Abstract

MALLYA, ASHOK ULLAL. Specifying and Resolving Temporal Commitments. (Under the direction of Munindar P. Singh).

Commitments are a powerful representation for modeling multiagent protocols, especially for applications such as electronic commerce, where contracts are a natural component of the desired interactions. Previous approaches have considered the semantics of commitments and how to check compliance with them. However, these approaches, although valuable, do not capture some of the subtleties that arise in applications of commitments in real-life settings. In particular, practical contracts and institutions have subtle temporal properties. The present thesis develops a rich representation for the temporal content of commitments. This enables us to capture realistic contracts and institutions rigorously, and avoid subtle ambiguities. Consequently, this approach enables us to reason about questions of great practical import, for example, whether and when exactly a commitment is satisfied or breached and whether it is or ever becomes unenforceable.

SPECIFYING AND RESOLVING TEMPORAL COMMITMENTS

BY

ASHOK ULLAL MALLYA

A THESIS SUBMITTED TO THE GRADUATE FACULTY OF

NORTH CAROLINA STATE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

RALEIGH

DECEMBER 2002

APPROVED BY:

PETER R. WURMAN

DENNIS R. BAHLER

MUNINDAR P. SINGH

CHAIR OF ADVISORY COMMITTEE

To my parents.

Biography

Ashok Mallya was born on September 10, 1978 in Mangalore, India. He received his Bachelors Degree in Computer Science from the Karnataka Regional Engineering College, Surathkal, India – now known as the National Institute of Technology, Surathkal – in the summer of 2000. He has been a Masters student at the North Carolina State University, in the Department of Computer Science, since August 2000.

Acknowledgment

You can take a kid to college, but you cannot make him think.

Dr. Singh did not take me to college, but he made me think in more ways than I thought possible. His constant encouragement has been inspiring.

I thank Dr. Bahler and Dr. Wurman for being on my thesis committee and suggesting changes that improved this work.

I thank, in no particular order, Raghu, Pınar, Subhayu and Amit for pretending to be interested in this thesis and giving valuable feedback. I thank Amit also for letting me win at Tennis and Badminton once in a while, just to keep me happy.

I thank Mike Maximillien, Zhengang Cheng, and Dr. Bin Yu for their opinions about this work, and Ruben, Bhargava, Vishwas, and Kolli for making our apartment a comfortable place to go back to after long hours of work.

I thank NC State University and the United States of America for making it very easy for an honest graduate student to earn a good living.

I thank Rasquinha, Sandhya, Mathai, and Pise for the Friday-evening debates.

I also owe a debt of gratitude to Champions Pool and Billiards and Mitch's Tavern; to Heather at Champions and Kristen at Mitch's for being good hostesses.

I thank Shalini and Dinesh for giving me a home away from home.

I thank Rashmi and Vikram for their support and friendship.

I am forever indebted to my parents for all that they have done for me.

There are many more people that have helped in one way or another during the course of this work. Although I cannot thank them individually here for fear of making the acknowledgments section larger than the actual thesis, they will always be gratefully remembered.

This research was supported partly by the North Carolina State University, and partly by the National Science Foundation under grant DST-0139037.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Objectives

Protocols help streamline the complex interactions that can take place between autonomous, heterogeneous agents in a multiagent system. A special application setting is electronic commerce, where the agents represent different parties that do business online.

The role of commitments in modeling such rich interactions is widely recognized, because they enable the key content of an interaction to be represented and reasoned about, especially in the face of opportunities or exceptions. Commitments are thus more expressive than traditional formalisms such as finite state machines. Commitments have begun to be operationalized in recent work. Yolum and Singh [2002b; 2002a] introduce commitment machines to show how to build and execute commitment-based protocols and show how to reason about commitment protocols in the event calculus. Fornara and Colombetti [2002] capture key aspects of the commitment life-cycle and further advance the idea of commitments as a data structure. Some compliance aspects of commitment protocols in a branching-time temporal logic with potential causality have also been studied [Venkatraman and Singh, 1999].

Motivation. The above approaches show that commitments provide a viable representational framework for designing, executing, and validating flexible protocols in multiagent systems. However, current approaches take an impoverished view of the temporal aspects of commitments. This can prove to be a drawback for their use in real systems, since business deals and legal agreements are usually not merely simple *if-then-else* rules about atomic propositions. They typically involve many clauses and sub-clauses, and most services have subtle time periods of reference. The following is an informal list of some properties that are relevant in practice, but not naturally handled by current approaches.

- *Time intervals*. Contracts and real-life commitments in general often involve time bounds. Having these time bounds simplifies decisions about the satisfaction or breach of commitments, which is one of the reasons traditional representations (e.g., paper documents) rely on them.
- *Maintenance*. Current work on commitments has concentrated on achievement conditions, whereas in real-life settings commitments are as likely to be about the maintenance of certain conditions. For example, a typical service-level agreement (SLA) may involve committing to maintaining network connectivity during business hours.
- *Temporal anaphora*. A particular variety of time bounds arise in the notion of temporal anaphora, as introduced by Partee [1984]. A promise such as "I will send you the goods" or a claim such as "I tried to call you five times" involves an implicit range of salient times within which the specified action occurred or will occur. Although we are not concerned here with commonsense reasoning, our representational framework for commitments should be able to accommodate the results of such reasoning.

Point-based temporal logics, which are commonly used in distributed system specifications, are inadequate to express the above requirements. Accordingly, we develop an extension of the well-known branching-time logic, Computation Tree Logic (CTL) [Emerson, 1990], which can capture the cases of interest here. Our main contribution is in applying this extension to commitments and showing how commitments can be specified using it and how their satisfaction or breach can be detected. Further, the temporal aspects of commitments are independent of the domain-specific semantics of the condition that the commitment is about, so that we can reason about the temporal aspects of commitments in a domain-independent manner.

Challenges. We use the fishmarket protocol as a framework for our motivating examples to show how our approach applies to various problems in a simplified, but still quite realistic, electronic commerce setting [Rodríguez-Aguilar et al., 1998].

The fishmarket protocol is a bidding protocol in which the price of the commodity being auctioned increases as bidding proceeds in the auction. The protocol is based on traditional Spanish fishmarkets where the seller of the fish is the auctioneer, who shouts out the price of the fish to a crowd of bidders. The bidders say "yes" if they are willing to buy at that price, or say "no" (or stay quiet) if they are not interested in buying at that price. The auctioneer goes on raising the price of the fish and announcing it as long as there are multiple bidders who are ready to buy the fish for the announced price. Normally, we expect the number of bidders who say "yes" to decrease with increasing price. The auction ends when only one bidder says "yes" for a particular price. The fish is sold to that bidder for that price.

The following cases of agent interactions that could take place in a fishmarket-based protocol motivate our study and results.

Example 1 The fishmarket wants to assure its buyers that it will hold an auction on at least one day this week. How would we specify this commitment? ▮

Example 2 The fishmarket promises the buyers that sometime today (a Tuesday) it will bring about the condition that there will be a continuous auction on weekends. Such a statement makes no sense in natural language, since something about the future can be promised today, but not brought about today. But can agents that do not comprehend natural language as well as we do detect the impossibility of resolution of such commitments? ▮

Example 3 An auctioneer in the fishmarket assures potential buyers that the fish will remain fresh for two days, or she will give them their money back. However, the auction will end two hours from now, after which she can leave the auction house and has no obligations towards the buyers or the auction house. How can buyers detect such clauses that render a seemingly well intentioned commitment unenforceable? ▮

These questions are answered in Sections 4.1, 4.2, and 4.3, respectively.

Organization. The organization of the paper is as follows. Chapter 2 provides an introduction to the background concepts of branching time and commitments. Chapter 3 develops our technical approach, including our formal language and how to reason with it. Sections 4 explains our results on the resolution of commitments that use temporally qualified propositions, and Chapter 5 summarizes our proposal and identifies directions of further research.

Chapter 2

Background Concepts

We next briefly explain our model of time and our temporal logic, and introduce the notion of commitments.

2.1 The Temporal Framework

We use a discrete, branching-time model, as shown in Figure 2.1.

The temporal model has the following features:

F_1. The world is a set of discrete *moments* in time. $\mathbb{M}$ is the set of all possible moments.

F_2. Moments are partially ordered by the relation $\prec$, which indicates temporal precedence. That is, if moment m_1 occurs before m_2, then $m_1 \prec m_2$. Also, $m_x \preceq m_y$ is defined as $m_x \prec m_y$ or $m_x = m_y$, where $m_x = m_y$ iff m_x and m_y both denote the same moment. The relation $\prec$ is transitive, asymmetric, and irreflexive.

F_3. Each moment is associated with a state of the world, which is fully defined by the state of all the propositions that hold at that moment. A condition q is said to be achieved when when a state is reached in which q holds.

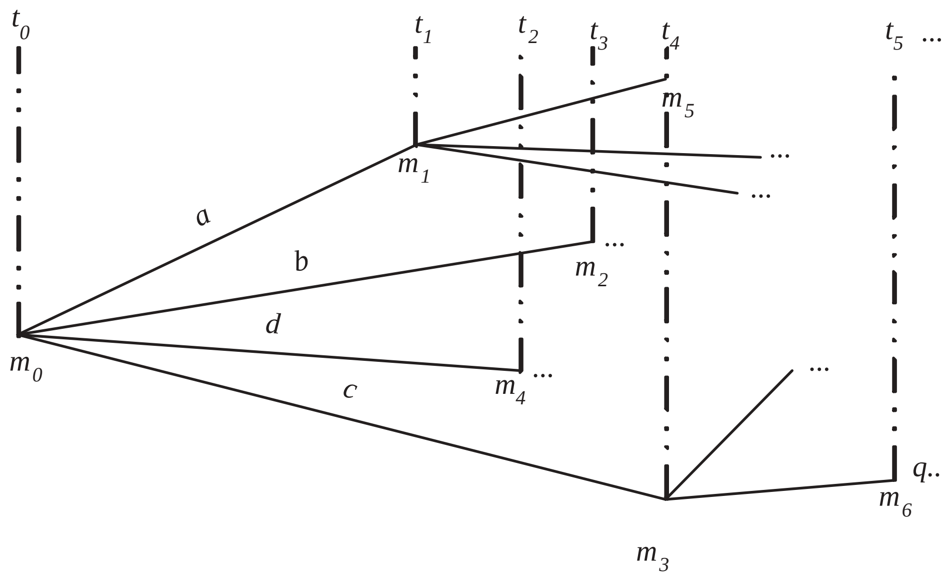

Figure 2.1: A schematic representation of the model of time

F_4. The past is considered linear, and the future branches into as many branches as there are outcomes of actions that are possible at that moment (for the agents in the system).

F_5. We use timestamps to be able to represent deadlines and fixed intervals. Each timestamp denotes an instant of time. $\mathbb{T}$ denotes the set of all timestamps. Timestamps are assumed to be discrete.

F_6. Moments and timestamps are assumed to be equally dense; i.e, we can relate exactly one timestamp with every moment and exactly one moment in any scenario with every timestamp. The function τ assigns a timestamp to every moment, i.e.,

$$\tau : \mathbb{M} \mapsto \mathbb{T}$$

F_7. The relation $<$ defines a total ordering over the set $\mathbb{T}$. That is, $(\forall t_x, t_y \in \mathbb{T} : t_x = t_y$ or $t_x < t_y$ or $t_y < t_x)$.The relation $<$ is transitive, asymmetric, and irreflexive.

F$_8$. Since $\prec$ is a partial ordering and $<$ is a total ordering,

$$\forall m_1, m_2 \in \mathbb{M} : m_1 \prec m_2 \Rightarrow \tau(m_1) < \tau(m_2)$$

However,

$$\forall t_1, t_2 \in \mathbb{T}, t_1 < t_2 :$$
$$\forall m_1, m_2 \in \mathbb{M} : (\tau(m_1) = t_1 and \tau(m_2) = t_2) \not\Rightarrow (m_1 \prec m_2)$$

For example, in Figure 2.1, $t_2 < t_3$, $\tau(m_4) = t_2$, and $\tau(m_2) = t_3$. However, $m_4 \not\prec m_2$.

F$_9$. A *scenario* S at a moment is a maximal set of moments containing the given moment and all moments along some branch in the future of the given moment. It represents one possible path of progression of the universe. The set of all scenarios at a moment m is represented by $\mathbb{S}_m$. A scenario is a subset of $\mathbb{M}$. A scenario S has the following properties:

- S is *rooted*; i.e., $S \neq \{\}$
- S is *linear*; i.e., $(\forall m_1, m_2 \in S : m_1 = m_2$ or $m_1 \prec m_2$ or $m_2 \prec m_1)$

For example, in Figure 2.1, the path $m_0 m_1 m_5$ is a scenario rooted at m_0

We use an extension of Computational Tree Logic (CTL) [Emerson, 1990] as the temporal logic to model commitments. We now introduce the components of the CTL language and their intended meanings.

1. *Boolean operators*. The logical operators true, false, $\neg$, $\vee$, $\wedge$ carry their usual meaning.

2. *Linear time operators*. These are operators that apply over a particular scenario.

- U: A proposition $p\mathsf{U}q$, read *p until q*, is true at a moment m_i on a scenario, iff q holds at some moment m_x in the future on the given scenario and p holds at all moments between m_i and m_x.
- F: A proposition $\mathsf{F}p$, read *eventually p*, means that p holds at some point on the future in the given scenario. It is equivalent to $\mathsf{true}\mathsf{U}p$.
- G: A proposition $\mathsf{G}p$, read *always p*, means that p holds at all moments in the future on the given scenario.

G and F are duals $\mathsf{G}p \Leftrightarrow \neg\mathsf{F}\neg p$.

3. *Branching temporal operators*. These are operators that apply over a set of scenarios.

   - E: The proposition $\mathsf{E}p$ means that p holds *in some scenario* rooted at the present moment. It is the existential quantifier operator over the set of all branches emanating from the present moment.
   - A: The proposition $\mathsf{A}p$ means that p holds *in all scenarios* rooted at the present moment. It is the universal quantifier operator over the set of all branches emanating from the present moment.

   E and A are duals. $\mathsf{E}p \Leftrightarrow \neg\mathsf{A}\neg p$.

2.2 Commitments

Commitments are obligations that one agent has towards another because of a promise. They can, in principle, be legally enforced. Commitments are created because of the social structure of agent communities. The social structure itself may have been formed as a result of other commitments like obligations, rights, and privileges, as Castelfranchi [1993] explains in his "first principles" of commitments. Singh [1999] gives a computational perspective.

2.2.1 Representation and Semantics

A commitment, written C*(d, x, y, p)*, relates a debtor x, a creditor y, and a condition p [Yolum and Singh, 2002b]. The commitment has a unique identifier d. When a commitment is created, x becomes responsible to y for satisfying the condition p; the commitment is said to be satisfied when the condition p holds. There can be at most one commitment with a particular identifier in our entire model.

2.2.2 Commitment Operations

Commitments are created, satisfied, and transformed in certain ways. The following operations that can be performed on commitments were introduced in [Singh, 1999].

1. CREATE*(x, c)* establishes the commitment *c* in the system. This operation can only be performed by the debtor *x* of the commitment.

2. DISCHARGE*(x,c)* satisfies the commitment *c*. It can only be performed by the debtor *x* of the commitment.

3. CANCEL*(x, c)* cancels the commitment *c*. It can only be performed by the debtor *x* of the commitment. Generally, cancellation of a commitment is followed by the creation of another one to compensate for it.

4. RELEASE*(y, c)* releases the debtor in the commitment c from the commitment c. This operation can only be performed by the creditor *y* of the commitment.

5. ASSIGN*(y, z, c)* replaces *y* with *z* as the creditor of the commitment c. This operation can only be performed by the creditor *y* of the commitment.

6. DELEGATE*(x, z, c)* replaces *x* with *z* as the debtor of the commitment c. This operation can only be performed by the debtor *x* of the commitment.

We note that a commitment has to be created using the CREATE operation for it to exist. We also introduce two predicates in Section 3.3 that help us capture the notion of fulfillment of a commitment rigorously.

2.2.3 Commitment Predicates

For every operation on commitments listed in Section 2.2.2 we introduce a corresponding predicate. These predicates are represented by the same name as the operation, with lowercase letters. The predicates are true at the moment where the corresponding operation succeeds. For example, if an agent x invokes a CREATE*(x,c)* operation, and this operation succeeds at a moment m_i, then the predicate *create(x,c)* is said to be true at the moment m_i.

2.2.4 Commitment Identifiers

Every commitment is assumed to have a unique identifier that helps to distinguish it from other commitments that may have the same debtor, the same creditor, and the same condition. For example, if I promise to pay you $5 twice, then a single payment of $5 should not suffice. The predicates in question also apply to specific commitments, i.e., respecting their unique identifiers. For example, I may cancel one of my two commitments to pay $5 without automatically canceling the other commitment. The identifiers come from a domain $\mathbb{D}$, which can be thought of as formed of the natural numbers if one likes.

Chapter 3

Technical Framework

This chapter introduces the concept of time intervals, describes the formal language for our scheme, and introduces two key predicates dealing with the resolution of commitments.

3.1 Temporal Commitments

In real-life settings where commitments are used, commitments often must be satisfied either within a fixed, bounded interval or at a specified instant in the future. A temporal specification that is used to specify commitments should therefore be able to specify time-bounded propositions.

Temporal Quantification

The temporal commitment structure specified by Fornara and Colombetti [2002] forms the basis for our temporal commitment scheme. Every condition specified in the commitment language is potentially specified with the following elements:

- A time interval.
- A temporal quantifier.

- A proposition.

We use timestamps to denote endpoints of time intervals. Two timestamps, like t_l and t_u are used to represent an interval that begins at t_l and ends at t_u, both instants inclusive. For any such time interval, either $t_l < t_u$ or $t_l = t_u$.

An interval provides a set of instants over which to evaluate logical propositions. We introduce the following *temporal quantifiers* to quantify over instants in the interval:

1. $[\,]$ is interpreted as an existential quantifier over a time interval. That is, $[t_1, t_2]p$ means the proposition p has to hold at one or more instants in the interval beginning at t_1 and ending at t_2.

2. $\overline{[\,]}$ is interpreted as a universal quantifier over a time interval. That is, $\overline{[t_1, t_2]}p$ means the proposition p has to hold at every instant in the interval beginning at t_1 and ending at t_2.

The existential and the universal temporal quantifiers are duals of each other.

$$\neg[t_l, t_u]p \Leftrightarrow \overline{[t_l, t_u]}\neg p \tag{3.1}$$

$$\neg\overline{[t_l, t_u]}p \Leftrightarrow [t_l, t_u]\neg p \tag{3.2}$$

Propositions that are temporally quantified are referred to as *temporally quantified propositions*, and commitments which contain such propositions in their conditions are called *temporal commitments*.

3.2 The Formal Language

The following is a grammar for $\mathcal{T}$, the propositional language we use, expressed in Backus-Naur Form (BNF). In this grammar, the uppercase letters denote nonterminals; the lowercase letters denote lexical items that are not analyzed by this grammar; A stands for any agent symbol; $\rightarrow$ is a meta-symbol of the BNF; items enclosed within $\{$ and $\}$ can appear zero or more times; all other symbols are terminals. T is the unique starting symbol for the language of $\mathcal{T}$.

G_1. $T \rightarrow T \wedge T$

G_2. $T \rightarrow a \mid \neg a \mid a \vee T \mid C \mid P \mid Q$

G_3. $T \rightarrow \mathsf{A}L$

G_4. $L \rightarrow [I, I]T \mid \overline{[I, I]}T$

G_5. $L \rightarrow T\mathsf{U}T$

G_6. $C \rightarrow \neg\mathsf{C}(d, A, A, T) \mid \mathsf{C}(d, A, A, T)$

G_7. $Q \rightarrow \mathit{satisfied}(C) \mid \mathit{breached}(C)$

G_8. $P \rightarrow \mathit{cancel}(A, C) \mid \mathit{delegate}(A, A, C) \mid \mathit{assign}(A, A, C) \mid \mathit{release}(A, C) \mid \mathit{create}(A, C)$

In the grammar, a is an atomic proposition in the domain, d is a unique commitment identifier, and I is a timestamp; i.e., $I \in \mathbb{T}$.

The features of $\mathcal{T}$ are:

- $\mathcal{T}$ accepts temporally quantified propositions as well as propositions without temporal quantification (which we shall refer to as simple propositions).

- $\mathcal{T}$ allows arbitrary nesting of time intervals. The implications of such a nesting are discussed in Chapter 4.

3.2.1 Notational Convention for Propositions

As a convention, we shall use

- p and q to denote simple propositions, which may be a single literal or a conjunction or disjunction of literals, where such a literal is either an atomic proposition or a commitment predicate.
- p_t and q_t to denote temporally quantified propositions or temporal commitments, and are collectively referred to as *temporal propositions*.

Any proposition that is denoted by a unique symbol in the language of the domain we are modeling is an atomic proposition. Atomic propositions are assumed to be known by all participants and to be understood by all participants to have the same meaning. For example, p might denote the atomic proposition "The auction house is in open." This is a simple proposition. q might denote "An auction is in progress." This is also a simple proposition. $p \wedge q$ is therefore a simple proposition meaning "The auction house is open *and* there is an auction in progress." $\neg p \vee q$, which is equivalent to $p \Rightarrow q$, is also a simple proposition which means "If the auction house is open, then an auction is in progress."

However, if we quantify an atomic proposition or a simple proposition with a time interval and a temporal quantifier, it becomes a temporally quantified proposition. Continuing with the above examples, $\overline{[t_l, t_u]}\neg p$ is a temporally quantified proposition; it means "At all times from t_l to t_u, the auction house is not open." $[t_l, t_u](p \wedge q)$ is also a temporally quantified proposition; it means "The auction is open and there is an auction in progress at least for one instant between t_l and t_u."

3.3 New Predicates for Commitments

We propose the addition of two new predicates, *breached(c)* and *satisfied(c)*, indicating violation and fulfillment of the given commitment, respectively.

Whenever a commitment is created, it is neither breached nor satisfied. Eventually, it might be breached and thus remain breached forever; or it may be be satisfied and remain satisfied forever. That is, in contrast to the predicates corresponding to commitment operations (as introduced in Section 2.2.2), both *breached* and *satisfied* are stable [Francez and Forman, 1996]. However, a commitment may remain forever possible without being satisfied, in which case it will never be breached nor satisfied. This has the effect of applying a three-valued logic for the satisfaction of commitments.

Section 3.4 defines a language for specifying rules about the behavior of commitments, using the concepts developed here.

3.4 Semantics

We now describe the semantics for the language $\mathcal{T}$.

- For a proposition p, $M \models_m p$ means that a model M satisfies proposition p at moment m. $M \models_{S,m} p$ means that the model M satisfies p at moment m in the scenario S.
- As is customary in formal semantics, we define an *interpretation* $\mathbb{I}$, which labels each moment with the elements that are interpreted as being true at that moment. In our case, the elements are the atomic propositions of the domain along with the elementary facts about commitments, i.e., which operations on commitments are successfully performed at the given moment. In a practical system, these elements would be specified in some manner external to the

logic. For example, the domain atomic propositions might be observed to be true and the commitment-predicates might be mapped to various linguistic or other conventions. For instance, a $create(\cdot,\cdot)$ predicate might be taken to hold when a party submits a form over the Web.

Let Φ be a set containing the domain atomic propositions as well as operations on commitments as generated from the nonterminal T in our grammar. Then $\mathbb{I} : \mathbb{M} \mapsto \mathcal{P}(\Phi)$.

Let $M = \langle \mathbb{A}, \mathbb{M}, \prec, \mathbb{I}, \mathbb{T} \rangle$ be a model for the formal language $\mathcal{T}$, where $\mathbb{M}$, $\mathbb{T}$, and $\prec$ have the meaning as explained in Section 2.1, $\mathbb{A}$ is a set of agent symbols, and $\mathbb{I}$ is an interpretation as defined above.

The semantics of $\mathcal{T}$ is given by the following postulates. For convenience, we define c as $\mathsf{C}(d,x,y,p)$ or $\mathsf{C}(d,x,y,p_t)$ and $active(c)$ as an abbreviation for $\neg(cancel(x,c) \vee delegate(x,\cdot,c) \vee assign(y,\cdot,c) \vee release(y,c) \vee discharge(x,c))$. The meaning of $active(c)$ is that the commitment c has not been operated upon using a commitment operation.

R$_1$. $M \models_m p$ iff $p \in \mathbb{I}(m)$ where $p \in \Phi$

R$_2$. $M \models_m p \wedge q$ iff $M \models_m p$ and $M \models_m q$

R$_3$. $M \models_m p \vee q$ iff $M \models_m p$ or $M \models_m q$

R$_4$. $M \models_m \neg p$ iff $M \not\models_m p$

R$_5$. $M \models_m \mathsf{A}p$ iff $(\forall S : S \in \mathbb{S}_t \Rightarrow M \models_{S,m} p)$

R$_6$. $M \models_m \mathsf{E}p$ iff $(\exists S : S \in \mathbb{S}_t \Rightarrow M \models_{S,m} p)$

R$_7$. $M \models_{S,m} p\mathsf{U}q$ iff $(\exists m_1 \in S : m \preceq m_1$ and $M \models_{S,m_1} q$ and $(\forall m_2 : m \preceq m_2 \preceq m_1 \Rightarrow M \models_{S,m_2} p))$

R$_8$. $M \models_{S,m} [t_l, t_u]p$ iff $(\exists m_l, m_u \in S : \tau(m_l) = t_l$ and $\tau(m_u) = t_u$ and $(\exists m_x : m_l \preceq m_x \preceq m_u$ and $M \models_{m_x} p))$

R$_9$. $M \models_{S,m} \overline{[t_l, t_u]}p$ iff $(\exists m_l, m_u \in S : \tau(m_l) = t_l$ and $\tau(m_u) = t_u$ and $(\forall m_x : m_l \preceq m_x \preceq m_u$ and $M \models_{m_x} p))$

R$_{10}$. $M \models_m \mathit{satisfied}(c)$ iff $(\exists m_3 : m_3 \preceq m$ and $M \models_{m_3} \mathit{discharge}(x, c)$ and $(\exists m_1 : m_1 \prec m_3$ and $M \models_{m_1} \mathit{create}(x, c)$ and($\forall m_2 : m_1 \preceq m_2 \prec m_3 \Rightarrow M \models_{m_2} \mathit{active}(c))))$

R$_{11}$. $M \models_m \mathit{breached}(c)$ iff $(\exists m_3 : m_3 \preceq m$ and $M \models_{m_3} \mathsf{AG}\neg \mathit{discharge}(x, c)$ and $(\exists m_1 : m_1 \preceq m_3$ and $M \models_{m_1} \mathit{create}(x, c)$ and $(\forall m_2 : m_1 \preceq m_2 \preceq m_3 \Rightarrow M \models_{m_2} \mathit{active}(c)$

R$_{12}$. $M \models_m c$ iff $(\exists m_1 : M \models_{m_1} \mathit{create}(x, c)$ and $(\forall m_2 : m_1 \preceq m_2 \preceq m$ and $M \models_{m_2} \mathit{active}(c)))$

Apart from these semantics, we impose the following constraints on the model to capture operations on commitments.

C$_1$. A commitment cannot be created more than once with a given identifier. This captures the uniqueness of commitment identifiers (Section 2.2.4).

$$M \models_m \mathit{create}(x, \mathsf{C}(d, x, y, p_t)) \Rightarrow \\ \forall m_1 : m \prec m_1 (\forall x_1, y_1, p'_t : M \not\models_{m_1} \mathit{create}(x_1, \mathsf{C}(d, x_1, y_1, p'_t))$$

C$_2$. When a commitment is assigned , a new commitment with the new creditor is created with the new creditor, and the old commitment no longer holds. In other words, an ASSIGN leads to a new commitment being created.

$$M \models_m (\mathit{assign}(y, z, c)) \Rightarrow M \models_m (\mathsf{AG}\neg c \wedge \mathit{create}(z, c'))$$

where $c \equiv \mathsf{C}(d, x, y, p_t)$ and $c' \equiv \mathsf{C}(d', x, z, p'_t)$.

C_3. When a commitment is delegated, it is as good as canceled, but a new commitment is created with a different debtor. In other words, a DELEGATE leads to new a commitment being created.

$$M \models_m (delegate(x, z, c)) \Rightarrow M \models_m (\mathsf{AG}\neg c \wedge create(z, c'))$$

where $c \equiv \mathsf{C}(d, x, y, p_t)$ and $c' \equiv \mathsf{C}(d', z, y, p'_t)$.

C_4. When a commitment is created, it is neither breached nor satisfied.

$$M \models_m create(c) \Rightarrow \neg satisfied(c) \text{ and } \neg breached(c)$$

Note that in both the delegation and the assignment rules, c' has a condition p'_t, which might be different from p_t in that the time bounds on the newly created commitment may be different. However, such issues of assignment of new time bounds and how they should be decided are in the realm of the study of responsibility among agents, and are beyond the scope of our work. We discuss this issue in Section 5.

3.5 Commitment Life Cycle

Using the above semantics, we can infer additional results about the life cycle of commitments, especially dealing with the breach or satisfaction of a temporal commitment. First we establish the following simple but important lemmas indicating the stability of the $breach(\cdot)$ and $satisfied(\cdot)$ predicates.

Lemma 1 $M \models_m breached(c)$ *iff* $M \models_m \mathsf{AG} breached(c)$

Lemma 2 $M \models_m \mathit{satisfied}(c)$ *iff* $M \models_m \mathsf{AG}\mathit{satisfied}(c)$

Lemma 3 $M \models_m \neg \mathit{satisfied}(c) \not\Rightarrow M \models_m \mathit{breached}(c)$

Lemma 4 $M \models_m \neg \mathit{breached}(c) \not\Rightarrow M \models_m \mathit{satisfied}(c)$

The following observations are also provably correct with respect to our semantics. Below, p and q represent simple propositions; t_l, t_u, and the t_i's represent instants in time.

A commitment that is canceled does not exist in the system. Recall that because of unique identifiers, a commitment cannot be created again. However, a new commitment with the same debtor, creditor, and proposition as an older commitment can be created, but the logic has nothing to say about it.

Observation 1 $M \models_m (\mathit{cancel}(x, c)) \Rightarrow M \models_m \mathsf{AG}\neg c$ *where* $c \equiv \mathsf{C}(d, x, y, p_t)$.

When a debtor is released from a commitment, that commitment no longer holds.

Observation 2 $M \models_m (\mathit{release}(y, c)) \Rightarrow M \models_m \mathsf{AG}\neg c$ *where* $c \equiv \mathsf{C}(d, x, y, p_t)$.

The semantic rules R_{10}. and R_{11}., and the Lemmas 1 and 2 are diagrammatically given in Figures 3.1 and 3.2.

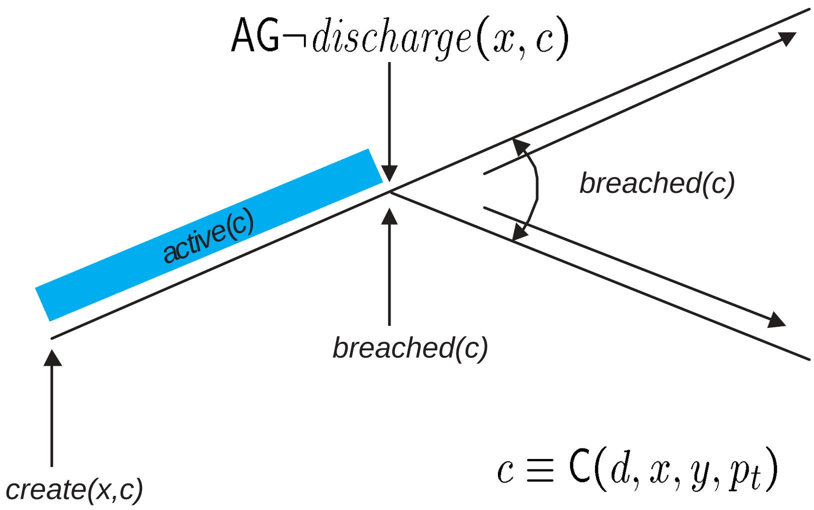

Figure 3.1: Temporal behavior of the $breached(\cdot)$ predicate

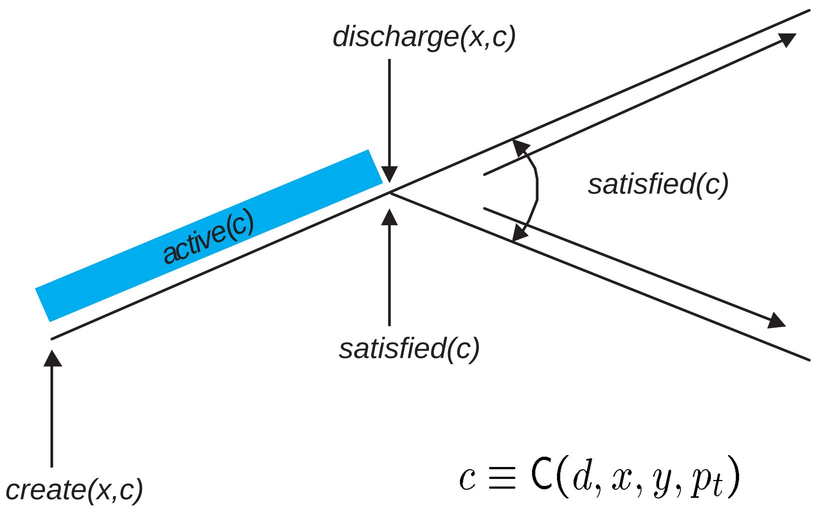

Figure 3.2: Temporal behavior of the $satisfied(\cdot)$ predicate

Chapter 4

Resolving Temporal Commitments

A temporal commitment is resolvable if its satisfaction or breach can be determined at some moment. Under certain conditions, the unresolvability of a temporal commitment can be ascertained even before the specified time interval occurs.

We now discuss some cases where a temporally quantified proposition is not resolvable, and develop methods to detect such cases. Based on the resolvability of such propositions, we can detect satisfaction or breach of commitments that have temporally quantified propositions as their condition.

4.1 Nested Interval Expressions

The temporal propositional language given in Section 3.2 allows for propositions to be nested within multiple levels of time intervals. Although there are many nested intervals whose interpretation in common language does not make sense or induces redundancy, some nested time intervals do make sense in real-life situations. We give examples of both meaningful and meaningless nested interval propositions in this section.

Consider the proposition $[t_1, t_2](\overline{[t_3, t_4]}p)$, where $t_1 < t_3 < t_4 < t_2$ (case 4.1, Figure 4.1). This proposition means that p will hold at all moments in the interval $\overline{[t_3, t_4]}$, and this interval will occur at least once within the time interval $[t_1, t_2]$. This proposition could therefore be used to express events that take a finite, continuous amount of time at every occurrence within an interval, like the event mentioned in Example 1 in Chapter 1.

Example 1 mentions that the auction house wants to assure its buyers that there will be at least one auction held this week.

Solution 1 $[t_1, t_2]$ would denote this week, $\overline{[t_3, t_4]}$ would denote the duration of one auction, and p would denote "An auction in progress." Hence "At least one auction this week" could be represented by $([t_1, t_2](\overline{[t_3, t_4]}p))$, and the commitment made by the auction house to its buyers could be represented as $\mathsf{C}(d, Au, b, [t_1, t_2](\overline{[t_3, t_4]}p))$, where Au represents the auction house, and b represents a buyer. ▮

Note that our current grammar does not say anything about specifying timestamps relative to one another; i.e., specifying relationship between timestamps within a formula formed according to the rules of our grammar. Such an extension to the grammar would be very beneficial in specifying statements like "sometime before an event", but would be computationally expensive. This issue is discussed in Section 5.

Allen [1983] defines 13 possible temporal relationships between any two given time intervals. Figure 4.1 shows a visual interpretation of of these 13 relationships for the two intervals contained in the temporally quantified proposition $[t_1, t_2](\overline{[t_3, t_4]}q)$; i.e., the intervals $[t_1, t_2]$ and $\overline{[t_3, t_4]}$. In the figure, time moves forward (increases) from left to right. The shaded portions are intervals of the type $\overline{[t_l, t_u]}$, and the unshaded portions are of the type $[t_l, t_u]$. 13 such diagrams can be constructed *for each combination* of temporal quantifiers applied to each of the intervals, but we show only one interval-quantifier combination pair as an example.

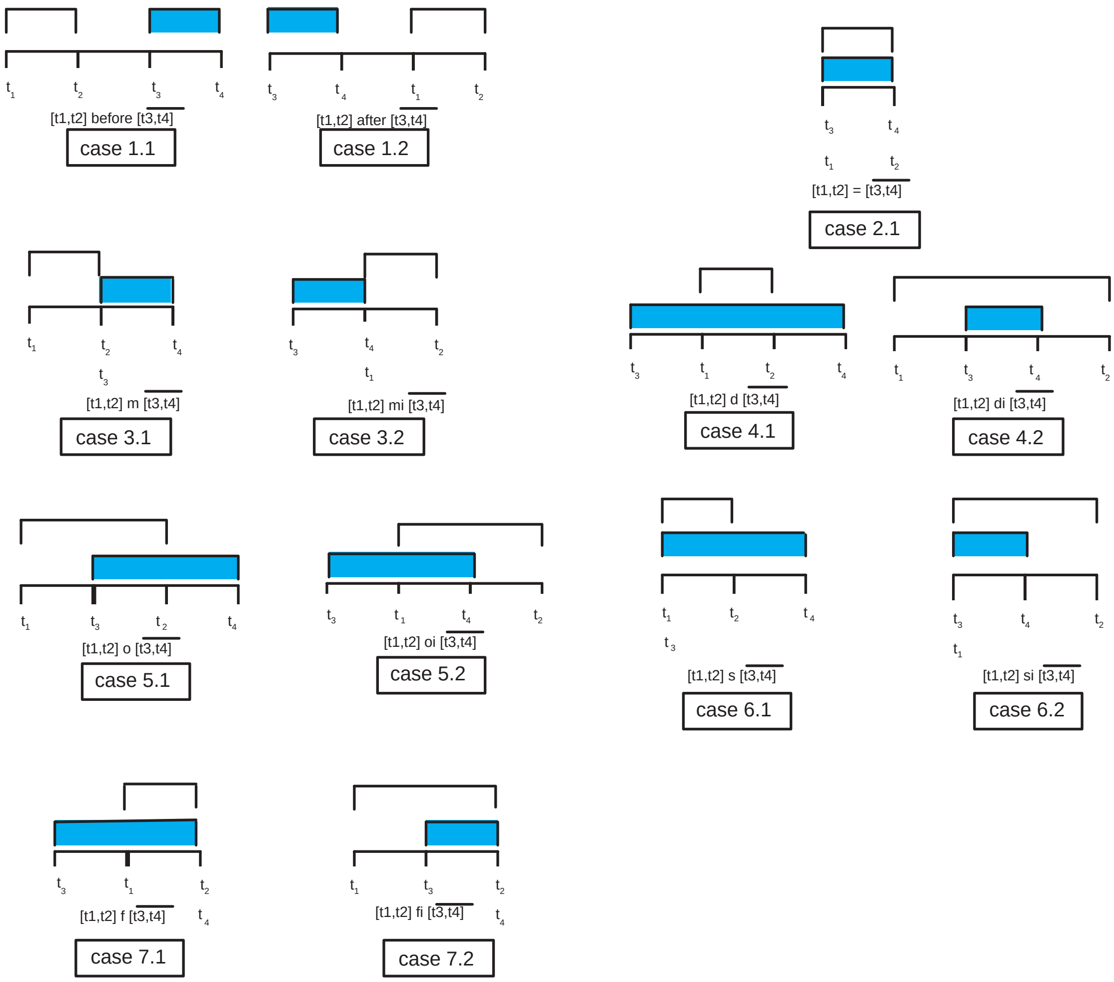

Figure 4.1: Allen's intervals for $[t_1, t_2](\overline{[t_3, t_4]}p)$

If we consider nested temporally quantified propositions as being conditions of commitments, we can see which kinds of nesting will make the success of the commitment unresolvable. Cases 1.1, 3.1, 4.1, 5.1, and 6.1 are not resolvable, but cases 1.2, 2.1, 3.2, 4.2, 5.2, 6.2, 7.1, and 7.2 can be resolved for reasons listed below. The term *inner proposition* is used to refer to the temporal proposition $\overline{[t_3, t_4]}p$.

- In cases 1.1, 3.1, 4.1, 5.1, and 6.1 the inner proposition's time interval does not complete until after the the outer time interval completes. The inner interval has references to instants in the future. Since the future cannot be seen in advance, these cases cannot be resolved.

- In cases 1.2, 2.1, 3.2, 4.2, 5.2, 6.2, 7.1, and 7.2 the inner proposition's success can be resolved at at least one instant within the interval of the outer proposition. Since the outer proposition has an existentially quantified time interval $[t_1, t_2]$ and we have at least one instant of resolvability, these cases can be resolved.

We now formalize the notions of nested intervals and results about their resolution. These results can be used to detect some commitment protocol violations before the actual violation occurs due to unresolvability.

Definition 1 A temporally quantified proposition is said to be positive-resolvable at an instant if its value is known to be true at that instant; it is said to be negative-resolvable at an instant if its value is known to be false at that instant. ▮

Definition 2 A temporal commitment is said to be positive-resolvable at an instant if its satisfaction can be known at that instant; it is said to be negative-resolvable at an instant if its breach can be known at that instant. ▮

Definition 2 means that if one of $satisfied$ or $breached$ predicates of a temporal commitment is true, then the commitment has been resolved. Expressed in the negative, this definition means that a temporal commitment is unresolved at an instant if both the $satisfied$ and the $breached$ predicates for that commitment are false at that instant.

We use the following notation to denote some important instants with respect to an interval. Below, p_t is a temporal proposition, and r is used to denote resolvability.

- $r_l^+(p_t)$ represents the earliest instant at which p_t is positive-resolvable.
- $r_u^+(p_t)$ represents the latest instant at which p_t is positive-resolvable.

- $r_l^-(p_t)$ represents the earliest instant at which p_t is negative-resolvable.
- $r_u^-(p_t)$ represents the latest instant at which p_t is positive-resolvable.

We assume that the past is observable because it has already occurred, and the future is not.

The following lemmas state results about earliest and latest bounds respectively, of the detection of resolvability for a simple proposition p. These bounds form the base case for calculating resolvability of propositions that have intervals nested to any arbitrary depth and the resolvability of temporal commitments.

Lemma 5 $r_l^+([t_l, t_u]p) = t_l,\ r_u^+([t_l, t_u]p) = t_u$

Lemma 6 $r_l^-([t_l, t_u]p) = t_u,\ r_u^-([t_l, t_u]p) = t_u$

Lemma 7 $r_l^+(\overline{[t_l, t_u]}p) = t_u,\ r_u^+(\overline{[t_l, t_u]}p) = t_u$

Lemma 8 $r_l^-(\overline{[t_l, t_u]}p) = t_l,\ r_u^-(\overline{[t_l, t_u]}p) = t_u$

Table 4.1 and Table 4.2 summarize the above lemmas.

Interval	Earliest breached at (r_l^-)	Earliest satisfied at (r_l^+)
$[t_l, t_u]$	t_u	t_l
$\overline{[t_l, t_u]}$	t_l	t_u

Table 4.1: The earliest instant of resolvability of a temporally quantified proposition

We make the following observations using Table 4.1 and Table 4.2, wher p_t is a temporally quantified proposition:

- p_t is not positive-resolvable at any instant before $r_l^+(p_t)$.
- p_t is not negative-resolvable at any instant before $r_l^-(p_t)$.
- p_t *is* positive-resolvable at any instant after $r_u^+(p_t)$.

Interval	Latest breached at (r_u^-)	Latest satisfied at (r_u^+)
$[t_l, t_u]$	t_u	t_u
$\overline{[t_l, t_u]}$	t_u	t_u

Table 4.2: The latest instant of resolvability of a temporally quantified proposition

- p_t *is* negative-resolvable at any instant after $r_u^-(p_t)$.

Figures 4.2 and 4.3 are a diagrammatic representations of the bounds on resolvability of a proposition with a universally quantified interval and an existentially quantified interval respectively.

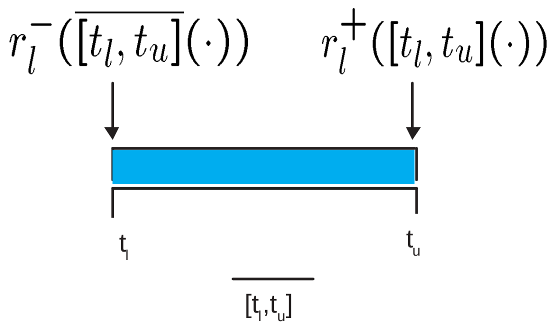

Figure 4.2: Bounds on resolvability of $\overline{[t_l, t_u]}(\cdot)$

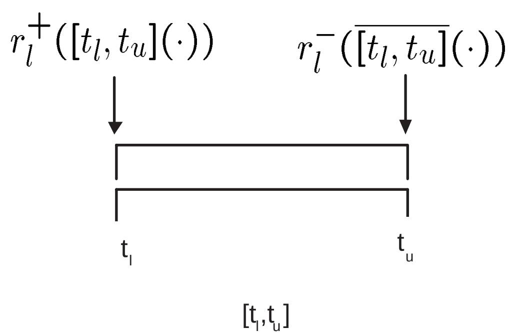

Figure 4.3: Bounds on resolvability of $[t_l, t_u](\cdot)$

Next, we utilize the theory about resolution of temporally quantified propositions that we have developed to solve some of the challenges that were outlined earlier.

4.2 Resolving Nested Interval Expressions

Using the rules in Section 4.1, we can now see why some of the two-level interval nesting cases shown in Figure 4.1 were determined to be unresolvable.

In cases 1.1, 4.1, and 5.1, we cannot determine the if the proposition $\overline{[t_3, t_4]}q$ is true within the outer interval $[t_1, t_2]$. More specifically, the earliest instant at which the satisfaction of $\overline{[t_3, t_4]}q$ can be determined is t_4, which is beyond t_2, the latest instant for the satisfaction of $[t_1, t_2]p$. As a consequence, the expression $[t_1, t_2](\overline{[t_3, t_4]}q)$ as a whole cannot be resolved, which is why commitments whose conditions are temporally quantified propositions of these types are disadvantageous for the creditor.

We now use the concepts of earliest and latest instants of resolvability to solve Example 2 of Chapter 1. This example mentions that the auction house promises to bring about the condition that there will be a continuous auction on the upcoming weekend. However, the auction house promises to bring about the condition today (a Tuesday).

Solution 2 To model Example 2, the auction house Au makes a commitment to a set of buyers b. The commitment is $\mathsf{C}(d, Au, b, [t_1, t_2](\overline{[t_3, t_4]}q)$, where $t_2 < t_3$, $[t_1, t_2]$ denotes the interval "today" (a Tuesday), $\overline{[t_3, t_4]}$ denotes the interval "this weekend", and q is an atomic proposition that means "There will be a continuous auction". The condition of this commitment corresponds to case 1.1 of Figure 4.1. In this case, $\overline{[t_3, t_4]}q$ cannot be resolved at least until t_4, and $[t_1, t_2](\cdot)$ *has* to be resolved at most by t_2. But since $t_2 < t_4$, this condition cannot be resolved. Hence the commitment cannot be satisfied. Formally, $r_u^+([t_1, t_2](\cdot)) < r_l^+(\overline{[t_3, t_4]}q)$. ▮

As a final example of an unresolvable commitment, consider a case $\overline{[t_1, t_2]}(\overline{[t_3, t_4]}q)$, as shown in Figure 4.4 where $t_3 < t_1 < t_2 < t_4$. The outer interval is not resolvable, since the inner proposition cannot be resolved at all instants in it. Formally, $r_u^+(\overline{[t_1, t_2]}(\cdot)) < r_l^+(\overline{[t_3, t_4]}q)$.

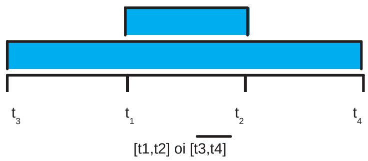

Figure 4.4: A schematic for $\overline{[t_1, t_2]}(\overline{[t_3, t_4]}p)$, where $t_2 < t_3$

To summarize, the following conditions are necessary to ensure resolvability of a temporally quantified proposition.

- A temporally quantified proposition of the form $[t_l, t_u]p_t$ must have at least one instant in the interval t_l, t_u, at which p_t it is resolvable
- A temporally quantified proposition of the form $\overline{[t_l, t_u]}p_t$ must have p_t resolvable at all instants in the interval t_l, t_u

Since we classify commitments as either breached or satisfied based on the intervals in their condition, the following lemmas hold:

Lemma 9 $M \models_m create(x, c) \Rightarrow (\forall S : M \models_{S,m} \mathsf{F} satisfied(c) \Rightarrow M \models_{S,m} \mathsf{F} \neg active(c))$

Lemma 10 $M \models_m create(x, c) \Rightarrow M \models_m \mathsf{AF}(\neg active(c) \vee breached(c))$

where $c \equiv \mathsf{C}(d, x, y, p_t)$.

We have shown how unresolvable commitments can be detected. In cases where an unresolvable commitment is as good (or as bad) as one that is breached, such resolution results will enable earlier detection of protocol violations, and is of practical importance in situations where prevention is better than cure.

4.3 Disjunctive Forms

Another important aspect of commitment resolution concerns commitments whose conditions are disjunctions of temporally-quantified propositions, which can also be modeled as implications or conditional commitments. We call such commitments *disjunctive commitments*.

Disjunctive commitments regularly arise in common business interactions and can sometimes lead to what we call *the warranty paradox,* where a disjunctive temporal commitment turns out to be favorable to the debtor and unfavorable to the creditor. More specifically, the warranty paradox is a situation where a merchant provides a warranty on some good she sells, but also specifies some clauses, which render the warranty void before the customer can ascertain the quality of the good. This can happen if ascertaining the quality of the good takes more time than the time over which the warranty for that good is valid.

The debtor of a disjunctive commitment is committed to satisfying at least one of the disjuncts. But the different bounds on positive- and negative-resolution of these disjuncts can lead to an unfavorable situation for the creditor of the commitment, as in Example 3.

Intuitively, we reason as follows about the satisfiability of a disjunction of temporal propositions: *A disjunction of temporal propositions is satisfiable if it has not already been satisfied, and at least one of the disjuncts is still resolvable*.

Let $L_1 \vee L_2 \vee \ldots \vee L_N$ denote the disjunction of temporally quantified propositions, in which t_{early} denotes the earliest instant among all instants contained in the intervals of the L_i's, and t_{late} denotes the latest. Then, for the disjunction to be satisfiable, the requirement is:

$$\forall m_x, m_{early} \preceq m_x \preceq m_{late}$$

$$\forall i : M \models_{m_x} \neg satisfied(L_i) \Rightarrow$$

$$\exists j : (M \models_{m_x} \neg breached(L_j)) \wedge (\tau(m_x) < r_u^+(L_j)) \tag{4.1}$$

where, $\tau(m_{early}) = t_{early}$, and $\tau(m_{late}) = t_{late}$.

Let us apply Equation 4.1 to Example 3 from Chapter 1. This example describes a scenario where the auctioneer who sold the fish guarantees that it will remain fresh for two days or the buyers will be given their money back. However, the fish market will close operations in two hours, after which the auctioneers commitments made in the fish market would be invalid.

Solution 3 Example 3 can be modeled by the commitment $\mathsf{C}(d, au, b, (\overline{[t_3, t_4]}\, \mathit{fresh}(\mathit{fish}) \vee [t_1, t_2] payback))$, where the atomic proposition "fresh (fish)" means the the fish are fresh, the atomic proposition "$payback$" represents the warranty that the auctioneer gives by saying she will refund the money if the fish rot before two days, au represents the auctioneer, and b represents the buyer. We see that there exists a moment in the set of all moments of the disjuncts, such that every literal is either breached or is beyond the upper bound of its positive-resolution. If the fish rots three hours after it was bought, then the only proposition that is not yet breached at that point is the payback guarantee by the auctioneer. However, the upper bound of positive resolvability of this proposition has passed. More formally,

$$\exists m_x, m_{early} \preceq m_x \preceq m_{late} :$$
$$\forall i : (M \models_{m_x} breached(L_i)) \vee (\tau(m_x) < r_u^+(L_i))$$

∎

Hence the warranty that the auctioneer provided on the freshness of the fish is unfavorable for the customer, since the warranty becomes invalid before the buyer can ascertain the freshness of the fish. Thus we show how the warranty paradox can be captured in our scheme of temporal commitments.

We provide one more example to clarify the concept of the warranty paradox.

Example 4 An electronics manufacturer sells computers with a 90-day money back warranty on the parts. However, if the company goes bankrupt within a month after a customer bought a computer, then the warranty is void because the company has no money to pay anyone who makes a claim. ▮

However, the violation of the warranty in this example cannot be detected beforehand because the bankruptcy of the company cannot be foreseen.

We have shown in this chapter how the three examples given in Chapter 1, which cannot be handled by commitments without deadlines, can be represented and reasoned about in our scheme. We have provided examples which illustrate how our scheme decouples the actual content of propositions from their temporal implications by specifying time intervals explicitly. This decoupling allows us to reason about temporal aspects of commitments without regard to the atomic propositions used.

Chapter 5

Discussion

The concept of deadlines in commitments is doubtless necessary for practical uses of commitments. Traditionally, deadlines are not formally modeled and are hidden within the atomic propositions. However, an explicit formulation of temporal commitments, as developed above is highly desirable. It offers a uniform treatment of operational characteristics across domains. We have shown how a such a system of commitments with deadlines can be developed, and used to reason about the possibility of satisfaction of commitments. Our approach not only allows for expression of statements that involve temporal anaphora, but also decouples the temporal quantification from the proposition itself, thus allowing us to reason about the temporal aspect, without regard to the meaning of the propositions. Our scheme for temporal quantification of propositions is related to the notion of temporal quantification presented by Fornara and Colombetti [2002], but develops interesting results about the resolution of fulfillment of commitments.

Grosof *et al.* develop semantics for systems that represent business rules using *Courteous Logic Programs* [1999]. Their approach uses explicit rules to use to decide between conflicting rules. However, temporal constraints are specified as part of the language, not as a separate concept. Such a specification is too simplistic and *ad hoc*, and does not allow reasoning about the temporal aspects independent of the domain specific details.

Minsky and Ungureanu [2000] propose *Law Governed Interaction*, which is a way of specifying laws about rules and hierarchies of rules, so priorities between rules can be resolved. This approach too does not emphasize temporal aspects.

Dignum *et al.* [1996] describe a temporal deontic logic that helps specify obligations and constraints. Their logic is based on obligations, which are similar in spirit to commitments. The logic has one generalized obligation, which is used to derive four specific type of obligations: One to perform some action with respect to the begin and end conditions of an interval, one to perform an action depending on another, one to perform some action immediately and one to periodically perform some action. Their theory of obligations, however, considers only actions of unit length. Maintenance conditions for states of the world cannot be handled. Our approach, on the other hand, can not only specify all the above kinds of obligation as commitments, but also model actions of variable duration.

Research Directions

Our work on temporal aspects of commitments is far from complete. We have identified three important related topics that need to be explored to extend and improve upon this work.

Potential Causality

One direction is to marry this approach with the work on compliance checking based on potential causality as developed by Venkatraman and Singh [1999]. Their work develops a theory for verifying compliance of an agent to a protocol which is modeled by commitments. Commitments are created because of messages passed between agents, and all messages in a protocol can be ordered with respect to an observer because messages are tagged with vector timestamps. A local model of messages is built by an agent, and this model is used to check if all commitments made to the agent have been fulfilled.

Our results on resolution of commitments and unenforceable commitments applies at a finer granularity than theirs, since we do not work with protocols, but only with individual commitments. Also, their work does not use the concept of deadlines on commitments. Compliance verification in agent interaction protocols could be done more effectively if a fusion of our work with Venkataraman and Singh's is achieved.

Relativized Timestamps

Our current specification of the grammar for $\mathcal{T}$ does not allow us to specify timestamps relative to one another. That is, we cannot say $([t_1, t_2]p) \wedge ([t_3, t_4]q) \wedge (t_1 < t_3) \wedge (t_2 < t_4)$. Such a statement would be of value in specifying statements that involve an event that ends before some other event.

Adding the property of specifying relative time intervals to our grammar would be simple but such a grammar would be computationally expensive. We leave this as future work.

Delegation and Responsibility

Another direction for further research is to extend our scheme to accommodate concepts of responsibility and trust to clearly define the semantics of commitment operations such as DELEGATE and ASSIGN.

Realistic delegations of tasks oftentimes require that the delegator be accountable for some or all of the consequences of the actions that the delegatee performs to accomplish the task that was delegated.

Although we have specified rules about the effect of the DELEGATE and the ASSIGN operations on the life-cycle of a commitment in Section 3.5, canceling a commitment when it is delegated is a naive approach. A clear concept of responsibility and its flow with the delegation or assignment of a commitment needs to be worked out, so that the model can represent richer agent interactions and model more complex social structures effectively.

Bibliography

James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *Proceedings of the AAAI-93 Workshop on AI and Theories of Groups and Organizations: Conceptual and Empirical Research*, 1993.

Frank Dignum, Hans Weigand, and Egon Verharen. Meeting the deadline: On the formal specification of temporal deontic constraints. In *International Syposium on Methodologies for Intelligent Systems*, pages 243–252, 1996. URL `citeseer.nj.nec.com/dignum96meeting.html`.

E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. North-Holland, Amsterdam, 1990.

Nicoletta Fornara and Marco Colombetti. Operational specification of a commitment-based agent communication language. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 535–542. ACM Press, 2002.

Nissim Francez and Ira R. Forman. *Interacting Processes: A Multiparty Approach to*

Coordinated Distributed Programming. ACM Press and Addison-Wesley, New York, 1996.

Benjamin N. Grosof, Yannis Labrou, and Hoi Y. Chan. A declarative approach to business rules in contracts: Courteous logic programs in xml. In Michael P. Wellman, editor, *Proceedings 1st Annual ACM Conf. on Electronic Commerce, EC'99, Denver, CO, USA, 3–5 November 1999*. ACM Press, 1999.

Naftaly H. Minsky and Victoria Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 9(3):273–305, 2000.

Barbara Partee. Nominal and temporal anaphora. *Linguistics and Philosophy*, 7(3):287–324, 1984.

Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia, and Carles Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.

Munindar P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

Mahadevan Venkatraman and Munindar P. Singh. Verifying compliance with commitment protocols: Enabling open Web-based multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, September 1999.

Pınar Yolum and Munindar P. Singh. Commitment machines. In *Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL-01)*. Springer-Verlag, 2002a.

Pınar Yolum and Munindar P. Singh. Flexible protocol specification and execution: Applying event calculus planning using commitments. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. ACM Press, July 2002b.