Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-92-10

An Interval-based Temporal Logic in a Multivalued Setting

Mathias Bauer

July 1992

Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Philips, SEMA Group Systems, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

DFKI Documents

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
mCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz:
Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

D-92-01
Stefan Bussmann: Simulation Environment for Multi-Agent Worlds - Benutzeranleitung
50 Seiten

D-92-02
Wolfgang Maaß: Constraint-basierte Plazierung in multimodalen Dokumenten am Beispiel des Layout-Managers in WIP
111 Seiten

D-92-03
Wolfgan Maaß, Thomas Schiffmann, Dudung Soetopo, Winfried Graf: LAYLAB: Ein System zur automatischen Plazierung von Text-Bild-Kombinationen in multimodalen Dokumenten
41 Seiten

D-92-06
Hans Werner Höper: Systematik zur Beschreibung von Werkstücken in der Terminologie der Featuresprache
392 Seiten

D-92-07
Susanne Biundo, Franz Schmalhofer (Eds.):
Proceedings of the DFKI Workshop on Planning
65 pages

D-92-08
Jochen Heinsohn, Bernhard Hollunder (Eds.):
DFKI Workshop on Taxonomic Reasoning Proceedings
56 pages

D-92-09
Gernod P. Laufkötter: Implementierungsmöglichkeiten der integrativen Wissensakquisitionsmethode des ARC-TEC-Projektes
86 Seiten

D-92-10
Jakob Mauss: Ein heuristisch gesteuerter Chart-Parser für attributierte Graph-Grammatiken
87 Seiten

D-92-12
Otto Kühn, Franz Schmalhofer, Gabriele Schmidt:
Integrated Knowledge Acquisition for Lathe Production Planning: a Picture Gallery (Integrierte Wissensakquisition zur Fertigungsplanung für Drehteile: eine Bildergalerie)
27 pages

D-92-13
Holger Peine: An Investigation of the Applicability of Terminological Reasoning to Application-Independent Software-Analysis
55 pages

D-92-15
DFKI Wissenschaftlich-Technischer Jahresbericht 1991
130 Seiten

D-92-16
Judith Engelkamp (Hrsg.)
Verzeichnis von Softwarekomponenten für natürlichsprachliche Systeme
189 Seiten

D-92-17
Elisabeth André, Robin Cohen, Winfried Graf, Bob Kass, Cécile Paris, Wolfgang Wahlster (Eds.):
UM92: Third International Workshop on User Modeling, Proceedings
254 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-92-21
Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

An Interval-based Temporal Logic in a Multivalued Setting

Mathias Bauer
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3,
W – 6600 Saarbrücken
GERMANY
bauer@dfki.uni-sb.de

Abstract

We describe the embedding of the semantic notions and modal operators of a first-order temporal logic based on time intervals in a multivalued setting. Truth values will be realized as functions from time intervals to "ordinary" truth values like t and f. The main emphasis lies on the realization of the various modal operators contained in the temporal logic as operations on the functional truth values. We show that it is possible to obtain an efficient system sufficient for tasks in the area of diagnostic reasoning.

Contents

1 Introduction

Reasoning about a changing world requires mechanisms going beyond the scope of classical predicate logic. Thus, several temporal logics have been proposed to solve problems in areas like hardware and software verification (e.g., [13]), planning ([2]), reasoning about actions (e.g., [9], [15]), and plan recognition (e.g., [4]).

Among these logics those equipped with operators supporting a compositional approach (e.g., [14], [10]) gained particular importance because they allow some kind of modular reasoning in which formulas can be combined to statements about temporally more complex situations. The operator enabling this kind of reasoning is often referred to as *chop* ($\mathcal{C}$). The semantics of logics containing it is usually based on the notion of *intervals* as sequences of states in contrast to logics like *tense logic* (cf. [5]) based on time *points* and Allen's temporal logic with intervals that are not built up from single states (cf. [1]).

Unfortunately, decision procedures for propositional temporal logics as described in [14] are non-elementary, i.e., of exponential height in the nesting depth of the chop operator. If the demand for completeness is relaxed, however, it is possible to implement inference systems of clearly smaller complexity for tasks in the field of diagnostic reasoning (e.g., [9], [4]).

The system MVL described in [6] suggests itself as a basis for such an implementation as it provides a proof system to which modal operators like *chop* can be added (cf. [8]). The crucial idea of MVL (which stands for "Multivalued Logics") is to keep the inference machine and the "bookkeeping" about truth values separated from each other. Thus, exchanging the set of possible truth values while retaining the prover results in a system for a totally different logic.

The aim of this paper is to describe the theoretical foundations of an implementation of FTL, a first-order version of the temporal logic introduced in [14], in the MVL setting and its realization, possible applications, and limitations. The reasons for embedding this logic in MVL are twofold: On the one hand, we will see how MVL generalizes the usual concepts of Kripke-style modal operators and sets of truth values underlying a certain logic (cf. section 3.3). On the other hand, this enables us to give an efficient implementation of a restricted inference machine for FTL (cf. section 3.4).

Sections 2 and 3.1 will introduce syntax and semantics of our temporal logic and the foundations of the MVL system, resp. In sections 3.2 through 3.4, we will describe how to embed our interval-based logic and its modal operators in MVL. As truth values we will use functions from time intervals to "ordinary" values including t and f and demonstrate how complex modal operators can be realized as operations on these functions. We will also address the problem of how to represent the set of intervals and the truth functions so that an effective computation is possible. In section 3.5, the computational overhead caused by the modal operators is shown to play a minor role concerning the complexity of the whole system. Finally, we will consider limitations and possible applications of the resulting system.

2 The Temporal Logic FTL

The temporal logic FTL ("First-order Temporal Logic") presented in this section essentially corresponds to the extension of the system PTL(U,X,C) as described in [14] to first-order logic.

2.1 Syntax

Given a denumerable set X of variables and a signature S, the set of formulas of FTL comprises T, F, and the usual set of first-order formulas with quantifiers $\forall$ and $\exists$ and the connectives $\neg, \wedge, \vee, \rightarrow$, and $\leftrightarrow$ over S and X. Besides, it contains all formulas of the form $\bigcirc p$ (*"next"*), $(p\ \mathcal{U}\ q)$ (*"until"*), and $(p\ \mathcal{C}\ q)$ (*"chop"*). By Φ_0 we denote the set of atomic first-order formulas. The versions of *next* and *chop* presented here are often referred to as "strong next" and "strong chop".

Before presenting the formal semantics of FTL, we give an intuitive description of the meaning of the modal formulas introduced above: We want to consider a formula $\bigcirc p$ true in an interval σ if p is true if we consider the situation one state later, i.e., if p is true in the interval obtained from σ by removing its first state.

We say $(p\ \mathcal{U}\ q)$ holds in an interval σ if q holds sometime within σ and p holds *all of the time before* within σ.

The chop operator $\mathcal{C}$ provides a possibility to *compose* two formulas p and q by concatenating the intervals in which they hold. Considered differently, chop allows to *split up* an interval σ in which $(p\ \mathcal{C}\ q)$ holds into two subintervals σ_1 and σ_2 where p and q hold, resp.

2.2 Semantics of FTL

Definition 2.1 Any subset of Φ_0 is called a *state*. Let Σ be the power set 2^{Φ_0} of Φ_0. Then the elements of $\mathcal{I} = \Sigma^+ \cup \Sigma^\omega$ are called *intervals*.

The idea behind this definition is that a state contains just those atomic formulas true at a certain moment in time. Intervals are non-empty sequences of states and thus model the truth or falsity of formulas over time.
We need some operations on intervals:

Definition 2.2 Let $\sigma, \sigma_1, \sigma_2$ be intervals. Then the *length* of σ is defined by

$$|\sigma| = \begin{cases} \omega, & \text{if } \sigma \text{ is infinite} \\ n, & \text{if } \sigma = \langle S_0, ..., S_n \rangle \end{cases}$$

The *composition* of σ_1 and σ_2 is

$$\sigma_1 \oplus \sigma_2 = \begin{cases} \sigma_1, & \text{if } |\sigma_1| = \omega \\ \langle S_0, ..., S_n, S_{n+1}, ... \rangle, & \text{if } \sigma_1 = \langle S_0, ..., S_n \rangle \text{ and } \sigma_2 = \langle S_n, S_{n+1}, ... \rangle \end{cases}$$

The *nth suffix* of $\sigma = \langle S_0, ... \rangle$ is $\sigma^{(n)} = \langle S_n, ... \rangle$.

Relating FTL to classical modal logics with Kripke-style semantics, we can regard intervals as possible worlds. The accessibility relation between worlds can be described by terminating subintervals defined below:

Definition 2.3 Let σ_1, σ_2 be intervals. Then we define $R_t \subseteq \mathcal{I} \times \mathcal{I}$ by

$$\sigma_1 \; R_t \; \sigma_2 \;\; :\Longleftrightarrow \;\; \sigma_1 = \sigma_2^{(1)},$$

and call the first suffix σ_1 of σ_2 *the first terminating subinterval* of σ_2.
We denote the transitive closure of R_t by R_t^+ and the reflexive, transitive closure of R_t by $\overline{R}_t$.
If $\sigma_1 \overline{R}_t \sigma_2$, then there is a unique interval σ' such that $\sigma_2 = \sigma' \oplus \sigma_1$. This interval is the *complement* of σ_1 in σ_2 and is denoted by $compl_{\sigma_2}(\sigma_1)$.

So, a world σ_1 is accessible from σ if σ_1 is a terminating subinterval of σ. Now we are ready to define the notion of satisfiability.

Definition 2.4 Let D denote the non-empty set over which we interpret the logical variables of X. Then $\rho : X \to D$ is called an *assignment.* Let $\sigma = \langle\, S_0, \ldots \rangle \in \mathcal{I}$ be an interval, $x \in X$ a variable, $P \in \Phi_0$ an atomic first-order formula, p, q FTL formulas. Then we have

$$
\begin{array}{lcl}
\sigma \models T & & \\
\sigma \not\models F & & \\
\sigma \models P & \Longleftrightarrow & P \in S_0 \\
\sigma \models \neg p & \Longleftrightarrow & \sigma \not\models p \\
\sigma \models p \wedge q & \Longleftrightarrow & \sigma \models p \text{ and } \sigma \models q \\
\sigma \models \exists x.p & \Longleftrightarrow & \text{there is an assignment } \rho \text{ such that } \sigma \models p^x_{\rho(x)} \\
\sigma \models \bigcirc p & \Longleftrightarrow & \sigma^{(1)} \models p \\
\sigma \models (p \;\mathcal{U}\; q) & \Longleftrightarrow & \text{there is } \sigma'', \sigma'' \overline{R}_t \sigma, \text{ such that } \sigma'' \models q \text{ and for all } \sigma' : \\
 & & \text{if } \sigma' \overline{R}_t \sigma \text{ and } \sigma'' R_t^+ \sigma', \text{ then } \sigma' \models p \\
\sigma \models (p \;\mathcal{C}\; q) & \Longleftrightarrow & \text{there are } \sigma', \sigma'' \text{ such that } |\sigma'| < \omega, \sigma' \models p, \sigma'' \models q, \\
 & & \text{and } \sigma = \sigma' \oplus \sigma''
\end{array}
$$

Concerning the other connectives and the universal quantifier, we use the usual recursive definitions.

It should be noted that the truth value of a non-modal formula only depends on the first state of an interval.

On the basis of the operators defined so far, we can derive other modalities useful in temporal reasoning. Examples are

$$\Diamond p \;\; :\equiv \;\; (T \;\mathcal{U}\; p)$$

and its dual

$$\Box p \;\; :\equiv \;\; \neg\Diamond\neg p\,.$$

These operators are called *sometimes* and *always*. In "classical" modal logics, they correspond to the modalities *possibly* and *necessarily*. Introducing the abbreviation

$$empty \;\; :\equiv \;\; \neg\bigcirc T$$

to denote the end of an interval, we can derive the so-called *weak* versions of chop and next:

$$(p\ \mathcal{C}\ q) \quad :\equiv \quad ((p\ \mathcal{C}\ q) \vee (p \wedge \Box\neg empty))$$

$$\odot p \quad :\equiv \quad empty \vee \bigcirc p\,.$$

3 Review of MVL and the Embedding of FTL

The MVL system by Ginsberg is an attempt to capture many sorts of reasoning within the field of artificial intelligence in a uniform framework (cf. [6]). The basic idea is to split up inference into two parts: One in which the actual process of reasoning takes place – realized by a theorem prover – and one in which some kind of "bookkeeping" of the results obtained from the inference machine is done.

As an example, one might imagine a system for probabilistic reasoning where the bookkeeping consists of combining the numerical values assigned to the formulas used and pruning formulas whose probability is below a certain threshold. Other examples given by Ginsberg are ATMS and default reasoning systems.

Ginsberg formalizes the bookkeeping part of reasoning by attaching two kinds of labels to each bit of information: one describing the amount of knowledge available about a certain statement and one indicating the degree of certainty about its validity. On the basis of these labels, sets of truth values can be given the internal structure of a so-called *bilattice*, which is advantageous with regard to several aspects:

1. **Modularity.** It is possible to develop theorem provers suitable for many different object logics independent of the actual choice of underlying truth values, since the definition of a bilattice forms a unique interface to the bookkeeping part of the inference machine. Selecting a new set of possible truth values yields a reasoner for a totally different logic although the original prover is further used (cf. [7] and [8]).

2. **Efficiency.** As we will see, it is possible to exploit the additional information represented in the bilattice structure during the inference process to render it more efficient.

3. **Modal operators.** It is easy to introduce new modal operators into a given logic, as they can essentially be expressed using primitive operations on the elements of a bilattice. Besides forming the basis for efficient implementations, this is also interesting from a theoretical point of view, as this approach generalizes both the classical concept of Kripke-style modal operators and Moore's autoepistemic operator L (cf. [12]). Thus, we are able to introduce modal operators of arity > 1 (in fact, the FTL operators $\mathcal{U}$ and $\mathcal{C}$ exceed Kripke's approach) and to compare different modal logics within a single uniform framework.

In section 3.1, we will describe the formal basis for the truth values to be chosen and the way in which the closure of a certain set of propositions is computed using this basis. Section 3.2 shows how functions can serve as truth values in this sense and applies these results to FTL. In sections 3.3 and 3.4, we consider the MVL concept of modal operators as described in [8] and its application to our temporal logic before we finally give some complexity results in 3.5.

3.1 Mathematical Preliminaries of MVL

The fundamental notion in connection with MVL truth values is that of a *bilattice* defined below:[1]

Definition 3.1 ([6]) A *bilattice* is a sextuple $(B, \wedge, \vee, \cdot, +, \neg)$ such that

1. $(B, \wedge, \vee)$ and $(B, \cdot, +)$ are both complete lattices.
2. $\neg : B \rightarrow B$ is a mapping with
   (a) $\neg^2 = 1$, and
   (b) $\neg$ is a lattice homomorphism from $(B, \wedge, \vee)$ to $(B, \vee, \wedge)$ and from $(B, \cdot, +)$ to itself.

If the operations $\wedge, \vee, +$, and $\cdot$ distribute with respect to each other, the bilattice is called *distributive*. If only $\wedge, \vee$ and $\cdot, +$ each distribute with respect to each other, it is called *t-distributive* and *k-distributive*, resp.

The elements of a bilattice can be considered as truth values if its operations are interpreted in the following way: The two pairs of operations $\wedge, \vee$ and $\cdot, +$ each induce a partial order on the elements of B, denoted by $\leq_t$ and $\leq_k$, resp. If x and y are elements of B with $x \leq_t y$, we interpret this by saying that y represents a truth value that is "nearer to truth" than the one represented by x. In other words, a formula assigned the truth value y is considered "more true" than one assigned x. An example for this ordering is $f \leq_t t$. Thus, $\leq_t$ represents the degree of certainty about the validity of a certain statement.

If on the other hand $x \leq_k y$, we say that y stands for a greater amount of knowledge about a certain fact than x. If we allow for some truth value u (*unknown*), we have $u \leq_k f$ and $u \leq_k t$, whereas t and f are incomparable with respect to $\leq_k$.

Completing the truth values used so far with another element denoted by $\perp$ which stands for "both t and f", we obtain the smallest non-trivial bilattice F representing the set of truth values used in first-order logic in MVL. Figure 1 shows this bilattice where $\leq_t$ increases from left to right and $\leq_k$ from bottom to top. The role of the $\neg$ operation is to invert the $\leq_t$ order while retaining $\leq_k$. We now relate the bilattice operations $\wedge, \vee$, and $\neg$ to the interpretation of the elements of the bilattice as truth values. As the notation indicates, there is a strong similarity between these operations in a bilattice and their syntactical counterparts within logic. Considering the bilattice F for example, we get $t \wedge f = f$, $t \vee f = t$, $\neg t = f$ etc., just as expected from traditional logic.[2]

[1] The presentation of the mathematical foundations of MVL in this paper has to concentrate on the most essential topics. For an elaborated description, the reader is referred to [6].

[2] $\wedge$ yields the greatest lower bound glb_t of its arguments w.r.t. $\leq_t$ and $\vee$ the least upper bound lub_t.

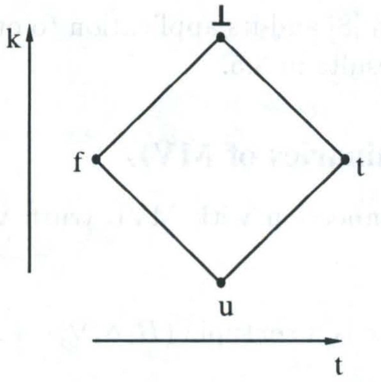

Figure 1: The bilattice F for first-order logic.

Definition 3.2 ([6]) Given some logical language $\mathcal{L}$ and a bilattice B, a *truth assignment* is a mapping $\phi : \mathcal{L} \rightarrow B$.

In MVL, such a mapping corresponds to a declarative database. In conventional logics, inferencing a formula p from a set S of axioms consists in checking if p is a member of the deductive closure of S. The multivalued counterpart for this process is to compute the truth value of p in the *closure* $cl(\phi)$ of a truth assignment ϕ.

Definition 3.3 ([6]) A truth assignment ϕ is *closed* if for all $p, q, p_1, p_2, ... \in \mathcal{L}$

1. $\phi(\bigwedge_i p_i) \geq_k \bigwedge_i \phi(p_i)$,
2. $\phi(\neg p) = \neg\phi(p)$, and
3. if $p \models q$, then $\phi(q) \geq_t \phi(p)$.

This means that – as already mentioned – the behaviour of the bilattice operation $\neg$ perfectly corresponds to negation in logic (2.), and that a formula q can't be "less true" than a formula p entailing it (3.). The content of clause 1. becomes clear when considering a formula $p \wedge \neg p$ where both conjuncts are assigned u but their conjunction obviously should be assigned f.

According to the "classical" approach to logic where the closure of some set T of sentences is defined to be the intersection of all deductively closed sets containing T, we define the closure $cl(\phi)$ of a truth assignment ϕ by:

Definition 3.4 ([6]) $cl(\phi) = \prod \{\psi \mid \psi \geq_k \phi$ and ψ is closed$\}$, where ϕ and ψ are compared pointwise.

To give a *constructive* account of closure, we will restrict ourselves here to the case of so-called *canonically grounded* bilattices where each element x can be expressed uniquely by a sum $x = g_t(x) + g_f(x)$, the *t-grounding* and *f-grounding* of x, resp. They correspond to the primitive bits of information x is composed of.[3] The bilattice F and the one used for FTL are both canonically grounded.

Now let p and q be sentences of our logical language such that $p \models q$ and $\phi(p) = x$. According to definition 3.3, we have $cl(\phi)(q) \geq_t x$ and thus $cl(\phi)(q) \geq_k g_t(x)$ (since

[3] In the bilattices considered here, $g_t(x) = x \vee u$, $g_f(x) = x \wedge u$. E.g., $g_t(\bot) = t$, $g_f(\bot) = f$.

$x = g_t(x) + g_f(x) \geq_k g_t(x)$). So the contribution of p to q's truth value is $g_t(\phi(p))$. If there are many sentences entailing q, this knowledge has to be accumulated by summing over all t-groundings.

For a set $S \subseteq \mathcal{L}$ of sentences we define $\phi(S) := \bigwedge_{p \in S} \phi(p)$. If $p \in \mathcal{L}$, we denote by $\pi_+(p)$ and $\pi_-(p)$ the sets of all subsets of $\mathcal{L}$ entailing p and $\neg p$, resp.:

$$\pi_+(p) = \{S \mid S \models p\}, \quad \pi_-(p) = \{S \mid S \models \neg p\}.$$

A truth assignment ϕ is called *$\neg$-closed* if $\phi(\neg p) = \neg\phi(p)$ for all p. This property can be characterized by the following lemma:

Lemma 3.1 ([6]) A truth assignment ϕ is $\neg$-closed iff

$$\phi(p) = \sum_{q \equiv_\neg p} \phi(q) + \sum_{q \equiv_\neg \neg p} \neg\phi(q)$$

for all p, where $q \equiv_\neg p$ if there exists a nonnegative integer n such that $p = \neg^{2n} q$ or vice versa.

Now assume we are given a $\neg$-closed truth assignment ϕ.[4] Then we have

Theorem 3.1 ([6]) The closure of ϕ is given by

$$\begin{aligned} cl(\phi)(p) &= \sum_{S \in \pi_+(p)} [\phi(S) \vee u] + \sum_{S \in \pi_-(p)} [\neg\phi(S) \wedge u] \\ &= \sum_{S \in \pi_+(p)} g_t[\phi(S)] + \sum_{S \in \pi_-(p)} \neg g_t[\phi(S)] \end{aligned}$$

This result implies a method to effectively compute the closure by steadily pruning formulas from the search space whose truth value is $<_k$ than the truth value already accumulated during the previous steps of the proof, since they can't make a real contribution to $cl(\phi)(p)$. Furthermore, the summation over $\pi_-(p)$ may be left out if we are only interested in the *truth* of p, i.e., if we want to show $cl(\phi)(p) \geq_k t$.

3.2 Functional Truth Values

For some applications it is not sufficient to use some kind of "atomic" truth values like t and f. Instead, it might be convenient to employ *mappings* from a given set to some bilattice as truth values. Ginsberg describes this for a simple temporal logic with truth values $g : I\!N \rightarrow F$ from the set of time points to the classical truth values (cf. [8], [9]). After describing the principal concept of functional truth values, we will see in this section how this technique can be applied to FTL.

Given a set S and a bilattice B, the set B^S of functions from S to B obviously inherits the bilattice property from B if the operations $\wedge, \vee, \cdot, +, \neg$ are computed pointwise.

Taking S to be the set of time intervals represented by $I\!N^2$, and B to be the first-order bilattice F, we obtain the new bilattice of truth values of FTL, denoted by B_I.

[4]If ϕ doesn't meet this property, we compute its $\neg$ closure according to lemma 3.1.

Intervals are represented as pairs of natural numbers $i = (a, l)$, where a is the first state and l is the number of states of i. Thus, the elements of B_I are *total* functions over $I\!N^2$.

Some practical problems arise with this approach: How to represent the infinitely large set $I\!N^2$ and the functions $g : I\!N^2 \to F$?

We begin with the second question and assume the general case of a bilattice B^S. If we can put some order on the set S, it is possible to make the representation of the functions g more compact by only listing those points of S explicitly where the value of g changes and assuming g to be constant between two such so-called *exception points*. As in general there will be no natural total order available for S, we organize S as a directed acyclic graph (DAG) with an induced partial order $\preceq$:

$$s_1 \preceq s_2 \ :\Longleftrightarrow \ \text{there is a path from } s_1 \text{ to } s_2 \text{ in } S.$$

To effectively represent this DAG, it is sufficient to have its root and a function computing the common successors of a given pair of points.

The required structure for the interval DAG D_I can be extracted from definition 2.4. As our time begins with state 0, we define the root of D_I to be $(0, 1)$, i.e., the interval consisting only of state 0. Examining those intervals sharing some common properties yields four classes:

1. Intervals of the form $(a, 1)$ consisting of a single state. They correspond to the embedding of time *points* into intervals. Within D_I, they yield a path $(0,1) \to (1,1) \to (2,1) \to \ldots$ called the *α-branch* of D_I.

2. Intervals *beginning with the same state*, as a non-modal formula is valid in (a, l) iff it is in (a, l') for any $l' > 0$. For each $a \in I\!N$, these form a path $(a,1) \to (a,2) \to (a,3) \to \ldots$, the *$\beta_a$-branch* of D_I.

3. Intervals *ending in the same state*. They all are terminating subintervals of a common greater interval and form pathes $(0, l) \to (1, l-1) \to \ldots \to (l-1, 1)$. A sequence of this form is called the *γ_l-branch* of D_I as for all $(a', l') \in \gamma_l$: $a' + l' = l$. By $\gamma_{a,l}$ we denote that part of γ_{a+l} beginning with (a, l).

4. "Empty" intervals $(a, 0)$. They are only mentioned for completeness as any formula is considered *unknown* in such an interval. In the following, we won't regard this kind of intervals anymore.

Figure 2 shows a part of the resulting graph D_I.

This structure of D_I allows to pass truth information along the edges to represent all kinds of relations between intervals described in definition 2.4. We now demonstrate how the concepts developed so far can be applied.

Example 3.1 Let p be a first-order formula *true* in state 2 and *unknown* everywhere else. According to definition 2.4, p is true in *all* intervals beginning with state 2. This can be expressed by assigning p a truth function $g_1 : I\!N^2 \to F$ with

$$g_1((a, l)) = \begin{cases} t, & a = 2, l \geq 1 \\ u, & otherwise \end{cases}$$

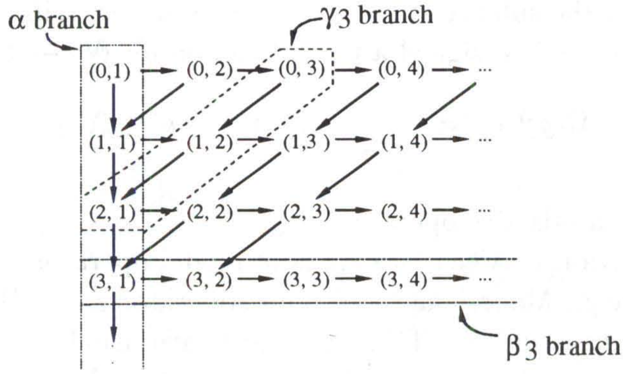

Figure 2: The DAG D_I of intervals.

Listing g_1's exception points and their respective truth values yields $\langle (0,1)-u,\ (2,1)-t,\ (3,1)-u\rangle$. In this case, we use inheritance along the β_2 branch of D_I and the fact that the *interval* $(2,1)$ of the α branch corresponds to *state* 2.

If we want to express that an FTL formula q is true in *all* states of an interval, e.g., $(5,3)$, we use a truth function g_2 with

$$g_2(w) = \begin{cases} t, & w \in \{(5,3),(6,2),(7,1)\} \\ u, & otherwise \end{cases}$$

The exception point list of g_2 is $\langle (0,1)-u,\ (5,3)-t,\ (5,4)-u\rangle$, where t is passed along the $\gamma_{5,3}$ branch of D_I.

3.3 Realization of Modal Operators

As already mentioned in section 3.1, the bilattice operations like "$\wedge$" play two distinct roles: Besides being a function on elements of a bilattice, they occur as binary operators in our logical language. In [8], Ginsberg generalizes this view to arbitrary operations:

> "...bilattice operations can be viewed *in general* as establishing semantic meanings for their syntactic counterparts. These syntactic counterparts are generally referred to as *modal operators*."

Definition 3.5 ([8]) Given a bilattice B, any n-ary function $g : B^n \to B$ is a *modal operator*.

Usually, modal operators are given a semantics using Kripke's approach of possible worlds (cf.[11]). Given an accessibility relation r among worlds and a modal operator Ω, where Ωp is intended to be valid in a world w if p holds in all worlds w' accessible from w, we can define the semantics of this operator by conceptually introducing a function Γ that takes a formula p and a world w and returns the truth value of p in w.[5] Then we have

$$\Gamma(\Omega p, w) \;=\; \bigwedge_{\substack{w' \\ r(w,w')}} \Gamma(p, w').$$

[5] Ω corresponds to the classical "necessity" operator.

How can this be related to the results of the previous section? Taking time intervals as possible worlds and the subinterval relationship $\overline{R}_t$ as accessibility relation,[6] we get for a formula Ωp, where p is assigned a truth function $g : \mathbb{N}^2 \rightarrow F$:

$$\Omega(g)(w) \;=\; \bigwedge_{w' \; \overline{R}_t \; w} g(w') \qquad (w \in \mathbb{N}^2)$$

In this equation, Ω is a bilattice operator.
There exists a distinction between two kinds of modal operators: The so-called *non-deductive* operators, e.g., Moore's autoepistemic operator L (cf. [12]), do not respect the bilattice operations $\cdot$ and $+$. These operators are usually not given a possible-worlds semantics, but are characterized by some functional relationship between the truth values of their "input" formulas and their results. The class of *deductive* modal operators is characterized by the classical Kripke semantics and comprises operators like those of necessity and possibility. When computing the closure $cl(\phi)$ of a truth assignment ϕ, the following inequality holds for deductive operators Ω:

$$cl(\phi)(\Omega(p_1, ..., p_n)) \;\geq_k\; \Omega(cl(\phi)(p_1), ..., cl(\phi)(p_n)),$$

while there is a strict equality for the non-deductive ones.

Proposition 3.1 ([8]) Any modal operator on a distributive bilattice B that can be written in terms of $\wedge, \neg, +$, constant functions and projections is deductive.

Projections are functions $\pi_{w,w'}$ indexed by two possible worlds w, w' that – when supplied with a truth function g and a world w'' – return the value of g at w' if $w = w''$ and u otherwise; i.e., such a projection exactly represents the property of accessibility of w' from w.

3.4 FTL Modal Operators

In the following, we will describe how the basic modal operators of FTL introduced in section 2 can be expressed as bilattice operations on B_I.
Let $g, g_1, g_2 \in B_I$ be truth functions $\mathbb{N}^2 \rightarrow F$, and $w, w', w'' \in \mathbb{N}^2$ pairs of natural numbers representing intervals.
Then we can express the *next* operator $\bigcirc$ by

$$next(g)(w) \;=\; g(w') \qquad (1)$$

where w' is uniquely determined by $w' \, R_t \, w$. This corresponds to the intuitive semantics of *next*: To see if $\bigcirc p$ is valid in an interval w, check p's truth value in its first terminating subinterval w'. Obviously, the result of *next* is itself an element of B_I.

With the other basic operators, things become slightly more complicated. Consider the operator *until*. Translating its semantics given in definition 2.4 into the MVL formalism yields the following: Given two truth functions $g_1, g_2 \in B_I$, we can determine

[6] I.E., w' is accessible from w iff $w' \overline{R}_t w$.

the value of $until(g_1, g_2)$ at w by checking if there is some terminating subinterval w'' such that $g_2(w'') = t$ and for all subintervals w' of w *before* w'': $g_1(w') = t$. We get

$$until(g_1, g_2)(w) = \bigvee_{w''\overline{R}_t w} \Big[\bigwedge_{\substack{w'\overline{R}_t w \\ w'' R_t^+ w'}} g_1(w') \wedge g_2(w'') \Big]. \tag{2}$$

Remark: In (2), lub_t and glb_t are guaranteed to exist even in the infinite case because – according to definition 3.1 – $(B_I, \wedge, \vee)$ is a *complete* lattice. In contrast to traditional logic, we can therefore express existential and universal propositions about worlds by disjunctions and conjunctions, resp.

For the *chop* operator, the translation is similar: Check if there is a terminating subinterval w' of w such that $g_2(w') = t$ and for the complement w'' of w' in w: $g_1(w'') = t$. Again, we get a disjunction

$$chop(g_1, g_2)(w) = \bigvee_{w'\overline{R}_t w} [g_1(compl_w(w')) \wedge g_2(w')]. \tag{3}$$

Here, the above remark is also valid yielding the correctness of (3). As a consequence of (1), (2), (3), and proposition 3.1, we get the following result:

Corollary 3.1 The modal operators *next* ($\bigcirc$), *until* ($\mathcal{U}$), and *chop* ($\mathcal{C}$) are deductive.

From the equations listed above, an actual implementation of these operators can be easily derived. For this, notice that

1. $(a', l')\ R_t\ (a, l) \iff l > 1, a' = a + 1$, and $l' = l - 1$.
2. $(a', l')\ \overline{R}_t\ (a, l) \iff a + l = a' + l', l' > 0$, and $a' \geq a$.
3. $(a', l')\ R_t^+\ (a, l) \iff a + l = a' + l', l' > 0$, and $a' > a$.
4. $compl_{(a,l)}((a + i, l - i)) = (a, i + 1)$.

So we get

$$next(g)((a, l)) = g((a + 1, l - 1)) \tag{4}$$

and

$$until(g_1, g_2)((a, l)) = \bigvee_{i=0}^{l-1} \Big[\bigwedge_{j=0}^{i-1} g_1((a + j, l - j)) \wedge g_2((a + i, l - i)) \Big]. \tag{5}$$

The implementation of *chop* is done by

$$chop(g_1, g_2)((a, l)) = \bigvee_{i=0}^{l-1} [g_1((a, i + 1)) \wedge g_2((a + i, l - i))]. \tag{6}$$

To complete this section, we will consider the derived modal operators *always* ($\Box$) and *sometimes* ($\Diamond$). As mentioned in section 2, *sometimes* can be obtained by

$$\Diamond p :\equiv (T\ \mathcal{U}\ p).$$

Denoting the truth functions assigned to T and p by g_T and g_p, resp. (i.e., $\forall w \in I\!N^2 : g_T(w) = t$), and inserting them into (2), we immediately get

$$\begin{aligned} sometimes(g_p)(w) &= until(g_T, g_p)(w) \\ &= \bigvee_{w''\overline{R}_t w} \Big[\bigwedge_{\substack{w'\overline{R}_t w \\ w''R_t^+ w'}} t \wedge g_p(w'') \Big] \quad \text{(cf. def. of } g_T) \\ &= \bigvee_{w''\overline{R}_t w} g_p(w'') \end{aligned} \tag{7}$$

Taking advantage of the duality of *sometimes* and *always*, it is possible to derive its realization by

$$\begin{aligned} always(g)(w) &= \neg sometimes(\neg g)(w) \\ &= \neg \bigvee_{w'\overline{R}_t w} \neg g(w') \\ &= \bigwedge_{w'\overline{R}_t w} g(w') \end{aligned} \tag{8}$$

In general, formulas like (2) and (3) cannot be effectively computed as they may contain infinitely large disjunctions and conjunctions. The representation of the set of intervals in a DAG and of the truth functions by only listing their exception points, however, admits these computations for many cases.

Example 3.2 In example 3.1, we assigned a truth function g_2 to a formula q with the intention to express that q is true during the whole interval $(5,3)$. Applying *always* to this function yields $always(g_2) = g_2$, i.e., g_2 in fact formalized what we intended. Consider another formula r with truth function g_3 where

$$g_3(w) = \begin{cases} t, & w \in \{(7,2),(8,1)\} \\ u, & otherwise \end{cases}$$

i.e., r is true in the whole interval $(7,2)$. Then the truth value of $(q\ \mathcal{C}\ r)$ is a function $chop(g_2, g_3) = g_4$ with

$$g_4(w) = \begin{cases} t, & w \in \{(5,4),(6,3),(7,2)\} \\ u, & otherwise \end{cases}$$

that is represented using the exception points $\langle (0,1) - u,\ (5,4) - t,\ (5,5) - u \rangle$. Applying *next* to g_3, i.e., computing the truth value of $\bigcirc r$, yields $next(g_3) = g_5$ with

$$g_5(w) = \begin{cases} t, & w \in \{(6,3),(7,2)\} \\ u, & otherwise \end{cases}$$

We have a slightly different view on modal operators than Ginsberg has. According to his understanding of modal operators, *next* should *modify* the truth value of r by pushing it one step into the future, whereas the above result is the truth function obtained from the query "In which intervals is $\bigcirc r$ true?"

3.5 Complexity Considerations

For the case of *chop*, we will describe an algorithm for (6) exploiting the sparsity of exception points of truth functions and compute its complexity.

Let $g_1, g_2 \in B_I$, $exc(g_1), exc(g_2)$ their respective sets of exception points with $|exc(g_1)| = n_1$ and $|exc(g_2)| = n_2$. As indicated in equation (6), to compute the value of $chop(g_1, g_2)$ at (a, l), we have to consider each possible splitting of (a, l) into two subintervals (a_1, l_1) and (a_2, l_2) such that $(a, l) = (a_1, l_1) \oplus (a_2, l_2)$ and combine the respective values of g_1 and g_2 using "$\wedge$". All these intermediate results are put together in one disjunction yielding the final value at (a, l). In a first step, we have to complete both sets in the following way:

1. Each $(a_2, l_2) \in exc(g_2)$ can be combined with $(a_1, l_1) \in exc(g_1)$ iff $a_1 + l_1 - 1 = a_2$, i.e., iff $(a_1, l_1) \in \gamma_{a_2+1}$. So, we need all points on γ_{a_2+1} where g_1 changes its value. Apart from the points of $\gamma_{a_2+1} \cap exc(g_1)$, these are all those elements (a_1, l_1) of $exc(g_1)$ passing their truth values into γ_{a_2+1} by inheritance. We have to collect their projections $(a_1, a_2 - a_1 + 1)$ onto γ_{a_2+1}. In the worst case, we therefore have to consider each member of $exc(g_1)$ for each of $exc(g_2)$ and obtain a complexity of $O(n_1 \cdot n_2)$.

2. In the other direction, any $(a_1, l_1) \in exc(g_1)$ is combinable with any $(a_2, l_2) \in exc(g_2)$ where $a_1 + l_1 - 1 = a_2$, i.e., with all members of β_{a_2}. Just as in 1., we have to compute the projections of all points (a_2', l_2') of $exc(g_2)$ influencing the truth values along β_{a_2}. These are the points $(a_2, a_2' + l_2' - a_2)$. The complexity of this step is again $O(n_1 \cdot n_2)$.

3. Combining the points obtained from the two previous steps according to (6) yields the same complexity again.

So, the computational overhead caused by *chop* is merely $O(n_1 \cdot n_2)$. For *until*, the process is similar and takes the same time, whereas *next* can be implemented to consume linear time.

4 Applications

One possible application for a system as the one described is in the field of plan recognition. Assume we are given some observed actions $a_1(t_1)$ and $a_2(t_2)$ with their actual parameters t_i and exact temporal information about their occurrences and some plan hypotheses P_1 and P_2 written as FTL formulas, where

$$\forall x_1, x_2, x_3. P_1(x_1, x_2, x_3) \equiv (a_1(x_1) \wedge \Diamond (a_2(x_2) \, \mathcal{C} \, a_3(x_3)))$$
$$\forall x_1, x_2, x_3. P_2(x_1, x_2, x_3) \equiv (a_1(x_1) \wedge \bigcirc a_2(x_2) \wedge \bigcirc\bigcirc a_3(x_3))$$

Using ordinary deduction, we can infer which of P_1 and P_2 is *not* a valid hypothesis for an explanation of the observed action sequence. Assume $a_1(t_1)$ is observed in state 5, i.e., in the interval $(5, 1)$, and $a_2(t_2)$ in state 10. Trying to derive $\neg P_1$ from this database yields a truth function g_{P_1} with value t everywhere except for all intervals

$(5, l)$ of β_5 with $l \geq 6$, where g_1 yields u. The reason for this result is the fact that any interval in which P_1 could hold has to begin with state 5 and include at least state 10 where $a_2(t_2)$ takes place. The derivation of $\neg P_2$, however, yields a constant truth function g_{P_2} with value t. The result of these inferences is that P_1 is consistent with the observations and thus a valid hypothesis for the observed action sequence – in contrast to P_2 that is not.

If later on action $a_3(t_3)$ is observed – e.g., at state 12 – we can even derive P_1 with a truth function g'_{P_1} that yields t for all $(5, l)$ with $l \geq 8$ as now each interval satisfying P_1 starts at state 5 and includes at least state 12.

In [4], a similar approach to plan recognition with a temporal modal logic is described.

5 Conclusions

We introduced a modal temporal logic FTL based on work described in [14] and the basic concepts of Ginsberg's MVL presented in [6]. The main emphasis lay on the translation of FTL into the MVL formalism, where the choice of functions $\mathbb{N}^2 \rightarrow F$ as truth values – as counterparts for its interval-based semantics – and the implementation of FTL's modal operators as functions over these truth values played a central role. We finally showed that it is even possible to give efficient implementations for these concepts by exploiting some constraints on the structure of truth values.

As expected, this efficiency is not for free. For example, truth functions changing their value infinitely often (e.g., from state to state) can't be represented using the methods described.

Another – perhaps even more serious – drawback lies in the limitation of possible inferences caused by MVL. To reason about a formula Ωp containing a modal operator Ω, p is required to have a *concrete truth value* that can serve as input for the *function* Ω. Axiom schemata like $\bigcirc A \rightarrow \Diamond A$ are conceptually not supported. Thus, the applicability of inference systems based on MVL is limited to cases of ***diagnostic*** reasoning, where a set of observations with their actual truth values is given. In such a situation, tasks like temporal projection are also solvable by using modal operators that "push" certain truth values into the future. Examples are reasoning about actions (e.g., Ginsberg's treatment of the Yale Shooting Problem (cf. [9] and [3])), plan recognition as described in [4], and all kinds of fault diagnosis. For such tasks, a more powerful – and less efficient – prover is generally not needed, but can be replaced by an inference system as the one described above.

Acknowledgements

I'd like to thank Gaby Merziger and Dietmar Dengler for fruitful discussions and Jana Köhler for her comments on an earlier version of this paper.

References

[1] J. Allen. Towards a General Theory of Action and Time. *Artif. Intell.*, 23:123–154, 1984.

[2] J. Allen. Planning as Temporal Reasoning. In *Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[3] A. Baker and M. Ginsberg. Temporal Projection and Explanation. In *Proc. of the 11th International Joint Conference on AI*, 1989.

[4] M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Köhler, and G. Merziger. Integrated Plan Generation and Recognition – a Logic-based Approach. In W. Brauer, editor, *4. Internationaler GI-Kongreß Wissensbasierte Systeme*. Springer, 1991.

[5] J. Burgess. Basic Tense Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol. II*, pages 89–133, 1984.

[6] M. Ginsberg. Multivalued Logics: A Uniform Approach to Inference in AI. *Computational Intelligence*, 4:265–316, 1988.

[7] M. Ginsberg. A Circumscriptive Theorem Prover. *Artif. Intell.*, 39:209–230, 1989.

[8] M. Ginsberg. Bilattices and Modal Operators. *J. of Logic and Computation*, 1, 1990.

[9] M. Ginsberg. Computational Considerations in Reasoning about Action. In *Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[10] R. Hale. Temporal Logic Programming. In A. Galton, editor, *Temporal Logics and Their Applications*, pages 91–119. Academic Press, 1987.

[11] S. Kripke. Semantical Considerations on Modal Logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[12] R. Moore. Semantical Considerations on Nonmonotonic Logic. *Artif. Intell.*, 25:75–94, 1985.

[13] B. Moszkowski. A Temporal Logic for Multi-level Reasoning about Hardware. *Computer*, 18:10–19, 1985.

[14] R. Rosner and A. Pnueli. A Choppy Logic. In *Proc. of the Symposium on Logic in Computer Science*, pages 306–313, June 1986.

[15] Y. Shoham. Time for Action: On the Relation between Time, Knowledge and Action. In *Proc. of the 11th International Joint Conference on AI*, pages 954–960, 1989.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-91-17
Andreas Dengel, Nelson M. Mattos:
The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein:
A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder:
Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

RR-92-01
Werner Nutt: Unification in Monoidal Theories is Solving Linear Equations over Semirings
57 pages

RR-92-02
Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Hönes, Frank Fein, Michael Malburg:
Π_{ODA}: The Paper Interface to ODA
53 pages

RR-92-03
Harold Boley:
Extended Logic-plus-Functional Programming
28 pages

RR-92-04
John Nerbonne: Feature-Based Lexicons: An Example and a Comparison to DATR
15 pages

RR-92-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Michael Schulte, Rainer Stark:
Feature based Integration of CAD and CAPP
19 pages

RR-92-06
Achim Schupetea: Main Topics of DAI: A Review
38 pages

RR-92-07
Michael Beetz:
Decision-theoretic Transformational Planning
22 pages

RR-92-08
Gabriele Merziger: Approaches to Abductive Reasoning - An Overview -
46 pages

RR-92-09
Winfried Graf, Markus A. Thies:
Perspektiven zur Kombination von automatischem Animationsdesign und planbasierter Hilfe
15 Seiten

RR-92-10
M. Bauer: An Interval-based Temporal Logic in a Multivalued Setting
17 pages

RR-92-11
Susane Biundo, Dietmar Dengler, Jana Koehler:
Deductive Planning and Plan Reuse in a Command Language Environment
13 pages

RR-92-13
Markus A. Thies, Frank Berger:
Planbasierte graphische Hilfe in objektorientierten Benutzungsoberflächen
13 Seiten

RR-92-14
Intelligent User Support in Graphical User Interfaces:
1. InCome: A System to Navigate through Interactions and Plans
Thomas Fehrle, Markus A. Thies
2. Plan-Based Graphical Help in Object-Oriented User Interfaces
Markus A. Thies, Frank Berger

22 pages

RR-92-15
Winfried Graf: Constraint-Based Graphical Layout of Multimodal Presentations
23 pages

RR-92-16
Jochen Heinsohn, Daniel Kudenko, Berhard Nebel, Hans-Jürgen Profitlich: An Empirical Analysis of Terminological Representation Systems
38 pages

RR-92-17
Hassan Aït-Kaci, Andreas Podelski, Gert Smolka: A Feature-based Constraint System for Logic Programming with Entailment
23 pages

RR-92-18
John Nerbonne: Constraint-Based Semantics
21 pages

RR-92-19
Ralf Legleitner, Ansgar Bernardi, Christoph Klauck
PIM: Planning In Manufacturing using Skeletal Plans and Features
17 pages

RR-92-20
John Nerbonne: Representing Grammar, Meaning and Knowledge
18 pages

RR-92-21
Jörg-Peter Mohren, Jürgen Müller
Representing Spatial Relations (Part II) -The Geometrical Approach
25 pages

RR-92-22
Jörg Würtz: Unifying Cycles
24 pages

RR-92-24
Gabriele Schmidt: Knowledge Acquisition from Text in a Complex Domain
20 pages

RR-92-25
Franz Schmalhofer, Ralf Bergmann, Otto Kühn, Gabriele Schmidt: Using integrated knowledge acquisition to prepare sophisticated expert plans for their re-use in novel situations
12 pages

RR-92-26
Franz Schmalhofer, Thomas Reinartz, Bidjan Tschaitschian: Intelligent documentation as a catalyst for developing cooperative knowledge-based systems
16 pages

RR-92-27
Franz Schmalhofer, Jörg Thoben: The model-based construction of a case-oriented expert system
18 pages

RR-92-29
Zhaohur Wu, Ansgar Bernardi, Christoph Klauck: Skeletel Plans Reuse: A Restricted Conceptual Graph Classification Approach
13 pages

DFKI Technical Memos

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel: ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang:
Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh
A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

An Interval-based Temporal Logic in a Multivalued Setting

Mathias Bauer

DFKI-RR-92-10

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9000 8).

An Interval-based Temporal Logic in a Multivalued Setting

Mathias Bauer

RR-92-10

Research Report