Games for the μ-calculus

Damian Niwiński* Igor Walukiewicz*

Institute of Informatics, Warsaw University,
Banacha 2, 02-097 Warsaw, POLAND
niwinski,igw@mimuw.edu.pl

Abstract

Given a formula of the propositional μ-calculus, we construct a tableau of the formula and define an infinite game of two players of which one wants to show that the formula is satisfiable, and the other seeks the opposite. The strategy for the first player can be further transformed into a model of the formula while the strategy for the second forms what we call a refutation of the formula. Using Martin's Determinacy Theorem, we prove that any formula has either a model or a refutation. This completeness result is a starting point for the completeness theorem for the μ-calculus to be presented elsewhere. However, we argue that refutations have some advantages of their own. They are generated by a natural system of sound logical rules and can be presented as regular trees of the size exponential in the size of a refuted formula. This last aspect completes the small model theorem for the μ-calculus established by Emerson and Jutla [3]. Thus, on a more practical side, refutations can be used as small objects testifying incorrectness of a program specification expressed by a μ-formula, we illustrate this point by an example.

1 Introduction

It is now common to view computer programs as state transformers, that is, actions that can change one state of computer hardware to another. The notion of change is intrinsic in modal logic which admits a hypothesis that the world may change; rather than a single unique world, one considers there multiple possible worlds and relations between them, as, for example, the states of an environment evolving in time. This aspect makes the modal logic a valuable tool for reasoning about program behavior.

*Supported by Polish KBN grant No. 2 1192 91 01

A variety of logical formalisms have been proposed in this context, including Hoare logic, Dijkstra's weakest precondition calculus, dynamic logic, temporal logic and automata-based formalisms. A uniform mathematical framework subsuming all of these and more is provided by the μ-calculus, a formalism permitting characterization of correctness properties as extremal fixed points of predicate transformers [13, 2, 8].

In spite of its great expressive power, a propositional version of the μ-calculus [8] was proven decidable in deterministic single exponential time [3], which makes this logic interesting for applications to verification of real life systems.

Although the μ-calculus, and especially the propositional μ-calculus attracted much interest [8, 17, 1, 5] the attempts to find a complete finitary axiomatization for the calculus systematically failed. Even for strictly weaker modal logics such as PDLΔ no finitary complete axiom system has been known.

A complete axiomatization of any calculus is still interesting even when it is decidable. The main gain is that a complete axiomatization gives us deeper insight into the calculus. On a more practical side, we can think of a prover instead of a model checker for verifying properties. An advantage of a prover could be that it would allow an interaction between computer and user, in which a user might give some hints, for example, an auxiliary formula for an application of a cut rule. This would help the computer in what is essentially an exponential task.

The present paper constitutes the first part of the work the final goal of which is to prove a completeness theorem for the μ-calculus. The second part which gives an actual system and proves its completeness is due to the second author [19, 20] and will be presented at length elsewhere.

In the present paper, we introduce a concept of *refutation* which can be viewed as some approximation of (or substitute for) the concept of proof. Refutations are trees generated by some natural system of tableau rules which are sound logical rules, but, unlike proofs, refutations may have infinite branches. We show that any formula of the propositional μ-calculus has either a model or a refutation. This result is an advance toward the completeness theorem for the μ-calculus to be proved in the second part of the work (for a preliminary version, see [19]) . Indeed, the proof system proposed there is designed in such a way that a proof of a valid μ-formula φ can be obtained from a refutation of $\neg\varphi$. Thus the completeness theorem follows from the above mentioned completeness result for refutations.

We believe, however, that the concept of refutation and the relevant result are of some interest of their own, and this is at least for two reasons.

First, refutations turn out to be small objects: using results from automata theory, we show that a refutation of a μ-calculus formula can be always made a regular tree of the size exponential in the size of the formula. This result completes the small model theorem for the μ-calculus due to Emerson and Jutla [3]. Combining the two results, we can state the following.

> A formula of the propositional μ-calculus has either a model or a refutation, any of which may be chosen to be a regular tree which can be presented as a graph of exponential size, produced in exponential time.

This suggests that refutations may play some role in program verification as small objects that witness the validity of μ-formulas; we illustrate this point by some examples. In this context, it would be interesting to estimate the length of "real" proofs in the existing proof systems, as the system for the μ-calculus given in [19], or a system for much weaker logic PDL [9]. The question seems to be difficult, but we know that the length of proofs in any reasonable proof system must be at least exponential and we are tempted to conjecture that it is much bigger than that.

The second aspect of refutations which is, we believe, of some theoretical interest is the connection with determinacy of certain infinite games. In our paper we consider an infinite game that is played on a tableau of a μ-calculus formula. Roughly speaking, one of the players wants to show that the formula is satisfiable, and the other seeks the opposite. The game is determined by Martin's determinacy theorem [10]. Now it turns out that a winning strategy (if it exists) for the first player induces a model of the formula, while a strategy for his/her opponent can be identified with a refutation.

Infinite games were studied in set theory [10, 11], they also appear in automata theory in several different proofs that have been proposed for Rabin's Complementation Lemma [7, 12, 5]. The present paper exhibits yet another aspect of such games which is an intriguing link between determinacy ("one of the player has a winning strategy") and completeness ("a formula has a model or a refutation").

The paper is organized as follows. After a preliminary Section 2, we introduce, in Section 3, a system of rules for generating tableaux of the μ-calculus formulas. We present also two subsystems that can generate what we call refutations and pre-models, respectively. The concept of a game is introduced in Section 4, where we also use the Martin's Theorem to show that any μ-calculus formula has either a refutation or a pre-model. In Section 5, we show that a pre-model can indeed be transformed into a model and *vice versa*. In Section 6, we consider the complexity of the constructions in consideration. In the Appendix A, we illustrate by a simple example, how our concept of refutation can be helpful in the analysis of specifications.

2 Syntax and Semantics

Let $Prop = \{p, q, \ldots\}$ be a set of propositional letters, $Var = \{X, Y, \ldots\}$ a set of variables and $Act = \{a, b, \ldots\}$ a set of actions. Formulas of the μ-calculus over

these three sets can be defined by the following grammar:

$$F := Var \mid Prop \mid \neg Prop \mid F \vee F \mid F \wedge F \mid \langle Act \rangle F \mid [Act]F \mid \mu Var.F \mid \nu Var.F$$

Observe that we allow negations to occur only before propositional letters. The negation of a formula is defined inductively by DeMorgan laws and the equivalences $\neg[a]\alpha \equiv \langle a \rangle \neg\alpha$ and $\neg\mu X.\alpha(X) \equiv \nu X.\neg\alpha(\neg X)$. Symbol *ff* is an abbreviation of a formula $p \wedge \neg p$ for some propositional constant p. We will sometimes use $\sigma X.\alpha(X)$ to denote $\mu X.\alpha(X)$ or $\nu X.\alpha(X)$. Formulas are interpreted in Kripke models of the form $\mathcal{M} = \langle S, R, \rho \rangle$, where:

- S is a nonempty set of states,
- R is a function assigning a binary relation $R(a)$ on S to each action a in Act.
- ρ is a function assigning a set of states to each propositional letter in $Prop$.

For a given model $\mathcal{M}$ and an assignment $Val : Var \to \mathcal{P}(S)$, the set of states in which a formula α is true, $\| \alpha \|^{\mathcal{M}}_{Val}$ is defined inductively as follows (we will omit superscript $\mathcal{M}$ when it causes no ambiguity):

$$\begin{aligned}
\| X \|_{Val} &= Val(X) \\
\| p \|_{Val} &= \rho(p) \\
\| \neg p \|_{Val} &= S - \rho(p) \\
\| \alpha \wedge \beta \|_{Val} &= \| \alpha \|_{Val} \cap \| \beta \|_{Val} \\
\| \alpha \vee \beta \|_{Val} &= \| \alpha \|_{Val} \cup \| \beta \|_{Val} \\
\| \langle a \rangle \alpha \|_{Val} &= \{ s : \exists s'.(s, s') \in R(a) \wedge s' \in \| \alpha \|_{Val} \} \\
\| [a]\alpha \|_{Val} &= \{ s : \forall s'.(s, s') \in R(a) \ \Rightarrow s' \in \| \alpha \|_{Val} \} \\
\| \mu X.\alpha(X) \|_{Val} &= \bigcap \{ S' \subseteq S : \| \alpha \|_{Val[S'/X]} \subseteq S' \} \\
\| \nu X.\alpha(X) \|_{Val} &= \bigcup \{ S' \subseteq S : S' \subseteq \| \alpha \|_{Val[S'/X]} \}
\end{aligned}$$

We will also write $\mathcal{M}, s, Val \models \alpha$ to mean that $s \in \| \alpha \|^{\mathcal{M}}_{Val}$. Formula φ will be called valid, in symbols $\models \varphi$, iff it is true in every state of every model.

Definition 2.1 A variable X in $\sigma X.\alpha(X)$ is *guarded* iff every occurrence of X in $\alpha(X)$ is in scope of some modality operator $\langle\rangle$ or []. We say that a formula is guarded iff every bound variable in the formula is guarded.

Lemma 2.2 (Kozen) *Every formula is equivalent to some guarded formula.*

Proof

Let φ be a formula, we show how to obtain an equivalent guarded formula.

Suppose $\varphi = \mu X.\alpha(X)$ and $\alpha(X)$ is a guarded formula. Suppose X is unguarded in some subformula of $\alpha(X)$ of the form $\sigma Y.\beta(Y,X)$ and Y is guarded in $\sigma Y.\beta(Y,X)$. Then one can use the equivalence $\sigma Y.\beta(Y,X) \equiv \beta(\sigma Y.\beta(Y,X),X)$ to obtain a formula with all unguarded occurrences of X outside the fixpoint operator. This way we obtain a formula equivalent to $\alpha(X)$ but with all unguarded occurrences of X not in the scope of a fixpoint operator.

Now using the laws of classical propositional logic we can transform this formula to conjunctive normal form (considering formulas of the form $\langle a\rangle\gamma$ and $[a]\gamma$ as propositional constants). This way we obtain a formula

$$(X \vee \alpha_1(X)) \wedge \ldots \wedge (X \vee \alpha_i(X)) \wedge \beta(X) \tag{1}$$

where all occurrences of X in $\alpha_1(X), \ldots, \alpha_i(X), \beta(X)$ are guarded. Observe that it may happen that some of $\alpha_i(X)$ may be just false and $\beta(X)$ may be true.

Variable X occurs only positively in (1) because it did so in our original formula. Formula (1) is equivalent to

$$(X \vee (\alpha_1(X) \wedge \ldots \wedge \alpha_i(X))) \wedge \beta(X)$$

We will show that $\mu X.(X \vee \bar{\alpha}(X)) \wedge \beta(X)$ is equivalent to $\mu X.\bar{\alpha}(X) \wedge \beta(X)$. It is obvious that

$$\models (\mu X.\bar{\alpha}(X) \wedge \beta(X)) \;\Rightarrow\; (\mu X.(X \vee \bar{\alpha}(X)) \wedge \beta(X))$$

Let $\gamma(X)$ and $\bar{\gamma}(X)$ stand for $\bar{\alpha}(X) \wedge \beta(X)$ and $(X \vee \bar{\alpha}(X)) \wedge \beta(X)$ respectively. To prove the other implication it is enough to observe that $\models \bar{\gamma}(\mu X.\gamma(X)) \Rightarrow \mu X.\gamma(x)$ as the following calculation shows:

$$\begin{array}{rl} ((\mu X.\gamma(X)) \vee \bar{\alpha}(\mu X.\gamma(X))) \wedge \beta(\mu X.\gamma(X)) & \Rightarrow \\ ((\bar{\alpha}(\mu X.\gamma(X)) \wedge \beta(\mu X.\gamma(X))) \vee \bar{\alpha}(\mu X.\gamma(X))) \wedge \beta(\mu X.\gamma(X)) & \Rightarrow \\ \bar{\alpha}(\mu X.\gamma(X) \wedge \beta(\mu X.\gamma(X)) & \end{array}$$

□

Henceforth we will consider only guarded formulas. This restriction is not necessary but simplifies some of the definitions to follow (see Remark 3.7).

3 Tableaux

In this section we present a system of rules for constructing a tableau for a formula. Tableaux will serve as arenas for a game we will consider later. We also define two kinds of substructures of tableaux: quasi-model and quasi-refutation.

It is convenient to introduce the concept of a definition list [15] which will name the fixpoint subformulas of a given formula in order of their nesting. We will employ the technique of reusing definition constants as described in [16].

We extend vocabulary of the μ-calculus by a countable set *Dcons* of fresh symbols that will be referred to as *definition constants* and usually denoted $U, V, \ldots$ These new symbols are now allowed to appear positively in formulas, like propositional variables.

A *definition list* is a finite sequence of equations :

$$\mathcal{D} = ((U_1 = \sigma_1 X.\alpha_1(X)), \ldots, (U_n = \sigma_n X.\alpha_n(X))$$

where $U_1, \ldots, U_n \in DCons$ and $\sigma_i X.\alpha_i(X)$ is a formula such that all definition constants appearing in α_i are among $U_1, \ldots, U_{i-1}$. We call U_i a μ-constant or a ν-constant according to $\sigma_i = \mu$ or $\sigma_i = \nu$. We assume that $U_i \neq U_j$ and $\alpha_i \neq \alpha_j$, for $i \neq j$. If $i < j$ then U_i is said to be *older* than U_j (U_j younger then U_i) with respect to the definition list $\mathcal{D}$.

Given a formula α, we construct a definition list for α by means of the contraction operation $\rangle\!\alpha\!\langle$ defined recursively as follows:

1. $\rangle\!p\!\langle = \rangle\!\neg p\!\langle = \rangle\!X\!\langle = \rangle\!U\!\langle = \emptyset$;
2. $\rangle\!\langle a\rangle\alpha\!\langle = \rangle\![a]\alpha\!\langle = \rangle\!\alpha\!\langle$;
3. $\rangle\!\alpha \wedge \beta\!\langle = \rangle\!\alpha \vee \beta\!\langle = \rangle\!\alpha\!\langle \circ \rangle\!\beta\!\langle$, operation $\circ$ is defined below;
4. $\rangle\!\mu X.\alpha(X)\!\langle = ((U = \mu X.\alpha(X)), \rangle\!\alpha(U)\!\langle)$ where U is new;
5. $\rangle\!\nu X.\alpha(X)\!\langle = ((U = \nu X.\alpha(X)), \rangle\!\alpha(U)\!\langle)$ where U is new.

The operation $\rangle\!\alpha\!\langle \circ \rangle\!\beta\!\langle$ is defined as follows. First, we make sure that the definition constants used in $\rangle\!\alpha\!\langle$ are disjoint form those used in $\rangle\!\beta\!\langle$. Then if it happens that $(U = \gamma) \in \rangle\!\alpha\!\langle$ and $(V = \gamma) \in \rangle\!\beta\!\langle$, we delete the definition from the list $\rangle\!\beta\!\langle$ and replace V with U in $\rangle\!\beta\!\langle$. This may cause other formulas to be doubly defined and we deal with them in the same way.

For a formula α and a definition list $\mathcal{D}$ containing all definition constants occurring in α we define the *expansion operation* $\langle\!\alpha\!\rangle_{\mathcal{D}}$, which subsequently replaces definition constants appearing in the formula by the right hand sides of the defining equations,

$$\langle\!\alpha\!\rangle_{\mathcal{D}} = \alpha[\alpha_n/U_n]\ldots[\alpha_1/U_1] \ , \quad \text{where } \mathcal{D} = (U_1 = \alpha_1), \ldots, (U_n = \alpha_n)$$

A *tableau sequent* is a pair $(\Gamma, \mathcal{D})$, where Γ is a finite set of formulas and $\mathcal{D}$ is a definition list that includes all the definition constants occurring in Γ.

A *tableau axiom* is a sequent $\Gamma \vdash_{\mathcal{D}}$ such that some formula and its negation occur in Γ.

We extend expansion operations to tableau sequents in a natural way:

$$\langle\!\langle \Gamma \vdash_{\mathcal{D}} \rangle\!\rangle_{\mathcal{D}} = \{\langle\!\langle \gamma \rangle\!\rangle_{\mathcal{D}} : \gamma \in \Gamma\}$$

In what follows we will often drop the prefix "tableau" if it is clear from the context.

Below we present the set of rules for constructing tableaux. These rules can be considered as logical rules when read upside-down. We write them with premises below the line because it is more appropriate for tableaux construction; we prefer to view a tableau as a tree expanding downwards. This style also puts emphasis on the fact that this rules are used to construct tableaux rather than proofs.

Definition 3.1 Let $\mathcal{S}$ be the following set of tableau rules :

$$(and) \qquad \frac{\alpha \wedge \beta, \Gamma \vdash_{\mathcal{D}}}{\alpha, \beta, \Gamma \vdash_{\mathcal{D}}}$$

$$(or) \qquad \frac{\alpha \vee \beta, \Gamma \vdash_{\mathcal{D}}}{\alpha, \Gamma \vdash_{\mathcal{D}} \quad \beta, \Gamma \vdash_{\mathcal{D}}}$$

$$(cons) \qquad \frac{U, \Gamma \vdash_{\mathcal{D}}}{\alpha(U), \Gamma \vdash_{\mathcal{D}}} \quad \text{whenever } (U = \sigma X.\alpha(X)) \in \mathcal{D}$$

$$(\mu) \qquad \frac{\mu X.\alpha(X), \Gamma \vdash_{\mathcal{D}}}{U, \Gamma \vdash_{\mathcal{D}}} \quad \text{whenever } (U = \mu X.\alpha(X)) \in \mathcal{D}$$

$$(\nu) \qquad \frac{\nu X.\alpha(X), \Gamma \vdash_{\mathcal{D}}}{U, \Gamma \vdash_{\mathcal{D}}} \quad \text{whenever } (U = \nu X.\alpha(X)) \in \mathcal{D}$$

$$(all\langle\rangle) \qquad \frac{\Gamma \vdash_{\mathcal{D}}}{\{\alpha, \{\beta : [a]\beta \in \Gamma\} \vdash_{\mathcal{D}} \ : \langle a\rangle\alpha \in \Gamma\}}$$

where in the last rule each formula in Γ is a propositional constant, a variable, a negation of one of them or a formula of the form $\langle b\rangle\beta$ or $[b]\beta$ for some action b and a formula β (in other words, no other rule is applicable to $\Gamma \vdash_{\mathcal{D}}$).

Observe that each rule, except (or) and $(all\langle\rangle)$, has exactly one premise. The rule (or) has two premises and the number of premises in the rule $(all\langle\rangle)$ is equal to the number of formulas of the form $\langle a\rangle\alpha$ in Γ and may be 0.

The system $\mathcal{S}_{mod}$ is obtained from $\mathcal{S}$ by replacing the rule (or) by two rules (or_{left}) and (or_{right}) defined in the obvious way.

The system $\mathcal{S}_{ref}$ is obtained from $\mathcal{S}$ by replacing the rule $(all\langle\rangle)$ by the rule

$$(\langle\rangle) \qquad \frac{\langle a\rangle\alpha, \Gamma \vdash_{\mathcal{D}}}{\alpha, \{\beta : [a]\beta \in \Gamma\} \vdash_{\mathcal{D}}}$$

with the same restrictions on formulas in Γ as in the case of $(all\langle\rangle)$ rule.

Remark 3.2 If we consider a sequent $\Gamma \vdash_{\mathcal{D}}$ as a formula $\bigwedge \langle\!|\Gamma|\!\rangle_{\mathcal{D}} \Rightarrow \mathit{ff}$ then the rules of the system $\mathcal{S}_{ref}$ become sound logical rules written upside-down.

Definition 3.3 Given a positive guarded formula γ, a *tableau* for γ is any labeled tree $\langle K, L\rangle$, where K is a tree and L a labeling function, such that

1. the root of K is labeled with $\gamma \vdash_{\mathcal{D}}$ where $\mathcal{D} = |\!|\gamma|\!|$,
2. if $L(n)$ is a tableau axiom then n is a leaf of K,
3. if $L(n)$ is not an axiom then the sons of n in K are created and labeled according to the rules of the system $\mathcal{S}$.

The definition of a *quasi-model* of γ is defined in a similar to that of a tableau, except that the system $\mathcal{S}_{mod}$ is used instead of $\mathcal{S}$ and we impose the additional requirement that *no* leaf is labeled by a tableau axiom.

The definition of a *quasi-refutation* of γ is similar to that of tableau, except that the system $\mathcal{S}_{ref}$ is used instead of $\mathcal{S}$ and we impose the additional requirement that *every* leaf is labeled by a tableau axiom. ∎

Remark 3.4 Each quasi-model, as well as a quasi-refutation can be obtained from a tableau by cutting off some nodes.

Definition 3.5 Given a path $\mathcal{P}$ of a tableau $\mathcal{T} = \langle T, L\rangle$, a *trace* on $\mathcal{P}$ will be a function $\mathcal{T}r$ assigning a formula to every node in some initial segment of $\mathcal{P}$ (possibly to the whole $\mathcal{P}$), satisfying the following conditions:

1. If $\mathcal{T}r(n)$ is defined then $\mathcal{T}r(n) \in L(n)$.
2. Suppose $\mathcal{T}r(m)$ is defined and let $n \in \mathcal{P}$ be a son of m.
   - If a rule different from $(all\langle\rangle)$ is applied in m then $\mathcal{T}r(n)$ is defined, and
     - if the rule does not reduce the formula $\mathcal{T}r(m)$ then $\mathcal{T}r(n) = \mathcal{T}r(m)$;
     - if $\mathcal{T}r(m)$ is reduced by the rule applied in m then $\mathcal{T}r(n)$ is one of the results of the reduction (nondeterministically).
   - Suppose the rule $(all\langle\rangle)$ is applied in m, and the label of n is $\alpha, \{\beta : [a]\beta \in L(m)\} \vdash_{\mathcal{D}}$, where $\langle a\rangle\alpha \in L(m)$. Then,
     - if $\mathcal{T}r(m) = [a]\beta$ then $\mathcal{T}r(n)$ is defined and equals β;
     - if $\mathcal{T}r(m) = \langle a\rangle\alpha$ then $\mathcal{T}r(n)$ is defined and equals α;
     - otherwise, $Tr(n)$ is undefined. (Note that, in this last case, $\mathcal{T}r(m)$ is the last element of the trace.)

Definition 3.6 A constant U *regenerates* on the trace $\mathcal{T}r$ if for some i, $\alpha_i = U$ and $\alpha_{i+1} = \alpha(U)$, where $(U = \sigma X.\alpha(X)) \in \mathcal{D}$

The trace $\mathcal{T}r$ is called ν-*trace* iff it is infinite and some ν-constant is the oldest constant which regenerates infinitely often. The trace is called μ-*trace* iff some μ-constant is the oldest constant which regenerates infinitely often on it. ∎

Remark 3.7 Observe that any infinite trace in a tableau for a guarded formula is either ν or μ-trace. This is because every rule except regeneration decreases the size of the formula and every formula is eventually reduced. This second fact is true because on any trace there cannot be two regenerations of the same definition constant without application of (*mod*) rule in between. Note that this last property does not hold for unguarded formulas.

Definition 3.8

— A quasi-model $\mathcal{PM}$ is called *pre-model* iff any infinite trace on any path of $\mathcal{PM}$ is a ν-trace.

— A quasi-refutation of γ is called a *refutation* of γ iff, on every infinite path of it, there exists a μ-trace.

We will show in the next sections that a formula is satisfiable iff it has a pre-model, and that it is unsatisfiable iff it has a refutation. The condition laid on pre-models is due to an observation that if, for some structure $\mathcal{M}$ and a state s, $\mathcal{M}, s \models \mu X.\alpha(X)$ then the smallest ordinal τ s.t. $\mathcal{M}, s \models \alpha^\tau(ff)$ must be a successor ordinal. Hence $\mathcal{M}, s \models \alpha(\alpha^\sigma(ff))$ for some $\sigma < \tau$. That is, in the process of investigating "the reasons" why a μ-formula is satisfied, we have managed to reduce the index of the formula. Since the ordinals are well ordered, it means that we will need to regenerate particular instance of the μ-formula only finitely many times. The condition on refutations is dual and obtained from analysis of a game we are going to describe in the next section.

4 Games

In this section we show that any formula γ has either a pre-model or a refutation.

Let $\mathcal{T}$ be a tableau for γ. We define an infinite game $\mathcal{G}(\mathcal{T})$ for two players, to be played on $\mathcal{T}$. Intuitively, player I will try to show that γ is satisfiable and player II that it is not. Our two players play the game as follows.

- game starts in the root of $\mathcal{T}$,
- in any (*or*) node, i.e. node where (*or*) rule was applied player I chooses one of the sons,

- in any ($all\langle\rangle$) node, player II chooses one of the sons,
- in other nodes which are not leaves, automatically the only son is chosen.

The result of such a game is either a finite or an infinite path of the tableau $\mathcal{T}$. A path can be finite only when it ends in a leaf which can be labeled either by an axiom or by an unreducible sequent but not an axiom. In the former case player II wins and in the latter case player I is the winner. If the resulting path is infinite, then player II wins iff there exists a μ-trace on the path.

Our interest in such a game is justified by the following.

Proposition 4.1 There is a strategy for the first player in the game $\mathcal{G}(\mathcal{T})$ iff there is a pre-model for γ contained in $\mathcal{T}$. There is a strategy for the second player in the game $\mathcal{G}(\mathcal{T})$ iff there is a refutation for γ contained in $\mathcal{T}$.

Proof
If there is a pre-model of γ contained in $\mathcal{T}$ then the strategy for player I is to stay in the nodes belonging to this pre-model. Conversely, a strategy for the first player induces a pre-model as follows. The root of $\mathcal{T}$ is of course included in a pre-model. If a node is included in a pre-model and this is a position where player I has to play, we select the son designated by the strategy. If player II is to play, we select all the sons. An argument for the case of refutations is similar. □

Clearly, at most one of the players may have a winning strategy in the game $\mathcal{G}(\mathcal{T})$. Therefore, a formula cannot have a pre–model and a refutation at the same time. However, it is not obvious that it must have one of them. Indeed, an infinite games may be undetermined, i.e. with no strategy existing for either player [11].

In what follows, we show that the game $\mathcal{G}(\mathcal{T})$ is nevertheless determined, *viz* there is always a winning strategy for one of the players. This fact can be deduced from general theory of infinite games of Gale and Stewart as considered, e.g. in [10, 11]. We briefly recall the definition of these games.

A game $G(Y, A)$ is defined by an *arena* Y and a *winning set* A, say, for the first player. Here $Y \subseteq X^*$ is a set of strings over some set X which is closed under initial segments, and such that any string in Y has a prolongation in Y. Let $\mathcal{F}(Y) \subseteq X^\omega$ be the set of infinite sequences which have all finite prefixes in the arena Y. The winning condition A is a subset of $\mathcal{F}(Y)$.

The players I and II pick alternatively elements in X constructing by this an infinite sequence from $\mathcal{F}(Y)$, $x_1, x_2, \ldots$. At infinity, the player I wins if the selected sequence is in A, otherwise II is the winner. The set $\mathcal{F}(Y)$ can be equipped with the Cantor topology, that is a topology induced by the metric $d(u, v) = 2^{-n}$ whenever $u \neq v$ and n is the least position such that $u(n) \neq v(n)$. Martin [10] proved that if A is a Borel set then the game is determined.

It remains to choose X, Y and A in such a way that our game could be presented as a Gale and Stewart game with a Borel winning condition. We chose X to be the set of all sequents that can appear in a tableau for γ plus some dummy symbol. Then the arena Y will be obtained from the tableau $\mathcal{T}$ by extending paths ending with a leaf by repetitions of the dummy symbol. The winning set $A \subseteq \mathcal{F}(Y)$ will be of course the set of paths that neither contain a μ-trace nor pass thorough an axiom sequent. Notice that the natural definition of this set involves an existential second-order quantifier. In order to show that A is indeed Borel, we first observe that the set of all infinite sequences over X that do not contain infinite traces and do not pass through axiom sequents, is an ω-regular language, and hence it is Σ_2^0 in the Borel hierarchy over X^ω (c.f., e.g., [18]). Now A can be obtained by intersecting this set with $\mathcal{F}(Y)$, and the last is a closed set since it is the set of all infinite paths of a tree. Thus A is a Borel subset of X^ω with the Cantor topology. In order to see that it is Borel also in $\mathcal{F}(Y)$, observe that every set $Z \subseteq \mathcal{F}(Y)$ which is closed in X^ω is also closed in $\mathcal{F}(Y)$.

Thus, from Martin's theorem and Proposition 4.1 we deduce the following.

Proposition 4.2 Let γ be arbitrary formula and $\mathcal{T}$ a tableau of γ. Then there exists a pre-model of γ in $\mathcal{T}$ or a refutation of γ in $\mathcal{T}$.

5 Characterization

In this section we prove that γ is satisfiable iff there exists a pre-model for it. From the results of the previous section, it will follow that γ is not satisfiable (*viz* $\neg\gamma$ is valid) iff there exits a refutation for $\neg\gamma$. Similar results concerning pre-model were proved in [17] and [16].

It will be convenient to use a characterization of the extremal fixpoints in terms of possibly transfinite induction. We introduce two new constructs $\mu^\tau X.\alpha(X)$ and $\nu^\tau X.\alpha(X)$, where τ is any ordinal, with the following semantics:

— $\| \mu^0 X.\alpha(X) \|_{Val} = \emptyset$, $\| \nu^0 X.\alpha(X) \|_{Val} = S$,

— $\| \sigma^{\tau+1} X.\alpha(X) \|_{Val} = \| \alpha(X) \|_{Val[\|\sigma^\tau X.\alpha(X)\|_{Val}/X]}$ (σ stands for μ or ν),

— $\| \mu^\tau X.\alpha(X) \|_{Val} = \bigcup_{\tau'<\tau} \| \mu^{\tau'} X.\alpha(X) \|_{Val}$, for τ limit ordinal,

— $\| \nu^\tau X.\alpha(X) \|_{Val} = \bigcap_{\tau'<\tau} \| \nu^{\tau'} X.\alpha(X) \|_{Val}$, for τ limit ordinal.

Then we have:

$$\begin{aligned} \| \mu X.\alpha(X) \|_{Val} &= \bigcup_\tau \| \mu^\tau X.\alpha(X) \|_{Val} \\ \| \nu X.\alpha(X) \|_{Val} &= \bigcap_\tau \| \nu^\tau X.\alpha(X) \|_{Val} \end{aligned}$$

We extend the notion of definition list from Section 3, by allowing equations of the form $(U = \sigma^{\tau} X.\alpha(X))$. The concept of expansion $\langle\!|\alpha|\!\rangle$ extends immediately.

Now we introduce the notion of the signature similar to that considered by Streett and Emerson [17].

Definition 5.1 Fix a formula β without free variables, a definition list $\mathcal{D}$ containing all definition constants occurring in β, and a state s of a model $\mathcal{M}$ such that $\mathcal{M}, s \models \langle\!|\beta|\!\rangle_{\mathcal{D}}$. Let $U_{k_1}, U_{k_2}, \ldots, U_{k_{d^\mu}}$ be all μ-constants occurring in $\mathcal{D}$. We define a *signature* of β in s, $Sig_{\mathcal{D}}(\beta, s)$, as the least (in the lexicographical ordering) sequence of ordinals $(\tau_1, \ldots, \tau_{d^\mu})$ such that $\mathcal{M}, s \models \langle\!|\beta|\!\rangle_{\mathcal{D}'}$, where $\mathcal{D}'$ is a definition list constructed from $\mathcal{D}$ by replacing the i-th definition of a μ-constant $(U_{k_i} = \mu X.\alpha_{k_i}(X)) \in \mathcal{D}$ by $(U_{k_i} = \mu^{\tau_i} X.\alpha_{k_i}(X))$ for each $i = 1, \ldots, d^\mu$.

We first show that signatures behave well with respect to formula reduction.

Lemma 5.2 For any state s of a model $\mathcal{M}$, definition list $\mathcal{D}$ and formulas $\alpha, \beta, \mu X.\alpha(X), \nu X.\alpha(X)$ such that every definition constant occurring in them occurs also in $\mathcal{D}$:

— If $\mathcal{M}, s \models \langle\!|\alpha \wedge \beta|\!\rangle_{\mathcal{D}}$ then $Sig_{\mathcal{D}}(\alpha \wedge \beta, s) = \max(Sig_{\mathcal{D}}(\alpha, s), Sig_{\mathcal{D}}(\beta, s))$.

— If $\mathcal{M}, s \models \langle\!|\alpha \vee \beta|\!\rangle_{\mathcal{D}}$ then $Sig_{\mathcal{D}}(\alpha \vee \beta, s) = Sig_{\mathcal{D}}(\alpha, s)$ or $Sig_{\mathcal{D}}(\alpha \vee \beta, s) = Sig_{\mathcal{D}}(\beta, s)$.

— If $\mathcal{M}, s \models \langle\!|\langle a\rangle\alpha|\!\rangle_{\mathcal{D}}$ then $Sig_{\mathcal{D}}(\langle a\rangle\alpha, s) = Sig_{\mathcal{D}}(\alpha, t)$ for some t s.t. $(s, t) \in R^{\mathcal{M}}(a)$.

— If $\mathcal{M}, s \models \langle\!|[a]\alpha|\!\rangle_{\mathcal{D}}$ then $Sig_{\mathcal{D}}([a]\alpha, s) = \sup\{Sig_{\mathcal{D}}(\alpha, t) : (s, t) \in R^{\mathcal{M}}(a)\}$.

— If $\mathcal{M}, s \models \langle\!|\nu X.\alpha(X)|\!\rangle_{\mathcal{D}}$ and $(V = \nu X.\alpha(X)) \in \mathcal{D}$ then $Sig_{\mathcal{D}}(\nu X.\alpha(X), s) = Sig_{\mathcal{D}}(V, s)$.

— If $\mathcal{M}, s \models \langle\!|\mu X.\alpha(X)|\!\rangle_{\mathcal{D}}$ and $(U_i = \mu X.\alpha(X)) \in \mathcal{D}$ is the i-th of the μ-constants in $\mathcal{D}$ then the prefixes of length $i - 1$ of $Sig_{\mathcal{D}}(\mu X.\alpha(X), s)$ and $Sig_{\mathcal{D}}(U_i, s)$ are equal.

— If $\mathcal{M}, s \models \langle\!|W|\!\rangle_{\mathcal{D}}$ and $(W = \sigma X.\alpha(X)) \in \mathcal{D}$ then $Sig_{\mathcal{D}}(W, s)$ is equal to $Sig_{\mathcal{D}}(\alpha(W), s)$ if W is a ν constant. If W is the i-th μ-constant in $\mathcal{D}$ then the latter signature is smaller and the difference is at the position i or less (in fact the difference is exactly at the position i but a weaker statement is enough for our purposes).

Proof

We will consider only the last case. Suppose $\mathcal{M}, s \models \langle\!|U_{k_i}|\!\rangle_{\mathcal{D}}$, where U_{k_i} is the i-th definition constant from $\mathcal{D}$. Let $(U_{k_i} = \mu X.\alpha_{k_i}(X)) \in \mathcal{D}$, remember that only definition constant older than U_{k_i} can appear in $\alpha_{k_i}(X)$. Let $Sig_{\mathcal{D}}(U_{k_i}, s) =$

$(\tau_1, \ldots, \tau_n)$ and $\mathcal{D}'$ be a definition list obtained from $\mathcal{D}$ by replacing the j-th μ-constant definition $(U_{k_j} = \mu X.\alpha_{k_j}(X)) \in \mathcal{D}$ by $(U_{k_j} = \mu^{\tau_j} X.\alpha_{k_j}(X))$ for every $j = 1, \ldots, n$. Let us denote $\langle\!|\alpha_i(X)|\!\rangle_{\mathcal{D}'}$ by $\beta(X)$. From the definition of the signature, we have $\mathcal{M}, s \models \mu^{\tau_i} X.\beta(X)$. Observe that τ_i must be a successor ordinal hence $\mathcal{M}, s \models \beta(\mu^{\tau_i - 1} X.\beta(X))$ which implies the thesis of the lemma. □

Proposition 5.3 If a positive guarded sentence γ is satisfiable then any tableau for γ contains a pre-model for γ as its subtree.

Proof
The proof is based on a Streett and Emerson's proof [17], a similar idea was also used in [15]. We present this proof in order to show the duality with the proof of the converse proposition.

Let us take a sentence γ and a model $\mathcal{M} = \langle S^{\mathcal{M}}, R^{\mathcal{M}}, \rho^{\mathcal{M}} \rangle$ in which γ is satisfiable. Let $\mathcal{D} = |\!|\gamma|\!| = (U_1 = \gamma_1), \ldots, (U_d = \gamma_d)$ and let

$$(U_{k_1} = \mu X.\alpha_{k_1}(X)), \ldots, (U_{k_{d^\mu}} = \mu X.\alpha_{k_{d^\mu}}(X))$$

be the subsequence of all μ-definitions in $\mathcal{D}$.

Given a tableau $\mathcal{T}$ for γ, we will construct a pre-model $\mathcal{PM} = \langle K, L \rangle$ as a subtree of $\mathcal{T}$. Starting from the root of $\mathcal{T}$, we will subsequently select the nodes of $\mathcal{T}$ that will be included in $\mathcal{PM}$. With every node n of $\mathcal{PM}$ under construction, we will associate a state s_n of $\mathcal{M}$ such that $\mathcal{M}, s_n \models \langle\!|L(n)|\!\rangle$.

The root of $\mathcal{T}$ becomes the root of $\mathcal{PM}$ and, for the associated state, we choose any state of $\mathcal{M}$ in which the formula γ is satisfied.

Suppose that we have already selected a node n of $\mathcal{PM}$ with an associated state s_n. We will show how to proceed from this point depending on what rule was used in the node n of $\mathcal{T}$:

1. If the (or) rule was applied to $L(n) = \alpha \vee \beta, \Gamma \vdash_{\mathcal{D}}$ then select the son of n with formula α if $Sig_{\mathcal{D}}(\alpha, s_n) \leq Sig_{\mathcal{D}}(\alpha \vee \beta, s_n)$ and the son with β otherwise. Associate the state s_n with the chosen node.

2. If the $(all\langle\rangle)$ rule was applied then select all sons of n. For any son n' of n, there is a formula of the form $\langle a \rangle \alpha$ the reduction of which resulted in the label of n'. For the node n', choose a state t such that $(s_n, t) \in R(a)$ and $Sig_{\mathcal{D}}(\langle a \rangle \alpha, s_n) \geq Sig_{\mathcal{D}}(\alpha, t)$. (Such a state exists, by Lemma 5.2.)

3. If n has no sons then it must be labeled by an unreducible sequent which is not an axiom. This is because $\langle\!|L(n)|\!\rangle$ is satisfied in the state s_n, hence a formula and its negation cannot simultaneously appear in $L(n)$.

4. For all other rules, just take the only son of n in $\mathcal{T}$ as the next node of $\mathcal{PM}$ and associate the state $s_{n'} = s_n$ with it.

We will show that $\mathcal{PM}$ constructed in this way is indeed a pre-model for γ. As we mentioned above, every leaf of $\mathcal{PM}$ is labeled by an unreducible sequent which is not an axiom. Hence $\mathcal{PM}$ is a quasi-model and it only remains to show that any infinite trace $\mathcal{T}r = \{\alpha_n\}_{n \in P}$ on any path P is a ν-trace. Suppose to the contrary, that we can find a trace $\mathcal{T}r$ such that the oldest constant regenerated infinitely often on it is some i-th μ-constant U_{k_i}. Clearly, after some point n_0, U_{k_i} must be the oldest constant which will be regenerated on the trace.

Observe that Lemma 5.2 implies that, from the point n_0, the prefix of length i of the signatures of formulas in $\mathcal{T}r$ never increases. Indeed, the only way the prefix could increase without regeneration of a definition constant older than U_{k_i} is an application of the rule (μ) with a constant older than U_{k_i}. But then the next reduction on the trace would be a regeneration of the constant older than U_{k_i}.

Lemma 5.2 also says that the prefix actually decreases after each regeneration of U_{k_i}. Since we have assumed that U_{k_i} is regenerated infinitely often, we obtain a contradiction, because the lexicographical ordering on sequences of bounded length over a well ordering is also a well ordering. This shows that every trace of $\mathcal{PM}$ is a ν-trace, hence $\mathcal{PM}$ is a pre-model. □

Now we will show implication in the other direction, i.e. we show, given a pre-model for γ, how to construct a structure where γ is satisfied.

Definition 5.4 Given a pre-model $\mathcal{PM} = \langle K, L \rangle$, the *canonical structure* for $\mathcal{PM}$ is a structure $\mathcal{M} = \langle S^{\mathcal{M}}, R^{\mathcal{M}}, \rho^{\mathcal{M}} \rangle$ such that:

1. $S^{\mathcal{M}}$ is the set of all nodes of $\mathcal{PM}$ which are either leaves or to which $(all\langle\rangle)$ rule was applied. For any node n of $\mathcal{PM}$ we will denote by des_n the closest descendant of n, or n itself, belonging to $S^{\mathcal{M}}$. (Note that des_n is well defined. Indeed, since the formulas in consideration are guarded, each infinite path in the pre-model must contain infinitely many nodes where the $(all\langle\rangle)$ rule is applied and there are no other branching points in the tree.)

2. $(s, s') \in R^{\mathcal{M}}(a)$ iff there is a son n of s with $des_n = s'$, such that $L(n)$ was obtained from $L(s)$ by reducing a formula of the form $\langle a \rangle \alpha$.

3. $\rho^{\mathcal{M}}(p) = \{s : p \text{ occurs in the sequent } L(s)\}$.

Proposition 5.5 If there exists a pre-model $\mathcal{PM}$ for a positive guarded sentence γ then γ is satisfiable in the canonical structure for $\mathcal{PM}$.

Proof
Let $\mathcal{PM} = \langle K, L \rangle$ be a pre-model of a sentence γ. Let $\mathcal{M} =< S^{\mathcal{M}}, R^{\mathcal{M}}, \rho^{\mathcal{M}} >$ be the canonical structure for $\mathcal{PM}$. Let $\mathcal{D} = \{\!|\gamma|\!\}$ be a definition list and let $(V_1 = \nu X.\beta_1(X)), \ldots, (V_{d^\nu} = \nu X.\beta_{d^\nu}(X))$ be a sublist of ν-definitions from $\mathcal{D}$.

Suppose that $\mathcal{M}, des_{n_0} \not\models \gamma$, where n_0 is the root of $\mathcal{PM}$. From now on we will proceed in a similar way to the proof of Proposition 5.3.

First we define the notion of a ν-signature, $Sig^{\nu}_{\mathcal{D}}(\alpha, s)$, which is defined for a formula α and state s s.t. $\mathcal{M}, s \not\models \alpha$:

$$Sig^{\nu}_{\mathcal{D}}(\alpha, s) = Sig_{\neg\mathcal{D}}(\neg\alpha, s)$$

Definition list $\neg\mathcal{D}$ is obtained from the definition list $\mathcal{D}$ by replacing each definition $(U = \sigma X.\alpha(X)$ with $(U = \neg\sigma X.\alpha(X))$. (Recall that we have allowed negation to occur only before propositional constants, but the negation of an arbitrary formula can be defined by the De Morgan laws and other dualities of the μ-calculus.) Observe that Lemma 5.2 translates to ν-signatures after interchanging μ with ν, $\langle\rangle$ with $[]$ and conjunction with disjunction.

Next we show that the assumption $\mathcal{M}, des_{n_0} \not\models \gamma$ implies that we can construct a μ-trace on some path P of $\mathcal{PM}$, which contradicts the hypothesis that $\mathcal{PM}$ is a pre-model of γ.

We will simultaneously construct a path P and a trace $\mathcal{T}r = \{\alpha_n\}_{n \in P}$. The first element will be of course $\alpha_{n_0} = \gamma$. Now suppose that we have constructed $\mathcal{T}r$ up to an element $\alpha_m \in L(m)$, such that $\mathcal{M}, des_m \not\models \langle\!|\alpha_m|\!\rangle$. We select the next element $\alpha_{m'}$ as follows:

— if a rule different from $(all\langle\rangle)$ was applied to $L(m)$ then there is only one son m' of m and:

- if α_m wasn't reduced by this rule then $\alpha_{m'} = \alpha_m$,
- if $\alpha_m = \varphi \wedge \psi$ was reduced then choose $\alpha_{m'} = \varphi$ if $Sig^{\nu}_{\mathcal{D}}(\varphi \wedge \psi, des_m) \geq Sig^{\nu}_{\mathcal{D}}(\varphi, des_m)$, else choose $\alpha_{m'} = \psi$,
- if $\alpha_m = \varphi \vee \psi$ then choose $\alpha_{m'}$ to be the formula which occurs in $L(m')$,
- in other subcases just take the resulting formula as the next element of the trace.

— if $(all\langle\rangle)$ rule was applied then:

- if $\alpha_m = \langle a\rangle\varphi$ then there is a son m' of m the label of which was obtained by reducing this formula. Take $\alpha_{m'} = \varphi$
- if $\alpha_m = [a]\varphi$ then, because $des_m \not\models \alpha_m$, there exists a state t such that $(des_m, t) \in R^{\mathcal{M}}(a)$ and $Sig^{\nu}_{\mathcal{D}}(\varphi, t) \leq Sig^{\nu}_{\mathcal{D}}([a]\varphi, s)$. Observe that, from the definition of our structure, this means that there is some son m' of m, such that $\varphi \in L(m')$ and $des_{m'} = t$. So let us take $\alpha_{m'} = \varphi$.

If the constructed trace were finite then from the definition of the canonical structure and the restrictions on the application of the rule $(all\langle\rangle)$ it follows that either the last element of the trace, α_m, is a propositional constant p or its

negation, or m is a leaf of $\mathcal{PM}$ and α_m is a formula of the form $[a]\psi$. Then, from the definition of the canonical structure, we have that $\mathcal{M}, des_m \models \alpha_m$, a contradiction.

Using an argument about signatures similar to the one from Proposition 5.3, we can show that the oldest constant regenerated infinitely often on $\mathcal{T}r$ must be a μ-constant. But $\mathcal{PM}$ is a pre-model hence all its traces are ν-traces, contradiction. □

It should be clear that Propositions 5.3 and 5.5 hold not only for positive guarded sentences but for all positive guarded formulas. Putting them together we obtain:

Theorem 5.6 There exists a pre-model of a positive guarded formula γ iff γ is satisfiable.

Finally, using our results from previous section we obtain also a characterization of valid formulas.

Theorem 5.7 A positive guarded formula $\neg\gamma$ is valid iff there exists a refutation for γ.

Proof
Consider any tableau $\mathcal{T}$ for γ. Since there is no model for γ, we cannot find a pre-model in $\mathcal{T}$ hence from Proposition 4.2 there is a refutation in $\mathcal{T}$.

In other direction, if $\mathcal{P}$ is a refutation for γ then we know that in the tableau $\mathcal{T}$, of which $\mathcal{P}$ is a subtree, we cannot find a pre-model. Hence from Proposition 5.3 it follows that γ is not satisfiable and $\neg\gamma$ is valid. □

6 Complexity

In this section we give bounds on the size of a refutation of a formula and consider the computational complexity of the problem for finding a refutation.

Emerson and Jutla [4] gave the exact bound of the decidability problem for the μ-calculus, by showing that the satisfiability problem for the μ-calculus is decidable in deterministic exponential time. The completeness of the problem for this complexity class follows from the lower bound for PDL due to Fischer and Ladner(1979) [6]. Streett and Emerson [17] also show a *small model theorem* for the propositional μ-calculus which, combined with the later results of Emerson and Jutla [4] tells that *if a sentence has a model then it has a finite model of the size exponential in the size of the sentence.* Considering the proof method used in [4], we can restate this result as follows.

Theorem 6.1 (Emerson and Jutla) *There is an algorithm which decides if a μ-calculus sentence γ is satisfiable in time $exp(|\gamma|)$. If this is the case, the algorithm constructs a model of the size $exp(|\gamma|)$ in time $exp(|\gamma|)$.*

The key fact needed for this theorem is a result on complexity of Rabin tree automata that is also shown in [4].

Theorem 6.2 (Emerson and Jutla) *There is an algorithm which, given a Rabin automaton with n states and m pairs, decides in time $\mathcal{O}(n^m)$ whether the automaton accepts some tree. If it is the case, the algorithm constructs in time $\mathcal{O}(n^m)$ a regular tree of the size $\mathcal{O}(n)$ accepted by the automaton.*

This last result can be also used for obtaining the upper bound for the size and the time of construction of refutations.

Let a μ-calculus sentence γ be given. Let Σ be the alphabet consisting of all the sequents that can appear in a tableau of γ. Observe that $|\Sigma| = \exp(|\gamma|)$. Since the rules of the system S are nondeterministic, there may be many tableaux of γ. Clearly, we can construct a tableau $\mathcal{T}$ of γ, which is a *regular tree* of size $\exp(|\gamma|)$, and the construction can be performed in time $\exp(|\gamma|)$. We construct a Büchi automaton on infinite words over Σ which nondeterministically chooses a trace from a path in $\mathcal{T}$ and accepts iff this is a μ-trace. It is easy to see that $\mathcal{O}(|\gamma|)$ states are sufficient for this job. Applying the Safra determinization construction [14], we get a Rabin automaton on ω-words with $\exp(|\gamma|)$ states and $\mathcal{O}(|\gamma|)$ pairs. Finally, following the construction of $\exp(|\gamma|)$ states and $\mathcal{O}(|\gamma|)$ pairs, which accepts precisely the refutations of γ. This automaton is actually obtained as a product of the above Rabin automaton on ω-words and an automaton checking local consistency i.e. that a tree is a quasi-refutation contained in $\mathcal{T}$. We can therefore state the following.

Theorem 6.3 There is an algorithm which, given a μ-calculus sentence γ, constructs a model of size $\exp(|\gamma|)$ or a refutation of size $\exp(|\gamma|)$. The algorithm runs in time $\exp(|\gamma|)$.

Acknowledgement

The second author would like to express his deep gratitude to professor Dexter Kozen for introduction to the μ-calculus and long discussions which inspired ideas of this work.

References

[1] Mads Dam. CTL* and ECTL* as a fragments of the modal μ-calculus. In *CAAP'92*, volume 581 of *LNCS*, pages 145–165, 1992.

[2] E. Allen Emerson and E.M. Clark. Characterizing correctness properties of parallel programs using fixpoints. In *Seventh International Colloqium on Automata, Languages and Programming*, pages 169–181, 1980.

[3] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. In *29th IEEE Symp. on Foundations of Computer Science*, 1988.

[4] E. Allen Emerson and Charanjit S. Jutla. On simultaneously determinizing and complementing ω-automata. In *LICS'89*, 1989.

[5] E.Allen Emerson and C.S. Jutla. Tree automata, mu calculus and determinacy. In *Proc. FOCS 91*, 1991.

[6] M.J. Fisher and R.E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[7] Yuri Gurevich and Leo Harrington. Trees, automata and games. *Journal of the ACM*, 1982.

[8] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[9] Dexter Kozen and R. Parikh. An elementary proof of the completeness of the PDL. *Theoretical Computer Science*, 14:113–118, 1981.

[10] D.A. Martin. Borel determinacy. *Ann. Math.*, 102:363–371, 1975.

[11] Y.N. Moschovakis. *Descriptive Set Theory*, volume 100 of *Studies in Logic*. North-Holand, 1980.

[12] A.A. Muchnik. Games on infinite trees and automata with dead ends. *Semiotics and Information*, 24:17–44, 1984. in Russian.

[13] D.M.R. Park. Fixpoint inductinon and proof of program semantics. In B. Meltzer and D. Michie, editors, *Machine Inteligence*, volume 5, pages 59–78. Edinburgh University Press, 1970.

[14] Shmuel Safra. On the complexity of ω-automata. In *29th IEEE Symp. on Foundations of Computer Science*, 1988.

[15] Colin P. Stirling and David J. Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89:161–171, 1991.

[16] C.S. Stirling. Modal and teporal logics for processes. to appear in LNCS.

[17] Robert S. Street and E. Allan Emerson. An automata theoretic procedure for the propositional mu-calculus. *Information & Computation*, 81:249–264, 1989.

[18] Wolfgang Thomas. Automata on infinite objects. In J.van Leeuven, editor, *Handbook of Theoretical Computer Science Vol.B*, pages 995–1072. Elsvier, 1990.

[19] Igor Walukiewicz. Gentzen-type axiomatization for pal. *Theoretical Computer Science*, 118:67–79, 1993.

[20] Igor Walukiewicz. On completeness of the μ-calculus. In *LICS '93*, pages 136–146, 1993.

A An Example

In this section we would like to show an example of the use of refutations. Suppose that we are given two specifications of modules. The first one

$$F_1 = \mu X.[a]X \vee (\nu Y.[a]Y \wedge C)$$

requires that on any execution of the program, from some point a condition C must be satisfied. The second one says that there always should be an execution such that, from every point it can reach a state where C is unsatisfied

$$F_2 = \nu X.\langle a\rangle X \wedge (\mu Y. < a > Y \vee \neg C)$$

We would like to find out why this two requirements are contradictory, or, in other words, why the conjunction of the two formulas is unsatisfiable.

First we name all fixpoint subformulas of $F_1 \wedge F_2$ and obtain a definition list $\mathcal{D}$:

$$\begin{aligned} U_1 &= \mu X.[a]X \vee (\nu Y.[a]Y \wedge C) \\ V_1 &= \nu Y.[a]Y \wedge C \\ V_2 &= \nu X.\langle a\rangle X \wedge (\mu Y. < a > Y \vee \neg C) \\ U_2 &= \mu Y. < a > Y \vee \neg C \end{aligned}$$

Below we present a refutation of $F_1 \wedge F_2$; some simplifications were used for notational convenience.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F_1, F_2 \vdash_{\mathcal{D}}}}{U_1, V_2 \vdash_{\mathcal{D}}}}{[a]U_1 \vee V_1, V_2 \vdash_{\mathcal{D}}}}{[a]U_1, V_2 \vdash_{\mathcal{D}}} \qquad \dfrac{}{V_1, V_2 \vdash_{\mathcal{D}}}}{[a]U_1, \langle a\rangle V_2, \langle a\rangle U_2 \vee \neg C \vdash_{\mathcal{D}}}}{[a]U_1, \langle a\rangle V_2, \langle a\rangle U_2 \vdash_{\mathcal{D}}} \quad \dfrac{[a]U_1, \langle a\rangle V_2, \neg C \vdash_{\mathcal{D}}}{\boxed{U_1, V_2 \vdash_{\mathcal{D}}}}}{\boxed{U_1, V_2 \vdash_{\mathcal{D}}}}$$

In this part of the refutation two nodes in frames are the same as the node just below the root, hence we have two cycles here. The part of the tableau starting from the node labeled $V_1, V_2 \vdash_{\mathcal{D}}$ will be presented below.

From first part of the refutation, we can read that there is a μ-trace with regenerations of U_1 on both cycles. This shows that in any potential model of $F_1 \wedge F_2$, we must have a state which satisfies $\langle\!\langle V_1, V_2 \vdash_{\mathcal{D}} \rangle\!\rangle$. The rest of the tableau for $F_1 \wedge F_2$ shows that there is no way to satisfy $\langle\!\langle V_1, V_2 \vdash_{\mathcal{D}} \rangle\!\rangle$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{V_1, V_2 \vdash_{\mathcal{D}}}}{[a]V_1, C, \langle a\rangle V_2, \langle a\rangle U_2 \vee \neg C \vdash_{\mathcal{D}}}}{[a]V_1, \langle a\rangle V_2, \langle a\rangle U_2 \vdash_{\mathcal{D}}} \quad \dfrac{}{\neg C, C \vdash_{\mathcal{D}}}}{V_1, U_2 \vdash_{\mathcal{D}}}}{[a]V_1 \wedge C, \langle a\rangle U_2 \vee \neg C \vdash_{\mathcal{D}}}}{\boxed{V_1, U_2 \vdash_{\mathcal{D}}}} \qquad \dfrac{}{C, \neg C \vdash_{\mathcal{D}}}$$

In this part of the refutation we have one cycle, marked by the sequent in a frame, and two leaves labeled by axioms. This part of the refutation shows that in order to satisfy $V_1 \wedge V_2$, one must satisfy $V_1 \wedge U_2$. But this is impossible because the marked cycle has a μ-trace with a regeneration of U_2.