ID

ID–TR: 1994-145

DTU

ISSN 0902–2821

Henrik Reif Andersen

A Polyadic Modal μ-Calculus

Department of Computer Science
Technical University of Denmark
Building 344
DK-2800 Lyngby, Denmark

Instituttet for Datateknik
Danmarks Tekniske Universitet
Bygning 344
2800 Lyngby

ID

ID–TR: 1994-145

DTU

ISSN 0902–2821

Henrik Reif Andersen

A Polyadic Modal μ-Calculus

October 1994

Department of Computer Science
Technical University of Denmark
Building 344
DK-2800 Lyngby, Denmark

Telephone
+45 45 93 33 32

Instituttet for Datateknik
Danmarks Tekniske Universitet
Bygning 344
2800 Lyngby


Printed by LTT Tryk, DTU, Lyngby
October 1994

A Polyadic Modal μ-Calculus

Henrik Reif Andersen*

Abstract

The propositional μ-calculus of Kozen extends modal logic with fixed points to achieve a powerful logic for expressing temporal properties of systems modelled by labelled transition systems. We further extend Kozen's logic with *polyadic* modalities to allow for expressing also quite naturally behavioural relations like bisimulation equivalence and simulation preorders.

We show that the problem of model checking is still efficiently decidable, giving rise to efficient worst-case algorithms for verifying the infinity of behavioural relations expressible in this *polyadic modal μ-calculus.* Some of these algorithms compete in efficiency with carefully hand-crafted algorithms found in the literature. In spite of this result, the validity problem turns out to be highly undecidable. This is in contrast to the propositional μ-calculus where it is decidable in deterministic exponential time. It follows as a corollary, that – also in contrast to the propositional μ-calculus – the polyadic modal μ-calculus does not have the finite-model property nor does it admit a finitary axiomatisation.

1 Introduction

Since Kozen introduced the propositional μ-calculus [Koz83] as an extension of dynamic logic, much effort has been put in solving traditional logical questions concerning expressiveness, decidability and axiomatisations for this syntactically simple, yet powerful logic. The logic has turned out to be useful in verifying dynamic properties of concurrent systems and has spawned research into model checking algorithms and proof systems for relating processes and assertions, e.g. [EL86, Lar88, SW91, Win89, Cle90, AC88, CS91, And94, Sti85, Win86, Win90, BS90b, AW92].

We briefly review the more theoretical of these results as relevant for this paper – consult e.g. Stirling [Sti92] for a more detailed account. Although Kozen's propositional μ-calculus has expressive power comparable to Rabin's non-elementary decidable, second order monadic theory of n-successors (SnS), [Hüt90], [Niw88], the validity problem is decidable. In fact, it is decidable in deterministic exponential time [EJ88], which is

*Department of Computer Science, Technical University of Denmark, Building 344, 2800 Lyngby, Denmark. E-mail: hra@id.dth.dk. Phone: +45 45933332. Fax: +45 42884530. Supported by the Danish Technical Research Council.

surprisingly efficient considering the expressiveness. It is further known to possess the finite-model property: Any satisfiable assertion is satisfiable in a finite model [Koz88]. Recently, also a finitary axiomatisation (although more involved than the one originally suggested by Kozen [Koz83]) has been found by Walukiewicz [Wal93].

We shall present an extension of Kozen's logic with polyadic modalities, which we call the *polyadic modal μ-calculus*, and change models to *tuples* of transition systems, thereby getting closer to the *relational* μ-calculus [dBdR73] and *predicate* μ-calculus [HP73].[1] As we will prove, this dramatically changes the picture: The validity problem becomes highly undecidable, more precisely Π^1_1-hard (see Rogers [Rog67] for a definition). However, the model checking problem remains efficiently decidable for finite models, which is the pragmatic reason for considering the logic. In fact, global and local model checking algorithms for the propositional μ-calculus can still be applied, and they will provide efficient algorithms for verifying the infinity of behavioural relations expressible in the polyadic modal μ-calculus. Some of these algorithms compete in efficiency with carefully hand-crafted algorithms found in the literature.

As it will be shown the Π^1_1-hardness and the decidability of model checking together implies that pμK *cannot* have the finite-model property, nor will it admit any finitary (i.e. recursively enumerable) axiomatisation.

2 The Polyadic Modal μ-Calculus

In this section we will introduce the logic, the models and state a few basic results. We shall assume the existence of an infinite set of *actions*, Act. Now, as models we consider tuples of pointed labelled transition systems:

Definition 1 A *pointed, labelled transition system* is a triple $T = (S, \rightarrow, i)$, where S is a set of *states*, $\rightarrow \subseteq S \times Act \times S$ is a *transition relation*, and $i \in S$ is an *initial state*, such that all states in S are reachable from i through transitions of $\rightarrow$. We shall write $s \stackrel{a}{\rightarrow} s'$ for $(s, a, s') \in \rightarrow$. We abbreviate a *tuple of pointed, labelled transition systems* $(T_1, \ldots, T_n)$ as $\vec{T}$. $\Box$

In the sequel, we will use the term "transition system" as an abbreviation for "pointed, labelled transition system." A tuple of transition systems $\vec{T}$ can be thought of as a parallel composition of n sequential systems that do not interact at all. With this interpretation the tupling can form an important ingredient in defining other more appropriate parallel compositions (see for example Winskel [Win84] or Andersen [And93]).

The assertions of *polyadic modal μ-calculus*, pμK, are constructed from the following grammar:

$$A \ ::= \ \neg A \mid A_0 \vee A_1 \mid \langle a \rangle_i A \mid X \mid \mu X.A$$

where $a \in Act$, i is a non-zero natural number $1, 2, \ldots$, X ranges over a set $AssnVar$ of assertion variables, and the minimum fixed points $\mu X.A$ are formed subject to the

[1] In choosing the name the *polyadic modal μ-calculus* for our logic, we are following Stirling [Sti92] and using the adjective *modal* instead of *propositional* to reflect the close relationship with modal logic.

syntactic monotonicity condition that X must only appear inside an even number of negations in A. Standard abbreviations like $A_0 \wedge A_1 =_{\text{def}} \neg(\neg A_0 \vee \neg A_1)$, $A_0 \rightarrow A_1 =_{\text{def}} \neg A_0 \vee A_1$, $[a]_i A =_{\text{def}} \neg\langle a\rangle_i \neg A$, $ff =_{\text{def}} \mu X.X$, $tt =_{\text{def}} \neg ff$, and the maximum fixed-point operator $\nu X.A =_{\text{def}} \neg\mu X.\neg A[\neg X/X]$ will be used freely.

The only new construction compared to Kozen's logic is the indexed diamond-modality $\langle a\rangle_i A$, which is going to hold for tuples of global states for which the i'th system can make an a-transition to a state such that the new global state satisfy A. Hence, these modalities allow local properties to be expressed directly.

Notice, that the subset of assertions achieved by fixing i to 1 coincides with Kozen's logic, which we refer to as μK.

Definition 2 The *arity* of an assertion A, denoted by $ar(A)$, is the maximum i that occurs in A and it is 0 if A contains no modalities. □

Let $\mathbf{1}$ be the one-point transition system $(\{\bullet\}, \emptyset, \bullet)$. For any n-tuple of transition systems $\vec{T}$ let $f_m(\vec{T})$ be defined as follows:

$$f_m(T_1,\ldots,T_n) \;=\; \begin{cases} (T_1,\ldots,T_n) & \text{if } n \geq m \\ (T_1,\ldots,T_n,\underbrace{\mathbf{1},\ldots,\mathbf{1}}_{m-n}) & \text{if } n < m \end{cases}$$

I.e. $f_m(\vec{T})$ has arity at least m, and agrees with $\vec{T}$ on the first n coordinates.

Given an assertion A and a tuple $\vec{T}$ of arity $n \geq ar(A)$ with states $S_1,\ldots,S_n$ and transition relations $\rightarrow_1,\ldots,\rightarrow_n$, the semantics $[\![A]\!]_{\vec{T}}\rho$ will be a subset of $S =_{\text{def}} S_1 \times \cdots \times S_n$ defined relative to an environment ρ assigning subsets to the free variables of A. Hence, by structural induction, define $[\![A]\!]_{\vec{T}}\rho$ as follows (we leave out the subscript $\vec{T}$ for convenience):

$$\begin{aligned} [\![\neg A]\!]\rho &= S \setminus [\![A]\!]\rho \\ [\![A_0 \vee A_1]\!]\rho &= [\![A_0]\!]\rho \cup [\![A_1]\!]\rho \\ [\![\langle a\rangle_i A]\!]\rho &= \{(s_1,\ldots,s_m) \in S \mid \exists s_i'.\ s_i \xrightarrow{a}_i s_i' \text{ and } (s_1,\ldots,s_i',\ldots s_m) \in [\![A]\!]\rho\} \\ [\![X]\!]\rho &= \rho(X) \\ [\![\mu X.A]\!]\rho &= \bigcap\{U \subseteq S \mid [\![A]\!]\rho[U/X] \subseteq U\}, \end{aligned}$$

where in the last clause we have used the Knaster-Tarski [Tar55] characterisation of the least fixed point as the intersection of all prefixed points. We extend the semantics to tuples $\vec{T}$ of arity $n < ar(A)$ by the use of $f_{ar(A)}$. Hence, for such a $\vec{T}$ take $[\![A]\!]_{\vec{T}}\rho =_{\text{def}} [\![A]\!]_{f_{ar(A)}(\vec{T})}\rho$.

Definition 3 A tuple $\vec{T}$ with initial states $i_1,\ldots,i_n$ *satisfy* a closed assertion A, written $\vec{T} \models A$, if and only if, $(i_1,\ldots,i_n) \in [\![A]\!]_{\vec{T}}\rho$, for all ρ. A closed assertion A is *valid*, written $\models A$, if and only if, for all tuples of transition systems $\vec{T}$, $\vec{T} \models A$, and it is *satisfiable* if there exists a tuple $\vec{T}$ satisfying A. □

Notice, that as usual in classical logics with negation, A is valid, if and only if, its negation $\neg A$ is not satisfiable.

We shall in the sequel often make use of the familiar notion of *bisimulation* on transition systems introduced by Park [Par81] and explored in a process algebraic framework by Milner [Mil89]. Recall, that two transition systems $T_1 = (S_1, \rightarrow_1, i_1)$ and $T_2 = (S_2, \rightarrow_2, i_2)$ are bisimilar, written $T_1 \sim T_2$, if there exists a *bisimulation* between i_1 and i_2, i.e. a relation $R \subseteq S_1 \times S_2$ with $(i_1, i_2) \in R$, such that for all $(s_1, s_2) \in R$,

- if $s_1 \stackrel{a}{\rightarrow}_1 s_1'$ then $\exists s_2'.\ s_2 \stackrel{a}{\rightarrow}_2 s_2'$ & $(s_1', s_2') \in R$, and
- if $s_2 \stackrel{a}{\rightarrow}_2 s_2'$ then $\exists s_1'.\ s_1 \stackrel{a}{\rightarrow}_1 s_1'$ & $(s_1', s_2') \in R$.

In fact, $\sim$ is the maximum fixed point of the monotonic function F on $Pow(S_1 \times S_2)$ given by $F(R) =$ "the set of (s_1, s_2) satisfying the two conditions above." The following fact on the invariance of satisfaction under $\sim$ is easily shown:

Lemma 1 *If $\vec{T}$ and $\vec{T}'$ are n-tuples of transition systems with $T_i \sim T_i'$ for $1 \leq i \leq n$, then, for all closed assertions A, $\vec{T} \models A$, if and only if, $\vec{T}' \models A$.*

It is easy to see that $\sim$ is an equivalence relation on transition systems. If $T = (S, \rightarrow, i)$ is a transition system, let the quotient under $\sim$ be defined by $T/\sim = (S/\sim, \rightarrow/\sim, [i]_\sim)$, where $[s]_\sim$ denotes the equivalence class containing s, $S/\sim = \{[s]_\sim \mid s \in S\}$, and $[s]_\sim(\stackrel{a}{\rightarrow}/\sim)[s']_\sim$, iff, $\exists s''.\ s \stackrel{a}{\rightarrow} s''$ & $s'' \sim s'$. Then clearly, $(T/\sim) \sim T$, hence, according to the lemma, $T/\sim$ satisfies the same assertions as T. Moreover, for two bisimilar transition systems T_1 and T_2 their quotients will be isomorphic:

Lemma 2 *If $T_1 \sim T_2$ then $T_1/\sim$ and $T_2/\sim$ are isomorphic (as sets).*

Example 1 Bisimilarity gives an example of what can be expressed in pμK. Given a finite set $L \subseteq Act$ of actions. Take

$$\mathcal{B} \equiv \nu R. \bigwedge_{a \in L} ([a]_1\langle a\rangle_2 R \wedge [a]_2\langle a\rangle_1 R).$$

Then $\mathcal{B}$ is an assertion of arity 2, and it is not hard to prove that two transition systems T_1 and T_2 with labels in L are bisimilar, if and only if, $(T_1, T_2) \models \mathcal{B}$. □

The construction of $\mathcal{B}$ is quite simple. Similarly, assertions for other equivalences and preorders can be constructed directly from their defining equations. The only annoyance is that we have to restrict the actions to a predefined *finite* set. What is really needed is a universal quantification over *all* actions. We shall later, in section 4, show that model checking will still be decidable for certain classes of such assertions. However, for the negative results we are going to show on validity we shall restrict attention to the simpler language, thereby making the results stronger.

3 Validity

In this section we will prove that validity is undecidable, or to be more precise, show that the set of closed assertions A for which $\models A$ holds is Π^1_1-hard (see Rogers [Rog67] for a definition).

The proof of undecidability is by a reduction from the *Recurring Domino Problem* due to Harel [Har85] (Harel's problem R2). We use the problem in the following version: A *recurring domino system* is a tuple $\mathcal{D} = (D, H, V, d_0, d_r)$ consisting of a finite set of dominoes D, rules for how dominoes can be arranged horizontally $H \subseteq D \times D$ and vertically $V \subseteq D \times D$, an initial domino $d_0 \in D$, and a recurrence domino $d_r \in D$. A tiling of the plane is a function $f : \mathbb{N} \times \mathbb{N} \to D$ satisfying

1. $f(0,0) = d_0$,
2. $\forall i, j \in \mathbb{N}.\ (f(i,j), f(i+1,j)) \in H$,
3. $\forall i, j \in \mathbb{N}.\ (f(i,j), f(i,j+1)) \in V$,
4. the set $\{j \in \mathbb{N} \mid f(0,j) = d_r\}$ is infinite.

Without the last condition we obtain a version of the undecidable, (co-r.e.) Domino Problem of Wang [Wan61]. This last *recurrence condition* makes the problem shift from the arithmetical to the analytical hierarchy: Harel showed that the set of encodings of recurring domino systems for which a valid tiling exists is Σ^1_1-complete.

The reduction proceeds by computably constructing for each recurring domino system $\mathcal{D}$ an assertion $\Theta(\mathcal{D})$ that will be satisfiable, if and only if, $\mathcal{D}$ has a solution. Hence, having described such a computation, satisfiability of pμK is at least as hard as the recurring domino problem, i.e. Σ^1_1-hard (cf. Rogers [Rog67, Chap.14.8]).[2]

In constructing $\Theta(\mathcal{D})$ we shall make heavy use of Lemma 1 and Lemma 2. First, however, we observe how a tiling can be encoded as a transition system. Each grid point will be modelled by a state that can reach its neighbour to the right through an h-transition and its neighbour above by a v-transition. The domino located at the grid point will be modelled by a transition out of the state with the domino as label, see Figure 1. Similarly, any transition system that forms an h-v-grid with a domino transition at each grid point, gives immediately rise to a tiling.

The requirements that make a transition system look like a grid will be expressed "up to bisimilarity" – in view of Lemma 1 we can do no better. As an example let us consider how to express that the initial state must have exactly one h-transition. The best we can hope for is to express that there is at least one h-transition and that all h-transitions have bisimilar successor states. However, the last part cannot be expressed directly. Intuitively, the reason is the inability of the box-modality to relate different successors. Instead, we

[2] Recall that the argument is: Suppose by way of contradiction that validity for pμK belonged to some lower complexity class, e.g. Σ^0_n. But then, through the computable reduction, also the Recurring Domino Problem would belong to this class. A contradiction.

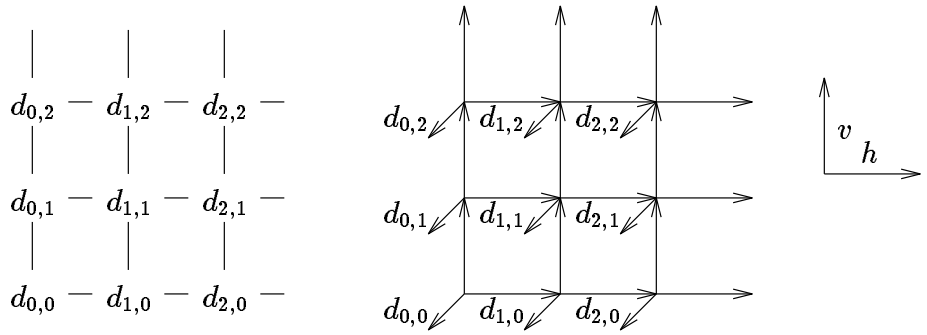

Figure 1: A colouring and its corresponding transition system. Horizontal transitions are labelled h, vertical transitions v.

employ the crucial trick of taking *two* copies of the transition system. If T is the transition system for which we want to express that it has a unique h-transition, we construct an assertion for the tuple (T, T) that will ensure this. The assertion is

$$\mathrm{UNIQ}_h =_{\mathrm{def}} \langle h\rangle_1 tt \wedge [h]_1[h]_2\mathcal{B},$$

where $\mathcal{B}$ is the assertion from Example 1 expressing bisimilarity. Clearly, if $(T,T) \models \langle h\rangle_1 tt \wedge [h]_1[h]_2\mathcal{B}$ then T can perform an h-action, and all h-successors of the first copy of T must be bisimilar to all h-successors of the second. This implies that the quotient $T/\sim$ has *exactly one* h-transition.

By similar arguments we can use the assertion

$$\mathrm{COMM} =_{\mathrm{def}} [h]_1[v]_1[v]_2[h]_2\mathcal{B}$$

to ensure that a pair of states – assuming all states have both h and v successors – constitute the lower-left point of an h-v square, again, of course, only up to bisimilarity. Now, let ALWAYS(A) be the assertion $\nu X.\bigwedge_{a\in\{h,v\}\cup D}[a]_1[a]_2X \wedge A$ expressing that A holds everywhere when the two transition systems perform matching transitions. Furthermore, let UNIQ_D be the straightforward generalisation of UNIQ_h that ensures that exactly one of the domino transitions are possible. Take

$$\mathrm{GRID(D)} =_{\mathrm{def}} \mathrm{ALWAYS}(\mathrm{UNIQ}_h \wedge \mathrm{UNIQ}_v \wedge \mathrm{UNIQ}_D \wedge \mathrm{COMM})$$

and observe that if $(T,T) \models \mathrm{GRID}(\mathcal{D})$ then $T/\sim$ is really a grid like the one in Figure 1.

What remains are the conditions 1-4 ensuring that the tiling is valid. They are handled by the following assertions defined relative to a recurring domino system $\mathcal{D} = (D, H, V, d_0, d_r)$:

1. $\langle d_0\rangle_1 tt$,
2. $\mathrm{ALWAYS}(\bigwedge_{d,d'\in D}\langle d\rangle_1 tt \wedge \langle h\rangle_1\langle d\rangle_1 tt \rightarrow ((d,d') \in H))$,

3. ALWAYS$(\bigwedge_{d,d' \in D} \langle d \rangle_1 tt \wedge \langle v \rangle_1 \langle d \rangle_1 tt \rightarrow ((d,d') \in V))$,

4. $\nu X.(\langle v \rangle_1 X \wedge \mu Y.(\langle v \rangle_1 Y \vee \langle d_r \rangle_1 tt))$,

where in condition 2 and 3 the membership tests, depending on the actual H and V, must be replaced by tt or ff as appropriate. Let RULES($\mathcal{D}$) be the conjunction of these four. Then take

$$\Theta(\mathcal{D}) = \text{GRID}(\mathcal{D}) \wedge \text{RULES}(\mathcal{D}) \wedge \mathcal{B}.$$

Suppose (T_1, T_2) satisfies this assertion of arity 2. Then, clearly by the last conjunct, $T_1 \sim T_2$. Hence, by Lemma 2 $(T_1/\sim)$ and $(T_2/\sim)$ are isomorphic and by Lemma 1, $(T_1/\sim, T_2/\sim) \models \Theta(\mathcal{D})$ showing, by the construction of GRID($\mathcal{D}$) and RULES($\mathcal{D}$), that $\mathcal{D}$ has a solution. For the other direction assume that $\mathcal{D}$ has a solution. Then it follows easily from the construction that (T, T) satisfy $\Theta(\mathcal{D})$ and therefore $\Theta(\mathcal{D})$ is satisfiable. We have reduced the Recurring Domino Problem to satisfiability in pμK, which, by application of Harel's result, provides a proof of:

Theorem 1 *The satisfiability problem for pμK is Σ^1_1-hard.*

It follows as an immediate corollary that the validity problem for pμK is Π^1_1-hard.

Corollary 1 *There exists no finitary (that is, recursively enumerable) axiomatisation of pμK, and pμK does not have the finite model property.*

The first Corollary follows from A being valid, if and only if, $\neg A$ is satisfiable. The proof of the last Corollary depends on model checking being decidable for finite models (see Theorem 2 below), and proceeds by way of contradiction. Assume that any satisfiable assertion were satisfiable in a finite model. Using the model checker all pairs of finite models and assertions could be enumerated, and computably checked for satisfaction. But this gives a recursive enumeration of all satisfiable assertions, implying that the satisfiability problem is in Σ^0_1 contradicting the Σ^1_1-hardness.

Readers familiar with the notion of alternation depth, introduced by Emerson and Lei [EL86], will observe that $\Theta(\mathcal{D})$ only has alternation depth one, thus is among the simplest of the assertions of pμK. It is unknown to the author whether the above lower bound, which seems to be tight for alternation depth one (i.e. validity for this subset of assertions is Π^1_1-*complete*), is also tight for higher alternation depths.

4 Model Checking

Given an n-ary tuple $\vec{T}$ of transition systems $(T_1, \ldots, T_n)$, the product $pr(\vec{T})$ of these is the labelled transition system $(S, \rightarrow, i)$ defined by taking

$$\begin{array}{rcl} S & =_{\text{def}} & S_1 \times \cdots \times S_n \\ i & =_{\text{def}} & (i_1, \ldots, i_n) \end{array}$$

and defining $\rightarrow \subseteq S \times (Act \times \mathbb{N}) \times S$ by

$$(s_1, \ldots, s_n) \stackrel{a,i}{\rightarrow} (s'_1, \ldots, s'_n) \;\Leftrightarrow_{\text{def}}\; s_i \stackrel{a}{\rightarrow}_i s'_i \text{ and } \forall j (1 \leq j \leq n).j \neq i \;\Rightarrow\; s_j = s'_j.$$

Similarly, let $pr(A)$ be the homomorphic map on assertions with $pr(\langle a \rangle_i A) = \langle (a,i) \rangle pr(A)$. Then it is not hard to see that $pr(\vec{T})$ is a transition system of arity 1 and $pr(A)$ is an assertion of arity (at most) 1 with actions in $Act \times \mathbb{N}$ such that

$$\vec{T} \models A \;\Leftrightarrow\; pr(\vec{T}) \models pr(A).$$

Hence, if each of the T_i are finite, so is $pr(\vec{T})$ and we have reduced the problem of model checking in pμK to model checking in μK. Using the algorithm from Andersen [And94] or Cleaveland, Dreimüller and Steffen [CDS92], we get from this reduction:

Theorem 2 *There exists an algorithm for deciding $\vec{T} \models A$, for A closed and $\vec{T}$ an m-tuple of finite transition systems with states $S_1, \ldots, S_m$, which runs in time $O(|A|^k(|S_1| \cdots |S_m|)^{k-1}|T_1| \cdots |T_m|)$, where k is the alternation depth of A.*

(See Emerson and Lei [EL86] for a definition of alternation depth, or Andersen [And93] for a slightly stronger notion of alternation. The theorem is valid for this stronger notion.)

Actually, we can extend this result to certain assertions from an even richer logic. First, we add to the logic simultaneous fixed points and quantification over actions. Let $\alpha, \beta, \ldots$ range over a set of action variables. The syntax of pμK$_\exists$ contains the constructions of pμK extended with

$$A \;::=\; \cdots \mid \exists \alpha.A \mid (\alpha = a) \mid \langle \alpha \rangle_i A \mid \mu X_1, \ldots, X_n.A_1, \ldots, A_n \downarrow X_j$$

where $1 \leq j \leq n$ and again X_i must appear under an even number of negations inside each of the $A_{i'}$'s. To give the semantics for pμK$_\exists$, environments ρ now also assigns actions to action variables, hence: $[\![\exists \alpha.A]\!]\rho = \bigcup\{[\![A]\!]\rho[a/\alpha] \mid a \in Act\}$, and

$$[\![\langle \alpha \rangle_i A]\!]\rho = \{(s_1, \ldots, s_m) \in S \mid \exists s'_i.\; s_i \stackrel{\rho(\alpha)}{\rightarrow}_i s'_i \text{ and } (s_1, \ldots, s'_i, \ldots s_m) \in [\![A]\!]\rho\}.$$

The semantics of the simultaneous fixed point $\mu X_1, \ldots, X_n.A_1, \ldots, A_n \downarrow X_j$ is a straightforward extension: The j'th projection of the minimum fixed point is given by the Knaster-Tarski fixed-point theorem. (The fixed-point assertion $\mu X.A$ is now an abbreviation for $\mu X.A \downarrow X$. The context should avoid any confusion between the two notations.) As before we shall freely use standard abbreviations, e.g. $\forall \alpha.A =_{\text{def}} \neg \exists \alpha. \neg A$.

Table 1 gives examples of behavioural relations that can be expressed in pμK$_\exists$.

5 Quotienting

In order to describe a model checking algorithm for pμK$_\exists$, we shall introduce a *quotienting operator*, inspired by Andersen and Winskel [AW92] (see also Larsen and Xinxin [LX90]),

Bisimulation [Mil89]	$\mathcal{B} =_{\text{def}} \nu R.\forall\alpha.[\alpha]_1\langle\alpha\rangle_2 R \wedge [\alpha]_2\langle\alpha\rangle_1 R$
Weak bisimulation [Mil89]	$\mathcal{W} =_{\text{def}} \nu R.\forall\alpha.[\alpha]_1\langle\langle\alpha\rangle\rangle_2 R \wedge [\alpha]_2\langle\langle\alpha\rangle\rangle_1 R$
Obs. congruence [Mil89]	$\mathcal{C} =_{\text{def}} \forall\alpha.[\alpha]_1\langle\tau\rangle_2^*\langle\alpha\rangle_2\langle\tau\rangle_2^*\mathcal{W} \wedge [\alpha]_2\langle\tau\rangle_1^*\langle\alpha\rangle_1\langle\tau\rangle_1^*\mathcal{W}$
Ready simulation [Blo89]	$\mathcal{R} =_{\text{def}} \nu R.\forall\alpha.[\alpha]_1\langle\alpha\rangle_2 R \wedge (\langle\alpha\rangle_1 tt \leftrightarrow \langle\alpha\rangle_2 tt)$
Inv. ready sim.	$\mathcal{R}^{-1} =_{\text{def}} \nu R.\forall\alpha.[\alpha]_2\langle\alpha\rangle_1 R \wedge (\langle\alpha\rangle_1 tt \leftrightarrow \langle\alpha\rangle_2 tt)$
Ready bisimulation [Blo89]	$\mathcal{RB} =_{\text{def}} \mathcal{R} \wedge \mathcal{R}^{-1}$
Simulation preorder [Mil89]	$\mathcal{S} =_{\text{def}} \nu R.\forall\alpha.[[\alpha]]_1\langle\langle\alpha\rangle\rangle_2 R$
Prebisimulation [Mil81]	$\mathcal{P} =_{\text{def}} \nu R.\forall\alpha.[\alpha]_1\langle\alpha\rangle_2 R \wedge (Cv_1 \rightarrow (Cv_2 \wedge [\alpha]_2\langle\alpha\rangle_1 R))$
Abbreviations: $\langle\alpha\rangle_i^* A =_{\text{def}} \mu X.\langle\alpha\rangle_i X \vee A$, $\langle\langle\alpha\rangle\rangle_i A =_{\text{def}} ((\alpha = \tau) \wedge A) \vee \langle\tau\rangle_i^*\langle\alpha\rangle_i\langle\tau\rangle_i^* A$, $Cv_i =_{\text{def}} \mu X.[\tau]_i X$	

Table 1: Some behavioural relations as assertions in pμK$_\exists$

that can reduce the arity of an assertion by quotienting with a transition system. By repeated application of this operator, the result will be an assertion of arity zero. I.e. an assertion including only the boolean connectives and fixed points. Applying algorithms from [And94, And93] such an assertion can be evaluated efficiently, in a local or global fashion, to achieve the answer to the model checking problem. This will be described in more details in the full paper. Here we only give a sketch. The main point is that an assertion of arity zero can be evaluated globally in time $O(n^{k-1}m)$ and locally in time $O(n^{k-1}m\log n)$, where n is the number of variables in the assertion, $k \geq 1$ is the alternation depth, and $m \geq n$ is the total size of the assertion. (Alternatively, the global algorithm of Cleaveland, Dreimüller and Steffen [CDS92] and the local of Larsen [Lar92] could be applied.)

Hence, it is of major concern to keep the size of these small. It is for this purpose, we included simultaneous fixed points in pμK$_\exists$. Moreover, we shall convert any assertion into *positive, normal form*. This is done by allowing as proper operators $[\alpha]_i, \wedge, \forall, (\alpha \neq a)$, and ν pushing all negations inwards, ending up with a negation free assertion. We then use a trick of Arnold and Crubille [AC88] and add a variable for each action-variable closed subassertion. Now, as argued in [And94, And93] if there are no action quantifiers and action variables at all, the size of a quotient $A/_i s$ with respect to a state s in a transition system T_i, is $O(|A||T_i|)$.[3] In the presence of the quantifiers and action variables, the result might get exponentially larger, since subassertions with free action variables might get duplicated in the process.[4] However, we will give some simple optimised quotienting rules that will turn out to give succinct quotients for all the relations of Table 1.

In presenting the quotienting operator (see Figure 2), we are using a little extra notation: If $\vec{X} = (X_1, \ldots, X_m)$ then $\vec{X}_s = (X_{1s}, \ldots, X_{ms})$, and if $\vec{A} = (A_1, \ldots, A_m)$ then $\vec{A}/_i s = (A_1/_i s, \ldots, A_m/_i s)$. The following theorem is a mild generalisation of [AW92,

[3] Actually, for this statement to be true for arbitrary alternation depths, a slightly generalised version of the simultaneous fixed points, allowing reference to several projections from the same fixed-point assertion, is needed. See [And93] for details.

[4] The reader is encouraged to try, for instance, $\exists\alpha.\langle a\rangle_1 \cdots \langle a\rangle_1\langle\alpha\rangle_1 A/_1 i_1$, where i_1 points an m-state transition system with a-transitions between all states.

$$
\begin{array}{rcl}
A_0 \vee A_1/_i s & = & (A_0/_i s) \vee (A_1/_i s) \\
\langle a \rangle_j A/_i s & = & \begin{cases} \bigvee\{A/_i s' \mid s \xrightarrow{a}_i s'\} & \text{if } i = j \\ \langle a \rangle_j (A/_i s) & \text{if } i \neq j \end{cases} \\
\langle \alpha \rangle_j A/_i s & = & \begin{cases} \bigvee\{(\alpha = a) \wedge (A/_i s') \mid s \xrightarrow{a}_i s'\} & \text{if } i = j \\ \langle \alpha \rangle_j (A/_i s) & \text{if } i \neq j \end{cases} \\
\exists \alpha . A/_i s & = & \exists \alpha . (A/_i s) \\
(\alpha = a)/_i s & = & (\alpha = a) \\
X/_i s & = & X_s \\
(\mu \vec{X} . \vec{A} \downarrow X_j)/_i s & = & \mu \vec{X}_{s_1} \ldots \vec{X}_{s_n} . (\vec{A}/_i s_1) \ldots (\vec{A}/_i s_n) \downarrow X_{j s}
\end{array}
$$

Optimised rules:

$$
\begin{array}{rcl}
\exists \alpha . \langle \alpha \rangle_j A/_i s & = & \begin{cases} \bigvee\{A[a/\alpha]/_i s' \mid s \xrightarrow{a}_i s'\} & \text{if } i = j \\ \exists \alpha . \langle \alpha \rangle_j (A/_i s) & \text{if } i \neq j \end{cases} \\
\exists \alpha . A_0 \vee A_1/_i s & = & (\exists \alpha . A_0/_i s) \vee (\exists \alpha . A_1/_i s)
\end{array}
$$

Figure 2: The quotienting operator for the "disjunctive" fragment. The conjunctive fragment $(\wedge, [], \forall\alpha, \nu)$ is dual. Note: s is a state in a finite transition system T_i with transition relation $\rightarrow_i$ and states $\{s_1, \ldots, s_n\}$.

Theorem 8] (see also Larsen and Xinxin [LX90]):

Theorem 3 *Assume A is a closed assertion and for $1 \leq i \leq m$, T_i is a finite transition system. Then, $(T_1, \ldots, T_m) \models A$, if and only if, $(T_1, \ldots, T_{j-1}, T_{j+1}, \ldots, T_m) \models A/_j i_j$, where the quotienting is with respect to the initial state i_j of T_j.*

For all the relations of Table 1 it can readily be shown that the number of variables of the resulting arity zero assertions are $O(n_1 n_2)$ for transition systems T_1 and T_2 with n_1 respectively n_2 states, and the total sizes of the resulting arity zero assertions are all $O(m_1 m_2)$ for transition systems T_1 and T_2 with m_1 respectively m_2 transitions. Moreover, these assertion *do not contain any remaining quantifiers*, hence the running time of the algorithms will depend on the alternation depth of the assertion.[5] For alternation depth one, using the formula above, the worst-case running time is $O(m_1 m_2)$. For alternation depth two, the worst-case time complexity is $O(n_1 n_2 m_1 m_2)$. Table 2 summarises the algorithms obtained in this way, and compares the complexity with known hand-crafted algorithms for the relations; l_i is the number of labels in the transition system T_i.

Of course, for a carefully studied equivalence as strong bisimulation, taking full advantage of the knowledge of the relation being an equivalence and therefore representing it compactly by its equivalence classes, the pμK$_\exists$-algorithm cannot compete.[6] But for the preorders the algorithms are quite efficient. Moreover, this includes preorders that have yet to be defined and studied!

[5] Remaining quantifiers could have been replaced by finite disjunctions when the finite set of labels would be known, but this potentially gives an exponential blow-up in size.

[6] unless the number of labels is $O(m)$, i.e. there are asymptotically as many labels as transitions.

Relation	pμK$_\exists$-algorithm	special algorithm	reference for sp. alg.
Bisimulation	$O(m_1 m_2)$	$O(lm \log n)$	[CP89]
Weak bisimulation	$O(n_1 n_2 m_1 m_2)$	$O(lm^2 \log n)$	[ESTT91]
Obs. congruence	$O(n_1 n_2 m_1 m_2)$	$O(lm^2 \log n)$	[ESTT91]
Ready simulation	$O(m_1 m_2)$	$O(mn)$	[BP92]
Ready bisimulation	$O(m_1 m_2)$	$O(mn)$	[BP92]
Simulation preorder	$O(n_1 n_2 m_1 m_2)$	$O((n_1 n_2)^2 m_1 m_2)$	Naive fixed-point comp.
Prebisimulation	$O(m_1 m_2)$	$O(m_1 m_2)$	[CS91]
Abbreviations: $m = m_1 + m_2, n = n_1 + n_2, l = l_1 + l_2$.			

Table 2: Worst-case time complexities of algorithms for behavioural relations

Furthermore, by applying one of the local model checking algorithms instead, we get local (or "on-the-fly") algorithms for all the relations too. (The local algorithm of [And93] is only a logarithmic factor worse than the global used in Theorem 2.)

Example 2 (Characteristic Formulae) A simple application of the quotienting operator is in the construction of *characteristic formulae*, formulae that characterise an equivalence class of some behavioural equivalence, or downwards (or upwards) closed sets of preorders. For example, $(\mathcal{W}/_2 i)$ holds for transition systems weakly bisimilar to the transition system pointed by i, and $(\mathcal{R}/_2 i)$ holds for transition systems that ready simulates the transition system pointed by i. Thus we get "for free" characteristic formulae of the infinity of relations expressible in pμK$_\exists$ – including those treated by Steffen [Ste89]. □

6 Related and Future Work

The idea of using temporal logic model checking for verification was introduced by Clarke and Emerson [CE81]. A lot of work on model checking with the temporal logic CTL* has since then taken place in Clarke's group at CMU. In particular, in recent years a heuristic based on compact representations of boolean functions due to Bryant [Bry86] – called BDD's – has turned out to be quite successful, see e.g. Burch et.al. [BCM$^+$90]. The algorithms considered in this paper are efficient from a worst-case point of view and fall short when the BDD-heuristic is successful. However, when the heuristics fail it is important to have efficient worst-case algorithms.

The idea of using model checking algorithms to verify behavioural relations is already present in Cleaveland and Steffen [CS91] and Larsen [Lar92]. Our contribution is to formalise this idea by presenting a logic for expressing such relations, the polyadic modal μ-calculus, which, despite the decidability of model checking, turns out to possess properties quite different from the propositional μ-calculus. (In [BCM$^+$90] a related encoding of behavioural relations is described. However, Park's full predicate μ-calculus [Par81] is used for this purpose.)

A potential useful feature of pμK and pμK$_\exists$ is the possibility of combining the behavioural relations with more conventional dynamic properties, being able to express properties like "eventually weakly bisimilar", thus allowing for example initialisations in the two systems to be different.

As Theorem 1 and the corollary shows pμK is certainly a non-trivial extension of μK. Another corollary, discovered when proving Theorem 1, is the impossibility of expressing the property of being a "product of transition systems" within μK. To see why this is true, assume that P was such an assertion. Hence $T \models P$, if and only if, there exists T_1, T_2 with $pr(T_1, T_2) \sim T$. Then the assertion $pr(\Theta(\mathcal{D})) \wedge P$ will have a model in μK, if and only if, $\Theta(\mathcal{D})$ has a model within pμK. But since satisfiability is decidable for μK and not for pμK this is clearly impossible.

It would be interesting to see how far the language pμK could be enriched and the technique of quotienting still being used in getting efficient algorithms. Surely, we cannot go all the way to a full relational μ-calculus, since there are monotonic functions which require exponential many boolean operators to be described (see for example Boppana and Sipser [BS90a]) and therefore cannot give rise to compact arity zero assertions. But pμK$_\exists$ shows that extensions *are* possible.

Acknowledgements

David Harel's and Thiagarajan's lectures and subsequent discussions with them at the Summerschool in Logical Methods in Concurrency at Aarhus University in August 1993 provided useful information about Recurring Domino Problems. Other participants at the Summerschool, in particular, Peter Sestoft and Ole Høgh Jensen provided valuable comments. Thanks are also due to Kim Guldstrand Larsen and Liu Xinxin for comments on solving process equations that convinced me about the undecidability of validity for pμK.

References

[AC88] André Arnold and Paul Crubille. A linear algorithm to solve fixed-point equations on transitions systems. *Information Processing Letters*, 29:57–66, 1988.

[ADCR89] G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors. *Proceedings of ICALP*, volume 372 of *LNCS*. Springer-Verlag, 1989.

[And93] Henrik Reif Andersen. *Verification of Temporal Properties of Concurrent Systems*. PhD thesis, Department of Computer Science, Aarhus University, Denmark, June 1993. PB-445.

[And94] Henrik Reif Andersen. Model checking and boolean graphs. *Theoretical Computer Science*, 126(1):3–30, April 1994.

[AW92] Henrik Reif Andersen and Glynn Winskel. Compositional checking of satisfaction. *Formal Methods In System Design*, 1(4), December 1992.

[BCM+90] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 428–439. IEEE Computer Society Press, 1990.

[BK90] J.C.M. Baeten and J.W. Klop, editors. *Proceedings of CONCUR '90*, volume 458 of *LNCS*. Springer-Verlag, 1990.

[Blo89] Bard Bloom. *Ready Simulation, Bisimulation, and the Semantics of CCS-Like Languages*. PhD thesis, Massachusetts Institute of Technology, August 1989.

[BP92] Bard Bloom and Robert Paige. Computing ready simulations efficiently. Draft, July 1992.

[Bry86] R.E. Bryant. Graph-based algorithms for boolean function manipulation. *Transactions on Computers*, 8(C-35):677–691, 1986.

[BS90a] R.B. Boppana and M. Sipser. The complexity of finite functions. In Jan van Leeuwen, editor, *Algorithms and Complexity*, volume A of *Handbook of Theoretical Computer Science*, chapter 14. Elsevier, 1990.

[BS90b] Julian C. Bradfield and Colin P. Stirling. Verifying temporal properties of processes. In Baeten and Klop [BK90], pages 115–125.

[CDS92] Rance Cleaveland, Marion Dreimüller, and Bernhard Steffen. Faster model checking for the modal mu-calculus. In v.Bochmann and Probst [vP92], pages 383–394.

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs, Workshop, Yorktown Heights, New York, May 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1981.

[Cle90] Rance Cleaveland. Tableau-based model checking in the propositional mu-calculus. *Acta Informatica*, 27:725–747, 1990.

[CP89] Jiazhen Cai and Robert Paige. Program derivation by fixed point computation. *Theoretical Computer Science*, 11:197–261, 1989.

[CS91] Rance Cleaveland and Bernhard Steffen. Computing behavioural relations, logically. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proceedings of ICALP*, volume 510 of *LNCS*, pages 127–138. Springer-Verlag, July 1991.

[dBdR73] J.W. de Bakker and W.P. de Roever. A calculus for recursive program schemes. In *1st International Colloquium on Automata, Languages and Programming* [ica73], pages 167–196.

[EJ88] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *IEEE Foundations of Computer Science*, 29:328–337, 1988.

[EL86] E. Allen Emerson and Chin-Luang Lei. Efficient model checking in fragments of the propositional mu-calculus. In *Symposium on Logic in Computer Science, Proceedings*, pages 267–278. IEEE, 1986.

[ESTT91] Klaus Estenfeld, Hans-Albert Schneider, Dirk Taubner, and Erik Tidén. Computer aided verification of parallel processes. In A. Pfitzmann and E. Raubold, editors, *VIS '91 Verlässliche Informationssysteme*, volume 271 of *Informatik Fachberichte*, pages 208–226, Darmstadt, 1991. Springer-Verlag.

[Har85] David Harel. Recurring dominoes: Making the highly undecidable highly understandable. *Ann. Disc. Math*, 24:51–72, 1985.

[HP73] Peter Hitchcock and David Park. Induction rules and termination proofs. In *1st International Colloquium on Automata, Languages and Programming* [ica73], pages 225–251.

[Hüt90] Hans Hüttel. SnS can be modally characterized. *Theoretical Computer Science*, 74(2):239–248, August 1990.

[ica73] *1st International Colloquium on Automata, Languages and Programming.* North-Holland, 1973.

[Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27, 1983.

[Koz88] Dexter Kozen. A finite model theorem for the propositional μ-calculus. *Studia Logica*, 47:233–242, September 1988.

[Lar88] Kim G. Larsen. Proof systems for Hennessy-Milner logic with recursion. In M. Dauchet and M. Nivat, editors, *Proceedings of CAAP, Nancy, Franch*, volume 299 of *Lecture Notes in Computer Science*, pages 215–230, March 1988.

[Lar92] Kim G. Larsen. Efficient local correctness checking. In v.Bochmann and Probst [vP92].

[LX90] Kim G. Larsen and Liu Xinxin. Compositionality through an operational semantics of contexts. In M.S. Paterson, editor, *Proceedings of ICALP*, volume 443 of *LNCS*, pages 526–539. Springer-Verlag, 1990.

[Mil81] Robin Milner. A modal characterisation of observable machine behaviours. In G. Astesiano and C. Böhm, editors, *CAAP '81*, number 112 in LNCS, pages 25–34. Springer-Verlag, 1981.

[Mil89] Robin Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Niw88] Damian Niwiński. Fixed points vs. infinite generation. In *Proceedings, Third Annual Symposium on Logic in Computer Science*, pages 402–409, Edinburgh, Scotland, July 5–8 1988. IEEE Computer Society.

[Par81] David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Proceedings of Theoretical Computer Science, 5th GI-Conference, Karlsruhe, March 23-25, 1981*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981. The full predicate mu-calculus source.

[Rog67] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill Book Company, 1967.

[Ste89] Bernhard Steffen. Characteristic formulae for CCS with divergence. In Ausiello et al. [ADCR89], pages 723–733.

[Sti85] Colin Stirling. A complete modal proof system for a subset of SCCS. volume 185 of *Lecture Notes in Computer Science*, pages 253–266. Springer-Verlag, 1985.

[Sti92] Colin Stirling. Modal and Temporal Logics. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 477–563. Oxford University Press, 1992.

[SW91] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89(1):161–177, 1991.

[Tar55] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[vP92] G. v.Bochmann and D. K. Probst, editors. *Proceedings of the 4th Workshop on Computer Aided Verification, CAV'92, June 29 - July 1, 1992, Montreal, Quebec, Canada*, volume 663 of *LNCS*. Springer-Verlag, 1992.

[Wal93] Igor Walukiewicz. On completeness of the μ-calculus. In *Proceedings, Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 136–146, Montreal, Canada, 19–23 June 1993. IEEE Computer Society Press.

[Wan61] Hao Wang. Proving theorems by pattern recognition II. *Bell System Technical Journal*, 40:1–41, 1961.

[Win84] Glynn Winskel. Synchronisation trees. *Theoretical Computer Science*, 34:33, 1984.

[Win86] Glynn Winskel. A complete proof system for SCCS with modal assertions. *Fundamenta Informaticae*, IX:401–420, 1986.

[Win89] Glynn Winskel. A note on model checking the modal ν-calculus. In Ausiello et al. [ADCR89], pages 761–772.

[Win90] Glynn Winskel. On the compositional checking of validity. In Baeten and Klop [BK90], pages 481–501.