Logic of Probabilistic Arguments

Subrata Das[1]

Abstract

We present a logic for reasoning with probabilistic arguments to help decision making under uncertainty. The syntax of the logic is essentially modal propositional, and arguments of decision makers are expressed as sentences of the logic, with associated supports drawn from a probability dictionary. To aggregate a set of arguments for and against some decision options, we construct a Bayesian belief network based on the argument set without requiring any additional information from the decision-maker. Evidence converted from the underlying knowledge of the decision maker is posted at the relevant nodes of the belief network to compute probability distributions, and hence rankings, among the decision options. Decision-making based on such rankings of decision options is therefore guaranteed to be consistent with probability theory. We develop possible world semantics of the logic, and establish soundness and completeness results. We illustrate the proposed decision-making framework in the context of a concrete example.

1 Introduction

Human decision-making can be regarded as a complex information processing activity, which, according to (Rasmussen, 1983), is divided into three broad categories, corresponding to activities at three different levels of complexity. At the lowest level is skill-based sensorimotor behavior, representing the most automated, largely unconscious level of skilled performance such as deciding to brake upon seeing a car ahead. At the next level is rule-based behavior, exemplified by simple procedural skills for well-practiced, simple tasks such as inferring the condition of a game-playing field based on the current weather. Knowledge-based behavior represents the most complex cognitive processing, used to solve difficult and sometimes unfamiliar problems, for making decisions that require dealing with various factors and uncertain data. Examples of this type of processing include determining the status of a game (i.e. a sporting event), given that there is transport disruption. Our focus here is to develop an argumentation framework to support human decision making at the knowledge base level by providing suggestions as to alternative courses of action, and help determine the most suitable. Human decision makers often weigh the available alternatives and select the most promising one based on the associated pros and cons. The proposed argumentation framework, similar to the one developed in (Das et al. 1997; Das and Grecu, 2000; Fox and Das, 2000), therefore naturally supports human decision-makers by augmenting and complementing their own cognitive capabilities.

Two important requirements must be met if we are to develop a practical and useful decision support system: the system must be declarative and robust. The declarative nature of the system ensures a human readable representation of knowledge and human-like reasoning with knowledge. Robustness of the system ensures its ability to cope with uncertain or missing data in situations where the required knowledge is unavailable in the underlying knowledge base. We plan to make our proposed framework declarative via the use of a high-level logical syntax for representing arguments, including probabilities to represent their strengths. The robustness is assured via representations that allow computations over a range of values, and the use of Bayesian belief network technology (Pearl, 1988) to support combining diverse evidence of arguments for and against decision alternatives. The belief network formalism supports probabilistic reasoning over the causal and evidential relations combining knowledge from decision makers and the current set of beliefs, so that the system can derive probability estimates for adopting particular decision options.

To summarize our framework, we use the syntax of modal propositional logic for representing arguments, and include probabilities to represent their strengths. For the purpose of aggregation of arguments, we automatically transform a set of arguments for and against some decision options into a belief network. The generated belief network then forms the basis for computing aggregated evidence for the decision options according to the strengths of the arguments. This hybrid approach has the following advantages:

- Arguments are expressed in a human readable syntax of modal propositional logic, along with a probability dictionary for expressing their strengths.
- The possible world semantics of the logic that we develop is intuitive to decision makers, as decision options simply correspond to various possibilities mapped to possible worlds.
- Aggregation is carried out on a belief network that is automatically constructed out of available arguments, and no additional knowledge needs to be acquired.

The rest of the paper is organized as follows: Section 2 presents an argumentation-based decision-making framework. Section 3 presents the underlying logic of arguments in the proposed framework. Section 4 presents an approach to argument aggregation via Bayesian belief networks. Section 5 presents a concrete example to illustrate the syntax and semantics of the logic and the argumentation and aggregation process. Each of Section 3 and Section 4 can be read independently of the other, but the example in Section 5 requires understanding of both the logic and the aggregation process. Throughout the paper we use the single example of the status of a ball game, which is scheduled to occur sometime today. Proof of theorems and propositions stated in the paper have been omitted due to space limitations. The proofs can be found in (Fox and Das, 2000).

2 Decision Making via Argumentation

This section presents the non-temporal version of the argumentation-based decision-making framework that was

[1] CHARLES RIVER ANALYTICS, Cambridge, MA 02138, USA

developed in (Fox and Das, 2000; Das et al. 1997), but focusing only on probabilistic arguments. We first provide a brief historical background of argumentation. Then we provide a concrete example to illustrate the use of argumentation, followed by the formal 'domino' model of argumentation and a knowledge representation language for expressing decision constructs and beliefs and knowledge in the model.

2.1 Brief Background in Argumentation

Toulmin in his book (Toulmin, 1956) discussed how difficult it is to cast everyday practical arguments into classical deductive form. He claimed that arguments needed to be analyzed using a richer format than the simple if-then form of classical logic. He characterizes practical argumentation by means of the scheme in Figure 1.

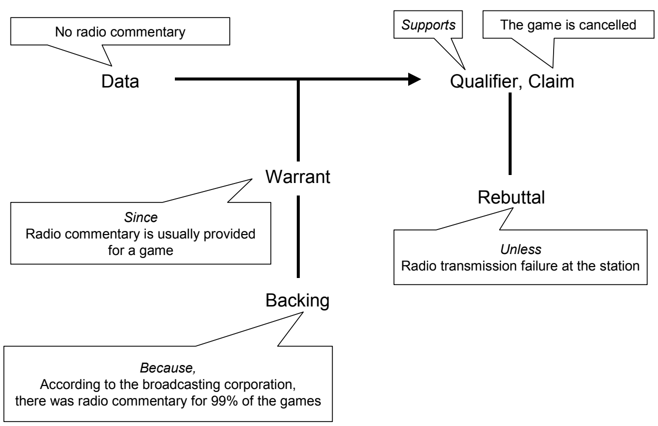

Figure 1: Toulmin's model of argumentation

As shown in Figure 1, Toulmin's model decomposes an argument into a number of constituent elements: 1) Claim: the point a decision maker is trying to make; 2) Data: the facts about a situation provided to support the claim; 3) Warrant: statements indicating general ways of arguing; 4) Backing: generalizations providing explicit support for an argument; 5) Qualifier: phrases showing the confidence an argument confers on a claim; 6) Rebuttal: acknowledges exceptions or limitations to the argument. To illustrate, consider an argument claiming that the game, which was supposed to be held today, has been cancelled. The facts or beliefs (that is, data) on which this claim is made are that there is no radio commentary for the game in question. General principles or rules, such as "radio commentary is usually provided for a game", warrant the argument, based on statistical research published by the broadcasting corporation, which is the backing. Since the argument is not conclusive we insert the qualifier "supports" in front of the claim, and note the possibility that the conclusion may be rebutted on other grounds, such as failure of radio transmission of the commentary.

Our approach is to transform Toulmin's work to a more formal setting, much the same way as in (Fox et al, 1992). We too deal with the concepts of warrant and rebuttal, but as very simple prepositional arguments for and against. We do not deal with first-order sentences that are more suitable for representing backings in Toulmin's model. We introduce the use of a single qualifier called 'support'.

2.2 Example Decision Making Process

We explain here the argumentation based decision-making framework in (Fox and Das, 2000), continuing with our ball-game example as shown in Figure 2.

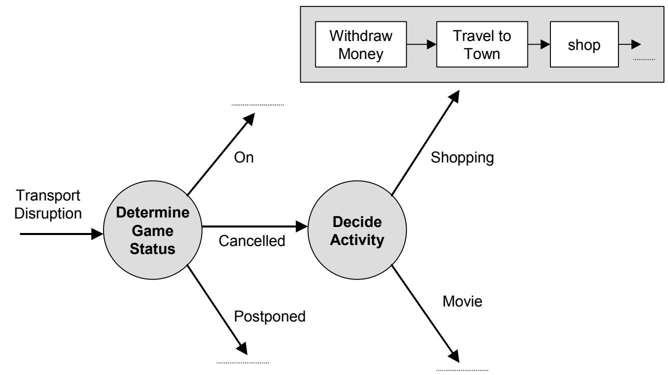

Figure 2: Decision-making flow

The process starts when the decision maker observes transport disruption on the way to catch a public transport (e.g. a bus) to go to town for the game. The newly discovered transport status then becomes the decision maker's belief. Given that the decision maker "believes" that there is transport disruption, it raises a "goal" of finding the status of the game. It then infers from its common sense knowledge that there are three possible or "candidate" states of the game, On, Cancelled, and Postponed, and so constructs arguments for and against these alternatives. These arguments use other beliefs of his, based on observations such as the weather and radio commentary. In this case the balance of "argument" is in favor of the game being cancelled, and this conclusion is added into the decision maker's database of beliefs.

Given this new belief regarding the cancelled status of the game, a new goal is raised, i.e. to plan for alternative activities. As in determining the status of the game, here there are two options for alternative activities, shopping and going to a movie, and the decision maker once again constructs arguments for the alternatives, taking into account transport, cost, etc., and recommends going shopping as the most preferred alternative activity on the basis of the arguments. The adoption of a shopping "plan" leads to an appropriate schedule of "actions" involved in shopping, such as withdrawing money, traveling to town, going to stores, etc. The effects of these actions are recorded in the decision maker's database, which may lead to further goals, and so on.

2.3 The Domino Model

Figure 3, the 'domino' model, captures graphically the decision-making framework, where the outer chain of arrows in the figure represents the above example decision-making process. Within our proposed framework, a decision schema has several component parts: an evoking situation, a goal, one or more candidates, and one or more commitment rules.

A situation describes, as a boolean expression on the database of beliefs, the situation or event which initiates decision making. For example, a belief that an abnormality (e.g. transport disruption) is present may lead to a choice between alternative possible causes/effects of it.

A goal is raised as soon as the evoking situation occurs. In particular, the belief that an abnormality is present may raise the goal of determining its cause or effects. For example, if transport is disrupted then one of its possible effects is the cancellation of the game, so therefore the goal is to determine game status. On the other hand, if there is no radio commentary then a goal is to determine the status of the game, as its cancellation causes no radio commentary. Typically, a goal is represented by a property that the decision maker tries to bring about.

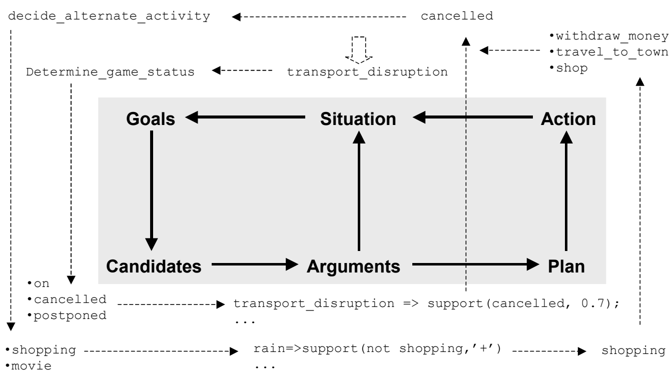

Figure 3: Domino process view of the example

Candidates are a set of alternative decision options, such as {on, cancelled, postponed}. In principle the set of candidates may be defined extensionally (as a set of propositions) or intentionally (by rules), but we only consider the former case here.

Arguments are modal-propositional rules that define the arguments that are appropriate for choosing between candidates for the decision. Argument schemas are typically concerned with evidence when the decision involves competing hypotheses (beliefs), and with preferences and values when the decision is concerned with actions or plans.

Commitment rules define the conditions under which the decision may be recommended, or taken autonomously, by the decision maker. It may include logical and/or numerical conditions on the argument and belief databases.

The following section represents a decision schema and its components as described above into a decision construct.

2.4 Decision Constructs

The concept of the domino decision scheme and its components is captured in a high-level declarative syntax. Figure 4 gives the decision construct representing the 'Determine Game Status' decision circle in Figure 2. All decisions have an evoking situation which, if the decision maker believes it to be true, raises the corresponding goal. The three possible paths from the decision circle go to the following three alternative pathways: on, cancelled, and postponed. These candidates are represented explicitly in the decision construct. The arguments and commitments within a decision construct are also represented directly.

The decimal number in an argument represents the probabilistic measure of support given by the argument to the decision candidate. The basic idea is that an argument is a reason to believe something or a reason to act in some way and an argument schema is a rule for generating such reasons during decision making. The more arguments there are for a candidate belief or action, then the more a decision maker is justified in committing to it. The aggregation function can be a simple "weighing up of pros and cons" (netsupport), but it represents a family of more or less sophisticated functions by which we may assess the merit of alternative candidates based on the arguments about them.

```
decision:: game_status
    situation
        transport_disruption
    goal
        determine_game_status
    candidates
        on;
        cancelled;
        postponed
    arguments
        transport_disruption => support(cancelled, 0.7);
        not radio_commentary => support(not on, 0.9);
        not rain => support(on, 0.95);
        bad_economy => support(not cancelled, 0.6);
        bad_economy & free_slot => support(postponed, 0.7);
    commits
        netsupport(X, U) & netsupport(Y, V) &
        netsupport(Z, W) & U > V & U > W => add(X).
```

Figure 4: Example decision construct

In general, an argument schema is like an ordinary inference rule with

support(<candidate>, <sign>)

as its consequent, where <sign> is drawn from a set called a dictionary. The <sign> represents, loosely, the confidence that the inference confers on the candidate. The dictionary may be strictly quantitative (e.g. the numbers in the [0,1] interval) or qualitative (e.g. the symbols {+, -} or {pro, con}). Here we are dealing with probabilistic arguments and <sign> is drawn from the probability dictionary [0,1]. An example argument from the decision construct in Figure 4 is

```
transport_disruption =>
        support(cancelled,0.7)
```

where <candidate> is 'cancelled'. Informally, the argument states that if there is transport disruption then there is 70% chance that the game will be cancelled. The rest of the arguments of the decision construct provide support for and against the decision options based on the evidence of radio commentary, weather, and hosting club's economic condition, and availability of free slots for rescheduling the game. A knowledge base for the decision maker consists of a set of definitions of this and other kinds of tasks.

A decision maker considers the decision game_status in Figure 4 for activation when the belief transport_disruption is added to the database. When the decision maker detects this, it checks whether any of the candidates has already been committed. If not, the decision will be activated and the goal determine_game_status is raised; otherwise no action is taken. While the goal is raised, further information about the situation (e.g. the weather) can be examined to determine whether the premises of any argument schemas are instantiated.

A commitment rule is like an ordinary rule with one of

```
add(<property>)
schedule(<plan>)
```

as its consequent. The former adds a new belief to the knowledge base and the latter causes an action to be scheduled as follows (see Figure 5):

```
decision:: alternative_activity
    situation
        cancelled
    goal
        decide_alternative_ activity
    candidates
        shopping;
        movie
    arguments
        rain => support(no shopping, 0.8);
        …
    commits
        … .
```

Figure 5: Example decision construct

See (Fox and Das, 2000) for information on how to deal with a scheduled plan that is committed. When a decision is in progress then, as additional arguments become valid, the decision's commitment rules are evaluated to determine whether it is justified to select a candidate. A commitment rule will often make use of an aggregation function such as 'netsupport' but this is not mandatory. The netsupport function evaluates collections of arguments for and against any candidate to yield an overall assessment of confidence and establish an ordering over the set of candidates; this ordering may be based on qualitative criteria or on quantitative assessment of the strength of the arguments. This function has the form:

```
netsupport(<candidate>, <support>)
```

In section 4, we implement the 'netsupport' function using an algorithm for evidence propagation in belief networks (Pearl, 1988; Jensen, 1996).

3 Logic of Arguments

The section presents the underlying logic of the argumentation-based decision-making framework, L_{Arg}, as described above, its possible world semantics, and the soundness and completeness results.

3.1 The Syntax

Suppose *P* is the set of all propositions, representing properties and actions, and includes the special property symbol $\top$ (true). Note that the logic does not distinguish between properties and actions; rather they are treated uniformly as propositions. L_{Arg} is essentially a propositional logic extended with certain modal operators. The modal operators $\langle bel \rangle$ and $\langle goal \rangle$ of L_{Arg} correspond to beliefs (Fagin, 1988; Hintikka, 1962) and goals (Cohen and Levesque, 1990) respectively. Propositions are supported by *collections of arguments*, and the confidence in a proposition or argument is represented by a number between 0 and 1. Suppose *D* is the dictionary [0, 1] with the top element Δ as 1. In addition, for each dictionary symbol $d \in D$, we have a modal "support" operator $\langle sup_d \rangle$ in L_{Arg}. The *formulae* (or *assertions*) of L_{Arg} extend the domain of propositional formulae to the domain of *formulae* as follows:

- *propositions* are formulae.
- $\langle bel \rangle F$ and $\langle goal \rangle F$ are formulae, where *F* is a formula.
- $\langle sup_d \rangle F$ is a formula, where *F* is a formula and *d* is in the dictionary *D*.
- $\neg F$ and $F \wedge G$ are formulae, where *F* and *G* are formulae.

We take $\perp$ (false) to be an abbreviation of $\neg\top$. Other logical connectives and the existential quantifier are defined using $\neg$ and $\wedge$ in the usual manner.

3.2 Example Sentences and Arguments

We provide here some example sentences of L_{Arg} that are translations of the decision construct shown in Figure 4. The situation and goal portion in the decision `game_status` is translated to the following modal rule:

$$\langle bel \rangle transport\_disruption \rightarrow \langle goal \rangle determine\_game\_status$$

The above L_{Arg} sentence states that if *transport_disruption* is believed, then a goal is *determine_game_status*. A goal is considered to be achieved as soon as it becomes true. In the context of the decision *game_status*, this is reflected in the following formulae:

$$\langle bel \rangle (on \wedge \neg cancelled \wedge \neg postponed) \rightarrow \langle bel \rangle determine\_game\_status$$
$$\langle bel \rangle (cancelled \wedge \neg on \wedge \neg postponed) \rightarrow \langle bel \rangle determine\_game\_status$$
$$\langle bel \rangle (postponed \wedge \neg on \wedge \neg cancelled) \rightarrow \langle bel \rangle determine\_game\_status$$

Figure 6: Translation of the goal in the decision construct shown in Figure 4

The first of the above four sentences (Figure 6) states that if it is believed that the game is on, but neither cancelled nor postponed, then *determine_game_status* is believed. In other words, the earlier goal *determine_game_status* is considered achieved upon believing that the game is on. The L_{Arg} representations for the arguments in the diagnosis decision are (Figure 7):

$$\langle bel \rangle transport\_disruption \rightarrow \langle sup_{0.7} \rangle cancelled$$
$$\langle bel \rangle \neg radio\_commentary \rightarrow \langle sup_{0.9} \rangle \neg on$$
$$\langle bel \rangle \neg rain \rightarrow \langle sup_{0.95} \rangle on$$
$$\langle bel \rangle bad\_economy \rightarrow \langle sup_{0.6} \rangle \neg cancelled$$
$$\langle bel \rangle (bad\_economy \wedge free\_slot) \rightarrow \langle sup_{0.7} \rangle postponed$$

Figure 7: Translation of the arguments in the decision construct shown in Figure 4

3.3 The Axioms

The axioms of L_{Arg} are divided into classical and modal axioms. For classical axioms, we consider every instance of a propositional tautology to be an axiom, and we also have the *modus ponens* inference rule. L_{Arg} adopts a standard set of axioms and inference rules of beliefs and goals in its reasoning and decision making, which can be found in (Cohen and Levesque, 1990; Meyer et al, 1991). A detailed explanation can be found in (Fox and Das, 2000). The L_{Arg} axioms and inference rules are:

$\neg\langle bel \rangle \perp, \neg\langle goal \rangle \perp$
$\langle bel \rangle F \wedge \langle bel \rangle (F \rightarrow G) \rightarrow \langle bel \rangle G$
$\langle bel \rangle F \rightarrow \langle bel \rangle \langle bel \rangle F$
$\neg\langle bel \rangle F \rightarrow \langle bel \rangle \neg \langle bel \rangle F$
$\langle goal \rangle F \wedge \langle goal \rangle (F \rightarrow G) \rightarrow \langle goal \rangle G$
$\langle bel \rangle F \rightarrow \langle goal \rangle F$
if $\vdash F$ then $\vdash \langle bel \rangle F$

We now present a set of axioms for the modal operator $\langle sup_d \rangle$. First of all, there can be no support for an inconsistency and this is axiomatized as follows:

$$\neg\langle sup_d \rangle \perp, \text{ for every } d \in D$$

The following inference rule states that the support operator is closed under implication. In other words, if F has support d and $F \to G$ is valid in L_{Arg} then G too has support d.

$$\text{if } \vdash F \to G \text{ then } \vdash \langle sup_d \rangle F \to \langle sup_d \rangle G, \text{ for every } d \in D$$

A valid L_{Arg} formula always has the highest support:

$$\text{if } \vdash F \text{ then } \vdash \langle sup_\Delta \rangle F$$

Support operators can be combined to obtain a single support operator by using the following axiom:

$$\langle sup_{d1} \rangle F \wedge \langle sup_{d2} \rangle G \to \langle sup_{d1 \otimes d2} \rangle (F \wedge G)$$

where $\otimes$: $D \times D \to D$ is the function for computing supports for assertions derived through material implication. The axiom states that if $d1$ and $d2$ are supports for F and G respectively then $\otimes(d1, d2)$ (or $d1 \otimes d2$ in infix notation) is a derived support for $F \wedge G$. Note that $d \otimes \Delta = d$, for every d in D. If $F = G$, then the above axiom basically aggregates two arguments for the decision option F. Such aggregation via belief networks will be presented in the following section. The following axiom says that every level of evidence for an assertion also implies every level of evidence for the assertion lower than the evidence:

$$\langle sup_{d1} \rangle F \to \langle sup_{d2} \rangle F, \text{ where } d2 \leq d1$$

3.4 Possible World Semantics

A *model* of L_{Arg} is a tuple

$$\langle W, V, R_b, R_s, R_g \rangle$$

in which W is a set of possible worlds. A world consists of a set of qualified assertions outlining what is true in the world. V is a valuation that associates each world with a subset of the set of propositions. In other words,

$$V: W \to \Pi(P)$$

where P is the set of propositions and $\Pi(P)$ is the power set of P. The image of the world w under the mapping V, written as $V(w)$, is the set of all propositions which are true in the world w. This means that p holds in w for each p in $V(w)$.

The relations R_b, R_s and R_g are the accessibility relations for beliefs, supports and goals respectively. For example, the relation R_b relates a world w to a set of worlds considered possible by the decision-maker from w. If there are n candidates for a decision that are active in a world w then there are n possible worlds.

The relation R_s is a *hyperelation* which is a subset of the set

$$W \times D \times \Pi(W)$$

Semantically, if $\langle w, d, W' \rangle \in R_s$ then there is an amount of support d for committing to one of the possible worlds in W' from the world w, where W' is non-empty. In other words, the support d is for the set of assertions uniquely characterized by the set of worlds W'.

An assertion is a *belief* of a decision maker at a world w if and only if it is true in all possible worlds that are accessible from the world w by R_b. Note that the members of R_s have been considered to be of the form $\langle w, d, W' \rangle$ rather than $\langle w, d, w' \rangle$. The main reason is that the derivability of $\langle sup_d \rangle F$ means F is true only in a "subset" of the set of all possible worlds accessible from w. If F is true in all possible worlds accessible from w then we would have had $\langle bel \rangle F$, which implies the highest form of support for F that is greater than or equal to d.

Due to the axioms related to the modal operator $\langle bel \rangle$, the standard set of properties that will be possessed by the accessibility relation R_b is:

Model Property 1: R_b is serial, transitive, and euclidean

The requirement that a decision maker may not believe in something that is inconsistent guarantees the existence of a possible world, which is the seriality property. The explanation for R_b being transitive and euclidean can be found in (Chellas, 1980; Lemmon, 1977).

The hyperelation R_s satisfies the following properties due to the axioms related to the modal operator $\langle sup_d \rangle$:

Model Property 2: For every w, w_1, w_2 in W and d, d' in D, the relation R_s satisfies the following conditions:

- if $\langle w, d, W' \rangle \in R_s$ then $W' \neq \emptyset$.
- if $\langle w, d, W' \rangle \in R_s$ then $\langle w, d', W' \rangle \in R_s$, for every $d' \leq d$
- $\langle w, \Delta, W \rangle \in R_s$.
- if $\langle w, d1, W_1 \rangle, \langle w, d2, W_2 \rangle \in R_s$ then $\langle w, d1 \otimes d2, W_1 \cap W_2 \rangle \in R_s$, provided $W_1 \cap W_2 \neq \emptyset$.

Explanation of each of these restrictions on R_s can be found in (Das and Fox, 2000).

Aggregation of arguments introduces a hierarchy of preferences among the set of all possible worlds accessible from w by the relation R_b. The maximal elements and possibly some elements from the top of the hierarchy of this preference structure will be called *goal worlds*. The relation R_g, which is a subset of R_b, relates the current world to the set of goal worlds. Only one of the goal worlds is committed for transition from the current world based on the aggregated support. This world will be called the *committed world*.

An assertion is a *goal* in a world w if and only if it is true in every goal world accessible from w by the accessibility relation R_g. Axiom $\neg\langle goal \rangle \perp$ introduces the seriality property on the accessibility relation R_g. Axiom $\langle bel \rangle F \to \langle goal \rangle F$ restricts R_g to a subset of R_b, that is, the set of goal worlds is a subset of the set of all possible worlds.

Model Property 3

- R_g is serial
- $R_g \subseteq R_b$: for every w and w' in W, if $w\, R_g\, w'$ then $w\, R_b\, w'$

The semantics of supports, beliefs and goals are as follows. Given a model $M = \langle W, V, R_b, R_s, R_g \rangle$, the truth values of formulae with respect to a world w are determined by the rules given below:

$\models_{Mw} \top$
$\models_{Mw} p$ iff $p \in V(w)$
$\models_{Mw} \langle sup_d \rangle F$ iff there exists $\langle w, d, W' \rangle$ in R_s such that $\models_{Mw} F$, for every $w' \in W'$
$\models_{Mw} \langle bel \rangle F$ iff for every w' in W such that $w\, R_b\, w'$, $\models_{Mw} F$
$\models_{Mw} \langle goal \rangle F$ iff for every w' in W such that $w\, R_g\, w'$, $\models_{Mw} F$
$\models_{Mw} \neg F$ iff $\not\models_{M\langle w,} F$
$\models_{Mw} F \wedge G$ iff $\models_{Mw} F$ and $\models_{Mw} G$

A formula F is said to be *true* in model M if and only if $\models_{Mw} F$, for every w in W. A formula F is said to be *valid* if F is true in every model.

Suppose Γ is the class of all models satisfying Model Property 1, Model Property 2, and Model Property 3. Then the soundness and completeness theorem establishes the fact that L_{Arg} is determined by Γ.

4 Aggregation of Probabilistic Arguments via Belief Networks

This section presents our approach to aggregating arguments via Bayesian belief network technology. This aggregation process is a meta-level reasoning that takes the clauses in the underlying knowledge base as input. The reasoning at the object or knowledge base level is carried out using the logic L_{Arg}. We first provide a brief background in the technology and then present the details of the approach.

4.1 Review of Bayesian Belief Networks

A Bayesian belief network (Pearl, 1988; Jensen, 1996) is a graphical, probabilistic knowledge representation of a collection of variables describing some domain. The nodes of the belief network denote the variables and the links denote causal relationships between the variables. The topology encodes the *qualitative* knowledge about the domain. Conditional probability tables (CPTs) encode the *quantitative* details (strengths) of the causal relationships between a node and its parents. In other words, the CPTs are *local* joint probability distributions involving subsets of the whole domain. For example, if a variable, *x*, is 4-valued and has one parent variable, *y*, which is 3-valued, then *x*'s CPT can be represented as a 3x4 table where the $(i,j)^{th}$ entry is $p(x_j|y_i)$. The belief network of Figure 8 encodes the relationships over a simple domain consisting of the six binary variables, *Injury, Rain*, *Game, Transport*, *Electricity*, and *Commentary*.

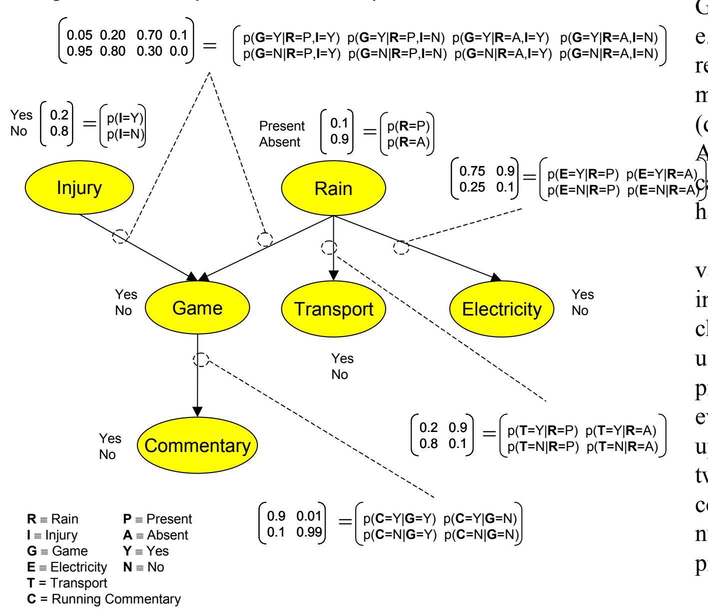

Figure 8: Simple Bayesian belief network

The topology captures the commonsense knowledge that:

1. *Rain* causes *Transport* disruption
2. *Rain* causes *Electricity* failure
3. *Game* causes running *Commentary* on the radio
4. *Injury* and *Rain* prevent *Game* from being played

As shown in Figure 8, the CPT specifies the probability of each possible value of the child variable conditioned on each possible combination of parent variable values. For example, the probability of having electricity given that rain is present is 0.75, whereas the probability of having electricity given clear skies is 0.9.

The structure of a belief network encodes other information as well. Specifically, the lack of links between certain variables represents a lack of direct causal influence, that is, they indicate conditional independence relations. This belief network encodes many independence relations, for example,

1. $Electricity \perp Transport \mid Rain$
2. $Commentary \perp \{ Rain, Electricity \} \mid Game$

where '⊥' is read 'is independent of' and '|' is read 'given.' Once the value of *Rain* is known, the value of *Transport* adds no further information about *Electricity*. Similar conditional independence assertions hold for other variables.

A central feature of the BN formalism is that the belief vector is decomposed as a product of the total *causal* evidence at *x*, which comes from *x*'s parents, and the total *diagnostic* evidence at *x*, which comes from *x*'s children. Root nodes are special cases; they require some initial estimate for their causal evidence vectors. Belief vectors generally change as new evidence regarding any of the variables is added to the network. Thus, if we obtain new evidence of electricity being present, our initial belief about rain, i.e. (Present = 0.1, Absent = 0.9), should be revised accordingly, e.g. to (Present = 0.2, Absent = 0. 8). This is an example of *diagnostic* reasoning from effects back to possible causes. This new evidence should also cause us to revise our belief vector for Game to reflect a higher probability that the game will be played, e.g. to (Yes = 0.91, No = 0.09). This is an example of *causal* reasoning from causes to effects. Thus, belief nets can support the model-based anomaly diagnosis both by hypothesis generation (diagnostic reasoning) and hypothesis testing (causal reasoning). Additionally, the topologies of the networks themselves can capture the structure and interconnection of the components at hand in an aggregate and easily understood manner.

When new evidence is posted to a variable in a BN, that variable updates its own belief vector, then sends out messages indicating updated predictive and diagnostic support vectors to its children and parent nodes respectively. These messages are then used by the other nodes to update their belief vectors and propagate their own updated support vectors. The separation of evidence yields a propagation algorithm (Pearl, 1988) in which update messages need only be passed in one direction between any two nodes following posting of evidence. Thus, the algorithm's complexity in a polytree type of network is proportional to the number of links in the network. This separation also automatically prevents the possibility of double-counting evidence.

In summary, a Bayesian Belief Network (Pearl, 1988; Lauritzen and Spiegelhalter, 1988) offers these principal advantages compared to other probabilistic reasoning methods:

1. Its use of cause/effect relationships is intuitive.
2. Its probability estimates are guaranteed to be consistent with probability theory.

The following section details our use of belief network technology for aggregating arguments for and against decision options.

4.2 Aggregation of Arguments

An argumentation based decision-making framework like the one described here is functionally similar to classical rule-based experts systems, with the following exceptions:

- It deals with more expressive knowledge in the form of arguments, than simply rules and a variety of dictionaries.
- It incorporates an inference mechanism which is capable of aggregating arguments for and against decision options and therefore more general than simple forward chaining.

While various types of classical, modal, and temporal logics can be used to represent and reason deductively with arguments, inferencing schemes within logics are insufficient for aggregating arguments, as the typical aggregation process is a meta-level reasoning involving sets of arguments. We propose here a scheme for aggregating arguments via Bayesian belief networks. The evidence propagation mechanism in belief networks implements both abductive and deductive inference schemes. While it is easier to elicit a set of arguments, constructing a belief network involves a more methodical approach to knowledge elicitation, and is usually much more time consuming. But a major advantage of an argumentation based framework is that support can be provided for making decisions even with a very few arguments, making the framework highly robust. But the propagation algorithm in a belief network fails to work even if a single entry within a CPT of the network is missing.

As pointed out in (Korver and Lucas, 1993), due to differences in the type of knowledge represented and in the formalism used to represent uncertainty, much of the knowledge to building an equivalent belief network could not be extracted from a rule-based expert system. In our approach, we will be able to extract the network structure fully, but cannot extract every entry in the conditional probability tables. The missing probabilities for variable states are assumed by default to be equally distributed. There are various approaches (Krause, 2000) to learning belief networks from sample data sets. For example, the approach taken in (Heckerman, 1996; Ramoni and Sebastiani, 1997) considers cases where both network structures and probabilities can be learned. The major assumption for learning probabilities from a complete data set is that the distribution for the variable representing probability vectors is considered to be *Dirichlet*. On the other hand, the *Gibbs sampling* technique is often employed to deal with incomplete data sets. Such techniques can be easily incorporated within our approach to estimate the probabilities that were assumed by default, provided relevant sample data sets are available.

Jitnah et. al. (2000) generates rebuttals in a Bayesian argumentation system based on normative and user models, represented in belief networks, that are manually constructed beforehand. The tutoring system proposed in (Conati et. al., 1997) automatically generates and updates belief networks during its interaction with the student for solving a problem. However, these approaches are only vaguely related to our approach to building a belief network, which is to be used for aggregating arguments, and does not seek for additional knowledge from the decision maker. We first construct fragments of networks using the arguments relevant to the decision-making task at hand. Note that, given a network fragment with a variable, and its parents and CPT, the fragment can be equivalently viewed as a set of arguments. For example, consider the network fragment in Figure 9, which states that player injury and rain together can determine the status of the game.

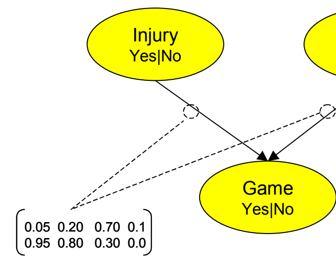

Figure 9: Example belief network fragment

Each column of the CPT yields an argument for and an argument against a state of the variable Game. For example, if there is player injury and it rains then there is an argument for a game with support 0.05.

```
injury & rain => support(game, 0.05)
```

Since the arguments are probabilistic, corresponding to the above argument there will be another argument which states that if there is player injury and it rains then there is an argument against the game with support 1 – 0.05, that is, 0.95, yielding the following:

```
injury & rain => support(not game, 0.95)
```

The rest of the entries of the CPT can be translated to arguments in a similar manner.

Continuing with our illustration of the network construction process from a set of arguments, consider the decision construct shown in Figure 4. Each argument with a single antecedent is translated to a network fragment containing two random variables corresponding to the antecedent and the consequent of the argument. For example, the argument

```
transport_disruption =>
                 support(cancelled, 0.7)
```

is translated to the network fragment on the left of Figure 10, which has two nodes or random variables: one for the antecedent `transport_disruption` and the other one for the decision option in the consequent. Since a particular decision option may occur in consequents of many arguments, their corresponding nodes in the network fragments are numbered to avoid ambiguity. Thus, the consequent of the above argument is translated to a node labeled Cancelled-1.

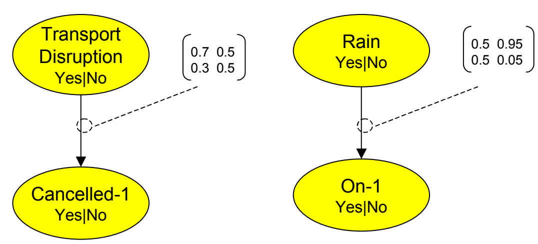

Figure 10: Belief network fragments by converting arguments

The following entry in the CPT comes directly from the argument:

$$P(\mathit{Cancelled}\text{-}1 = \mathit{Yes} \mid \mathit{Transport\ Disruption} = \mathit{Yes}) = 0.7$$

$$P(\mathit{Cancelled}\text{-}1 = \mathit{No} \mid \mathit{Transport\ Disruption} = \mathit{Yes}) = 0.3$$

The above type of probabilities will be equivalently written as the following:

$P(Cancelled\text{-}1 \mid Transport\ Disruption) = 0.7$

$P(not\ Cancelled\text{-}1 \mid Transport\ Disruption) = 0.3$

In case of no transport disruption, we have no information relating it to the cancellation of the game. Therefore, the probability distribution among the cancellation and non-cancellation states is even (uniform) given there is no transport disruption:

$P(Cancelled\text{-}1 \mid not\ Transport\ Disruption) = 0.5$

$P(not\ Cancelled\text{-}1 \mid not\ Transport\ Disruption) = 0.5$

Similarly, the network fragment on the right of Figure 10 is obtained by translating the argument

```
not rain => support(on, 0.95)
```

In this case, the above argument generates the following entries of the CPT:

$P(On\text{-}1 \mid not\ Rain) = 0.95$

$P(not\ On\text{-}1 \mid not\ Rain) = 0.05$

Since we cannot say anything about the state of the game given rain, the other two entries of the CPTs are as follows:

$P(On\text{-}1 \mid Rain) = 0.5$

$P(not\ On\text{-}1 \mid Rain) = 0.5$

An argument with multiple conditions is translated into a network fragment in a similar manner. Consider the following argument for postponing the game that has two conditions:

```
bad_economy & free_slot =>
        support(postponed, 0.7)
```

The translated network is shown in Figure 11. Observe that we are only able to fill in only one column of the CPT and each of the rest of the columns is uniformly distributed.

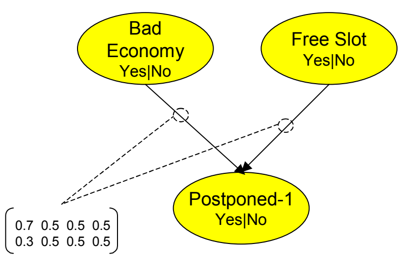

Figure 11: Belief network fragment by converting arguments with multiple conditions

After translating each individual argument to a belief network fragment, the next task is to aggregate arguments for and against each decision option. The heuristic used here is that the probability distribution of the two states of the variable corresponding to a decision option after the aggregation is proportional to the number of arguments for and against the decision option. For example, if we have three arguments for the decision option On via the three nodes On-1, On-2, and On-3, and no arguments against then we have the following probabilities for and against On:

$P(On \mid On\text{-}1, On\text{-}2, On\text{-}3) = 1.0$

$P(not\ On \mid On\text{-}1, On\text{-}2, On\text{-}3) = 0.0$

On the other hand, for example, if we have two arguments for the decision option On via the two nodes On-1 and On-2 and one argument against via the node On-3 then we have the following:

$P(On \mid On\text{-}1, On\text{-}2, not\ On\text{-}3) = 2/3$

$P(not\ On \mid On\text{-}1, On\text{-}2, not\ On\text{-}3) = 1/3$

This is illustrated in Figure 12.

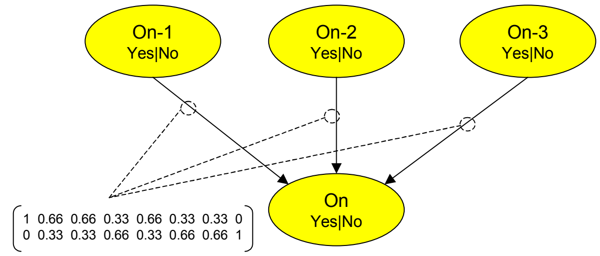

Figure 12: Belief network fragments by converting arguments for/against a decision option

Now that we have network fragments for arguments for and against individual decision options, we need to combine these arguments to rank the decision options. For this, we create a random variable with the states corresponding to the decision options for the task at hand. In the context of our example, we create a random variable called Game with three states On, Cancelled, and Postponed. The variable has three parents corresponding to the three decision options. The decision options are ranked based on the aggregation of arguments for and against the decision options; the values of the CPT are determined accordingly. For example, if we have aggregated evidence for each of the three decision options On, Cancelled, and Postponed, then the probability distribution of the variable Game is evenly distributed as follows:

$P(Game = On \mid On, Cancelled, Postponed) = 0.33$

$P(Game = Cancelled \mid On, Cancelled, Postponed) = 0.33$

$P(Game = Postponed \mid On, Cancelled, Postponed) = 0.33$

Note that we have the same probability distribution when we have aggregated evidence against each of the three decision options. On the other hand, for example, if we have aggregated evidence for each of the two decision options On and Cancelled, and aggregated evidence against the decision option Postponed, then the probability distribution on the states of the variable Game is as follows:

$P(Game = On \mid On, Cancelled, not\ Postponed) = 0.5$

$P(Game = Cancelled \mid On, Cancelled, not\ Postponed) = 0.5$

$P(Game = Postponed \mid On, Cancelled, not\ Postponed) = 0.0$

This is illustrated in Figure 13.

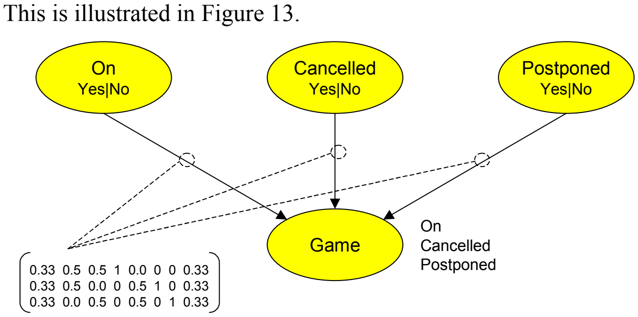

Figure 13: Belief network fragment for aggregating arguments for/against decision options

Figure 14 shows the combined network for aggregating the arguments of the decision construct in Figure 4. Such a network has three blocks: the Argument Block, the Axiom Block, and the Aggregation Block. The Argument Block is constructed out of the network fragments obtained by translating the arguments in the decision construct. The Axiom Block, to some extent, implements a specific case of axiom $\langle sup_{d1}\rangle F \wedge \langle sup_{d2}\rangle G \rightarrow \langle sup_{d1\otimes d2}\rangle(F \wedge G)$ (when F = G). The Aggregation Block implements the commitment rule in the decision construct. Mismatch is expected between the network in Figure 8 and that of in Figure 14 as any complete network of the former type is carefully constructed via a knowledge elicitation effort. (One can always incorporate additional knowledge from experts into the constructed network for improved prediction.)

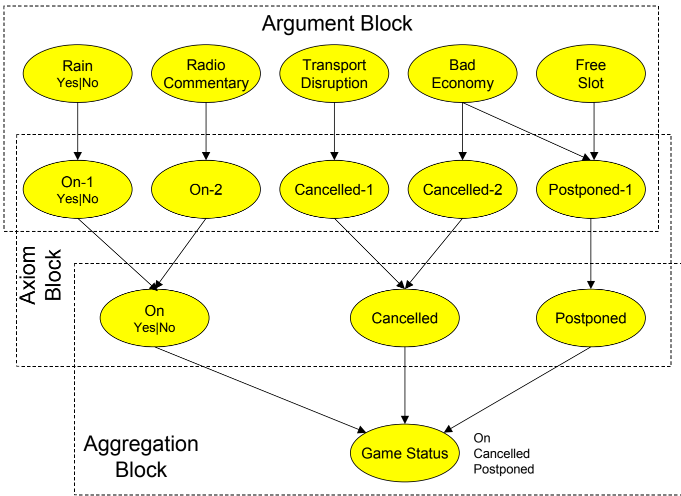

Figure 14: Combined belief network for argument aggregation

In the absence of any evidence, no arguments are generated and the *a priori* probabilities of the decision options are as follows:

$P(Game = On) = 0.33$

$P(Game = Cancelled) = 0.32$

$P(Game = Postponed) = 0.35$

No evidence in the network has been posted at this stage, not even for any prior beliefs on the variables. Now, given that there is transport disruption and rain, the network ranks the decision options based on the following posterior probabilities (as shown in the figure):

$P(Game = Postponed \mid Transport\ Disruption, Rain) = 0.37$

$P(Game = Cancelled \mid Transport\ Disruption, Rain) = 0.37$

$P(Game = On \mid Transport\ Disruption, Rain) = 0.26$

The dilemma occurs between the two decision options Cancelled and Postponed. If we now receive information about the unavailability of free slots then the network ranks the decision options as follows:

$P(Game = Cancelled \mid Disruption, Rain,\ not\ Free\ Slot) = 0.38$

$P(Game = Postponed \mid Disruption, Rain,\ not\ Free\ Slot) = 0.34$

$P(Game = On \mid Disruption,\ Rain,\ not\ Free\ Slot) = 0.28$

Based on the above probability distribution, the decision maker may decide to commit to the decision option Cancelled.

5 An Example

We present here a concrete example illustrating the proposed argumentation based decision-making process and belief network based aggregation.

Suppose the current world w_0 consists of the sentences in the syntax of L_{Arg}, shown in Figure 6 and Figure 7, obtained by translating the specification of the `game_state` decision, shown in Figure 4. In addition, we consider the following set of beliefs and knowledge (knowledge is defined as $F \wedge \langle bel\rangle F$) as part of the decision maker's knowledge base at w0:

$$\{rain, \langle bel\rangle transport\_disruption\}$$

We cannot uniquely define the valuation on w_0 as the set of formulae that characterize w_0 if it contains assertions that are only believed, such as $\langle bel\rangle transport\_disruption$. An example valuation S on w_0 is the following:

$$S = \{rain, transport\_disruption, cancelled\}$$

Since there are 3 candidates in the `game_state` decision (*on*, *cancelled*, and *postponed*) and we are dealing with probabilistic arguments, these three options will be considered mutually exclusive and exhaustive (which is not the case in general) for the purpose of aggregation:

C1 $= on$, C2 $= cancelled$, C3 $= postponed$

Consequently, there will be 3 possible worlds w_1, w_2, and w_3, whose valuations are as follows (see figure):

$V(w_1) = S \cup \{on, determine\_game\_status\}$

$V(w_2) = S \cup \{cancelled, determine\_game\_status\}$

$V(w_3) = S \cup \{postponed, determine\_game\_status\}$

Note that the presence of $\langle bel\rangle transport\_disruption$ in the knowledge base along with the argument

$$\langle bel\rangle transport\_disruption \rightarrow \langle sup_{0.7}\rangle cancelled$$

derives $\langle sup_{0.7}\rangle cancelled$ from the knowledge base. Now the argument $\langle bel\rangle \neg rain \rightarrow \langle sup_{0.95}\rangle on$ states that P(On | not Rain) = 0.95. But we have *rain* in the knowledge base and our implicit assumption is P(On | Rain) = 0.5. Therefore, $\langle sup_{0.5}\rangle on$ can be derived from the knowledge base.

The relations R_b and R_s in the model definition are defined as follows:

$R_b = \{\langle w_0, w_1\rangle, \langle w_0, w_2\rangle, \langle w_0, w_3\rangle\}$

$R_s = \{\langle w_0, 0.95, \{w_2\}\rangle, \langle w_0, 0.5, \{w_1\}\rangle\}$

Note that *determine_game_status* is true in each of the possible worlds and therefore this is a goal - since the set of goal worlds is a subset of the set of possible worlds. This corresponds to the provability of $\langle goal\rangle determine\_game\_status$ in the current world using $\langle bel\rangle transport\_disruption$ in conjunction with the formula

$$\langle bel\rangle transport\_disruption \rightarrow \langle goal\rangle determine\_game\_status$$

The goal is active in w_0 since game status is not yet determined or *determine_game_status* is not yet believed. We are assuming here that the L_{Arg} theorem prover is able to derive the negation of $< bel > determine\_game\_status$ from the current world by a mechanism similar to negation by failure. Belief network based aggregation process (as described in the last section) computes the supports for the candidates C1, C2, and C3 as follows:

Total support for: C1 $= 0.26$, C2 $= 0.37$, C3 $= 0.37$

The preference relation << among the set of possible worlds is derived as $w_1 << w_2$ and $w_1 << w_3$. The maximally preferred possible worlds are w_2 and w_3. The relation R_g in the model definition is now defined as follows (Figure 15):

$R_g = \{\langle w_0, w_2\rangle, \langle w_0, w_3\rangle\}$

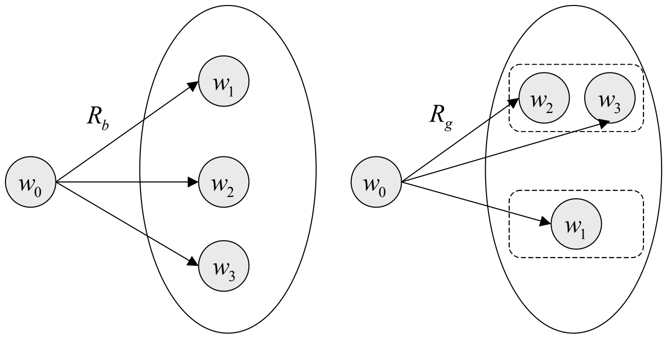

Figure 15: Relations between the current and possible worlds

This produces a dilemma. If the decision maker cannot gather any more evidence it may commit to w_2 by preferring w_2 to w_3. This involves adding the beliefs cancelled, not on, and not postponed to the current state of the database depending on the strength of support for them. In the new situation the goal to determine the status of the game will no longer be active, as *determine_game_status* it will be believed due to the presence of

$$\langle bel\rangle(cancelled \wedge \neg on \wedge \neg postponed) \rightarrow \langle bel\rangle determine\_game\_status$$

and the beliefs in *cancelled*, $\neg$*on*, and $\neg$*postponed*. Alternatively, if additional evidence is available to the decision-maker about the hosting club's financial situation, say $\langle bel\rangle\neg bad\_economy$, that will increase the total support for C1 as follows:

Total support for: C1 = 0.26, C2 = 0.41, C3 = 0.33

The revised valuation on each w_i will be as before except the additional evidence $\neg bad\_economy$ changes its truth value. The relations R_s and R_g may be redefined as follows:

$$R_s = \{\langle w_0, 0.95, \{w_2\}\rangle, \langle w_0, 0.5, \{w_1\}\rangle, \langle w_0, 0.5, \{w_2\}\rangle\}$$

$$R_g = \{\langle w_0, w_2\rangle\}$$

Since w_2 is the only goal world, the decision-maker considers w_2 as the committed world. Changing to the committed world from the current world involves adding $\langle bel\rangle cancelled$ and $\langle bel\rangle\neg on$, $\langle bel\rangle\neg postponed$ to the database as the decision-maker's beliefs. Adding $\langle bel\rangle cancelled$ to the database will trigger the decision for alternative activity (shown in Figure 5) and the decision making process continues as before.

6 Conclusion

In this paper, we have presented L_{Arg}, a logic for reasoning with probabilistic arguments, along with an approach for aggregating arguments via Bayesian belief networks. The semantics of L_{Arg} is given by enhancing the traditional possible world semantics with a new accessibility relation for support, and the soundness and completeness result is established. In the future, we plan to deal with more general forms of arguments than just propositional sentences, and enhance our proposed aggregation algorithm to aggregate temporal arguments via dynamic belief networks.

Acknowledgements

The author would like to thank Dave Lawless and John Fox for many helpful discussions on the topic, and the two anonymous reviewers for their constructive criticisms, that have helped to improve the paper enormously.

7 References

[1] Chellas, B. 1980. "Modal Logic." Cambridge, U.K.: Cambridge University Press.

[2] Cohen, P. R. and Levesque, H. J. (1990). "Intention is choice with commitment." *Artificial Intelligence*, Vol. 42, 13-361.

[3] Conati, C., Gertner, A., van Lehn, K., and Druzdzel, M. (1997). "On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks." *Proceedings of the Sixth International Conference on User Modeling*.

[4] Das, S., Fox, J., Elsdon, D., and Hammond, P. (1997). "A flexible architecture for autonomous agents", *Journal of Experimental and Theoretical AI*, 9(4): 407-440.

[5] Das, S. and Grecu, D. (2000). "COGENT: Cognitive agent to amplify human perception and cognition." *Proceedings of the 4th Int. Conf. On Autonomous Agents*, Barcelona, June.

[6] Fagin, R. 1988. "Belief, Awareness, and Limited Reasoning." *Artificial Intelligence*, 34(1):39–76.

[7] Fox, J. and Das, S. K. "Safe and Sound: Artificial Intelligence in Hazardous Applications," AAAI-MIT Press, June 2000.

[8] Fox, J., Krause, P, and Ambler, S. (1992). "Arguments, contradictions, and practical reasoning." *Proceedings of the Tenth European Conference on Artificial Intelligence*, Vienna, August, pp. 623-626.

[9] Heckerman, D. (1996). "A tutorial on learning with Bayesian networks." Microsoft Technical Report MSR-TR-95-06.

[10] Hintikka, J. 1962. "Knowledge and Belief." Ithaca, N.Y.: Cornell University Press.

[11] Jensen, F.V. (1996). "An Introduction to Bayesian Networks." Springer-Verlag.

[12] Jitnah, N., Zukerman, I., McConachy, R., and George, S. (2000). "Towards the generation of rebuttals in a Bayesian argumentation system." *Proceedings of the 1st Int. Natural Language Generation Conf.*, pp. 39-46.

[13] Korver, M. and Lucas, P. (1993). "Converting a rule-based expert system into belief network." *Medical Informatica*, Vol. 18(3), pp. 219-241.

[14] Krause, P. J. (1998). "Learning probabilistic networks." *The Knowledge Engineering Review*, Vol. 13:4, pp. 321-325

[15] Lauritzen, S. L. and D. J. Spiegelhalter (1988). "Local computations with probabilities on graphical structures and their applications to expert systems." *Journal of the Royal Statistical Society*, B 50 (2), pp.154-227.

[16] Lemmon, E. J. 1977. "An Introduction to Modal Logic." Basil, U.K.: Blackwell.

[17] Meyer, J.-J., and Vreeswijk, G. A. (1991). "Epistemic Logic for Computer Science: A Tutorial. Part 1." *Bulletin of European Association for Theoretical Computer Science* (EATCS) 44(4): 242–270.

[18] Pearl, J. (1988). "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference." San Mateo, CA, Morgan Kaufmann.

[19] Ramoni, M. and Sebastiani, P. (1997). "Learning Bayesian networks from incomplete databases." Technical Report KMI-TR-43, The Open University, UK.

[20] Rasmussen, J. (1983). "Skills, Rules and Knowledge: Signals, Signs and Symbolism, and Other Distinctions in Human Performance Models." *IEEE Transactions on Systems, Man, and Cybernetics*, 12: 257-266.

[21] Toulmin, S. 1956. The Uses of Argument. Cambridge, U.K.: Cambridge University Press.