Influence Diagrams Involving Time

Søren Holbech Nielsen
holbech@cs.auc.dk

Department of Computer Science
Aalborg University

10th June 2003

Aalborg University

Department of Computer Science, Fredrik Bajers Vej 7E, 9220 Aalborg Øst

TITLE:

Influence Diagrams Involving Time

SUBJECT:

Decision Support Systems

PROJECT GROUP:

E2-101

MEMBERS OF PROJECT GROUP:

Søren Holbech Nielsen

SUPERVISOR:

Thomas D. Nielsen

SEMESTER:

DAT6

PROJECT PERIOD:

1st of February 2003 -
10th of June 2003

NUMBER OF COPIES:

4

NUMBER OF PAGES:

Report: 106 pages
Total: 116 pages

Abstract: IDITs was originally proposed in [Broe et al., 2003] as a representation language for decision problems involving quantitative measures of time. IDITs is suggested as a representation language, which caters for aspects of time. However, the temporal semantics of elements in IDITs presented in [Broe et al., 2003] are flawed. In this report, we suggest a new set of ordering semantics and a definition of welldefinedness that builds on this new ordering. Furthermore, a method to check an IDIT for welldefinedness is given, and the representation language of IDITs is enhanced to cater for more aspects of time, including varying orderings of decisions.

[Broe et al., 2003] also neglects to present a method for solving decision problems modelled as IDITs, but do suggest a sketch for such a method. In the latter part of this report, we explore the boundaries of this sketch and identify a subset of IDITs that can be solved using this approach. Our method succeeds in handling continuous variables as parents of discrete decisions through exploitation of constraints induced by the nature of time.

Preface

This report constitutes my master thesis in the field of decision support systems. It was developed at the Dat6 semester at the Department of Computer Science at Aalborg University. The project period spans the spring semester of 2003, from the 1st of February 2003 to the 10th of June 2003.

I would like to thank Thomas D. Nielsen for exceptional supervision and commitment during the semester. Furthermore, I would also like to thank Mads Broe and Rune Jeppesen for being enthusiastic debating partners throughout the semester.

Søren Holbech Nielsen

Contents

1 Introduction **1**

2 Influence Diagrams Involving Time **5**

2.1 The Original Representation Language 5

2.2 Alterations of the Original Framework 19

2.3 Temporal Ordering in IDITs 29

3 Solutions to IDITs **47**

3.1 Solutions to Decision Problems 47

3.2 Solutions to IDITs 51

3.3 Solving IDITs 57

4 Conclusion **105**

A Summary **115**

Chapter 1

Introduction

Influence diagrams involving time (henceforth referred to as IDITs) is a framework for representing decision problems that involve quantitative measures of time. It is the result of an analysis of decision problems, frameworks traditionally used for modelling decision problems, and how these frameworks cope with aspects of time, carried out in [Broe et al., 2003]. The conclusion of this analysis is that none of the analyzed frameworks, viz. decision trees, influence diagrams, and valuation networks, are suitable for modelling decision problems involving time. This is because these cater only for qualitative aspects of time, such as ordering of decisions and observations, and some decision problems revolve around quantitative aspects of time, such as deadlines or entities who change as time progresses. Consequently, IDITs was developed as an alternative.

Frameworks that are traditionally used for modelling decision problems, such as influence diagrams, decision trees, and valuation networks, have a set of semantics associated with them that allows humans to read and understand modelled decision problems in an unambiguous manner. Furthermore, they include a syntax which, in conjunction with the semantics, renders decision problems solvable on a computer. That is, a strategy, which maximizes the expected utility of a decision taker, can be computed from a model of the decision problem.

IDITs is meant to be an extension of the influence diagrams framework, originally proposed in [Howard and Matheson, 1984], and is a compact and unambiguous framework, which portrays decision problems involving time in a fashion that should be easy to grasp for modellers experienced in modelling decision problems using influence diagrams. The extension is a true extensions, in the sense that an IDIT, modelling a decision problem that does not involve time, is an influence diagram, which can be reasoned about using the set of semantics traditionally associated with influence diagrams. Unfortunately, [Broe et al., 2003] fails to provide a clear semantical interpretation of modelled decision problems. Specifically, a temporal ordering of events and decisions in the problem is flawed. Furthermore, a method for solving IDITs modelling decision problems, which do involve time, is not given, as [Broe et al., 2003] settles on a sketch of such a method.

The sketch, although brief, brings to light the difficulty in solving a decision problem modelled as an IDIT: Decision problems are represented as decision graphs, in which points in time are represented as continuous variables. Time variables' impact on other variables and policies, which are both discrete, are not easily evaluated. In this report we, actually, give an example of an IDIT, modelling a decision problem involving time, which cannot be solved exactly using known algebraic manipulations. The basic problem of integrating continuous and discrete variables in the same decision graph, and more specifically influence diagrams, has been given a lot of attention in the research community. An approach to using continuous variables in influence diagrams, called Gaussian influence diagrams, are given in [Shachter and Kenley, 1989]. A Gaussian influence diagram consists of continuous variables only, where chance variables follow Gaussian distributions, potentially conditioned on other variables in the diagram. A more universally applicable approach is given in [Poland and Shachter, 1993], which describes a method for letting continuous chance variables in Bayesian networks follow a distribution which is a mixture of Gaussian distributions. [Madsen and Jensen, 2003] gives a solution method for influence diagrams involving a mixture of continuous Gaussian distributed variables and discrete variables, with the structural constraint that no discrete variable can be a child of a continuous variable. Finally, [Lerner et al., 2001] introduces a technique for mixing discrete and continuous variables in a Bayesian network, using softmax functions (traditionally applied in reasoning using neural networks) as conditional probability distributions for discrete chance variables given continuous parents. The inference method they propose is exact up to the accuracy of numerical integrations performed during evaluation. Thus, an approximation.

The basic problem involved in applying these techniques for solving IDITs, is that all of them assumes continuous variables follow Gaussian distributions, or mixtures thereof. Such variables have a strictly positive density for all real numbers, which does not suit the nature of a progression of time variables, which should be guaranteed to have probability 0 for configurations where their values decrease. In other words, the probability distributions of time variables should ensure that time never regresses. Furthermore, none of the techniques listed allow continuous parents of discrete decisions, and none of them give a full description of how to allow continuous parents of discrete chance variables in influence diagrams.

The problems arising from employing continuous variables can be circumvented through discretizing the variables prior to solving the influence diagram. One such technique is given in [Kozlov and Koller, 1997]. Another approach, which circumvents the problem of using non-Gaussian distributed continuous variables, is to apply sampling methods, such as those presented in [Charnes and Shenoy, 2003]. Unfortunately, this latter approach do not solve the problem of having continuous parents of discrete decisions. A solution method for IDITs, which utilizes sampling, can be found in [Broe and Jeppesen, 2003].

In this report, we complete the representation language of IDITs into a framework. We examine IDITs in depth, which reveals a number of problems inherent in its original formulation. We then reformulate IDITs in a form that does not suffer from these problems. Following this, a set of unambiguous semantics for temporal orderings is fleshed out, and the language is enhanced to provide additional possibilities for modelling time aspects of decision problems. Building on the new temporal ordering, we furthermore define what it means for an IDIT to be welldefined and provide a method for checking this. Following this is the last result, which is an examination of the boundaries of the solution sketch given in [Broe et al., 2003], resulting in a method that solves a subset of IDITs using approximations in the form of Taylor's series and Newton's method. The solution method avoids discretizing the continuous variables in the framework and does not require sampling, at the expense of only solving a subset of IDITs.

Overview

The report is divided intro four chapters, of which this is the first, and an appendix. Chapter 2 presents the IDITs representation language in its original form and describes a set of enhancements as well as the abovementioned semantical corrections. In Chapter 3 we describe what it means to solve a decision problem and gradually adapt the general discussion into the full solution method. Finally, in Chapter 4 a conclusion of the report is given.

Following the main part of the report, Appendix A contains a brief summary of the entire report.

Notation

The topics discussed in the report are of a somewhat abstract nature, as we are dealing with mathematical models on several layers of abstraction. Consequently, the report can be heavy on notation and concepts in places, so we have provided a list of notation and an index of concepts in the back of the report. Furthermore, the report has been printed using extra line spacing to allow for mathematical expressions to be interleaved in the text with little visual impact.

Some general conventions are not described in the list of notation, and we list them here instead: All sets are printed using a bold font, such as $\boldsymbol{S}$, all decision problems and IDITs are printed using a caligraphic font, like $\mathcal{I}$, all variables are printed using normal font, such as X, and all states of variables are presented in lower case letters, e.g. x and $\vec{d}$. Whenever we refer to a set of unnamed variables of an unspecified type, we denote it $\boldsymbol{Z}$, and sets of unnamed discrete variables are denoted $\boldsymbol{D}$. Decision variables are generally denoted by D, chance variables as C, and variables of an unspecified type as X or Y. All notation are occasionally subject to subscripts or superscripts.

Chapter 2

Influence Diagrams Involving Time

In this chapter we introduce IDITs, which is a representation language constructed for representing decision problems involving quantitative measures of time. The representation language is based on that of influence diagrams, and most of the semantics is similar. It was originally proposed in [Broe et al., 2003]. We introduce it informally and describe it formally in its original form in Section 2.1. For further elaborations on the original representation language and the motivation behind it, see [Broe et al., 2003]. In Section 2.2 some alterations, which address minor shortcomings of the original representation language, are described and incorporated in the formalization of IDITs. Section 2.3 introduces a temporal ordering relation for the elements in an IDIT, and a definition of what a welldefined IDIT is.

2.1 The Original Representation Language

An IDIT is a model of a decision problem and its association to a decision taker. The model is a directed acyclic graph, whose nodes represent decision and chance variables as well as local utility functions. In this report we refer to nodes representing variables as variables and nodes representing local utility functions as utilities, when this introduce no ambiguity. For a formal introduction to the basics of graphs and explanation of graph concepts used in this report, see [Broe et al., 2003].

Introduction to IDITs

The representation language is designed to deal with decisions that span periods of time. For instance, a farmer's decision on whether to harvest his fields using a thorough method, a quick method, or not at all would span a period of time ranging from an instant to several days. Given that decisions can span periods of time, and assuming further that no two decisions can take place simultaneously, it is clear, that a decision should have associated with it a point in time, where it initiates, and its duration. Collectively, we can encode this information by, for each decision, D, of a decision problem involving time, attaching two variables: The *initiation time* of the decision, denoted $\text{init}(D)$, and the *end time* of the decision, denoted $\text{end}(D)$. The period a decision, D, spans is, thus, the variable $\text{end}(D) - \text{init}(D)$. [Broe et al., 2003] further introduces an assumption called *no-delay*, which basically states that when a decision ends the next decision initiates immediately. That is, for two decisions, D_i and D_{i+1}, where D_{i+1} is the decision presented to the decision taker after having decided on D_i, it holds that $\text{init}(D_{i+1})$ equals $\text{end}(D_i)$. In other words, there is no unexplained delay between the two decisions. If we assume that the first decision of some decision problem is taken at some predefined point in time, e.g. 0, we can, due to the no-delay assumption, omit variables representing initiation times when describing the decision problem.
Some decisions might be worth postponing for the decision taker. The farmer, for example, might postpone his decision on whether to harvest, while some laboratory examines samples of his crops to estimate its quality. Representing aspects like this is accomplished by introducing some decisions regarding possible waiting periods. In the example the farmer would be faced with two decisions: The harvest decision and a decision on whether to wait for some period before deciding on the harvest decision and, if so, for how long. Such a decision is called a *wait decision*. As the exact length of the waiting period might be clouded in uncertainty, the wait decision decomposes into the decision itself and the resulting waiting time. [Broe et al., 2003] assumes that the decision itself only affects the actual waiting time and no other aspects of the IDIT. As such, the choice taken have no effect in itself, but only through the inherent actual waiting period resulting from it. Therefore, it is called a *non-intervening* choice.
Even though we deal with decisions that can be postponed, it is important to stress that we assume that no decision can be constrained to be taken at only select moments in time. According to our perception of modelled decision processes, decisions do not just appear or disappear. Some choice is always open for taking, no matter when the decision is initiated. In some cases, this choice might simply be to do nothing, but that is still a choice. Furthermore, as IDITs are supposed to model decision processes, we disregard circumstances and events which have time spans, i.e. initiation and end times are not modelled for these. This is elaborated on

later in this section. With these preliminaries on the nature of decision problems involving time dealt with, we look deeper into the constituents of IDITs.

Chance variables are exhaustive groupings of mutually exclusive circumstances or events that lie outside the decision taker's direct control, and *decision variables*, sometimes simply called *decisions*, are exhaustive groupings of mutually exclusive actions that are directly controllable by the decision taker. *Local utility functions* are assumed to be an additive decomposition of some *total utility function*, which is a real-valued function over the configurations of the variables in the diagram, which reflects the decision taker's preferences. When specifying the utility function, this decomposition property is usually exploited, and only the local utilities are defined. The chance variables are furthermore partitioned into *time variables*, which have continuous state spaces, and the remaining chance variables, referred to as *ordinary chance variables*, which all have finite and discrete state spaces. Likewise, decision variables are divided into *wait decisions*, which have continuous state spaces, and the remaining decision variables, *ordinary decisions*, which have finite and discrete state spaces. A time variable symbolizes the end time of exactly one decision, and a wait decision symbolizes a period of waiting time. For a variable, X, its state space is denoted as $\mathbf{sp}(X)$. For a set of variables, $\boldsymbol{S}$, the Cartesian product $\times\{\mathbf{sp}(X)|X \in \boldsymbol{S}\}$ is denoted as the state space of $\boldsymbol{S}$, written $\mathbf{sp}(\boldsymbol{S})$. If a variable, X, is known to be in some state, x, we say that it is *instantiated* and write $X = x$.

In an IDIT, ordinary chance variables are depicted as circles, ordinary decisions as rectangles, utilities as diamonds, time variables as double bordered semicircles, and wait decisions as double bordered rectangles. A time variable is only allowed to be in the diagram if it is directly associated with a decision, and a decision is at most allowed to have one time variable directly associated with it. A formal clarification of what it means for a time variable to be directly associated with a decision is given in the end of this section. For now, we rely on the reader's intuition.

In order to minimize the number of arcs in the diagram, a time variable and the associated decision are drawn as an entity consisting of a rectangle and a semicircle. We present an example of a decision problem involving time and an IDIT modelling it, before discussing finer aspects of the representation language.

Example 1

The example, which is inspired by a somewhat similar example in [Broe et al., 2003], revolves around the previously introduced farmer and his crops. Whenever a variable is introduced in the example its name is shown in parenthesis following the description of its meaning, like (*This*).

At the outset of the decision problem the farmer, who we refer to as Frank, is facing harvesting season. His crops are of some quality (Qc_1), which is hard to evaluate precisely. The only hint Frank has got is the amount of weed in the field (We_1). However, he can order a test (*Te*) of his crops' quality by an external laboratory, which has specialized in this sort of task. The test takes ten days and costs \$1000 to perform. No matter if Frank takes the test, his next decision is concerned with whether he should spray (Spr) his field against weed. He can choose to decide on this straight away, based solely on his subjective estimate of the state of his crops achieved from the information on the amount of weed in the field, or he can postpone the decision, until a test result (Re) is ready.

Depending on whether he sprays or not, the decision on spraying can take some time, and even after he has completed any spraying, government imposed health regulations prohibit him from harvesting in a period of seven days after this has taken place. Thus, depending on his choice on spraying, he must decide whether to wait for a while before deciding on harvesting (Ha). Another factor that might influence that choice is the result of the test of his crops. If he decided to spray without waiting for the test result, and he is forced to wait for seven days in addition to the period of, say, one day used on spraying, he could decide to wait an additional two days to view the test result before deciding whether to harvest. Of course, Frank has the option of taking a direct look at the current level of weed in his fields (We_2), which can give him some indication of the current quality of the crops (Qc_2), but his estimate will be more precise if he knows the result of the test of the quality before spraying.

Besides the estimated quality of the crops at harvesting time, Frank accesses further information in the form of the weather forecast (Wf). If it turns out to be raining for a good deal of the forthcoming days, even the quick harvesting method might take drastically longer to complete than expected, and furthermore, the value of the crops would diminish if it gets wet.

Considering further that every other farmer in the area is trying to beat Frank to the finishing line and get their crops onto the market, while it is still a sought after commodity, Frank must, throughout all of his decisions, bear in mind that the value of his crops, no matter the quality, is inversely proportional to the point in time he can deliver it.

The structure of Frank's decision problem is modelled by the IDIT portrayed in Figure 2.1. Time is measured in days.

Strictly speaking, the diagram in the figure is not a proper IDIT as described in [Broe et al., 2003] because of the dashed arrow from the time variable next to the decision Spr to Ha and the dashed arrow from the time variable next to the decision Ha' to Ha both being present. We return to this issue in Section 2.2.

The nodes that have not been introduced this far include the three utilities C_{Te}, C_{Spr}, and R_{Cr}, which represent the cost of any test being carried out, the cost associated with any spraying, and the eventual revenue of the harvested crops, respectively. Furthermore, two wait decisions, Spr' and Ha', symbolize the time periods Frank waits before deciding on Spr and Ha, respectively. The chance variable Gw represents the global weather situation, around the time Frank chooses whether to harvest. It affects the local weather during the harvesting period (W) and the previously introduced weather forecast. The double-bordered semicircles attached to decisions Spr', Spr, Ha', and Ha represents $\text{end}(Spr') = \text{init}(Spr)$, $\text{end}(Spr) = \text{init}(Ha')$, $\text{end}(Ha') = \text{init}(Ha)$, and $\text{end}(Ha)$, respectively. $\text{init}(Te)$, $\text{end}(Te)$,

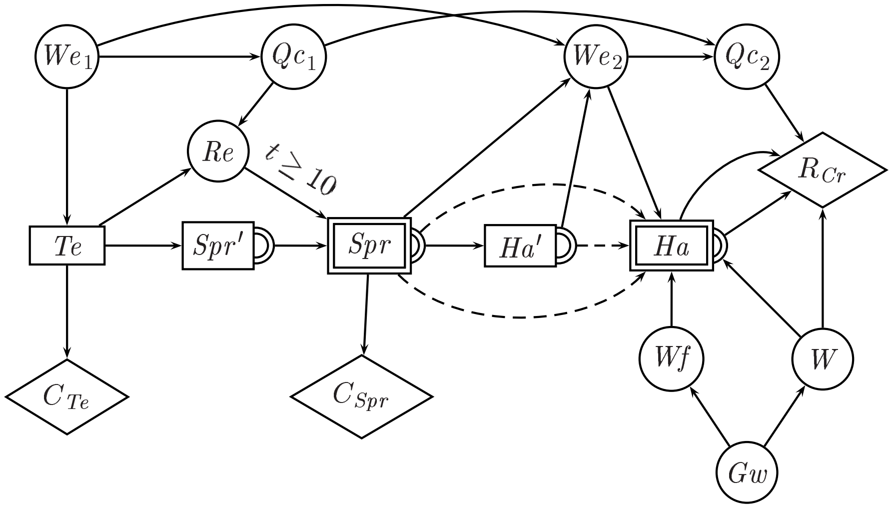

Figure 2.1: *An IDIT of the farmer's problem.*

and init(Spr') are all assumed to have the value 0, and end(Te) = init(Spr') is, therefore, not shown explicitly in the diagram. Throughout this report we refer back to this example and the variables and relationships introduced.

Semantics of IDITs

As stated previously, the time variables symbolize points in time and are each required to be associated with a decision. The semantics of the unique time variable associated with a decision is the point in time the decision has been implemented and any actions inherent in the choice chosen has been performed. If no time variable is associated with a decision, it is assumed to be taken instantaneously, and it is called an *instant decision*. Te in Example 1, for instance, is an instant decision, as it can be carried out in an instant, no matter if the choice is to order a test or to do nothing. Conversely, Ha is not an instant decision, even though it can be completed in an instant as well, by choosing not to harvest. Decisions with associated time variables we call *decisions involving time*. Wait decisions are required to be decisions involving time due to their semantics.

Arcs in an IDIT can be labelled, either dashed or solid, and represent either informational precedence, probabilistic dependencies, or functional dependencies. We go through the allowed possibilities one by one below.

A solid arc going into a decision variable represents informational precedence. That is, the state of the variable the arc emanates from is known immediately before deciding upon the decision represented by the decision variable. These kinds of arcs are called *informational arcs* and are allowed to have *guards* associated with them. A guard is a boolean function shown as a label on the arc, like it is the case on the arc from Re to Spr in Example 1. The guarded arc signifies that the variable the arc emanates from, called the *guarded* variable, is only observed when deciding upon the decision it goes into, if this decision is initiated at a point in time where the guard evaluates to **true**. As an example, Re in Example 1 is observed immediately before deciding upon Spr, only if $\text{init}(Spr)$ takes on a value greater than or equal to 10, mirroring the fact that the test takes ten days to complete.

The t referred to by a guard on an arc going into a decision, D, is always $\text{init}(D)$, and not any other points in time the observed variable happens to be probabilistically dependent on. This reflects the philosophical view that the time dependent observation or non-observation of a variable is solely a result of the point in time the observation is attempted. Other dependencies regarding observation might be thought of. One is to allow observation to hinge on configurations of other variables in the modelled decision problem. This lies outside the scope of this report, though, and the motivation for constructing the IDIT representation language in the first place. Adapting IDITs to care for these kinds of relationships might be a topic of future research. [Nielsen and Jensen, 2000] presents techniques for representing this in settings that do not involve time.

No-forgetting is assumed, which means that observed variables and decisions decided upon are remembered when deciding upon subsequent decisions. For instance, the state of We_1 in Example 1, observed when the farmer decides on Te, is remembered when deciding upon each of Spr', Spr, Ha', and Ha'. If no-forgetting was not assumed, a modeller of a decision problem would have to explicitly draw arcs from an observed variable, to every decision where it might be relevant, and the decision taker would remember it. Both requirements are not easily seen to be fulfilled: Knowing the state of a variable might allow a decision taker to choose a better choice at a decision, which seemingly have nothing to do with that variable, and if some decision problem spans several years the set of variables remembered correctly by the decision taker cannot be taken for granted. Some representation languages do not assume no-forgetting, e.g. LIMIDs[Lauritzen and Nilsson, 2001], but through explicitly assuming no-forgetting, the issues elaborated on above are avoided. In addition to no-forgetting, it is assumed that the value of time variables representing end times of decisions, which have been decided upon, are remembered at subsequent decisions.

Guarded variables are subject to a special kind of no-forgetting, called *extended no-forgetting*. Basically, every guarded arc going into a decision is “inherited” by following decisions. This means that, even if a variable is not observed at a decision, it might become observed before one of the next decisions are initiated and, thus, be reacted on. For example, *Re* in Example 1 would not be observed when deciding upon *Spr* if the initiation time of *Spr* is, say, 0. However, if Frank decides to spray with some chemical that takes four days to use and subsequently waits for the specified period of seven days before deciding upon harvesting, then *Re* will be observed immediately before deciding upon harvesting. This reflects the fact that the test result would be in Franks possession at day 11, where he initiates his decision on harvesting. Further elaboration on this topic is presented in Section 2.2.
A dashed arc going into a decision, D, from some variable, X, signifies that the state space of D is a function over the state space of X. In other words, the set of available choices at D is restricted by the value taken on by X. X is said to be in the *domain* of the *restriction function*, r_D, of D, written $X \in \mathrm{dom}(r_D)$. For instance, in Example 1, the set of available choices at *Ha* is restricted by the choice taken at *Spr* and the points in time represented by $\mathrm{end}(Spr)$ and $\mathrm{end}(Ha')$: Choosing to spray with some chemical and not waiting for the prescribed period of seven days render the choices for harvesting impossible. As the concept of restriction relies on states of the world, which are rendering certain choices impossible, it is required that no restricting variable can be unobserved. It would make little sense to be prevented from doing something, with no knowledge of why this is so. In graphical terms, no informational arcs are allowed to be both guarded and dashed.
Turning our attention from informational arcs, a solid arc going into a chance variable represents that the variable is probabilistically dependent on the variable the arc emanates from. If the parent variable is a time variable, the semantics associated with this is that the probability distribution of the chance variable varies over time. The probability distribution that is to be applied is then the one corresponding to the point in time represented by the parent time variable. Therefore, two time variables are not allowed to be parents of the same chance variable. We_2 in Example 1 is an example of a variable whose probability distribution varies over time. Only observed variables are allowed to follow probability distributions that vary over time. The reason for this is the semantics just described: The probability distribution that is to be applied corresponds to a point in time. If the variable is not observed, the point in time it is realized cannot be established uniquely, and the probability distribution that is to be applied can, consequently, not be identified.

The joint distribution of such an unknown point in time and the chance variable can be encoded by the marginal distribution for the chance variable, though.

At this point we return to the issue of events spanning periods of time, which we deemed prohibited in the beginning of this section. If an ordinary chance variable contains some state, which represents an event that spans a period of time, it would be possible to attach an initiation and end time to this variable. They would represent the point in time the event starts and the point in time it expires. However, as a chance variable is supposed to include states, which are mutually exclusive *and* exhaustive, some of the other states of the variable must represent the possibility of the event not happening. That begs the question of how we are supposed to determine an initiation time or, indeed, an end time of something that does not happen? In other words, the semantics of the initiation and end time of the chance variable seem to be defined for select states of the variable only.

This problem is not relevant for decision variables. To see why, we need to examine the nature of the two kinds of variables. Decision variables contain choices of which one must be selected by the decision taker. It makes no sense to enquire what the state of a decision variable is, at points in time prior to it being presented to the decision taker. Enquiring the state of a decision, at some point in time after the decision has been taken, is irrelevant as it remains fixed once it has been taken. Chance variables seen as representations of states of the world, can, on the other hand, always be enquired. Even if some event would happen in a given time interval, the variable takes on specific values in points in time outside of this interval. Therefore, initiation and end times of chance variables are associated with some semantical uncertainty. Far more meaningful to deal with the point in time the variable is observed and its state at this point. That is, conceive a chance variable as a snapshot of the state of the world, at the point in time it is observed. For instance, a chance variable representing the deposit on a bank account allows for no obvious initiation time nor end time. When we ask the state of such a chance variable, we are implying that what we really want to know, is the state of some variable, which represents the deposit on the account at some specific point in time, for instance the deposit on the account at January 1st. Therefore, in IDITs there is no initiation time nor end time of variables, but chance variables are allowed to be probabilistic dependent on a time variable if they are observed at a decision initiated at the point in time represented by that time variable.

The natural semantical interpretation of a guarded arc going into an ordinary chance variable, C, from some variable, X, would be that C is only probabilistic dependent on X if C is observed at points in time, at which the guard evaluates to **true**. In contrast to the semantics of guards on informational arcs, however, this information represents no structural significance to the decision process modelled by the IDIT, and the information is already found in the probability distribution of C. Using guarded arcs, in this case, is merely a visualization of a specific attribute of the probability distribution of C, namely that for some values of the parent time variable the state of C is independent of the state of X. Other attributes of the probability distribution, including independencies arising from instantiation of other variables than the time variable, seem to be equally relevant, but are not shown in influence diagrams, which IDITs are sought to be compatible with. Therefore, arcs into ordinary chance variables and utility functions are not allowed to be guarded.

Arcs into chance variables are allowed to be dashed, however, if the variable is a time variable. In that case, the arc indicates a probabilistic relationship of deterministic nature. An example could be an arc from a chance variable *Temperature* to a time variable, end(D), representing the time some decision, D, involving chemical reactions finishes. If the temperature is *low*, the time taken carrying out D would take, say, two hours more than it would, had it been *high*. To signify this predictable relationship, the arc from *Temperature* to end(D) should be dashed. A similar arrangement for arcs going into ordinary chance variables could be envisioned, but this conflicts with current standards in influence diagrams to which IDITs have been designed to be compatible. Therefore, dashed arcs into ordinary chance variables are prohibited. This topic is further discussed in Section 2.2.

Arcs into utilities indicate functional dependencies. That is, the local utility function represented by a utility node is a function over the state space of all variables that are parents of the utility. These arcs are only allowed to be solid and non-guarded. Guarded arcs are not allowed for the same reason as they are not allowed into chance variables, namely that it conflicts with the representation of influence diagrams, and that the information is already stored in the utility function itself. The same reasoning applies for not allowing dashed arcs. Of the parent variables of a utility, only one is allowed to be a time variable. If one such variable exists, it signifies that the utility takes on a different structure for each point in time, and that the actual structure is determined by the point in time represented by this time variable. R_{Cr} in Example 1 is dependent on the point in time Ha ends, for instance, in that the value of some fixed amount of crops of some fixed quality takes on different values dependent on the time it is sold.

In the graph there must exist a directed path including all decision and time variables. This path indicates the temporal ordering of these, in the sense that if a time or decision variable, X, is prior to a time or decision variable, Y, on this path, then the point in time represented by X, or $\text{init}(X)$ if X is a decision, is before or equal to the point in time represented by Y, or $\text{init}(Y)$ if Y is a decision. The diagram must be constructed, such that any pair of time variables are separated by at least one decision on this path. This is due to the previously introduced no-delay assumption.

The semantics associated with arcs described above share a common denominator: All arcs convey inter-variable structural aspects of decision problems. These kinds of aspects are called *qualitative* aspects. In contrast, we find that intra-variable aspects, such as state spaces, are not evident from the pictorial representation of the diagram. In addition to the graphical structure of an IDIT, we, therefore, define one or more *realizations* for it. A realization encapsulates some of these non-structural — also called *quantitative* — aspects of a decision problem and consists of probability distributions for chance variables, local utility functions, and restriction functions. Additional terms for qualitative and quantitative aspects are *global* and *local* aspects, respectively. Influence diagrams clearly divides qualitative and quantitative information into diagrams and realizations, and IDITs, which was designed to be compatible with influence diagrams, attempts to retain this division. A number of further restrictions apply to the topology of the IDIT, and we go through these after having set up a formal notation, as this allows us to discuss IDITs with greater precision. Realizations are also subject to restrictions that are easier understood using formal notation, and a thorough description of these is, therefore, postponed for now.

Formalization of IDITs

As described above, an IDIT, $\mathcal{I}$, is defined to be a directed acyclic labelled graph, $(\boldsymbol{W}^{\mathcal{I}}, \boldsymbol{L}^{\mathcal{I}}, \boldsymbol{E}^{\mathcal{I}})$, where $\boldsymbol{W}^{\mathcal{I}}$ consists of chance variables, decision variables, and local utility functions, $\boldsymbol{L}^{\mathcal{I}}$ is a set of labels, and $\boldsymbol{E}^{\mathcal{I}}$ is a set of arcs. The set of all chance variables in $\mathcal{I}$ is denoted as $\boldsymbol{V}^{\mathcal{I}}_C$, the set of all decision variables as $\boldsymbol{V}^{\mathcal{I}}_D$, the set of all time variables as $\boldsymbol{V}^{\mathcal{I}}_T$, the set of all wait decisions as $\boldsymbol{V}^{\mathcal{I}}_W$, and the set of all local utility functions as $\boldsymbol{V}^{\mathcal{I}}_U$. We have that $\boldsymbol{V}^{\mathcal{I}}_T \subseteq \boldsymbol{V}^{\mathcal{I}}_C$ and $\boldsymbol{V}^{\mathcal{I}}_W \subseteq \boldsymbol{V}^{\mathcal{I}}_D$. Furthermore, the set of all variables, $\boldsymbol{V}^{\mathcal{I}}_C \cup \boldsymbol{V}^{\mathcal{I}}_D$, is denoted as $\boldsymbol{V}^{\mathcal{I}}$, the set of ordinary chance variables, $\boldsymbol{V}^{\mathcal{I}}_C \setminus \boldsymbol{V}^{\mathcal{I}}_T$, as $\boldsymbol{V}^{\mathcal{I}}_{OC}$, and the set of ordinary decision variables, $\boldsymbol{V}^{\mathcal{I}}_D \setminus \boldsymbol{V}^{\mathcal{I}}_W$, as $\boldsymbol{V}^{\mathcal{I}}_{OD}$.

Thus,

$$\boldsymbol{V}_T^{\mathcal{I}} \cup \boldsymbol{V}_{OC}^{\mathcal{I}} \cup \boldsymbol{V}_W^{\mathcal{I}} \cup \boldsymbol{V}_{OD}^{\mathcal{I}} \cup \boldsymbol{V}_U^{\mathcal{I}} = \boldsymbol{W}^{\mathcal{I}},$$

where the sets on the left-hand side of the equality sign are pairwise disjoint. If the IDIT, $\mathcal{I}$, is obvious from the context we omit its name from the notation, e.g. simply write $\boldsymbol{V}_D$ instead of $\boldsymbol{V}_D^{\mathcal{I}}$.

The set of labels, $\boldsymbol{L}^{\mathcal{I}}$, consists of boolean functions having the real numbers as their domain. That is, $\boldsymbol{L}^{\mathcal{I}} \subseteq \{f | f : \mathbb{R} \to \{\mathbf{true}, \mathbf{false}\}\}$. The set of arcs, $\boldsymbol{E}^{\mathcal{I}}$, is partitioned into two disjoint sets: A set of solid arcs, $\boldsymbol{E}_s^{\mathcal{I}}$, and a set of dashed arcs, $\boldsymbol{E}_d^{\mathcal{I}}$. As for sets of variables, we omit the name $\mathcal{I}$ from the notation, when it is obvious from the context. An arc (X, Y, f) in $\boldsymbol{E}$ is to be interpreted as an arc emanating from node X going to node Y labelled with the function f. Arcs labelled with the constant function **true** are drawn with no label for sake of clarity. Arcs labelled with the constant function **false** are semantically equivalent to the absence of an arc, and are, therefore, not drawn in the diagram.

The placement of arcs labelled with non-constant functions is restricted to informational arcs. That is, if (X, Y, f) is in $\boldsymbol{E}$, and $f(t_1) \neq f(t_2)$ for two distinct t_1 and t_2 in $\mathbb{R}$, then Y must be in $\boldsymbol{V}_D$. Additionally, dashed arcs are only allowed going into decision or time variables. Thus, if (X, Y, f) is in $\boldsymbol{E}_d$, then Y must be in $\boldsymbol{V}_D \cup \boldsymbol{V}_T$.

The set of all parents of a node, X, i.e. the nodes from which an arc that goes into X emanates, we denote as $\mathbf{pa}(X)$, and the set of children, i.e. the nodes from which an arc emanating at X goes into, as $\mathbf{ch}(X)$. The set of all parents connected to a node, X, with dashed arcs we denote $\mathbf{pa}_d(X)$.

The previously mentioned temporal order of decisions and time variables is extended to an ordering relation, which imposes a partial order on all variables, denoted $\prec$. For any pair of time or decision variables, X and Y, X is temporally prior to Y, written $X \prec Y$, if and only if there is a path from X to Y. As stated previously, the ordering of time and decision variables induced from the diagram is required to be a total ordering. This defined ordering suffers from some flaws, all associated with guarded arcs. For instance, when guards are not fulfilled, and arcs consequently are perceived as not being present, we might experience a situation, where there is, in effect, no directed path between two decisions, and the ordering, thus, fails to emerge. We look further into these problems in Section 2.3. For now, we disregard these aspects and further state that for any ordinary chance variable, X, and some decision or time variable, Y, $X \prec Y$ if and only if (X, Y, f) is in $\boldsymbol{E}$, for some f in $\boldsymbol{L}$, or there exists some decision or time variable, Z, such that $X \prec Z$ and $Z \prec Y$. Furthermore, if an ordinary chance variable, X, is not a parent of any decision or

time variable, then $Y \prec X$, for any time or decision variable, Y. This refinement is subject to further discussion in Section 2.3 as well.
Some structural constraints need to be fulfilled for a graph to qualify as an IDIT:

- No node is allowed to have more than one time variable as parent, i.e. $|\mathbf{pa}(X) \cap \boldsymbol{V}_T| \leq 1$, for all X in $\boldsymbol{W}$. This restriction reflects that no variable or utility can be observed or realized at more than one point in time.

- A node has no children, if and only if it represents a utility. That is, $\mathbf{ch}(U) = \varnothing$ iff $U \in \boldsymbol{V}_U$. This requirement is similar to what is usually required of nodes in influence diagrams and seeks to prevent barren nodes and children of utility nodes, the latter having no clear semantical interpretation. Barren nodes are variables that influence no other part of the decision problem. They are sometimes included in models of decision problems in order to render the problem easier understood by people with preconceived notions of the mechanics underlying the problem. They are, when all is said and done, irrelevant to a solution method, such as the one presented in this report, though.

- There should exist a path, $(X_1, \ldots, X_n)$, in the diagram, such that $\boldsymbol{V}_D \cup \boldsymbol{V}_T \subseteq \{X_1, \ldots, X_n\}$. This path ensures that the temporal ordering, $\prec$, is a total ordering over all time and decision variables, but as mentioned above, this is subject of further discussion in Section 2.3.

- Each time variable must be a child of some decision variable. That is, if $T \in \boldsymbol{V}_T$ then $|\mathbf{pa}(T) \cap \boldsymbol{V}_D| \geq 1$. This structural requirement stems from the no-delay assumption introduced earlier in this section. If more decisions are parents of the same time variable, the maximal one, with respect to $\prec$, is the decision whose end time is represented by the time variable. The time variable is said to be directly associated with this decision. The remaining parent decisions are conditionals for the probability distribution of this end time variable.

- A wait decision must have exactly one child variable, and that variable must be a time variable, i.e. if $D \in \boldsymbol{V}_W$, then $\mathbf{ch}(D) = \{T\}$ and $T \in \boldsymbol{V}_T$. This requirement is meant to restrict the possible impact, on the variables in the rest of the diagram, of what is perceived as a non-intervening decision.

- There must be a dashed arc between any two time variables, which are consecutive in the order obtained from applying $\prec$ to the set of time variables. That is, for any $T_i, T_j \in \boldsymbol{V}_T$, where $T_i \prec T_j$, and there is no $T_k \in \boldsymbol{V}_T$, such that $T_i \prec T_k \prec T_j$, the arc $(T_i, T_j, \mathbf{true})$ must be in $\boldsymbol{E}_d$. This requirement is meant to reflect that a time variable cannot take on a value that is lower than the one the time variable before it did, or, in other words, that the time modelled always progresses and never regresses. Of course the structure only communicates that restrictions between time variables are in place. The actual restrictions, ensuring this progression of time, must be defined in the probability distributions of the variables. We call an arc from a time variable to a time variable a *temporal* arc.

- An ordinary chance variable is only allowed to have a time variable as parent, if it is observed when deciding upon a decision, which initiates at the point in time represented by this time variable. Formally, if $C \in \boldsymbol{V}_{OC}$, and there exists some $T \in \mathbf{pa}(C)$, where $T \in \boldsymbol{V}_T$, then there exists some D in $\mathbf{ch}(C)$, where $D \in \boldsymbol{V}_D$ and $\text{init}(D) = T$. The need for a unique "trigger point" for ordinary chance variables to be dependent on time, described previously, is the reason for this requirement.

- An arc is not allowed to be both dashed and guarded. Formally, if (X, Y, f) is in $\boldsymbol{E}_d$, then $f(t_i) = f(t_j)$, for all t_i and t_j in $\mathbb{R}$. The reasoning for this is the understanding that a variable, which can restrict a decision, cannot be unobserved or observed after the decision has been taken, since this could lead to paradoxes, as described above.

Given this formal syntax of IDITs and the above list of structural requirements the keen-eyed reader might protest that the IDIT pictured in Figure 2.1 is not really an IDIT. For instance, there are no temporal arcs, and there is no path through all time and decision variables. This is due to it being shown in its *compressed form*, as opposed to the *blown-up* version the formal syntax describes. The compressed form of an IDIT is a result of exploitation of two observations, namely that each time variable should have a decision as parent, and that the temporal arc from one time variable to the next is always present. By pictorially attaching all time variables to the decisions whose end times they represent no information is lost, as if the time variable was to be "ripped" from its parent decision, there would, in every case, be an arc from the decision to the time variable. Additionally, as the arcs between consecutive time variables are required to always be present, consistently and conventionally omitting them results in no information loss. Furthermore, the coupling of a time variable to the decision whose end time it represents, emphasizes the strong conceptual bond between these.

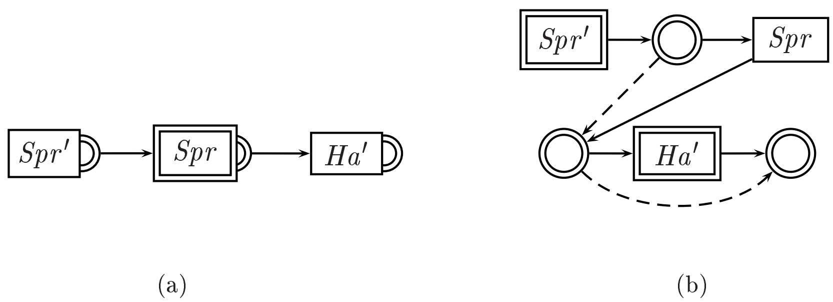

Figure 2.2: *a) Compressed form of an IDIT. b) Blown-up version of the same IDIT.*

Figures 2.2(a) and 2.2(b) show two versions of a part of the IDIT in Figure 2.1. The one in Figure 2.2(a) corresponds to the compact form, whereas the one in Figure 2.2(b) shows what the blown-up version would look like. The figures should convey the difference in clarity of the two schemes.

As mentioned earlier, each IDIT allows for one or more realizations. A realization for an IDIT, $\mathcal{I}$, is a four tuple, $(\mathbf{\Phi}^{\mathcal{I}}, \mathbf{\Psi}^{\mathcal{I}}, \mathbf{\Pi}^{\mathcal{I}}, \mathbf{\Gamma}^{\mathcal{I}})$, where the elements of $\mathbf{\Phi}^{\mathcal{I}}$ are probability distributions, the elements of $\mathbf{\Psi}^{\mathcal{I}}$ are local utility functions, the elements of $\mathbf{\Pi}^{\mathcal{I}}$ are density functions, and the elements of $\mathbf{\Gamma}^{\mathcal{I}}$ are restriction functions. More precisely, for each node, X, in $\boldsymbol{W}^{\mathcal{I}}$

- a conditional probability distribution $P(X|\mathbf{pa}(X))$ is in $\mathbf{\Phi}^{\mathcal{I}}$ iff $X \in \boldsymbol{V}^{\mathcal{I}}_{OC}$,
- a local utility function $u_X : \mathbf{sp}(\mathbf{pa}(X)) \hookrightarrow \mathbb{R}$, where $\hookrightarrow$ denotes a partial function, is in $\mathbf{\Psi}^{\mathcal{I}}$ iff $X \in \boldsymbol{V}^{\mathcal{I}}_{U}$,
- a density function $f_X : \mathbf{sp}(\mathbf{pa}(X)) \times \mathbb{R} \to [0; \infty[$, where $\int_{-\infty}^{\infty} f(\vec{c}, x)dx = 1$, for all $\vec{c}$ in $\mathbf{sp}(\mathbf{pa}(X))$, is in $\mathbf{\Pi}^{\mathcal{I}}$ iff $X \in \boldsymbol{V}^{\mathcal{I}}_{T}$, or
- a restriction function $r_X : \mathbf{sp}(\mathbf{pa}_d(X)) \hookrightarrow 2^{\mathbf{sp}(X)} \setminus \{\varnothing\}$ is in $\mathbf{\Gamma}^{\mathcal{I}}$ iff $X \in \boldsymbol{V}^{\mathcal{I}}_{D}$.

As for other sets, we omit the name of the IDIT in the notation if it is obvious from the context. It is worth noticing that, when specifying a realization for an IDIT, some configurations of parent variables for some decision or utility might be impossible. Consequently, the restriction or utility function value corresponding to these configurations can be difficult to specify by a modeller, and we, therefore, allow these functions to be partial.

The intuition behind a density function for a time variable, T, is that it, for

any configuration of $\mathbf{pa}(T)$, is a density function for T over the real numbers. A restriction function for a decision, D, for any configuration of $\mathbf{pa}(D)$, yields the possible choices when deciding upon D. Even though the guards on arcs can contain numerical attributes they are not seen as part of a realization, as their semantic is of a structural nature.
For a realization to make any sense, a restriction functions is required to never result in the empty set. That is, when deciding upon a decision, no matter the configuration of the parent variables, *some* choice is always possible. Furthermore, density functions are required to take on the value 0 for points in time, which precede the point represented by the unique parent time variable. That is, time progresses and never regresses.

2.2 Alterations of the Original Framework

Some aspects of IDITs, as introduced in Section 2.1, are not fully desirable, and in this section we, therefore, propose a set of alterations to the original representation language and its interpretation. The motivation for each alteration is presented along with the alteration proposal itself. As the original requirements on IDITs are modified, or new requirements are added, we state it in clearly marked **Requirement**'s. Each requirement assumes the existence of a labelled graph, $\mathcal{I} = (\boldsymbol{W}^{\mathcal{I}}, \boldsymbol{L}^{\mathcal{I}}, \boldsymbol{E}^{\mathcal{I}})$, as described in Section 2.1. Similarly, when we introduce concepts, which are referenced in the remainder of the report, we do so in clearly marked **Definition**'s.
None of the alterations presented in this section are required for IDITs to be a functioning representation language, but they are included as they increase the expressive powers or decrease the level of inconsistency in it. Alterations that actually fixes flaws in the semantics of the originally proposed representation language are presented in Section 2.3.

Presence of Dashed Arcs

The first alteration we propose is dropping the convention of drawing arcs into time variables dashed, if the parent variable has a functional influence on the time variable.

The reason for doing this is two-fold: First, the dashed arcs impose restrictions on the probability distributions for time variables stored in the realization. The distinction between qualitative and quantitative aspects is thus blurred, and an IDIT and its realization are tied closer together than necessary. Second, to ensure consistency arcs into ordinary chance variables that represent deterministic relationships would also have to be dashed. But that would conflict with the conventional semantics of influence diagrams, leaving IDITs incompatible. Hence, we allow only dashed arcs to go into decisions.

Requirement 1
Arcs which are dashed or labelled with a non-constant function may only go into a decision node. That is, if (X, Y, f) *is in* $\boldsymbol{E}_d^{\mathcal{I}}$*, or* $f(t_i) \neq f(t_j)$ *for distinct real numbers,* t_i *and* t_j*, then* Y *is in* $\boldsymbol{V}_D^{\mathcal{I}}$*.*

Realization Time Variables for Utilities

The second alteration stems from the observation that utilities, which take on values depending on the specific points in time they are realized, can in IDITs only be modelled if the moment of realization coincides with the end time of some decision. This might not always be the case, as can be seen by considering some financial utility, payed by a mailed check, which is not cashed until some time after the decision, which triggered the utility, ended. We remedy this, by allowing utility nodes to have associated their "own" time, in effect imposing an uncertainty on the value of utilities. We call these points in time *realization times* of the utilities and draw them in IDITs as semicircles attached to the utility nodes they are associated with. Semantically, they correspond to groupings of points in time where a utility might be realized, just like end time variables represent groupings of points in time where decisions end. We distinguish between the two kinds of variables by specifically referring to a variable representing the point in time a utility is realized as a *realization time variable*, or simply realization time, though. Like time variables, realization times must have probability distributions specified for them, and these can be parameterized by other variables. This is shown in the IDIT by drawing solid arcs from the affecting variables to the realization time. See Figure 2.3 for a depiction of a utility dependent on time, U, with its own realization time node, $\mathrm{real}(U)$. $\mathrm{real}(U)$ is affected by both $\mathrm{end}(D)$ and C_1, while U is a function over $\mathrm{real}(U)$ and C_2.

If the realization time of some utility always coincides with the end time of some

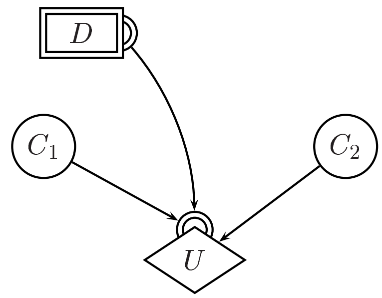

Figure 2.3: *A utility with its own realization time.*

decision, we leave out the semicircle and simply draw an arc from the end time of the decision to the utility, as described in Section 2.1. If the utility does not depend on time at all, we connect no time variable to it as parent.

We denote the set of all realization time variables in an IDIT, $\mathcal{I}$, as $\boldsymbol{V}_R^{\mathcal{I}}$ or, if the IDIT is obvious from the context, simply as $\boldsymbol{V}_R$.

As the point in time a utility, U, is realized, is modelled as a variable, $\text{real}(U)$, it is natural to enquire, whether variables and other utilities can depend on it, i.e. if other nodes than U are allowed to be a descendent of $\text{real}(U)$. In this report we choose not to allow this. First of all, we do not allow some decision to be a descendant of $\text{real}(U)$, as it, depending on the associated realizations, might introduce antinomies with regards to time. These antinomies arise, if both a time variable representing an end time of a decision, D_2, and a time variable representing the realization time of U are located along paths from one decision, D_1, to another, D_3, as shown in Figure 2.4. In that case we cannot uniquely determine, which of the variables that should act as initiation time of D_3, and even if we, consistently, always choose either the former or latter, some configurations of $\text{real}(U)$ and $\text{end}(D_2)$ would yield D_3 either initiating before D_2 ends, or $\text{real}(U)$ representing a point in time after D_3 is initiated, but still known immediately before it initiates. Both scenarios are antinomic. Furthermore, if we were to choose the variable representing the point farthest in time on a case by case basis, we would, in situations where $\text{real}(U)$ is interpreted as $\text{init}(D_3)$, violate the no-delay assumption, as there would be an unaccounted for delay in the decision process from ending D_2 to initiating D_3.

Allowing some ordinary chance variable, C, to be a child of a realization time, $\text{real}(U)$, is also prohibited, as this would, owing to the discussion in Section 2.1, require C to be observed. However, if C is observed, then some decision node must be a child of C and, consequently, a descendant of $\text{real}(U)$, which is undesirable due

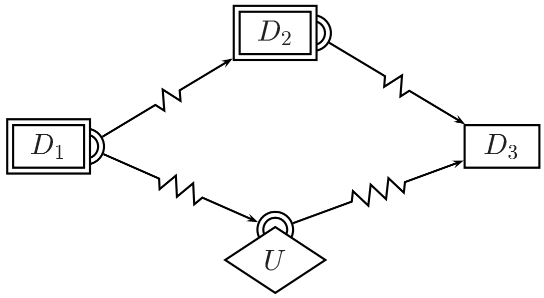

Figure 2.4: *An unattractive consequence of allowing decisions as descendants of utility realization times. The zig-zags on the arcs signal that there might be some intermediate nodes on the path between the node the arc emanates from and the node it goes into.*

to the reasons given above.

Finally, we might envision some utility node, U', other than U, being a child of real(U). Semantically, this would mean that both U and U' are realized at the point in time symbolized by real(U). But why attach real(U) to U and not U', then? Indeed, we could achieve a more balanced representation, of such a shared realization time, by tearing real(U) from U and representing it as a full double circle connected as a parent to both U and U'. This is not consistent with the approach used for representing end times of decisions in the rest of the diagram, though, and although the semantics of the two kinds of variables are different, we feel that the conceptual bond between a utility and its realization time is as relevant as that found between a decision and its end time. Therefore, we continue to draw realization times attached to utility nodes as semicircles, and simply abstain from connecting other utilities as children of the realization time.

Requirement 2

Realization time variables are only allowed to have one child, which must be a utility node. They are required to have one and only one time variable as parent. Formally, if real(U) is in $\boldsymbol{V}_R^{\mathcal{I}}$ *then* $\mathbf{ch}(real(U)) \subseteq \boldsymbol{V}_U^{\mathcal{I}}$, $|\mathbf{ch}(real(U))| = 1$, *and* $|\mathbf{pa}(real(U)) \cap \boldsymbol{V}_T^{\mathcal{I}}| = 1$.

Time Variables as Parents of Decisions

The third alteration to the structure of IDITs is that we, henceforth, allow decisions to have more than one time variable as parent. IDITs, as they were described in Section 2.1, prohibit all variables from having more than one time variable as parent, as this could lead to confusion on which time variable that represents the initiation time or instantiation time of the decision or chance variable, respectively. The restriction is of a pedagogical nature when applied to arcs going into decision variables, though, as the requirement on a directed path through all decision and time variables ensures that the initiation time of a decision can be deduced from the diagram, even if more time variables are parents of the decision.
Having several time variables as parents of one decision would, when no-forgetting is assumed, be useless if they were all connected with solid arcs. In Figure 2.1, however, the variable Ha is a child of both end(Spr) and end(Ha'). The reason why it is attractive to have several time variables as parents of one decision, even if the extra arcs, due to no-forgetting, seems redundant, thus becomes clear: Some restriction functions might vary according to the stretch in time between two decisions. Even though the states of the time variables end(Ha') and end(Spr) are remembered at decision Ha, the restricting effect can only be conveyed to a reader by drawing the dashed arcs. Consequently, from now on we allow decisions to have more than one time variable as parent, even if a set of parent time variables, with more than one connected with a solid arc, is redundant information.

Requirement 3
No chance or utility node can have more than one time variable as parent. That is, if X is in $\boldsymbol{V}_C^{\mathcal{I}} \cup \boldsymbol{V}_U^{\mathcal{I}}$, then $|\mathbf{pa}(X) \cap (\boldsymbol{V}_T^{\mathcal{I}} \cup \boldsymbol{V}_R^{\mathcal{I}})| \leq 1$.

Inheritance of Guarded Arcs

While on the subject of informational arcs we comment on the nature of guarded arcs, which gives rise to the fourth alteration of IDITs. Guarded arcs, as explained in Section 2.1, are inherited by decisions following the decisions the arcs go into. This is the sane approach if an arc is guarded with a boolean function, which yields **false** for initial points in time, but from some point in time starts yielding **true**, like it is the case in Example 1, where the result of a test is unavailable initially, but becomes available later on. By allowing arcs to be inherited by later decisions, the guard is evaluated once for each decision, and the guarded variable, thus, gets more chances of being observed. However, applying the same reasoning, if the guard is of an inverse nature, i.e. it evaluates to **true** for early points in time but **false** for later ones, simply inheriting the arc would mean that, somehow, the guarded variable becomes unobserved as time progresses. Even if the phenomenon represented by the guarded variable becomes physically unobservable, we might assume that, if it was observable previously, its state can be remembered. Therefore, we alter the semantics of guarded arcs: We still interpret a guarded arc into a decision, D, to mean that the guarded variable, X, is observed immediately before deciding on D, provided that the guard evaluates to **true** at $\text{init}(D)$. However, for any decision, D_i, that is a descendant of D, we define X to be observed immediately before, D_i, if either the guard evaluates to **true** at $\text{init}(D_i)$, or it evaluated to **true** for $\text{init}(D_j)$, where D_j is D or some ancestor of D_i and descendant of D. In other words, once some variable is observed, it stays observed, even if the circumstances allowing for the observation expires.

Varying Ordering of Decisions

A further alteration to the structure of the representation language concerns the requirement on the graph to be acyclic. We now abandon the requirement for the IDIT to be a directed acyclic graph and allow the graph to contain cycles under special circumstances. This alteration causes problems for the previously introduced temporal ordering of variables, $\prec$, as this was heavily based on the acyclic property of IDITs. But, as were also mentioned, this ordering is subject to some other flaws, and we, therefore, disregard it for the moment and return to the matter in Section 2.3.
Allowing cycles is attractive as it allows for specification of sets of decisions that are not necessarily taken in a predetermined order, but according to the point in time they, as a group, are initiated. An example should clarify this: The IDIT in Figure 2.5 contains two decisions, D_2 and D_3, which are taken either in the order D_2 then D_3 or in the order D_3 then D_2. The determining factor is what time the wait decision D_1 ends: If $\text{end}(D_1)$ is less than 10, then D_3 is taken prior to D_2, whereas if $\text{end}(D_1)$ is greater than or equal to 10, then D_2 is taken prior to D_3. This is seen from the guards on the arcs forming the cycle (D_2, D_3, D_2). When the decision taker is done taking D_1, the guards on arcs going into the next decision, which in this case is either D_1 or D_2, are evaluated, and arcs with guards that evaluate to **false**

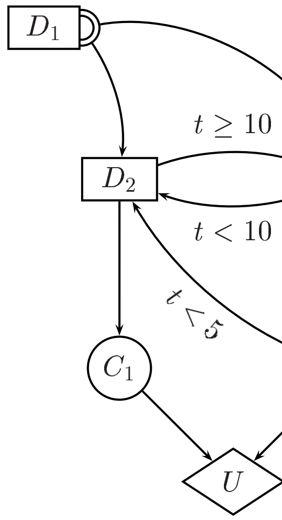

Figure 2.5: *An example of two decisions that are not taken in a predefined order.*

are considered to be non-existent, as described in Section 2.1. No matter what point in time end(D_1) represents, exactly one of the guards on the two arcs evaluates to **true**. The cycle is thus "broken", and the ordering of D_2 and D_3 is evident from the resulting diagram.

Two key observations regarding this arrangement should be noted, though. First, none of the decisions involved in the cycle is a decision involving time. If one of the decisions, say D_2, had been a decision involving time, it would not be clear what point in time, end(D_1) or end(D_2), the guard $t \geq 10$ refers to: If D_3 initiates before D_2, then t would refer to end(D_1), and if D_3 initiates after D_2, then it would refer to end(D_2). But as we do not know whether D_3 initiates before D_2, until the guards on the arcs are evaluated, the guards cannot be evaluated, and a seemingly inextricably problem thus arises. The second key observation is that the guards on the two arcs are mutually exclusive and exhaustive, thereby guaranteeing that the cycle is broken before any of its constituent decisions are decided on.

Three approaches to cycles, which honour these two observations, are

- either to disallow cycles and thereby varying decision orderings,
- to allow cycles involving instant decisions only, or
- instead of using t, use some other notation, such as end(D).t, to signify what time variable each guard is referring to.

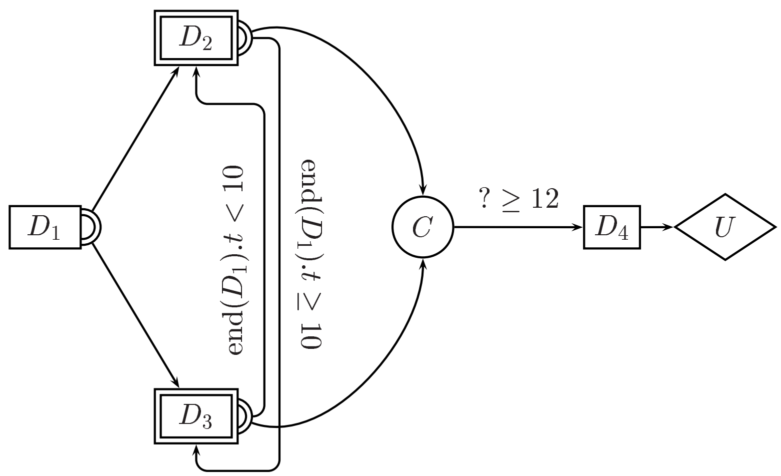

Figure 2.6: *Problems arising from using more elaborate notation.*

The first approach we dismiss as it limits functionality, and the second approach we treated in the previous paragraphs. The third, however, seems to be the most powerful approach, as it puts little restriction on the constructs which can be modelled. However, we take it that the approach would severely damage clarity of IDITs and at the same time cripple some of the flexibility of the language. This last point can be seen by studying a modified version of the IDIT pictured in Figure 2.5, using the more elaborate guards suggested as approach number three. The new IDIT is pictured in Figure 2.6. Clearly, no ambiguities arise, when determining whether the guards on the arcs connecting D_2 and D_3 evaluate to **true**, but the guard on the arc between C and D_4 is problematic. The variable that should replace the question mark is either $\text{end}(D_2).t$ or $\text{end}(D_3).t$, depending on which of the two decisions, D_2 or D_3, that is taken first. However, we do not know, which it is, until D_1 has completed, and certainly not when we draw the diagram in the first place. The problem could be remedied by using $\text{init}(D_4)$, but this variable is not drawn explicitly in the diagram, and we believe that referring to it in guards would, therefore, lead to confusion on behalf of the reader.

Another problem, related to using more elaborate notation on guards, is the process of determining, whether a cycle is guaranteed to be broken before any of its constituent decisions are decided on. Assume for a moment that, in Figure 2.6, the guard from D_2 to D_3 was specified as $\text{end}(D_0).t \geq 10$, and that D_0 is some decision prior to D_1. Whether the cycle would be broken, when D_1 ends, is now less clear than it was before. If the two time variables, $\text{end}(D_0)$ and $\text{end}(D_1)$, were related, such that, in addition to the previously mentioned requirement on time not regressing, we had that

$$\text{end}(D_0) < 10 \Rightarrow \text{end}(D_1) < 10,$$

for instance, the cycle would be guaranteed to be broken. But this would not be evident from the diagram. Taking the thought experiment even further we might imagine cycles encompassing a sizeable number of decisions, in which the guards would refer to lots of variables, and consequently, few modellers would be able to distinguish a legal cycle from an illegal one. It might be possible to automate the process of checking whether a cycle is guaranteed to be broken, even if the guards in it refer to several time variables. However, this would call for analysis and comparisons of density functions and is outside the scope of this report.
Concluding on this discussion we settle on allowing cycles involving instant decisions only and shun the elaborate notation mentioned in the previous paragraph. At this point we also briefly touch upon the subject of probabilistic guards. That is, guards which evaluates to **true** with some probability, and not deterministically, given its parameters, e.g. $t \leq A, A \sim N(6, 2)$. Such guards would have to be prohibited from appearing in cycles, as the cycles could not be guaranteed to be broken at any point in time. We do not touch upon probabilistic guards in the remainder of this report.

Requirement 4
A cycle must consist only of instant decisions and ordinary chance variables. That is, if there is a path from a variable X to X, then $X \in \boldsymbol{V}^{\mathcal{I}}_{OD} \cup \boldsymbol{V}^{\mathcal{I}}_{OC}$, and $\mathbf{ch}(X) \cap \boldsymbol{V}^{\mathcal{I}}_{T} = \varnothing$.

Definition of IDITs

In conclusion we define the IDIT and its realizations. A number of requirements are left untouched from Section 2.1, and we list them here for convenience.

Requirement 5
There must be a temporal arc between any two time variables following each other in the temporal order. That is, there exists a path, $(T_1, \ldots, T_n)$, through all time variables, $\boldsymbol{V}^{\mathcal{I}}_{T} = \{T_1, \ldots, T_n\}$, indicating the temporal ordering of these.

Note that temporal arcs, which can be deduced from the rest of the diagram, are not shown in the compact form of IDITs shown in most figures in this report.

Requirement 6
There must be a directed path through all decision and time variables. That is, there must exist a path, $(X_1, \ldots, X_n)$, *in* $\mathcal{I}$ *such that* $\boldsymbol{V}_T^{\mathcal{I}} \cup \boldsymbol{V}_D^{\mathcal{I}} \subseteq \{X_1, \ldots, X_n\}$.

Requirements 4 and 6 guarantee a total ordering of non-instant decisions with respect to time variables.

Requirement 7
A node has no children if and only if it is a utility node, i.e. for any node X *in* $\boldsymbol{W}^{\mathcal{I}}$ *we have that* $\mathbf{pa}(X) = \varnothing$ *iff* X *is in* $\boldsymbol{V}_U^{\mathcal{I}}$.

Requirement 7, thus, handles barren node removal as described in Section 2.1.

Requirement 8
Each time variable must be a child of at least one decision. That is, if T *is in* $\boldsymbol{V}_T^{\mathcal{I}}$, *then* $|\mathbf{pa}(T) \cap \boldsymbol{V}_D^{\mathcal{I}}| \geq 1$.

The reason that each time variable needs a decision as parent is that their semantical interpretation is to be end times of decisions. Hence, it makes no sense to talk about time variables with no association to a decision.

Requirement 9
Each wait decision has only one child, and that child is a time variable. More formally, we have that if D *is in* $\boldsymbol{V}_W^{\mathcal{I}}$, *then* $\mathbf{ch}(D) = \{T\}$, *where* T *is in* $\boldsymbol{V}_T^{\mathcal{I}}$.

As stated in Section 2.1, this is because we perceive a wait decision as a non-intervening decision, which can only affect other parts of the decision problem through the actual time spent waiting.

Requirement 10
An ordinary chance variable is only allowed to have a time variable as parent if it is observed immediately before a decision, which initiates at the point in time represented by the time variable. In other words, if there is a T *in* $\mathbf{pa}(C) \cap \boldsymbol{V}_T^{\mathcal{I}}$, *for some* C *in* $\boldsymbol{V}_{OC}^{\mathcal{I}}$, *then there is a* D *in* $\mathbf{ch}(C) \cap \boldsymbol{V}_D^{\mathcal{I}}$ *such that* $init(D) = T$.

The reasoning for this was elaborated on in Section 2.1.
Given these requirements, we define IDITs, as they are used in the rest of the report.

Definition 2.1

Let $\mathcal{I} = (\boldsymbol{W}^{\mathcal{I}}, \boldsymbol{L}^{\mathcal{I}}, \boldsymbol{E}^{\mathcal{I}})$ *be a directed labelled graph, whose nodes,* $\boldsymbol{W}^{\mathcal{I}}$*, consist of ordinary chance variables,* $\boldsymbol{V}^{\mathcal{I}}_{OC}$*, ordinary decisions,* $\boldsymbol{V}^{\mathcal{I}}_{OD}$*, wait decisions,* $\boldsymbol{V}^{\mathcal{I}}_{W}$*, time variables,* $\boldsymbol{V}^{\mathcal{I}}_{T}$*, realization time variables,* $\boldsymbol{V}^{\mathcal{I}}_{R}$*, and utility functions,* $\boldsymbol{V}^{\mathcal{I}}_{U}$*. Furthermore, let the set of labels,* $\boldsymbol{L}^{\mathcal{I}}$*, consist of boolean functions over the real numbers, i.e.* $\boldsymbol{L}^{\mathcal{I}} \subseteq \{f | f : \mathbb{R} \to [\mathbf{true}, \mathbf{false}]\}$*, and the set of edges,* $\boldsymbol{E}^{\mathcal{I}}$*, be partitioned into a set of solid edges,* $\boldsymbol{E}^{\mathcal{I}}_{s}$*, and a set of dashed edges,* $\boldsymbol{E}^{\mathcal{I}}_{d}$*. If* $\mathcal{I}$ *conforms to Requirements 1 to 10, then* $\mathcal{I}$ *is an* IDIT*.*

Additionally, the realizations we will deal with are defined as following.

Definition 2.2

Let $\mathcal{I}$ *be an IDIT. Then the four tuple,* $(\boldsymbol{\Phi}^{\mathcal{I}}, \boldsymbol{\Psi}^{\mathcal{I}}, \boldsymbol{\Pi}^{\mathcal{I}}, \boldsymbol{\Gamma}^{\mathcal{I}})$*, where the elements of* $\boldsymbol{\Phi}^{\mathcal{I}}$ *are probability distributions, the elements of* $\boldsymbol{\Psi}^{\mathcal{I}}$ *are local utility functions, the elements of* $\boldsymbol{\Pi}^{\mathcal{I}}$ *are density functions, and the elements of* $\boldsymbol{\Gamma}^{\mathcal{I}}$ *are restriction functions, is a* realization *of* $\mathcal{I}$ *if for each node,* X*, in* $\boldsymbol{W}^{\mathcal{I}}$

- *a conditional probability distribution* $P(X|\mathbf{pa}(X))$ *is in* $\boldsymbol{\Phi}^{\mathcal{I}}$ *iff* $X \in \boldsymbol{V}^{\mathcal{I}}_{OC}$*,*
- *a local utility function* $u_X : \mathbf{sp}(\mathbf{pa}(X)) \hookrightarrow \mathbb{R}$ *is in* $\boldsymbol{\Psi}^{\mathcal{I}}$ *iff* $X \in \boldsymbol{V}^{\mathcal{I}}_{U}$*,*
- *a density function* $f_X : \mathbf{sp}(\mathbf{pa}(X)) \times \mathbb{R} \to [0; \infty[$*, where* $\int_{-\infty}^{\infty} f(\vec{c}, x)dx = 1$*, for all* $\vec{c}$ *in* $\mathbf{sp}(\mathbf{pa}(X))$*, is in* $\boldsymbol{\Pi}^{\mathcal{I}}$ *iff* $X \in \boldsymbol{V}^{\mathcal{I}}_{T} \cup \boldsymbol{V}^{\mathcal{I}}_{R}$*, and*
- *a restriction function* $r_X : \mathbf{sp}(\mathbf{pa}_d(X)) \hookrightarrow 2^{\mathbf{sp}(X)} \setminus \{\varnothing\}$ *is in* $\boldsymbol{\Gamma}^{\mathcal{I}}$ *iff* $X \in \boldsymbol{V}^{\mathcal{I}}_{D}$*.*

As can be seen, the only difference, between Definition 2.2 and the one described in Section 2.1, is the allowance for realization time variables, to which density functions are associated.

2.3 Temporal Ordering in IDITs

In Section 2.1, when the temporal ordering relation, $\prec$, was introduced, we briefly commented that it suffers from some flaws in conjunction with guards on arcs. We further allowed cycles in IDITs in Section 2.2 resulting in even more strains on $\prec$. In this section we explain in detail, why the original ordering relation is not sufficient for reading IDITs, and propose a new ordering relation for nodes in IDITs. Furthermore, we define what it means for an IDIT to be welldefined and provide a

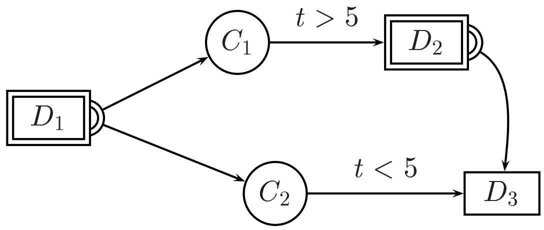

Figure 2.7: *An example of problems related to the ≺-relation.*

method for checking this.

Why ≺ fails

We start of with highlighting the flaws of ≺. In the example IDIT pictured in Figure 2.7 they are prevalent. According to ≺, the decision problem modelled by this diagram seems to make little sense: If we try to list the variables according to ≺, we get the total ordering

$$D_1 \prec \text{end}(D_1) \prec C_1 \prec D_2 \prec \text{end}(D_2) \prec C_2 \prec D_3,$$

which conflicts with the intuitive notion obtained from the diagram that C_1 can only be observed at points in time after 5, whereas C_2 can only be observed prior to this. Furthermore, if taking D_1 ends at, say, time 2, we seem to encounter a situation where which decision is next is undefined. On the one hand there is a directed path from D_2 to D_3, which, according to ≺, means that decision D_2 should be taken prior to decision D_3. On the other hand the semantics of a guarded arc, whose guard evaluates to **false**, is equivalent to a non-existing arc. Thus, as we, from the directed path with no intermediate time or decision variables from end(D_1) to D_2, can deduce that init(D_2) is equivalent to end(D_1), we know that the guard $t > 5$ is not fulfilled, and consequently, the very same directed path, which allowed us to reach this conclusion, ceases to exist. Choosing D_3 as the next decision instead is not a solution, even though the path from D_1 to D_3 continues to be there when the guard $t < 5$ is evaluated. This is because of there still being a directed path from D_2 to D_3, stating that D_2 should be taken before D_3. However, if we assume that D_1 has been taken and instantiate end(D_1) to some value, we can disregard them and conclude that, even though the directed path from D_1 to D_2 is broken, because there is a directed path from D_2 to D_3, D_2 must be the first decision in

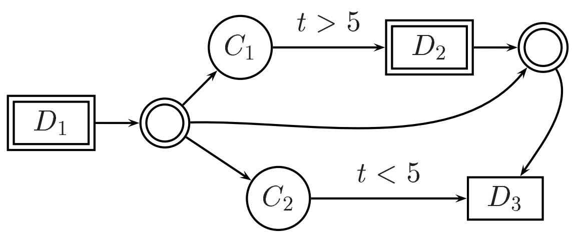

Figure 2.8: *An example of problems related to the ≺-relation — further elucidated using temporal arcs.*

the remaining part of the diagram. Thus, D_2 must be following D_1 in the temporal order. In other words, in any new decision problem, arising from deciding on D_1 and instantiating end(D_1), we can easily identify the first decision presented to the decision taker, viz. D_2.

If the diagram in Figure 2.7 was shown in the blown-up version, with all temporal arcs in place, as in Figure 2.8, we would immediately see that $\text{end}(D_1) \prec \text{end}(D_2)$. In conjunction with the relationships $\text{end}(D_2) \prec D_3$, $D_1 \prec \text{end}(D_1)$, and $D_2 \prec \text{end}(D_2)$, this would allow for only two orderings:

$$D_1 \prec \text{end}(D_1) \prec D_2 \prec \text{end}(D_2) \prec D_3$$

and

$$D_1 \prec D_2 \prec \text{end}(D_1) \prec \text{end}(D_2) \prec D_3.$$

As the second of these orderings clearly violates the no-delay assumption, the only ordering, which fulfills the assumptions, is the first one. Notice that no additional information is portrayed by the diagram, though, as the only other possibility of placing temporal arcs, i.e. an arc from end(D_2) to end(D_1), is not allowed, as it would result in a cycle involving time variables.

This exercise seems quite elaborate and yet the conclusion so vague: the definition of ≺ clearly states that D_2 follows D_1 in the ordering obtained by ≺, if *and only if* there is a directed path from D_1 to D_2, which ceases to be the case, if we instantiate end(D_1) to some value less than 5. If we add temporal arcs to the diagram the ordering seems much clearer, although no new information is conveyed. Thus, it must be the ordering relation, which is not welldefined. Furthermore, and worse, according to ≺, we have that C_1 precedes both D_2 and D_3 in the temporal ordering, but when end(D_1) is instantiated to a value less than 5 that conclusion seems dubious, as the guard clearly states that C_1 is observable, only when the time has passed 5, which is not the case when the decision following D_1 is initiated. This problem of observable variables cannot be remedied simply through instantiating variables or disregarding parts of the IDIT. Consequently, we must define a new and weaker ordering relation, $\prec'$, which imposes only a partial ordering on decision variables, to accommodate for cycles.

In addition to these problems, extended no-forgetting suffers from a semantical oversight, which has reflected on $\prec$. The omission is connected to the situation in which an ordinary chance variable, C, is a parent of a time variable, T, but not any decisions prior to T. Semantically, this means that the state of C affects the point in time represented by T. An example of this is the weather variable, W, which affects the time it takes to harvest, modelled as an impact on end(Ha). In such cases, it is reasonable to assume that the impacting variable is observed, as it directly affects the time it takes to take a decision. Rarely are we in a situation, in which the completion of some task have a time span which is noticeably more or less than usual, with no explanation as to why this is so.

When we add the assumption that ordinary chance variables affecting a time variable are observed, a logical step is to work this assumption into the extended no-forgetting assumption. That is, in addition to remembering variables observed at decisions, we also remember variables having an impact on time variables. The new ordering relation should conform to this, by explicitly letting chance variables affecting a time variable, be prior to this in the temporal ordering.

A New Ordering Relation

Following the discussion above we define a decision, D, to be prior to another variable, X, in the temporal ordering of variables in an IDIT, $\mathcal{I}$, written $D \prec'_{\mathcal{I}} X$, if there is a directed path, from D to X in $\mathcal{I}$, comprising no guarded arcs. The reasoning behind this is similar to the one applied in influence diagrams. An example of this is the decision D_1, which is prior to end(D_1), C_1, and C_2 in the IDIT in Figure 2.7. It is, however, not prior to D_2 nor D_3 by virtue of this rule alone, as the paths from D_1 to both of them comprise guarded arcs. Second, we define a time variable, T_i, to be prior to a time variable, T_j, if there is a directed path, P, from T_i to T_j. This is justified if there is a path consisting only of temporal arcs from T_i to T_j. Requirement 5 guarantees the existence of such a path from either T_i to T_j or from T_j to T_i. Requirement 4 and the existence of P tell us that it must be the former, and concluding that $T_i \prec'_{\mathcal{I}} T_j$ is, thus, justified. For instance, we concluded a little earlier that the time variable end(D_1) in the IDIT in Figure 2.7 had to be prior to the time variable end(D_2) using a similar argument.

A decision, D, which is a descendant of a time variable, T, is defined to be following T in the temporal ordering. This is justified, as the point in time represented by T must either be init(D), if no other time variables exists on paths from T to D, or some point in time prior to init(D), otherwise. As an example, this, in addition to transitive closure introduced later, is the rule which allows us to conclude that D_2 is following D_1 in the IDIT in Figure 2.7. We furthermore define an ordinary chance variable, C, to be following a time variable, T, if C is a parent of a decision, D, such that init(D) $= T$, and C is not prior to T. As the arc from C to D can be seen as being guarded, either by a genuine guard, if such a guard is shown in the diagram, or the trivial guard, $t = t$, its observation depends on the value of T, and hence, it cannot be prior to T in the temporal ordering. The additional requirement on C not being prior to T in the temporal ordering is practically redundant, as that would imply C is being observed at some decision, D', initiating before the point in time represented by T. In that case, C would also be a parent of D', and the arc from C to D would, consequently, be redundant due to no-forgetting. An example of a relationship such as this, is the variable Re in Example 1, which follows end(Spr'), in the temporal ordering.

Additionally, we define an ordinary chance variable, C, to be prior to a decision variable, D, if C is a parent of D connected with an unguarded arc. In this case C is always observed prior to deciding on D, and we may then safely assume that $C \prec' D$. Had the arc between the two been guarded, we cannot conclude the same, and the ordering of the two variables is thus unknown. This is reflected in Figure 2.7, where the positions of the two variables C_1 and C_2 in the temporal order is undefined prior to instantiating end(D_1). The counterpart of this rule is that a chance variable, C, is prior to a time variable, T, if C is a parent of T. An example of this is the variable W in Figure 2.1, which is prior to the time variable end(Ha).

Analogous to orderings of variables in influence diagrams, we define an ordinary chance variable, C, that is not a parent of a decision variable, D, or any decision which might be prior to D in the temporal ordering, to be following D. This rule only differs from the one used in influence diagrams, by the specific check for C being a parent of some decision which might prior to D. In influence diagrams it is sufficient to check whether C is prior to D, but in the IDIT in Figure 2.7 this would lead us to conclude that C_1 is prior to D_3 in the temporal ordering, which is not necessarily the case. An example of this rule is the variable C_1 in Figure 2.5, which

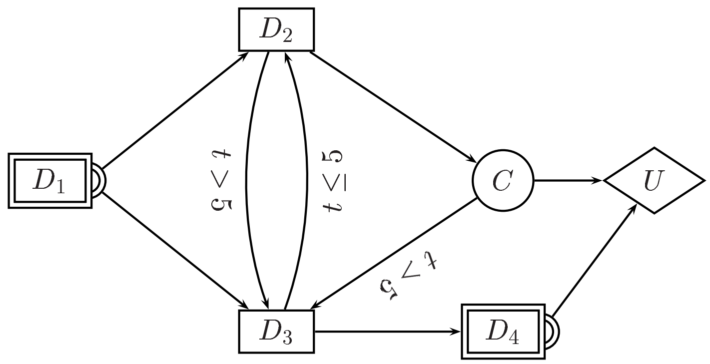

Figure 2.9: *Example showing the need for one of the $\prec'$-rules.*

is not a parent of any decision, and therefore, follows them all in the temporal ordering. Notice that C_2, in the same IDIT, does not fall into this category, as it is a parent of D_2, and thus might be observed prior to D_2. It is, however, following D_1. Similarly, we define a time variable, T, which is not prior to a decision, D, to be following D in the temporal ordering. The reason for this, is that, as T does not represent a point in time prior to initiation of D, it must be a point in time after this. An example of the need for this rule, is the IDIT in Figure 2.9. Here the ordering of D_2 and end(D_4), i.e. $D_2 \prec'_{\mathcal{I}}$ end(D_4), is determined by this rule.
Finally, we extend $\prec'_{\mathcal{I}}$ to its transitive closure, i.e. $X \prec'_{\mathcal{I}} Y$ and $Y \prec'_{\mathcal{I}} Z$ implies $X \prec'_{\mathcal{I}} Z$, which seems a natural convention, as we are dealing with events in the ever progressing flow of time.

Definition 2.3
The partial temporal ordering of elements in an IDIT, $\mathcal{I}$, is the transitive closure of the ordering relation, $\prec'_{\mathcal{I}}$, having the following characteristics:

- *if there is a directed path, comprising no guarded arcs, from a decision variable, D, to some other variable, X, in $\mathcal{I}$, then $D \prec'_{\mathcal{I}} X$,*
- *if there is a directed path from a time variable, T, to a time or decision variable, X, in $\mathcal{I}$, then $T \prec'_{\mathcal{I}} X$,*
- *if an ordinary chance variable, C, is an unguarded parent of a time or decision variable, X, in $\mathcal{I}$, then $C \prec'_{\mathcal{I}} X$,*

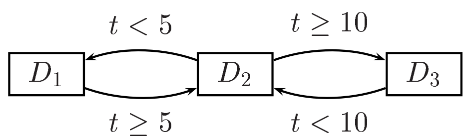

Figure 2.10: *An example of an IDIT that is not welldefined, as no unique first decision can be identified.*

- *if an ordinary chance variable, C, is not a parent of a decision, D, or any other decision D', where $D \not\prec'_{\mathcal{I}} D'$, in $\mathcal{I}$, then $D \prec'_{\mathcal{I}} C$,*
- *if a time variable, T, is not prior to a decision, D, in $\mathcal{I}$, then $D \prec'_{\mathcal{I}} T$, and*
- *if an ordinary chance variable, C, is a parent of a decision, D, in $\mathcal{I}$, and $T \prec'_{\mathcal{I}} D$ for some time variable, T, then $T \prec'_{\mathcal{I}} C$.*

The extended notation applying subscripts, used in the above definition, is abandoned when the IDIT is obvious from the context.
Applied to the diagram of Figure 2.7 this new relation yields the following ordering of decision and time variables:

$$D_1 \prec' \mathrm{end}(D_1) \prec' D_2 \prec' \mathrm{end}(D_2) \prec' D_3.$$

The ordering relationships of C_1 and C_2 are undefined except for both of them following $\mathrm{end}(D_1)$ and C_2 following $\mathrm{end}(D_2)$, mirroring that C_1 and C_2 are not necessarily observed before any decision. Returning briefly to cycles we see that the ordering relationships of variables in the IDIT pictured in Figure 2.5 are the transitive closure of the relationships

$$D_1 \prec' \mathrm{end}(D_1),\ \mathrm{end}(D_1) \prec' D_2,\ \mathrm{end}(D_1) \prec' D_3,\ D_2 \prec' C_1,\ \text{and}\ D_3 \prec' C_2.$$

No total temporal ordering of all variables can be obtained from these relationships, but if some decision, which is prior to all other decisions according to $\prec'$, can be identified, we can, given instantiations of it and its end time and through evaluation of guards, identify the next decision and the set of variables, observed immediately before that next decision initiates. Some diagrams, such as the one presented in Figure 2.10, do not have this quality, and we, therefore, say that such diagrams are not *welldefined*.

Welldefined IDITs

Before we define this notion of welldefinedness, precisely, we introduce some auxiliary concepts and results. These are referenced extensively throughout the rest of the report. We start with simple and intuitive concepts arising from applying $\prec'$ to sets of variables.

Definition 2.4
Let $\boldsymbol{Z}$ be a set of variables in an IDIT, $\mathcal{I}$. A variable, X, in $\boldsymbol{Z}$ is then said to be the first *variable of $\boldsymbol{Z}$, if $X \prec'_{\mathcal{I}} Y$, for all other variables Y in $\boldsymbol{Z}$.*

As examples of this definition, the decision D_1 is the first variable of the set of decision variables in the IDIT in Figure 2.5, and the decision D_3 is the first variable of the set $\{D_3, C_2\}$ in the same IDIT. Notice that the definition says nothing about the existence of a first variable. In fact, this cannot be guaranteed, as is evident from the set of ordinary chance variables in the IDIT in Figure 2.5. In the report, we treat the concept of first variables rather casually and refer to them in an intuitive manner, e.g. "the first time variable" and "the first decision following X" to mean "the first variable of the set of time variables" and "the first variable of the set consisting of decisions, which follow X in the temporal ordering obtained from $\prec'$", respectively.

Definition 2.5
Let $\boldsymbol{Z}$ be a set of variables in an IDIT, $\mathcal{I}$. A variable, X, in $\boldsymbol{Z}$ is then said to be the last *variable of $\boldsymbol{Z}$, if $Y \prec'_{\mathcal{I}} X$, for all other variables Y in $\boldsymbol{Z}$.*

An example of a last variable is end(D_1) in Figure 2.5, which is the last variable in the set of time variables. Similar to the concept of first variable, there is no guarantee of existence, and we refer to last variables in an intuitive manner in the remaining part of the report.

Definition 2.6
Let $\mathcal{I}$ be an IDIT and X and X' be two variables in $\mathcal{I}$. A variable, Y, is then said to be an intermediate *variable between X and X', if $X \prec'_{\mathcal{I}} Y$ and $Y \prec'_{\mathcal{I}} X'$.*

An example of this definition is end(D_1) in Figure 2.5, which is an intermediate variable between D_1 and C_2. As for first and last variables, the existence of intermediate variables between two variables cannot be guaranteed, and we use rather casual language in referring to these.

In addition to these definitions building on $\prec'$, we introduce the concept of instantiations:

Definition 2.7

Let $\mathcal{I}$ be an IDIT and X a variable in $\mathcal{I}$. Then an IDIT in which X is known to be in some state, $x \in \mathbf{sp}(X)$, is called an instantiation *of $\mathcal{I}$ on X to the value x. We write this as $\mathcal{I}[X \mapsto x]$.*

Examples of instantiations of the IDIT, $\mathcal{I}$, in Figure 2.5, assuming that the state space of D_1 is $\{d_1, \neg d_1\}$ and the state space of C_2 is $\{c_2, \neg c_2\}$, are $\mathcal{I}[D_1 \mapsto d_1]$ and $\mathcal{I}[C_2 \mapsto \neg c_2]$, whereas $\mathcal{I}[D_1 \mapsto x]$ and $\mathcal{I}[Y \mapsto y]$ are not. $\mathcal{I}[D_1 \mapsto x]$ is not an instantiation as x is not a state of D_1, and $\mathcal{I}[Y \mapsto y]$ is not an instantiation as Y is not a variable in $\mathcal{I}$.

The extra information on the state of a variable can cause graphical representations of the IDIT to change: When an IDIT is instantiated on a time variable, T, all guards on arcs going into intermediate decisions, between T and the first time variable following T, can be evaluated with t being the value T is instantiated to. Arcs with guards, which evaluate to **true**, can then be exchanged for arcs with no label. An arc with a guard, which evaluates to **false**, on the other hand, must be removed. However, owing to the discussion of inheritance of guarded arcs in Section 2.2, new arcs, with the same guard, must be added from the guarded variable to decisions following the first time variable following T.

Likewise, in an instantiation on a variable, X, which is in the domain of some restriction function for a decision, D, to the value x, the restriction function for D,

$$r_D : \mathbf{sp}(\mathbf{pa}_d(D)) \to 2^{\mathbf{sp}(D)},$$

can be exchanged for the function

$$r'_D : \mathbf{sp}(\mathbf{pa}_d(D) \setminus \{X\}) \to 2^{\mathbf{sp}(D)},$$

where $r'_D(\vec{c}) = r_D(\vec{c}, x)$, for all $\vec{c}$ in $\mathbf{sp}(\mathbf{pa}_d(D) \setminus \{X\})$, after which the dashed arc from X to D is rendered solid.

Instantiations of the IDIT, $\mathcal{I}$, in Figure 2.7, $\mathcal{I}[\text{end}(X) \mapsto 2]$ and $\mathcal{I}[\text{end}(X) \mapsto 12]$, are shown in Figures 2.11 and 2.12, respectively. Note that, as an instantiation is an IDIT with added information, it is reasonable to talk of instantiations of instantiations. For notational convenience we write $\mathcal{I}[\{X_1, X_2, \ldots, X_n\} \mapsto (x_1, x_2 \ldots, x_n)]$, or $\mathcal{I}[\boldsymbol{S} \mapsto \vec{x}]$,

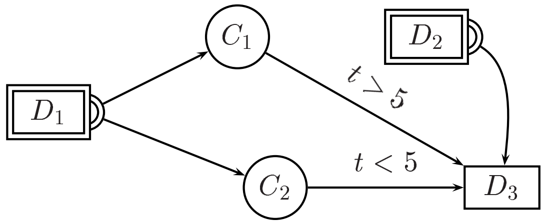

Figure 2.11: *An instantiation of the IDIT in Figure 2.7 corresponding to end*(X) *being* 2.

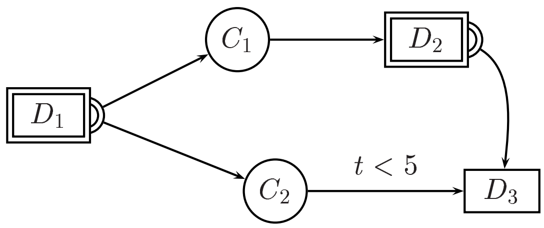

Figure 2.12: *An instantiation of the IDIT in Figure 2.7 corresponding to end*(X) *being* 12.

where $\boldsymbol{S}$ is $\{X_1, X_2, \ldots, X_n\}$, to denote the instantiation

$$\mathcal{I}[X_1 \mapsto x_1][X_2 \mapsto x_2] \cdots [X_n \mapsto x_n].$$

Also for notational convenience, we use the term IDIT to mean an IDIT with zero or more instantiated variables, unless otherwise explicitly stated.
Not all instantiations are sensible, though. For instance, an instantiation which include a decision, but not the time variable stating when the decision initiates, would constitute a paradoxical situation. We define the sensible instantiations:

Definition 2.8
Let $\mathcal{I}[\boldsymbol{Z} \mapsto \vec{c}]$ *be an instantiation of an IDIT,* $\mathcal{I}$*, on a set of variables,* $\boldsymbol{Z}$*, to the values* $\vec{z}$*. Then* $\mathcal{I}[\boldsymbol{Z} \mapsto \vec{z}]$ *is said to be a* temporally allowable *instantiation if,*

- *for all pairs of time variables,* $T_i = t_i$ *and* $T_j = t_j$*, in* $\boldsymbol{Z}$*, where* $T_i \prec' T_j$*, we have that* $t_i \leq t_j$*, and*

- *there exists no non-instantiated time variable,* T*, in* $\mathcal{I}[\boldsymbol{Z} \mapsto \vec{z}]$ *and* X *in* $\boldsymbol{Z}$*, such that* $X \not\prec'_{\mathcal{I}[\boldsymbol{Z} \mapsto \vec{z}]} T$*.*

In words, we require that the values of time variables do not violate the requirement on time progression, and we do not allow a variable to be observed or decided upon, unless all time variables, which precedes it in the temporal ordering, have been instantiated. A temporal allowable instantiation on all variables in an IDIT, which do not violate the restriction function of any decision, we call a *decision scenario*. For notational convenience we regard an IDIT with no instantiated variables as a temporally allowable instantiation.

At this point, we introduce a short hand notation, which renders methods introduced in the remainder of this report more elegantly expressed. For an IDIT, $\mathcal{I}$, which contains both instantiated and non-instantiated time variables, we denote the set of intermediate decision variables between the last instantiated time variable and the first non-instantiated time variable as $\boldsymbol{ID}_{\mathcal{I}}$. The intuition behind this is that each decision in $\boldsymbol{ID}_{\mathcal{I}}$ initiates at the point in time the decision problem modelled by $\mathcal{I}$ starts, and are, thus, part of that part of the decision problem, which is current. For instance, in the IDIT, $\mathcal{I}$, in Figure 2.12, which models a decision problem in which D_1 has ended at time 12, $\boldsymbol{ID}_{\mathcal{I}}$ consists of D_2, meaning that D_2 is the only decision initiating at time 12. In an IDIT, $\mathcal{I}$, which only contains instantiated time variables, $\boldsymbol{ID}_{\mathcal{I}}$ is defined to be the set of decisions following the last time variable. If $\mathcal{I}$ contains only non-instantiated time variables, $\boldsymbol{ID}_{\mathcal{I}}$ is the set of decisions prior to the first time variable. For instance, $\boldsymbol{ID}_{\mathcal{I}}$ consists of D_1 in Figure 2.5. Finally, in IDITs containing no time variables, $\boldsymbol{ID}_{\mathcal{I}}$ equals $\boldsymbol{V}_D$, corresponding to all decisions being taken in the same instant.

These definitions and notational conventions aside, we note some useful aspects of $\prec'$. First and foremost, it is clearly the case that in any IDIT, which conforms to Requirement 5, a total ordering of all time variables can be identified. Furthermore, as this ordering is induced from temporal arcs, which, by definition, cannot be guarded, no amount of instantiation of variables can alter it.

Another useful result is that in any IDIT, $\mathcal{I}$, for any decision variable, D, and time variable, T, we can determine, whether $D \prec'_{\mathcal{I}} T$, or $T \prec'_{\mathcal{I}} D$. This result is immediately obtained from Requirements 4 and 6, which allow only the ordering of instant decisions to vary, and as time variables are disallowed in cycles, even the ordering of the instant decisions relative to time variables are fixed. Again, no amount of instantiation can change these ordering relations.

Building on these notions, we define a welldefined IDIT:

Definition 2.9

Let $\mathcal{I}$ be an IDIT. The we say that $\mathcal{I}$ is structurally welldefined, *or simply welldefined,* *if, for any temporally allowable instantiation, $\mathcal{I}'$, for each decision, D, in $\boldsymbol{ID}_{\mathcal{I}'}$ and variable, X, in $\boldsymbol{V}_{\mathcal{I}} \setminus D$, either $D \prec' X$ or $X \prec' D$.*

Intuitively, for all temporally allowable instantiations, the ordering of all decisions, which are prior to the first non-instantiated time variable, is a total ordering, and the set of variables observed at each of those decisions can be uniquely determined. This definition tells us that no matter what points in time time variables represent, as long as they constitute an temporally allowable instantiation, the next decision to decide upon can always be identified.

Checking Welldefinedness

Definition 2.9 cannot be applied mechanically to verify that a specific IDIT is welldefined, though. That would call for a check of all temporally allowable instantiations, of which there, even for IDITs containing only a single time variable, is an infinite number. Instead we construct an operational method for examining whether an IDIT is welldefined. Before presenting the method, formally, we reveal the workings of it, by applying Definition 2.9 to the example IDIT, $\mathcal{I}$, in Figure 2.5, using intuition rather than strict adherence to the wording of the definition.

The approach, we take, is to exploit that even if there is an infinite number of allowable instantiations of a given IDIT, there is only a finite number of different structures derivable from it. That is, even if we can instantiate variables in an infinite number of ways, these instantiations can be grouped into sets with similar structures.

Looking at $\mathcal{I}$ in Figure 2.5, we see that there is a maximum of eight different structures of variables that conforms to the restrictions laid down by $\mathcal{I}$. These are portrayed in Figure 2.13. By applying the rules of $\prec'$, it can easily be seen that some of these structures do not fulfill the requirement on a clear ordering of decisions and unique set of observed variables. Therefore, we need to be sure that no temporally allowable instantiations result in one of those structures.

To get any further, we observe that the structure, which corresponds to a temporally allowable instantiation, is a function of the instantiated time variables only, as the structure is uniquely determined by the evaluation of guards, which in turn are functions over time variables, only. Therefore, we need only focus on the values of time variables in temporally allowable instantiations. As a result of this observations, we can divide the temporally allowable instantiations into groups, corresponding to how many time variables they encompass. In the case of the IDIT $\mathcal{I}$, we thus group the temporal allowable instantiations into two sets: One where $\text{end}(D_1)$ is instantiated and one where it is not. Next we need to subdivide these sets into groups based on their structure.

The group of instantiations, where no time variables are instantiated, can only result in one structure of the decision variables and observed variables prior to the first non-instantiated time variable, viz. $\text{end}(D_1)$, as guards are functions over time variables only. The set of decisions prior to $\text{end}(D_1)$ consists of a single variable, D_1, and the ordering of its elements is, trivially, total. Likewise, as no variables are parents of D_1, the set of observed variables can be unambiguously determined. Thus, all temporally allowable instantiations not involving $\text{end}(D_1)$ fulfills the requirements of Definition 2.9.

When we move on to checking the temporally allowable instantiations including $\text{end}(D_1)$, we can exploit the work we just completed on the instantiations that did not include $\text{end}(D_1)$: As the ordering of decision variables with respect to time variables are total in $\mathcal{I}$, none of the decision variables prior to $\text{end}(D_1)$ can be involved in the structural changes arising from instantiation of $\text{end}(D_1)$. Thus, the decisions prior to $\text{end}(D_1)$ do not need to be checked when we examine whether a temporally allowable instantiation involving $\text{end}(D_1)$ fulfills the requirements in Definition 2.9.

As we attempt to subdivide the group of temporally allowable instantiations involving $\text{end}(D_1)$, according to their structure, we encounter a potential problem. We mentioned that only time variables affect this division, so the problem eventually boils down to splitting the state space of $\text{end}(D_1)$ according to its effect on the guards $t < 5$, $t < 10$, and $10 \leq t$. In this specific example this can be accomplished quite easy through identifying the critical points 5 and 10, and then splitting the state space of $\text{end}(D_1)$ accordingly. However, for some guards, such as "t is a prime", this straightforward splitting is undecidable. Therefore and in the rest of the report, we assume that all guards are of the form

$$g(t) = \bigvee_i t \in I_i,$$

where the I_i's are intervals of the real line.

As we have identified three intervals $]-\infty;5[$, $[5;10[$, and $[10;\infty[$ in which the structural changes resulting from instantiating $\text{end}(D_1)$ are the same, we can split the temporally allowable instantiations including $\text{end}(D_1)$ into three groups. Instantiations in all groups agree on the structure of decisions prior to $\text{end}(D_1)$, and the

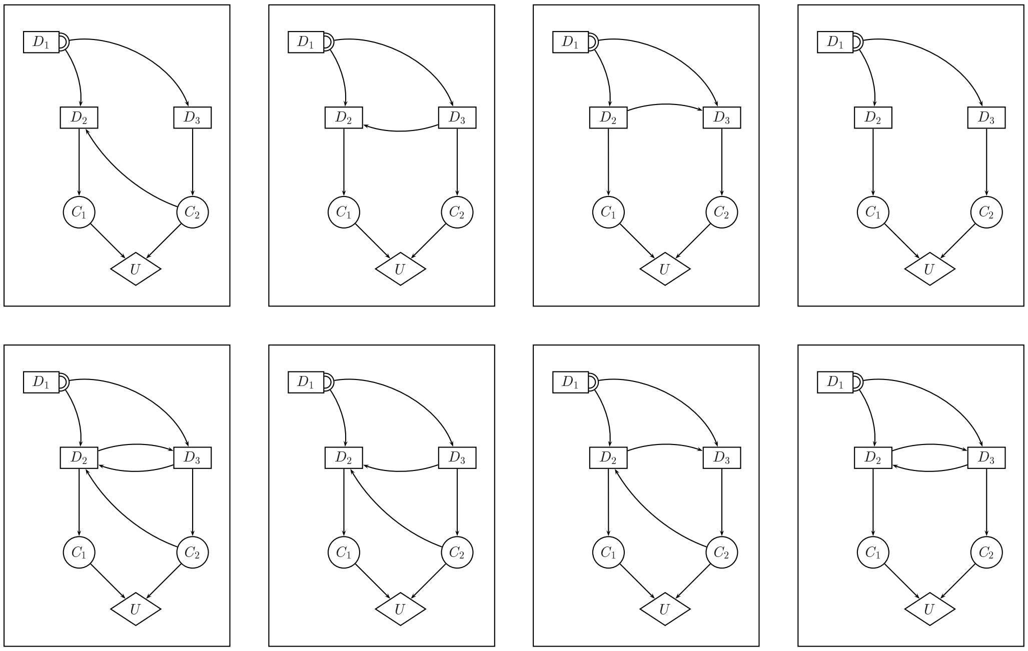

Figure 2.13: *The possible structures of variables in the IDIT in Figure 2.5.*

instantiations in each individual group agree on the structure of the remaining part of the IDIT. We can visualize this process as a tree, which is illustrated in Figure 2.14. As can be seen from the resulting structures in the three leaves, each of the three groups of temporal allowable instantiations fulfills the requirements for welldefinedness, and we can conclude that $\mathcal{I}$ is welldefined. In the remainder of the report, we will refer to trees, constructed by a process such as this, as *split trees*.
The method we have just described can be generalized to one that can be applied for checking IDITs including an arbitrary number of time variables. Such a method, for checking whether an IDIT, $\mathcal{I}$, is welldefined, is presented below and we elaborate on the details, which set it apart from the one just given, afterwards. The method takes as parameter a starting point in time, t, which for most problems would be 0, but could be set to minus infinity or any number for that matter. The starting point represents, when the decision problem modelled by the IDIT is initiated, that is, the minimum value the first time variable can possibly take on.

Method 2.10 (Input: IDIT $\mathcal{I}$, and point in time t)

1. Identify $\boldsymbol{ID}_{\mathcal{I}}$
2. Evaluate whether the instantiation that is $\mathcal{I}$ fulfills the requirements for welldefinedness, through checking if a total ordering of all decisions in $\boldsymbol{ID}_{\mathcal{I}}$ can be obtained from $\prec'_{\mathcal{I}}$, and if all arcs into decisions in $\boldsymbol{ID}_{\mathcal{I}}$ are without guards. If this is not the case,

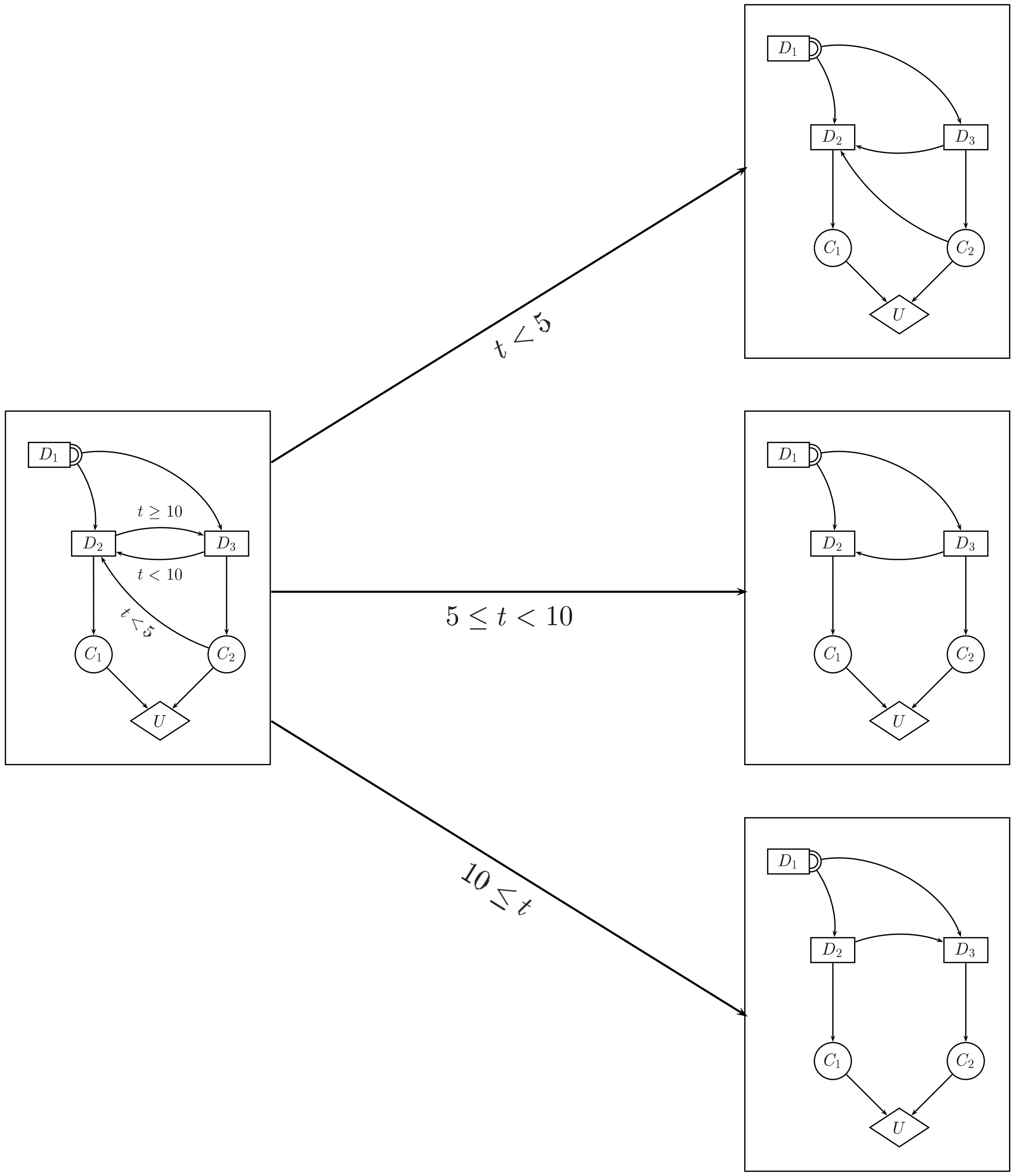

Figure 2.14: *The tree constructed by the method for checking welldefinedness of the IDIT in Figure 2.5.*

stop and report failure. If the test is a success, and no non-instantiated time variables remains in $\mathcal{I}$, stop and report success. Otherwise, let T be the first non-instantiated time variable in $\mathcal{I}$, and continue.

3. Let $\boldsymbol{G}$ be the set of guards on arcs in $\mathcal{I}$ going into the decisions in $\boldsymbol{ID}_{\mathcal{I}[T\mapsto t]}$.

4. Partition the points in time from t to infinity into a minimal set of subsets, $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n\}$, such that each guard in $\boldsymbol{G}$ evaluate to the same value for all points in time in each $\boldsymbol{T}_i$. That is, for all $\boldsymbol{T}_i$ and all f in $\boldsymbol{G}$, $f(t_a)$ equals $f(t_b)$, for any two points, t_a and t_b, in $\boldsymbol{T}_i$.

5. For each subset, $\boldsymbol{T}_i$, construct the IDIT $\mathcal{I}[T \mapsto t_i]$, where t_i is the least element of $\boldsymbol{T}_i$, and recursively check whether $\mathcal{I}[T \mapsto t_i]$ is welldefined for the point in time t_i. If one or more of these instantiations is not welldefined, then $\mathcal{I}$ is not welldefined, otherwise it is welldefined and success is reported.

There are two main differences between Method 2.10 and the one illustrated by the example: Most obvious, Method 2.10 is recursive. Second, it does not generate groups of instantiations, but a sort of generalized representative of each of these groups.

That Method 2.10 is recursive is due to it handling more than one time variable. When we split the group of instantiations in the example, we did so according to how the first time variable, $\text{end}(D_1)$, affected the guards on arcs into decisions prior to the first time variable following $\text{end}(D_1)$, which did not exist. At the same time we reasoned why the structure of decisions prior to $\text{end}(D_1)$ was left untouched by instantiating D_1, and therefore why we could disregard this part of the IDIT when checking instantiations including $\text{end}(D_1)$. When we are dealing with a second time variable, T_2, we can employ this reasoning again and consider only the effect of T_2 on the part of the IDIT that follows it.

This apparently suggests an iterative method, in which parts of the IDIT between two time variables are checked one after the other. However, the value a time variable, T, is instantiated to can affect the structure of variables following the time variable following T. This is because guards that do not evaluate to true are inherited by subsequent decisions and their sets of observed variables, thus, depend on more points in time than just their initiation time. Consequently, we need to employ a recursive strategy.

We can contend ourself with not constructing groups of instantiations, but rather representatives of such groups, due to three observations. First, when we split a group of temporally allowable instantiations on some time variable T, we know that the instantiations in the group all agree on the structure of the decisions prior to T, and it, thus, does not matter if we chose a single representative for this. Second, consider the group of temporally allowable instantiations corresponding to one of the subgroups of instantiations — say, those where T is instantiated to a value in $[t_1; t_2] \cup \cdots \cup [t_{n-1}; t_n]$, where $i < j$ implies $t_i \leq t_j$: No matter what value in this interval we choose to instantiate T to, the ordering of decisions prior to the first time variable following T will be the same. Likewise, for the sets of observed variables. Finally, when T is instantiated to some value t', the possible values of instantiated time variables, following T in the temporal order in a temporally allowable instantiation, are limited to those in $[t'; \infty[$. Therefore, by choosing to instantiate T to the lowest possible value, t_1, the set of possible values of following time variables encompass the possible values had we chosen any other t' in $[t_1; t_2] \cup \cdots \cup [t_{n-1}; t_n]$. Thus, by choosing the lowest possible value for a split variable, it suffices to use a representative from a group of instantiations.

Although we use the "lowest possible value", or equivalently, the minimal element of a set, as instantiation value in this method, some intervals, such as $]4; 5[$ have no minimal element. In such a case, we choose to use the abstract "value" $]4$, meaning the number which is less than any number in $[4; \infty[$ except for 4 which it is greater than, as instantiation value. That this "value" do not have the properties of real numbers, such as the ability to be a part of a sum or multiplication, does not hinder us from using it in this case, as all we are using it for, is comparisons.

This section concludes our description of IDITs and the semantics used in this report. We have described the representation language both in its original form and with some alterations that enhances the language. In the remainder of the report when we refer to IDITs, we are referring to the representation language defined in Section 2.2, and when we use temporal orderings of nodes, we are referring to the semantics introduced in this section.

Chapter 3

Solutions to IDITs

So far no method for solving decision problems modelled with the IDIT representation language has existed. In this chapter we describe a method, which solves a subset of these, and apply it in an example. The chapter is divided into three sections. Section 3.1 is a general discussion of what a solution to a decision problem is. Section 3.2 is a description of what it means to solve an IDIT in particular, and Section 3.3 presents the method we have devised for solving IDITs, as well as the example of this.

3.1 Solutions to Decision Problems

The purpose of the representation language IDITs is primarily to be a standard, in which decision problems involving time can be modelled compactly and unambiguously, and for which the models can easily be interpreted by human beings. In short: Supplying a means for discussing and communicating decision problems in a sound manner. Furthermore, the representation language has a syntax and semantics, which allow models to be fed to a computer. Once a decision problem has been modelled by an IDIT, it is, therefore, possible to have methods that, given the model, can reason about the problem. One method, which is strongly desirable, is an automated solution method. Solution, in this case, meaning a prescription for which choices a decision taker should choose at the various decisions, given previous observations and choices, in order to maximize his expected utility. We formalize this notion using terms and concepts traditionally used in describing solutions to decision problems.

The formalization given in this section is written in general and abstract terms, in order to allow the reader to focus on what the essence of a solution is, instead of details pertaining to solutions of IDITs. In Section 3.2 we transform the concepts and terms into IDIT specific equivalents, which take advantage of the information on structural constraints that an IDIT contains.

As the term "decision problem" is unspecified at this point, we briefly list what we consider the bare essentials of a decision problem in this general discussion. A decision problem, $\mathcal{P}$, contains a set of chance variables, $\boldsymbol{V}_C^{\mathcal{P}}$, and a set of decision variables, $\boldsymbol{V}_D^{\mathcal{P}}$, collectively denoted $\boldsymbol{V}^{\mathcal{P}}$. We allow each variable to be continuous or discrete. In addition to the variables, $\mathcal{P}$ must specify a probability distribution for the chance variables given the decisions, i.e. $P(\boldsymbol{V}_C^{\mathcal{P}}|\boldsymbol{V}_D^{\mathcal{P}})$, and a utility function over the state space of all variables, $u^{\mathcal{P}} : \mathbf{sp}(\boldsymbol{V}^{\mathcal{P}}) \rightarrow \mathbb{R}$. The semantics of these concepts are similar to the ones specific to IDITs given in Section 2.1. Notice that a decision problem, in this context, contains no information on when a variable can be observed during the decision process, or whether it can be observed at all. Furthermore, it says nothing about the ordering of decisions, or assumptions on no-forgetting and no-delay. It is merely a description of what possible states of the world this problem is defined over, which parts are under direct control by the decision taker, how likely the remaining parts are, and how valuable each configuration of variables is to the decision taker. An example of a decision problem could, thus, be the farming problem, described in Example 1, stripped of any ordering information.

Before defining what a solution to a decision problem is, we introduce its constituent elements.

Definition 3.1

Let D be a decision variable in a decision problem, $\mathcal{P}$, and $\boldsymbol{P}$ a subset of $\boldsymbol{V}^{\mathcal{P}} \setminus \{D\}$. Then a function $\delta_D : \mathbf{sp}(\boldsymbol{P}) \rightarrow \mathbf{sp}(D)$ is called a policy *for D given $\boldsymbol{P}$.*

Intuitively, we may think of a policy, $\delta_D : \mathbf{sp}(\boldsymbol{P}) \rightarrow \mathbf{sp}(D)$, as a function, which given a configuration over a set of variables observed or decided upon in the past, $\boldsymbol{P}$, yields a choice from the decision D. An example of a policy for the decision variable Te in Example 1 given We_1 could be a function, which yields the choice *order test*, if We_1 is *much*, and *do not order test* if We_1 is *little*. Another example of a policy for Te, could be a function which yields *do not order test* if Ha is *quick* and We_2 is *little*, and *order test* for all other configurations of the two variables. This latter example would have no value for a decision taker, though, as both variables cannot be observed when deciding upon Te. The latter policy is rendered invalid by the ordering constraints given in the IDIT. Generally, we say that under an ordering, $\lhd$, over the variables in a decision problem, $\mathcal{P}$, a policy, $\delta_D : \mathbf{sp}(\boldsymbol{P}) \rightarrow \mathbf{sp}(D)$, is *valid* if for any variable X in $\boldsymbol{V}^{\mathcal{P}}$, we have that X is in $\boldsymbol{P}$ if and only if $P \lhd D$. In Example 1 the policy just described is, thus, not valid under the ordering $\prec'$.

In order to identify valid policies for a decision problem, $\mathcal{P}$, we assume that a configuration, $\vec{c}$, over the variables in $\boldsymbol{V}^{\mathcal{P}}$ uniquely determines the ordering of these variables. That is, we can define a function, $o^{\mathcal{P}} : \mathbf{sp}(\boldsymbol{V}^{\mathcal{P}}) \rightarrow \boldsymbol{O}^{\mathcal{P}}$, where $\boldsymbol{O}^{\mathcal{P}}$ is the set of all possible ordering of the variables in $\boldsymbol{V}^{\mathcal{P}}$, yielding the ordering of variables given a configuration over these. In IDITs, for instance, the ordering of variables can be found from the configuration of time variables.

Next, we define the formal equivalent of the previously mentioned prescription.

Definition 3.2

Let $\mathcal{P}$ be a decision problem and $\lhd$ some ordering over the variables in $\boldsymbol{V}^{\mathcal{P}}$. Then a set

$$\bigcup_{D \in \boldsymbol{V}_D^{\mathcal{P}}} \{\delta_D : \mathbf{sp}(\{X \in \boldsymbol{V}^{\mathcal{P}} | X \lhd D\}) \rightarrow \mathbf{sp}(D)\},$$

is a strategy *for $\mathcal{P}$ under the ordering $\lhd$. We denote this $\boldsymbol{S}_{\lhd}^{\mathcal{P}}$.*

A strategy for a decision problem under some ordering is, thus, a set of valid policies: One for each decision and the set of past variables for this decision. Given a decision problem, $\mathcal{P}$, we call a set,

$$\bigcup_{\lhd \in \boldsymbol{O}^{\mathcal{P}}} \boldsymbol{S}_{\lhd}^{\mathcal{P}},$$

a *strategy* for $\mathcal{P}$. The policies in a strategy $\boldsymbol{S}^{\mathcal{P}}$ which are valid under some ordering, $\lhd$, we also denote $\boldsymbol{S}_{\lhd}^{\mathcal{P}}$. In the report, we denote the set of all strategies for a decision problem, $\mathcal{P}$, as $\boldsymbol{\Delta}_{\mathcal{P}}$.

In order to describe the impact of policies and strategies on the expected utility of a decision problem, we introduce policy-induced probability distributions. This concept is of a similar nature to the probabilities of future decisions presented in [Nilsson and Jensen, 1999].

Definition 3.3

Let δ_D be a policy for a decision variable, D, given a set of past variables, $\boldsymbol{P}$, in a decision problem, $\mathcal{P}$. Then the probability distribution, $P_{\delta_D}(D|\boldsymbol{P})$, defined as

$$P_{\delta_D}(d|\vec{p}) = \begin{cases} 1 & \textit{if } \delta_D(\vec{p}) = d \\ 0 & \textit{otherwise,} \end{cases}$$

where d *is in* $\mathbf{sp}(D)$ *and* $\vec{p}$ *is in* $\mathbf{sp}(\boldsymbol{P})$, *is the* δ_D-induced probability distribution.

The δ_D-induced probability distribution, thus, represents the probability of the decision D, given the set of variables $\boldsymbol{P}$, if D is decided upon by a decision taker who follows δ_D.
We extend the concept of policy-induced probability distributions to strategy-induced probability distributions under some ordering.

Definition 3.4
Let $\boldsymbol{S}_{\lhd}^{\mathcal{P}}$ *be a strategy under some ordering,* $\lhd$, *for a decision problem,* $\mathcal{P}$, *with probability distribution* $P(\boldsymbol{V}_C^{\mathcal{P}}|\boldsymbol{V}_D^{\mathcal{P}})$. *The probability distribution,*

$$P_{\boldsymbol{S}_{\lhd}^{\mathcal{P}}}(\boldsymbol{V}^{\mathcal{P}}) = P(\boldsymbol{V}_C|\boldsymbol{V}_D) \prod_{\delta_D:\mathbf{sp}(\boldsymbol{P})\rightarrow\mathbf{sp}(D)\in\boldsymbol{S}_{\lhd}^{\mathcal{P}}} P_{\delta}(D|\boldsymbol{P}),$$

is then called the $\boldsymbol{S}_{\lhd}^{\mathcal{P}}$-induced probability distribution.

Thus, a strategy-induced probability distribution under some ordering is a joint distribution over chance and decision variables reflecting the probability of these, given that the decisions are decided upon by a decision taker, which follows that strategy and that the ordering of variables is the one the strategy is specified over.
In the beginning of this section, we briefly stated that a solution to a decision problem was a prescription for choices at all decisions given previous choices and observations. With the concepts introduced above we can define this precisely.
In the definition below, and henceforth, we use a $\downarrow$-notation on real-valued functions. For the function $f : \mathbf{sp}(\boldsymbol{Z} = \boldsymbol{C} \cup \boldsymbol{D}) \rightarrow \mathbb{R}$, where the variables in $\boldsymbol{C}$ are continuous and the variables in $\boldsymbol{D}$ are discrete, the expression $f(\boldsymbol{Z})^{\downarrow \boldsymbol{Z}^-}$, where $\boldsymbol{Z}^-$ is a subset of $\boldsymbol{Z}$, denotes the function $f^- : \mathbf{sp}(\boldsymbol{Z}^-) \rightarrow \mathbb{R}$ where

$$f^-(\vec{z}) = \sum_{\vec{d}\in\mathbf{sp}(\boldsymbol{D}\setminus\boldsymbol{Z}^-)} \int_{\mathbf{sp}(\boldsymbol{C}\setminus\boldsymbol{Z}^-)} f(\vec{d},\vec{c},\vec{z})d\vec{c},$$

for all $\vec{z}$ in $\mathbf{sp}(\boldsymbol{Z}^-)$. We say that f^- is the *projection* of f down-to $\boldsymbol{Z}^-$. If $\boldsymbol{Z}^-$ is the empty set, then $f(\boldsymbol{Z})^{\downarrow \boldsymbol{Z}^-}$ is a constant.

Definition 3.5
Let $\mathcal{P}$ *be a decision problem. Then an* optimal *strategy for* $\mathcal{P}$ *is*

$$\arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{P}}} \left(P_{\boldsymbol{S}^{\mathcal{P}}_{o^{\mathcal{P}}(\boldsymbol{V}^{\mathcal{P}})}}(\boldsymbol{V}^{\mathcal{P}}) \cdot u^{\mathcal{P}}(\boldsymbol{V}^{\mathcal{P}})\right)^{\downarrow\emptyset}.$$

The quantity, that is sought maximized, is denoted the *expected utility* of $\mathcal{P}$ under the ordering $o^{\mathcal{P}}(\boldsymbol{V}^{\mathcal{P}})$ given $\boldsymbol{S}$. As an offshoot of this definition, we define an *optimal policy* to be a policy, which is part of an optimal strategy. Given a decision problem we also designate an optimal strategy as a *solution* to the decision problem. The process, in which a solution to a decision problem is obtained, we call *solving* the decision problem, and a method for doing this we call a *solution method*.

3.2 Solutions to IDITs

The concepts introduced in the previous section were given in order to present a smooth transition from the rather casual, but intuitive, initial definition of what it means to solve a decision problem, to the mathematical cogent definition presented in Definition 3.5. However, as the definitions given are abstract and general, they also fail to take advantage of the additional information contained in an IDIT of a decision problem. An IDIT contains information on informational precedence, ordering constraints on decisions, probabilistic independencies among variables, as well as a decomposition of the total utility function. In this section we exploit some of this information and present a set of IDIT specific definitions, which render the eventual task of solving the decision problem easier.

Required Policies

The definition of a solution given in Section 3.1 reflects that a prescription for choices given previous choices and observations, at the face of it, would need to take into account all orderings of variables. However, if a decision problem is modelled as an IDIT, the set of possible orderings are drastically reduced, as non-guarded arcs in the diagram allow us to determine ordering restrictions between variables.

For instance, in Example 1, a policy for the decision Spr given the set of variables $\{\text{end}(Spr'), Wf\}$ would not make any sense. Both as the variable Wf cannot be observed before Spr is decided upon, and as knowing $\text{end}(Spr')$ would, because of no-forgetting, imply that the variables Spr', Te, and We_1 are also known. The sets

$\{\mathrm{end}(Spr'), Spr', Te, We_1\}$ and $\{\mathrm{end}(Spr'), Spr', Te, We_1, Re\}$ are the only possible sets of known variables when deciding upon Spr. Whether or not Re is observed, depends solely on the value of $\mathrm{end}(Spr')$. Consequently, we define a required policy for an IDIT:

Definition 3.6
Let δ_D be a policy for a decision, D, in an IDIT, $\mathcal{I}$. Then we call δ_D a required *policy for $\mathcal{I}$ if there is a temporally allowable instantiation, $\mathcal{I}[\boldsymbol{X} \mapsto \vec{x}]$, such that δ_D is valid under $\prec'_{\mathcal{I}[\boldsymbol{X}\mapsto\vec{x}]}$.*

In other words, only if there exists some genuine situation, in which a policy is needed, do we require it to be specified in a strategy for the IDIT.
Identifying required policies is not always easy, though, as guarded information arcs can be inherited by subsequent decisions and the truth values of some guards might imply specific truth values of others, as noted in Section 2.3. However, these structural changes are all functions of time variables, and in order to see whether a policy is required, it, therefore, suffices to consider instantiations of time variables only. Thus, the set of required policies constituting a strategy, $\boldsymbol{S}$, for an IDIT, $\mathcal{I}$, is

$$\bigcup_{\vec{t}\in\mathbf{sp}(\boldsymbol{V}_T^{\mathcal{I}})} \bigcup_{D\in\boldsymbol{V}_D} \{\delta_D : \mathbf{sp}(\boldsymbol{P}_{\mathcal{I},D,\vec{t}}) \to \mathbf{sp}(D)\},$$

where

$$\boldsymbol{P}_{\mathcal{I},D,\vec{t}} = \{X \in \boldsymbol{V}^{\mathcal{I}} | X \prec'_{\mathcal{I}[\boldsymbol{V}_T^{\mathcal{I}}\mapsto\vec{t}]} D\}.$$

In what follows, we use the short hand notation $\boldsymbol{S}_{o^{\mathcal{I}}(\vec{t})}$ to mean $\boldsymbol{S}^{\mathcal{I}}_{\triangleleft}$, where $\triangleleft$ is some ordering consistent with $\prec'_{\mathcal{I}[\boldsymbol{V}_T^{\mathcal{I}}\mapsto\vec{t}]}$.
Clearly, the sets of policies in $\boldsymbol{S}_{o^{\mathcal{I}}(\vec{t}_i)}$ and $\boldsymbol{S}_{o^{\mathcal{I}}(\vec{t}_j)}$, where $\vec{t}_i \neq \vec{t}_j$, for some strategy $\boldsymbol{S}$, would for many configurations, $\vec{t}_i$ and $\vec{t}_j$, be the same. For instance, a strategy, $\boldsymbol{S}$, for the IDIT in Figure 2.5 would consists of the policies

$$\delta_{D_1} : \mathbf{sp}(\varnothing) \to \mathbf{sp}(D_1),$$
$$\delta_{D_2} : \mathbf{sp}(\{D_1, \mathrm{end}(D_1), D_3, C_2\}) \to \mathbf{sp}(D_2), \text{ and}$$
$$\delta_{D_3} : \mathbf{sp}(\{D_1, \mathrm{end}(D_1)\}) \to \mathbf{sp}(D_3),$$

for any of the configurations of time variables where $\mathrm{end}(D_1)$ is less than 5. In order to utilize these similarities, we need to group the instantiations of time variables into sets of instantiations, which share a similar structure. Such a grouping is performed by Method 2.10, and in the next section we show how it can be used in the context of finding an optimal strategy for an IDIT.

As for decision problems in general, an optimal strategy for an IDIT is a strategy, which maximizes the expected utility. However, we can express this more concisely by using the factorization of probability distributions and utility functions stored in a realization. That is, an optimal strategy for an IDIT, $\mathcal{I}$, with realization $(\mathbf{\Phi}^{\mathcal{I}}, \mathbf{\Psi}^{\mathcal{I}}, \mathbf{\Pi}^{\mathcal{I}}, \mathbf{\Gamma}^{\mathcal{I}},)$, is

$$\arg\max_{\boldsymbol{S}\in\mathbf{\Delta}_{\mathcal{I}}} \left(\prod_{\pi\in\mathbf{\Pi}^{\mathcal{I}}} \pi \prod_{\phi\in\mathbf{\Phi}^{\mathcal{I}}} \phi \prod_{\delta\in\boldsymbol{S}_{o^{\mathcal{I}}(\boldsymbol{V}_T^{\mathcal{I}})}} P_\delta \left(\sum_{\psi\in\mathbf{\Psi}^{\mathcal{I}}} \psi \right) \right)^{\downarrow\varnothing} .$$

In the report, we regard two strategies for an IDIT, which yield the same expected utility, as equivalent.

Legal Policies

Policies, which are defined over sets of variables that, due to observability, can never constitute sets of past variables, are not the only policies that we can dismiss: Assume a decision, D, has a restriction function, r_D, which given some configuration, $\vec{p}$, over the variables $\boldsymbol{P}$, prevents a choice, d, to be taken when deciding upon D. A policy which advises the decision taker to take choice d, when observing that the variables $\boldsymbol{P}$ is instantiated as $\vec{p}$, is consequently flawed, as the advice cannot be followed. Therefore, we define a *legal* policy. In this definition, we use the $\downarrow$-operator on configurations over variables. For a configuration, $\vec{z}$, over the variables $\boldsymbol{Z}$, we denote by $\vec{z}^{\downarrow\boldsymbol{Z}'}$, where $\boldsymbol{Z}'$ is a subset of $\boldsymbol{Z}$, the configuration over the variables in $\boldsymbol{Z}'$ obtained from $\vec{z}$ by dropping coordinates of variables in $\boldsymbol{Z} \setminus \boldsymbol{Z}'$.

Definition 3.7

Let δ_D be a policy for a decision, D, with restriction function r_D, given a set of past variables, $\boldsymbol{P}$, in an IDIT, $\mathcal{I}$. If, for all configurations, $\vec{p}$, over $\boldsymbol{P}$, $\delta_D(\vec{p})$ is in $r_D(\vec{p}^{\downarrow\mathrm{dom}(r_D)})$, then we say that δ_D is a legal policy.

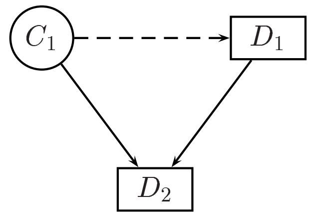

Figure 3.1: Not all policies for D_1 and D_2 make sense.

A strategy for an IDIT, which consists of only legal policies, are said to be legal as well. Thus, when searching for an optimal strategy, $\boldsymbol{S}$, for an IDIT, we must take care not to include any policies, which are not legal, in it. By considering this prior to searching for optimal strategies the search space is reduced, and the search is, potentially, more effective.

Not only can we focus our attention on legal policies, several of these policies can be disregarded as well. Consider the IDIT in Figure 3.1, where the state spaces of C_1 is $\{c_1, \neg c_1\}$, the state space of D_1 is $\{d_1, \neg d_1\}$, and the state space of D_2 is $\{d_2, \neg d_2\}$. The restriction function for D_1 is defined as

$$r_{D_1}(c_1) = \{d_1, \neg d_1\}$$

and

$$r_{D_1}(\neg c_1) = \{d_1\}.$$

In this case, two policies for D_1, δ_{D_1} and δ'_{D_1}, where

$$\begin{aligned} \delta_{D_1}(c_1, d_1) &= \delta'_{D_1}(c_1, d_1), \\ \delta_{D_1}(c_1, \neg d_1) &= \delta'_{D_1}(c_1, \neg d_1), \text{ and} \\ \delta_{D_1}(\neg c_1, d_1) &= \delta'_{D_1}(\neg c_1, d_1), \end{aligned}$$

but

$$\delta_{D_1}(\neg c_1, \neg d_1) \neq \delta'_{D_1}(\neg c_1, \neg d_1),$$

are equivalent advisers for a decision taker, as the only case, in which they differ, is one that cannot occur.

These restrictions, arising from distinguishing between policies that are legal and those that are not, are utilized in the next section, where we use an adaptation of Method 2.10 to solve an IDIT.

Representing Policies

Having defined exactly what a solution to an IDIT is, we need to address a fundamental problem before proposing a method for finding it, namely how to handle policies over continuous variables. That is, whether such policies can have finite representations. If it is not possible to do this, no solution method would ever finish outputting a solution and no decision taker would be able to use it.

Evidently, any strategy for an IDIT must contain a finite number of policies, as there is only a finite number of decisions and a finite number of combinations of variables, which can be past variables for decisions. Thus, we need only concern ourselves with representing individual policies in a finite manner. In solutions for influence diagrams, policies have traditionally been stored as tables, with an entry for each configuration of the past variables, stating the policy value of this configuration. In IDITs, however, we need to deal with continuous variables, in the form of time variables and wait decisions, and the table approach can, therefore, not be applied directly.

Two approaches for representing policies defined over continuous variables exists, though. Either the policy can be stored as a finite mathematical expression, or the continuous variables in the domain can be discretized according to their effect on the policy. Unfortunately, none of the approaches is ideal in all situations. The problem inherent in the former is that it might not always be possible to construct an expression, which can be evaluated within a reasonable time frame. The problem associated with the latter is that the continuous variables in the domain of some policies might require an infinite number of discretization intervals, for the policies to be represented in sufficient detail. However, in most cases we may settle for a satisfying solution. That is, storing a policy, which is not an optimal policy, but which can be represented using discretization or as a relatively simple function, and which yields an expected utility not substantially lower than the one offered by an optimal policy. When dealing with points in time, it is quite reasonable to use approximations: Initiating a decision at some exact point in time is rarely possible and it might be hard to justify that a utility should yield radically different values for points in time close to one another.

However, not all continuous domains can easily be discretized. Example 1 provides an example of the requirement on infinite discretization intervals. If we ignore concepts such as winter and life span of crops, and assume that, no matter when the farmer arrives at the Ha decision, he would gain maximum expected utility by harvesting, no matter the state of the crops and the weather, we need an infinite number of discretization intervals for the policy δ_{Ha}: For each possible pair of states, $t_{Spr'}$ and $t_{Ha'}$, of end(Spr') and end(Ha'), respectively, we need to store either a choice *quick* or *thorough* or the choice *no harvesting*, depending upon whether the time span between the points in time $t_{Spr'}$ and $t_{Ha'}$ is more than seven. No finite discretization intervals for end(Spr') and end(Ha') can capture this. In this case, we can circumvent the problem by specifying the policy over a discretization of the difference of the two variables in addition to discretizations of the variables themselves. A variable, such as the difference between the value of two time variables, which is defined as a deterministic function of other variables, we call a *derived* variable.

In this example the need for letting the policy vary according to a derived variable arose from the restriction function for the decision. In fact, the problem we solved, through using derived variables, would also be present when specifying the restriction function, as part of the realization, in the first place. In general, if a restriction function for a decision is not constant, we can construct a derived discrete variable, which take on values corresponding to this function, and thereby, we can conclude that all policies, which differs due to a restriction function, can be represented through this scheme. However, handling the derived variable in solution methods might not be as straightforward.

In this report we make some assumptions that renders the possibility of two continuous variables in the domain of a policy impossible. Therefore, we can restrict ourselves to policies defined over one continuous variable. These we represent as tables over the discrete variables in their domain, and with each cell containing a finite list of mutually exclusive and exhaustive intervals of the states in the state space of the continuous variable, and a corresponding choice from the decision. For wait decisions, we store each choice as a simple function of the value, t, of the continuous variable, such as $k - t$, where k is some constant.

Of course, even as we restrict our attention to policies varying over one continuous variable only, we still cannot be sure that we can construct a finite list of intervals, as there might be an infinite number of intervals over which the policy differs, for even this single variable. A solution is to divide the state space of the continuous variable into subsets, which do not necessarily constitute intervals, but this begs the question as to whether these subsets can be described in a finite manner. We leave these problems, as fortunately, the workings of the solution method we present guarantee a finite number of intervals.

3.3 Solving IDITs

In this section we present a method for solving IDITs, which is an extension of Method 2.10. We introduce the method through an example, before presenting the method in full.

Introducing the Problem

The method, we present in this section, builds on the structure of the method of solving decision trees, the method for solving asymmetric influence diagrams presented in [Nielsen and Jensen, 2000], and the method for solving valuation networks given in [Demirer and Shenoy, 2001]. The method presented here differs radically in some areas, though, most having to do with the continuous nature of time variables. As the method is a hybrid of elimination of variables and message passing in a split tree, it is not obvious why it identifies in an optimal strategy. To better understand the problems associated with elimination, which is specific to IDITs, we present a rather elaborate example, which should help the reader obtain some intuition on the structure of the method and why it works, allowing him to focus on the details of the method presented later in this section.
The example involves a number of general observations. To better communicate these, we employ a change in typography when they arise and return to the standard example typography again afterwards.

Example 2
The IDIT, we want to solve, is the IDIT, $\mathcal{I}$, presented in Figure 3.2. It is a slightly altered version of the IDIT we used as example in presenting Method 2.10. The changes, which are the addition of the node C_0 and the arcs connecting it to D_1, D_2, and D_3, have been introduced in order to render this example more interesting. We assume that all non-time variables are binary and denote the states of a variable, X, as x and $\neg x$.
The realization of $\mathcal{I}$, we work with, consists of the probability distributions given in Tables 3.1(a) through 3.2(a), the restriction function given in Table 3.2(b), the utility function given in Table 3.3, and the density function for $\text{end}(D_1)$, which is χ^2, with 5 degrees of freedom if D_1 is d_1, and 10 degrees of freedom if D_1 is $\neg d_1$. A plot of the density functions for $\text{end}(D_1)$ is shown in Figure 3.3.
The realization is chosen somewhat arbitrarily, and no specific semantics are given for the variables. A pair of relationships warrants emphasizing, though: First, the utility function,

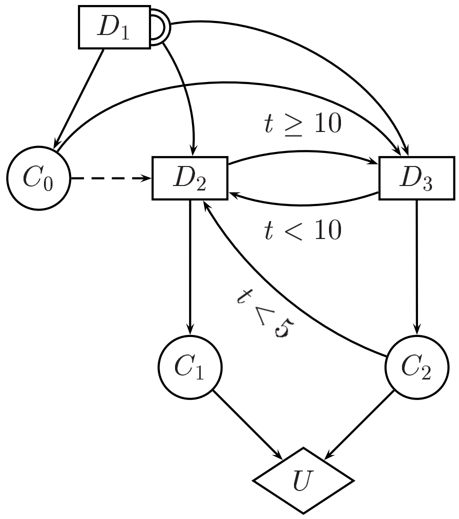

Figure 3.2: *The IDIT we want to find an optimal strategy for.*

	d_1	$\neg d_1$
c_0	0.2	0.7
$\neg c_0$	0.8	0.3

(a)

	d_2	$\neg d_2$
c_1	0.05	0.4
$\neg c_1$	0.95	0.6

(b)

Table 3.1: *(a): The probability distribution* $P(C_0|D_1)$. *(b): The probability distribution* $P(C_1|D_2)$.

	d_3	$\neg d_3$
c_2	1	0.1
$\neg c_2$	0	0.9

(a)

c_0	$\{d_2, \neg d_2\}$
$\neg c_0$	$\{d_2\}$

(b)

Table 3.2: *(a): The probability distribution* $P(C_2|D_3)$. *(b): The restriction function* $r_{D_2} : \mathbf{sp}(C_0) \to 2^{\mathbf{sp}(D_2)} \setminus \{\varnothing\}$.

	c_1	$\neg c_1$
c_2	40	20
$\neg c_2$	0	30

Table 3.3: *The utility function* U.

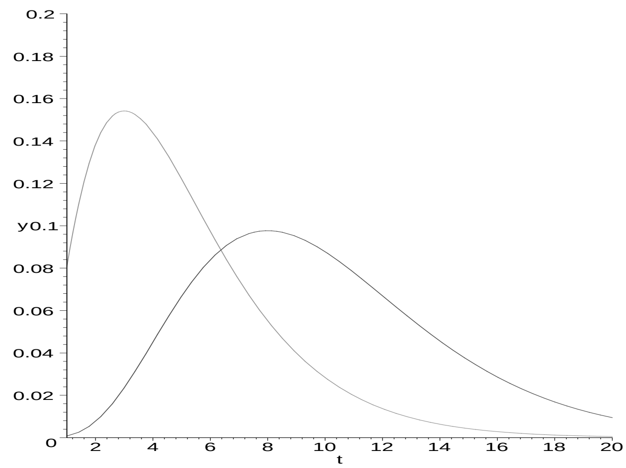

Figure 3.3: *The density functions for end*(D_1)*. The fair line is* 5 *degrees of freedom, and the dark one is* 10 *degrees of freedom*

U, is structured so that knowing the state of C_2, when deciding upon D_2, is desirable. Second, the choice of d_1 in D_1 yields a faster decision, which in turn, renders the observation of C_2 prior to deciding on D_2 more likely. However, this would, most likely, also render the choice $\neg d_2$ impossible and, consequently, the top utility of 40 unlikely. Thus, no candidate, for an optimal strategy, seems an obvious choice, and this exercise is, therefore, not trivial.

Solving $\mathcal{I}$

When identifying an optimal strategy, we start by limiting ourselves to the set of strategies, which suit the information constraints in the IDIT. As we mentioned in Section 3.2, the trees constructed by the method in Section 2.3 allow us to identify which policies are required for a strategy for a decision problem modelled as an IDIT. This is done through creating groups of instantiations of the IDIT, which share the same structure. In order to capture the constraints imposed by restriction functions we need to expand on the method, such that it constructs trees, in which the groups of instantiations, not only share a common structure, but also share the same state spaces of variables.
It turns out that integrating splitting of decision scenarios, according to state spaces of variables, into the process described in Method 2.10 is quite straightforward: Recall that Method 2.10 works its way through time variables in the order dictated by $\prec'$. Whenever a time variable, T, is encountered, the process splits the current group of decision scenarios, according to the value of T, and recursively invokes itself on the resulting groups. As mentioned, we need to split the groups of decision

scenarios, according to the state spaces of decisions in them, as well. Thus, we must adjust Method 2.10 so that whenever it encounters a variable, which affects the state spaces of subsequent decisions, it splits the group of decision scenarios accordingly and recurses. Fortunately, the group of decision scenarios of a welldefined IDIT, handled in each invocation of the method, are guaranteed to have the same ordering of decisions and observed variables prior to the first non-instantiated time variable, T'. Therefore, if some of these variables are in the domain of some restriction function, we can split the current group of decision scenarios according to how these variables affect the state space of the decisions, before handling T'. In summary, there is little difference in how a variable in the domain of a restriction function and a time variable should be handled. Consequently, we refer to both kinds of variables as split variables.

Example 2
For the IDIT $\mathcal{I}$ we can identify two split variables: The time variable $\text{end}(D_1)$ and the variables in the domain of r_{D_2}, i.e. C_0. We observe the ordering of split variables to be $\text{end}(D_1) \prec' C_0$. Thus, we must start by splitting on $\text{end}(D_1)$. This task was performed in Section 2.3, and the resulting tree, with the addition of C_0, is displayed in Figure 3.4. Next, we split the decision scenarios on C_0 resulting in the tree shown in Figure 3.5. For ease of reference we have labelled the IDITs in the individual nodes, such that IDIT $\mathcal{I}_{xy}$ is the IDIT found as the y'th child of the x'th child of the root, and IDIT $\mathcal{I}_x$ is the x'th child of the root. Although the leaf nodes pairwise seem to contain similar IDITs, the state spaces of D_2 differ: In the ones, where C_0 is instantiated to c_0, the state space of D_2 consists of d_2 and $\neg d_2$, and in the ones, where C_0 is instantiated to $\neg c_0$, the state space of D_2 consists only of d_2.
We end up with six groups of decision scenarios containing decisions with similar statespaces and similar ordering of variables:

$$\begin{aligned}
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [0;5[\text{ and } \vec{z}^{\downarrow\{C_0\}} = c_0\} = \mathcal{I}_{11},\\
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [0;5[\text{ and } \vec{z}^{\downarrow\{C_0\}} = \neg c_0\} = \mathcal{I}_{12},\\
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [5;10[\text{ and } \vec{z}^{\downarrow\{C_0\}} = c_0\} = \mathcal{I}_{21},\\
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [5;10[\text{ and } \vec{z}^{\downarrow\{C_0\}} = \neg c_0\} = \mathcal{I}_{22},\\
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [0;\infty[\text{ and } \vec{z}^{\downarrow\{C_0\}} = c_0\} = \mathcal{I}_{31}, \text{ and}\\
&\{\vec{z} \in \mathbf{sp}(\boldsymbol{V}^{\mathcal{I}}) | \vec{z}^{\downarrow\{\text{end}(D_1)\}} \in [10;\infty[\text{ and } \vec{z}^{\downarrow\{C_0\}} = \neg c_0\} = \mathcal{I}_{32},
\end{aligned}$$

where the orderings are $\prec'_{\mathcal{I}_{11}}$, $\prec'_{\mathcal{I}_{12}}$, $\prec'_{\mathcal{I}_{21}}$, $\prec'_{\mathcal{I}_{22}}$, $\prec'_{\mathcal{I}_{31}}$, and $\prec'_{\mathcal{I}_{32}}$, respectively. In the rest of this example, we let $\delta_{D_i}^{\boldsymbol{S},\mathcal{I}_{jk}}$ denote the policy for D_i under the ordering of variables $\prec'_{\mathcal{I}_{jk}}$ in the strategy $\boldsymbol{S}$.

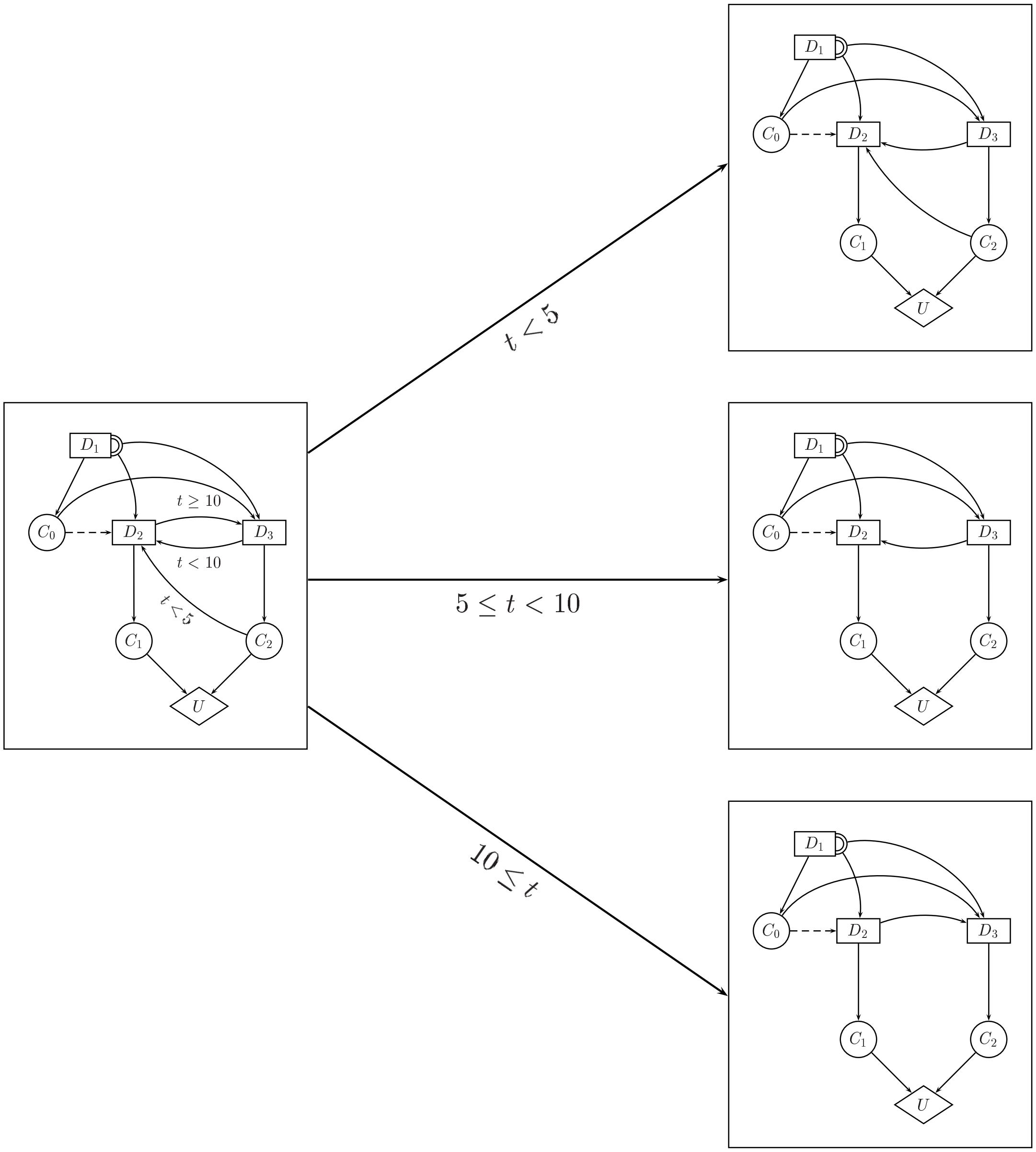

Figure 3.4: *The tree constructed from $\mathcal{I}$ by splitting on end(D_1).*

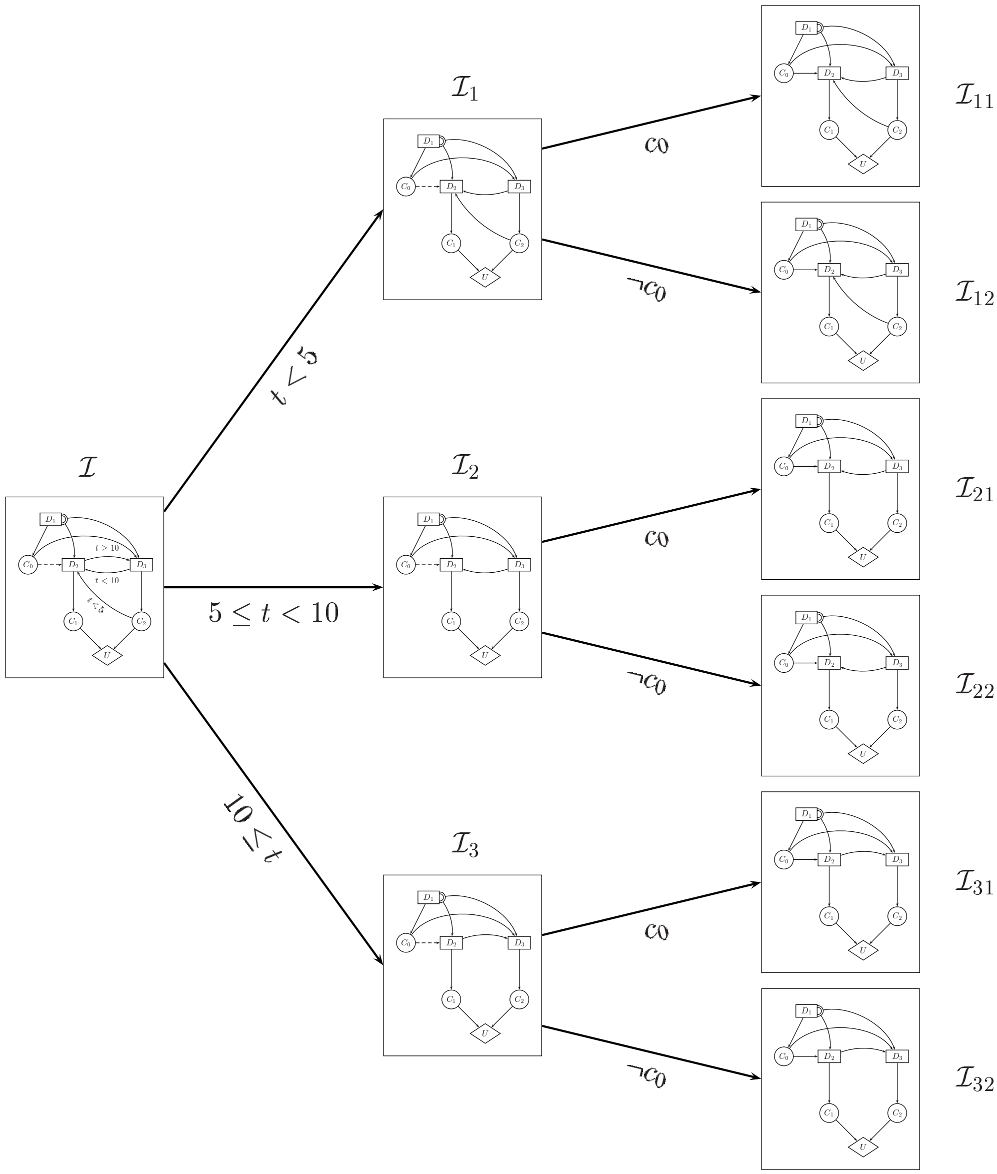

Figure 3.5: *The tree constructed from $\mathcal{I}$ by splitting first on end(D_1) and then C_0.*

We turn out attention to the expression for an optimal strategy, $\boldsymbol{S}'$, for $\mathcal{I}$:

$$\begin{aligned}
\boldsymbol{S}' &= \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} \left(\prod_{\pi\in\boldsymbol{\Pi}^{\mathcal{I}}} \pi \prod_{\phi\in\boldsymbol{\Phi}^{\mathcal{I}}} \phi \prod_{\delta\in\boldsymbol{S}_{o^{\mathcal{I}}(\boldsymbol{V}_{T}^{\mathcal{I}})}} P_\delta \left(\sum_{\psi\in\boldsymbol{\Psi}^{\mathcal{I}}} \psi \right)\right)^{\downarrow\varnothing} \\
&= \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} \int_{-\infty}^{\infty} \sum_{x_{C_0}\in\mathbf{sp}(C_0)} \sum_{x_{C_1}\in\mathbf{sp}(C_1)} \sum_{x_{C_2}\in\mathbf{sp}(C_2)} \sum_{x_{D_1}\in\mathbf{sp}(D_1)} \sum_{x_{D_2}\in\mathbf{sp}(D_2)} \sum_{x_{D_3}\in\mathbf{sp}(D_3)} \\
&\quad f(x_{\text{end}(D_1)}|x_{D_1}) P(x_{c_0}|x_{D_1}) P(x_{c_1}|x_{D_2}) P(x_{c_2}|x_{D_3}) \\
&\quad \cdot P_{\delta_{D_1}}^{\boldsymbol{S},\mathcal{I}[\text{end}(D_1)\mapsto x_{\text{end}(D_1)}]}(x_{D_1}|x_{C_0}, x_{C_1}, x_{C_2}, x_{D_2}, x_{D_3}, x_{\text{end}(D_1)}) \\
&\quad \cdot P_{\delta_{D_2}}^{\boldsymbol{S},\mathcal{I}[\text{end}(D_1)\mapsto x_{\text{end}(D_1)}]}(x_{D_2}|x_{C_0}, x_{C_1}, x_{C_2}, x_{D_1}, x_{D_3}, x_{\text{end}(D_1)}) \\
&\quad \cdot P_{\delta_{D_3}}^{\boldsymbol{S},\mathcal{I}[\text{end}(D_1)\mapsto x_{\text{end}(D_1)}]}(x_{D_3}|x_{C_0}, x_{C_1}, x_{C_2}, x_{D_1}, x_{D_2}, x_{\text{end}(D_1)}) \\
&\quad \cdot U(x_{C_1}, x_{C_2}) dx_{\text{end}(D_1)}.
\end{aligned}$$

As the ordering of D_1 and $\text{end}(D_1)$ relative to every other variable is the same in all six groups of decision scenarios identified above, we may rewrite the expression above to

$$\begin{aligned}
\boldsymbol{S}' = \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} & \sum_{x_{D_1}\in\mathbf{sp}(D_1)} P_{\delta_{D_1}}^{\boldsymbol{S}}(x_{D_1}) \int_{-\infty}^{\infty} f(x_{\text{end}(D_1)}|x_{D_1}) \sum_{x_{C_0}\in\mathbf{sp}(C_0)} P(x_{c_0}|x_{D_1}) \\
& \sum_{x_{C_1}\in\mathbf{sp}(C_1)} \sum_{x_{C_2}\in\mathbf{sp}(C_2)} \sum_{x_{D_2}\in\mathbf{sp}(D_2)} \sum_{x_{D_3}\in\mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2}) P(x_{c_2}|x_{D_3}) \\
& \cdot P_{\delta_{D_2}}^{\boldsymbol{S},\mathcal{I}[\text{end}(D_1)\mapsto x_{\text{end}(D_1)}]}(x_{D_2}|x_{C_0}, x_{C_1}, x_{C_2}, x_{D_1}, x_{D_3}, x_{\text{end}(D_1)}) \\
& \cdot P_{\delta_{D_3}}^{\boldsymbol{S},\mathcal{I}[\text{end}(D_1)\mapsto x_{\text{end}(D_1)}]}(x_{D_3}|x_{C_0}, x_{C_1}, x_{C_2}, x_{D_1}, x_{D_2}, x_{\text{end}(D_1)}) \\
& \cdot U(x_{C_1}, x_{C_2}) dx_{\text{end}(D_1)}.
\end{aligned}$$

We chose to split this sum into six parts, each corresponding to one of the groupings of decision scenarios identified above, by splitting the integration interval and unfolding the sum over states of C_0.

$$
\begin{aligned}
\boldsymbol{S}' = \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} & \sum_{x_{D_1}\in\mathbf{sp}(D_1)} P^{\boldsymbol{S}}_{\delta_{D_1}}(x_{D_1})\Bigg(\int_{-\infty}^{5} f(x_{\mathrm{end}(D_1)}|x_{D_1})\Bigg(P(c_0|x_{D_1}) \\
& \sum_{x_{C_1}\in\mathbf{sp}(C_1)}\sum_{x_{C_2}\in\mathbf{sp}(C_2)}\sum_{x_{D_2}\in\mathbf{sp}(D_2)}\sum_{x_{D_3}\in\mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2})P(x_{c_2}|x_{D_3}) \\
& \cdot P_{\delta^{\boldsymbol{S},\mathcal{I}_{11}}_{D_2}}(x_{D_2}|c_0, x_{C_2}, x_{D_1}, x_{D_3}, x_{\mathrm{end}(D_1)})P_{\delta^{\boldsymbol{S},\mathcal{I}_{11}}_{D_3}}(x_{D_3}|c_0, x_{D_1}, x_{\mathrm{end}(D_1)}) \\
& \cdot U(x_{C_1}, x_{C_2})\Bigg) + \Bigg(P(\neg c_0|x_{D_1}) \sum_{x_{C_1}\in\mathbf{sp}(C_1)}\sum_{x_{C_2}\in\mathbf{sp}(C_2)} \\
& \sum_{x_{D_2}\in\mathbf{sp}(D_2)}\sum_{x_{D_3}\in\mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2})P(x_{c_2}|x_{D_3}) \\
& \cdot P_{\delta^{\boldsymbol{S},\mathcal{I}_{12}}_{D_2}}(x_{D_2}|\neg c_0, x_{C_2}, x_{D_1}, x_{D_3}, x_{\mathrm{end}(D_1)})P_{\delta^{\boldsymbol{S},\mathcal{I}_{12}}_{D_3}}(x_{D_3}|\neg c_0, x_{D_1}, x_{\mathrm{end}(D_1)}) \\
& \cdot U(x_{C_1}, x_{C_2})\Bigg)dx_{\mathrm{end}(D_1)} + \\
& \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
& + \int_{10}^{\infty} f(x_{\mathrm{end}(D_1)}|x_{D_1})\Bigg(P(c_0|x_{D_1}) \\
& \sum_{x_{C_1}\in\mathbf{sp}(C_1)}\sum_{x_{C_2}\in\mathbf{sp}(C_2)}\sum_{x_{D_2}\in\mathbf{sp}(D_2)}\sum_{x_{D_3}\in\mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2})P(x_{c_2}|x_{D_3}) \\
& \cdot P_{\delta^{\boldsymbol{S},\mathcal{I}_{31}}_{D_2}}(x_{D_2}|c_0, x_{D_1}, x_{\mathrm{end}(D_1)})P_{\delta^{\boldsymbol{S},\mathcal{I}_{31}}_{D_3}}(x_{D_3}|c_0, x_{D_1}, x_{D_2}, x_{\mathrm{end}(D_1)}) \\
& \cdot U(x_{C_1}, x_{C_2})\Bigg) + \Bigg(P(\neg c_0|x_{D_1}) \sum_{x_{C_1}\in\mathbf{sp}(C_1)}\sum_{x_{C_2}\in\mathbf{sp}(C_2)} \\
& \sum_{x_{D_2}\in\mathbf{sp}(D_2)}\sum_{x_{D_3}\in\mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2})P(x_{c_2}|x_{D_3}) \\
& \cdot P_{\delta^{\boldsymbol{S},\mathcal{I}_{32}}_{D_2}}(x_{D_2}|\neg c_0, x_{D_1}, x_{\mathrm{end}(D_1)})P_{\delta^{\boldsymbol{S},\mathcal{I}_{32}}_{D_3}}(x_{D_3}|\neg c_0, x_{D_1}, x_{D_2}, x_{\mathrm{end}(D_1)}) \\
& \cdot U(x_{C_1}, x_{C_2})\Bigg)dx_{\mathrm{end}(D_1)}\Bigg).
\end{aligned}
\tag{3.1}
$$

At this point the structure of the expression, we need to maximize, is similar to that of the split tree in Figure 3.5. We have not used any special properties of the involved functions, which suggests that a splitting of the expected utility of any IDIT, can be constructed in a similar fashion.

	d_2	$\neg d_2$
c_2	21	28
$\neg c_2$	28.5	18

Table 3.4: *The utility function U^*.*

At this point we need to calculate the sums in the six subexpressions, of which

$$\sum_{x_{C_1} \in \mathbf{sp}(C_1)} \sum_{x_{C_2} \in \mathbf{sp}(C_2)} \sum_{x_{D_2} \in \mathbf{sp}(D_2)} \sum_{x_{D_3} \in \mathbf{sp}(D_3)} P(x_{c_1}|x_{D_2})P(x_{c_2}|x_{D_3})$$
$$\cdot P_{\delta_{D_2}^{S,\mathcal{I}_{11}}}(x_{D_2}|c_0, x_{C_2}, x_{D_1}, x_{D_3}, x_{\text{end}(D_1)})P_{\delta_{D_3}^{S,\mathcal{I}_{11}}}(x_{D_3}|c_0, x_{D_1}, x_{\text{end}(D_1)})$$
$$\cdot U(x_{C_1}, x_{C_2}) \tag{3.2}$$

is one, before we can move on to summing over C_0 and D_1 and integrating over end(D_1). This is no coincidence. If we study the IDITs in the leaves of the split tree in Figure 3.5, C_0, D_1, and end(D_1) are all either instantiated or prior to an instantiated variable in the ordering $\prec'_{\mathcal{I}}$. Variables which are neither instantiated nor prior to an instantiated variable in an IDIT, $\mathcal{I}$, we call *free variables*. We focus on the subexpression in (3.2), where the free variables, thus, are C_1, C_2, D_2, and D_3. By rearranging sums we get

$$\sum_{x_{D_3} \in \mathbf{sp}(D_3)} P_{\delta_{D_3}^{S,\mathcal{I}_{11}}}(x_{D_3}|c_0, x_{D_1}, x_{\text{end}(D_1)}) \sum_{x_{C_2} \in \mathbf{sp}(C_2)} P(x_{c_2}|x_{D_3})$$
$$\cdot \sum_{x_{D_2} \in \mathbf{sp}(D_2)} P_{\delta_{D_2}^{S,\mathcal{I}_{11}}}(x_{D_2}|c_0, x_{C_2}, x_{D_1}, x_{D_3}, x_{\text{end}(D_1)}) \sum_{x_{C_1} \in \mathbf{sp}(C_1)} P(x_{c_1}|x_{D_2})U(x_{C_1}, x_{C_2}). \tag{3.3}$$

Thus we must sum over variables in the order C_1, D_2, C_2, and then D_3. This ordering is consistent with the inverse of $\prec'_{\mathcal{I}_{11}}$.

Elimination of Variables

We evaluate the the sub expression

$$\sum_{x_{C_1} \in \mathbf{sp}(C_1)} P(x_{C_1}|x_{D_2})U(x_{C_1}, x_{C_2})$$

right away, and get a utility function, U^*, defined over C_2 and D_2. U^* is shown in Table 3.4. This we refer to as *marginalizing out* C_1 or, equivalently, *eliminating* C_1 from $\mathcal{I}_{11}$.
Replacing

$$\sum_{x_{C_1} \in \mathbf{sp}(C_1)} P(x_{C_1}|x_{D_2})U(x_{C_1}, x_{C_2})$$

with $U^*(x_{C_2}, x_{D_2})$ we get

$$\sum_{x_{D_3} \in \mathbf{sp}(D_3)} P_{\delta_{D_3}^{\boldsymbol{S},\mathcal{I}_{11}}}(x_{D_3}|c_0, x_{D_1}, x_{\mathrm{end}(D_1)}) \sum_{x_{C_2} \in \mathbf{sp}(C_2)} P(x_{c_2}|x_{D_3})$$
$$\cdot \sum_{x_{D_2} \in \mathbf{sp}(D_2)} P_{\delta_{D_2}^{\boldsymbol{S},\mathcal{I}_{11}}}(x_{D_2}|c_0, x_{C_2}, x_{D_1}, x_{D_3}, x_{\mathrm{end}(D_1)})U^*(x_{C_2}, x_{D_2}).$$

To find a policy $P_{\delta_{D_2}^{\boldsymbol{S},\mathcal{I}_{11}}}$ which maximizes this expression for all states of D_1, all states of $\mathrm{end}(D_1)$ in $[-\infty; 5[$, and C_0 being c_0, we can focus on the last part of the expression:

$$\sum_{x_{D_2} \in \mathbf{sp}(D_2)} P_{\delta_{D_2}^{\boldsymbol{S},\mathcal{I}_{11}}}(x_{D_2}|D_1, \mathrm{end}(D_1), c_0, x_{C_2}, x_{D_3})U^*(x_{C_2}, x_{D_2}). \tag{3.4}$$

This is equivalent to the expression

$$U^*(x_{C_2}, \delta_{D_2}^{\boldsymbol{S},\mathcal{I}_{11}}(D_1, \mathrm{end}(D_1), c_0, x_{C_2}, x_{D_3})),$$

so, in order to maximize it, we can write

$$\max_{d \in \mathbf{sp}(D_2)} U^*(x_{C_2}, d),$$

which yield a value of 28 if x_{C_2} is c_2 and 28.5 otherwise. The corresponding states of D_2 can be found by

$$\arg \max_{d \in \mathbf{sp}(D_2)} U^*(x_{C_2}, d),$$

yielding the choice $\neg d_2$ in case that C_2 is c_2, and d_2 otherwise. Hence, we have identified a policy which maximizes (3.4):

$$\delta_{D_2}^{\boldsymbol{S},\mathcal{I}_{11}}(D_1, \mathrm{end}(D_1), C_0, C_2, D_3) = \begin{cases} d_2 & \text{if } C_2 = \neg c_2 \\ \neg d_2 & \text{if } C_2 = c_2. \end{cases}$$

No matter how the remaining parts of (3.3) may evaluate, this policy must be part of an optimal strategy. This is because it is conditioned on $\mathrm{end}(D_1)$ being less than 5 and C_0 being c_0 and, thus, only affects the part of Equation (3.1) that constitutes (3.4), which it maximizes.

By substituting the expression in (3.4) with a utility function over C_2, which yields the value 28 if C_2 is c_2 and 28.5 if C_2 is $\neg c_2$, we can disregard D_2 in (3.3), henceforth. We refer to this as marginalizing D_2 out, or eliminating D_2 from $\mathcal{I}_{11}$. Note that the process of marginalizing out a decision is, thus, different from that of marginalizing out a chance variable. In the former we maximize over states and in the latter we sum. Furthermore, in the former we note, for each configuration of past variables, the state which yields the maximum utility.

Continuing eliminating free variables in $\mathcal{I}_{11}$ in the same manner, we must marginalize out, first C_2, and then D_3. We skip the details, which are similar to those for marginalizing out

C_1 and D_2, and simply state that the resulting optimal policy is

$$\delta_{D_3}^{\boldsymbol{S},\mathcal{I}_{11}}(D_1, \text{end}(D_1), C_0) = \neg d_3,$$

meaning that as long as end(D_1) is less than 5 and C_0 is c_0 we should always choose $\neg d_3$ at D_3. The expected utility of this is 28.45, which can easily be verified.
The process just described corresponds to traditional elimination of variables in an influence diagram. Actually, it corresponds exactly to eliminating variables from $\mathcal{I}_{11}$ interpreted as an influence diagram. We can perform similar processes on the IDITs in the remaining leaves of the split tree. The resulting expression is

$$\begin{aligned}\boldsymbol{S}' = \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} \sum_{x_{D_1}\in\mathbf{sp}(D_1)} P^{\boldsymbol{S}}_{\delta_{D_1}}(x_{D_1}) \Bigg(\\ \cdot \int_{-\infty}^{5} f(x_{\text{end}(D_1)}|x_{D_1}) \Big(P(c_0|x_{D_1}) \cdot 28.5\Big) + \Big(P(\neg c_0|x_{D_1}) \cdot 27.75\Big) dx_{\text{end}(D_1)} \\ \cdot + \int_{5}^{10} f(x_{\text{end}(D_1)}|x_{D_1}) \Big(P(c_0|x_{D_1}) \cdot 28.2\Big) + \Big(P(\neg c_0|x_{D_1}) \cdot 26.5\Big) dx_{\text{end}(D_1)} \\ \cdot + \int_{10}^{\infty} f(x_{\text{end}(D_1)}|x_{D_1}) \Big(P(c_0|x_{D_1}) \cdot 28.2\Big) + \Big(P(\neg c_0|x_{D_1}) \cdot 26.5\Big) dx_{\text{end}(D_1)} \Bigg).\end{aligned}$$

We say that the maximum expected utilities of the IDITs in the leaves of the split tree, have been *absorbed* into the IDITs in the internal nodes of the tree. Notice that the maximum expected utilities absorbed from the leaves corresponding to end(D_1) being greater than 5, are the same. This is because, in both cases, no information is obtained by the decision taker between deciding upon D_2 and D_3, and the ordering of the two, therefore, does not affect the resulting utility.

Next, we eliminate C_0, which results in the following expression

$$
\begin{aligned}
\boldsymbol{S}' =& \arg\max_{\boldsymbol{S}\in\boldsymbol{\Delta}_{\mathcal{I}}} \\
=& P^{\boldsymbol{S}}_{\delta_{D_1}}(d_1)\Bigg(\int_0^5 f(x_{\text{end}(D_1)}|d_1)27.89dx_{\text{end}(D_1)} \\
&+ \int_5^{10} f(x_{\text{end}(D_1)}|d_1)27.8dx_{\text{end}(D_1)} \\
&+ \int_{10}^{\infty} f(x_{\text{end}(D_1)}|d_1)27.8dx_{\text{end}(D_1)}\Bigg) \\
&+ P^{\boldsymbol{S}}_{\delta_{D_1}}(\neg d_1)\Bigg(\int_0^5 f(x_{\text{end}(D_1)}|(\neg d_1)28.24dx_{\text{end}(D_1)} \\
&+ \int_5^{10} f(x_{\text{end}(D_1)}|(\neg d_1)27.925dx_{\text{end}(D_1)} \\
&+ \int_{10}^{\infty} f(x_{\text{end}(D_1)}|(\neg d_1)27.925dx_{\text{end}(D_1)}\Bigg).
\end{aligned} \tag{3.5}
$$

According to the temporal order, $\prec'$, we should next eliminate end(D_1). Studying the expression in Equation (3.5), however, reveals that this is no easy task. Even though we can move the constants outside the integrals, we are still left with evaluating integrals over the density function of end(D_1). As end(D_1) follows a χ^2-distribution, of which no known closed form expression, presently, exists[Nist, 2003], this is impossible.

As we are really not that interested in the actual maximum expected utility of $\mathcal{I}$, but rather a strategy which maximizes this, we can employ approximation techniques instead. One such technique is sampling, in which we, for each possible policy, δ_{D_1}, sample the value of

$$
\begin{aligned}
&P_{\delta_{D_1}}(d_1)\left(\int_0^5 f(x_{\mathrm{end}(D_1)}|d_1)27.89dx_{\mathrm{end}(D_1)}\right.\\
&+\int_5^{10} f(x_{\mathrm{end}(D_1)}|d_1)27.8dx_{\mathrm{end}(D_1)}\\
&\left.+\int_{10}^{\infty} f(x_{\mathrm{end}(D_1)}|d_1)27.8dx_{\mathrm{end}(D_1)}\right)\\
&+P_{\delta_{D_1}}(\neg d_1)\left(\int_0^5 f(x_{\mathrm{end}(D_1)}|(\neg d_1)28.24dx_{\mathrm{end}(D_1)}\right.\\
&+\int_5^{10} f(x_{\mathrm{end}(D_1)}|(\neg d_1)27.925dx_{\mathrm{end}(D_1)}\\
&\left.+\int_{10}^{\infty} f(x_{\mathrm{end}(D_1)}|(\neg d_1)27.925dx_{\mathrm{end}(D_1)}\right),
\end{aligned}
$$

a fixed number of times. Then we calculate the average of the samples taken for each policy and choose the policy with the maximum average as the optimal one. A problem arising from applying this technique is that it assumes a fixed configuration of past variables. If there would happen to be some time variable, T, in the past of D_1, we would, theoretically, need to sample for an infinite number of configurations of past variables, which is a perpetual task. This could be remedied by discretizing T, but the choice of discretization intervals of T is not obvious. Furthermore, the technique can be time and space consuming, as the number of samples in some cases would need to be high in order to obtain a satisfying degree of confidence in the result. [Charnes and Shenoy, 2003] present a method that utilizes sampling for influence diagrams, which could allow discretization of time variables to be of fine granularity, while leaving the calculation of expected utility computationally feasible. [Broe and Jeppesen, 2003] presents a solution method for IDITs utilizing sampling.

A more crude approach is to only allow integrable density functions for time variables. Furthermore, we would have to require that the resulting functions from integrating over these would be integrable too. Similarly, all kinds of utility functions, which could arise during elimination of variables, should be integrable. Clearly, this approach limits the number of decision problems that can be specified and solved using IDITs severely.

A more flexible take on this last approach is to approximate all continuous functions by polynomials, as these are infinitely integrable, and sums and products of polynomials are polynomials as well. The process of converting an arbitrary continuous function to a polynomial can be time consuming, though, and some functions would need to be approximated by polynomials of a very high degree, implying requirements on time and space for a solution method. However, approximation using polynomials has some advantages as well, most having to do with avoidance of discretization issues. When we approximate using polynomials we need to be aware of the nature of these when the variable in their domain goes to infinity or minus infinity. In most cases the value of the polynomial will go to either infinity or minus infinity as well. Therefore, we must limit the areas of integration. For most decision problems the span of time is assumed to start at some constant, such as 0, and the lower limit is therefore not a problem.

Example 2

In order to round off this example, we choose to approximate the density functions for end(D_1) by polynomials, and limit the areas of integration to the values in $[0; 40]$. The resulting expression is

$$\begin{aligned} \boldsymbol{S}' = &\arg \max_{\boldsymbol{S} \in \boldsymbol{\Delta}_{\mathcal{I}}} \\ &P^{\boldsymbol{S}}_{\delta_{D_1}}(d_1) \cdot (0.58 \cdot 27.89 + 0.34 \cdot 27.8 + 0.07 \cdot 27.8) \\ &+P^{\boldsymbol{S}}_{\delta_{D_1}}(\neg d_1) \cdot (0.11 \cdot 28.24 + 0.45 \cdot 27.925 + 0.42 \cdot 27.925) \\ &= \arg \overset{\boldsymbol{S}}{\max_{\boldsymbol{S} \in \boldsymbol{\Delta}_{\mathcal{I}}}} P_{\delta_{D_1}}(d_1) \cdot 27.57 + P^{\boldsymbol{S}}_{\delta_{D_1}}(\neg d_1) \cdot 27.40. \end{aligned}$$

Thus, the optimal policy for D_1, is to choose d_1, although the expected utility of choosing $\neg d_1$ is roughly the same.

Preliminaries for the Solution Method

The purpose of the example just given was to introduce the main structure of the solution method, and to hint at why it identifies an optimal strategy when invoked on an IDIT. However, several problems arising from eliminating variables in IDITs have not been touched upon during the example. We do so, as they become relevant in the presentation of the method below, and hope that the reader, through the example, has obtained the breath of view necessary to focus on these details instead of the overall structure of the method. The solution method described here solves only a subset of IDITs. Throughout the description below we need to introduce a set of assumptions. Whenever this need arises we emphasize the assumption in a paragraph by itself and comment on the restrictions it implies. Before we present the method, some preliminaries need to be laid down, though.

As the elaborate example, concerning solution of the IDIT in Figure 3.2, showed, we have to handle situations in which a utility function over a time variable is only piecewise continuous. We introduce a set of formal notation and some concepts for handling such functions. We start by defining a *partition* of the real numbers as a finite subset of the real numbers, $\boldsymbol{I} = \{a_1, \ldots, a_{n+1}\}$, where $i < j$ implies $a_i < a_j$, and say that it *generates* a series of $n + 2$ intervals,

$$]-\infty; a_1[, [a_1; a_2[, \ldots, [a_n; a_{n+1}[, [a_{n+1}; \infty[.$$

We refer to an interval, $[a_i; a_{i+1}[$, as the i'th interval of $\boldsymbol{I}$. The series of intervals generated by the empty set consists of a single interval, $]-\infty; \infty[$. We use the notation $\boldsymbol{I}^{<x}$ to denote the number of elements in $\boldsymbol{I}$ which are smaller than or equal to x. For example, we have that the partition $\boldsymbol{I} = \{2, 7\}$ generates the intervals $]-\infty; 2[$, $[2; 7[$, and $[7; \infty[$, and that $\boldsymbol{I}^{<4}$ is 1 and $\boldsymbol{I}^{<7}$ is 2. We thus have the relationship: If x is in $[a_i; a_{i+1}[$ then $\boldsymbol{I}^{<x}$ is i.

Let $\boldsymbol{I}_{f'} = \{a_1, \ldots, a_{n+1}\}$ be a partition and $(f_0 : \mathbb{R} \to \mathbb{R}, \ldots, f_{n+1} : \mathbb{R} \to \mathbb{R})$ a series of continuous functions. Then the function $f' : \mathbb{R} \to \mathbb{R}$, where

$$f'(x) = \begin{cases} f_0(x) & \text{if } x \in]-\infty; a_1[\\ f_1(x) & \text{if } x \in [a_1; a_2[\\ \vdots & \vdots \\ f_n(x) & \text{if } x \in [a_n; a_{n+1}[, \\ f_{n+1}(x) & \text{if } x \in [a_{n+1}; \infty[, \end{cases}$$

is the *piecewise continuous function* over the partition $\boldsymbol{I}_{f'}$ of the functions $(f_0, \ldots, f_{n+1})$. We use subscripts on f' to access the functions in $(f_0, \ldots, f_{n+1})$, i.e. f'_i denotes the function f_i. As an example, we have plotted part of the piecewise continuous function, f, over the partition $\boldsymbol{I}_f = \{2, 7\}$ of the functions $(0.5x + 2, 10/x, -0.15x^2 + 15)$ in Figure 3.6

Notice that we have chosen to have all intervals of the form $[a; b[$, instead of, say, $]a; b]$. That is, the cutting point on the real line is always included in the interval containing the higher numbers. This is a notational convenient convention and in all intents and purposes does not affect the reasoning presented, as the intervals are used only as boundaries of integral domains. Considered from a different perspective, we can also argue that, as time variables are continuous, the probability of a decision scenario, in which a time variable takes on a specific point in time is zero, and thus, it does not matter which group of decision scenarios it is included in.

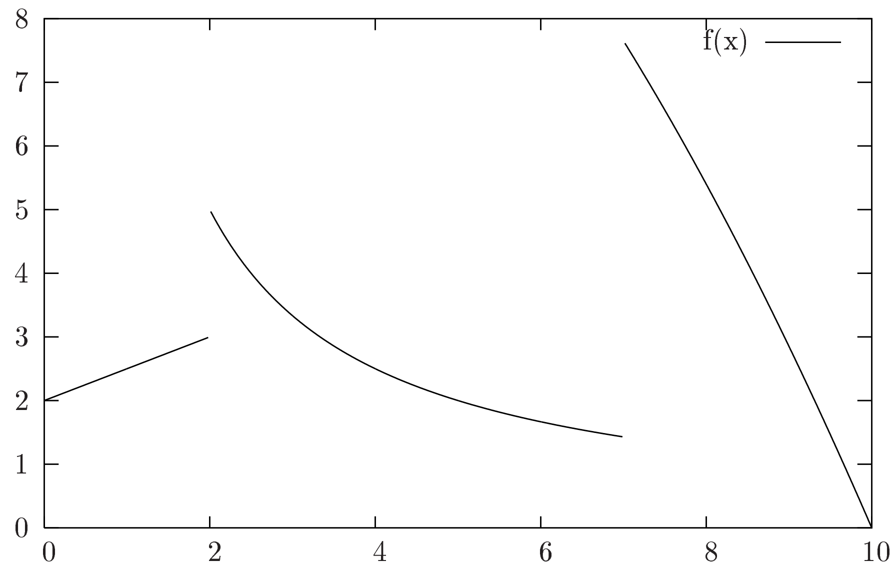

Figure 3.6: *A piecewise continuous function.*

During the solution process we need to calculate sums of piecewise continuous functions. Addition of piecewise continuous functions are somewhat cumbersome. One needs to take into account that they might not be defined over the same intervals: Let f and f' be piecewise continuous functions over the partitions $\boldsymbol{I}_f$ and $\boldsymbol{I}_{f'}$, respectively. Then the sum $f + f'$ is the piecewise continuous function over the partition $\boldsymbol{I}_{f+f'} = \boldsymbol{I}_f \cup \boldsymbol{I}_{f'} = \{a_1, \dots, a_{n+1}\}$, of the functions

$$(f + f')_i = f_{\boldsymbol{I}_f^{<a_i}} + f'_{\boldsymbol{I}_{f'}^{<a_i}},$$

for all i in $\{1, \dots, n+1\}$, and $(f + f')_0 = f_0 + f'_0$. Notice that if we regard any continuous function, f_c, as being piecewise continuous over the partition $\varnothing$ and f_0 being f_c, then the above definition corresponds to the standard definition of the sum of two continuous functions. Furthermore, a sum of a piecewise continuous function, f, and a continuous function, f_c, results in a piecewise continuous function, $f + f_c$, over the same defining intervals as f, with each $(f + f_c)_i$ being $f_c + f_i$. The product of piecewise continuous functions are defined analogous to the sum, except that multiplication of individual functions are applied instead of addition. Thus, we may in all matters regard continuous functions as piecewise continuous function, and therefore, do not distinguish between them during manipulation of utilities.

Given a piecewise continuous function, f, we, furthermore, define the short hand notation $f^{\downarrow[a;b[}$ to mean the product $f \cdot g$, where g is the piecewise continuous function over the partition $\boldsymbol{I}_g = \{a, b\}$ of $(g_0 = 0, g_1 = 1, g_2 = 0)$. Intuitively, $f^{\downarrow[a;b[}$ takes on the value of f in the interval $[a; b[$ and the value 0 everywhere else. We denote it as the *projection* of f down-to the interval $[a; b[$.

The main idea of the solution method is to approximate continuous functions by polynomials, as manipulations of these, such as addition, multiplication, and differentiation, can be carried out mechanically. We work with polynomials over one or two variables. A polynomial, p, of degree n over one variable, x, is defined as

$$p(x) = \sum_{i=0}^{n} C[p]_i x^i,$$

where $C[p]_0, \ldots, C[p]_n$ are real numbers, which we call coefficients. The C[function-name]-notation we use throughout the report when dealing with polynomials. Similarly, a polynomial, p, of degree (n, m) over two variables, x and y, is defined as

$$p(x, y) = \sum_{i=0}^{n} \sum_{i=0}^{m} C[p]_{ij} x^i y^j,$$

where $C[p]_{00}, \ldots, C[p]_{nm}$ are real numbers.

As stated in Chapter 1, we employ approximation in the form of Taylor's series. We define these formally, and refer the interested reader to [Apostol, 1974] for further information on them.

Definition 3.8

Let f be a function, which is infinitely differentiable over the interval $[a; b]$. Then the Taylor's series *of f on $[a; b]$ about a point, c, in $[a; b]$ is the polynomial*

$$\sum_{i=0}^{\infty} \frac{f^{(i)}(c)}{i!} (x - c)^i,$$

where $f^{(i)}(c)$ is the i'th derivative of f at c.

The point c is usually called the *point of expansion* of the series.
As we cannot deal with infinite polynomials, we utilize that we can rewrite

$$\begin{aligned}\sum_{i=0}^{\infty} \frac{f^{(i)}(c)}{i!}(x-c)^i &= \sum_{i=0}^{n} \frac{f^{(i)}(c)}{i!}(x-c)^i + \sum_{j=n+1}^{\infty} \frac{f^{(j)}(c)}{j!}(x-c)^j \\ &= \sum_{i=0}^{n} \frac{f^{(i)}(c)}{i!}(x-c)^i + r_n(x).\end{aligned}$$

It can be shown [Apostol, 1974] that $\lim_{n\to\infty} r_n(x) = 0$, if there exists some constant, k, such that $|f^{(n)}(x)| \leq k^n$, for all x in the interval $[a;b]$. Therefore, when dealing with utility and density functions, for which this is true, we can chose a Taylor's series of a finite degree as an approximation to the original function. Throughout the remainder of the report we assume all Taylor's series are of a fixed degree, N. This discussion imposes the following assumption on the given utility and density functions, as well as probability distributions dependent on time variables:

Assumption 1
Any density function for a time variable, utility function over a time variable, or probability distribution with a time variable in its domain, $f : \mathbf{sp}(\boldsymbol{Z}) \times \mathbb{R} \to \mathbb{R}$, must be differentiable an arbitrary number of times with respect to the time variable. Furthermore, for each configuration $\vec{z}$ of the variables in $\boldsymbol{Z}$, there must exist some constant, k, such that $|f(\vec{z})^{(n)}(x)| \leq k^n$, for all x in $\mathbb{R}$.

This assumption is rather strict, but due to an additional assumption, introduced below, we can loosen it a bit.
Of course, an approximation, f', of a density function, f, rarely is a density function itself, i.e. $\int_{-\infty}^{\infty} f'(t)dt$ do not necessarily evaluate to 1. The approximation can be transformed into a density function by dividing each coefficient of f' by $\int_{-\infty}^{\infty} f'(t)dt$, though. Such an operation we refer to as *normalizing* f'. Similarly, the Taylor series of a probability distribution, $P(C|\boldsymbol{D}, T)$ for a given configuration, $\vec{d}$, of the variables in $\boldsymbol{D}$, $P'(c_1|\vec{d}, T), \ldots, P'(c_n|\vec{d}, T)$, do not necessarily sum to 1 for each t in $\mathbf{sp}(T)$. This can be solved by adding

$$1 - \sum_{c_i \in \mathbf{sp}(C)} P(c_1|\vec{d}, T)$$

to one of the series, such as $P'(c_1|\vec{d}, T)$.
The value of N we assume to be set by the user of our method, but some dynamic

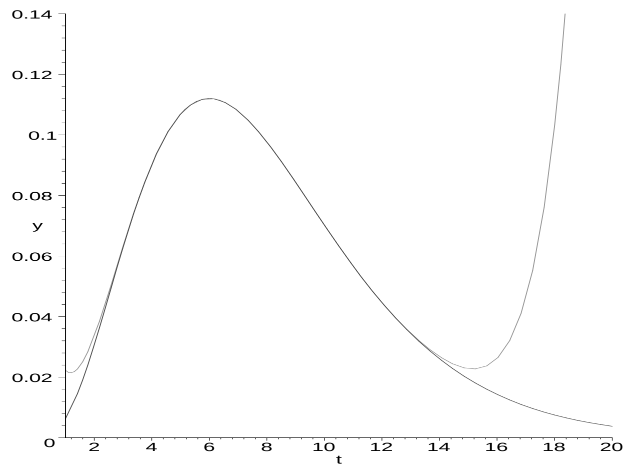

Figure 3.7: *The Taylor's series (fair line) of a density function (dark line) for a χ^2-distributed variable.*

adjustment method could be incorporated in the method some time in the future. We do not touch upon this again in this report. We use $\mathcal{T}_f([a;b],c)$ to denote the Taylor's series of f on $[a;b]$ about c. In Figure 3.7 the density function of a variable following a χ^2-distribution with 6 degrees of freedom as well as a Taylor's series, on the real line of degree 10 about 8, of this are shown.

Specifically catering for piecewise continuous functions, furthermore, allows us to approximate utility functions and probability distributions using Taylor's series piece by piece. Obviously, this is useful if the function is specified piece by piece, but it can also result in faster approximation, as lower values of N can be used with little loss in precision. As an example of this, we present two approximations of the function $f(t) = e^t t^6 \sin t$ over the interval $[0; 20]$ in Figures 3.8 and 3.9. The first approximation is based on the first 40 derivatives of f, whereas the second uses only the first 10, but uses them four times. In both cases 40 evaluations of derivatives in a point need to be evaluated, but whereas the first approach needs calculation of 40 derivatives the second method needs only 10. It is, thus, faster to approximate the function piece by piece. Of course there is a limit to the gains in approximation speed, as N cannot be less than 0. Furthermore, the actual speed gain or penalty of piecewise approximation in the solution method itself is not obvious. We do not present a full time-complexity analysis of our solution method, but do evaluate complexity of the more intriguing steps, to hint at the complexity of the issue. Unfortunately, we cannot allow density functions to be approximated piecewise, which is further elaborated on below.

For notational convenience, we introduce the notation $\mathcal{T}_f$, where f is a piecewise

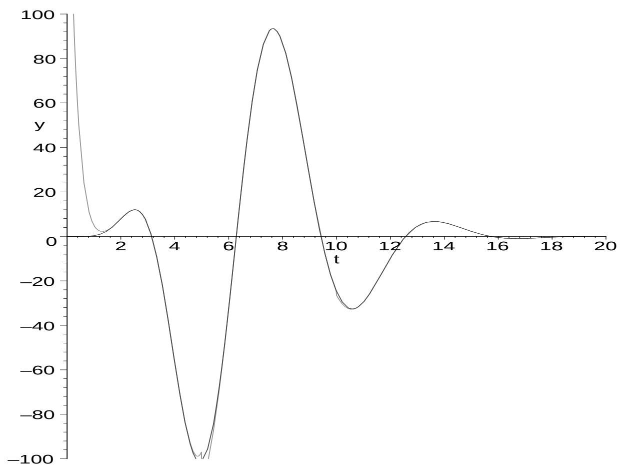

Figure 3.8: *A Taylor's series (fair line) of the function* $e^t t^6 \sin t$ *(dark line) over the interval* $[0;20]$ *with* N *being* 40.

continuous function over the partition $\boldsymbol{I}_f$ of the functions $(f_0,\ldots,f_{n+1})$, to mean the piecewise continuous function over the partition $\boldsymbol{I}_f$ of the functions $(f_0^*,\ldots,f_{n+1}^*)$, where

$$f_i^* = \mathcal{T}_{f_i}\left([t_i;t_{i+1}[, \frac{t_{i+1}-t_i}{2}\right),$$

for i in $\{1,\ldots,n\}$,

$$f_0^* = \mathcal{T}_{f_0}\left(]\infty;t_1[, t_1-1\right),$$

and

$$f_{n+1}^* = \mathcal{T}_{f_{n+1}}\left([t_{n+1};\infty[, t_{n+1}\right).$$

As we approximate all utility functions by polynomials of a finite degree, the approximations will invariably start to monotonically decrease or increase after some point in time. In Example 2 we needed to integrate over utility functions from zero to infinity, and this will also be necessary in the method presented below. This cannot be performed when utility functions decrease or increase as described, and we, therefore, assume that the IDITs we solve have a time limit, t_e, before which the decision taker wants the decision process completed at all costs. That is, all utility functions dependent on time, either directly or indirectly, yields 0 for points in time after t_e, no matter the configuration of other variables in the IDIT. By "indirectly", we mean that a utility is d-connected to the time variable given the set of observed variables and decisions prior to the time variable in the temporal ordering.

This assumption is not as restrictive as it appears. First of all, if a decision taker

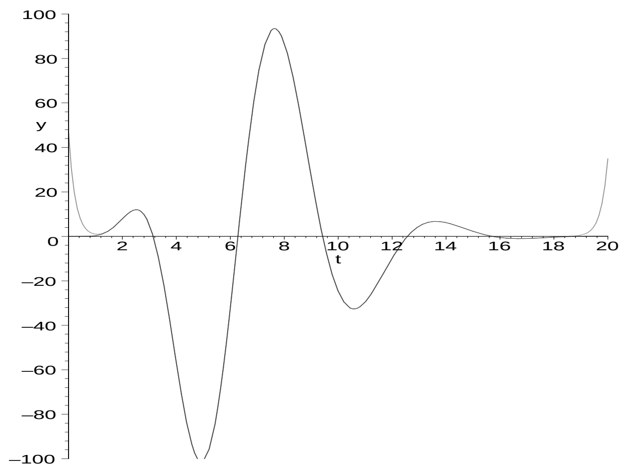

Figure 3.9: *A piecewise Taylor's series (fair line) of the function* $e^t t^6 \sin t$ *(dark line) over the intervals* $[0;5[$, $[5;10[$, $[10;15[$, *and* $[15;20]$ *with* N *being* 10.

is a human being there is a very natural limit after which the utility should be of no concern to him. Second, if an IDIT contains no wait decisions, the probability of decision scenarios where the last time variable takes on a high value, will get increasingly smaller, and thus, the contribution of these to the expected utility of any decision is negligible. If an IDIT does contain a wait decision, and we cannot apply this argument, we rely on most naturally occurring utilities dependent on time being highest for initial points in time and steadily decresing afterwards. That is, many decision takers prefer a payoff today rather than tomorrow. In such cases, there must come some point in time, t_e^*, after which each utility for each configuration of other variables either is so low that the differences from one configuration to the other becomes negligable, or it is constant. In both scenarios, the part of the decision problem that succeeds t_e^* can be disregarded with little impact on the result.

Assumption 2

A time limit to the decision process, t_e*, must be fixed before an IDIT can be solved. All utilities dependent on time variables, must yield the value* 0 *for all points in time after* t_e*. Likewise, utilities indirectly dependent on time, such as utilities dependent on an ordinary chance variable which in turn depends on a time variable, should yield* 0 *for points in time after* t_e*.*

Given this assumption and a utility function or density function, f, in an IDIT, we may approximate $f^{\downarrow[0;t_e[}$, instead of f, when identifying Taylor series. Thus, the degree of approximation, N, are closely bound to the value of t_e. When t_e is raised, either N or the acceptable inaccuracy stemming from approximation would need to be raised as well. Furthermore, Assumption 2 and the observation that utility functions can be approximated piece by piece, allow us to loosen the wording of Assumption 1 to only require density functions, utility functions, and probability distributions to be differentiable an arbitrary number of times over the interval $[0; t_e[$, as opposed to the real line. Furthermore, we can allow utility functions and probability distributions to fulfill this requirement on a piece by piece basis.

Since we assume all utility functions are 0 outside the interval $[0; t_e[$, for any given configuration of other variables, we must also assume that each utility does not take on negative values in the interval $[0; t_e[$. If this was not assumed, we would have scenarios in which the points in time outside this interval yield more attractive utilities than some of those inside it. However, by adding a large constant to such utilities beforehand, we can disregard this problem. Furthermore, as utilities generally are required to be unique up to any given positive linear transformation, which adding a constant is, this is not a limitation.

With this notation and understanding of approximation methods used, we can move on to the actual algorithm.

Solution Method

The method we propose for solving IDITs is inspired by the method for solving asymmetric influence diagrams presented in [Nielsen and Jensen, 2000]. Throughout the description, we use lower case greek letters to denote functions which are not necessarily part of the original specification of the IDIT, but possibly results of previous calculations. Method 3.9 is the method that solves its input IDIT, $\mathcal{I}$, and returns an optimal strategy, $\boldsymbol{S}$. It is really a shell for the solution method itself, and takes care of initialization of the IDIT in preparation to the actual solving process. Method 3.10, which is the main part of the solution method, takes as argument an IDIT, $\mathcal{I}$, with a realization, $(\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Pi}, \mathbf{\Gamma})$, a starting point in time, t_s, and an end point in time t_e, and it produces a set of policies for free decisions in $\mathcal{I}$, $\boldsymbol{S}$, as well as two sets of functions, $\mathbf{\Phi}'$ and $\mathbf{\Psi}'$, which are results of manipulations of the functions given in $(\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Pi}, \mathbf{\Gamma})$. The method recursively invokes itself and also, occasionally, calls Method 3.11, which takes care of the actual elimination of variables. All methods are explained immediately after they have been presented.

Method 3.9
Input: IDIT $\mathcal{I}$, realization $(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Pi}, \boldsymbol{\Gamma})$, and end point in time t_e.
Output: Optimal strategy, $\boldsymbol{S}$.

1. For each utility, $u : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$, where $\boldsymbol{D}$ is some subset of $\boldsymbol{V}_{OD} \cup \boldsymbol{V}_{OC}$ and T is a time variable, in $\boldsymbol{\Psi}$, approximate the function $\psi : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$ as
$$\psi(\vec{d}) = \mathcal{T}_{u(\vec{d})\downarrow[0;t_e[},$$
for all $\vec{d}$ in $\mathbf{sp}(\boldsymbol{D})$. Replace u with ψ.

2. For each probability distribution with a time variable in its domain, $P(C|\boldsymbol{D}, T)$, where $\boldsymbol{D}$ is some subset of $\boldsymbol{V}_{OD} \cup \boldsymbol{V}_{OC}$, in $\boldsymbol{\Phi}$, construct the probability distribution $P^*(C|\boldsymbol{D}, T)$ as
$$P^*(c|\vec{d}) = \mathcal{T}_{P(c|\vec{d})\downarrow[0;t_e[},$$
for all c in $\mathbf{sp}(C)$ and $\vec{d}$ in $\mathbf{sp}(\boldsymbol{D})$. Then let $P^*(c'|\vec{d})$, for some arbitrary c' in $\mathbf{sp}(C)$, be given as
$$P^*(c'|\vec{d}) + \left(1 - \sum_{c \in \mathbf{sp}(C)} P^*(c|\vec{d})\right),$$
and replace $P(C|\boldsymbol{D}, T)$ with $P^*(C|\boldsymbol{D}, T)$.

3. For each density function, $f : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$, where $\boldsymbol{D}$ is some subset of $\boldsymbol{V}_{OD} \cup \boldsymbol{V}_{OC}$ and T is a time variable, in $\boldsymbol{\Pi}$, construct the function $\pi : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$ as
$$\pi(\vec{d}) = \left(\frac{\mathcal{T}_{f(\vec{d})\downarrow[0;t_e[}}{\int_0^{t_e} \mathcal{T}_{f(\vec{d})\downarrow[0;t_e[} dt}\right)^{\downarrow[0;t_e[},$$
for all $\vec{d}$ in $\boldsymbol{D}$.
Replace f with π.

4. Run Method 3.10 on $\mathcal{I}$, $(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Pi}, \boldsymbol{\Gamma})$, 0, and t_e. Denote the result $(\boldsymbol{\Phi}', \boldsymbol{\Psi}', \boldsymbol{S})$. The set $\boldsymbol{\Phi}'$ should contain a constant, 1, and $\boldsymbol{\Psi}$ should contain only a constant representing the maximum expected utility of $\mathcal{I}$.

5. Return $\boldsymbol{S}$.

The workings of Method 3.9 should be pretty straightforward to understand. In Step 1 each utility function is first projected down-to the interval $[0; t_e[$ in accordance with Assumption 2, and then a Taylor's series is constructed for each piece of the function. When a function is piecewise continuous it might be the case that the functions of adjacent intervals are essentially the same. In that case, we can speed up the solution method by joining these intervals into one. Examining whether two functions are the same, can be difficult when the initial functions are

given, but once they have been converted to polynomials, it can be done by $N+1$ comparisons. Although we, attempting to be clear and concise, do not write this operation specifically, it can be inserted after most of the operations described later in this section. We do not touch upon it again in this report.
In Step 2 each probability distribution with a time variable in its domain is approximated and projected down-to the interval $[0;t_e[$. In accordance with the discussion above, the resulting set of functions are then normalized to be a probability distribution.
Step 3 approximates the density functions, which are not allowed to be piecewise from the start. After a new function has been projected down-to the interval $[0;t_e[$ and approximated, it is normalized to be a density function. This completes the initialization steps of Method 3.9.
In Step 4 the method calls Method 3.10, which as mentioned is the main part of the solution method. Due to the recursive calls performed by that method, the result returned to Method 3.9 is a triple of sets. The first set contains probability distributions of ordinary chance variables, which have not been eliminated from the IDIT, the second contains utility functions over variables that have not been eliminated, and the last contains a set of policies that should constitute an optimal strategy. Obviously, the first set should contain a function over the empty set, i.e. a constant, which should be 1, and similarly the second set should contain a constant, indicating the maximum expected utility of the IDIT. We end the solving process by returning the optimal strategy in Step 5.

Method 3.10
Input: IDIT $\mathcal{I}$, realization $(\mathbf{\Phi},\mathbf{\Psi},\mathbf{\Pi},\mathbf{\Gamma})$, points in time t_s and t_e.
Output: Sets of probability distributions, $\mathbf{\Phi}'$, and utility functions, $\mathbf{\Psi}'$, over the variables in $\mathcal{I}$, which are not free, and an optimal strategy, $\boldsymbol{S}$, for free decisions in $\mathcal{I}$, given the variables that are not free.

1. Examine whether non-instantiated split variables are in $\mathcal{I}$. If so, let X denote the first of these. If not, let $\boldsymbol{S}$ be the empty set and skip to Step 4.

2. If X is not a time variable, skip to Step 3. Else,

   i Let $\boldsymbol{G}$ be the set of guards on arcs in $\mathcal{I}$ into the decisions in $\boldsymbol{ID}_{\mathcal{I}[X\mapsto t_s]}$. If X is furthermore in the domains of some restriction functions, $r_{D_1},\dots,r_{D_k}$, then let $\boldsymbol{R}$ be the set of boolean functions over X determining its impact on state spaces of $D_1,\dots,D_k$,
   $$\bigcup_{i=1}^{k}\{r_{D_i}(\vec{c},x)|\vec{c}\in \mathbf{sp}(\mathrm{dom}(r_{D_i}))\},$$
   otherwise, let $\boldsymbol{R}$ be $\varnothing$.

ii Partition the points in time from t_s to t_e into a set of intervals, $[t_s = t_1; t_2[, \ldots, [t_n; t_{n+1} = t_e[$, containing points in time having similar impact on guards in $\boldsymbol{G}$ and restriction functions in $\boldsymbol{R}$. That is, for any interval, $[t_i; t_{i+1}[$, any guard, g, in $\boldsymbol{G}$, any restriction function, r', in $\boldsymbol{R}$, and any two points, t_j and t_k, in $[t_i; t_{i+1}[$, we have that $g(t_j) = g(t_k)$ and $r'(t_j) = r'(t_k)$.

iii Let $\boldsymbol{F}_X$ be the set of free variables in $\mathcal{I}[X \mapsto t_s]$, and $\boldsymbol{\Phi}_{\boldsymbol{F}_X}$ the subset of $\boldsymbol{\Phi}$ containing probability distributions having some variable in $\boldsymbol{F}_X$ in their domain. Furthermore, let $\boldsymbol{\Psi}_{\boldsymbol{F}_X}$ and $\boldsymbol{\Pi}_{\boldsymbol{F}_X}$ be defined in similar ways. Let $\boldsymbol{\Phi}^*$ be the set $\boldsymbol{\Phi} \setminus \boldsymbol{\Phi}_{\boldsymbol{F}_X}$, and $\boldsymbol{\Psi}^*$ and $\boldsymbol{\Pi}^*$ be defined similarly.

iv For each interval, $[t_i; t_{i+1}[$, do the following

  a Construct the IDIT $\mathcal{I}[X \mapsto t_i]$, and the updated set of restriction functions arising from this instantiation, $\boldsymbol{\Gamma}_{X=t_i}$.

  b Recursively invoke Method 3.10 on $\mathcal{I}[X \mapsto t_i]$ and the realization $(\boldsymbol{\Phi}_{\boldsymbol{F}_X}, \boldsymbol{\Psi}_{\boldsymbol{F}_X}, \boldsymbol{\Pi}_{\boldsymbol{F}_X}, \boldsymbol{\Gamma}_{X=t_i})$ with the starting point in time being t_i and the ending point being t_e.

  Denote the resulting triples as $(\boldsymbol{\Phi}_1, \boldsymbol{\Psi}_1, \boldsymbol{S}_1), \ldots, (\boldsymbol{\Phi}_n, \boldsymbol{\Psi}_n, \boldsymbol{S}_n)$.

v For each utility ψ in each $\boldsymbol{\Psi}_i$, where X is not in $\text{dom}(\psi)$ and ψ is not in $\boldsymbol{\Psi}_j$ for all j in $\{1, \ldots, n\}$, condition ψ on the value of X being in $[t_i; t_{i+1}[$. That is, remove ψ from $\boldsymbol{\Psi}_i$ and replace it with the function $\psi' : \text{dom}(\psi) \times \mathbf{sp}(X) \to \mathbb{R}$, where, for each $\vec{z}$ in $\text{dom}(\psi)$, $\boldsymbol{I}_{\psi'(\vec{z})} = \{t_i, t_{i+1}\}$ and $\psi'(\vec{z})_1 = \psi(\vec{z})$.
  Then replace each utility ψ in each $\boldsymbol{\Psi}_i$, where X is in $\text{dom}(\psi)$, with $\psi^{\downarrow [t_i; t_{i+1}[}$.

vi Let

$$\boldsymbol{\Phi} = \boldsymbol{\Phi}^* \cup \bigcup_{i=1}^{n} \boldsymbol{\Phi}_i, \quad \boldsymbol{S} = \bigcup_{i=1}^{n} \boldsymbol{S}_i, \quad \text{and } \boldsymbol{\Psi} = \boldsymbol{\Psi}^* \cup \bigcup_{i=1}^{n} \boldsymbol{\Psi}_i.$$

3. i Let $\boldsymbol{F}_X$ be the set of free variables in $\mathcal{I}[X \mapsto x]$, where x is some state in the state space of X, and $\boldsymbol{\Phi}_{\boldsymbol{F}_X}$ be the subset of $\boldsymbol{\Phi}$ containing probability distributions having a variable in $\boldsymbol{F}_X$ in their domain. Furthermore, let $\boldsymbol{\Psi}_{\boldsymbol{F}_X}$ and $\boldsymbol{\Pi}_{\boldsymbol{F}_X}$ be defined in similar ways. Let $\boldsymbol{\Phi}^*$ be the set $\boldsymbol{\Phi} \setminus \boldsymbol{\Phi}_{\boldsymbol{F}_X}$, and $\boldsymbol{\Psi}^*$ and $\boldsymbol{\Pi}^*$ be defined similarly.

ii For each state, x, in $\mathbf{sp}(X)$, do the following:

  a Construct the IDIT $\mathcal{I}[X \mapsto x]$, and the updated set of realization functions arising from this instantiation, $\boldsymbol{\Gamma}_{X=x}$.

  b Recursively invoke Method 3.10 on $\mathcal{I}[X \mapsto x]$ and the realization $(\boldsymbol{\Phi}_{\boldsymbol{F}_X}, \boldsymbol{\Psi}_{\boldsymbol{F}_X}, \boldsymbol{\Pi}_{\boldsymbol{F}_X}, \boldsymbol{\Gamma}_{X=x})$ with the starting point in time being t_s and the ending point t_e.

  Denote the resulting triples as $(\boldsymbol{\Phi}_1, \boldsymbol{\Psi}_1, \boldsymbol{S}_1), \ldots, (\boldsymbol{\Phi}_n, \boldsymbol{\Psi}_n, \boldsymbol{S}_n)$.

iii For each utility ψ in $\mathbf{\Psi}_i$, where X is not in dom(ψ) and ψ is not in $\mathbf{\Psi}_j$ for all j in $\{1, \ldots, n\}$, condition ψ on X being x. That is, remove ψ from $\mathbf{\Psi}_i$ and replace it with the function $\psi' : \text{dom}(\psi) \times \mathbf{sp}(X) \to \mathbb{R}$, where $\psi'(\vec{z}, x') = \psi(\vec{z})$ if x' is x and 0 otherwise, for all $\vec{z}$ in dom(ψ) and x' in $\mathbf{sp}(X)$.

iv Let

$$\mathbf{\Phi} = \mathbf{\Phi}^* \cup \bigcup_{i=1}^{n} \mathbf{\Phi}_i, \quad \boldsymbol{S} = \bigcup_{i=1}^{n} \boldsymbol{S}_i, \quad \text{and } \mathbf{\Psi} = \mathbf{\Psi}^* \cup \bigcup_{i=1}^{n} \mathbf{\Psi}_i.$$

4. Eliminate all free variables from the functions in $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Pi}$ using Method 3.11 with some elimination order consistent with the inverse of $\prec'_{\mathcal{I}}$, the starting point t_s, and the ending point t_e. Denote the result $(\mathbf{\Phi}', \mathbf{\Psi}', \boldsymbol{S}')$.

5. Return $(\mathbf{\Phi}', \mathbf{\Psi}', \boldsymbol{S} \cup \boldsymbol{S}')$.

Method 3.10 basically branches into three cases depending upon the nature of the first split variable, X, in $\mathcal{I}$. If no X can be identified, it means that the ordering of variables and state spaces of decisions are the same for all decision scenarios in $\mathcal{I}$. In that case we can immediately proceed to Step 4 where all free variables in $\mathcal{I}$ are eliminated using Method 3.11.

If, on the other hand, a split variable X can be identified we must split the group of decision scenarios corresponding to $\mathcal{I}$ on X. This step is represented by Steps 2 and 3, corresponding to X being a time variable or not. The process in both steps are similar, but minor details are different due to X being either a time variable, and thus, continuous, or an ordinary decision or chance variable, and hence, discrete. We briefly note that X cannot be a wait decision, as these are not allowed to influence anything but their own end time, and consequently cannot be in a restriction function of any decision.

As the processes in Steps 2 and 3 are similar, we comment only on the one in Step 2, as this is the most complex one and contains the same problems as the one in Step 3. Initially, in parts i and ii, the state space of X is divided into intervals, according to its effect on $\mathcal{I}$. This is similar to the approach given in Method 2.10, and we therefore do not go into it in detail. What is worth noticing, though, is that this partitioning do not need to be of a specific granularity. That is, any partitioning, which fulfills the requirement on a similar effect on guards and restriction functions, will do. Furthermore, this approach forces us to assume that no restriction function is a function over more than one time variable, as the intervals cannot easily be determined otherwise.

Assumption 3
No decision variable can have two time variables in the domain of its restriction function.

Obviously, this assumption excludes the example IDIT given in Example 1 and similar IDITs from being solved.
In the case where X is not a time variable, we assume that it is discrete and split on its individual states instead of intervals. This is a reasonable assumption as the only non-time variables that are continuous is wait decisions, and as these are prohibited from restricting other decisions and do not appear in guards, they cannot be split variables.
Next, as the set of variables, which is in an instantiation of $\mathcal{I}$ on X, is the same for all values in the state space of X, we can select any value we like when determining this set. We chose, t_s, and identifies the set of free variables in $\mathcal{I}[X \rightarrow t_s]$. These are the subset of free variables in $\mathcal{I}$ that cannot be eliminated in this invocation of Method 3.10, since their ordering or state space is dependent on the value of X. Therefore, we construct the subproblems corresponding to each interval and recursively solve these.
The results of all subproblems should replace the original functions in the realization. However, some of the utility functions are not obtained from all of the recursive calls. These are therefore conditioned on X being in the corresponding interval. This happens in v. This step contains two implicit assumptions: First, that no time variable different from X is in the domains of utility functions absorbed from the subproblems, and second, that no two probability distributions over the same domain, but yielding different probabilities, are absorbed from the subproblems. In order to argue for the second of these assumption, it is sufficient to realize that:

- the only parts of Methods 3.10 and Methods 3.11 that produce new probability distributions or manipulate existing ones are the elimination procedures for ordinary chance variables, ordinary decisions, and time variables in Method 3.11
- these are commutative, in the sense that it does not matter which order variables are eliminated from them,
- the sets of variables that is eliminated in each subproblem is the same, and
- each subproblem is invoked on the same set of probability distributions.

That is, each invocation starts from the same situation, applies the same set of operations, which can be applied in any order without affecting the result, and consequently, ends up in the same situation.
In order to be sure that no utility function absorbed from a recursive call is defined over a time variable different from X, we need another assumption:

Assumption 4
For any two time variables, T and T', where $T \prec' T'$, and any node, X, that is a descendant of T', we have that T is d-separated from X, given the chance variables in $\mathbf{ch}(T)$.

This is the most limiting assumption we must introduce for the proposed solution method to work. It is not only needed at this point, but at several points in the elimination procedures in Method 3.11. To give the reader a better understanding of the implications of this assumption, we present a few examples of IDITs that do not fulfill it in Figure 3.10.
The IDIT in (a) does not fulfill Assumption 4 as the utility U is both a descendant of end(D_2), a child of end(D_1), and hence, d-connected to end(D_1) given end(D_2). In the slightly changed situation, modelled in the IDIT in (b), the utility U, which is still a descendant of end(D_2), is not a child of end(D_1). It is, however, still d-connected to end(D_1) through real(U). That both of these situations do not make a lot of sense can easily be argued for: As the realization time of U in both cases — end(D_1) and real(U), respectively — can be a point in time prior to deciding upon D_3, it goes against common sense to have the resulting choice of D_3 influencing U.
The situation modelled in the IDIT in (c), on the other hand, cannot be said to be senseless. We have a variable, C_3, which affects two chance variables dependent on time, C_1 and C_2. Several situation where such a setup is included can be thought of. For instance, C_3 could represent a global physical circumstance, such as humidity or temperature, and C_1 and C_2 could be observations of the same phenomenon, such as number of athletes still participating in an amateur marathon race, at two different points in time, end(D_1) and end(D_3). Sadly, as C_2 is d-connected to end(D_1) through C_3 and a descendant of end(D_2) as well, this IDIT does not fulfill Assumption 4.
Having described Assumption 4, we argue why it allows us to conclude that no time variable different from X is in the domains of utility functions absorbed from only some of the subproblems: Assume the opposite, namely that a utility, ψ, with some time variable T different from X in its domain, is returned from some of the subproblems only. First of all, we note that T must be prior to X in the temporal

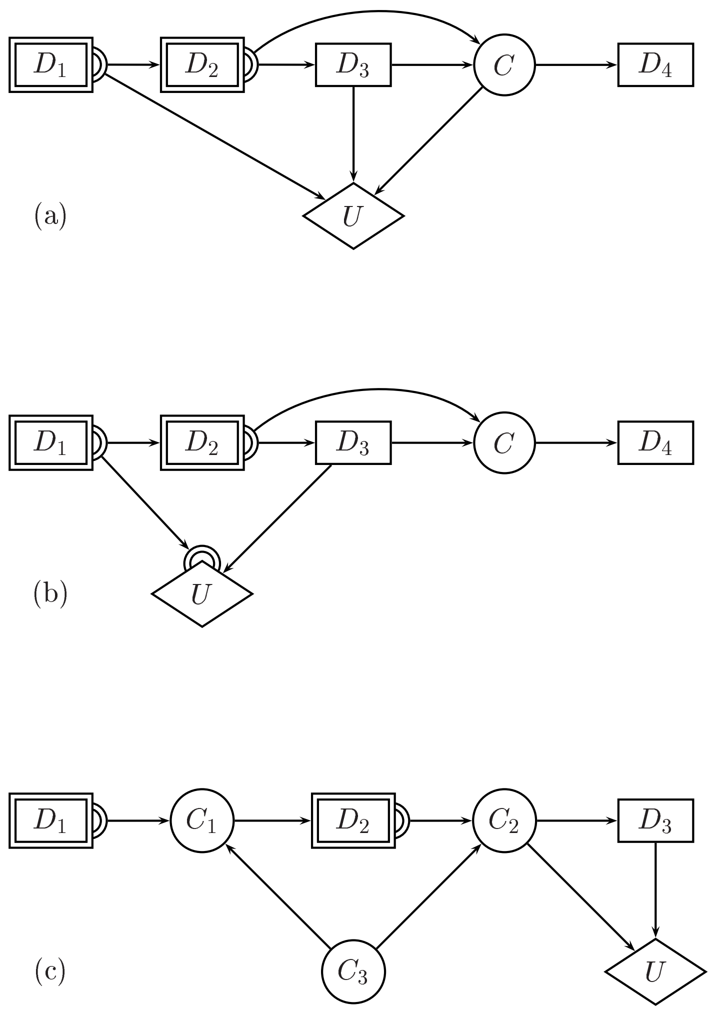

Figure 3.10: *IDITs such as these we do not attempt to solve.*

ordering, as it would have been eliminated in the subproblems otherwise. The reason why ψ is only returned from some subproblems, must be because of the asymmetric differences arising from X. That is, ψ must be the result of elimination of some decision D, where D's set of observed variables or D's state space is dependent on the value of X. If T is in the domain of ψ, then there must exist some node, Y, which T is d-connected to given the chance variables among its children, and which is a descendant of D. Furthermore, as D is eliminated in the subproblems, it must be following X in the temporal ordering and, thus, be a descendant of X. Consequently, Y is a descendant of X, thereby violating Assumption 4.

Returning to the description of Method 3.10, we reach Step 4, where all free variables in $\mathcal{I}$ are eliminated from the functions in $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Pi}$, with respect to the points in time t_s and t_e. The elimination of each variable is handled by Method 3.11, which needs an elimination order, consistent with the inverse of $\prec'$. That is, if $X \prec' Y$, then Y should be eliminated prior to X, whereas if both $X \not\prec' Y$ and $Y \not\prec' X$, then the elimination ordering of the two is without significance. We conjecture that any method for constructing these elimination orderings can be used, including the standard junction tree based method used for solving influence diagrams, as described in [Jensen, 2001]. For simplicity, we may simply assume that the exact elimination sequence is chosen at random.

After all free variables have been eliminated from the functions in $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Pi}$, we return what remains of them and the calculated optimal policies along with optimal policies calculated in any recursive steps to the calling method. If $\mathcal{I}$ is the original IDIT, the free variables encompass all variables in $\mathcal{I}$ and the resulting functions should, therefore, be only constants. It should be noted that no density functions are returned to the calling method. This is because

- no new density functions are produced by any of the elimination procedures in Method 3.11, and
- all density functions given to Method 3.10 as input are density functions for free time variables in $\mathcal{I}$, which are all eliminated in Step 4.

We proceed to presenting Method 3.11.

Method 3.11
Input: IDIT, $\mathcal{I}$, set of probability distributions, $\mathbf{\Phi}$, set of density functions, $\mathbf{\Pi}$, set of utility functions, $\mathbf{\Psi}$, points in time, t and t_e, and elimination order $(X_1, \ldots, X_n)$.
Output:

1. For each variable X in the ordering $(X_1, \ldots, X_n)$, eliminate X from $(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{S})$ using the appropriate elimination technique, from those described below.

2. Return the transformed sets $(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{S})$.

The basic structure of Method 3.11 is a loop where the variables to be eliminated are treated one after the other in the given order of elimination. Each elimination is performed by switching on the type of the variable to be eliminated and its parents in $\mathcal{I}$, and then applying the corresponding transformation on involved functions. As this structure is fairly basic, we focus on explaining the transformation in details, which constitute the larger part of the rest of the report.
All throughout the descriptions below, whenever we write "for a configuration $\vec{d}$ over the variables $\boldsymbol{D}$", we assume that the state space of each decision, D, has been updated according to the value of the restriction function, given the values of remaining variables in $\boldsymbol{D}$. This assumption are not needed for the solution to be found, but prevents calculation of function values that are irrelevant for a solution.

Case 1

The first case we consider is the elimination of a time or realization time variable, T, which is a child of another time variable, a wait decision, or both. This is by far the most complex case. Although this parent variable can be a wait decision, for the sake of clarity, we choose to denote it T' as if it was a time variable. If T is a child of both a wait decision, W, and a time variable, T'', we denote by T' the continuous variable $T'' + W$. In order to do calculations in the case described here we make an additional assumption, which consists of three nearly identical requirements:

Assumption 5
The density function, f, for a time variable, T, given another time variable, T', is, for all configurations, $\vec{d}$, of other variables in its domain, specified as a density function over the span in time from T' to T. That is, the density function, $f_{T-T'}$, for the continuous variable $T - T'$ given $\vec{d}$, is $f(t|t', \vec{d})$, for all real numbers, t and t'. Furthermore, $f_{T-T'}(t-t', \vec{d})$ is 0 for all points in time, t and t', where $t-t' \leq 0$.

The density function, f, for a time variable, T, given a wait decision, W, is, for all configurations, $\vec{d}$, of other variables in its domain, specified as a density function over the span in time from W to T. That is, the density function, f_{T-W}, *for the continuous variable* $T - W$ *given* $\vec{d}$*, is* $f(t|w, \vec{d})$*, for all real numbers,* t *and* w*. Furthermore,* $f_{T-T'}(t-w, \vec{d})$ *is* 0 *for all points in time,* t *and* t'*, where* $t-t' \leq 0$*.*

Likewise, the density function, f*, for a time variable,* T*, given another time variable,* T'*, and a wait decision,* W*, is, for all configurations,* $\vec{d}$*, of other variables in its domain, specified as a density function over the span in time from* $T'+W$ *to* T*. That is, the density function,* $f_{T-(T'+W)}$*, for the continuous variable* $T-(T'+W)$ *given* $\vec{d}$*, is* $f(t|t', w, \vec{d})$*, for all real numbers,* t*,* t'*, and* w*. Furthermore,* $f_{T-(T'+W)}(t-(t'+w), \vec{d})$ *is* 0 *for all real numbers,* t*,* t'*, and* w*, where* $t-(t'+w) \leq 0$*.*

Observing the requirement on time not regressing, the guiding lines given in this assumption represent a very natural way of specifying probability distributions for time variables given their predecessors. At least this author cannot come up with any counter examples.

When eliminating time variables, we utilize that all variables in the domain of their density function are prior to the time variable in the temporal ordering. This is not necessarily the case of realization time variables, as these have no specified ordering relative to their parent chance variables. However, by refining the temporal ordering to place a realization time variable after each of its parent variables, allow us to apply the same reasoning for these as time variables.

When T is to be eliminated the only functions in $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Pi}$ having T in their domain must be the density function for $T - T'$, $\pi_{T-T'}$, a set of utility functions, which combine additively into, ψ_T, and possibly some probability distributions, $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2, T)$. The reason why there cannot exist more density functions with T in its domain is that no new density functions are produced by any of the elimination procedures, and time variables following T in the temporal ordering must have been eliminated at this point.

As stated previously, the elimination ordering must respect the inverse of $\prec'$. That means that all ordinary chance variables, which are descendants of T must have been eliminated at this point. Therefore, if T is in the domain of a probability distribution, $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2, T)$, a variable, X, in $\boldsymbol{Z}_1$ is not a descendant of T. Furthermore, no variables among T's descendants can be considered instantiated at this point. Consequently, T must be d-separated from X given its parents, and we may simply replace $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2, T)$ in $\mathbf{\Phi}$ with $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2)$ equaling $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2, t_i)$ for some random t_i in $[t_s; t_e[$, such as t_s.

What remains is to replace $\pi_{T-T'} : \mathbf{sp}(\boldsymbol{D}_1) \times \mathbf{sp}(T-T') \rightarrow \mathbb{R}$ and $\psi : \mathbf{sp}(\boldsymbol{D}_2) \times \mathbf{sp}(T) \rightarrow \mathbb{R}$ with a new utility function, $\psi' : \mathbf{sp}(\boldsymbol{D}_1 \cup \boldsymbol{D}_2 = \boldsymbol{D}) \times \mathbf{sp}(T') \rightarrow \mathbb{R}$. Before we construct ψ' we explain why the sets $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ can be considered to be subsets of $\boldsymbol{V}_{OC} \cup \boldsymbol{V}_{OD}$. First, none of the elimination steps in Method 3.11 construct new density functions, and $\pi_{T-T'}$, therefore, can be defined over the time variables T and T' only. Second, none of the elimination steps constructs utility functions over two time variables and due to Assumption 4 such utilities cannot exist in the diagram from the start, so ψ cannot be defined over other time variables than T.

As when eliminating variables in influence diagrams, we need to construct, ψ', for each configuration, $\vec{d}$, of variables in $\boldsymbol{D}$. These functions should each be the expected value of ψ given T' and $\vec{d_2} = \vec{d}^{\downarrow \boldsymbol{D}_2}$. As each $\psi(\vec{d_2})$ might be piecewise continuous, we need to define each $\psi'(\vec{d})$ as a piecewise continuous function also. Since $\pi_{T-T'}$ is continuous over the interval $[0; t_e]$ we do not need to take this into account when identifying the intervals for each $\psi'(\vec{d})$, and we can, therefore, simply let $\boldsymbol{I}_{\psi'(\vec{d})}$ equal $\boldsymbol{I}_{\psi(\vec{d_2})}$ for each $\vec{d}$ in $\boldsymbol{D}$. Notice that if we had let density functions be piecewise continuous over the interval $[0; t_e[$, the partition $\boldsymbol{I}_{\psi'(\vec{d})}$ could not have been determined this way. This is because a density function is defined over the variable $T - T'$ rather than T itself. Therefore, the resulting partition of the resulting utility function for T', would be a function of T', leaving us with calculating an infinite number of partitions.

We let $\psi'(\vec{d})_i$ be 0, for each i in $\{0, \ldots, \boldsymbol{I}^{<t}_{\psi'(\vec{d})} - 1\}$. That is, the expected utility of any value of T' less than t, is not needed in future computations and is, therefore, simply set to 0. The remaining parts of $\psi'(\vec{d})$ is found as following: For each i in $\{\boldsymbol{I}^{<t}_{\psi'(\vec{d})}, \ldots, |\boldsymbol{I}_{\psi'(\vec{d})}|\}$ we let $\psi'(\vec{d})_i$ be defined as

$$\psi'(\vec{d})_i(t') = \int_{-\infty}^{\infty} \pi_{T-T'}(\vec{d_1})(t-t')\psi(\vec{d_2})(t)dt,$$

where $\vec{d_1}$ is $\vec{d}^{\downarrow \boldsymbol{D}_1}$, for all real numbers t'. By utilizing that $\pi_{T-T'}(\vec{d_1})(t-t')$ is 0 for all points in time, t, less than t', that t_e is an upper limit after which all utilities yield 0, that $\psi(\vec{d_2})$ is defined piecewise, and that all functions are polynomials of degree

N, we get

$$\begin{aligned}
\psi'(\vec{d})_i(t') &= \int_{-\infty}^{\infty} \pi_{T-T'}(\vec{d}_1)(t-t')\psi(\vec{d}_2)(t)dt \\
&= \int_{t'}^{t_e} \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j (t-t')^j \psi(\vec{d}_2)(t)dt \\
&= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \left(\int_{t'}^{t_{i+1}} (t-t')^j \sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l t^l dt + \right. \\
&+ \left. \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \int_{t_k}^{t_{k+1}} (t-t')^j \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m t^m dt \right) \\
&= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \left(\sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l \int_{t'}^{t_{i+1}} (t-t')^j t^l dt + \right. \\
&+ \left. \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m \int_{t_k}^{t_{k+1}} (t-t')^j t^m dt \right).
\end{aligned}$$

Using the Binomial Theorem[Edwards and Penney, 1998], we can replace $(t-t')^x$ with $\sum_{y=0}^{x}\binom{x}{y}(-1)^{x-y}t'^{x-y}t^y$, and we get

$$\begin{aligned}
\psi'(\vec{d})_i(t') &= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \left(\sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l \int_{t'}^{t_{i+1}} (t-t')^j t^l dt + \right. \\
&\quad \left. + \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m \int_{t_k}^{t_{k+1}} (t-t')^j t^m dt \right) \\
&= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \left(\sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l \int_{t'}^{t_{i+1}} \sum_{n=0}^{j} \binom{j}{n} (-1)^{j-n} t'^{j-n} t^j t^l dt \right. \\
&\quad \left. + \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m \int_{t_k}^{t_{k+1}} \sum_{n=0}^{j} \binom{j}{n} (-1)^{j-n} t'^{j-n} t^j t^m dt \right) \\
&= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \sum_{n=0}^{j} \binom{j}{n} (-1)^{j-n} \left(\sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l t'^{j-n} \int_{t'}^{t_{i+1}} t^{j+l} dt \right. \\
&\quad \left. + \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m t'^{j-n} \int_{t_k}^{t_{k+1}} t^{j+m} dt \right) \\
&= \sum_{j=0}^{N} C[\pi_{T-T'}(\vec{d}_1)]_j \sum_{n=0}^{j} \binom{j}{n} (-1)^{j-n} \left(\sum_{l=0}^{N} C[\psi(\vec{d}_2)_i]_l \frac{t_{i+1}^{j+l+1} - t'^{j+l+1}}{j+l+1} t'^{j-n} \right. \\
&\quad \left. + \sum_{k=i+1}^{|\boldsymbol{I}_{\psi'(\vec{d})}|-1} \sum_{m=0}^{N} C[\psi(\vec{d}_2)_k]_m \frac{t_{k+1}^{j+m+1} - t_k^{j+m+1}}{j+m+1} t'^{j-n} \right)
\end{aligned} \tag{3.6, 3.7}$$

As this is a polynomial of degree $2N$, we need to approximate it by a polynomial of a degree N, before removing ψ and $\pi_{T_T'}$ from $\mathbf{\Psi}$ and $\mathbf{\Pi}$, respectively, and inserting ψ' in $\mathbf{\Psi}$.

It is worth noticing that if the numbers $\binom{x}{y}$ have been evaluated before hand, the evaluation time of the expression in (3.7) is $O(N^3|\boldsymbol{I}|^1)$, where $|\boldsymbol{I}|$ is the maximum number of intervals a utility function is split into due to initial specification and split variables. Considering that the expression only yield one of the needed polynomials, ψ_i, we end up with a total evaluation time of $O(N^3|\boldsymbol{I}|^2)$, prior to approximation down-to a polynomial of degree N. Thus, for sufficiently large approximations of degree n, the complexity of this operation benfits from a division of the domain into intervals and approximations to a lesser degree over each of them.

Case 2

The case where we are eliminating a time variable, T, which have no time variable nor wait decision as parent, is roughly similar to Case 2. The only difference is that the utility function, ψ', resulting from eliminating T is not a function over another time variable or wait decision. We show how to derive ψ', given a density function, $\pi_T : \mathbf{sp}(\boldsymbol{D}_1) \times \mathbf{sp}(T) \rightarrow \mathbb{R}$, and a utility function $\psi : \mathbf{sp}(\boldsymbol{D}_2) \times \mathbf{sp}(T) \rightarrow \mathbb{R}$. For the same reasons as when eliminating a time variable with another time variable as parent, $\boldsymbol{D} = \boldsymbol{D}_1 \cup \boldsymbol{D}_2$ must be a subset of $\boldsymbol{V}_{OC} \cup \boldsymbol{V}_{OD}$, and we derive ψ' for each configuration, $\vec{d}$, over $\boldsymbol{D}$:

$$\psi'(\vec{d}) = \int_{-\infty}^{\infty} \pi_T(\vec{d}_1)(t)\psi(\vec{d}_2)(t)dt,$$

where $\vec{d}_1$ denotes $\vec{d}^{\downarrow \boldsymbol{D}_1}$ and $\vec{d}_2$ denotes $\vec{d}^{\downarrow \boldsymbol{D}_2}$. Using the same tricks as in the derivations above, we get

$$\begin{aligned}
\psi'(\vec{d}) &= \sum_{i=0}^{N} C[\pi_T(\vec{d}_1)]_i \sum_{j=1}^{|\boldsymbol{I}_{\psi(\vec{d}_2)}|-1} \int_{t_j}^{t_{j+1}} \sum_{k=0}^{N} C[\psi(\vec{d}_2)_j]_k t^{i+k} dt \\
&= \sum_{i=0}^{N} C[\pi_T(\vec{d}_1)]_i \sum_{j=1}^{|\boldsymbol{I}_{\psi(\vec{d}_2)}|-1} \sum_{k=0}^{N} C[\psi(\vec{d}_2)_j]_k \int_{t_j}^{t_{j+1}} t^{i+k} dt \\
&= \sum_{i=0}^{N} C[\pi_T(\vec{d}_1)]_i \sum_{j=1}^{|\boldsymbol{I}_{\psi(\vec{d}_2)}|-1} \sum_{k=0}^{N} C[\psi(\vec{d}_2)_j]_k \frac{t_{j+1}^{i+k+1} - t_j^{i+k+1}}{i+k+1}.
\end{aligned}$$

Similar to the situation above, we need to approximate this result before inserting it in $\boldsymbol{\Psi}$. The evaluation of this thus takes time $O(N^2|\boldsymbol{I}|)$.

Case 3

When eliminating a wait decision, W, with a time variable, T, as parent, we need only consider one function, viz. the utility $\psi : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(W+T) \rightarrow \mathbb{R}$ in $\boldsymbol{\Psi}$. This is because W only occurred in one function from the start, $f_{T'-(W+T)}$, for some time variable, T'. When T' was eliminated only a utility over $W+T$ was produced while $f_{T'-(W+T)}$ was removed. For the same reasons as given for elimination of time variables, $\boldsymbol{D}$ must be a subset of $\boldsymbol{V}_{OC} \cup \boldsymbol{V}_{OD}$.
As we eliminate W from ψ we need to identify a strategy that, given a configuration

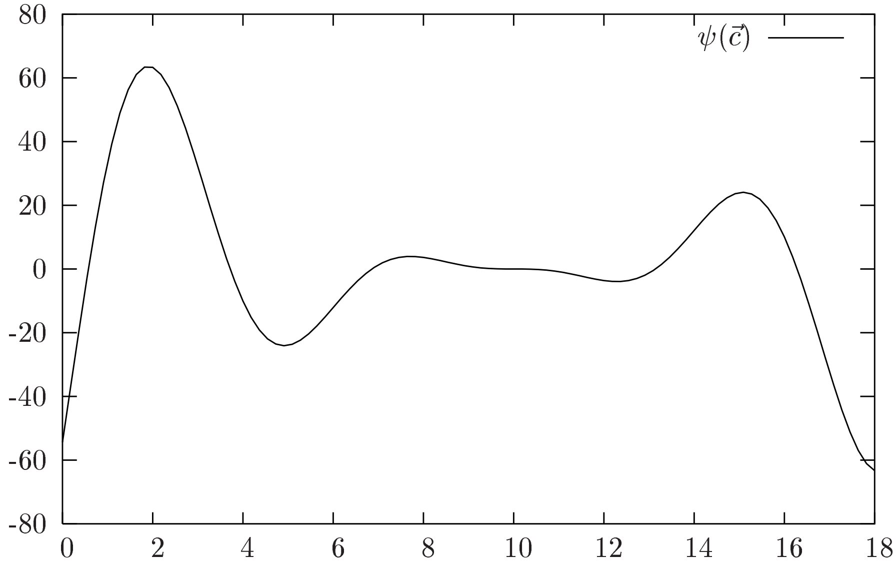

Figure 3.11: *An example utility function for a wait decision.*

$\vec{d}$ over the variables in $\boldsymbol{D}$, and a point in time, t, represented by T, yields the choice from $[t; t_e]$ that maximizes ψ. If we study the example utility function, $\psi(\vec{d})$, plotted in Figure 3.11, we can see how these choices must be described:

- If t is less than 2, where $\psi(\vec{d})$ is at a global maximum, the best advice is to wait until time 2. In other words, the optimal choice is to wait for $2 - t$ time units.
- If t is more than 2, but still less than 3, the best advice is not to wait. That is, the optimal choice is to wait for 0 time units.
- In the time span from 3 to 15, the best advice is again to wait. The optimal choice is thus to wait for $15 - t$ time units.
- In the remaining time of the interval from 0 to 18, the optimal choice is again to wait for 0 time units.

Thus, an optimal policy, δ_W, would be defined as following:

$$\delta_W(\vec{d})(t) = \begin{cases} 2 - t & \text{if } t < 2 \\ 0 & \text{if } 2 \leq t < 3 \\ 15 - t & \text{if } 3 \leq t < 15 \\ 0 & \text{if } 15 \leq t < 18 \end{cases}$$

We arrive at this conclusion through a simple procedure. Once we have established that the points 0, 2, 3, 15, and 18 are the places on the real line the policy should change, we can apply a simple set of rules to each interval between them to determine the policy. However, some utility functions might be piecewise continuous, and we have to take that into account. Furthermore, we need a method of finding extrema of the utility function, which reduces to finding roots in its derivative. We find roots of a function, f, by the application of Newton's method, which given an initial guess of a root, x_1, calculates a new root candidate, x_2, using the formula

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)},$$

where f' is the first derivative of f. The process continues iteratively until the difference $x_{n+1} - x_n$ is smaller than some fixed threshold value. Newton's method can under some circumstances fail to locate a root, even though such one exists, and in that case human intervention might be necessary, or another approximation method may be used. For more information on Newton's method, see [Edwards and Penney, 1998].

The iterative procedure for locating an optimal policy for a given configuration, $\vec{d}$, is as following: Initially, we differentiate each piece of the utility function $\psi(\vec{d})$ with regards to t. We locate roots of the resulting functions in their respective intervals. This is done using Newton's method. We denote the roots $r_1, \ldots, r_n$. Next we denote $\boldsymbol{P} = \boldsymbol{I}_{\psi(\vec{d})} \cup \{r_1, \ldots, r_n\}$ as the set of points of interest, and set

$$r = \arg\max_{t \in \boldsymbol{P}} \psi(\vec{d})(t).$$

Then we set the policy for all points in time, t', prior to r to $r - t'$.

In the iterative step we identify which of the points larger than r that gives rise to the highest value of $\psi(\vec{d})$. If several of these points exists we choose the minimum one. Let this be r'. Then we construct the function $f(t) = \psi(\vec{d})(t) - \psi(\vec{d})(r')$ and find its root, r^*, in the interval $]r; r'[$, if such a thing exists.

If no root exists, $\psi(\vec{d})$ is either larger than $\psi(\vec{d})(r')$ over the interval $[r; r'[$ or less than or equal to $\psi(\vec{d})(r')$ over $[r; r'[$. In the first case we set the policy for this interval to 0. In the second case we set it to $r' - t$.

If a root exists, we are in a situation such as the one presented in the example above, and we can set the policy for $[r; r^*[$ to 0 and the policy for $[r^*; r'[$ to $r' - t$.

Next we set r' to be r and iterate. When we run out of candidates for maximums in $\boldsymbol{P}$ we stop the iteration.

This method does not take into account restrictions on the state space of W. The alterations needed for this would, depending on how simple these restrictions are, imply a more thorough examination of the utility function. We chose not to focus on this here and simply assume:

Assumption 6
No arcs into wait decisions may be dashed.

Having this assumption as a basic part of the representation language of IDITs is actually not that reckless. Whenever we are in a situation where we are told that we "cannot wait for that long", or that we "need to wait for at least" some specific amount of time, the implicit understanding of this is "or else...". In other words, a restriction on a wait decision could be modellable as an sudden decrease or increase in some utility connected to the time variable following the wait decision. Therefore, we do not see Assumption 6 as a limitation on the number of decision problems that can be solved.

The remaining bit of work is to construct a new utility function, ψ', over $\mathbf{sp}(\{T\} \cup \boldsymbol{D})$. For each configuration, $\vec{d}$, over $\boldsymbol{D}$, where $\delta_W(\vec{d})$ is the optimal policy just found, we let $\psi'(\vec{d})$ be given as following: For all intervals where $\delta_W(\vec{d})$ is not given as 0, but as $k - t$, for some k in $[t; t_e[$, we let $\psi'(\vec{d})$ be the function defined as $f(t) = \psi(\vec{d})(r)$. For all intervals where $\delta_W(\vec{d})$ is 0, we let $\psi'(\vec{d})$ be $\psi(\vec{d})$.

Case 4

The case where a wait decision, W, with no time variable as parent is to be eliminated is quite simple. As for wait decisions with time variables as parents, we can assume that only a utility function, $\psi : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(W) \to \mathbb{R}$, has W in its domain. We need to find an optimal policy for W — a process which, for each configuration $\vec{d}$ over the variables in $\boldsymbol{D}$, encompass locating the value, $m_{\vec{d}}$, of W corresponding to the global maximum of $\psi(\vec{d})$ over $[0; t_e[$. This process was described as part of the explanation of how to eliminate wait decisions with time variables as parents, so we do not repeat it here.

Once these global maxima have been identified, we can replace ψ with an utility, $\psi' : \mathbf{sp}(\boldsymbol{D}) \to \mathbb{R}$, defined as

$$\psi'(\vec{d}) = \psi(\vec{d}, m_{\vec{d}}),$$

for all $\vec{d}$ in $\mathbf{sp}(\boldsymbol{D})$. Furthermore, we construct the optimal policy $\delta_W : \mathbf{sp}(\boldsymbol{D}) \to [0; t_e[$, as

$$\delta_W(\vec{d}) = m_{\vec{d}},$$

for all $\vec{d}$ in $\mathbf{sp}(\boldsymbol{D})$.

Case 5

When an ordinary decision, D, is eliminated we need to manipulate a set of probability distributions, $\boldsymbol{\Phi}_D$, having D in their domain, and a set of utilities, $\boldsymbol{\Psi}_D$, with a similar property. No density functions can have D in their domain at this point, because such a density function would be defined over a time variable following D in the temporal ordering, which should have been eliminated at this point.
Like for time variables, we start by removing D from any probability distribution $P(\boldsymbol{Z}_1|\boldsymbol{Z}_2, D)$, as D must be d-separated from any variable in $\boldsymbol{Z}_1$. Following this, we branch into two cases: First case, is when no time variable is in the domain of any of the utility functions in $\boldsymbol{\Psi}_D$. The second case is when only one time variable is in the domains of the utility functions in $\boldsymbol{\Psi}_D$. We can never be in the case that two time variables, T and T', are both in the domains of functions in $\boldsymbol{\Psi}_D$, and that the sum of these functions is not constant over either T or T'. To see this, we need a conjecture:

Conjecture 1
Let T be a time variables in an IDIT, d-connected to some variable X given its the chance variables amongst its children. Then T is d-connected to any node that is a child of X.

We assume without loss of generality that $T \prec' T'$, and that ψ_T is an utility function with T in its domain. This means that there is some node, X, which is a descendant of D and, according to Conjecture 1, d-connected to T given the chance variables amongst its children. Furthermore, as both T and T' is in the domain of a utility in $\boldsymbol{\Psi}_D$, they cannot have been eliminated at this point, which indicates that D is following both of them in the temporal ordering. This in turn tells us that D is a descendant of T', that X is a descendant of T', and Assumption 4 is, therefore, violated.
We consider first the case in which no time variable is in the domain of utility functions in $\boldsymbol{\Psi}_D$. This is similar to the procedure used for eliminating decision variables in influence diagrams and is, therefore, not presented in great detail. We let $\psi : \mathbf{sp}(\boldsymbol{D} \cup \{D\}) \to \mathbb{R}$ be the sum of utilities in $\boldsymbol{\Psi}_D$. Then for each configuration, $\vec{d}$,

of variables in $\boldsymbol{D}$, we let the policy for D be

$$\delta_D(\vec{d}) = \arg \max_{d \in \mathbf{sp}(D)} \psi(\vec{d}, d),$$

and the maximum expected utility, $\psi' : \mathbf{sp}(\boldsymbol{D}) \to \mathbb{R}$, be defined as

$$\psi'(\vec{d}) = \psi(\vec{d}, \delta_D(\vec{d})).$$

Finally, we replace the utilities in $\boldsymbol{\Psi}_D$ in $\boldsymbol{\Psi}$ with ψ' and store δ_D in $\boldsymbol{S}$.
If a time variable, T, is in the domains of the functions in $\boldsymbol{\Psi}_D$, we sum all utilities in $\boldsymbol{\Psi}_D$ into one, $\psi : \mathbf{sp}(\boldsymbol{D} \cup \{D\}) \times \mathbf{sp}(T) \to \mathbb{R}$. We construct the optimal policy, δ_D, for D as following: For each configuration, $\vec{d}$, of $\boldsymbol{D}$, we construct the functions $\psi(\vec{d}, d_i) - \psi(\vec{d}, d_j)$, for each pair of distinct states, d_i and d_j, in $r_D(\vec{d}^{\downarrow \mathrm{dom}(r_D)})$, where r_D is the restriction function for D in $\boldsymbol{\Gamma}$. We then use Newton's method on these functions and locate their roots. These points, along with discontinuities, $\boldsymbol{I}_{\psi(\vec{d})}$, in $\psi(\vec{d})$, are the points in time where our policy may change. We denote them $\boldsymbol{P}_{\psi(\vec{d})}$. As identification of these points is the main purpose of constructing the $\psi(\vec{d}, d_i) - \psi(\vec{d}, d_j)$-functions, we may chose to only construct one of the functions $\psi(\vec{d}, d_i) - \psi(\vec{d}, d_j)$ and $\psi(\vec{d}, d_j) - \psi(\vec{d}, d_i)$ for each pair of states, d_i and d_j.
Finally, for each interval $[t_i; t_{i+1}[$ generated by $\boldsymbol{P}_{\psi(\vec{d})}$ we let

$$\delta_D(\vec{d})(t) = \arg \max_{d \in r_D(\vec{d})} \psi(\vec{d}, d) \left(\frac{t_{i+1} - t_i}{2} \right),$$

for all t in $[t_i; t_{i+1}[$.
The utility function, $\psi' : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$, which is to replace the functions in $\boldsymbol{\Psi}_D$ in $\boldsymbol{\Psi}$, we derive for each configuration, $\vec{d}$, of $\boldsymbol{D}$ as following: First, we let $\boldsymbol{I}_{\psi'(\vec{d})}$ be $\boldsymbol{P}_{\psi(\vec{d})}$. Then for each interval $[t_i; t_{i+1}[$ generated by $\boldsymbol{I}_{\psi'(\vec{d})}$ we let

$$\psi'(\vec{d}) = \psi(\vec{d}, \delta_D(\vec{d}, t_i)).$$

Case 6

Elimination of an ordinary chance variable, C, involves manipulation of functions in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. No density function, π, in $\boldsymbol{\Pi}$, for a time variable, T, can have C in its domain, as that would imply that $C \prec' T$, and hence, that T should have been eliminated at this point. We call the sets of functions with C in their domain $\boldsymbol{\Phi}_C$

and $\mathbf{\Psi}_C$, respectively.
Each function in $\mathbf{\Psi}_C$ can have only one time variable in its domain, as that is the case from the start, and no elimination procedure produces utility functions over two time variables. At this point we divide the description into three cases, depending on whether no, one, or more time variables are in the domains of functions in $\mathbf{\Phi}_C$.

No Time Variables in Domains of Functions in $\mathbf{\Phi}_C$

First, we replace the functions in $\mathbf{\Phi}_C$ in $\mathbf{\Phi}$ with the function, $\phi' : \mathbf{sp}(\boldsymbol{D}_1) \rightarrow [0;1]$, where $\boldsymbol{D}_1$ is the set of variables in domains of functions $\mathbf{\Phi}_C$ except C, defined as

$$\phi'(\vec{d}_1) = \sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}_1^{\downarrow \mathrm{dom}(\phi)}),$$

for each configuration, $\vec{d}_1$, over the variables in $\boldsymbol{D}_1$. Second, we replace each function, ψ, in $\mathbf{\Psi}_C$ in $\mathbf{\Psi}$ with the expected value of ψ, ψ'. If ψ does not have a time variable in its domain, we let ψ' be $\psi' : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \mathrm{dom}(\psi) \setminus \{C\}) \rightarrow \mathbb{R}$, defined as

$$\psi'(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$.
If ψ has a time variable, T, in its domain, we let ψ' be $\psi' : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \mathrm{dom}(\psi) \setminus \{C, T\}) \times \mathbf{sp}(T) \rightarrow \mathbb{R}$, defined as

$$\psi'(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$. The main difference between this expression and the one before is that the resulting $\psi'(\vec{d})$'s are polynomials.
We have divided up the sum of utilities, and calculated each expected utility individually, which is not the standard solution technique for influence diagrams. In these the utilities in $\mathbf{\Psi}_C$ are additively combined and then the expected value of this combination is constructed. The two approaches can easily be shown to yield the same result, though. To see why we have chosen this approach, study the IDIT in Figure 3.12. If we add up the resulting utilities when eliminating C, we end up with a utility over two time variables, which the rest of the solution method depends on

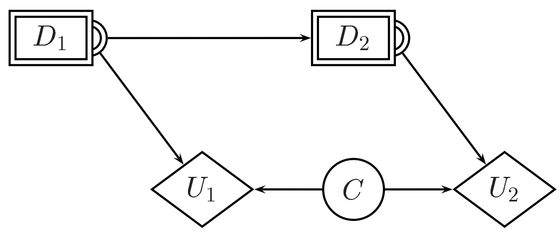

Figure 3.12: *IDITs where we have to be careful not to sum the local utility functions.*

never happens.

One Time Variable in Domains of Functions in $\mathbf{\Phi}_C$

When there is one or more time variables, $\boldsymbol{T}$, in the domain of the functions in $\mathbf{\Phi}_C$, we need a different approach and some additional results. We first assume that only one time variable, T, is in $\boldsymbol{T}$:

We replace the functions in $\mathbf{\Phi}_C$ in $\mathbf{\Phi}$ with the function, $\phi' : \mathbf{sp}(\boldsymbol{D}_1) \times \mathbf{sp}(T) \to [0;1]$, where $\boldsymbol{D}_1$ is the set of variables in domains of functions $\mathbf{\Phi}_C$ except C and T, defined as

$$\phi'(\vec{d}_1) = \sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}_1^{\downarrow \mathrm{dom}(\phi)}),$$

for each configuration, $\vec{d}_1$, over the variables in $\boldsymbol{D}_1$. This is a polynomial, which can be of degree $|\mathbf{\Phi}_C|N$, since each probability distribution might be a polynomial of degree N. As we work with polynomials of a fixed degree only, we approximate ϕ^* by a Taylor's series. That is, we replace the functions in $\mathbf{\Phi}_C$ in $\mathbf{\Phi}$ with the function $\phi' : \mathbf{sp}(\boldsymbol{D}_1) \times \mathbf{sp}(T) \to [0;1]$ defined as

$$\phi'(\vec{d}_1) = \mathcal{T}_{\phi^*(\vec{d}_1)},$$

for each configuration, $\vec{d}_1$, over the variables in $\boldsymbol{D}_1$.

Second, we replace each function, ψ, in $\mathbf{\Psi}_C$ in $\mathbf{\Psi}$ with the expected value of ψ, ψ'. If ψ does not have a time variable in its domain, we let ψ^* be $\psi^* : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \mathrm{dom}(\psi) \setminus \{C\}) \times \mathbf{sp}(T) \to \mathbb{R}$, defined as

$$\psi^*(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \mathbf{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$. Each $\psi^*(\vec{d})$ is not a polynomial, so we need to approximate it. Consequently, we replace ψ in $\boldsymbol{\Psi}$ with the function $\psi' : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$ defined as

$$\psi'(\vec{d}) = \mathcal{T}_{\psi^*(\vec{d})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$.

If ψ has a time variable, T', in its domain, and T' is the same variable as T, then we let ψ^* be $\psi^* : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \mathrm{dom}(\psi) \setminus \{C\}) \times \mathbf{sp}(T) \to \mathbb{R}$, defined as

$$\psi^*(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$. As above, each $\psi^*(\vec{d})$ is not a polynomial, so we replace ψ in $\boldsymbol{\Psi}$ with $\psi' : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T) \to \mathbb{R}$ defined as

$$\psi'(\vec{d}) = \mathcal{T}_{\psi^*(\vec{d})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$.

If ψ has a time variable, T', different from T, in its domain, we cannot apply the above operations directly, as we need to be sure we do not construct a utility with two time variables in its domain. We refer to the discussion in the next paragraph on why a utility cannot be non-constant over more than one time variable, T^*, and simply state that the resulting utility must be constant over at least one of the variables. We, therefore, let ψ^* be $\psi^* : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \mathrm{dom}(\psi) \setminus \{C\}) \times \mathbf{sp}(T^*) \to \mathbb{R}$, defined as

$$\psi^*(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$. As twice before, each $\psi^*(\vec{d})$ is not necessarily a polynomial, so we replace ψ in $\boldsymbol{\Psi}$ with $\psi' : \mathbf{sp}(\boldsymbol{D}) \times \mathbf{sp}(T') \to \mathbb{R}$ defined as

$$\psi'(\vec{d}) = \mathcal{T}_{\psi^*(\vec{d})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$.

More than one Time Variable in the Domains of Functions in $\boldsymbol{\Phi}_C$

When there is more time variables, $\boldsymbol{T}$, in the domains of the functions in $\boldsymbol{\Phi}_C$, we replace the functions in $\boldsymbol{\Phi}_C$ in $\boldsymbol{\Phi}$ with the function, $\phi' : \mathbf{sp}(\boldsymbol{D}_1) \times \mathbf{sp}(\boldsymbol{T}) \to [0;1]$, where $\boldsymbol{D}_1$ is the set of variables in domains of functions $\boldsymbol{\Phi}_C$ except C and those in $\boldsymbol{T}$, defined as

$$\phi'(\vec{d}_1) = \sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}_1^{\downarrow \mathrm{dom}(\phi)}),$$

for each configuration, $\vec{d}_1$, over the variables in $\boldsymbol{D}_1$. This is not a polynomial, but a sum over a product of polynomials. We denote it as a *compound expression*. We do not evaluate it to a polynomial at this point but simply store ϕ' in $\boldsymbol{\Phi}$ instead of the functions in $\boldsymbol{\Phi}_C$. Any applicable encoding scheme, such as a list of ordinary chance variables, $\boldsymbol{C}$, followed by a list of polynomials, $\boldsymbol{PL}$, can be used to represent a function such as

$$\sum_{X \in \boldsymbol{C}} \prod_{p \in \boldsymbol{PL}} p,$$

and we do not make any assumptions on this representation. We must, however, take care that this unevaluated function do not interfere with the workings of the other parts of the solution method. The only parts of the solution method that manipulates probability distributions in ways other than dropping variables from their domains, is the two cases described above. As both of these are conditioned on there not being two time variables in domains of functions in $\boldsymbol{\Phi}_C$, we can be sure that these probability distributions do not get handled by anything other than this part of the solution method.

We then examine each utility, $\psi : \mathbf{sp}(D_2) \times \mathbf{sp}(C) \times \mathbf{sp}(T) \to \mathbb{R}$, in $\boldsymbol{\Psi}_C$. When we construct the expected utility, $\psi^* : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \boldsymbol{D}_2) \times \mathbf{sp}(\boldsymbol{T} \cup \{T\}) \to \mathbb{R}$, as

$$\psi^*(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$, we end up with a non-polynomial over several time variables, $\boldsymbol{T} \cup \{T\}$. However, it will always be the case that ψ^* is constant over all time variables, except possibly for one. To see this, let $\boldsymbol{T}_{\not c}$ denote the set of time variables in $\boldsymbol{T} \cup \{T\}$ for which ψ^* is not constant. Furthermore, let T_m denote the variable in $\boldsymbol{T}_{\not c}$ farthest in the temporal ordering. As T_m has not been eliminated it is clearly the case that $T_m \prec'_{\mathcal{I}} C$. ψ^* is not constant over T_m and therefore it must have some node, X, amongst the descendants of C, as descendant. Then let T' be some other time variable in $\boldsymbol{T}_{\phi}$, which would have to be prior to T_m in the temporal ordering. As ψ^* is not constant over T' and all chance variables that are children of T' is known when T_m is known, we conclude that T' is d-connected to C given the chance variables amongst its children, and, according to Conjecture 1, thus to X. This is a violation of Assumption 4, and T' can therefore not exist. Hence, ψ^*, varies only over one variable.

We then let $\psi' : \mathbf{sp}(\boldsymbol{D} = \boldsymbol{D}_1 \cup \boldsymbol{D}_2) \times \mathbf{sp}(T_m) \rightarrow \mathbb{R}$, be given as

$$\psi^*(\vec{d}) = \frac{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)}) \psi(c, \vec{d}^{\downarrow \mathrm{dom}(\psi)})}{\sum_{c \in \mathbf{sp}(C)} \prod_{\phi \in \boldsymbol{\Phi}_C} \phi(c, \vec{d}^{\downarrow \mathrm{dom}(\phi)})},$$

for each configuration, $\vec{d}$, over the variables in $\boldsymbol{D}$. If this is not a polynomial of degree N, we approximate it and replace ψ in $\boldsymbol{\Psi}$ with the approximation. Otherwise, we replace it with ψ'.

The only open question remaining, is regarding the point in the method where a compound expression, ϕ, is removed from $\boldsymbol{\Psi}$: Whenever a time variable is eliminated, and consequently removed from the domain of ϕ, we check if there are still two time variables in the domain of ϕ, and if not, we evaluate ϕ, and store the result in $\boldsymbol{\Phi}$.

Structural Correctness of the Solution Method

In the paragraphs above, we have given several arguments as to why the elimination procedures and short cuts we have applied are sound. We still need to argue why the overall structure of the solution method produces an optimal strategy for a given IDIT $\mathcal{I}$. As a formal proof of this would be rather elaborate and would not differ much from the one given in [Nielsen and Jensen, 2000], we chose only to present a sketch.

To prove that the structure of the method constructs an optimal strategy, given that the elimination procedures are correct, we apply a conversion of IDITs to decision trees, and utilize that the averaging-out-and-folding-back algorithm is known to produce an optimal strategy for decision problems modelled as decision trees [Jensen, 2001].

We need to argue that our solution method can construct a split tree where each variable in $\mathcal{I}$ is treated as a split variable: Each ordinary decision or chance variable in $\mathcal{I}$, which are not in the domain of a restriction function, can be used as a split variable. This can be seen from Step 3 in Method 3.10, which does not require of the variables it splits on, that the resulting subproblems are of a different structure. Hence, we may split on all ordinary decision and chance variables in this step, if we so desire.

As already mentioned above, when splitting on a time variable, the partitioning of the numbers in the interval $[t_s; t_e[$, need not be of a specific granularity, as long as the requirement on similarity of decision scenarios in the resulting subproblems is fulfilled. Therefore, we can discretize each time variable to any given level of precision and still apply out method. Similarly, by discretizing wait decisions we can let Step 3 in Method 3.10 split on the states of these as well.

Thus, by discretizing continuous variables, to an arbitrarily fine level of granularity, we end up with the general structure of the averaging-out-and-folding-back algorithm.

Future Work

Clearly, this solution method suffers from some flaws, all due to the assumptions introduced. The subset of IDITs we can solve are limited, most noticeable because of Assumption 4, which, among other things, prohibits an unobserved variable to influence time dependent variables not dependent on the same time variable. As already mentioned this excludes a great deal of decision scenarios, and a topic of future research would be to alter the parts of the solution method that depends on it. Furthermore, an area where improvements are needed, is in the handling of restriction functions. The method, as presented, does not allow wait decisions to be restricted, and ordinary decisions cannot be restricted by more than one time variable.

Apart from completing the solution method, it would be interesting to analyze the complexity of the method in terms of N, t_e, $|\boldsymbol{I}|$, and the number of nodes in an IDIT. The result of such an analysis could perhaps be used for deciding an optimal value of N and optimal number of approximation intervals for each utility and probability distribution, given an IDIT and a value for t_e.

Apart from completing the solution method, a topic of interest is implementation of a modelling and solving tool for IDITs. This tool could warn the user when an IDIT violating the assumptions is constructed.

Chapter 4

Conclusion

As this report forms the documentation of a period of study as well as a means of communicating the result, this conclusion consists of two parts. First, some concluding remarks on the scientific status and value of the results given in the report, and second, a brief account on the knowledge that has been obtained by this author in the process.

The starting point for this report is a representation language for representing decision problems involving quantitative aspects of time. This representation language suffers from some serious flaws and some minor quirks, which we corrected in Chapter 2. The most damaging flaw was the lack of a clear temporal ordering of elements in IDITs. Through an analysis we have highlighted the flaws in the existing ordering, and from this analysis, we have created a new temporal order operator, which takes asymmetry arising from quantitative time into account. Building on this work, we have managed to construct a definition of what IDITs that make sense, that is, which can be considered welldefined. A method for checking IDITs for being welldefined has also been constructed.

The temporal ordering operator is inspired by the ordering operator used in asymmetric influence diagrams [Nielsen and Jensen, 2000], but takes on a quite different form, due to guarded arcs into decisions being inherited by subsequent decisions. Similarly, the concept of instantiation, used in other representation languages, has been adapted to cater for this as well.

In addition to these results, we have also presented a solution method which solves a subset of IDITs. Even considering the limitations on the IDITs which can be solved, the solution method is interesting as it avoids discretizing continuous variables and does so without utilizing sampling. Continuous variables have so far not been integrated into influence diagrams with unequivocal success. The problem of specifying policies for decisions over continuous variables has so far eluded solution. In our case, we have exploited the restrictions and nature of time, and hence, have solved it in this specific case.

Seen from a personal learning perspective, this project has been rich on challenges. The problems connected to identifying a temporal ordering relation have mostly been dealt with using reflection and pondering while studying orderings of influence diagrams in detail. Before deciding upon an approach for solving IDITs, several other approaches, including sampling and discretization, was studied to a point where a choice could be made on a solid foundation. As most of the methods for dealing with continuous variables that exist are based heavily on properties of Gaussian distributions, few ideas from these sources have been applicable, whereas the structure of solutions to asymmetric influence diagrams have been an inspiration to the structure of the solution method presented in this report.

Bibliography

[Apostol, 1974] Apostol, T. M. (1974). *Mathematical Analysis*. Addison-Wesley, 2nd edition.

[Broe and Jeppesen, 2003] Broe, M. and Jeppesen, R. (2003). Influence diagrams involving time, a framework for decision problems involving time.

[Broe et al., 2003] Broe, M., Jeppesen, R., and Nielsen, S. H. (2003). Representing decision problems involving time.

[Charnes and Shenoy, 2003] Charnes, J. M. and Shenoy, P. P. (2003). A forward monte carlo method for solving influence diagrams using local computation. Working paper.

[Demirer and Shenoy, 2001] Demirer, R. and Shenoy, P. P. (2001). Sequential valuation networks: A new graphical technique for asymmetric decision problems. In Benferhat, S. and Besnard, P., editors, *Lecture Series in Artificial Intelligence*, number 2143, pages 252–265. Springer-Verlag.

[Edwards and Penney, 1998] Edwards, C. H. and Penney, D. E. (1998). *Calculus with analytic geometry*. Prentice Hall, 5th edition.

[Howard and Matheson, 1984] Howard, R. A. and Matheson, J. E. (1984). Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis*, pages 721–762.

[Jensen, 2001] Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

[Kozlov and Koller, 1997] Kozlov, A. V. and Koller, D. (1997). Nonuniform dynamic discretization in hybrid networks. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 314–325.

[Lauritzen and Nilsson, 2001] Lauritzen, S. L. and Nilsson, D. (2001). Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251.

[Lerner et al., 2001] Lerner, U., Segal, E., and Koller, D. (2001). Exact inference in networks with discrete children of continuous parents. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, pages 319–328.

[Madsen and Jensen, 2003] Madsen, A. L. and Jensen, F. (2003). Mixed influence diagrams. In *Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*.

[Nielsen and Jensen, 2000] Nielsen, T. D. and Jensen, F. V. (2000). Representing and solving asymmetric decision problems.

[Nilsson and Jensen, 1999] Nilsson, D. and Jensen, F. V. (1999). Probabilities of future decisions. Proceedings from the International Conference on Informational Processing and Management of Uncertainty in knowledge-based Systems (IPMU), pages 1454–1461.

[Nist, 2003] Nist (2003). Nist/sematech e-handbook of statistical methods. `http://www.itl.nist.gov/div898/handbook/eda/section3/eda3666.htm`.

[Poland and Shachter, 1993] Poland, W. B. and Shachter, R. D. (1993). Mixtures of gaussians and minimum relative entropy techniques for modeling continuous uncertainties. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*.

[Shachter and Kenley, 1989] Shachter, R. D. and Kenley, C. R. (1989). Gaussian influence diagrams. *Management Science*, 35(5):527–550.

Notation

$\mathrm{init}(D)$	the time variable that represents the initiation time of the decision D	p6
$\mathrm{end}(D)$	the time variable that represents the end time of the decision D	p6
$\mathbf{sp}(X)$	the state space of the variable X	p7
$\boldsymbol{S}_1 \times \boldsymbol{S}_2$	the Cartesian product of the sets $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$	p7
$X = x$	the knowledge that the variable X is in the state x	p7
$\mathrm{dom}(f)$	the domain of the function f	p11
$\boldsymbol{W}^{\mathcal{I}}$	the set of all nodes in the IDIT $\mathcal{I}$	p14
$\boldsymbol{L}^{\mathcal{I}}$	the set of all labels in the IDIT $\mathcal{I}$	p14
$\boldsymbol{E}^{\mathcal{I}}$	the set of all edges in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_C^{\mathcal{I}}$	the set of all chance variables in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_D^{\mathcal{I}}$	the set of all decisions in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_T^{\mathcal{I}}$	the set of all time variables in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_W^{\mathcal{I}}$	the set of all wait decisions in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_U^{\mathcal{I}}$	the set of all local utility functions in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}^{\mathcal{I}}$	the set of all variables in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_{OC}^{\mathcal{I}}$	the set of all ordinary chance variables in the IDIT $\mathcal{I}$	p14
$\boldsymbol{V}_{OD}^{\mathcal{I}}$	the set of all ordinary decisions in the IDIT $\mathcal{I}$	p14
$\boldsymbol{E}_s^{\mathcal{I}}$	the set of all solid edges in the IDIT $\mathcal{I}$	p15
$\boldsymbol{E}_d^{\mathcal{I}}$	the set of all dashed edges in the IDIT $\mathcal{I}$	p15
$\mathbf{pa}(X)$	the set of parent variables for the node X	p15
$\mathbf{ch}(X)$	the set of child nodes for the node X	p15
$\mathbf{pa}_d(X)$	the set of parent variables connected with dashed edges for the node X	p15
$\prec$	the ordering relation used in [Broe et al., 2003]	p15
$\boldsymbol{\Phi}^{\mathcal{I}}$	the set of probability distributions in a realization for the IDIT $\mathcal{I}$	p18
$\boldsymbol{\Psi}^{\mathcal{I}}$	the set of local utility functions in a realization for the IDIT $\mathcal{I}$	p18
$\boldsymbol{\Pi}^{\mathcal{I}}$	the set of density functions in a realization for the IDIT $\mathcal{I}$	p18
$\boldsymbol{\Gamma}^{\mathcal{I}}$	the set of restriction functions in a realization for the IDIT $\mathcal{I}$	p18
$\hookrightarrow$	a partial function	p18
$\mathrm{real}(U)$	the realization time variable of the utility U	p20
$\boldsymbol{V}_R^{\mathcal{I}}$	the set of all realization time variables in the IDIT $\mathcal{I}$	p21
$\prec'_{\mathcal{I}}$	the temporal ordering relation of elements in the IDIT $\mathcal{I}$	p32
$\mathcal{I}[X \mapsto x]$	the instantiation of the IDIT $\mathcal{I}$ in which X is known to be x	p37

$\boldsymbol{ID}_{\mathcal{I}}$	the set of decisions in an IDIT $\mathcal{I}$ initiating at the point in time the decision problem modelled by $\mathcal{I}$ starts	p39
$\boldsymbol{O}^{\mathcal{P}}$	The set of possible orderings of variables in the decision problem $\mathcal{P}$	p49
$o^{\mathcal{P}}(\vec{z})$	the ordering of variables in the decision problem, $\mathcal{P}$, when the variables are instantiated as $\vec{z}$	p49
$\boldsymbol{S}^{\mathcal{P}}_{\lhd}$	the policies in the strategy $\boldsymbol{S}$, which are valid under the ordering $\lhd$	p49
$\boldsymbol{\Delta}_{\mathcal{P}}$	the set of strategies for a decision problem $\mathcal{P}$	p49
$P_{\delta_D}(D\|\boldsymbol{P})$	the δ_D-induced probability distribution for the policy δ_D	p49
$f(\boldsymbol{S})^{\downarrow \boldsymbol{S}'}$	the real-valued function over $\boldsymbol{S}'$ obtained from the funtion f by summing and/or integrating over all variables in $\boldsymbol{S} \setminus \boldsymbol{S}'$	p50
$P_{\boldsymbol{S}^{\mathcal{P}}_{\lhd}}(\boldsymbol{V}^{\mathcal{P}})$	the $\boldsymbol{S}$-induced probability distribution for the strategy $\boldsymbol{S}$ for the decision problem $\mathcal{P}$	p50
$\boldsymbol{P}_{\mathcal{I},D,\vec{t}}$	the set of past time variables for the decision D in the IDIT $\mathcal{I}[\boldsymbol{V}^{\mathcal{I}}_{T} \mapsto \vec{t}]$	p52
$\boldsymbol{S}_{o^{\mathcal{I}}(\vec{t})}$	$\boldsymbol{S}^{\mathcal{I}}_{\lhd}$, where $\lhd$ is some ordering consistent with $\prec'_{\mathcal{I}[\boldsymbol{V}^{\mathcal{I}}_{T} \mapsto \vec{t}]}$	p52
$\vec{x}^{\downarrow \boldsymbol{S}}$	the configuration over the variables in $\boldsymbol{S}$ obtained from $\vec{x}$ by dropping coordinates not corresponding to a variable in $\boldsymbol{S}$	p53
$\delta_D^{\boldsymbol{S},\mathcal{I}}$	the policy for D in $\boldsymbol{S}$ under the ordering $\prec'_{\mathcal{I}}$	p60
$\boldsymbol{I}^{<x}$	the number of elements in the partition $\boldsymbol{I}$ less than or equal to x	p71
$\boldsymbol{I}_f$	the partition the function f is defined over	p71
f_i	the function defined over the i'th interval generated by $\boldsymbol{I}_f$	p71
$f^{\downarrow [a;b[}$	the function that takes on the value of f on points in $[a;b[$ and 0 everywhere else	p73
$C[f]_i$	the coefficient corresponding to x^i in the polynomial f	p73
N	the degree of Taylor's series in the solution method	p74
$\mathcal{T}_f$	the piecewise approximation of the function f	p75
$\mathcal{T}_f([a;b],c)$	the Taylor's series of f on $[a;b]$ about c	p75
t_e	a point in time after which all utilities in an IDIT should yield 0	p77
$\|\boldsymbol{I}\|$	the maximum number of intervals a utility function is split into due to initial specification and split variables	p91

Index

absorption, 67

blown-up version, 17

chance variable, 7
compressed form, 17

decision, 7
decision involving time, 9
decision scenario, 39
decision variable, 7
derived variable, 56
domain, 11

eliminating variable, 65, 66
end time, 6
expected utility, 51
extended no-forgetting assumption, 11, 32

first variable, 36
free variable, 65

guard, 10
guarded variable, 10

IDIT, 5, 14, 27, 29
informational arc, 10
initiation time, 6
instant decision, 9
instantiated variable, 7
instantiation, 37
intermediate variable, 36

last variable, 36
legal policy, 53
legal strategy, 54
local utility, 7

marginalizing variable out, 65, 66

no-delay assumption, 6
no-forgetting assumption, 10
non-intervening choice, 6
normalizing, 74

optimal policy, 51
optimal strategy, 50
ordinary chance variable, 7
ordinary decision variable, 7

partition, 71
piecewise continuous function, 71
policy, 48
policy-induced probability distribution, 50
projection, 50, 73

qualitative aspect, 14
quantitative aspect, 14

realization, 14, 29
realization time, 20
realization time variable, 20
required policy, 52
restriction function, 11

solution method, 51
solution to decision problem, 51
solving a decision problem, 51
split tree, 42
strategy, 49
strategy under ordering, 49
strategy-induced probability distribution, 50

Taylor's series, 73
temporal arc, 17
temporal ordering, 32, 34
temporally allowable instantiation, 38
time variable, 7
total utility, 7

utility, 7

valid policy, 49

wait decision, 6, 7
welldefined, 35, 39

Appendix A

Summary

This report deal with a representation language for decision problems involving quantitative aspects of time called influence diagrams involving time, or simply IDITs.
For some time there has existed a number of frameworks for representing and solving decision problems, including influence diagrams, valuation networks, and decision trees. None of them cope very well with quantitative measures of time, which was uncovered in [Broe et al., 2003]. Consequently, a new framework was needed. [Broe et al., 2003] suggests a representation language, called IDITs, which is supposed to be a compact and unambiguous language compatible with influence diagrams, in the sense that an IDIT of a decision problem involving no aspects of quantitative time should be interpretable as an influence diagram with no modification. [Broe et al., 2003] neglects to turn the representation language into a full framework, meaning that both a set of unambiguous semantics and a solution method is lacking. In this report both of these missing results are developed.
Chapter 2 contain a description of IDITs. In short, these are directed acyclic graphs whose nodes represent chance and decision variables and local utilities. Arcs in the graph represent either probabilistic dependencies, informational constraints, or functional dependencies. This far IDITs resemble influence diagrams. However, IDITs allow for a subset of the chance variables to represent points in time where decisions end, and thereby to be continuous. Furthermore, decisions can be continuous if they denote decisions on lengths of waiting periods encountered during the process described by the decision problem. Asymmetry arising from quantitative time are included in the diagram by the means of guarded informational arcs and restriction functions for decisions.

We enhance IDITs by furthermore allowing utilities to depend on points in time not necessarily representing an end time of a decision in the decision problem. Additionally, we allow the ordering of decisions, which do not span a period of time, to vary according to the time previous decisions have ended, and modify the rules for inheriting guards in the diagram, to better reflect the nature of time dependent observation. A temporal ordering relation which takes into account the asymmetry of IDITs is then presented, and a definition of welldefinedness is derived from this relation. We furthermore construct a method that checks whether an IDIT is welldefined.

At this point we have completed IDITs as a representation language and can, thus, in Chapter 3, construct a solution method on a solid foundation. Our approach to constructing a solution method, takes outset in an introduction to solutions to decision problems in general, and is then outlined through an elaborate example before being presented in full. The structure of our solution method follows the structure of solution methods for solving asymmetric decision problems in [Nielsen and Jensen, 2000] and [Demirer and Shenoy, 2001], but the details are different. We chose to approximate continuous functions in IDITs by Taylor's series and use algebraic manipulations of these in order to eliminate variables from the IDIT. Specifically, we are, due to asymmetry, required to cater for piecewise continuous functions, which further allow us to approximate continuous utility functions with greater precision using the same resources.

The resulting solution method is not universally applicable, as it builds on a series of assumptions on the nature of the given IDIT. It can be argued for that most of these assumptions are fulfilled by the vast majority of IDITs. However, one of them is a real limitations. Future research should seek to eliminate the need for this assumption.