E_1\,\mathrm{gamb}(T_1) \oplus \overline{E}_1 \mathcal{Z} | \mathrm{ev}(T)).$$

Hence we have found a $\mathcal{Z}$ and a $Z \in \mathcal{Z}$ required to apply Strong Backward Conditioning. Finally, by $E_1\,\mathrm{gamb}(U_1) = E_1\,\mathrm{gamb}(V_1)$, and Strong Backward Conditioning, we have

$$\mathrm{gamb}(U_1) \subseteq \mathrm{opt}(\mathrm{gamb}(T_1) | E_1 \cap \mathrm{ev}(T)).$$

This argument applies for any index i and therefore Eq. (B.3.2) has been shown, establishing the result. □

Proof of Lemma 3.14. In this proof we require Eq. (1.10). For clarity, let $A = \mathrm{ev}(T) = \mathrm{ev}(T_i)$. We first show that

$$\mathrm{norm}_{\mathrm{opt}}(T) \supseteq \mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i) \right).$$

Consider a normal form decision $U \in \mathrm{norm}_{\mathrm{opt}}(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i))$. To show that $U \in \mathrm{norm}_{\mathrm{opt}}(T)$, we must show that $U \in \mathrm{nfd}(T)$ and $\mathrm{gamb}(U) \subseteq \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T))$. The former is obvious, and the latter is established by Eq. (3.7):

$$\mathrm{gamb}(U) \subseteq \mathrm{gamb}\left(\mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i) \right) \right) = \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)).$$

Next we show that

$$\mathrm{norm}_{\mathrm{opt}}(T) \subseteq \mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i)\right).$$

Let $U \in \mathrm{norm}_{\mathrm{opt}}(T)$. To show that $U \in \mathrm{norm}_{\mathrm{opt}}(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i))$ we must show that $U \in \mathrm{nfd}(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i))$ and that

$$\mathrm{gamb}(U) \subseteq \mathrm{gamb}\left(\mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i)\right)\right).$$

The latter requirement follows immediately from Eq. (3.7):

$$\mathrm{gamb}(U) \subseteq \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{gamb}\left(\mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i)\right)\right).$$

We now prove that $U \in \mathrm{nfd}(\bigsqcup_{i=1}^{n} \mathrm{norm}_{\mathrm{opt}}(T_i))$. Let V be U with the root node removed, that is, $U = \sqcup V$. Clearly, for some k, $V \in \mathrm{nfd}(T_k)$. It suffices to show that $V \in \mathrm{norm}_{\mathrm{opt}}(T_k)$. Let $\{X\} = \mathrm{gamb}(U) = \mathrm{gamb}(V)$. We know that $X \in \mathrm{opt}(\mathrm{gamb}(T)|A)$, and also that $X \in \mathrm{gamb}(T_k)$. If we can prove that $X \in \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_k)) = \mathrm{opt}(\mathrm{gamb}(T_k)|A)$, then $V \in \mathrm{norm}_{\mathrm{opt}}(T_k)$. Indeed, using Preservation Under Addition and $\mathrm{gamb}(T_k) \subseteq \mathrm{gamb}(T)$ we have

$$\mathrm{opt}(\mathrm{gamb}(T_k)|A) \supseteq \mathrm{gamb}(T_k) \cap \mathrm{opt}(\mathrm{gamb}(T)|A).$$

So we have shown that indeed $X \in \mathrm{opt}(\mathrm{gamb}(T_k)|A)$, establishing the claim. □

Proof of Lemma 3.15. Let A and B be non-empty events, and $\mathcal{X}$ be a non-empty finite set of gambles, and Z be a gamble, such that the following properties hold: $A \cap B \neq \emptyset$, $\overline{A} \cap B \neq \emptyset$, $\mathcal{X}$ is $A \cap B$-consistent, Z is $\overline{A} \cap B$-consistent, and there are $X, Y \in \mathcal{X}$ such that $AX = AY$, $X \in \mathrm{opt}(\mathcal{X}|A \cap B)$, and $AX \oplus \overline{A}Z \in \mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. If it is not possible to construct such a situation, then opt satisfies Backward Conditioning automatically. Otherwise, to prove that Backward Conditioning holds, we must show that $Y \in \mathrm{opt}(\mathcal{X}|A \cap B)$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\mathrm{ev}(T) = B$, $\mathrm{gamb}(T_1) = \mathcal{X}$, and $\mathrm{gamb}(T_2) = \{Z\}$. Since $\mathcal{X}$ is $A \cap B$-consistent and Z is $\overline{A} \cap B$-consistent, we know from Definition 1.17 that there is such a T. We have $\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(\mathrm{gamb}(T)|B) = \mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. So, $AX \oplus \overline{A}Z \in \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T))$, and of course $AX \oplus \overline{A}Z = AY \oplus \overline{A}Z$.

Therefore, any normal form decision in $\mathrm{nfd}(T)$ that induces the gamble $AY \oplus \overline{A}Z$ must be in $\mathrm{norm}_{\mathrm{opt}}(T)$. In particular, by Lemma B.2 there is a normal form decision $U \in \mathrm{nfd}(T_1)$ such that $\mathrm{gamb}(U) = \{Y\}$, and a normal form decision $V \in \mathrm{nfd}(T_2)$ such that $\mathrm{gamb}(V) = \{Z\}$. So $AU \odot \overline{A}V \in \mathrm{nfd}(T)$ and $\mathrm{gamb}(AU \odot \overline{A}V) = \{AY \oplus \overline{A}Z\}$. Indeed, because $AX \oplus \overline{A}Z \in \mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$, it follows that $AU \odot \overline{A}V \in \mathrm{norm}_{\mathrm{opt}}(T) = \mathrm{back}_{\mathrm{opt}}(T)$. By definition, $\mathrm{back}_{\mathrm{opt}}(T) = \mathrm{norm}_{\mathrm{opt}}(A\,\mathrm{back}_{\mathrm{opt}}(T_1) \odot \overline{A}\,\mathrm{back}_{\mathrm{opt}}(T_2))$, and so it must hold that $U \in \mathrm{back}_{\mathrm{opt}}(T_1) = \mathrm{norm}_{\mathrm{opt}}(T_1)$. Whence, $\mathrm{gamb}(U) \subseteq \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)) = \mathrm{opt}(\mathcal{X}|A \cap B)$. Since $\mathrm{gamb}(U) = \{Y\}$, we have $Y \in \mathrm{opt}(\mathcal{X}|A \cap B)$, establishing Backward Conditioning. □

Proof of Lemma 3.16. Let A be any non-empty event, and $\mathcal{X}_1, \ldots, \mathcal{X}_n$ be any non-empty finite sets of A-consistent gambles. Let $T = \bigsqcup_{i=1}^n T_i$ be a consistent decision tree, with $\mathrm{gamb}(T_i) = \mathcal{X}_i$ for each i, and where $\mathrm{ev}(T) = A$. The existence of T is assured by A-consistency of $\mathcal{X}$ (see Definition 1.17). We have, with repeated applications of Eq. (1.10) and Eq. (1.8),

$$\begin{aligned}
\mathrm{opt}\left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A\right) &= \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T)) \\
&= \mathrm{gamb}\left(\mathrm{norm}_{\mathrm{opt}}\left(\bigsqcup_{i=1}^n \mathrm{back}_{\mathrm{opt}}(T_i)\right)\right) \\
&= \mathrm{opt}\left(\mathrm{gamb}\left(\bigsqcup_{i=1}^n \mathrm{back}_{\mathrm{opt}}(T_i)\right) \middle| A\right) \\
&= \mathrm{opt}\left(\bigcup_{i=1}^n \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_i)) \middle| A\right) \\
&= \mathrm{opt}\left(\bigcup_{i=1}^n \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_i)) \middle| A\right) \\
&= \mathrm{opt}\left(\bigcup_{i=1}^n \mathrm{opt}(\mathrm{gamb}(T_i)|A) \middle| A\right) \\
&= \mathrm{opt}\left(\bigcup_{i=1}^n \mathrm{opt}(\mathcal{X}_i|A) \middle| A\right).
\end{aligned}$$

□

Proof of Lemma 3.17. Let A and B be non-empty events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let $\mathcal{X}$ be a non-empty finite set of $A \cap B$-consistent gambles, and let Z be an $\overline{A}\cap B$-consistent gamble. Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree, where $\mathrm{ev}(T) = B$, $\mathrm{gamb}(T_1) = \mathcal{X}$, and $\mathrm{gamb}(T_2) = \{Z\}$. The existence of T is assured by $A \cap B$-consistency of $\mathcal{X}$ and $\overline{A} \cap B$-consistency of Z. By assumption,

$$\begin{aligned}\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) &= \mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T)) \\ &= \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(A\,\mathrm{back}_{\mathrm{opt}}(T_1) \odot \overline{A}T_2)).\end{aligned}$$

From Eq. (1.10), we have $\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T)) = \mathrm{opt}(\mathrm{gamb}(T)|B)$. Similarly, with repeated applications of Eq. (1.10) and Eq. (1.7),

$$\begin{aligned}\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(A\,\mathrm{back}_{\mathrm{opt}}(T_1) \odot \overline{A}T_2)) &= \mathrm{opt}(\mathrm{gamb}(A\,\mathrm{back}_{\mathrm{opt}}(T_1) \odot \overline{A}T_2)|B) \\ &= \mathrm{opt}(A\,\mathrm{gamb}(\mathrm{back}_{\mathrm{opt}}(T_1)) \oplus \overline{A}Z|B) \\ &= \mathrm{opt}(A\,\mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) \oplus \overline{A}Z|B) \\ &= \mathrm{opt}(A\,\mathrm{opt}(\mathrm{gamb}(T_1)|A \cap B) \oplus \overline{A}Z|B).\end{aligned}$$

Finally we note that $\mathrm{gamb}(T) = A\mathcal{X} \oplus \overline{A}Z$, and $\mathrm{gamb}(T_1) = \mathcal{X}$. Therefore,

$$\mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B) = \mathrm{opt}(A\,\mathrm{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z|B) \subseteq A\,\mathrm{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

□

B.4 Proofs for Weak Subtree Perfectness Theorem

This section provides proofs for some intermediate results required for Theorem 3.23.

Proof of Lemma 3.25. Let A and B be non-empty events, $\mathcal{X}$,be a non-empty finite set of gambles, and Z be a gamble, such that the following properties hold: $A\cap B \neq \emptyset$, $\overline{A} \cap B \neq \emptyset$, $\mathcal{X}$ is $A \cap B$-consistent, Z is $\overline{A} \cap B$-consistent, and there are $X, Y \in \mathcal{X}$ such that $AX = AY$, $X \in \mathrm{opt}(\mathcal{X}|A\cap B)$, and $AX \oplus \overline{A}Z \in \mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. If it is not possible to construct such a situation, then opt satisfies Backward Conditioning automatically. Otherwise, to prove that Backward Conditioning holds, we must show that $Y \in \mathrm{opt}(\mathcal{X}|A \cap B)$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\mathrm{ev}(T) = B$, $\mathrm{gamb}(T_1) = \mathcal{X}$, and $\mathrm{gamb}(T_2) = \{Z\}$. Since $\mathcal{X}$ is $A \cap B$-consistent and Z is $\overline{A} \cap B$-consistent, we know from Definition 1.17 that there is such a T. By Lemma B.2, there is a normal form decision $U \in \mathrm{nfd}(T_1)$ such that $\mathrm{gamb}(U) = \{Y\}$, and a normal form decision in $V \in \mathrm{nfd}(T_2)$ such that $\mathrm{gamb}(V) = \{Z\}$.

Since, by assumption, $AX = AY$, obviously $AX \oplus \overline{A}Z = AY \oplus \overline{A}Z$, and hence, also $AY \oplus \overline{A}Z \in \mathrm{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. Therefore, by definition of $\mathrm{norm}_{\mathrm{opt}}$, $AU \odot \overline{A}V \in \mathrm{norm}_{\mathrm{opt}}(T)$. In particular, $U \in \mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T))$, where N is the root of T_1. Because $\mathrm{norm}_{\mathrm{opt}}$ is subtree perfect for normal form decisions, it follows that also $U \in \mathrm{norm}_{\mathrm{opt}}(\mathrm{st}_N(T)) = \mathrm{norm}_{\mathrm{opt}}(T_1)$. Again applying the definition of $\mathrm{norm}_{\mathrm{opt}}$, we conclude that indeed $Y \in \mathrm{opt}(\mathcal{X}|A \cap B)$. □

Proof of Lemma 3.26. Let A be a non-empty event, and let $\mathcal{X}$ be a non-empty finite set of A-consistent gambles. Let $\mathcal{Y}$ be a non-empty subset of $\mathcal{X}$. Let $T = T_1 \sqcup T_2$, where $\mathrm{ev}(T) = A$, $\mathrm{gamb}(T_1) = \mathcal{Y}$, and $\mathrm{gamb}(T_2) = \mathcal{X}$. Let N be the root of T_1.

If $\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y} = \emptyset$ then Preservation Under Addition holds automatically. Suppose $\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. By definition of $\mathrm{norm}_{\mathrm{opt}}$, N appears in at least one element of $\mathrm{norm}_{\mathrm{opt}}(T)$, and

$$\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y} = \mathrm{gamb}(\mathrm{st}_N(\mathrm{norm}_{\mathrm{opt}}(T))),$$

and by subtree perfectness of normal form decisions

$$\begin{aligned} &\subseteq \mathrm{gamb}(\mathrm{norm}_{\mathrm{opt}}(T_1)) \\ &= \mathrm{opt}(\mathcal{Y}|A). \end{aligned}$$

□

Proof of Lemma 3.27. Let A and B be non-empty events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let $\mathcal{X}$ be a non-empty finite set of $A \cap B$-consistent gambles, and let Z be an $\overline{A} \cap B$-consistent gamble. Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree, where $\mathrm{ev}(T) = B$, $\mathrm{gamb}(T_1) = \mathcal{X}$, and T_2 is simply a normal form decision with $\mathrm{gamb}(T_2) = \{Z\}$. The existence of T is assured by $A \cap B$-consistency of $\mathcal{X}$ and $\overline{A} \cap B$-consistency of $\{Z\}$. Let N be the root of T_1.

Consider any gamble $AX \oplus \overline{A}Z \in \operatorname{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. By Lemma B.2, there is a $U \in \operatorname{nfd}(T_1)$ such that $\operatorname{gamb}(U) = X$. By definition of $\operatorname{norm}_{\mathrm{opt}}$, it follows that $AU \odot \overline{A}T_2 \in \operatorname{norm}_{\mathrm{opt}}(T)$, and hence, in particular, $U \in \operatorname{st}_N(\operatorname{norm}_{\mathrm{opt}}(T))$. By subtree perfectness for normal form decisions, $U \in \operatorname{norm}_{\mathrm{opt}}(T_1)$. Again applying the definition of $\operatorname{norm}_{\mathrm{opt}}$, we find that $X \in \operatorname{opt}(\mathcal{X}|A \cap B)$, thus indeed $AX \oplus \overline{A}Z \in A \operatorname{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z$, whence Backward Mixture is established. $\square$