

A Modal μ -Calculus for Durational Transition Systems

Helmut Seidl
FB IV – Informatik
Universität Trier
D-54286 Trier, Germany
email: `seidl@psi.uni-trier.de`

February 29, 1996

Abstract

Durational transition systems are finite transition systems where every transition is additionally equipped with a duration. We consider the problem of interpreting μ -formulas over durational transition systems. In case the formula contains only operations minimum, maximum, addition, and sequencing, we show that the interpretation is not only computable but (up to a linear factor) as efficiently computable as the interpretation of μ -formulas over ordinary finite transition systems.

1 Introduction

The μ -calculus, a propositional modal logic with fixpoint operators, was introduced by Dexter Kozen as a tool for the description and analysis of the behavior of possibly infinite computations [11]. The expressive power of the modal μ -calculus is intimately connected to finite-state automata on infinite trees [20, 6, 7, 4]. Classically, a μ -formula denotes a predicate on states. Typical properties to be expressed and analyzed are *safety* and *liveness* assertions. The formula $\nu x.\langle a \rangle x$, for example, denotes the set of all states allowing for an infinite sequence of a -actions. Especially, the modal operator $\langle a \rangle$ (for action a) constructs a property of the actual state from a property of a next state. Thus, it relates presence to a

(possibly infinite) future. For detailed explanation of various modal (and temporal) logics consult, e.g., C. Stirling in [18].

The problem with which we are concerned here, however, is not only to determine properties of states but also quantitative information. We are interested in questions like: *How long does it take a process to reach a state with a certain property?* One approach to answer such questions has been suggested by E.A. Emerson in [8]. Emerson extends the μ -calculus by allowing “bounded” fixpoint operators. A (closed) formula $\mu t X.\phi$ holds true if the t -th approximation of the fixpoint of ϕ holds true. Thus, some upper bound t can be verified for the time an event is going to happen.

Opposed to Emerson, we no longer interpret closed formulas as predicates or, equivalently, as mappings $Q \rightarrow \{0, 1\}$ for some base set Q . Instead, we employ mappings $t : Q \rightarrow D$ for some time domain D . In this paper we consider D as a subinterval of

$$-\infty < 0 < 1 < 2 < \dots < n < \dots < \infty$$

i.e., the non-negative integers extended by $-\infty$ (to express unaccessibility) and ∞ . The “logical” connectives “ \vee ” and “ \wedge ” are replaced with maximum and minimum, respectively. Additionally, we consider addition “ $+$ ” (appropriately extended) and the sequencing operator “ $;$ ” which returns the second argument only provided the first one is different from $-\infty$. Otherwise, the result is $-\infty$. While addition

is useful for modeling the time behavior of two processes being executed one after the other, the “;” may serve for a primitive form of conditional: the second argument is only counted provided the event corresponding to the first will ever occur.

A corresponding theory of processes with durational actions was developed by Gorrieri, Rocchetti and Stancampiano [9]. For the analysis of complexity-like properties of programs, e.g., the number of accesses to some variable, least solutions of (non-hierarchical) systems of equations over non-negative integers (extended by ∞) have been considered in [15, 17]. This corresponds to the interpretation of μ -formulas *without* occurrences of the greatest fixpoint operator.

A formula ϕ of the μ -calculus may contain occurrences both of the least fixpoint operator μ and the greatest fixpoint operator ν . For our interpretation, we consider *complete lattices* D with least element \perp and top element \top and monotonic functions. Recall the fixpoint theorem of Tarsky–Knaster which states that every monotonic function $f : D \rightarrow D$ (D a complete lattice) both has a least and a greatest fixpoint. In case, f is upward continuous, i.e., commutes with least upper bounds of ascending chains, the least fixpoint can be explicitly described by $\mu x.f x = \bigsqcup_{n \geq 0} f^n \perp$. Accordingly, if f is downward continuous, i.e., commutes with greatest lower bounds of descending chains, the greatest fixpoint is given by $\nu x.f x = \bigsqcap_{n \geq 0} f^n \top$. Indeed, the set of non-negative integers equipped with their natural ordering and extended by $-\infty$ and ∞ forms a complete lattice. It is denoted by $[-\infty, \infty]$. The *intervals* $[k, n] \subseteq [-\infty, \infty]$ given by $\{j \in [-\infty, \infty] \mid k \leq j \leq n\}$, $-\infty \leq k \leq n \leq \infty$, form complete sublattices. Note that the two-point lattice **2** can be viewed as a special case of an interval containing just two elements (e.g., $-\infty$ and 0, or: k and $k + 1$). Since $[-\infty, \infty]$ satisfies the *descending chain condition* (abbreviated: dcc), every monotonic function between intervals automatically is downward continuous.

In this paper we show that the interpretation of μ -formulas over $Q \rightarrow D$ can be effectively

computed – even when the chosen time interval D is infinite. Moreover, we find that it is not only computable but, even more, as efficiently (at least up to a linear factor) computable as the interpretation over a two point interval.

The rest of the paper is organized as follows. The next section defines durational transition systems and the modal μ -calculus. Section 3 introduces the technically more flexible concept of “hierarchical systems of equations” and shows how to translate a μ -formula ϕ together with a durational transition system T into a hierarchical system S of equations such that the solution of S equals the interpretation of ϕ over T . Section 4 provides basic facts about hierarchical systems together with a Lower Bound Theorem and derives an efficient algorithm that computes solutions. Actually, this algorithm consists in iteratively applying some traditional algorithm for solving hierarchical systems of equations over the two-point domain. By the translation of Section 3, this algorithm can be used to compute the interpretation of a modal μ -formula over a durational transition system. Section 5 concludes. Because of lack of space some more technical proofs have been omitted. They can be found in the long version of the paper [16].

2 Durational Transition Systems and the μ -Calculus

A *durational transition system* T consists of a finite set of *states* Q , a finite set of *actions* A , a set $\Delta \subseteq Q \times A \times Q$ of *transitions* together with a *duration function* $d : \Delta \rightarrow D$. For transition τ , duration $d\tau$ describes the amount of time necessary to execute τ .¹ Usually, the set A of actions is fixed. Therefore, we denote T by the triple (Q, Δ, d) . A *basic time assignment* p for T is a mapping $p : Q \rightarrow D$. p assigns a time value pq to every state q .

The set $L\mu$ of formulas of the μ -calculus consists of all formulas given by the grammar:

$$\phi ::= p \mid x \mid \phi_1 \square \phi_2 \mid \langle a \rangle \phi' \mid [a] \phi' \mid \mu x. \phi' \mid \nu x. \phi'$$

¹We use a bracket convention for function application similar to functional languages like ML.

where p is a basic time assignment and \square is contained in some suitable set of binary operators Ω .

The set of operations, we are basically concerned with is $\Omega = \{\sqcup, \sqcap, +, ;\}$. “ \sqcup ” denotes maximum, “ \sqcap ” denotes minimum, “ $+$ ” denotes usual addition of non-negative integers, extended by $-\infty + x = -\infty$ for all x , and $\infty + y = \infty$ provided $y > -\infty$; finally, “ $;$ ” denotes the sequence operator defined by $x; y = -\infty$ if $x = -\infty$ and $x; y = y$ otherwise. For convenience, we will denote syntactic objects in Ω and their meaning as functions $D^2 \rightarrow D$ by the same symbol.

Consider a μ -formula ϕ where the free variables of ϕ are contained in \mathcal{F} , and assume we are given a durational transition system $T = (Q, \Delta, d)$ together with a set \mathcal{P} of basic time assignments for T . An *environment* for the free variables of ϕ is a mapping $\rho : \mathcal{F} \rightarrow Q \rightarrow D$. The meaning or *interpretation* $\llbracket \phi \rrbracket_D \rho$ of ϕ relative to environment ρ is defined as follows (again, if D is understood, we omit subscript D):

- basic time assignment: $\llbracket p \rrbracket \rho = p$;
- variable: $\llbracket x \rrbracket \rho = \rho x$ if $x \in \mathcal{F}$;
- operator application: $\llbracket \phi_1 \square \phi_2 \rrbracket \rho q = (\llbracket \phi_1 \rrbracket \rho q) \square (\llbracket \phi_2 \rrbracket \rho q)$ for $\square \in \Omega$;
- existential modal quantification: $\llbracket \langle a \rangle \phi \rrbracket \rho q = \bigsqcup_{\tau=(q,a,q') \in \Delta} (d\tau + \llbracket \phi \rrbracket \rho q')$
- universal modal quantification: $\llbracket [a] \phi \rrbracket \rho q = \bigsqcap_{\tau=(q,a,q') \in \Delta} (d\tau + \llbracket \phi \rrbracket \rho q')$
- least fixpoint: $\llbracket \mu x. \phi \rrbracket \rho = z_0$ iff z_0 is the least fixpoint of the monotone mapping $z \mapsto \llbracket \phi \rrbracket \rho[z/x]$;
- greatest fixpoint: $\llbracket \nu x. \phi \rrbracket \rho = z_0$ iff z_0 is the greatest fixpoint of the monotone mapping $z \mapsto \llbracket \phi \rrbracket \rho[z/x]$.

By considering modalities and taking durations into account which are connected with transitions, we are able to compute quantitative measures of how long a process may take. We do *not* consider several clock variables and

deal with linear constraints between them as in [1] or [10]. Instead, *absolute* time (resp. duration) values are computed. We illustrate this by the following simple example formulas.

Example 1 Consider a durational transition system $T = (Q, \Delta, d)$ with set of actions $A = \{a, b\}$ and durations $d\tau = 1$ for every transition τ . For some subset $Q' \subseteq Q$ let p and p' denote the basic time assignments defined by $p q = -\infty$ if $q \notin Q'$ and $p q = 0$ otherwise, resp. $p' q = \infty$ if $q \notin Q'$ and $p' q = 0$ otherwise.

- Consider $\psi \equiv \nu x. \langle a \rangle x$. Then $\llbracket \psi \rrbracket \emptyset q$ equals ∞ if q allows for an infinite sequence of a -actions, and $-\infty$ otherwise².
- Let $\bar{\psi} \equiv \mu y. [a] y$ denote the formal “complement” of ψ . Then $\llbracket \bar{\psi} \rrbracket \emptyset q$ equals ∞ if all paths of a -actions starting in q are finite, and $-\infty$ otherwise. Thus, $\bar{\psi}$ is indeed complementary to ψ .
- Now consider $\phi \equiv \mu x_1. p \sqcup \langle b \rangle x_1$. $\llbracket \phi \rrbracket \emptyset q$ returns the length of the *longest* path of b -actions from q to some state in Q' – provided it exists. If there are arbitrarily long paths of this type, ∞ is returned. If no such paths exist, $\llbracket \phi \rrbracket \emptyset q = -\infty$.
- Finally, let $\phi' \equiv \nu x_2. p' \sqcap [b] x_2$. ϕ' is *not* the precise formal complement of ϕ . Instead, $\llbracket \phi' \rrbracket \emptyset q$ returns the length of the *shortest* sequence of b -actions from q to some state in Q' – if it exists. If such a state is not reachable from q , the result is ∞ .

□

3 Hierarchical Systems of Equations

In this section we introduce hierarchical systems of equations and show how μ -formulas together with transition systems and environments can be translated into equivalent hierarchical systems of equations. As right hand

²By “ \emptyset ” we denote the environment with empty domain.

sides of equations we allow expressions built up from formal variables from some set \mathcal{Y} and constants by application of operators from Ω . The set of all these expressions is denoted by $\mathcal{E}_{\Omega \cup D}(\mathcal{Y})$. Every expression $f \in \mathcal{E}_{\Omega \cup D}(\mathcal{Y})$ denotes a function $\llbracket f \rrbracket_D : (\mathcal{Y} \rightarrow D) \rightarrow D$ (as usual, if D is understood, we omit subscript D). This function is upward continuous since all operations in Ω are upward continuous.

A *hierarchical* system of equations with free variables from \mathcal{F} consists of a *base set* S of equations $x = f_x, x \in \mathcal{X}$, where for every x , f_x is an expression in $\mathcal{E}_{\Omega \cup D}(\mathcal{X} \cup \mathcal{F})$; together with a *hierarchy* on \mathcal{X} . The hierarchy consists of a system \mathcal{H} of subsets $\sigma \subseteq \mathcal{X}$ together with a qualification $\kappa : \mathcal{H} \rightarrow \{\mu, \nu\}$ of subsets in \mathcal{H} . Intuitively, the hierarchical structure on S is introduced in order to reflect the nesting of scopes of variables bound by ν - or μ -operators. Therefore, \mathcal{H} must satisfy the following three conditions:

- $\mathcal{X} \in \mathcal{H}$ but $\emptyset \notin \mathcal{H}$;
- for $\sigma, \sigma' \in \mathcal{H}$ with $\sigma \cap \sigma' \neq \emptyset$, either $\sigma \subseteq \sigma'$ or $\sigma' \subseteq \sigma$;
- if $x \in \sigma$ and $x' \in \mathcal{X}$ occurs in f_x then for every $\sigma' \in \mathcal{H}$, $x' \in \sigma'$ implies $\sigma \subseteq \sigma'$ or $\sigma' \subseteq \sigma$.

Usually, if κ is understood, we write \mathcal{H} for the hierarchy, and if \mathcal{H} is understood, we write S for the hierarchical system. We also denote the fact that $\kappa(\sigma) = \mu$ by $\sigma : \mu$ (resp. $\kappa(\sigma) = \nu$ by $\sigma : \nu$).

A *subsystem* of S relative to some $\mathcal{X}' \subseteq \mathcal{X}$ is given by the base set $S_{\mathcal{X}'}$ of equations $x = f_x, x \in \mathcal{X}'$, together with hierarchy $\mathcal{H}_{\mathcal{X}'}$ which is the restriction of \mathcal{H} to $S_{\mathcal{X}'}$, i.e., it consists of system $\mathcal{H}_{\mathcal{X}'} = \{\emptyset \neq \mathcal{X}' \cap \sigma \mid \sigma \in \mathcal{H}\}$ together with qualification $\kappa_{\mathcal{X}'}$ mapping $\mathcal{X}' \cap \sigma$ to $\kappa(\sigma')$ for the least $\sigma' \in \mathcal{H}$ with $\mathcal{X}' \cap \sigma = \mathcal{X}' \cap \sigma'$.

An environment ρ for S is a mapping $\rho : \mathcal{F} \rightarrow D$. The *solution* of S relative to environment ρ in D , denoted by $\llbracket S \rrbracket_D \rho$, is a mapping $\mathcal{X} \rightarrow D$ (again, we sometimes suppress the subscript). It is defined by induction on $\#\mathcal{H}$. If $\#\mathcal{H} = 1$, then $\mathcal{H} = \{\mathcal{X}\}$. W.l.o.g. assume $\mathcal{X} : \mu$. Then

the solution of S relative to ρ is the least fixpoint of the function $G : (\mathcal{X} \rightarrow D) \rightarrow \mathcal{X} \rightarrow D$ where $G \beta x = \llbracket f_x \rrbracket (\rho + \beta)$. Here, operator “+” combines two functions with disjoint domains into one.

For $\#\mathcal{H} > 1$, let $\sigma_1, \dots, \sigma_m$ be the maximal elements of $\mathcal{H} \setminus \{\mathcal{X}\}$. The free variables of subsystems $S_j = S_{\sigma_j}$ are contained in $\mathcal{F} \cup \mathcal{X}'$ where $\mathcal{X}' = \mathcal{X} \setminus (\sigma_1 \cup \dots \cup \sigma_m)$. Therefore let $\sigma = \sigma_1 \cup \dots \cup \sigma_m$, and define $R : (\mathcal{F} \cup \mathcal{X}' \rightarrow D) \rightarrow \sigma \rightarrow D$ by $R (\rho + \beta) x = \llbracket S_j \rrbracket (\rho + \beta) x$ whenever $x \in \sigma_j$. Then mapping $x' \mapsto \llbracket S \rrbracket \rho x', x' \in \mathcal{X}'$, is the least fixpoint of function $G : (\mathcal{X}' \rightarrow D) \rightarrow \mathcal{X}' \rightarrow D$ where $G \beta x = \llbracket f_x \rrbracket (\rho + \beta + R (\rho + \beta))$. Finally, for $x \in \sigma_j, j = 1, \dots, m$, $\llbracket S \rrbracket \rho x = \llbracket S_j \rrbracket (\rho + \beta') x$ where $\beta' : \mathcal{X}' \rightarrow D$ is given by $\beta' x' = \llbracket S \rrbracket \rho x', x' \in \mathcal{X}'$.

Example 2 Assume S consists of the equations:

$$\begin{aligned} x_1 &= (x_1 \sqcup 6) \sqcap x_4 \\ x_3 &= x_3 + 1 \\ x_2 &= (x_1 \sqcap x_2) \sqcup 0 \\ x_4 &= x_3 \sqcap (x_2 + 1) \end{aligned}$$

where hierarchy \mathcal{H} together with the qualification are given by:

$$\begin{aligned} \{x_1, x_2, x_3, x_4\} &: \mu, & \{x_3\} &: \nu, \\ \{x_1, x_2\} &: \nu, & \{x_1\} &: \mu \end{aligned}$$

System S has no free variables. We find that

$$\begin{array}{c|cccc} i & 1 & 2 & 3 & 4 \\ \hline \llbracket S \rrbracket \emptyset x_i & 6 & 6 & \infty & 7 \end{array}$$

□

For the following let ϕ denote a formula with free variables from \mathcal{F} , $T = (Q, \Delta, d)$ a durational transition system and ρ an environment for the free variables in ϕ . W.l.o.g. let us assume that the names of all variables, bound or free, are distinct. Let $\mathcal{X} = \mathcal{X}_\mu \cup \mathcal{X}_\nu$ denote the set of bound variables where \mathcal{X}_μ and \mathcal{X}_ν denote the sets of variables bound by μ - and ν -operators respectively. The key idea of the translation into a hierarchical system of equations is the observation that the values

of subformulas for every state can be treated independently. Especially, if these values are known for some states, they can be removed. This leaves us with a simpler system which, however, is no longer in direct correspondence with a formula. Hence, hierarchical systems of equations are a much more flexible tool than formulas.

For every subformula ϕ' of ϕ and $q \in Q$, we introduce a new formal variable $y_{\phi',q}$. In order to compute the value of ϕ (w.r.t. T and ρ) we construct a system S consisting of a defining equation $y_{\phi',q} = f_{\phi',q}$ for every formal variable where $\phi' \notin \mathcal{F}$. By case distinction over the form of subformulas ϕ' of ϕ the right hand sides $f_{\phi',q}$ are defined as follows:

- If $\phi' \equiv p$ then $f_{\phi',q} \equiv pq$;
- If $\phi' \equiv \phi_1 \sqcap \phi_2$ then $f_{\phi',q} \equiv y_{\phi_1,q} \sqcap y_{\phi_2,q}$;
- If $\phi' \equiv \langle a \rangle \phi''$ then $f_{\phi',q} \equiv \bigsqcup_{\tau=(q,a,q') \in \Delta} (d\tau + y_{\phi'',q'})$;
- If $\phi' \equiv [a] \phi''$ then $f_{\phi',q} \equiv \bigsqcap_{\tau=(q,a,q') \in \Delta} (d\tau + y_{\phi'',q'})$;
- If $\phi' \equiv \nu x. \phi''$ then $f_{\phi',q} \equiv y_{x,q}$; additionally, $f_{x,q} \equiv y_{\phi'',q}$;
- If $\phi' \equiv \mu x. \phi''$ then $f_{\phi',q} \equiv y_{x,q}$; additionally, $f_{x,q} \equiv y_{\phi'',q}$.

Especially, the set of free variables of S is contained in $\mathcal{F}_Q = \{y_{x,q} \mid x \in \mathcal{F}, q \in Q\}$.

For each variable $x \in \mathcal{X}_\mu$, let ϕ_x denote the subexpression of ϕ such that $\mu x. \phi_x$ is a subexpression of ϕ ; accordingly for each variable $x \in \mathcal{X}_\nu$, ϕ_x denotes the subexpression of ϕ such that $\nu x. \phi_x$ is a subexpression of ϕ . W.l.o.g. ϕ itself is of the form $\mu x'. \phi'$ or $\nu x'. \phi'$ for some x' . For $x \in \mathcal{X}$, let σ_x denote the set of variables $y_{\phi',q}$ where ϕ' is a subexpression of ϕ_x and $q \in Q$. Then define \mathcal{H} as the set of all $\sigma_x, x \in \mathcal{X}$, where $\sigma_x : \mu$ if $x \in \mathcal{X}_\mu$ and $\sigma_x : \nu$ otherwise. We find:

Theorem 1 *Assume ϕ is a modal μ -formula with free variables from \mathcal{F} , $T = (Q, \Delta, d)$ is a durational transition system, and $\rho : \mathcal{F} \rightarrow D$*

an environment. Let S denote the hierarchical system of equations constructed above, and define $\rho_Q : \mathcal{F}_Q \rightarrow D$ by $\rho_Q y_{x,q} = \rho x q$. Then $\llbracket \phi \rrbracket \rho q = \llbracket S \rrbracket \rho_Q y_{\phi,q}$ for all $q \in Q$. \square

4 Efficient Computation of Solutions

We start by collecting basic algorithmic facts on hierarchical systems of equations. First note that all functions occurring in our context are indeed upward (and hence also downward) continuous. Recall that in general, this assumption is wrong since the ν -operator need not preserve upward continuity.

Example 3 Define $F : [-\infty, \infty]^2 \rightarrow [-\infty, \infty]$ by $F(x, y) = (x \sqcap y) - 1$ where postfix operation “ -1 ” denotes the predecessor function extended with $\infty - 1 = \infty$ and $0 - 1 = -\infty - 1 = -\infty$. Clearly, F is not only monotonic but upward continuous. Now let $f : [-\infty, \infty] \rightarrow [-\infty, \infty]$ be defined by $fx = \nu y. F(x, y)$. f is still monotonic. However, $fx = -\infty$ for $x < \infty$ and $fx = \infty$ for $x = \infty$. Hence, f is no longer upward continuous. \square

If the involved functions are no longer upward continuous, a fixpoint computation by iterating upto ordinal ω may no longer work. However, in [16] we show that all occurring functions remain upward (and hence also downward) continuous – provided all right hand sides of equations are taken from $\mathcal{E}_{\Omega \cup D}(\mathcal{X})$ for some set of variables \mathcal{X} .

Next, we observe:

Fact 1 *Assume S is a hierarchical system of equations $x = f_x, x \in \mathcal{X}$, over some complete partial order D whose free variables are contained in \mathcal{F} . Assume $\mathcal{X} = X \cup Y$ where $X \cap Y = \emptyset$, $\rho : \mathcal{F} \rightarrow D$ is an environment for S and $\rho_Y : Y \rightarrow D$ is given by $\rho_Y y = \llbracket S \rrbracket \rho y$. Then for all $x \in X$, $\llbracket S \rrbracket \rho x = \llbracket S_X \rrbracket (\rho + \rho_Y) x$ \square*

Fact 1 states that occurrences of variables in S may safely be replaced with the value provided for them by the solution.

Let S denote a hierarchical system of equations $x = f_x$, $x \in \mathcal{X}$, where $f_x \in \mathcal{E}_{\Omega \cup D}(\mathcal{X} \cup \mathcal{F})$. It turns out that it is sometimes convenient to assume that the right hand sides f_x of S are of one of the forms z or $z_1 \sqcap z_2$ where z, z_1, z_2 are variables or elements of D and $\sqcap \in \Omega$. Furthermore,

- Whenever $f_x \in D$ then x does not occur in any right hand side $f_{x'}$;
- Whenever $f_x \equiv z_1 \sqcap z_2$ then either z_1 or z_2 is a variable;
- $-\infty$ or ∞ do not occur as operands of “ \sqcup ” or “ \sqcap ”;
- whenever $f_x \equiv z_1; z_2$ then z_1 is a variable;
- 0 or $-\infty$ do not occur as operands of “ $+$ ”.

Systems with these properties are called *reduced*. We have:

Fact 2 *For every hierarchical system S of equations defining variables $x \in \mathcal{X}$, a reduced system S' with variables $x \in \mathcal{X}'$, can be constructed with $\mathcal{X} \subseteq \mathcal{X}'$ in time $O(|S|)$ such that for every ρ , $\llbracket S \rrbracket \rho x = \llbracket S' \rrbracket \rho x$ for all $x \in \mathcal{X}$. \square*

Example 4 Consider the hierarchical system of equations from Example 2. A corresponding reduced system S' may consist of the equations:

$$\begin{array}{ll} x_1 = y_1 \sqcap x_4 & y_1 = x_1 \sqcup 6 \\ x_2 = y_2 \sqcup 0 & y_2 = x_1 \sqcap x_2 \\ x_3 = x_3 + 1 & y_3 = x_2 + 1 \\ x_4 = x_3 \sqcap y_3 & \end{array}$$

where auxiliary variables $y_i, i = 1, 2, 3$, have been introduced to break complicated expressions into simple ones. The new hierarchy is given by the qualified sets:

$$\begin{array}{ll} \{x_1, x_2, x_3, x_4, y_1, y_2, y_3\} & : \mu \\ \{x_1, x_2, y_1, y_2\} & : \nu, \\ \{x_1, y_1\} & : \mu, \\ \{x_3\} & : \nu \end{array}$$

The following fact is well-known.

Fact 3 *The solution of any hierarchical system of equations S over some finite interval D relative to an environment ρ can be effectively computed. \square*

Let $\Psi_h(n, d)$ denote the complexity of computing the solution of a hierarchical system of size n and nesting depth d over interval D with $\#D = h$. Efficient algorithms for solving this problem (at least for $h = 2$) and their BDD based implementations (see, e.g., [5]) have extensively been considered in the literature. Among the theoretical results, we mention Andersen who shows in [3] that $\Psi_2(n, d) = O(n^d)$. Especially for the important case where the nesting depth equals one, he obtains a linear algorithm. Only recently, Long, Browne, Clarke, Jha, and Marrero came up with an improved algorithm for arbitrary finite D resulting in an upper bound $\Psi_h(n, d) = O(h \cdot n^2 \cdot n^{\frac{d}{2}})$.

In this paper we are not concerned with algorithms computing the solution of hierarchical systems over finite domains. Instead, we will decompose the computation of the solution of some system S over $[-\infty, \infty]$ into a linear sequence of problems consisting in the computation of solutions of systems over 2-point lattices. By taking the computation of a solution over 2-point lattices as a subroutine which is iteratively called, we take advantage from any improvement to the implementation of this basic algorithm. Accordingly, we will express complexity results always in terms of $\Psi_2 = \Psi_2(|S|, d)$. Here is the key theorem of this section.

Theorem 2 *The solution of a hierarchical system S of equations over $D \subseteq [-\infty, \infty]$ relative to environment ρ can be computed in time $O(|S| \cdot \Psi_2)$.*

Note that the complexity result in Theorem 2 is independent of the cardinality of D . By the translation of Section 3, we immediately obtain

\square the main result of our paper:

Theorem 3 *The interpretation of μ -formula ϕ over durational transition system T can be computed in time $O(n \cdot \Psi_2(n, d))$ where $n = |T| \cdot |\phi|$ and d is the nesting-depth of ϕ . \square*

Before we present an algorithm to prove Theorem 2, we observe that we always can compute the solutions for those variables which obtain values from an interval $[-\infty, h]$ for some $h < \infty$. Let D denote a subinterval of $[-\infty, \infty]$ and $H : D \rightarrow D \cap [-\infty, h]$ for some $h < \infty$ denote the function $Hx = x \sqcap h$. Let S denote a reduced hierarchical system of equations $x = f_x, x \in \mathcal{X}$, over interval $D \subseteq [-\infty, \infty]$. Let S_h denote the hierarchical system of equations obtained from S by replacing all occurrences of constants c with $Hc = c \sqcap h$ and all operators $\square \in \Omega$ with operators \square_h defined by $x \square_h y = H(x \square y) = (x \square y) \sqcap h$. We have:

Fact 4 $H \circ (\llbracket S \rrbracket_D \rho) = \llbracket S_h \rrbracket_{D \cap [-\infty, h]} (H \circ \rho)$ \square

H is injective on $\{d \in D \mid d < h\}$. Therefore by Facts 3 and 4 and choosing $h = 0$, we can determine in time $O(\Psi_2)$ those x with $\llbracket S \rrbracket \rho x = -\infty$.

Next we observe that whenever the solution for every variable is beyond a certain threshold h , we can safely remove all occurrences of constants $k \leq h$ as operands of “ \sqcap ” or “ \sqcup ” – provided we restrict at the same time the whole semantic domain D to interval $D \cap [h, \infty]$. Formally, we call hierarchical system S of equations over D *h -reduced* iff S is reduced and

- $k \leq h$ does not occur as an operand of “ \sqcap ” or “ \sqcup ”;
- “ $;$ ” does not occur in any right hand side;
- ∞ does not occur as an operand of “ $+$ ”.

Fact 5 *For reduced hierarchical system S over D with free variables from \mathcal{F} , environment $\rho : \mathcal{F} \rightarrow D$ where $\rho x > h$ for all $x \in \mathcal{F}$ and $\llbracket S \rrbracket_D \rho x > h$ for all $x \in \mathcal{X}$, an equivalent h -reduced system S' can be constructed in time $O(|S|)$ such that $\llbracket S \rrbracket_D \rho = \llbracket S' \rrbracket_{D \cap [h, \infty]} \rho$.*

Proof. First notice that S cannot contain any equation $x = f_x$ where $f_x \equiv z_1 \sqcap z_2$ with $z_1 \equiv c \leq h$ or $z_2 \equiv c \leq h$. Therefore, consider system S' of equations $x = f'_x, x \in \mathcal{X}$, which is obtained from S as follows.

- If $f_x \equiv c \sqcup x'$ or $f_x \equiv x' \sqcup c$ where $c \leq h$ then define $f'_x \equiv x'$;
- If $f_x \equiv \infty + x'$ or $f_x \equiv x' + \infty$ then define $f'_x \equiv \infty$;
- If $f_x \equiv y ; z$ for some variable y then define $f'_x \equiv z$;
- Otherwise, define $f'_x \equiv f_x$.

S' is h -reduced. Furthermore, $\llbracket S' \rrbracket_{D \cap [h, \infty]} \rho = \llbracket S \rrbracket_{D \cap [h, \infty]} \rho$. Thus, the assertion follows from:

Claim: Assume $h \geq 0$, S is a reduced hierarchical system of equations over D and ρ an environment for S such that $\llbracket S \rrbracket_D \rho x \geq h$ for all x . Then $\llbracket S \rrbracket_D \rho = \llbracket S \rrbracket_{D \cap [h, \infty]} \rho$. \square

Example 5 In our running example, all variables receive values > 5 . Thus, the corresponding 5-reduced system S' is obtained by replacing the equation $x_2 = y_2 \sqcup 0$ with $x_2 = y_2$.

Observe that the resulting system has no longer the same solution over $[-\infty, \infty]$ as the system from Example 4. Instead, $\llbracket S' \rrbracket_{[-\infty, \infty]} \emptyset x_i = -\infty$ for $i = 1, 2, 4$. However, $\llbracket S' \rrbracket_{[5, \infty]} \emptyset x_i$ still equals 6 for $i = 1, 2$ and 7 for $i = 4$. \square

The next proposition states that – provided we know that the values of the solution for all variables are at least h – we can easily determine the set of variables whose values match the given lower bound h .

Proposition 6 *Assume $-\infty \leq h < m$, and S is a reduced hierarchical system of equations $x = f_x, x \in \mathcal{X}$, over $D = [h, m]$, $\rho : \mathcal{F} \rightarrow D$ is an environment, and $\llbracket S \rrbracket_D \rho x \geq h$ for all x . Then the set of all x with $\llbracket S \rrbracket_D \rho x = h$ can be computed in time $O(\Psi_2)$.*

Proof. By Fact 4, we may assume that $-\infty < h$, and by Fact 5 that S is $(h-1)$ -reduced. We consider the mapping $H : [h, \infty] \rightarrow [h, h+1]$ defined by $Hx = x \sqcap (h+1)$. Applying Fact 4, we find that $H(\llbracket S \rrbracket_{[h,m]} \rho x) = \llbracket S_{h+1} \rrbracket_{[h,h+1]} (H \circ \rho) x$ for every x . Since H is injective on $\{h\}$, we conclude that it suffices to compute $\llbracket S_{h+1} \rrbracket_{[h,h+1]} (H \circ \rho)$. The latter, however, can be done in time $O(\Psi_2)$. \square

Example 6 For S' of Example 5, the test for equality with 6 results in the system:

$$\begin{array}{ll} x_1 = y_1 \sqcap x_4 & y_1 = x_1 \sqcup 6 \\ x_2 = y_2 & y_2 = x_1 \sqcap x_2 \\ x_3 = 7 & y_3 = 7 \\ x_4 = x_3 \sqcap y_3 \end{array}$$

where the hierarchy remains unchanged. The right hand sides 7 both for x_3 and y_3 are deduced from the observation that $x +_7 y = (x + y) \sqcap 7 = 7$ whenever $x \geq 6$ and $y > 0$. Thus, we conclude that $\llbracket S' \rrbracket_{[6,\infty]} \emptyset z > 6$ for $z \in \{x_3, y_3, x_4\}$. \square

Finally, we need a non-trivial estimate of a good lower bound for the values of variables given by the solution. The following proposition is crucial for proving the Lower Bound Theorem.

Proposition 7 *Assume S is a reduced hierarchical system of equations $x = f_x, x \in \mathcal{X}$, which does not contain occurrences of “+”, and let \mathcal{C} denote the set of constants occurring in right hand sides. Then for every $h \geq -\infty$ and $x \in \mathcal{X}$, $\llbracket S \rrbracket_{[h,\infty]} \rho x \in \mathcal{C} \cup \text{range}(\rho) \cup \{h, \infty\}$.*

Proof. Let $V = \mathcal{C} \cup \text{range}(\rho) \cup \{h, \infty\}$. For our proof we use the same notation as in the definition of the solution of S . We proceed by induction on $\#\mathcal{H}$. Assume $\sigma_1, \dots, \sigma_m$ are the maximal elements in $\mathcal{H} \setminus \{\mathcal{X}\}$, $\sigma = \sigma_1 \cup \dots \cup \sigma_m$ and $\mathcal{X}' = \mathcal{X} \setminus \sigma$. By induction hypothesis for every i and $x \in \sigma_i$, $\llbracket S_i \rrbracket (\rho + \beta) x \in V \cup \text{range}(\beta)$ for every $\beta : \sigma \rightarrow D$ (S_j abbreviates S_{σ_j}). Therefore, consider variables $x \in \mathcal{X}'$ (in case, $m = 0$, we obtain the base case where $\#\mathcal{H} = 1$). Let $R : (\mathcal{F} \cup \mathcal{X}' \rightarrow D) \rightarrow \sigma \rightarrow D$ be

defined by $R \beta y = \llbracket S_j \rrbracket \beta y$ whenever $y \in \sigma_j$; furthermore, define $G : (\mathcal{X}' \rightarrow D) \rightarrow \mathcal{X}' \rightarrow D$ by $G \beta x = \llbracket f_x \rrbracket (\rho + \beta + R(\rho + \beta))$. W.l.o.g. assume $\mathcal{X} : \mu$ and that the least fixpoint of G is given by $\beta = \bigsqcup_{t \geq 0} \beta^{(t)}$ where $\beta^{(0)} x = h$ and $\beta^{(t)} x = \llbracket f_x \rrbracket (\rho + \beta^{(t-1)} + R(\rho + \beta^{(t-1)}))$ for $t > 0$.

Claim: For all $x \in \mathcal{X}'$ and $t \geq 0$, $\beta^{(t)} x \in V$.

The proof of the claim follows by induction on t (using the induction hypothesis for the σ_i to deal with R). Since set V is finite, it is closed under least upper bounds and greatest lower bounds. It follows that $\llbracket S \rrbracket \rho x \in V$ for all $x \in \mathcal{X}'$. Hence by induction hypothesis for the S_i , also $\llbracket S \rrbracket \rho x \in V$ for all $x \in \sigma_i$ – which completes the proof of the assertion. \square

Theorem 4 *Assume $h \geq 0$, and S is an h -reduced hierarchical system of equations $x = f_x, x \in \mathcal{X}$, $\rho : \mathcal{F} \rightarrow [h+1, \infty]$ an environment, and $\llbracket S \rrbracket_{[h,\infty]} \rho x > h$ for all x . Let \mathcal{C} denote the set of constant right hand sides and constant operands of \sqcup or \sqcap occurring in S , and $c = \sqcap(\mathcal{C} \cup \text{range}(\rho))$ (in case, $\mathcal{C} \cup \text{range}(\rho) = \emptyset$, $c = \infty$).*

Then $\llbracket S \rrbracket_{[h,\infty]} \rho x \geq c$ for all x .

Proof. Let $b = \sqcap_x(\llbracket S \rrbracket_{[h,\infty]} \rho x)$. By assumption, $0 < b$. For a contradiction assume $b < c$. Let $\mathcal{X}' = \{x \in \mathcal{X} \mid \llbracket S \rrbracket \rho x = b\}$, and $\mathcal{X}'' = \mathcal{X} \setminus \mathcal{X}'$. Define S' as the subsystem of S with variables $x \in \mathcal{X}'$, and $\rho'' : \mathcal{X}'' \rightarrow [h, \infty]$ as the restriction of $\llbracket S \rrbracket_{[h,\infty]} \rho$ to \mathcal{X}'' . By Prop. 1, $\llbracket S' \rrbracket_{[h,\infty]} (\rho + \rho'') x = \llbracket S \rrbracket_{[h,\infty]} \rho x = b$ for all $x \in \mathcal{X}'$. Since $b \leq v$ for all $v \in V$ and also $b \leq \llbracket S \rrbracket_{[h,\infty]} \rho x$ for every $x \in \mathcal{X}$ we conclude that no right hand side $f_{x'}$ in S' may contain operator “+”. Hence, we can apply Prop. 7 which implies that $b \in V \cup \text{range}(\rho'')$. Since by assumption, $b \notin \text{range}(\rho'')$ and $b > h$, we conclude that $b \geq c$: contradiction! \square

We now have all prerequisites together to efficiently compute the solution of a hierarchical system of equations and thus prove Theorem 2.

Proof of Theorem 2. We only consider the case where $D = [-\infty, \infty]$. W.l.o.g. we can assume

that S is reduced, contains no free variables, and has only right hand sides $f_x \notin [-\infty, \infty]$. Furthermore, for some $h > -\infty$, $\llbracket S \rrbracket_{[h, \infty]} \emptyset x > h$ for all x . Then by Fact 5, we can w.l.o.g. additionally assume that S is h -reduced. Let \mathcal{C} the set of all constant operands of “ \sqcap ” and “ \sqcup ” occurring in S .

If $\mathcal{C} = \emptyset$ then by Theorem 4, $\llbracket S \rrbracket_{[h, \infty]} \emptyset x = \infty$ for all x . Therefore, assume $\mathcal{C} \neq \emptyset$ and let $c > 0$ denote the least element in \mathcal{C} . By Theorem 4, $\llbracket S \rrbracket_{[h, \infty]} \emptyset x \geq c$ for all x . We conclude that $\llbracket S \rrbracket_{[h, \infty]} \emptyset = \llbracket S \rrbracket_{[c, \infty]} \emptyset$. By Prop. 6 we know that the set of all i where $\llbracket S \rrbracket_{[c, \infty]} \emptyset x = c$ can be computed in time $O(\Psi_2)$.

Assume now that we have replaced all f_x with c whenever $\llbracket S \rrbracket_{[c, \infty]} \emptyset x \equiv c$ and afterwards constructed the corresponding reduced system. Then by Fact 5, we can remove all occurrences of c as operands of “ \sqcap ” or “ \sqcup ”. This procedure can be iterated until all constants in the system are removed. Thus, we obtain the following algorithm to compute the solution of a hierarchical system S of equations:

- (0) Compute the set of all x where $\llbracket S \rrbracket \emptyset x = -\infty$, and replace the corresponding right hand sides with $-\infty$.
- (1) Compute the set of all x where $\llbracket S \rrbracket \emptyset x = 0$, and replace the corresponding right hand sides with 0.
- (2) Construct the equivalent 0-reduced system.
W.l.o.g. no right hand side is contained in $[-\infty, \infty]$.
- (3) Compute the set \mathcal{C} of all constant operands of “ \sqcap ” or “ \sqcup ” occurring in S .
- (4) While $\mathcal{C} \neq \emptyset$ execute steps (4.1) to (4.5):
 - (4.1) Compute $c := \sqcap \mathcal{C}$.
 - (4.2) Determine the set J of all x where $\llbracket S \rrbracket_{[c, \infty]} \emptyset x = c$.
 - (4.3) For every $\in J$ replace every right hand side f_x with c .
 - (4.4) Construct the equivalent c -reduced system.

W.l.o.g. no right hand side is contained in $[-\infty, \infty]$.

(4.5) Recompute \mathcal{C} .

- (5) Replace every remaining $f_x \notin D$ with ∞ .

Since the size of system S is strictly decreasing, there are at most $|S|$ iterations of the loop. Since every iteration can be executed in time $O(\Psi_2)$ the result follows. \square

5 Conclusion

We have considered the interpretation of μ -formulas over durational transition systems. For simplicity, we only considered *forward* modalities where properties of the present state are expressed in terms of a (possibly) infinite future behavior. *Backward modalities* where properties of the present state are expressed in terms of the (always finite) past have been considered, e.g., by Vardi [19], Laroussinie and Schoebelen [12] and Laroussinie, Pinchinat and Schoebelen [13]. It is not difficult to see that (technically) these can be treated completely analogous. Our strategy consisted in two steps. First, we reduced the computation of such interpretations to the computation of the solution of a hierarchical system of equations. Secondly, we showed how the computation of the solution of a hierarchical system over the non-negative integers extended by $-\infty$ and ∞ can be obtained by iteratively computing solutions of hierarchical systems over a two-point lattice – provided the right hand sides are built up from constants and variables by operators “ \sqcap ”, “ \sqcup ”, “ $+$ ”, and “ $;$ ”.

Various problems remain open. Most of all, the expressiveness of μ -formulas equipped with interpretations over durational transition systems should be further investigated. Especially, corresponding temporal logics must be considered.

6 References

- [1] R. Alur, C. Courcoubetis, D.L. Dill: Model

- Checking in Dense Real Time. Inform. and Comput. 104, 2–34, 1993
- [2] R. Alur, D.L. Dill: A Theory of Timed Automata. TCS 126, 183–235, 1994
- [3] H.R. Andersen: Model checking and Boolean Graphs. Proc. ESOP’92, LNCS 582, 1–19, 1992
- [4] O. Bernholtz, M.V. Vardi, P. Wolper: An Automata–Theoretic Approach to Branching–Time Model Checking (Extended Abstract). Proc. Conf. Computer Aided Verification, LNCS 818, 142–155, 1994
- [5] J.R. Burch, E.M. Clarke, K.L. McMillan: Symbolic Model Checking 10^{20} States and Beyond. Inf. and Comput. 98, 142–170, 1992
- [6] E.A. Emerson, C.S. Jutla: Tree Automata, Mu–Calculus and Determinacy (Extended Abstract). Proc. 32nd IEEE Symp. on FOCS, 368–377, 1991
- [7] E.A. Emerson: Real–Time and the Mu–Calculus. Proc. REX Workshop on “Realtime: Theory and Practice”, LNCS 600, 176–194, 1992
- [8] E.A. Emerson, C.S. Jutla, A.P. Sistla: On Model–Checking for Fragments of μ –Calculus. Proc. Conf. on Computer Aided Verification, LNCS 697, 385–396, 1993
- [9] R. Gorrieri, M. Roccetti, E. Stancampiano: A Theory of Processes with Durational Actions. TCS 140, 73–94, 1995
- [10] T.A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine: Symbolic Model Checking for Real–Time Systems. Inform. and Comput. 111, 193–244, 1994
- [11] D. Kozen: Results on the Propositional μ –Calculus. TCS 27, 333–354, 1983
- [12] F. Laroussinie, Ph. Schoebelen: A Hierarchy of Temporal Logics with Past. Proc. 11th STACS, LNCS 775, 47–58, 1994; long version submitted to TCS
- [13] F. Laroussinie, S. Pinchinat, Ph. Schoebelen: Translations between Modal Logics of Reactive Systems. Special Issue on AMAST’93, TCS 140, 53–71, 1995
- [14] D.E. Long, A. Browne, E.M. Clarke, S. Jha, W.R. Marrero: An Improved Algorithm for the Evaluation of Fixpoint Expressions. Proc. Conf. on Computer Aided Verification, LNCS 818, 338–350, 1994
- [15] H. Seidl: Least Solution of Equations over \mathcal{N} . Proc. ICALP’94, LNCS 820, 400–411, 1994
- [16] H. Seidl: A Modal μ –Calculus for Durational Transition Systems. Tech. Report 95–08, University of Trier, June 1995.
- [17] H. Seidl: Integer Constraints to Stop Deforestation. To appear in: Proc. ESOP’96, 1995
- [18] C. Stirling: Modal and Temporal Logics. In: S. Abramsky, D. Gabbay, and T. Maibaum (eds.): Handbook of Logic in Computer Science, 477–563, 1992
- [19] M. Vardi: A Temporal Fixpoint Calculus. Proc. 15th POPL, 250–259, 1988
- [20] M.V. Vardi, P. Wolper: Automata–Theoretic Techniques for Modal Logics of Programs. JCSS 32, 183–221, 1986