Tableaux with Automata (Extended Abstract)

Oliver Friedmann and Martin Lange

Dept. of Computer Science, University of Munich, Germany

1 Introduction

Tableaux and automata are two different (but not unrelated) methodologies underlying decision procedures for various logics. In this paper we do not want to argue in favour or against one of these frameworks. Undeniably, each of them has their own advantages (and maybe also disadvantages) which are the reason for the success and usefulness of certain decision procedures. Instead we examine a situation in which, as we think, the best decision procedure is obtained by combining tableaux *with* automata using certain advantages of both and avoiding their disadvantages.

The problem we are going to tackle is the satisfiability (or, by duality, validity) problem for the modal μ-calculus and therefore for various other branching time logics as well. This is not unsolved [1], but on the one hand the modal μ-calculus is expressive enough to be of wide interest, and on the other it is structurally simple enough in order to attempt a clean presentation which hopefully does not distract through unnecessary detail from the combination of tableau and automata methods.

Basically all procedures for the satisfiability problem of the modal μ-calculus use an intermediate step in the attempt to construct a model for a given formula. In Emerson and Jutla's automata-theoretic approach this is the notion of pre-models for instance [1], here we will speak of pre-tableaux instead. Models of formulas are then obtained by discarding those pre-models that do not satisfy a certain well-foundedness condition, namely do not guarantee least fixpoint constructs to be fulfilled in a finite number of steps.

Here we will characterise satisfiability of formulas of the modal μ-calculus through the existence of tableaux which are infinite trees (pre-tableaux) with such an additional well-foundedness condition. The tableau method seems suitable in this step because of the flexibility it provides in the choice of rules. We therefore do not need Hintikka sets and, most importantly do not need a fixed (and usually overestimated) arity for trees that are recognised by tree automata for μ-calculus formulas. However, it turns out that non-well-foundedness is easily recognised by nondeterministic Büchi automata. We then use powerful tools from automata-theory, namely determinisation and complementation in order to turn this into a device that decides well-foundedness in the tableaux. In the end, satisfiability is reduced to the existence of an infinite tableau whose nodes are labeled with formula sets and states of a deterministic parity automaton. Deciding satisfiability is then reduced to the solving of parity games thus avoiding the re-introduction of automata theory or the invention of decision methods for infinite tableaux.

2 The Modal μ-Calculus

Let $\mathcal{V}$ be an infinite set of variables. Formulas of the modal μ-calculus (in positive normal form) over a set $\mathcal{P}$ of propositions are given as follows.

$$\varphi \quad ::= \quad q \mid \overline{q} \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $X \in \mathcal{V}$, $q \in \mathcal{P}$. We will write σ for either μ or ν whenever the type of fixpoint is negligible. They are interpreted over states s of a labeled transition systems $\mathcal{T} = (\mathcal{S}, \longrightarrow, \ell)$, where $(\mathcal{S}, \longrightarrow)$ is a directed graph and $\ell : \mathcal{S} \to 2^{\mathcal{P}}$ as follows. Let $\rho : \mathcal{V} \to 2^{\mathcal{S}}$.

$$\begin{array}{rcl}
s \models_\rho q & \text{iff} & q \in \ell(s) \\
s \models_\rho \overline{q} & \text{iff} & q \notin \ell(s) \\
s \models_\rho X & \text{iff} & s \in \rho(X) \\
s \models_\rho \varphi \vee \psi & \text{iff} & s \models_\rho \varphi \text{ or } s \models_\rho \psi \\
s \models_\rho \varphi \wedge \psi & \text{iff} & s \models_\rho \varphi \text{ and } s \models_\rho \psi \\
s \models_\rho \Diamond\varphi & \text{iff} & \exists t \in \mathcal{S} \text{ with } s \longrightarrow t \text{ and } t \models_\rho \varphi \\
s \models_\rho \Box\varphi & \text{iff} & \forall t \in \mathcal{S} : \text{ if } s \longrightarrow t \text{ then } t \models_\rho \varphi \\
s \models_\rho \mu X.\varphi & \text{iff} & \forall T \subseteq \mathcal{S} : \text{ if } \forall t \in T : t \models_{\rho[X \mapsto T]} T \text{ implies } t \in T \text{ then } s \in T \\
s \models_\rho \nu X.\varphi & \text{iff} & \exists T \subseteq \mathcal{S} \text{ s.t. } s \in T \text{ and } \forall t \in T : t \models_{\rho[X \mapsto T]} \varphi
\end{array}$$

We assume the reader to be familiar with the standard notions of syntactic subformulas $Sub(\varphi)$, free variables, closed formulas, substitution $\varphi[\psi/X]$ of all occurrences of a variable, etc.

A formula φ is in *normal form* iff there is only one fixpoint binder $\sigma X.\varphi'$ for every bound variable X and every occurring fixpoint binder $\sigma X.\varphi'$ is proper, meaning that $X \in Sub(\varphi')$. Clearly, every closed formula φ can easily be transformed into an equivalent closed formula ϑ in normal form. Then there is a function fp_ϑ that maps each variable X occuring in ϑ to the body ψ of its defining fixpoint subformula $\sigma X.\psi$.

A formula φ is in *guarded form* iff every occurrence of a bound variable X is in the scope of a modal operator $\Diamond$ or $\Box$ under its quantifier $\sigma X.\varphi'$. Every formula can be transformed into guarded form incurring a quadratic blow-up [2]. We assume from now on that every formula is guarded and in normal form.

Let φ be fixed and take two variables X, Y occurring in it. Then Y *depends on* X, written $X \succ_\varphi Y$, if X occurs freely inside of $fp_\varphi(Y)$. Let $\succ_\varphi^+$ be the transitive closure of $\succ_\varphi$. The *index* of φ is the maximal n in a chain $X_1 \succ_\varphi^+ X_2 \succ_\varphi^+ \ldots \succ_\varphi^+ X_n$ s.t. adjacent variables in this chain are of different fixpoint type μ or ν. A variable X is called *outermost* among a set of variables V if there is no $Y \in V$ s.t. $Y \succ_\varphi^+ X$.

$$(\mathtt{Or})\ \frac{\varphi_0 \vee \varphi_1, \Phi}{\varphi_i, \Phi} \qquad (\mathtt{And})\ \frac{\varphi_0 \wedge \varphi_1, \Phi}{\varphi_0, \varphi_1, \Phi} \qquad (\mathtt{FP})\ \frac{\sigma X.\varphi, \Phi}{X, \Phi} \qquad (\mathtt{Var})\ \frac{X, \Phi}{fp_\vartheta(X), \Phi}$$

$$(\mathtt{Mod})\ \frac{\Diamond\varphi_1, \ldots, \Diamond\varphi_n, \Box\psi_1, \ldots, \Box\psi_m, q_1, \ldots, q_k, \overline{p_1}, \ldots, \overline{p_h}}{\varphi_1, \psi_1, \ldots, \psi_m \quad \varphi_2, \psi_1, \ldots, \psi_m \quad \ldots \quad \varphi_n, \psi_1, \ldots, \psi_m}\ \text{if } q_i \neq p_j \text{ for all } i, j$$

Fig. 1. The rules for building pre-tableaux.

3 An Infinite Proof System for the Modal μ-Calculus

In the following we fix a formula ϑ for which satisfiability is to be decided. A *pre-tableau* for ϑ is a (possibly infinite but finitely-branching) tree in which nodes are labeled with subsets of $Sub(\vartheta)$, the set of subformulas of ϑ. The root is labeled with the singleton set containing ϑ, and successors in the tree are being built using the rules in Fig. 1.

The formula ϑ induces the *connection* relation $\rightsquigarrow \subseteq 2^{Sub(\vartheta)} \times Sub(\vartheta) \times 2^{Sub(\vartheta)} \times Sub(\vartheta)$ defined as follows. We have $\Phi, \varphi \rightsquigarrow \Psi, \psi$ iff there is an instance of a rule of Fig. 1 s.t.

- $\varphi \in \Phi$, $\psi \in \Psi$, and
- Φ is the conclusion (on top), Ψ is one of the premisses (below), and
- either φ is not principal in this rule application and $\psi = \varphi$, or φ is a principal formula in Φ and ψ is a replacement of φ.

For example, in rule (And), $\varphi_0 \wedge \varphi_1$ is connected to both φ_0 and φ_1. In rule (Mod), $\Box\psi_j$ is connected to ψ_j in any premiss, literals are not connected to anything, and $\Diamond\varphi_i$ is only connected to φ_i in the i-th premiss; etc.

A *thread* in an infinite pre-tableaux branch $\Phi_0, \Phi_1, \Phi_2, \ldots$ is an infinite sequence $\varphi_0, \varphi_1, \varphi_2, \ldots$ s.t. $\Phi_i, \varphi_i \rightsquigarrow \Phi_{i+1}, \varphi_{i+1}$ for every $i \in \mathbb{N}$. It is called μ-thread if the outermost variable occurring infinitely often in this sequence is of type μ. Otherwise it is called ν-thread.

A *tableau* for ϑ is a pre-tableau s.t. every finite branch ends in a node labeled with $\Box$-formulas and consistent literals only, and every infinite branch does not have a μ-thread. The following is not too difficult to show.

Proposition 1. *A formula ϑ is satisfiable iff there is a tableau for ϑ.*

The direction from right to left takes a tableau and collapses sequences of nodes between which no application of rule (Mod) occurs into a state of a transition system. The converse direction makes use of the fact that every formula has a pre-tableau. A model for ϑ is then traversed state-by-state in order to construct such a pre-tableau, and the global thread-conditions are shown to hold using approximants of least fixpoint formulas. An immediate consequence of this construction is the tree model property for the modal μ-calculus.

Proposition 2. *Every satisfiable formula ϑ of the modal μ-calculus has a tree model with out-degree bounded by the number of different $\Diamond$-subformulas of ϑ.*

4 Using Automata

The previous section characterises satisfiability through the existence of an infinite tableau. In this section we describe how automata can be used in order to decide this existence. Again, we fix a formula ϑ. It induces an *alphabet of rule applications*. Let

$$\begin{aligned}\Sigma_\vartheta \; := & \; \{\mathtt{And}(\varphi_0 \wedge \varphi_1) \mid \varphi_0 \wedge \varphi_1 \in \mathit{Sub}(\vartheta)\} \;\cup\; \{\mathtt{FP}(X), \mathtt{Var}(X) \mid X \in \mathit{Sub}(\varphi)\} \\ & \cup \{\mathtt{LOr}(\varphi_0 \vee \varphi_1), \mathtt{ROr}(\varphi_0 \vee \varphi_1) \mid \varphi_0 \vee \varphi_1 \in \mathit{Sub}(\vartheta)\} \;\cup\; \{\mathtt{Mod}\}\end{aligned}$$

With an infinite branch $\rho = \Phi_0, \Phi_1, \ldots$ of a pre-tableau for ϑ we associate a word $w_\rho \in \Sigma_\vartheta^\omega$ in the natural way: the i-th letter of w_ρ is $\mathtt{ROr}(\varphi_0 \vee \varphi_1)$ for instance iff Φ_{i+1} is obtained from Φ_i through an application of rule $(\mathtt{Or})$ on the principal formula $\varphi_0 \vee \varphi_1$ which is then replaced by its right disjunct φ_1; etc.

Lemma 1. *There is a nondeterministic Büchi automaton (NBA) $\mathcal{B}_\vartheta$ over Σ_ϑ s.t.*

- *the number of states in $\mathcal{B}_\vartheta$ is bounded by $(\mathit{ind}(\vartheta) + 1) \cdot |\mathit{Sub}(\vartheta)|$,*
- *$L(\mathcal{B}_\vartheta) = \{w \mid$ if $w = w_\rho$ for some ρ then ρ has a μ-thread $\}$.*

The NBA simply guesses threads by tracing single subformulas in its state set. Upon reading an input letter it knows whether the next rule application transforms the currently traced subformula or whether it remains the same on that thread. A parity condition that reflects the alternation depth of each variable inside ϑ can then be used in order to recognise the language at hand. At last on uses the standard and simple transformation of nondeterministic parity automata into NBA.

Next we will use a powerful automata-theoretic result in order to make the global thread conditions in tableaux algorithmically handable.

Theorem 1 ([3]). *For every NBA $\mathcal{B}$ with n states there is a deterministic parity automaton (DPA) $\mathcal{A}$ with $2^{\mathcal{O}(n \log n)}$ states and index $\mathcal{O}(n)$ s.t. $L(\mathcal{A}) = \overline{L(\mathcal{B})}$.*

Combining this theorem with Lemma 1 yields a DPA $\mathcal{A}_\vartheta$ with some exponentially sized state set Q that recognises exactly those sequences of rule applications that either do not correspond to a pre-tableau branch, or that do correspond to a branch which does not contain a μ-thread. This allows us to define a parity game G_ϑ with a node v_0 s.t. this node is won by player 0 iff ϑ is satisfiable. The nodes of the game are of the form $2^{\mathit{Sub}(\vartheta)} \times Q$, the designated node v_0 is $(\{\vartheta\}, q_0)$ where q_0 is the initial state of $\mathcal{A}_\vartheta$. A node $w = (\Psi, q')$ is a successor of $v = (\Phi, q)$ if a uniquely (for (Φ, q)) chosen rule is applied to Φ that yields Ψ as one of its premisses, this rule is represented by $r \in \Sigma_\vartheta$ and $\delta(q, r) = q'$ where δ is the transition function of $\mathcal{A}_\vartheta$. The node ownership in the game is determined by these uniquely chosen rules: player 0 owns nodes in which rule $(\mathtt{Or})$ is applied, player 1 owns nodes in which rule $(\mathtt{Mod})$ is applied. All other nodes can have at most one successor. Finally, the priority of a game node (Φ, q) is simply $\Omega(q)$ where Ω is the priority function of $\mathcal{A}_\vartheta$.

Proposition 3. *Let ϑ be a formula with $n = |Sub(\vartheta)|$ and $k = ind(\vartheta)$. There is a parity game G_ϑ with number of nodes bounded by $2^{\mathcal{O}(n^2 k \log(nk))}$ and a designated node v that is won by player 0 iff ϑ is satisfiable.*

Furthermore, the subgame induced by a winning strategy for player 0 is in effect a model for ϑ. This immediately yields a small model property of corresponding size for all levels of the alternation hierarchy of the modal μ-calculus. However, better bounds may be obtained by considering this construction in a modular fashion. This is summarised in the following statement.

Proposition 4. *Let ϑ be a formula and $\mathcal{A}_\vartheta$ be a deterministic automaton with n states that recognises among branches of a pre-tableau for ϑ exactly those that not contain a μ-thread. Then there is a game G_ϑ of the same acceptance condition as $\mathcal{A}_\vartheta$ and size $2^{|\vartheta|} \cdot n$ s.t. solving this game decides satisfiability of ϑ, and a model for ϑ can be found as a subgame of G_ϑ.*

This more abstract formulation avoids explicit determinisation of NBA. This is particularly beneficial when considering fragments of the modal μ-calculus. Take for instance the fragment in which no μ-bound variable occurs freely inside a ν-quantified formula. Then a μ-thread can be recognised using a co-Büchi condition since the corresponding automaton only has to check whether it eventually remains inside the unfolding of a μ-formula. Co-Büchi automata of size n can be determinised and then complemented to a deterministic Büchi automaton of size 2^{2n}. Hence, for this fragment we immediately obtain the small model property of size $2^{3|\vartheta|}$. Additionally, this fragment can be decided through a reduction to co-Büchi games, i.e. max-parity games in which only the priorities 0 and 1 occur which is easier than solving general parity games.

References

1. E. A. Emerson and C. S. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd Symp. on Foundations of Computer Science*, pages 368–377, San Juan, Puerto Rico, 1991. IEEE.
2. Radu Mateescu. Local model-checking of modal mu-calculus on acyclic labeled transition systems. In *TACAS '02: Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 281–295, London, UK, 2002. Springer-Verlag.
3. N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Proc. 21st Symp. on Logic in Computer Science, LICS'06*, pages 255–264. IEEE Computer Society, 2006.