BRICS

Basic Research in Computer Science

BRICS NS-95-1 I. Walukiewicz: The Propositional μ-calculus: Completeness and Related Results

Notes on the Propositional μ-calculus: Completeness and Related Results

Igor Walukiewicz

BRICS Notes Series NS-95-1

ISSN 0909-3206 February 1995

See back inner page for a list of recent publications in the BRICS Notes Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@daimi.aau.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.aau.dk/BRICS/`
`ftp ftp.brics.aau.dk (cd pub/BRICS)`

Notes on the Propositional μ-calculus: Completeness and Related Results

Igor Walukiewicz
BRICS [1]
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark
e-mail: igw@daimi.aau.dk

February, 1995

[1]**B**asic **R**esearch **i**n **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

Contents

1 Introduction

In this notes we consider propositional μ-calculus as introduced by Kozen in [9]. The main purpose of these notes is to present the completeness proof of the Kozen's axiomatisation of the μ-calculus [9]. To achieve this goal we develop tools which allow us to give relatively simple proofs of results for the logic like:

- syntactic characterisation of satisfiability and validity,
- small model theorem,
- decidability,
- equivalence of the μ-calculus over binary trees and Rabin automata,
- a notion of disjunctive formula and the proof that every formula is equivalent to a disjunctive formula,
- linear satisfiability checking algorithm for disjunctive formulas.

These notes are intended to supplement a 6 hours course given in February 1995 at BRICS centre. Because of the time limit some of the topics naturally connected to the subject have been omitted. In these notes we have deliberately tried to minimise the use of automata theory. This is why the remarks about the correspondence between the μ-calculus and automata theory are grouped in one small subsection, which is not intended to give the full overview of the correspondence. This choice also means that we will not consider the model checking problem; although some tools we develop are very similar to so called local model checking [16], we feel that one cannot give the full overview of the subject without mentioning automata theory and alternating automata in particular.

The contents of these notes are based on three papers [14, 8, 20].

1.1 Synopsis

We start by giving some preliminary definitions. First we present the syntax and the semantics of the μ-calculus. Then we introduce the notions of *positive* and *guarded* formulas. We also define a *new modality operator* which can replace two standard modalities. In these notes we will mostly work with positive guarded formulas and use the new modality operator. Next we introduce *binding functions* which is a tool we use to deal with subformulas of a given formula. We finish this preliminary section by defining automata on infinite strings and trees and stating theorems showing equivalence between Rabin and *parity* conditions. We will use automata in two of the proofs.

In the next section we present what we call operational semantics of formulas. We give a characterisation of satisfiability of a given formula in a given state by means of *markings* of infinite *tableaux*. This result makes very explicit the main tools used in the fundamental paper of Streett and Emerson [18].

After that we show some applications of operational semantics. We prove the small model theorem and the decidability result (i.e. the results from [18]).

We also define the notion of *refutation* and state the characterisation of validity (or rather unsatisfiability) by means of refutations [14]. We finish this section by presenting a new application of the operational semantics, namely *tableau equivalence*. We show that if two formulas have equivalent tableaux then they are equivalent. This result turns out to be very useful.

The next step is the definition of disjunctive formulas and the proof that every formula is equivalent to a disjunctive formula.

The following section gives some results concerning disjunctive formulas. We show that satisfiability testing is linear for this class of formulas and that there is a very straightforward method of constructing models for satisfiable disjunctive formulas. We also discuss the connections between disjunctive formulas and automata. We consider μ-calculus restricted to binary trees. In this case there is a straightforward connection between disjunctive formulas and parity automata on trees. In this restricted case it is still true that every formula is equivalent to a disjunctive formula. Hence we obtain (yet another) proof of Rabin's complementation lemma and the proof of the equivalence between monadic second order logic of two successors and the μ-calculus over binary trees [15, 5]. At the end of this section we argue that disjunctive formulas give rise to a new concept of automata. In contrast with usual notions of automata on infinite strings or trees, the notion of a run of these automata is defined for arbitrary transition systems. Hence these automata are designed to cope with arbitrary branching of the structure. Moreover they have exactly the same expressive power as the μ-calculus. A different concept of automata which can adapt to structures with varying degrees of nodes was proposed in [1] but this automata are in general stronger than the μ-calculus.

After this development we come back to the questions of provability in Kozen's axiom system. We first consider some restricted cases. It turns out that there is a very simple way of proving the negation of every unsatisfiable disjunctive formula. Then we the recall the definition of an aconjunctive formula from [9] and propose a small generalisation of this concept called *weakly aconjunctive formula*. In [9] Kozen showed a method of proving the negation of every unsatisfiable aconjunctive formula. As we need a generalisation of this result we introduce the notion of *thin refutation* to isolate the cases where the method still works.

Finally we prove our main theorem which states that for every formula φ there is a disjunctive formula $\widehat{\varphi}$ for which the implication $\varphi \Rightarrow \widehat{\varphi}$ is provable. This gives us the completeness theorem as an easy corollary. The proof is done by induction on the structure of φ and uses tableau equivalence in case of the greatest fixpoint operator and conjunction. In case of the least fixpoint we need to generalise the concept of tableau equivalence to *tableau consequence*. This notion is defined using infinite games and can be seen as an extension of the notion of simulation of transition systems.

2 Preliminaries

2.1 Syntax and semantics of the μ-calculus

Let $Prop = \{p, q, \ldots\} \cup \{\bot, \top\}$ be a set of propositional letters, $Var = \{X, Y, \ldots\}$ a set of variables and $Act = \{a, b, \ldots\}$ a set of actions. The set of formulas of the μ-calculus over this three sets is the smallest set such that:

- propositional letters and variables are formulas,
- if α, β are formulas then so are : $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$,
- if $a \in Act$ and α is a formula then $\langle a \rangle \alpha$ and $[a]\alpha$ are formulas. Later we will introduce one more construct $(a \rightarrow \Gamma)$, where Γ is a finite set of formulas,
- Let $\alpha(X)$ be a formula. We will say that an occurrence of X in $\alpha(X)$ is *positive* iff it is proceeded by an even number of negations. If all occurrences of X in $\alpha(X)$ are positive then $\mu X.\alpha(X)$ and $\nu X.\alpha(X)$ are formulas. Variable X is *bound* in $\mu X.\alpha(X)$ and $\nu X.\alpha(X)$.

Remark 2.1 We will assume that the modalities ($\langle a \rangle$, $[a]$) and the fixpoint operators (μ, ν) bind weaker than propositional connectives. Hence $\mu X.\langle a \rangle X \vee p$ is a shorthand for $\mu X.(\langle a \rangle (X \vee p))$. Later on in these notes we will use $(a \rightarrow \Gamma)$ construct which should simplify formula parsing.

In the following, $\alpha, \beta, \gamma, \varphi, \psi, \ldots$ will denote formulas, and $\Gamma, \Delta, \Theta, \Omega, \ldots$ will denote finite sets of formulas. We shall use σ to denote either μ or ν. Variables, propositional letters and their negations will be called *literals*.

Formulas are interpreted in *transition systems* of the form $\mathcal{M} = \langle S, R, \rho \rangle$, where:

- S is a nonempty set of states,
- $R : Act \rightarrow \mathcal{P}(S \times S)$ is a function assigning a binary relation on S to each action in Act.
- $\rho : Prop \rightarrow \mathcal{P}(S)$ is a function assigning a set of states to every propositional letter.

For a given model $\mathcal{M}$ and a valuation $Val : Var \rightarrow \mathcal{P}(S)$, 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):

$$\| p \|_{Val} = \rho(p) \quad \| \bot \|_{Val} = \emptyset \quad \| \top \|_{Val} = \emptyset$$

$$\begin{aligned}
\| X \|_{Val} &= Val(X) \\
\| \neg p \|_{Val} &= S - \rho(p) \\
\| \alpha \vee \beta \|_{Val} &= \| \alpha \|_{Val} \cup \| \beta \|_{Val} \\
\| \alpha \wedge \beta \|_{Val} &= \| \alpha \|_{Val} \cap \| \beta \|_{Val} \\
\| \langle a \rangle \alpha \|_{Val} &= \{s : \exists s'.(s, s') \in R(a) \wedge s' \in \| \alpha \|_{Val}\} \\
\| [a]\alpha \|_{Val} &= \{s : \forall s'.(s, s') \in R(a) \ \Rightarrow s' \in \| \alpha \|_{Val}\} \\
\| \mu X.\alpha(X) \|_{Val} &= \bigcap \{T \subseteq S : \| \alpha \|_{Val[T/X]} \subseteq T\} \\
\| \nu X.\alpha(X) \|_{Val} &= \bigcup \{T \subseteq S : T \subseteq \| \alpha \|_{Val[T/X]}\}
\end{aligned}$$

where $Val[T/X]$ is the valuation such that, $Val[T/X](X) = T$ and $Val[T/X](Y) = Val(Y)$ for every $Y \neq X$.

Remark 2.2 The meaning of a formula is the set of states where it is satisfied. We can consider that a formula $\alpha(X)$ with a free variable X defines a function from sets of states to sets of states $f_\alpha : \mathcal{P}(S) \to \mathcal{P}(S)$ which maps $T \subseteq \mathcal{P}(S)$ to $\| \alpha(X) \|_{Val[T/X]}$. If X occurs only positively in $\alpha(X)$ then f_α is a *monotone function*, i.e., $f_\alpha(T) \subseteq f_\alpha(T')$ whenever $T \subseteq T'$. Monotone functions on complete latices have the least and the greatest fixpoint. The least fixpoint is the intersection of all *pre-fixpoints*, i.e., sets T such that $f_\alpha(T) \subseteq T$. The greatest fixpoint is the sum of all *post-fixpoints*, i.e., sets T such that $T \subseteq f_\alpha(T)$.

The following proposition states that both $\mu X.\alpha(X)$ and $\nu X.\alpha(X)$ are fixpoints of the corresponding function. Its proof follows directly from the semantics.

Proposition 2.3 For any model $\mathcal{M}$ and valuation Val.

$$\| \mu X.\alpha(X) \|^{\mathcal{M}}_{Val} = \| \alpha(\mu X.\alpha(X)) \|^{\mathcal{M}}_{Val} \quad \| \nu X.\alpha(X) \|^{\mathcal{M}}_{Val} = \| \alpha(\nu X.\alpha(X)) \|^{\mathcal{M}}_{Val}$$

We shall say that a formula α is *satisfied* in a state s of a model $\mathcal{M}$ with a valuation Val, in symbols $\mathcal{M}, s, Val \models \alpha$, when $s \in \| \alpha \|^{\mathcal{M}}_{Val}$. Formula α is *valid* iff for every model $\mathcal{M}$, state s and valuation Val we have $\mathcal{M}, s, Val \models \alpha$. We will use $\alpha \Rightarrow \beta$ as an abbreviation of $\neg\alpha \vee \beta$. We say that two formulas α and β are *semantically equivalent* iff formula $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$ is valid. For example Proposition 2.3 states that $\mu X.\alpha(X)$ is semantically equivalent to $\alpha(\mu X.\alpha(X))$. We will usually say just equivalent instead of semantically equivalent.

Proposition 2.4 (Dualities) *For every formula α and action a, the formula $\langle a \rangle\alpha$ is semantically equivalent to $\neg[a]\neg\alpha$. Formula $\mu X.\alpha(X)$ is semantically equivalent to $\neg\nu X.\neg\alpha(\neg X)$.*

Exercises:

1 Prove Propositions 2.3 and 2.4.

2 Show that $\mathcal{M}, t_0 \models \mu X.[a]X$ iff there is no infinite path $t_0, t_1, t_2, \ldots$ with $(t_i, t_{i+1}) \in R(a)$ for $i = 0, 1, 2, \ldots$ What is the meaning of $\nu X.[a]X$?

3 Is there a transition system $\mathcal{M}$ and a valuation Val such that for every formula $\alpha(X)$:

$$\| \mu X.\alpha(X) \|^{\mathcal{M}}_{Val} = \| \nu X.\alpha(X) \|^{\mathcal{M}}_{Val}$$

2.2 Restrictions and extensions of the syntax

As our tools will be mostly syntactic, the form of the formula itself will be important to us. We are interested in restrictions on the use of some of the connectives of the calculus which do not restrict its expressive power. The main result of this kind will be the proof that every formula is equivalent to a disjunctive formula but this will be done only in Section 5. Here we will introduce the notions of *positive* and *guarded* formulas and show that every formula is equivalent to a positive guarded formula. We also introduce a special construct of the form $(a \to \Gamma)$ which can replace both $\langle a \rangle$ and $[a]$ modalities.

Definition 2.5 (Positive and guarded formulas) *A formula is* positive *iff all negations in the formula appear only before literals. Variable X in a formula $\alpha(X)$ is* guarded *iff every occurrence of X in $\alpha(X)$ is in the scope of some modality operator $\langle\ \rangle$ or $[\ ]$. We say that a formula φ is* guarded *iff for every subformula $\sigma X.\alpha(X)$ of φ, variable X is guarded in $\alpha(X)$.*

Example 2.6 The formula $\nu Y.X \vee \langle a \rangle Y$ is guarded while $\mu X.\nu Y.X \vee \langle a \rangle Y$ is not because X is not guarded.

In the following we will often restrict ourselves to positive guarded formulas. This is not essential but substantially simplifies many notions. The following proposition shows that such a restriction is "harmless" at least from the semantical point of view.

Proposition 2.7 (Kozen) *Every formula is equivalent to some positive guarded formula.*

Proof
Let φ be a formula, we first show how to obtain an equivalent guarded formula. The proof proceeds by induction on the structure of the formula with the only nontrivial cases being fixpoint constructors. We present here the case for the least fixpoint. The case for the greatest fixpoint is similar.

Assume that $\varphi = \mu X.\alpha(X)$ and $\alpha(X)$ is a guarded formula. Suppose X is unguarded in some subformula of $\alpha(X)$ of the form $\sigma Y.\beta(Y, X)$. Variable Y is guarded in $\sigma Y.\beta(Y, X)$ by the assumption. We can use the equivalence of $\sigma Y.\beta(Y, X)$ with $\beta(\sigma Y.\beta(Y, X), X)$ to obtain a formula with all unguarded occurrences of X outside the fixpoint operator. This way we obtain a formula equivalent to $\alpha(X)$ 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 the conjunctive normal form (considering fixpoint formulas and 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 some of $\alpha_j(X)$ may be just $\bot$ and $\beta(X)$ may be $\top$. Variable X occurs only positively in (1) because it did so in our original formula. Formula (1) is equivalent to

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

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

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

Let γ stand for $\mu X.\bar{\alpha}(X) \wedge \beta(X)$. To prove another implication it is enough to observe that γ is a pre-fixpoint of $\mu X.(X \vee \bar{\alpha}(X)) \vee \beta(X)$ as the following calculation shows:

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

If φ is a guarded formula then we use dualities of the μ-calculus (see Proposition 2.4) to produce an equivalent positive formula. It is easy to see that it will be still a guarded formula. □

In analysing the role of each connective of the μ-calculus we will see that the conjunction has two functions. To help us distinguish this two functions we introduce the following concept which is related to a normal form for the simple modal logic.

Definition 2.8 We extend the syntax of the μ-calculus by allowing new construction of the form $(a \rightarrow \Gamma)$, where a is an action and Γ is a finite (possibly empty) set of formulas. Such a formula is semantically equivalent to $\bigwedge\{\langle a\rangle\alpha : \alpha \in \Gamma\} \wedge [a] \bigvee \Gamma$. We adopt the convention that the conjunction of the empty set of formulas is the formula $\top$ and the disjunction of the empty set is $\bot$.

Remark 2.9 A formula $\langle a\rangle\alpha$ is equivalent to $(a \rightarrow \{\alpha, \top\})$ and a formula $[a]\alpha$ is equivalent to $(a \rightarrow \{\alpha\}) \vee (a \rightarrow \emptyset)$. It follows that any formula can be written with this new construction in place of modalities.

Remark 2.10 The notions of positive and guarded formulas can be extended in a straightforward way to formulas with this new construct.

Exercises:

1 Give a proof of the induction step for the greatest fixpoint case in the proof of Proposition 2.7.

2 Is there a transition system $\mathcal{M}$ and a valuation *Val* such that for every *guarded* formula $\mu X.\alpha(X)$:

$$\| \mu X.\alpha(X) \|_{Val}^{\mathcal{M}} = \| \nu X.\alpha(X) \|_{Val}^{\mathcal{M}}$$

3 Find a positive guarded formula equivalent to $\neg(a \rightarrow \Gamma)$ where Γ is a set of positive guarded formulas.

2.3 Binding definitions

In this subsection we would like to introduce some tools to deal with subformulas of a given formula. They are very similar to those used in [9] or [16].

We would like to have a different name (which will be a variable) for every fix-point subformula of a given formula. We will also introduce a notion of a binding function which will associate subformulas to names.

Definition 2.11 (Binding) *We call a formula* well named *iff every variable is bound at most once in the formula. For a variable X bound in a well named formula α there exists the unique subterm of α of the form $\sigma X.\beta(X)$, from now on called the* binding definition of X in α *and denoted $\mathcal{D}_\alpha(X)$. We will omit subscript α when it causes no ambiguity. We call X a ν-*variable *when $\sigma = \nu$, otherwise we call X a μ-*variable.

The function $\mathcal{D}_\alpha$ assigning to every bound variable its binding definition *in α will be called the* binding function *associated with α.*

Remark 2.12 Note that every formula is equivalent to a well-named one with some ad hoc consistent renaming of bound variables. The substitution of a formula β for all free occurrences of a variable X in α, denoted $\alpha[\beta/X]$, can be made modulo some consistent renaming of bound variables of β, so that the obtained formula $\alpha[\beta/X]$ is still well-named.

Definition 2.13 (Dependency order) *Given a formula α we define the* dependency order *over the bound variables of α, denoted $\leq_\alpha$, as the least partial order relation such that if X occurs in $\mathcal{D}_\alpha(Y)$ and $\mathcal{D}_\alpha(Y)$ is a subformula of $\mathcal{D}_\alpha(X)$ then $X \leq_\alpha Y$. We will say that a bound variable Y depends on a bound variable X in α when $X \leq_\alpha Y$.*

Remark 2.14 It is not the case that $X \leq_\alpha Y$ iff $\mathcal{D}(Y)$ is a subformula of $\mathcal{D}(X)$. For example when $\alpha = \mu X.(b \to \{X\}) \vee \nu Y.(a \to \{Y\})$ then variables X and Y are incomparable in $\leq_\alpha$ ordering. On the other hand if α is $\mu X.\nu Y.(a \to \{X\}) \vee \mu Z.(a \to \{Z \vee Y\})$ then $X \leq_\alpha Z$.

Definition 2.15 Given a formula α with an associated binding function $\mathcal{D}_\alpha$, for every subformula β of α, we will define the *expansion* of β with respect to $\mathcal{D}_\alpha$ as:

$$\langle\!\langle \beta \rangle\!\rangle_{\mathcal{D}_\alpha} = \beta[\mathcal{D}_\alpha(X_n)/X_n]\cdots[\mathcal{D}_\alpha(X_1)/X_1]$$

where the sequence $(X_1, X_2, \ldots, X_n)$ is a linear ordering of all bound variables of α compatible with the dependency partial order, i.e. if $X_i \leq_\alpha X_j$ then $i \leq j$.

Proviso: If not otherwise stated all considered formulas are assumed to be well named, positive and guarded. We also assume that all occurrences of $\langle a \rangle$ and $[a]$ modalities are replaced by appropriate formulas using $(a \to \Gamma)$ construct.

2.4 Automata on infinite objects

Let us briefly recall the concepts of finite automata on infinite words and trees (see [19] for a survey). We will need automata in some of the proofs. The kind of automata we would be interested most are automata with so called *parity or Rabin chain condition*. This condition is a special case of both Rabin and Street conditions and yet it is as powerful as any of these. As we will see this type of condition has very strong connections with the μ-calculus.

An *infinite word* over a finite alphabet Σ is a function $w : \mathcal{N} \to \Sigma$. We will sometimes use $w_0 w_1 \ldots$ notation for such words. The set of all infinite words over Σ is denoted Σ^ω; the set of all finite words over Σ is denoted Σ^*. An *infinite tree* over Σ is a function $t : \{l, r\}^* \to \Sigma$. Here l, r are two letters meaning left and right respectively. The root of a tree is an empty word denoted ε.

Definition 2.16 (Finite automata) *A finite automaton is a tuple $\langle Q, \Sigma, s_0, \delta\rangle$, where Q is a finite set of states, Σ is a finite set called an alphabet, $s_0 \in Q$ is an initial state, $\delta \subseteq Q \times \Sigma \times Q$ is a transition relation.*

Finite automata which run on infinite objects can be equipped with different accepting conditions. We will restrict ourselves only to two types of conditions.

Definition 2.17 (Acceptance conditions) *Let $\mathcal{A}$ be an automaton as above.* Parity acceptance condition *is given by a function $\mathcal{C} : Q \to \mathcal{N}$ assigning a natural number to every state of the automaton.* Rabin acceptance condition *is given by a set of pairs of subsets of Q, $\{(R_i, G_i) : i = 1, \ldots, k\}$.*

An finite automaton with a parity condition will be called parity automaton. Rabin automata *are finite automata with Rabin conditions.*

Definition 2.18 (Recognition) *A sequence $q_0, q_1, \ldots$ is called a run of $\mathcal{A}$ on a word $w_1 w_2 \ldots \in \Sigma^\omega$ iff $q_0 = s_0$ and for any $i \in \mathcal{N}$ we have $(q_i, w_i, q_{i+1}) \in \delta$. Let $Inf(q_0, q_1, \ldots)$ denote the set of states which appear infinitely often in the sequence.*

A run $r : \mathcal{N} \to Q$ is accepting *with respect to a parity condition $\mathcal{C}$ iff* $\min\{\mathcal{C}(q) : q \in Inf(r)\}$ *is even. The run is accepting with respect to a Rabin condition $\{(R_i, G_i) : i = 1, \ldots, k\}$ iff there exists an index $j \in \{1, \ldots, k\}$ such that $R_j \cap Inf(r) = \emptyset$ and $G_j \cap Inf(r) \neq \emptyset$.*

A word is accepted by the automaton iff it admits an accepting run. The language recognised *by $\mathcal{A}$ is the set of words accepted by $\mathcal{A}$.*

Remark 2.19 Parity automata are special case of Rabin automata. A parity condition $\mathcal{C} : Q \to \mathcal{N}$ is equivalent to the Rabin condition $\{(R_i, G_i) : R_i = \{q : \mathcal{C}(q) < 2i\}, G_i = \{q : \mathcal{C}(q) = 2i\}, i \in \mathcal{N}\}$. It is also true that for every Rabin automaton there exists a parity automaton recognising the same language but this translation is more complicated and involves increasing the number of states of the automaton.

Example 2.20 The automata presented in Figure 1 accept the language consisting of those infinite words over $\Sigma = \{a, b\}$ which contain only finitely many a's or only finitely many b's. Double circles denote initial states. Transition relation is defined by labeled arcs. The upper automaton is a Rabin automaton and its acceptance condition is written next to it. The lower automaton is a parity automaton, its acceptance condition is given just by the state numbers.

An automaton is called *deterministic* iff for every $q \in Q$ and $a \in \Sigma$ there is at most one $q' \in Q$ such that $(q, a, q') \in \delta$. McNaughton [11] showed that every Rabin automaton over infinite strings is equivalent to a deterministic one (Actually he showed this for different acceptance condition, so called Muller condition). Both automata in Figure 1 are deterministic. The following theorem in a more general form can be found in [12].

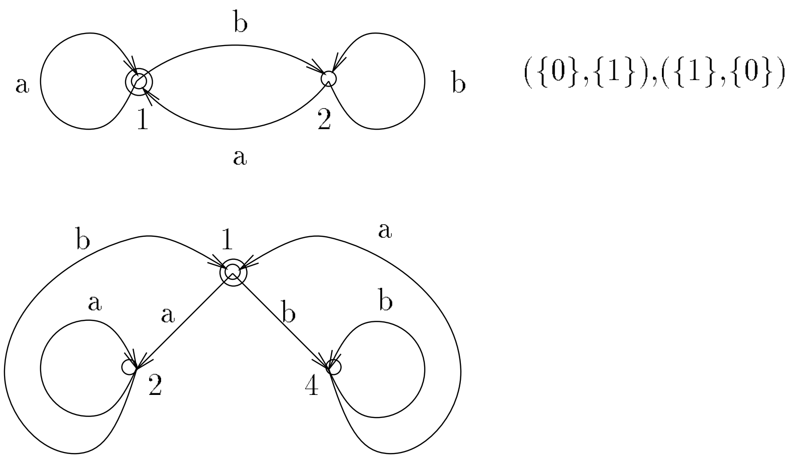

Figure 1: An automaton for $\{w \in \{a,b\}^* : |w|_a < \infty \text{ or } |w|_b < \infty\}$

Theorem 2.21 (Mostowski) *For every Rabin automaton on infinite words there is a deterministic parity automaton recognising the same language.*

Definition 2.22 *Tree automaton* is a tuple $\langle Q, \Sigma, s_0, \delta \rangle$ where Q, Σ, s_0 are as before and $\delta \subseteq Q \times \Sigma \times Q \times Q$. A run of a tree automaton on a tree $t : \{l,r\}^* \to \Sigma$ is a function $r : \{l,r\}^* \to Q$ such that $r(\varepsilon) = s_0$ and for any $w \in \{l,r\}^*$ we have $(r(w), t(w), r(wl), r(wr)) \in \delta$.

Definition 2.23 A tree automaton with parity or Rabin condition will be called *parity* or *Rabin* tree automaton respectively. A run of such an automaton is accepting iff for every path $\mathcal{P}$ of the tree the sequence of states on $\mathcal{P}$ satisfies corresponding condition.

Example 2.24 In Figure 2 we present a tree automaton over $\Sigma = \{a,b\}$ recognising the set of trees containing at least one a.

The states are marked with circles and the transitions are represented by boxes. The initial state is 1 and there are three transitions from it. Transition $(1,b,1,2)$ is represented by the box to the left labeled b. The parity acceptance condition is given by the state numbers.

Remark 2.25 A tree automaton is said to be deterministic iff for every $q \in Q$ and $a \in \Sigma$ there is at most one pair $(q_l, q_r) \in Q$ such that $(q, a, q_l, q_r) \in \delta$. It is not true that for every Rabin tree automaton there is an equivalent deterministic Rabin tree automaton.

Theorem 2.26 (Mostowski [12], Emerson & Jutla [5]) *For every Rabin tree automaton there is an equivalent parity tree automaton.*

As we said in the introduction we are not going to consider the model checking problem, i.e. the problem of deciding whether a formula is true in a given state of a given finite structure. Let us mention just one result which shows the connection between the μ-calculus and parity conditions.

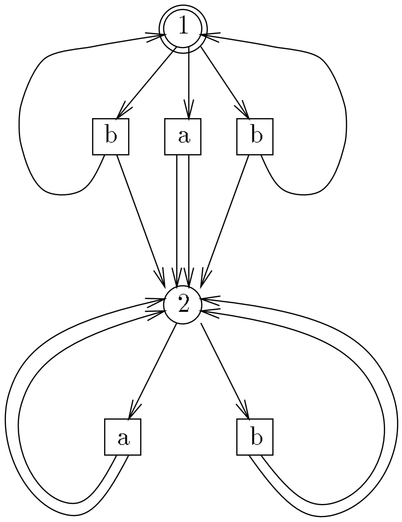

Figure 2: Tree automaton

Theorem 2.27 ([2]) *Model checking problem is equivalent under linear reductions to the testing emptiness of parity automata on trees. The problem is in NP and co-NP. On the other hand testing emptiness of Rabin automata is NP-complete.*

Exercises:

1 Construct Rabin and parity string automata recognising the complement of the language considered in Figure 1.

2 Show that there is no deterministic tree automaton recognising the same language as the tree automaton considered in Figure 2.

3 Tableaux, markings and "operational semantics"

In this section we present the characterisation of satisfiability by means of markings of infinite tableaux. We call this "operational semantics" of formulas because the idea of the characterisation comes from considering μ-calculus formulas as automata-like devises checking the properties of the structure. In the first subsection we give some motivating examples. Then we formalise the ideas and preset the characterisation result.

According to our proviso we will assume that all the formulas considered in this section are well named, positive, guarded and use $(a \to \Gamma)$ construct instead of $\langle a \rangle$ and $[a]$ modalities.

3.1 Formulas as automata

Here we would like to give some intuitions about the operational semantics for the μ-calculus formulas. We will pursue the idea that formulas are automata-like devices checking properties of the model.

Let us consider the task of checking whether a set of formulas Γ is satisfied in a state s of a model $\mathcal{M} = \langle S, R, \rho\rangle$ with a valuation *Val*. We will use $\{\alpha, \Gamma\}$ to stand for $\{\alpha\} \cup \Gamma$. As we will never consider sets of sets of formulas this notation should not be very confusing. The algorithm for checking satisfiability may look as follows:

1. To check that $\{\alpha \vee \beta, \Gamma\}$ is satisfied in s, check $\{\alpha, \Gamma\}$ or check $\{\beta, \Gamma\}$.

2. To check $\{\alpha \wedge \beta, \Gamma\}$, check $\{\alpha, \beta, \Gamma\}$.

3. To check $\{\sigma X.\alpha(X), \Gamma\}$, check $\{\alpha(\sigma X.\alpha(X)), \Gamma\}$. This step is called *regeneration.*

4. If only literals and formulas of the form $(a \rightarrow \Phi)$ appear in Γ (i.e. the above rules do not apply) then we must check that all the literals are satisfied in s and:

   (a) for every $(a \rightarrow \Phi) \in \Gamma$ and $\alpha \in \Phi$ we must find a state t with $(s,t) \in R(a)$ and check whether $\{\alpha\} \cup \{\bigvee\Theta : (a \rightarrow \Theta) \in \Gamma, \Theta \neq \Phi\}$ is satisfied in t,

   (b) for every action a such that a formula of the form $(a \rightarrow \Phi)$ belongs to Γ and every t such that $(s,t) \in R(a)$ we must find formulas $(a \rightarrow \Phi') \in \Gamma$ and $\alpha' \in \Phi'$ and check that in t the set $\{\alpha'\} \cup \{\bigvee\Theta : (a \rightarrow \Theta) \in \Gamma, \Theta \neq \Phi'\}$ is satisfied.

Observe that the procedure described above is nondeterministic and this nondeterminism shows in two places: reduction of disjunction and assignment of states to sets in reduction of modalities. We will call one execution of the procedure a *run.* Thus for a given formula and a state there may be many runs, some of them finite and some not. We want to find a condition on runs such that existence of a successful run would characterise satisfiability relation.

Let us give some examples. Consider a formula $\mu X.q \wedge (a \rightarrow \{p \vee X\})$. Let $\mathcal{M}_1$ be a model presented in Figure 3.

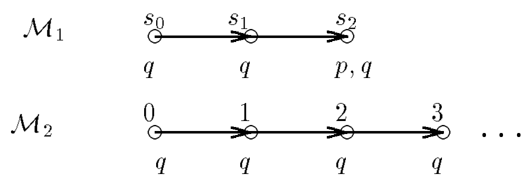

Figure 3: Models $\mathcal{M}_1$ and $\mathcal{M}_2$

It consists of three states s_0, s_1, s_2 and two a-transitions: $s_0 \xrightarrow{a} s_1$ and $s_1 \xrightarrow{a} s_2$. Let q be true in all the states and let p be true in s_2. A checking of satisfiability of our formula in this model can be pictured as follows:

$$
\begin{array}{l}
s_0 \models \{\mu X.q \wedge (a \rightarrow \{p \vee X\})\} \\
s_0 \models \{q \wedge (a \rightarrow \{p \vee X\})\} \\
s_0 \models \{q, (a \rightarrow \{p \vee X\})\} \\
s_1 \models \{p \vee X\} \\
s_1 \models \{X\} \\
s_1 \models \{q \wedge (a \rightarrow \{p \vee X\})\} \\
s_1 \models \{q, (a \rightarrow \{p \vee X\})\} \\
s_2 \models \{p \vee X\} \\
s_2 \models \{p\}
\end{array}
$$

We used variable X to denote the whole fixpoint formula $\mu X.q \wedge (a \rightarrow \{p \vee X\})$.

Remark 3.1 This example also shows that in clause 4b the condition that only special formulas appear in the label of Γ is necessary. Otherwise we could apply clause 4b at the very beginning and finish the procedure claiming that it is a success.

Now consider a model $\mathcal{M}_2$ which states are natural numbers, transitions labeled by a lead from a number to its successor, q is true in all the states and p is always false. In this model our satisfiability checking would look like this:

$$
\begin{array}{lll}
0 & \models & \{\mu X.q \wedge (a \rightarrow \{p \vee X\})\} \\
0 & \models & \{q \wedge (a \rightarrow \{p \vee X\})\} \\
0 & \models & \{q, (a \rightarrow \{p \vee X\})\} \\
1 & \models & \{p \vee X\} \\
1 & \models & \{X\} \\
1 & \models & \{q \wedge (a \rightarrow \{p \vee X\})\} \\
1 & \models & \{q, (a \rightarrow \{p \vee X\})\} \\
\vdots & \vdots & \qquad\vdots
\end{array}
$$

This is the only run of the procedure which does not fail in finitely many steps. It is easy to show that $\| \mu X.q \wedge (a \rightarrow \{p \vee X\}) \|_{Val}^{\mathcal{M}_2} = \emptyset$. Hence we are tempted to conclude that if our satisfiability checking process can't finish in a finite number of steps then the formula is not satisfied. This statement is not true. As an example let us take the formula $\nu X.q \wedge (a \rightarrow \{p \vee X\})$. Now the satisfiability checking process will look almost the same as before but it is easy to see that $\| \nu X.q \wedge (a \rightarrow \{p \vee X\}) \|_{Val}^{\mathcal{M}_2} = \mathcal{N}$.

Let us analyse the differences between this two checking examples. We will call a transition from X to $\alpha(X)$ *regeneration* of variable X. In first case some μ-variable was regenerated infinitely often during the checking. In the second case it was ν-variable. We will show in the next subsection that satisfiability checking process is successful iff (roughly speaking) there is no instance of a μ-variable regenerated finitely many times.

The intuition why it is the case comes from the Knaster-Tarski characterisation of fixpoints on a complete lattice by chains of approximations. In order to describe this approximations for every ordinal τ we introduce two new constructs $\mu^\tau X.\alpha(X)$ and $\nu^\tau X.\alpha(X)$ with the following semantics (for some $\mathcal{M} = \langle S, R, \rho\rangle$ and Val):

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

— $\| \sigma^{\tau+1} X.\alpha(X) \|_{Val} = \| \alpha(X) \|_{Val[\|\sigma^\tau X.\alpha(X)\|_{Val}/X]}$ (σ means 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.

Remark 3.2 This constructs are not μ-calculus formulas. This is an extension of the language which we will use only to state the following theorem and to define the notion of signature in Definition 3.14.

Using this constructs we can state:

Theorem 3.3 (Knaster-Tarski) *For every model $\mathcal{M}$ and valuation Val:*

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

From the theorem it follows that whenever we are to check $\mu X.\alpha(X)$ in some state s and $s \models \mu X.\alpha(X)$ then there is the least ordinal τ such that $s \models \mu^\tau X.\alpha(X)$. So $s \models \alpha(\mu^{\tau'} X.\alpha(X))$ for some ordinal $\tau' < \tau$. Hence each regeneration reduces the ordinal μ-formula carries. Because the ordinals are well ordered if we guess the run right then each instance of a μ-formula should be regenerated only finitely many times.

To see what we mean by an instance of a μ-formula consider the structure $\mathcal{M}_3$ presented in Figure 4

$$\mathcal{M}_3 \qquad \bullet 0 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{a} 5 \quad \ldots$$

Figure 4: Model $\mathcal{M}_3$

Consider a task of checking that the formula $\{\mu X.\nu Y.(a \to \{X\}) \vee (b \to \{Y\})\}$ is satisfied in state 0:

$$\begin{array}{lcl} 0 & \models & \{\nu X.\mu Y.(a \to \{X\}) \vee (b \to \{Y\})\} \\ 0 & \models & \{(a \to \{X\}) \vee (b \to \{Y\})\} \\ 0 & \models & \{(a \to \{X\})\} \\ 1 & \models & X \\ 1 & \models & \{(a \to \{X\}) \vee (b \to \{Y\})\} \\ 1 & \models & \{(b \to \{Y\})\} \\ 2 & \models & Y \\ 2 & \models & \{(a \to \{X\}) \vee (b \to \{Y\})\} \\ 2 & \models & \{(a \to \{X\})\} \\ 3 & \models & X \\ \vdots & \vdots & \vdots \end{array}$$

This is unique run which does not fail after finite number of steps. As there are infinitely many regenerations of the μ-variable on this unique run we are inclined to say that the formula is not satisfied in state 0. On the other hand it is not difficult to check that $\| \nu X.\mu Y.(a \to \{X\}) \vee (b \to \{Y\}) \|_{Val}^{\mathcal{M}_3} = \mathcal{N}$. The reason for this is that after each regeneration of μ-variable X there is a regeneration of ν-variable Y on which X depends ($Y \leq X$). Very vaguely one may say that each time the ν-variable is regenerated new instance of the μ-subformula is created.

Exercises:

1 Consider model $\mathcal{M}_3$ and the formula $\| \mu X.\nu Y.(a \to \{X\}) \vee (b \to \{Y\}) \|_{Val}^{\mathcal{M}_3}$. Calculate the set of states where the formula holds. How the runs of satisfiability checking procedure look like?

3.2 Formalisation

Let us formalise the above intuitions. We will first define a notion of a tableau for a formula. This is intended to represent all possible reductions of the formula which can be done during satisfiability checking. Then we will define a notion of a marking which will correspond to one run of the checking algorithm. We conclude this section by proving a characterisation of satisfiability of a formula in a given state by means of consistent markings.

Let us start with tableaux. A tableau for a formula will be constructed according to reduction rules which will reflect the steps of the checking procedure described above. This rules may be also considered as sound logical rules although some of them may seem strange from the logical point of view.

Each rule will have the form:

$$\frac{\Sigma_1, \ldots, \Sigma_n}{\Gamma}$$

where $\Gamma, \Sigma_1, \ldots, \Sigma_n$ are finite sets of formulas. The set below the line is called *conclusion*. The sets above the line are called *assumptions*. We will see an application of a rule as a process of reduction. Given a set of formulas Γ we want to derive, we apply the rule and obtain assumptions $\Sigma_1, \ldots, \Sigma_n$ which we can reduce further. We continue to write $\{\alpha, \Gamma\}$ as a shorthand for $\{a\} \cup \Gamma$. According to our proviso we assume that all the formulas considered in this section are well named, positive, guarded and use $(a \to \Gamma)$ construct instead of $\langle a \rangle$ and $[a]$ modalities.

Definition 3.4 For a formula γ and its binding function $\mathcal{D}_\gamma$ we define the system of tableau rules $\mathcal{S}^\gamma$ parameterised by γ or rather its binding function:

$$(and)\ \frac{\{\alpha,\beta,\Gamma\}}{\{\alpha\wedge\beta,\Gamma\}} \qquad (or)\ \frac{\{\alpha,\Gamma\}\quad\{\beta,\Gamma\}}{\{\alpha\vee\beta,\Gamma\}}$$

$$(\mu)\ \frac{\{\alpha(X),\Gamma\}}{\{\mu X.\alpha(X),\Gamma\}} \qquad (\nu)\ \frac{\{\alpha(X),\Gamma\}}{\{\nu X.\alpha(X),\Gamma\}}$$

$$(reg)\ \frac{\{\alpha(X),\Gamma\}}{\{X,\Gamma\}} \quad \text{whenever } X \text{ is a bound variable of } \gamma \text{ and } \mathcal{D}_\gamma(X)=\sigma X.\alpha(X)$$

$$(mod)\ \frac{\{\alpha\}\cup\{\bigvee\Theta : (a\to\Theta)\in\{\Gamma\},\Theta\neq\Phi\}\ \text{ for every } (a\to\Phi)\in\{\Gamma\},\alpha\in\Phi}{\{\Gamma\}}$$

with $\bigvee\emptyset$ interpreted as $\bot$.

Remark 3.5 The rule (mod) has as many premises as there are formulas in sets Φ, s.t., $(a\to\Phi)\in\Gamma$. For example

$$\frac{\{\alpha_1,\alpha_3\}\quad\{\alpha_2,\alpha_3\}\quad\{\alpha_1\vee\alpha_2,\alpha_3\}\quad\{\beta_1\}\quad\{\beta_2\}}{\{(a\to\{\alpha_1,\alpha_2\}),(a\to\{\alpha_3\}),(b\to\{\beta_1,\beta_2\})\}}$$

is an instance of the rule.

Definition 3.6 A *tableau* for γ is a pair $\langle T,L\rangle$, where T is a tree and L is a labeling function such that:

1. the root of T is labeled by $\{\gamma\}$,
2. the sons of every node are created and labeled according to the rules of the system $\mathcal{S}^\gamma$, with *rule* (mod) *applied only when no other rule is applicable.*

Leaves and nodes where (mod) rule was applied will be called *modal nodes*. The root of $\mathcal{T}$ and sons of modal nodes will be called *choice nodes*. We say that m is *near* n iff there is a path from n to m in a tableau without an application of (mod) rule in between.

Remark 3.7 Returning to our example of an instance of the rule (mod) from Remark 3.5. If a node n is labeled by the conclusion of this instance then n has five sons labeled by corresponding assumptions. We will call a son obtained by *reducing* an action a an *a-son*. In a sense one can consider that some edges of a tableau can be labeled with actions. In our example n has three a-sons and two b-sons. Node n is a modal node, its sons are choice nodes.

An example of a tableau is presented in Figure 5. Following computer science tradition our tableaux will always expand downwards so the root is at the top.

Definition 3.8 (Marking) *For a tableau $\mathcal{T}=\langle T,L\rangle$ we define its* marking *with respect to a structure $\mathcal{M}=\langle S,R,\rho\rangle$ and state s to be an assignment M of sets of states of $\mathcal{M}$ to the nodes of $\mathcal{T}$ such that:*

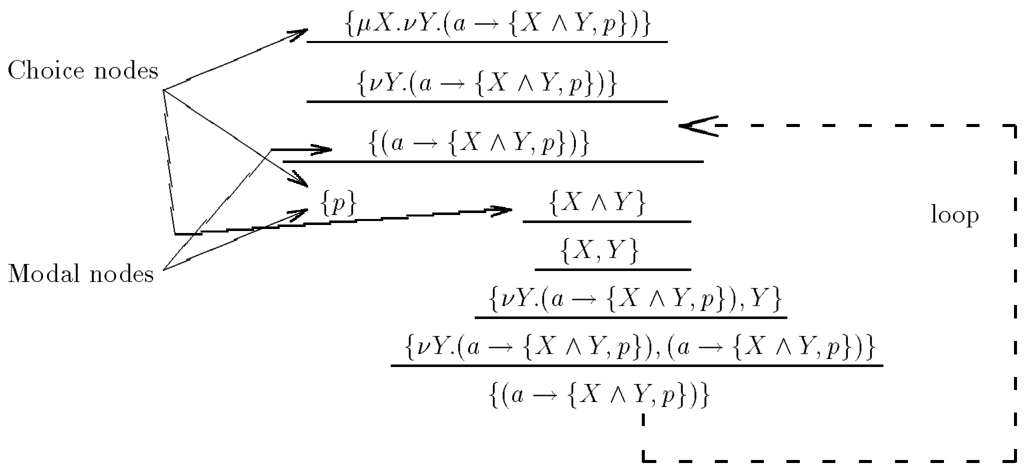

Figure 5: An example of a tableau.

1. *The root of $\mathcal{T}$ is marked with $\{s\}$.*

2. *If in a node n some rule other than (or) or (mod) was applied then the only son of n has the same marking as n.*

3. *If a node n has assigned some set of states $M(n)$ and rule (or) was applied in n, then one son is assigned some subset of $M(n)$ and the other son its complement.*

4. *If rule (mod) was applied in a node m then for every $s \in M(m)$ and every action a for which there exits a formula $(a \to \Gamma)$ in $L(m)$:*

   (a) *for every a-son n of m and every $s \in M(m)$ there exists a state $t \in M(n)$, s.t. $(s,t) \in R(a)$,*

   (b) *for every state $s \in M(m)$ and every state t such that $(s,t) \in R(a)$ there is an a-son n of m with $t \in M(n)$.*

Definition 3.9 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:

- If $\mathcal{T}r(n)$ is defined then $\mathcal{T}r(n) \in L(n)$.

- Let m be a node with $\mathcal{T}r(m)$ defined and let $n \in \mathcal{P}$ be a son of m. If a rule applied in m does not reduce formula $\mathcal{T}r(m)$ then $\mathcal{T}r(n) = \mathcal{T}r(m)$. If $\mathcal{T}r(m)$ is reduced in m then $\mathcal{T}r(n)$ is one of the results of the reduction. This should be clear for all the rules except (mod). In case m is a modal node and n is labeled by $\{\psi\} \cup \{\bigvee\Theta : (a \to \Theta) \in \Gamma, \Theta \neq \Psi\}$ for some $(a \to \Psi) \in L(m)$ and $\psi \in \Psi$, then $\mathcal{T}r(n) = \psi$ if $\mathcal{T}r(m) = (a \to \Psi)$ and $\mathcal{T}r(n) = \bigvee\Theta$ if $\mathcal{T}r(m) = (a \to \Theta)$ for some $(a \to \Theta) \in \Gamma$, $\Theta \neq \Psi$. Traces from other formulas end in node m.

Definition 3.10

— We say that there is a *regeneration* of a variable X on a trace $\mathcal{T}r$ on some path iff for some node m and its son n on the path $\mathcal{T}r(m) = X$ and $\mathcal{T}r(n) = \alpha(X)$, where $\mathcal{D}(X) = \sigma X.\alpha(X)$.

— We call a trace *μ-trace* iff it is an infinite trace (defined for the whole path) on which the smallest with respect to $\leq_\gamma$ ordering variable regenerated infinitely often is a μ-variable. Similarly a trace will be called a *ν-trace* iff it is an infinite trace where the smallest variable which regenerates i.o. is a ν-variable.

On our example in Figure 5 there is a finite trace to the leaf labeled $\{p\}$. There is a trace consisting of the leftmost formulas on the infinite path of the tableau. It is a μ-trace. There is also a ν-trace obtained by choosing rightmost formulas. Observe that the traces can split and merge. There are countably many ν-traces and uncountably many μ-traces on the infinite path of the tableau.

Remark 3.11 Every infinite trace is either a μ-trace or a ν-trace because all the rules except regenerations decrease the size of formulas and formulas are guarded hence every formula is eventually reduced. Observe that even though $\leq_\gamma$ is a partial ordering there is always the least variable required in the above definition.

We can now define what does it mean for a marking to be consistent.

Definition 3.12 (Consistent marking) *Using the notation from the Definition 3.8, a marking M of $\mathcal{T}$ with respect to $\mathcal{M}, s$ is called* consistent marking *with respect to $\mathcal{M}$, s, Val iff it satisfies the following conditions:*

local consistency *for every modal node m of $\mathcal{T}$ and state t of $\mathcal{M}$, if $t \in M(m)$ then $\mathcal{M}, t$, Val $\models \Gamma'$, where Γ' is the set of all the literals occurring in $L(m)$,*

global consistency *for every path $\mathcal{P} = n_0, n_1, \ldots$ of $\mathcal{T}$ such that $M(n_i) \neq \emptyset$ for every $i = 0, 1, \ldots$ there should be no μ-trace on $\mathcal{P}$.*

Theorem 3.13 Positive guarded formula φ is satisfied in a *finitary branching* structure $\mathcal{M}$, state s and valuation *Val* iff there is a tableau $\mathcal{T}$ for φ and a marking M of $\mathcal{T}$ consistent with $\mathcal{M}, s$, *Val*.

Proof

Let us fix a formula φ (which is well named, positive and guarded by our proviso). Let $\mathcal{D}_\varphi$ be its binding function.

First we introduce the notions of a signature and ν-signature similar to that considered by Streett and Emerson [18]. We extend the notion of binding function from Section 2.3, by allowing constructs of the form $\sigma^\tau X.\alpha(X)$ in the image of $\mathcal{D}_\varphi$ (as before σ stands for μ or ν). The concept of expansion $\langle\!|\alpha|\!\rangle_{\mathcal{D}_\varphi}$ extends immediately.

Definition 3.14 Let us assume that $X_1, X_2, \ldots, X_{d^\mu}$ $(Y_1, \ldots, Y_{d^\nu})$ are all μ-constants (ν-constants respectively) from the domain of $\mathcal{D}_\varphi$ listed in some order respecting $\leq_\varphi$ relation (smaller elements have smaller indices). Let us take a subformula β of φ.

If the formula $\langle\!\langle\beta\rangle\!\rangle_{\mathcal{D}_\varphi}$ is satisfied in a state s of a model $\mathcal{M}$ with a valuation Val then we can define a *signature* of β in s, $Sig(\beta, s)$, as the least, in lexicographical ordering, sequence of ordinals $(\tau_1, \ldots, \tau_{d^\mu})$ such that $\mathcal{M}, s, Val \models \langle\!\langle\beta\rangle\!\rangle_{\mathcal{D}'}$, where $\mathcal{D}'$ is a binding function obtained from $\mathcal{D}_\varphi$ by changing definitions of X_i (for $i = 1, \ldots, d^\mu$) from $\mathcal{D}_\varphi(X_i) = \mu X_i.\alpha_i(X_i)$ to $\mathcal{D}'(X_i) = \mu^{\tau_i} X_i.\alpha_i(X_i)$.

If the formula $\langle\!\langle\beta\rangle\!\rangle_{\mathcal{D}_\varphi}$ is *not* satisfied in a state s of a model $\mathcal{M}$ with a valuation Val then we can define a ν*-signature* of β in s, ${}^\nu Sig(\beta, s)$, as the least, in lexicographical ordering, sequence of ordinals $(\tau_1, \ldots, \tau_{d^\mu})$ such that $\mathcal{M}, s, Val \not\models \langle\!\langle\beta\rangle\!\rangle_{\mathcal{D}'}$, where $\mathcal{D}'$ is a binding function obtained from $\mathcal{D}_\varphi$ by changing definitions of Y_i (for $i = 1, \ldots, d^\nu$) from $\mathcal{D}_\varphi(Y_i) = \nu Y_i.\alpha_i(Y_i)$ to $\mathcal{D}'(Y_i) = \nu^{\tau_i} Y_i.\alpha_i(Y_i)$.

Remark 3.15 Of course signature of a formula depends not only on a state but also on valuation and definition list which is not taken into the account in our notation. This parameters will be always clear from the context.

It can be shown that signatures behave nicely with respect to a formula reduction namely:

Lemma 3.16 For every state s of a model $\mathcal{M}$, valuation Val, and α, β, $\mu X.\alpha(X)$, $\nu X.\alpha(X)$ subformulas of φ:

— If $\mathcal{M}, s, Val \models \langle\!\langle\alpha \wedge \beta\rangle\!\rangle_{\mathcal{D}_\varphi}$ then $Sig(\alpha \wedge \beta, s) = \max(Sig(\alpha, s), Sig(\beta, s))$.

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

— If $\mathcal{M}, s \models \langle\!\langle(a \to \Phi)\rangle\!\rangle_{\mathcal{D}_\varphi}$ then: (i) for every formula $\varphi \in \Phi$ there is a state t such that $(s,t) \in R(a)$ and $Sig(\varphi, t) \leq Sig((a \to \Phi), s)$, (ii) for any state t such that $(s,t) \in R(a)$, $Sig(\bigvee \Phi, t) \leq Sig((a \to \Phi), s)$.

— If $\mathcal{M}, s \models \langle\!\langle\nu X.\alpha(X)\rangle\!\rangle_{\mathcal{D}_\varphi}$ then $Sig(\nu X.\alpha(X), s) = Sig(\alpha(X), s)$.

— If $\mathcal{M}, s \models \langle\!\langle\mu X_i.\alpha_i(X_i)\rangle\!\rangle_{\mathcal{D}_\varphi}$ and X_i is i-th μ-variable in the domain of $\mathcal{D}_\varphi$ then the prefixes of length $i-1$ of $Sig(\mu X_i.\alpha_i(X_i), s)$ and $Sig(\alpha_i(X_i), s)$ are equal.

— If $\mathcal{M}, s \models \langle\!\langle Z\rangle\!\rangle_{\mathcal{D}_\varphi}$ and $\mathcal{D}_\varphi(Z) = \sigma Z.\alpha(Z)$ then $Sig(Z, s) = Sig(\alpha(Z), s)$ if Z is a ν-variable. If Z is i-th μ-variable then the second signature is smaller and the difference is at position i.

Similarly for ν-signatures but with interchanged roles of μ with ν, conjunction with disjunction and dual statement in $(a \to \Phi)$ case.

Proof

We will consider only the last case. Suppose $\mathcal{M}, s \models \langle\!\langle X_i\rangle\!\rangle_{\mathcal{D}_\varphi}$, where X_i is i-th μ-variable from the domain of $\mathcal{D}_\varphi$. Let $\mathcal{D}_\varphi(X_i) = \mu X_i.\alpha_i(X_i)$. As our ordering of μ-variables respects $\leq_\varphi$ relation only μ-variables with indices less than i can appear free in $\mu X_i.\alpha_i(X_i)$. Let $Sig(X_i, s) = (\tau_1, \ldots, \tau_n)$ and let $\mathcal{D}'$ be a definition list obtained from $\mathcal{D}_\varphi$ in the way described in Definition 3.14. Let us denote $\alpha_i(X_i)[\mathcal{D}'(X_{i-1})/X_{i-1}]\ldots[\mathcal{D}'(X_1)/X_1]$ by $\beta(X_i)$.

It should be clear that the signature of $\alpha_i(\mu X_i.\alpha_i(X_i))$ is the same as $\mu X_i.\alpha_i(X_i)$, which means that the signatures of X_i and $\alpha_i(X_i)$ are the same on positions smaller than i. From the definition of the signature we have $\mathcal{M}, s \models \mu^{\tau_i} X_i.\beta(X)$. 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. □

Proof of Thm 3.13 $\Rightarrow$ The proof is based on [18]. Let us first focus on the left to right implication. Suppose that φ is satisfied in a state s of a structure $\mathcal{M}$ with a valuation *Val*. Let $\mathcal{T}$ be a tableau for φ. We will construct a consistent marking M of $\mathcal{T}$ with respect to $\mathcal{M}, s$, *Val*.

- The root of $\mathcal{T}$ will be marked by s.
- If in a node some unary rule was applied then, as required, the only son of it will be marked with the same set of states as the father.
- If (or) rule was applied in a node n:

$$\frac{\{\alpha, \Gamma\} \quad \{\beta, \Gamma\}}{\{\alpha \vee \beta, \Gamma\}}$$

  then for any $s \in M(n)$ we put s into the marking of the son labeled $\{\alpha, \Gamma\}$ if $Sig(\alpha, s) < Sig(\beta, s)$ otherwise we put s into the marking of the other son.

- Suppose rule (mod) was applied in n and let $s \in M(n)$. If for some $(a \to \Phi) \in L(n)$, $\varphi \in \Phi$ and t with $(s, t) \in R(a)$ we have $Sig(\varphi, t) \leq Sig((a \to \Phi), s)$ then we put t into a marking of an a-son of n containing φ. Form Lemma 3.16 follows that every $t \in \{t : (s, t) \in R(a)\}$ appears in a marking of some a-son of n and in every a-son of n one of the states from the set appears.

It is obvious that the marking defined in such a way is locally consistent, i.e., for any modal node m and state $t \in M(m)$, all the literals occurring in the label of m are satisfied in t.

To check the other condition of consistency of the marking let $\mathcal{P}$ be a path of $\mathcal{T}$ and let every node of $\mathcal{P}$ be labeled by a nonempty set. We will show that there cannot be a μ-trace on $\mathcal{P}$. First observation is that because the structure is finitely branching, every node of $\mathcal{T}$ is marked by a finite number of states. This means that if for some node n and formula $\alpha \in L(n)$ we define its signature as $\max\{Sig(\alpha, s) : s \in M(n)\}$ then there will be actually state $s' \in M(n)$, s.t., $Sig(\alpha, s')$ is this maximum.

Let us assume that there is a μ-trace on $\mathcal{P}$ and that a μ-variable X_i is the smallest variable with respect to $\leq_\varphi$ ordering regenerated i.o. on this trace. Let us look at the signatures of formulas from the point when no variable smaller than X_i is regenerated on the trace. From the definition of the marking follows that from that moment signatures of formulas on the trace never increases on positions $1, \ldots, i-1$. If X_i is regenerated in a node n then $X_i \in L(n)$ and $\alpha_i(X_i)$ occurs in the label of the only son n' of n. From Lemma 3.16 follows that for any state $s \in M(n)$, signature $Sig(\alpha_i(X_i), s)$ is smaller, on position i than $Sig(X_i, s)$. Because $M(n)$ is finite, maximal signature also decreases.

This shows that from some moment maximal signature of formulas on the trace considered up to position i never increases and decreases every time X_i is regenerated. This is a contradiction because sequences of ordinals of bounded length are well ordered.

Proof of Thm 3.13 $\Leftarrow$ To prove the theorem in the direction from right to left let us assume that there is a tableau $\mathcal{T}$ for φ and its marking M consistent with respect to $\mathcal{M}, s, Val$. Assume conversely that $\mathcal{M}, s, Val \not\models \varphi$. We will show that this assumption leads to a contradiction.

We will show that there must be a μ-trace on a path of $\mathcal{T}$ such that every node of it is marked by some nonempty set of states. Suppose that we have constructed this hypothetical trace up to a node n, formula $\alpha \in L(n)$ is the last formula of it and $s \in M(n)$ is a state such that $\mathcal{M}, s, Val \not\models \alpha$. We proceed according to the rule which was applied in n.

- Suppose the rule is unary. If it was applied to α then the next element of the trace is the result of reduction of α, otherwise the next element is formula α itself. In case (and) rule was applied to $\alpha = \gamma_1 \wedge \gamma_2$ choose γ_1 if ${}^{\nu}Sig(\gamma_1, s)$ is smaller than ${}^{\nu}Sig(\gamma_2, s)$ or choose γ_2 otherwise. It is clear that the new last element of the trace is not satisfied in s.

- If rule (or) was applied in n then choose a son n' of n, s.t. $s \in M(n')$. The next element of the trace will be a result of reduction of α which appears in n' or α itself if it was not reduced by this application of the rule.

- If rule (mod) was applied in n then by definition of a consistent marking α cannot be a literal or a formula of the form $(a \to \emptyset)$. Hence it has the form $(a \to \Phi)$ with $\Phi \neq \emptyset$. In this case either:

  1. There is a formula $\varphi \in \Phi$ such that for every t with $(s,t) \in R(a)$ we have $t \not\models \varphi$ and ${}^{\nu}Sig(\varphi, t) \leq {}^{\nu}Sig((a \to \Phi), s)$. In this case we choose a son n' of n labeled by $\{\varphi\} \cup \{\bigvee \Theta : (a \to \Theta) \in L(n), \Theta \neq \Phi\}$. For the next state we take $t \in M(n')$ such that $(s,t) \in R(a)$.
  2. There is state t, s.t., $(s,t) \in R(a)$ and $t \not\models \bigvee \Phi$ with ${}^{\nu}Sig(\bigvee \Phi, t) \leq {}^{\nu}Sig((a \to \Phi), s)$. In this case take a son n' of n where t is in the marking. Our next formula is $\bigvee \Phi$ or $\psi \in \Phi$ depending on which one appears in $L(n)$.

Using similar arguments as in the proof of the left to right implication one can easily prove that constructed trace must be a μ-trace but this contradicts our assumption about consistency of the marking.

□

Remark 3.17 Observe that Theorem 3.13 is stated only for finitely branching structures and we need this restriction to prove the left to right implication. This restriction can be removed by slightly modifying the notion of the marking. We will not do it here as we do not need this stronger result. The fact we need is that every satisfiable formula has a finitary branching model. We will assume this fact without a proof.

4 Applications of operational semantics

This section is divided into two parts. In first we sketch how some known results can be proved using the operational semantics of formulas. The second is devoted to the notion of tableau equivalence which turns out to be very helpful in the further development.

4.1 Small model theorem, decidability, syntactic characterisations

Our operational semantics is very similar to a model checking algorithm in the style of Stirling and Walker [16] (in case of finite models) or [17] (in the general case). The important difference lies in the treatment of conjunction. This is connected to the notion of alternation which we try to omit in this notes. Let us instead consider decidability problem and the small model theorem [18].

Theorem 4.1 (Small model) *If a μ-calculus formula γ is satisfiable then it is satisfiable in a finite model having not more than $O(2^{|\gamma|})$ states.*

Proof
The proof is based on [18]. Let $\mathcal{T} = \langle T, L\rangle$ be a tableau for γ. Let a *pre-model* $\mathcal{PM}$ be a tree satisfying the following conditions:

- the root of of $\mathcal{T}$ belongs to $\mathcal{PM}$,
- if a choice node belongs to $\mathcal{PM}$ then exactly one modal node near it belongs to $\mathcal{PM}$,
- if a modal node belongs to $\mathcal{PM}$ then all its sons belong to $\mathcal{PM}$,
- if a modal node m belongs to $\mathcal{PM}$ then $\perp$ does not appear in m and there is no literal such that the literal and its negation occur in the label of m,
- there is no path of $\mathcal{PM}$ with a μ-trace.

It is easy to construct a model $\mathcal{M} = \langle S, R, \rho\rangle$ from the pre-model. Let S be the set of modal nodes of $\mathcal{PM}$, $(m, n) \in R(a)$ iff n is near some a-son of m, $m \in \rho(p)$ iff $p \in L(m)$. Let s_0 be the modal node of $\mathcal{PM}$ near the root of $\mathcal{T}$. Let $Val(X)$ be defined by $m \in Val(X)$ iff $X \in L(m)$.

It is quite easy to see that there is a consistent marking of $\mathcal{T}$ with respect to $\mathcal{M}, s_0, Val$. Hence by Theorem 3.13 we have $\mathcal{M}, s_0, Val \models \gamma$. Form the same theorem it follows that if γ is satisfiable then there is a pre-model for γ (see Remark 3.17). Thus we have:

Theorem 4.2 Formula γ is satisfiable iff there is a pre-model for γ.

Now there is a finite automaton on trees which recognises (codings of) pre-models for γ. Using Rabin conditions it is possible to construct such an automaton with $O(2^{|\gamma|})$ states and $|\gamma|$ pairs. Hossley and Rackoff proved in [7] that if Rabin automaton over one letter alphabet has an accepting run then it has a regular accepting run. From this follows the small model theorem without the bound on the size of the structure. Emerson using this result proved the following theorem [3]:

Theorem 4.3 (Emerson) *Suppose $\mathcal{A}$ is a Rabin automaton over a single letter alphabet. If $\mathcal{A}$ has an accepting run then there is a graph $\mathcal{G}$ with states of $\mathcal{A}$ as nodes which unwinds to an accepting run of $\mathcal{A}$.*

From the theorem it follows that there is a pre-model for γ which can be presented as a graph with no more than $O(2^{|\gamma|})$ nodes. It can be converted to a model of the same size.

□

In [4] it was shown how to test emptiness of a Rabin automaton in time $\mathcal{O}((mn)^{3n})$ where m is the number of states and n is the number of pairs of the automaton. This gives an exponential satisfiability testing procedure for the μ-calculus. The lower bound for this problem follows from the lower bound for PDL proved in [6].

Theorem 4.4 (Decidability) *The problem of deciding whether a given formula of the μ-calculus is satisfiable is EXPTIME complete in the size of the formula.*

In [14] the question was raised whether it is possible to characterise validity in the same way as satisfiability.

Definition 4.5 (Refutation) *A refutation $\mathcal{R}$ is a subset of $\mathcal{T}$ satisfying the following conditions:*

- *the root of $\mathcal{T}$ belongs to $\mathcal{R}$,*
- *if a choice node n belongs to $\mathcal{PM}$ then all modal nodes near n belong to $\mathcal{R}$,*
- *if a modal node m belongs to $\mathcal{R}$ then at most one son of m belongs to $\mathcal{R}$,*
- *if m has no sons in $\mathcal{R}$ then either $\perp$ occurs in the label of m or some literal and its negation occur in m,*
- *there is a μ-trace on every infinite path of $\mathcal{R}$.*

It was observed that from Martin's determinacy theorem [10] it follows that if there is no pre-model in $\mathcal{T}$ then one can find a *refutation* in $\mathcal{T}$. It is easy to see that there cannot be a refutation and a pre-model in the same tableau. This shows

Theorem 4.6 Formula γ is valid iff there is a refutation for $\neg\gamma$.

The similar analysis of complexity, as we did for pre-models, applies also to refutations. Summarising we have:

Theorem 4.7 There is an algorithm which, given a mu-calculus sentence γ, constructs a model of the size $O(2^{|\gamma|})$ or a refutation of the size $O(2^{|\gamma|})$. The algorithm runs in time $O(2^{|\gamma|})$.

In [21] a finitary proof system was proposed and it was show how to convert a refutation of a formula into a proof of the negation of the formula. Unfortunately the method used there does not seam to work for Kozen's axiomatisation.

4.2 Tableau equivalence

In this subsection we would like to present one more application of the operational semantics. We will define the notion of tableaux equivalence and show that if two formulas have equivalent tableaux then they are equivalent.

To give the definition of equivalence we will need to distinguish one more kind of formulas:

Definition 4.8 A *terminal formula* is a formula of the form $(a \to \emptyset)$ for some action a.

The meaning of a terminal formula $(a \to \emptyset)$ is that there are no a-transitions from a given state. Although we introduce them only here these formulas were already implicitly considered in the definition of the marking (see Definition 3.8).

In tableau equivalence we can abstract from the order of application of non-modal rules, but the structure of the tree designated by modal nodes will be very important.

Definition 4.9 We say that two tableaux $\mathcal{T}_1$ and $\mathcal{T}_2$ are *equivalent* iff there is a bijection $\mathcal{E}$ between choice and modal nodes (see Definition 3.6) of $\mathcal{T}_1$ and $\mathcal{T}_2$ such that:

1. $\mathcal{E}$ maps the root of $\mathcal{T}_1$ onto the root of $\mathcal{T}_2$, it maps choice nodes to choice nodes and modal nodes to modal nodes.

2. If n is a descendant of m then $\mathcal{E}(n)$ is a descendant of $\mathcal{E}(m)$. Moreover if for some action a, node n is an a-son of a modal node m then $\mathcal{E}(n)$ is an a-son of $\mathcal{E}(m)$.

3. For every modal node m, the sets of literals and terminal formulas occurring in $L(m)$ and in $L(\mathcal{E}(m))$ are equal.

4. There is a μ-trace on a path $\mathcal{P}$ of $\mathcal{T}_1$ iff there is a μ-trace on a path of $\mathcal{T}_2$ designated by the image of $\mathcal{P}$ under $\mathcal{E}$.

Theorem 4.10 If two positive guarded formulas have equivalent tableaux then they are equivalent. (According to our proviso we assume that formulas use $(a \to \Phi)$ notation instead of $\langle a\rangle\varphi$ and $[a]\varphi$.)

Proof
Let α, β be two formulas and $\mathcal{T}_\alpha, \mathcal{T}_\beta$ equivalent tableaux for α and β respectively. Let $\mathcal{E} : \mathcal{T}_\alpha \to \mathcal{T}_\beta$ denote the bijection showing the equivalence of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$. We will show that for any finitely branching structure $\mathcal{M}$, state s and valuation *Val*, we have $\mathcal{M}, s, Val \models \alpha$ iff $\mathcal{M}, s, Val \models \beta$.

Suppose $\mathcal{M}, s, Val \models \alpha$. By Theorem 3.13 there is a consistent marking M of $\mathcal{T}_\alpha$ with respect to $\mathcal{M}, s, Val$. This marking determines a consistent marking of $\mathcal{T}_\beta$. First to any modal or choice node n of $\mathcal{T}_\beta$ we assign the set $M(\mathcal{E}^{-1}(n))$. The labeling of any other internal node is uniquely determined by this assignment. Directly from the definition of equivalence it follows that defined marking is consistent with $\mathcal{M}, s, Val$.

Observe that $\mathcal{E}^{-1}$ is also a function showing the equivalence of $\mathcal{T}_\beta$ and $\mathcal{T}_\alpha$ hence there is a way of obtaining a consistent marking of $\mathcal{T}_\alpha$ from a consistent marking of $\mathcal{T}_\beta$.

□

Remark 4.11 Of course the inverse of the above theorem is not true. Despite this weakness the theorem will be very useful for proving equivalence of formulas.

Let us discuss the connection between this notion of equivalence and bisimulation. It is possible to define the notion of simulation between tableaux. It is convenient to use game metaphor here but we hope that the connections with the usual notion of simulation will be evident. Given two tableaux $\mathcal{T}_\alpha$, $\mathcal{T}_\beta$ we can define the game on this two tableaux by the following rules:

1. The starting position of the game is (r_α, r_β), where r_α and r_β are the roots of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ respectively.

2. If the position of the game is a pair of choice nodes (n_α, n_β) then player *I* chooses some modal node m_α near n_α and player *II* responds with a modal node m_β near n_β with the property that every literal and terminal formula appearing in m_β appears also in m_α. The new position of the game is (m_α, m_β).

3. If the game is in a position (m_α, m_β), both nodes being modal nodes then player *I* chooses a son of one of the nodes and the other player responds with a son of the other node of the same type (i.e. if both sons must be obtained by reduction of the same action).

The game may end after finite number of steps because one of the players cannot make a move. In this case the other player is the winner. If the game lasts forever then as the result we obtain a pair of infinite paths $\mathcal{P}_\alpha$ of $\mathcal{T}_\alpha$ and $\mathcal{P}_\beta$ of $\mathcal{T}_\beta$. Player *I* wins if there is no μ-trace on $\mathcal{P}_\alpha$ but there is a μ-trace on $\mathcal{P}_\beta$, otherwise player *II* is the winner.

We will say that $\mathcal{T}_\beta$ can *simulate* $\mathcal{T}_\alpha$ iff there is a winning strategy for the player *II* in the game described above. One can show that if β has a tableau that can simulate a tableau for α then for every model $\mathcal{M}$, state s and valuation *Val*, $\mathcal{M}, s, Val \models \alpha$ implies $\mathcal{M}, s, Val \models \beta$.

We can say that two tableaux are bisimilar iff each one can simulate the other. It is easy to see that this notion of bisimulation relates more tableaux than equivalence.

Please note that the game is asymmetric in step 2 but both tableaux are treated equally in step 3. This is connected to the fact that we have both $\langle a \rangle$ and $[a]$ modalities in the logic. It is possible to vary the definition of step 3. One such variation can be found in Definition 9.5.

We will not use this notion of bisimulation in this notes. In the completeness proof we will need even weaker notion of simulation (see Definition 9.5). Till that moment the notion of equivalence will be sufficient for us.

5 Disjunctive formulas

Operational semantics of formulas gives us some intuitions about the role of each connective. If we are to check that $\alpha \vee \beta$ holds, we choose (nondeterministically) one of the disjuncts. If we are to check $\mu X.\alpha(X)$ we try the equivalent formula $\alpha(\mu X.\alpha(X))$. When we check $\alpha \wedge \beta$ we must check that a state satisfies α and β. While disjunction acts like a nondeterministic choice, conjunction acts rather like universal branching of an alternating automaton. Such an alternating behaviour of a conjunction is a source of many difficulties. For example this is the only reason why our tableaux are labeled by sets of formulas and why the notion of trace is needed. One may ask whether we can avoid this difficulties?

From automata theory we know that alternating automata are equivalent to nondeterministic ones [13]. This suggests an idea that every formula should be equivalent to a formula which does not have universal branching behaviour represented by conjunction. Of course we cannot discard conjunctions completely from positive formulas as an example of the formula $(a \to \{p\}) \wedge (b \to \{q\})$ shows. Note that conjunction in this formula does not act as universal branching. It is rather an implicit conjunction from (usual, not alternating) automata on trees where transition relation forces one son to be labeled by a state q and the other one by q'. This implicit conjunction is the only form of conjunction that is present in fixpoint notation for the sets of trees defined by Niwiński [15]. It was proved that this fixpoint language has the same expressive power as *SnS*, monadic second order logic of n successors. Hence adding explicit conjunction to this language will not increase its expressive power.

Definition 5.1 (Special conjunctions and disjunctive formulas) *A conjunction $\alpha_1 \wedge \ldots \wedge \alpha_n$ is called* special *iff every α_i is either a literal or a formula of a form $(a \to \Gamma)$ and for any action a there is at most one conjunct of the form $(a \to \Gamma)$ among $\alpha_1, \ldots, \alpha_n$.*

The set of disjunctive formulas*, $\mathcal{F}_d$ is the smallest set defined by the following clauses:*

1. *every literal is a disjunctive formula,*

2. *if $\alpha, \beta \in \mathcal{F}_d$ then $\alpha \vee \beta \in \mathcal{F}_d$; if moreover X occurs only positively in α and not in the context $X \wedge \gamma$ for some γ, then $\mu X.\alpha, \nu X.\alpha \in \mathcal{F}_d$,*

3. $(a \to \Phi) \in \mathcal{F}_d$ *if* $\Phi \subseteq \mathcal{F}_d$,

4. special conjunction of disjunctive formulas is a disjunctive formula.

Remark 5.2 Many properties can be "naturally" expressed by disjunctive formulas. For example the properties q holds almost always and q holds infinitely often:

$$\mu X.(a \to \{X\}) \vee \nu Y.q \wedge (a \to \{Y\}) \qquad \nu X.\mu Y.(a \to \{Y\}) \vee (q \wedge (a \to \{X\}))$$

Remark 5.3 Modulo the order of application of (and) rules, disjunctive formulas have unique tableaux. Moreover on any infinite path there is one and only one infinite trace.

Theorem 5.4 For every (well named, positive and guarded) formula φ and every regular tableau $\mathcal{T}$ for φ (i.e. a tableau which is a regular tree) there is a disjunctive formula $\widehat{\varphi}$ with the tableau equivalent to $\mathcal{T}$.

Proof
Let $\mathcal{T} = \langle T, L\rangle$ be a regular tableau for φ. As $\mathcal{T}$ is a regular tree, it can be presented as a finite labeled graph $\mathcal{G} = \langle G, L_{\mathcal{G}}\rangle$, where G is a finite graph and $L_{\mathcal{G}}$ a labeling function.

We first show that it is possible to present $\mathcal{T}$ in a form of a tree with *back edges*, i.e., edges leading from some leaves to their ancestors. We will still apply tree-like terminology to such a structure, for example we will say that one node is a son of the other meaning that it is so in a tree obtained by forgetting about back edges.

Lemma 5.5 It is possible to construct a finite tree with back edges $\mathcal{T}_l = \langle T_l, L_l\rangle$, satisfying the following conditions:

1. $\mathcal{T}_l$ unwinds to $\mathcal{T}$.

2. Every node to which a back edge points can be assigned color magenta or navy in such a way that for any infinite path from the unwinding of $\mathcal{T}_l$ we have: there is a μ-trace on the path iff the highest node of $\mathcal{T}_l$ through which the path goes i.o. is colored magenta.

Proof
It is easy to see that there is a Rabin automaton on infinite strings recognising those paths of a tableau for φ which have a μ-trace on them. Form Theorem 2.21 it follows that there is an equivalent deterministic automaton $\mathcal{A}$ with a chain condition $\mathcal{C}$.

Given automaton $\mathcal{A}$ we construct our tree with back edges, $\mathcal{T}_l = \langle T_l, L_l\rangle$. Labeling function L_l will assign to nodes of $\mathcal{T}_l$ not sets of formulas but triples $(\textit{set of formulas}, \textit{state of } \mathcal{A}, \textit{node of } \mathcal{G})$. All such triples (Ω, q, k) will satisfy an additional requirement that $L_{\mathcal{G}}(k) = \Omega$.

- We label the root of $\mathcal{T}_l$ by a triple consisting of $\{\varphi\}$, a state q reachable from the start state of the automaton on letter $\{\varphi\}$ and the initial node of $\mathcal{G}$.

- If we have already constructed a node n of $\mathcal{T}_l$ labeled (Ω, q, k) then for any k' to which there is an edge from k in $\mathcal{G}$ we add a son of n labeled by (Ω', q', k') where, $\Omega' = L_{\mathcal{G}}(k')$ and q' is a state reachable from q on a letter Ω'. There is one exception to this rule. If a triple we are going to add is already a label of some ancestor m of n and no state with smaller than $\mathcal{C}(q')$ value of $\mathcal{C}$ function appears on the path from m to n then instead of creating a son of n we create a back edge from n to m. We color m magenta if $\mathcal{C}(q')$ is even otherwise we color m navy.

It should be clear that the constructed tree is finite and its unwinding with labeling restricted to the first components is just $\mathcal{T}$. If we take an infinite path $\mathcal{P}$ from the unwinding of $\mathcal{T}_l$ then the third components of the triples constitute a path of our initial tableau $\mathcal{T}$ and the second components constitute a run of our automaton on this path which is unique because the automaton is deterministic. For any path $\mathcal{P}$ there is the highest node m of $\mathcal{T}_l$ which $\mathcal{P}$ visits i.o. Let q be the state appearing in the label of m. From the construction of $\mathcal{T}_l$ we know that only sates with $\mathcal{C}$ value not smaller then $\mathcal{C}(q)$ appear on the path i.o. From the assignment of colors follows that if m is colored magenta then the run is accepting and there is a μ-trace on the path. Otherwise if m is colored navy the run is rejecting and all infinite traces on the path are ν-traces. $\square$

From the tree with back edges $\mathcal{T}_l$ we are going to construct a disjunctive formula which has a tableau equivalent to $\mathcal{T}$. We start from the leaves of $\mathcal{T}_l$ and going to the top assign a formula $\widehat{n}$ to each node n of $\mathcal{T}_l$ in the following way:

- If there are no edges going from n then in the label of n only literals and terminal formulas can occur. We let $\widehat{n}$ to be the conjunction of all the formulas appearing in the label of n.

- If there are edges going from n then we assume that every son of n has assigned some disjunctive formula. It will be convenient to assume that a formula assigned to a son is also assigned to an edge leading from n to this son. There can be also back edges leading from n to some ancestors of n and of course those ancestors have no formula assigned yet. To such a back edge from n to, say m, we assign a variable X_m (an index is a node to which the edge points). We first define an auxiliary formula γ depending on the rule which was applied in n.

  - If one of the rules $(\mu), (\nu), (reg)$ or (and) was applied in n then γ is exactly the same as the formula assigned to the only edge leading from n
  - If rule (or) was applied in n then there are two edges leading from n which have been assigned formulas ψ_1 and ψ_2. We let $\gamma = \psi_1 \vee \psi_2$.
  - If applied rule was (mod) then let Ψ_a be the set of all the formulas assigned to the edges leading from n to some node labeled by a result of reduction of action a. We let γ to be a conjunction of all the literals and terminal formulas appearing in $L(n)$ together with all the formulas of the form $(a \to \Psi_a)$.

If there are no back edges leading to n then $\widehat{n}$ is just γ. Otherwise let $\widehat{n} = \sigma X_n.\gamma$, where σ is μ or ν depending on whether n was colored magenta or navy respectively.

Observe that $\widehat{n_0}$ has only one tableau, call it $\widehat{\mathcal{T}}$. It is not difficult to check that $\widehat{\mathcal{T}}$ is equivalent to $\mathcal{T}$. Hence formula $\widehat{n_0}$ is semantically equivalent to $\widehat{\varphi}$.

Finally let us make the following useful observation which we will use in the completeness proof.

Observation 5.5.1 We can assume that $\widehat{n} \neq \widehat{m}$ whenever m is not near to n. We can assume that all back edges in $\mathcal{T}_l$ go from modal nodes to choice nodes.

Observe that while the assumption on back edges is harmless, it is not clear how easily satisfy the first requirement without a bit of "cheating". One solution would be to add countable number of constants denoting the truth. Having them we could add a different constant of this kind to every leaf of $\mathcal{T}_l$. As we have countable number of variables one such constant is enough because we can take formulas $\{\mu X_i.\top : i \in \mathcal{N}\}$.

We would like to stress that assumptions made in Observation 5.5.1 are not essential for the proofs which follow although simplify them a lot. $\square$

6 Applications of disjunctive formulas

This section is divided into two parts. In the first we show how to construct models for disjunctive formulas and that checking their satisfiability can be done in linear time. Second part is devoted to some connections with automata theory.

6.1 Satisfiability of disjunctive formulas

In Theorem 4.1 a general technique of model construction for the μ-calculus formulas was described. Till now it remains essentially the only known technique for model construction. It turns out that in case of disjunctive formulas model construction is much easier. This is described in the following theorem.

Theorem 6.1 A closed disjunctive formula α is satisfiable iff the formula β obtained from α by replacing all occurrences of μ-variables by $\bot$ and all occurrences ν-variables by $\top$ is satisfiable.

Proof
It is quite easy to show that if α is satisfiable then β is satisfiable. This can be done by induction on the structure of α (see the proof of Theorem 8.4 for the similar argument).

Let us assume that β is satisfiable. Let $\mathcal{T}_\alpha = \langle T_\alpha, L_\alpha \rangle$ and $\mathcal{T}_\beta = \langle T_\beta, L_\beta \rangle$ be tableaux for α and β respectively. Observe that T_β is a finite tree with no back edges which is "isomorphic" to β. Indeed if we consider β written as a tree then there is a direct correspondence between nodes of $\mathcal{T}_\beta$ and β. Tree T_α is an unwinding of a graph obtained from T_β by adding back edges from some nodes labeled $\{\bot\}$ or $\{\top\}$. Back edges from nodes labeled $\{\bot\}$ lead to nodes labeled with μ-variables. Similarly back edges from $\{\top\}$-nodes lead to ν-variables. We will assume that there is a function h which for any node of $\mathcal{T}_\alpha$ gives us the corresponding node of $\mathcal{T}_\beta$. This situation is presented schematically in Figure 6

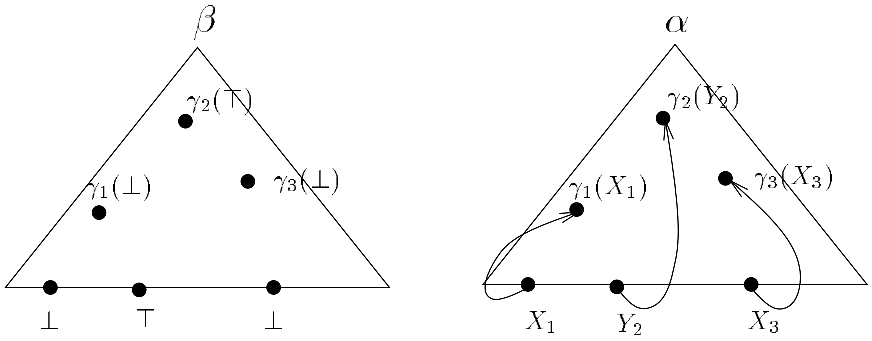

Figure 6: Tableaux $\mathcal{T}_\beta$ and $\mathcal{T}_\alpha$

Let M be a minimal (w.r.t. inclusion) marking of $\mathcal{T}_\beta$ consistent w.r.t. some arbitrary model for β. Let T'_β be the subtree of T_β consisting of the nodes marked with nonempty sets and let $T'_\alpha = h^{-1}(T'_\beta)$. As no leaf of T'_β can be labeled by $\{\bot\}$ it is easy to see that there is no μ-trace on any infinite path of T'_α. Also in every modal node m of T'_α there are the same literals as in the modal node $h(m)$ of T'_β. This shows that T'_α is a pre-model for α. Hence by Theorem 4.2 formula α is satisfiable. $\square$

Corollary 6.2 Satisfiability checking of disjunctive formulas can be done in linear time.

The above theorem can be also used to show how to prove the negation of an unsatisfiable disjunctive formula. We will show this in the next section as we need to have some proof system first.

6.2 Applications to automata theory

In this section we would like to discuss connections between parity automata on trees and disjunctive formulas. We will show that disjunctive formulas for the μ-calculus over binary trees closely correspond to parity automata. This gives yet another proof of Rabin's complementation lemma. We would also like to argue that disjunctive formulas can be considered as a natural generalisation of parity automata on trees to arbitrary transition systems.

Let us consider μ-calculus restricted to binary trees, that is suppose that the models of the μ-calculus are transition systems obtained from binary trees. Given a finite set of propositional constants $\{p_1, \ldots, p_k\}$ and a tree $t : \{l, r\}^* \to \Sigma$ over an alphabet $\Sigma = \mathcal{P}(\{p_1, \ldots, p_k\})$ we can consider it as a transition system $\langle\{l,r\}^*, R, \rho\rangle$ where $R(l) = \{(w, wl) : w \in \{l,r\}^*\}$, $R(r) = \{(w, wr) : w \in \{l,r\}^*\}$, and $\rho(p_i) = \{w \in \{l,r\}^* : p_i \in t(w)\}$.

For this restricted calculus we can take stronger version of rule (mod) which takes advantage of the information that there is always exactly one left and exactly one right successor.

$$(tmod) \quad \frac{\{\varphi : \varphi \in \Phi, (l \to \Phi) \in \Gamma\} \quad \{\varphi : \varphi \in \Phi, (r \to \Phi) \in \Gamma\}}{\{\Gamma\}}$$

Then we can modify the proofs of Theorems 4.10 and 5.4 in a straightforward way to show that any formula of the μ-calculus restricted to binary trees is equivalent to a disjunctive formula where all special conjuncts have the form $(l \to \{\alpha\}) \wedge (r \to \{\beta\}) \wedge \bigwedge \Gamma$, for some formulas α, β and a set of literals Γ. Tableau for such a formula can be presented as a tree with back edges as described in Lemma 5.5. This tree can be easily converted to an equivalent parity automaton. Here equivalent means that the automaton accepts exactly those trees in which root our initial formula is satisfied.

It is also quite straightforward to construct a disjunctive formula from a parity automaton. As μ-calculus over binary trees is closed under complementation this translations show that parity automata are closed under complementation. Put together with Theorem 2.26 we have got a proof of Rabin's complementation lemma:

Theorem 6.3 (Rabin) *Rabin tree automata are closed under complementation.*

This also gives a proof of the results from [15, 5]

Theorem 6.4 (Niwinski, Emerson & Jutla) *μ-calculus over binary trees is as expressive as Rabin tree automata hence equivalent to the monadic second order logic of two successors (S2S).*

One may ask what happens in the general situation when we allow arbitrary transition systems. In this case monadic second order logic (MS-logic for short) is not decidable hence the equivalence does not hold. This is an easy answer because the question was not exactly right. It is well known that μ-calculus cannot distinguish between a transition system and its unwinding which can easily be done in MS-logic.

More refined question would be then to ask what happens when we restrict to models which are unwindings of transition systems. In this case it is still true that the μ-calculus is weaker than MS-logic. In MS-logic one can say that a node has, say, two a-successors. This fact is not expressible in the μ-calculus.

This shows that there is a difference between the case when a degree of a node is know and the general case. This difference can be also exhibited on the automata level. Unwinding of every countable transition system can be encoded as a binary tree. From the results cited above it follows that for any μ-calculus formula there is an equivalent parity automaton where equivalent now means that the automaton recognises encodings of exactly those trees which are unwindings of transition systems from states where the formula holds. Nevertheless, for the reasons mentioned above it is not the case that for every automaton there exists an equivalent μ-calculus formula. We have argued above that disjunctive formulas for restricted μ-calculus correspond to parity automata. This allows us to consider disjunctive formulas for unrestricted μ-calculus to be a generalisation of the notion of a parity tree automaton to arbitrary transition systems.

7 Kozen's axiomatisation

Here we would like to present an axiomatisation of the μ-calculus proposed by Kozen [9]. We will adopt Gentzen-style formalisation because it fits nicely with the tableau rules we have. A sequent $\Gamma \vdash \Delta$ is a pair of finite sets of formulas. The meaning of $\Gamma \vdash \Delta$ is that implication $\bigwedge \Gamma \Rightarrow \bigvee \Delta$ is valid, or in other words that for every model $\mathcal{M}$ its state s and valuation Val if $\mathcal{M}, s, Val \models \bigwedge \Gamma$ then $\mathcal{M}, s, Val \models \bigvee \Delta$. The meaning of one sided sequent $\Gamma \vdash$ is that the conjunction of formulas from Γ is not satisfiable.

The system consists of the three sets of rules. First come the rules for propositional modal logic (so called system K):

$$(\neg) \qquad \frac{\Gamma \vdash \alpha, \Delta}{\neg\alpha, \Gamma \vdash \Delta} \qquad\qquad \frac{\alpha, \Gamma \vdash \Delta}{\Gamma \vdash \neg\alpha, \Delta}$$

$$(\wedge) \qquad \frac{\alpha, \beta, \Gamma \vdash \Delta}{\alpha \wedge \beta, \Gamma \vdash \Delta} \qquad\qquad \frac{\Gamma \vdash \alpha, \Delta \quad \Gamma \vdash \beta, \Delta}{\Gamma \vdash \alpha \wedge \beta, \Delta}$$

$$(\vee) \qquad \frac{\alpha, \Gamma \vdash \Delta \quad \beta, \Gamma \vdash \Delta}{\alpha \vee \beta, \Gamma \vdash \Delta} \qquad\qquad \frac{\Gamma \vdash \alpha, \beta, \Delta}{\Gamma \vdash \alpha \vee \beta, \Delta}$$

$$(\langle\rangle) \qquad \frac{\alpha, \{\beta : [a]\beta \in \Gamma\} \vdash \{\gamma : \langle a\rangle\gamma \in \Delta\}}{\langle a\rangle\alpha, \Gamma \vdash \Delta}$$

$$(cut) \qquad \frac{\Gamma \vdash \Delta, \gamma \quad \Sigma, \gamma \vdash \Omega}{\Gamma, \Sigma \vdash \Delta, \Omega}$$

Then we add two rules concerning the least fixpoint. First expresses the fact that the least fixpoint is a pre-fixpoint. The second is Park's least fixpoint rule which says that the least fixpoint is the least pre-fixpoint:

$$(\mu) \qquad \frac{\Gamma \vdash \alpha(\mu X.\alpha(X)), \Delta}{\Gamma \vdash \mu X.\alpha(X), \Delta}$$

$$(P) \qquad \frac{\alpha(\varphi) \vdash \varphi}{\mu X.\alpha(X) \vdash \varphi}$$

Finally as we add rules expressing dualities of the μ-calculus and defining

$(a \to \Phi)$ construct:

$$([\,])\qquad \frac{\Gamma \vdash \langle a\rangle\neg\alpha, \Delta}{[a]\alpha, \Gamma \vdash \Delta} \qquad\qquad \frac{\langle a\rangle\neg\alpha, \Gamma \vdash \Delta}{\Gamma \vdash [a]\alpha, \Delta}$$

$$(\nu)\qquad \frac{\Gamma \vdash \mu X.\neg\alpha(\neg X), \Delta}{\nu X.\alpha(X), \Gamma \vdash \Delta} \qquad\qquad \frac{\mu X.\neg\alpha(\neg X), \Gamma \vdash \Delta}{\Gamma \vdash \nu X.\alpha(X), \Delta}$$

$$(\_ \to \{\_\})\qquad \frac{[a]\bigvee\Phi, \{\langle a\rangle\alpha : \alpha \in \Phi\}, \Gamma \vdash \Delta}{(a \to \Phi), \Gamma \vdash \Delta} \qquad \frac{\Gamma \vdash \bigwedge\{\langle a\rangle\alpha : \alpha \in \Phi\} \wedge [a]\bigvee\Phi, \Delta}{\Gamma \vdash (a \to \Phi), \Delta}$$

Observe that there is a cut rule in the system. We don't know whether cut can be eliminated from the system at the expense of adding some finite set of "reasonable" rules.

A sequent $\Gamma \vdash \Delta$ is called an *axiom* iff $\Gamma \cap \Delta \neq \emptyset$ or $\bot \in \Gamma$ or $\top \in \Delta$. A *proof* of a sequent $\Gamma \vdash \Delta$ in the system is a finite tree constructed with the rules above which root is labeled by $\Gamma \vdash \Delta$ and all the leaves are axioms.

As all our tools were developed for positive guarded formulas we need to show that if provability is concerned we can restrict to such formulas.

Fact 7.1 Every formula is provably equivalent to a positive guarded formula.

Proof
Just observe that all the steps used in the proof of Proposition 2.7 use provable equivalences. □

A rule will be called *admissible* iff it is possible to prove the conclusions of the rule assuming that premises of the rule are additional axioms. The following rules were proved admissible in [9]:

$$(\langle\rangle)\qquad \frac{\{\psi\} \cup \{\bigvee\Theta : (a \to \Theta) \in \Gamma, \Theta \neq \Psi\} \vdash}{(a \to \Psi), \Gamma \vdash} \qquad \text{for some } \psi \in \Psi$$

$$(fix)\qquad \frac{\alpha(\mu X.\neg(\bigwedge\Gamma) \wedge \alpha(X)), \Gamma \vdash}{\mu X.\alpha(x), \Gamma \vdash}$$

$$(mon)\qquad \frac{\alpha \vdash \beta}{\varphi(\alpha) \vdash \varphi(\beta)} \quad X \text{ occurs only positively in } \varphi(X)$$

Our main goal is to prove the following theorem.

Theorem 7.2 (Completeness) *Kozen's axiomatisation of the propositional μ-calculus is complete, that is for every unsatisfiable formula φ there exists a proof of the sequent $\varphi \vdash$ in the system.*

In the next section we will consider some special cases when we put some syntactic restrictions on the form of φ. Among others we will prove the theorem for disjunctive formulas. In the last section we will show that every formula is provably equivalent to a disjunctive formula.

8 Special cases of the completeness theorem

In this section we will consider classes of formulas for which the provability is easier than in the general case. We recall a notion of an *aconjunctive formula* [9] and propose its slight generalisation called *weakly aconjunctive* formulas. Our goal is to obtain a generalisation of the main result from [9] which states that for every unsatisfiable aconjunctive formula the sequent $\varphi \vdash$ is provable. We introduce the notion of *thin refutation* which isolates the cases for which the original proof still goes through. From Theorem 4.6 we know that every unsatisfiable formula has a refutation. It turns out that in case of weakly aconjunctive formulas this refutation is thin.

8.1 Aconjunctiveness

Definition 8.1 (Aconjunctiveness) *Let φ be a formula, $\mathcal{D}_\varphi$ be its binding function and $\leq_\varphi$ dependency ordering (see Definition 2.11).*

— *We say that a variable X is* active *in ψ, a subformula of φ, iff there is a variable Y appearing in ψ with $X \leq_\varphi Y$.*

— *Let X be a variable with its binding definition $\mathcal{D}_\varphi(X) = \mu X.\gamma(X)$. Variable X is called aconjunctive iff for all subformulas of γ of the form $\alpha \wedge \beta$ it is not the case that X is active in α as well as in β.*

— *Variable X as above is called* weakly aconjunctive *iff for all subformulas of γ of the form $\alpha \wedge \beta$ if X is active both in α and β then $\alpha \wedge \beta$ is a special conjunction as defined in Definition 5.1*

— *Formula φ is called* (weakly) aconjunctive *iff all μ-variables in φ are (weakly) aconjunctive.*

In the following we will be interested only in weakly aconjunctive formulas. Definition of aconjunctive formulas was recalled just to give a comparison of the two notions.

From the next observation follows that all formulas appearing in a tableau for a weakly aconjunctive formula are weakly aconjunctive.

Proposition 8.2 Every formula appearing in a tableau for φ is a subformula of φ.

The next proposition states some closure properties of the class of weakly aconjunctive formula. Observe that weakly aconjunctive formulas are not closed under taking the least fixpoint.

Proposition 8.3 If $\gamma(X)$ and δ are weakly aconjunctive formulas then $\gamma[\delta/X]$, $\nu X.\gamma(X)$ and $\delta \wedge \gamma(X)$ are also weakly aconjunctive formulas.

Proof
As we consider only well named formulas when conjunction is formed we make

sure that the bound variables in δ and $\gamma(X)$ are different. With this observation it should be easy to see that $\nu X.\gamma(X)$ and $\delta \wedge \gamma(X)$ are weakly aconjunctive.

Also while performing substitution $\gamma[\delta/X]$ we keep bound variables of δ distinct from the bound variables of γ. Let $\alpha = \gamma[\delta/X]$ and let Y be a μ-variable of α. It may be either a bound variable from δ or from γ. If it is bound variable from γ then observe that because no bound variable of γ is free in δ for every $Y \leq_\alpha Z$, variable Z is a bound variable of γ. Hence Y is aconjunctive in α iff it was aconjunctive in γ. For the similar reason any μ-variable of δ is aconjunctive in α. $\square$

8.2 Proving (negations of) weakly aconjunctive formulas

First let us consider disjunctive formulas.

Theorem 8.4 For every unsatisfiable disjunctive formula α the sequent $\alpha\vdash$ is provable.

Proof

In Theorem 6.1 we have shown that α is satisfiable iff β obtained form α by replacing all μ-variables by $\perp$ and all ν-variables by $\top$ is satisfiable.

We prove the theorem by induction on the structure of α.

If α is a special conjunction $\alpha_1 \wedge \ldots \wedge \alpha_n$ then we have two cases. If $\alpha_i = \perp$ or $\alpha_i = \neg\alpha_j$ for some $i, j \in \{1, \ldots, n\}$ then $\alpha\vdash$ is easily provable. Otherwise one of the conjuncts must be of the form $(a \to \Gamma)$ and one of the formulas from Γ must be unsatisfiable. From induction assumption using rule $(\langle\rangle)$ we obtain the proof of $\alpha\vdash$.

If $\alpha = \gamma \vee \delta$ then by induction assumption we have proofs of $\gamma\vdash$ and $\delta\vdash$ so we can use the left $(\vee)$ rule.

If $\alpha = \mu X.\gamma(X)$ then as this formula is unsatisfiable so is $\gamma(\perp)$. By induction assumption there is a proof of $\gamma(\perp)\vdash$. Using derivable rule:

$$\frac{\gamma(\perp)\vdash}{\mu X.\gamma(X)\vdash}$$

we have the proof of $\mu X.\gamma(X)\vdash$.

If $\alpha = \nu X.\gamma(X)$ then we consider $\gamma(\top)$. It is of course a disjunctive formula. By Theorem 6.1 $\gamma(\top)$ is satisfiable iff $\nu X.\gamma(X)$ is satisfiable. As the later formula is not satisfiable we have by induction assumption the proof of $\gamma(\top)\vdash$ and we can use derivable rule:

$$\frac{\gamma(\top)\vdash}{\nu X.\gamma(X)\vdash}$$

$\square$

Remark 8.5 The proof of the above theorem crucially depends on the fact that we never run into the situation when μ is considered in a conjunction with some other formula. Observe that the rule:

$$\frac{\alpha(\bot), \Gamma \vdash}{\mu X.\alpha(X), \Gamma \vdash}$$

is not sound. This means that our proof method breaks down in the general case.

It should be mentioned that the infinitary rule:

$$\frac{\{\gamma^n(\bot), \Gamma \vdash \; : n \in \mathcal{N}\}}{\mu X.\gamma(X), \Gamma \vdash}$$

is sound because of the small model property. This rule obviously gives a complete system. From the small model property it follows that this rule can be replaced by a "finitary rule":

$$\frac{\gamma^n(\bot), \Gamma \vdash}{\mu X.\gamma(X), \Gamma \vdash} \quad n > 2^k, \text{ where } k \text{ is the size of the conclusion}$$

In [9] Kozen proved a result similar to Theorem 8.4 but for aconjunctive formulas. Below we present a minor generalisation of this result. We will need it in our completeness proof.

Definition 8.6 A refutation (see Definition 4.5) is called *thin* iff whenever a formula of the form $\alpha \wedge \beta$ is reduced in some node of the refutation and some variable is active in α as well as in β then either $\alpha \wedge \beta$ is a special conjunction or one of the conjuncts is immediately discarded by the use of the weakening rule:

$$\frac{\Gamma}{\{\alpha, \Gamma\}}$$

Theorem 8.7 Let φ be a formula. If there exists a thin refutation for φ then the sequent $\varphi \vdash$ is provable.

We will not prove this theorem here, its proof is a minor alternation of the proof from [9] and it is much more complicated than that of Theorem 8.4. Instead let us give an example of the method used in the proof.

Consider the sequent

$$\mu X.p \vee (a \to \{X\}), \nu Y.\neg p \wedge (a \to \{Y\}) \vdash$$

Below we present a refutation of this sequent.

$$\begin{array}{c}
\{\mu X.p \vee (a \to \{X\}), \nu Y.\neg p \wedge (a \to \{Y\})\} \\ \hline
\{p \vee (a \to \{X\}), \nu Y.\neg p \wedge (a \to \{Y\})\} \\ \hline
\{p \vee (a \to \{X\}), \neg p, (a \to \{Y\})\} \\ \hline
\begin{array}{cc}
\{p, \neg p, (a \to \{Y\})\} &
\begin{array}{c}
\{(a \to \{X\}), \neg p, (a \to \{Y\})\} \\ \hline
\{X, Y\} \\ \hline
\{p \vee (a \to \{X\}), \neg p, (a \to \{Y\})\} \\ \hline
\vdots
\end{array}
\end{array}
\end{array}$$

As the sequent just above the dots appears already in the refutation we have a loop which gives us an infinite path. On this path there is a μ-trace composed of the leftmost formulas of the path.

We convert this refutation into a finite proof by using rule (fix) in the first regeneration of $\mu X.p \vee (a \to \{X\})$. This allows us to "remember" the node where regeneration was performed. We can then use this information when we will be regenerating this formula once again in the same context. (Below we denote $\mu X.\neg(\nu Y.\neg p \wedge (a \to \{Y\})) \wedge (a \to \{X\})$ by γ and use variables to stand for corresponding formulas.)

$$
\begin{array}{c}
\mu X.p \vee (a \to \{X\}), \nu Y.\neg p \wedge (a \to \{Y\}) \vdash \\
\hline
p \vee (a \to \{\gamma\}), \nu Y.\neg p \wedge (a \to \{Y\}) \vdash \\
\hline
p \vee (a \to \{\gamma\}), \neg p, (a \to \{Y\}) \vdash \\
\hline
p, \neg p, (a \to \{Y\}) \vdash \qquad\qquad (a \to \{\gamma\}), \neg p, (a \to \{Y\}) \vdash \\
\hline
\gamma, Y \vdash \\
\hline
\gamma, \nu Y.\neg p \wedge (a \to \{Y\}) \vdash \\
\hline
\mu X.\neg(\nu Y.\neg p \wedge (a \to \{Y\})) \wedge (a \to \{X\}), \nu Y.\neg p \wedge (a \to \{Y\}) \vdash \\
\hline
\neg(\nu Y.\neg p \wedge (a \to \{Y\})), \nu Y.\neg p \wedge (a \to \{Y\}) \vdash
\end{array}
$$

The last sequent is clearly provable. The last step is obtained using (cut) rule with the obvious sequent.

The following easy consequence of Proposition 8.2 shows that Theorem 8.7 gives use means to deal with weakly aconjunctive formulas.

Fact 8.8 Every refutation for a weakly aconjunctive formula is a thin refutation.

9 Provable equivalence

Having Theorem 8.4 to prove completeness it is enough to show that for any unsatisfiable formula φ there is a disjunctive unsatisfiable formula $\widehat{\varphi}$ such that $\varphi \vdash \widehat{\varphi}$ is provable. Of course we could just take $\widehat{\varphi}$ to be $\bot$ but then the proof of this fact would be exactly as difficult as showing completeness. So in general we will look for more complicated formulas then $\bot$. Because we will prove this fact by induction on φ we clearly need to consider also satisfiable formulas.

This whole section is devoted to the proof of the following theorem:

Theorem 9.1 For every positive, guarded formula φ there is an equivalent disjunctive formula $\widehat{\varphi}$ such that $\varphi \vdash \widehat{\varphi}$ is provable.

The proof will proceed by induction on the structure of φ.

Case: φ is a literal In this case $\widehat{\varphi}$ is just φ.

Case: $\varphi = \alpha \vee \beta$ By the induction assumption there are disjunctive formulas $\widehat{\alpha}, \widehat{\beta}$ equivalent to α and β respectively. We let $\widehat{\alpha \vee \beta}$ to be $\widehat{\alpha} \vee \widehat{\beta}$ and of course $\alpha \vee \beta \vdash \widehat{\alpha} \vee \widehat{\beta}$ is provable because $\alpha \vdash \widehat{\alpha}$ and $\beta \vdash \widehat{\beta}$ are provable.

Case: $\varphi = (a \to \Phi)$ This case is very similar to the above.

Case: $\varphi = \nu X.\alpha(X)$ By the induction assumption there is a disjunctive formula $\widehat{\alpha}(X)$ equivalent to $\alpha(X)$. Of course $\nu X.\alpha(X)$ is equivalent to $\nu X.\widehat{\alpha}(X)$ and $\nu X.\alpha(X) \vdash \nu X.\widehat{\alpha}(X)$ is provable. Unfortunately $\nu X.\widehat{\alpha}(X)$ may not be a disjunctive formula. This is because X may occur in a context $X \wedge \gamma$ for some γ. Therefore we have to construct $\widehat{\varphi}$ from the scratch.

By Theorem 5.4 there is a disjunctive formula $\widehat{\varphi}$ which has the tableau equivalent to some regular tableau $\mathcal{T}$ for $\nu X.\widehat{\alpha}(X)$. By Theorem 4.10 the two formulas are equivalent. We are left to show that $\nu X.\widehat{\alpha}(X) \vdash \widehat{\varphi}$ is provable in Kozen's system. As every disjunctive formula is a weakly aconjunctive formula, by Proposition 8.3 we have that $\nu X.\widehat{\alpha}(X)$ is an weakly aconjunctive formula. Unfortunately we cannot directly apply Theorem 8.7 to $\nu X.\widehat{\alpha}(X), \neg\widehat{\varphi} \vdash$. This is because $\neg\widehat{\varphi}$ may not be a weakly aconjunctive formula. Nevertheless we know that the two formulas have equivalent tableaux and we can use this information.

Lemma 9.2 Suppose that we have a weakly aconjunctive formula α and a disjunctive formula δ which have equivalent tableaux. In this case the sequent $\alpha \vdash \delta$ is provable.

Proof
Let $\mathcal{T}_\alpha = \langle T_\alpha, L_\alpha \rangle$ and $\mathcal{T}_\delta = \langle T_\delta, L_\delta \rangle$ be tableaux for α and δ respectively. Let $\mathcal{E} : \mathcal{T}_\alpha \to \mathcal{T}_\delta$ be an equivalence function. We will construct a thin refutation $\mathcal{R}$ for $\alpha, \neg\delta \vdash$.

To facilitate the construction we will define correspondence function $\mathcal{C}_\alpha$ which assigns to every considered node of $\mathcal{R}$ (that is not to every node) a node of $\mathcal{T}_\alpha$ such that:

$(*)$ $L(n) = L_\alpha(\mathcal{C}_\alpha(n)) \cup \{\neg \bigwedge L_\delta(\mathcal{E}(\mathcal{C}_\alpha(n)))\}$

The root r of $\mathcal{R}$ will be of course labeled by $\alpha, \neg\delta \vdash$ and setting $\mathcal{C}_\alpha(r)$ to be the root of $\mathcal{T}_\alpha$ establishes condition $(*)$.

Observation 9.2.1 Suppose we have already constructed $\mathcal{R}$ up to a node n, $\mathcal{C}_\alpha(n)$ is a choice node and condition $(*)$ is satisfied. We can construct a finite part of $\mathcal{R}$ such that for each leaf m of the constructed part we can define $\mathcal{C}_\alpha(m)$ so that it is a modal node and $(*)$ is satisfied. Moreover the traces from n to m are *reflected*. This means that the traces form n to m are exactly the traces from $\mathcal{C}_\alpha(n)$ to $\mathcal{C}_\alpha(m)$ with an exception of the trace from $\neg \bigwedge L_\delta(\mathcal{E}(\mathcal{C}_\alpha(n)))$ to $\neg \bigwedge L_\delta(\mathcal{E}(\mathcal{C}_\alpha(m)))$ which corresponds to negated (unique) trace from $\mathcal{E}(\mathcal{C}_\alpha(n))$ to $\mathcal{E}(\mathcal{C}_\alpha(m))$.

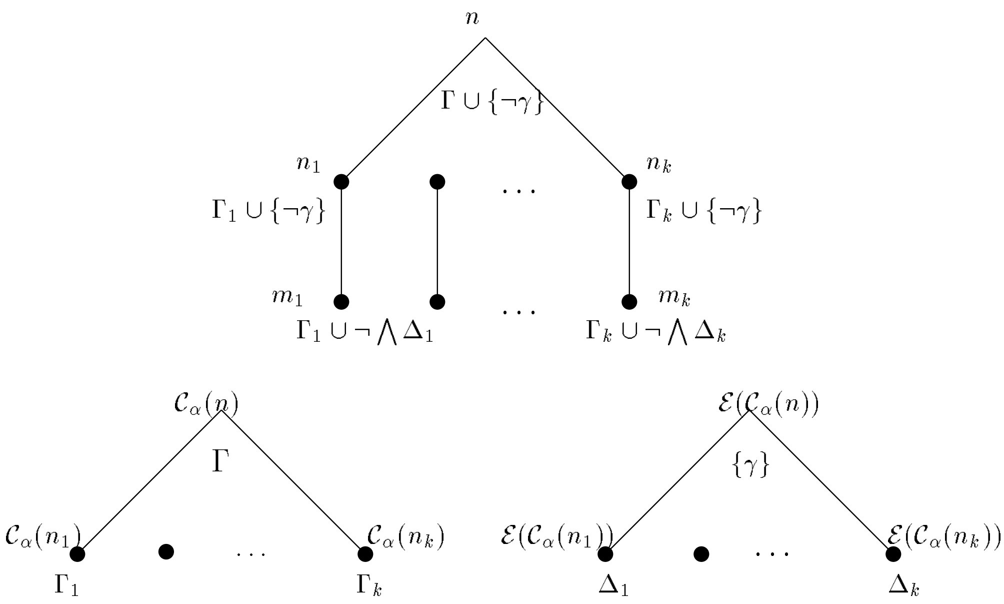

Figure 7: The proof of Observation 9.2.1. The labels of the nodes are written below the nodes.

Proof: The idea of the proof is presented in Figure 7. We know that $\mathcal{E}(\mathcal{C}_\alpha(n))$ is a choice node and as δ is a disjunctive formula $L_\delta(\mathcal{E}(\mathcal{C}_\alpha(n)))$ is one element set, say $\{\gamma\}$. Hence $L(n) = L_\alpha(\mathcal{C}_\alpha(n)) \cup \{\neg\gamma\}$. Let us apply as long as possible rules other than $(\langle\rangle)$ and weakening to all the formulas in $L(n)$ except $\neg\gamma$ in the same order as they were applied from $\mathcal{C}_\alpha(n)$. This way we obtain a part of a tree rooted in n with leaves $n_1, \ldots, n_k$. For every $j = 1, \ldots, k$ the label $L(n_j)$ contains $\neg\gamma$ and some set of formulas Γ_j to which only $(\langle\rangle)$ may be applicable. It is easy to see that every n_j corresponds to some modal node $\mathcal{C}_\alpha(n_j)$ near $\mathcal{C}_\alpha(n)$ with the property that $L_\alpha(\mathcal{C}_\alpha(n_j)) = \Gamma_j$. Let us look at the path from $\mathcal{E}(\mathcal{C}_\alpha(n))$ to $\mathcal{E}(\mathcal{C}_\alpha(n_j))$ in $\mathcal{T}_\delta$. Because δ is a disjunctive formula, on this path first only (σ), (reg) and (or) rules may be applied and then we have zero or more applications of (and) rule. Let us apply dual rules to $\neg\gamma$ (dual to (μ) is (ν), (reg) is self-dual). When it comes to application of (or) rule in $\mathcal{T}_\delta$, apply (and) rule followed by weakening to leave only the conjunct which appears on the path to $\mathcal{E}(\mathcal{C}_\alpha(n_j))$.

This way we arrive at a node m_j. If we define $\mathcal{C}_\alpha(m_j) = \mathcal{C}_\alpha(n_j)$ then its label can be presented as $L_\alpha(\mathcal{C}_\alpha(m_j)) \cup \{\neg \bigwedge L_\delta(\mathcal{E}(\mathcal{C}_\alpha(m_j)))\}$. $\Box$

Observation 9.2.2 Suppose we have constructed the refutation up to a node m. Assume that $\mathcal{C}_\alpha(m)$ is a modal node and $(*)$ is satisfied. We can construct a finite part of the refutation and define $\mathcal{C}(n)$ for each leaf of the constructed part so that $\mathcal{C}_\alpha(n)$ is a choice node, condition $(*)$ is satisfied and traces are reflected.

Proof: Let $\gamma = \bigwedge L_\delta(\mathcal{E}(\mathcal{C}_\alpha(m))) = \bigwedge\{\gamma_1, \ldots, \gamma_l\}$ and $\Gamma = L_\alpha(\mathcal{C}_\alpha(m))$. By $(*)$ we have $L(m) = \{\neg\gamma\} \cup \Gamma$. Node $\mathcal{E}(\mathcal{C}_\alpha(m))$ is a modal node hence every

γ_i is either a literal or formula of the form $(a \to \Phi)$. When we negate γ we obtain a disjunction of negations of such formulas. Let us apply (or) rule to eliminate all these disjunctions. This way we obtain new leaves $m_1, \ldots, m_l$. For any $i \in 1, \ldots, l$ node m_i is labeled by $\Gamma_j \cup \{\neg\gamma_i\}$. We use γ_i to decide what rule to apply in m_i.

- If γ_i is a literal or a terminal formula we are done because γ_i appears in Γ. This follows directly from $(*)$ and the definition of an equivalence of tableaux.
- If γ_i is of a form $(a \to \Phi)$ with $\Phi \neq \emptyset$ then negated it becomes $\bigvee\{[a]\neg\varphi : \varphi \in \Phi\} \vee \langle a \rangle \bigwedge\{\neg\varphi : \varphi \in \Phi\}$ or rather translation of this formula to $(a \to \Theta)$ notation. We apply disjunction rules as long as possible. This way we obtain a part of a tree. Any leaf u of this tree is labeled by Γ and one of the disjuncts.
  - Suppose this disjunct is $(a \to \emptyset)$. As $\Phi \neq \emptyset$ there is an a-son of $\mathcal{E}(\mathcal{C}_\alpha(m))$ so there is an a-son of $\mathcal{C}_\alpha(n)$. Hence there is $(a \to \Theta) \in \Gamma$ with $\Theta \neq \emptyset$. We obtain an axiom after one application of $(\langle\rangle)$.
  - If it is $(a \to \{\neg\varphi\})$ for some $\varphi \in \Phi$ then let m_δ be a son of $\mathcal{E}(\mathcal{C}_\alpha(m))$ labeled by $\{\varphi\}$. Node $\mathcal{E}^{-1}(m_\delta)$ is an a-son of $\mathcal{C}_\alpha(m)$. It has a label of the form $\{\psi\} \cup \{\bigvee\Theta : (a \to \Theta) \in \Gamma, \Theta \neq \Psi\}$ for some $(a \to \Psi) \in \Gamma$ and $\psi \in \Psi$. We apply $(\langle\rangle)$ rule to $(a \to \Psi)$ in $L(u)$ and obtain a son u' of u labeled $\{\psi\} \cup \{\bigvee\Theta : (a \to \Theta) \in \Gamma, \Theta \neq \Psi\} \cup \{\neg\varphi\}$. We let $\mathcal{C}_\alpha(u') = \mathcal{E}^{-1}(m_\delta)$.
  - If it is $(a \to \{\bigwedge\{\neg\varphi : \varphi \in \Phi\}, \top\})$ then we apply $(\langle\rangle)$ rule to this formula and obtain a son u' of u labeled

    $$\{\bigwedge\{\neg\varphi : \varphi \in \Phi\}\} \cup \{\bigvee\Theta : (a \to \Theta) \in \Gamma\}$$

    Let us apply (or) rule to one of $\bigvee\Theta$. This way we obtain a part of a tree with one leaf for every $\psi \in \Theta$. This leaf is labeled by

    $$\{\psi\} \cup \{\bigvee\Theta' : (a \to \Theta') \in \Gamma, \Theta' \neq \Theta\} \cup \{\bigwedge\{\neg\varphi : \varphi \in \Phi\}\}$$

    Let n_α be a son of $\mathcal{C}_\alpha(m)$ labeled by $\{\psi\} \cup \{\bigvee\Theta' : (a \to \Theta') \in \Gamma, \Theta' \neq \Theta\}$. Let $\{\varphi\}$ be the label of $\mathcal{E}(n_\alpha)$. To $\bigwedge\{\neg\varphi : \varphi \in \Phi\}$ we apply (and) rule followed by weakening to obtain a node u'' labeled by the sequent:

    $$\{\psi\} \cup \{\bigvee\Theta' : (a \to \Theta') \in \Gamma, \Theta' \neq \Theta\} \cup \{\neg\varphi\}$$

    Define $\mathcal{C}_\alpha(u'') = n_\alpha$

□

The above two observations describe the construction of $\mathcal{R}$ completely. All finite paths of the constructed tree end with axioms. For every infinite path $\mathcal{P}$ we have two possibilities. There may be a μ-trace on a path of $\mathcal{T}_\alpha$ designated by the image of $\mathcal{P}$ under $\mathcal{C}_\alpha$. If it is so then by trace reflection there is also a

μ-trace on $\mathcal{P}$. If there is no μ-trace on $\mathcal{C}_\alpha(\mathcal{P})$ then there cannot be a μ-trace on $\mathcal{E}(\mathcal{C}_\alpha(\mathcal{P}))$ because of the definition of tableau equivalence. Hence there is a ν-trace on $\mathcal{C}_\delta(\mathcal{P})$ which negated in $\mathcal{R}$ becomes a μ-trace.

This means that every finite path of $\mathcal{R}$ ends in a set containing a literal and its negation and on every infinite path there is a μ-trace. $\mathcal{R}$ is also a thin refutation because of the way we have constructed it and the fact that α is a weakly aconjunctive formula. Hence by Theorem 8.7 the sequent $\alpha \wedge \neg\delta \vdash$ is provable.

$\square$

Case: $\varphi = \alpha \wedge \beta$ By the induction assumption there are disjunctive formulas $\widehat{\alpha}, \widehat{\beta}$ equivalent to α and β respectively and such that sequents $\alpha \vdash \widehat{\alpha}$ and $\beta \vdash \widehat{\beta}$ are provable. Sequent $\alpha \wedge \beta \vdash \widehat{\alpha} \wedge \widehat{\beta}$ is of course provable. By Theorem 8.7 there is a disjunctive formula $\widehat{\varphi}$ which has a tableau equivalent to some regular tableau for $\widehat{\alpha} \wedge \widehat{\beta}$. By Proposition 8.3 formula $\widehat{\alpha} \wedge \widehat{\beta}$ is aconjunctive. We can use Lemma 9.2 to show that sequent $\widehat{\alpha} \wedge \widehat{\beta} \vdash \widehat{\varphi}$ is provable.

Case: $\varphi = \mu X.\alpha(X)$ This case is more complicated than the case for the greatest fixpoint. As in that case we have by the induction assumption a disjunctive formula $\widehat{\alpha}(X)$ equivalent to $\alpha(X)$. Unfortunately this time $\mu X.\widehat{\alpha}(X)$ may not be a weakly aconjunctive formula hence we cannot use the same argument as in the last two cases. Let us nevertheless carry on and see where modifications are needed.

By Theorem 5.4 there is a disjunctive formula $\widehat{\varphi}$ which has the tableau equivalent to some regular tableau $\mathcal{T}$ for $\mu X.\widehat{\alpha}(X)$. By Theorem 4.10 the two formulas are equivalent. We are left to show that $\mu X.\widehat{\alpha}(X) \vdash \widehat{\varphi}$ is provable in Kozen's system. Because $\mu X.\widehat{\alpha}(X)$ may not be an aconjunctive formula we cannot just use Lemma 9.2 to $\mu X.\widehat{\alpha}(X)$ and $\widehat{\varphi}$. What we can do is to use Park's induction rule (P) if we only prove the sequent $\widehat{\alpha}(\widehat{\varphi}) \vdash \widehat{\varphi}$.

This time by Proposition 8.3 we know that $\widehat{\alpha}(\widehat{\varphi})$ is an aconjunctive formula but we meet another obstacle preventing us from using Lemma 9.2. We don't know weather $\widehat{\alpha}(\widehat{\varphi})$ and $\widehat{\varphi}$ have equivalent tableaux. Indeed it may be the case that they don't have equivalent tableaux.

For example consider the formula

$$\varphi = \mu X.\nu Y.(a \rightarrow \{X \wedge Y, p\})$$

This is not completely honest example, as $\nu Y.(a \rightarrow \{X \wedge Y, p\})$ is not a disjunctive formula, but is good enough to show the problems. There are also "honest" examples for the same phenomenon.

A tableau for the formula looks as follows (several steps are omitted):

$$\begin{array}{c}
\underline{\{\mu X.\nu Y.(a \to \{X \wedge Y, p\})\}}\\
\underline{\{(a \to \{X \wedge Y, p\})\}}\\
\{p\} \qquad\qquad \underline{\{X \wedge Y\}}\\
\qquad\qquad \underline{\{(a \to \{X \wedge Y, p\})\}}\\
\vdots
\end{array}$$

Construction from the proof of Theorem 5.4 gives us a disjunctive formula $\widehat{\varphi} = \mu X.(a \to \{p, X\})$ which has the tableau equivalent to the one presented above. Let us see how the tableau for $\nu Y.(a \to \{\widehat{\varphi} \wedge Y, p\})$ looks like.

$$\begin{array}{c}
\underline{\{\nu Y.(a \to \{\widehat{\varphi} \wedge Y, p\})\}}\\
\underline{\{(a \to \{\widehat{\varphi} \wedge Y, p\})\}}\\
\{p\} \qquad\qquad\qquad \underline{\{\widehat{\varphi} \wedge Y\}}\\
\underline{\{\mu X.(a \to \{p, X\}), \nu Y.(a \to \{\widehat{\varphi} \wedge Y, p\})\}}\\
\underline{\{(a \to \{p, X\}), (a \to \{\widehat{\varphi} \wedge Y, p\})\}}\\
\{p\} \quad \underline{\{p, \widehat{\varphi} \wedge Y\}} \quad \underline{\{X, \widehat{\varphi} \wedge Y\}}\\
\vdots \qquad\qquad \vdots
\end{array}$$

Observe that this tableau has a modal node with three sons and in the tableau for $\widehat{\varphi}$ every modal node has at most one son. Hence this two tableaux cannot be equivalent.

What we need is some weaker notion of correspondence between tableaux but it should be strong enough for to give us something like Lemma 9.2. Below we propose such a notion which we call *tableau consequence*. This notion will be defined in terms of games on tableaux. To simplify the definition we will introduce the notion of a *wide tableau*.

Definition 9.3 *Wide tableaux* are constructed according to the same rules as tableaux but rule (mod) is replaced by $(wmod)$:

$$(wmod) \quad \frac{\{\psi\} \cup \{\bigvee \Theta : (a \to \Theta) \in \Gamma, \Theta \neq \Psi\} \qquad \text{for every } (a \to \Psi) \in \Gamma, \psi \in \Psi \qquad \{\bigvee \Theta : (a \to \Theta) \in \Gamma\}}{\{\Gamma\}}$$

Compared to (mod) rule $(wmod)$ has new assumptions, one for each action a such that there is a formula of the form $(a \to \Psi)$ in Γ. We will call sons of the old type $\langle a \rangle$*-sons*. The sons of the new type will be called $[a]$*-sons*. Observe that for any action a we can have at most one $[a]$-son of a node.

Remark 9.4 For example:

$$\frac{\{\alpha_1\} \quad \{\alpha_2\} \quad \{\alpha_1 \vee \alpha_2\} \quad \{\beta_1\} \quad \{\beta_2\} \quad \{\beta_1 \vee \beta_2\}}{\{(a \to \{\alpha_1, \alpha_2\}), (b \to \{\beta_1, \beta_2\})\}}$$

is an instance of ($wmod$) rule. There are two $\langle a\rangle$-sons, two $\langle b\rangle$-sons, one $[a]$-son and one $[b]$-son.

Rule (mod) with the same conclusion would have only four assumptions, $[a]$ and $[b]$-sons would be missing. In rule ($wmod$) we make explicit universal requirements of the conclusion. The meaning of $(a \to \{\alpha_1, \alpha_2\})$ is that there is a state reachable by action a where α_1 is satisfied (represented by $\langle a\rangle$-son labeled $\{\alpha_1\}$), there is a state where α_2 is satisfied (represented by $\langle a\rangle$-son labeled $\{\alpha_2\}$) and all the states reachable by action a must satisfy $\alpha_1 \vee \alpha_2$, this is represented by $[a]$-son labeled $\{\alpha_1 \vee \alpha_2\}$.

Definition 9.5 Given a pair of wide tableaux $(\widetilde{\mathcal{W}}, \mathcal{W})$, where $\widetilde{\mathcal{W}} = \langle \widetilde{T}, \widetilde{L}\rangle$ and $\mathcal{W} = \langle T, L\rangle$, we define a two player game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ with the following rules.

- The starting position is a pair of the roots of both tableaux.
- If a position of a play is $(\widetilde{n}, n)$, both nodes being choice nodes of $\widetilde{\mathcal{W}}$ and $\mathcal{W}$ respectively, then player I chooses a modal node $\widetilde{m}$ near $\widetilde{n}$ and player II replies by choosing a modal node m near n with the property that every literal and terminal formula from $L(m)$ appears in $\widetilde{L}(\widetilde{m})$.
- If a position is $(\widetilde{m}, m)$, a pair of modal nodes from $\widetilde{\mathcal{W}}$ and $\mathcal{W}$ respectively, then player I can choose a son n of m and player II has to respond with a son $\widetilde{n}$ of $\widetilde{m}$ of the same kind. That is if n is a $\langle a\rangle$-son then $\widetilde{n}$ must be a $\langle a\rangle$-son and if n is a $[a]$-son so must be $\widetilde{n}$.

The game may end in a finite number of steps because one of the players cannot make a move. In this case the opposite player wins. When the game has infinitely many steps we get as the result two infinite paths: $\widetilde{\mathcal{P}}$ from $\widetilde{\mathcal{W}}$ and $\mathcal{P}$ from $\mathcal{W}$. Player I wins if there is no μ-trace on $\widetilde{\mathcal{P}}$ but there is a μ-trace on $\mathcal{P}$, otherwise player II is the winner.

Definition 9.6 A strategy $\mathcal{S}$ for the second player in the game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ is a partial function giving for a position consisting of two choice nodes $(\widetilde{n}, n)$ and a modal node $\widetilde{m}$ near $\widetilde{n}$ a modal node $\mathcal{S}(\widetilde{m}, n)$ near n. If $(\widetilde{m}, m)$ is a pair of modal nodes and n is a son of m then the strategy gives us a son $\mathcal{S}(\widetilde{m}, n)$ of $\widetilde{m}$. A strategy is called *winning* for II iff it guarantees that player II wins the game no matter what the moves of player I are. This also implies that the strategy is defined for appropriate positions.

We will say that a wide tableau $\mathcal{W}$ is a *consequence* of a wide tableau $\widetilde{\mathcal{W}}$ iff player II has a wining strategy in $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$.

The definition of the game is based on the following intuition about wide tableaux. Wide tableau for a formula describes "operationally" semantics of a formula. In order to satisfy formulas in a choice node n we must provide a state which satisfies the label of one of the modal nodes near n. The sons of a modal node describe the transitions from a hypothetical state satisfying its label. $\langle a\rangle$-sons describe which a-successors are required and the $[a]$-son describes what are general conditions all a-successors must satisfy. In this way tableau of a formula describes all possible models of the formula.

The game is defined so that whenever II has a winning strategy from a position $(\widetilde{n}, n)$ then every model of the label of $\widetilde{n}$, $\widetilde{L}(\widetilde{n})$, is also a model of the label of n, $L(n)$. If $\widetilde{n}$ and n are both choice nodes then a model for $\widetilde{L}(\widetilde{n})$ must satisfy the label of one of the modal nodes near $\widetilde{n}$. Hence for every modal node near $\widetilde{n}$ we must show a modal node near n which label is implied by it. If $\widetilde{n}$, n are modal nodes then every $\langle a\rangle$-son of n describes a state the existence of which is required in order to satisfy $L(n)$. We must show that the existence of such a state is also required by $\widetilde{L}(\widetilde{n})$. The $[a]$-son of n represents general requirements on states reachable by action a imposed by $L(n)$. We must show that they are implied by general requirements in $\widetilde{L}(\widetilde{n})$

Now the following lemma can be proved using exactly the same method as in Lemma 9.2

Lemma 9.7 Suppose that we have a weakly aconjunctive formula α and a disjunctive formula δ such that there is a wide tableau for δ which is a consequence of a wide tableau for α. In this case sequent $\alpha \vdash \delta$ is provable.

Proof
Let $\mathcal{W}_\alpha$ and $\mathcal{W}_\delta$ be wide tableaux for α and δ respectively and let $\mathcal{S}$ be a strategy for player II in the game $\mathcal{G}(\mathcal{W}_\alpha, \mathcal{W}_\delta)$. Instead of having just one mapping $\mathcal{C}_\alpha$ we now define two $\mathcal{C}_\alpha$ and $\mathcal{C}_\delta$. They assign nodes of a the wide tableaux for α and δ respectively to some of the nodes of the thin refutation being constructed. Instead of condition $(*)$ we have now two conditions:

$(*_1)$ $L(n) = L_\alpha(\mathcal{C}_\alpha(n)) \cup \{\neg \bigwedge L_\delta(\mathcal{C}_\delta(n))\}$

$(*_2)$ Position $(\mathcal{C}_\alpha(n), \mathcal{C}_\delta(n))$ is reachable in game $\mathcal{G}(\mathcal{W}_\alpha, \mathcal{W}_\delta)$ when II plays according to $\mathcal{S}$.

The rest of the argument is very similar. □

To finish the completeness proof it is enough to show that there is a wide tableau for $\widehat{\varphi}$ which is a consequence of a wide tableau for $\widehat{\alpha}(\widehat{\varphi})$. First we will need some wide tableaux for $\mu X.\widehat{\alpha}(X)$ and $\widehat{\varphi}$. The following lemma shows how to obtain these.

Lemma 9.8 For a given pair of equivalent tableaux $\mathcal{T}_1$ and $\mathcal{T}_2$ for formulas φ_1 and φ_2 respectively, we can construct wide tableaux $\mathcal{W}_1$ for φ_1 and $\mathcal{W}_2$ for φ_2 such that $\mathcal{W}_2$ is a consequence of $\mathcal{W}_1$ and $\mathcal{W}_1$ is a consequence of $\mathcal{W}_2$.

Proof
As we will need the result only when one of the formulas is a disjunctive formula we will give the proof only for this special case. The general argument is very similar.

Suppose δ is a disjunctive formula and α is an arbitrary formula. Assume that we have two equivalent tableaux $\mathcal{T}_\alpha = \langle T_\alpha, L_\alpha\rangle$ and $\mathcal{T}_\delta = \langle T_\delta, L_\delta\rangle$. Let $\mathcal{E} : \mathcal{T}_\alpha \to \mathcal{T}_\delta$ be an equivalence function. We will construct wide tableaux $\mathcal{W}_\alpha, \mathcal{W}_\delta$ (with labeling functions L^w_α and L^w_δ respectively) and a strategy for player II in the game $\mathcal{G}(\mathcal{W}_\alpha, \mathcal{W}_\delta)$.

Observe that if we happen to construct $\mathcal{W}_\alpha$ and $\mathcal{W}_\delta$ up to two choice nodes (o_α, o_δ) and there is a node $F(o_\alpha)$ of tableau $\mathcal{T}_\alpha$ labeled by the same set as o_α then we can construct a part of $\mathcal{W}_\alpha$ up to the nearest modal nodes which will be exactly the same as the corresponding part of $\mathcal{T}_\alpha$. This is because the same rules are used in both cases. If moreover $L^w_\delta(o_\delta) = L_\delta(\mathcal{E}(F(o_\alpha)))$ then we can construct a part of $\mathcal{W}_\delta$ up to the nearest modal nodes which is exactly the same as $\mathcal{T}_\delta$ from $\mathcal{E}(F(o_\alpha))$. This means that we can use equivalence $\mathcal{E}$ for finding a modal node near o_δ for every modal node near o_α.

Assume now that we have a position consisting of two modal nodes (o_α, o_δ). Assume we have defined $F(o_\alpha)$ to be a modal node of $\mathcal{T}_\alpha$ such that

$(**)$ $L_\alpha(F(o_\alpha)) = L^w_\alpha(o_\alpha)$ and $L_\delta(\mathcal{E}(F(o_\alpha))) = L^w_\delta(o_\delta)$.

First we will find a $\langle a\rangle$-son of o_α for every $\langle a\rangle$-son of o_δ. The idea of how it is done is presented in Figure 8. Let o'_δ be a $\langle a\rangle$-son of o_δ. By condition $(**)$

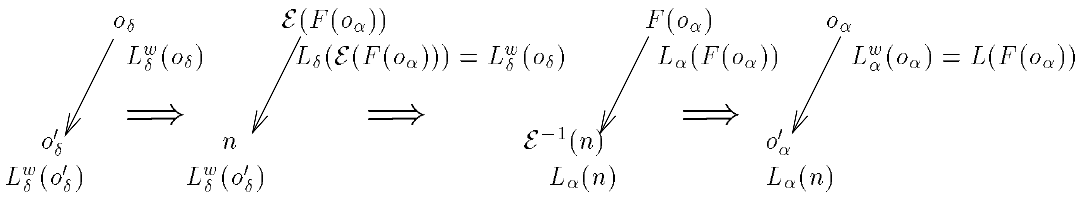

Figure 8: Finding a $\langle a\rangle$-son of o_α for a $\langle a\rangle$-son of o_δ.

there is an a-son n of $\mathcal{E}(F(o_\alpha))$ labeled by $L^w_\delta(o'_\delta)$. Using equivalence $\mathcal{E}^{-1}$ we obtain an a-son $\mathcal{E}^{-1}(n)$ of $F(o_\alpha)$. Once again using $(**)$ we get a $\langle a\rangle$-son o'_α of o_α with the same label as $\mathcal{E}^{-1}(n)$. Defining $F(o'_\alpha) = \mathcal{E}^{-1}(n)$ we get the position consisting of two choice nodes (o'_α, o'_δ) which was considered in the previous paragraph. Hence for every modal node o''_α near o'_α we can find a modal node o''_δ near o'_δ and define $F(o''_\delta)$ so that $(**)$ will be satisfied.

Now let us assume that o'_α is the $[a]$-son of o_α and let o'_δ be the $[a]$-son of o_δ. The label of o'_α is $\{\Phi_\alpha, \ldots, \Phi_i\} = \{\bigvee\Theta : (a \to \Theta) \in L^w_\alpha(o_\alpha)\}$. The label of o'_δ is $\{\bigvee\Psi\}$ as δ is a disjunctive formula.

For any modal node r_α near o'_α we will find a node r_δ near o'_δ so that condition $(**)$ will be satisfied. The idea is presented in Figure 9.

Let us assume that from o'_α we first reduce disjunctions from $\bigvee\Phi_1, \ldots, \bigvee\Phi_i$. Hence on the path to r_α we have a node labeled $\{\varphi_1, \ldots, \varphi_i\}$ where each φ_k is a disjunct form Φ_k. Now let n_α be a a-son of $F_\alpha(o_\alpha)$ labeled by $\{\varphi_1\} \cup \{\Phi_k : k = 2, \ldots, i\}$. There is a modal node m_α near n_α such that on the path from n_α to m_α for every $k = 2, \ldots, i$ formula φ_k is selected from $\bigvee\Phi_k$ and moreover for any trace from φ_k to some formula $\delta \in L^w_\alpha(r_\alpha)$ there is the same trace from φ_k to $\delta \in L_\alpha(m_\alpha)$ and vice versa. Next we take $\mathcal{E}(m_\alpha)$ and take n_δ, the a-son of $\mathcal{E}(F(o_\alpha))$ on the way to $\mathcal{E}(m_\alpha)$. It is labeled by some $\psi \in \Psi$. The $[a]$-son of o_δ is labeled by $\bigvee\Psi$ and there must be a node o''_δ below it labeled by $\{\psi\}$. Now take a modal node r_2 near o''_δ to which leads exactly the same chain of reductions as that form n_δ to $\mathcal{E}(m_\alpha)$. If we define $F(r_\alpha) = m_\alpha$ then we are back in a position when we have a pair of modal nodes (r_α, r_δ) for which $(**)$ holds and we can repeat the procedure.

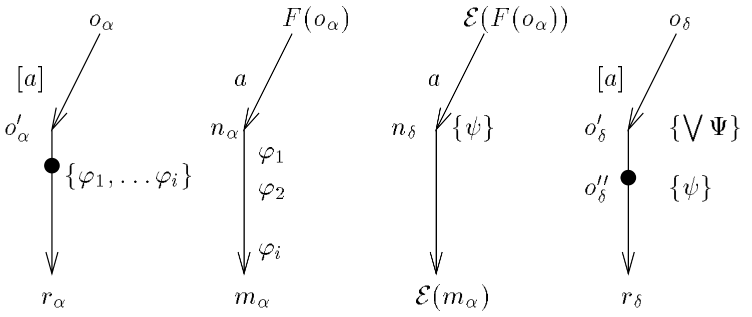

Figure 9: Finding r_δ for r_α.

This describes the strategy for player *II* in the game $\mathcal{G}(\mathcal{W}_\alpha, \mathcal{W}_\delta)$. From the fact that $\mathcal{E}$ is an equivalence it easily follows that this strategy is winning.

As the construction of $\mathcal{W}_\alpha$ and $\mathcal{W}_\delta$ did not depend on the strategy a similar argument shows that $\mathcal{W}_\alpha$ is a consequence of $\mathcal{W}_\delta$. $\square$

Lemma 9.9 Let $\mathcal{W}, \widehat{\mathcal{W}}$ be a pair of wide tableaux for $\mu X.\widehat{\alpha}(X)$ and $\widehat{\varphi}$ respectively constructed from $\mathcal{T}$ and $\widehat{\mathcal{T}}$ as in the Lemma 9.8. There is a wide tableau $\widetilde{\mathcal{W}}$ for $\widehat{\alpha}(\widehat{\varphi})$ of which $\mathcal{W}$ is a consequence.

Proof
Let L and $\widehat{L}$ by labeling functions of $\mathcal{W}$ and $\widehat{\mathcal{W}}$ respectively. Let $\mathcal{S}$ be a winning strategy in the game $\mathcal{G}(\widehat{\mathcal{W}}, \mathcal{W})$. There is a very close match between $\mathcal{W}$ and a wide tableau for $\widehat{\alpha}(\mu X.\widehat{\alpha}(X))$ as the only tableau rule applicable to $\{\mu X.\widehat{\alpha}(X)\}$ is (μ). Let us denote by $\mathcal{W}$ also the wide tableau for $\widehat{\alpha}(\mu X.\widehat{\alpha}(X))$.

By assumption $\widehat{\varphi}$ and $\widehat{\alpha}(X)$ are disjunctive formulas. We will use notation $\beta[\mu X.\widehat{\alpha}(X)/\widehat{\varphi}]$ and $\beta[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]$ to stand for the obvious replacement, it will be always the case that no free variable in $\mu X.\widehat{\alpha}(X)$ or $\widehat{\varphi}$ is bound by the context β.

From Proposition 8.2 we obtain

Observation 9.9.1 For every node $\widetilde{n}$ of $\widetilde{\mathcal{W}}$, every formula in $\widetilde{L}(\widetilde{n})$ will be either a disjunctive formula or of the form $\beta(\widehat{\varphi})$ with $\beta(X)$ being a disjunctive formula.

Recall that $\widehat{\mathcal{T}}$ was constructed from $\mathcal{T}$ using Theorem 5.4. Hence we can assume that conditions described in Observation 5.5.1 are satisfied. Easy analysis of the construction from Lemma 9.8 gives us the following.

Observation 9.9.2 For any two positions $(\widehat{m}, m)$, $(\widehat{n}, n)$ reachable in the game $\mathcal{G}(\widehat{\mathcal{W}}, \mathcal{W})$ when *II* plays according to $\mathcal{S}$ we know that whenever $\widehat{L}(\widehat{m}) = \widehat{L}(\widehat{n})$ then $L(m) = L(n)$. Moreover if $\widehat{m}, \widehat{n}$ are both choice nodes with different labels then the labels of modal nodes near $\widehat{m}$ are different from the labels of modal nodes near $\widehat{n}$.

The above observation will be quite important because it allows us to control contractions which take place. Instead of using this observation we could allow tableaux labeled with multisets.

To define a strategy from a position $(\widetilde{m}, m)$ we will use some additional information about the position. This information will come from the three functions $p(\widetilde{m})$, $nd(\widetilde{m})$ and $\widehat{nd}(\widetilde{m})$. The first function assigns a priority, that is a natural number or ∞, to every formula in $\widetilde{L}(\widetilde{m})$. The functions $nd(\widetilde{m})$ and $\widehat{nd}(\widetilde{m})$ are partial functions which assign nodes of $\mathcal{W}$ and $\widehat{\mathcal{W}}$ respectively to finite priorities from the image of $p(\widetilde{m})$, i.e., from the set $\{q \in \mathcal{N} : p(\widetilde{m})^{-1}(q) \neq \emptyset\}$.

This situation is presented in Figure 10. We have a position $(\widetilde{m}, m)$ in the main play of the game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ and for every natural number in the image of $p(\widetilde{m})$ we have a sub-play of $\mathcal{G}(\widehat{\mathcal{W}}, \mathcal{W})$ in a position given by functions $\widehat{nd}(\widetilde{m})$ and $nd(\widetilde{m})$. To make a move in the main play we consult strategy $\mathcal{S}$ for sub-plays.

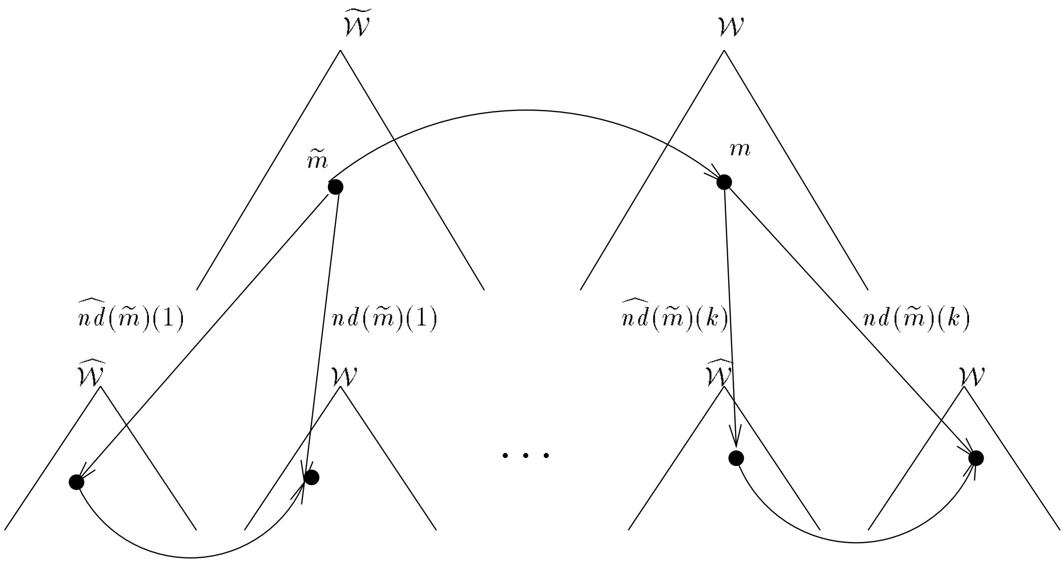

Figure 10: Auxiliary functions

For every considered position $(\widetilde{m}, m)$ we will have three conditions:

(i) for any $q \in \mathcal{N}$ in the image of $p(\widetilde{m})$, $L(\widehat{nd}(\widetilde{m})(q)) \subseteq \widetilde{L}(\widetilde{m})$,

(ii) for any $\psi \in L(m)$, there is q so that $\psi \in L(nd(\widetilde{m})(q))$.

(iii) for any $q \in \mathcal{N}$ in the image of $p(\widetilde{m})$, strategy $\mathcal{S}$ is defined for the position $(\widehat{nd}(\widetilde{m})(q), nd(\widetilde{m})(q))$.

The idea of a strategy in $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ is to consult $\mathcal{S}$ in every step and the above three conditions allow us to do this (see Figure 11). Whenever a position is a pair of modal nodes $(\widetilde{m}, m)$ and I chooses a son of m, this son is designated by some formula $\xi \in L(m)$. By condition (ii) we know that there is a $d \in \mathcal{N}$ and a son of $nd(\widetilde{m})(d)$ designated by ξ. By (iii) we can take an a-son of $\widehat{nd}(\widetilde{m})(d)$

which is a response of $\mathcal{S}$ to choosing this son. By (i) we can find an appropriate son of $\widetilde{m}$. The scheme of the reasoning in case when a position is a pair of choice nodes is similar. In what follows we will have conditions: $I1, I2, I3$ for pairs of modal nodes and conditions: $I1', I2', I3'$ for pairs of choice nodes, which are precise formulation of $(i), (ii), (iii)$ for appropriate cases.

We will describe the strategy for player II in a sequence of observations. First observation shows how we can define auxiliary functions for sons of $\widetilde{m}$, if only auxiliary functions for $\widetilde{m}$ are defined, so that the pair $(\widetilde{m}, m)$ satisfies some conditions which are precise formulation of conditions (i), (ii) and (iii).

Observation 9.9.3 Suppose the game is in a position $(\widetilde{m}, m)$ with both nodes being modal nodes. Let us assume that the following conditions hold:

$I1$ For every $q \in \mathcal{N}$, $p(\widetilde{m})^{-1}(q) \subseteq \widehat{L}(\widehat{nd}(\widetilde{m})(q)) \subseteq \bigcup\{p(\widetilde{m})^{-1}(q') : q' \leq q\}$,

$I2$ $L(m) \subseteq \bigcup_{q \in \mathcal{N}} L(nd(\widetilde{m})(q)) \cup \{\psi[\mu X.\widehat{\alpha}(X)/\widehat{\varphi}] : \psi \in p^{-1}(\widetilde{m})(\infty)\}$

$I3$ for any $q \in \mathcal{N}$ in the image of $p(\widetilde{m})$, strategy $\mathcal{S}$ is defined for the position $(\widehat{nd}(\widetilde{m})(q), nd(\widetilde{m})(q))$.

For every son $\widetilde{n}$ of $\widetilde{m}$ we can define $p(\widetilde{n})$ and $\widehat{nd}(\widetilde{n})$ so that:

$I1'$ For every $q \in \mathcal{N}$ in the image of $p(\widetilde{n})$: $p(\widetilde{n})^{-1}(q) = \widehat{L}(\widehat{nd}(\widetilde{n})(q))$

Proof: For every formula $\alpha \in \widetilde{L}(\widetilde{n})$ there is the least priority formula $\beta \in \widetilde{L}(\widetilde{m})$ which reduction gave us α. We let $p(\widetilde{n})(\alpha) = p(\widetilde{m})(\beta)$.

To define $\widehat{nd}(\widetilde{n})$ we have two cases depending on whether $\widetilde{n}$ is a $[a]$-son or a $\langle a\rangle$-son of $\widetilde{m}$. In the first case, for every q in the image of $p(\widetilde{n})$ we let $\widehat{nd}(\widetilde{n})(q)$ to be the $[a]$-son of $\widehat{nd}(\widetilde{m})(q)$.

In the second case the label of $\widetilde{n}$ is of the form

$$\{\psi\} \cup \{\bigvee \Theta : (a \to \Theta) \in \widetilde{L}(\widetilde{m}), \Theta \neq \Psi\}$$

for some $(a \to \Psi) \in \widetilde{L}(\widetilde{m})$ and $\psi \in \Psi$. Let $d = p(\widetilde{m})((a \to \Psi))$. For every priority $q \neq d$ we let $\widehat{nd}(\widetilde{n})(q)$ to be the $[a]$-son of $\widehat{nd}(\widetilde{m})(q)$ if such exists. If $p(\widetilde{n})(\psi) = d$ we let $\widehat{nd}(\widetilde{n})(d)$ to be the $\langle a\rangle$-son of $\widehat{nd}(\widetilde{m})(d)$ labeled with $\{\psi\}$. $\square$

Let us consider a situation when a position of the game consists of a pair of modal nodes $(\widetilde{m}, m)$ and player I chooses a $[a]$-son.

Observation 9.9.4 Assume the game is in the position $(\widetilde{m}, m)$ as described in Observation 9.9.3, that is, conditions $I1$, $I2$, $I3$ hold and auxiliary functions for all the sons of $\widetilde{m}$ are defined accordingly. Let n be the $[a]$-son of m and $\widetilde{n}$ be the $[a]$-son of $\widetilde{m}$. For the pair $(\widetilde{n}, n)$ we can define $nd(\widetilde{n})$ so that conditions $I1', I2', I3'$ will be satisfied where:

$I2'$ $L(n) \subseteq \bigcup_{q \in \mathcal{N}} L(nd(\widetilde{n})(q)) \cup \{\psi[\mu X.\widehat{\alpha}(X)/\widehat{\varphi}] : \psi \in p^{-1}(\widetilde{n})(\infty)\}$,

$I3'$ For any $q \in \mathcal{N}$ in the image of $p(\widetilde{n})$, $\widehat{nd}(\widetilde{n})(q) = \mathcal{S}(\widehat{nd}(\widetilde{m})(q), nd(\widetilde{n})(q))$.

Proof: Node n is labeled by

$$\{\bigvee\Theta : (a \to \Theta) \in L(m)\}$$

For every $\bigvee\Theta \in L(n)$ by $I2$ either there is the lowest priority q such that $(a \to \Theta) \in L(nd(\widetilde{m})(q))$ or $(a \to \Theta)[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in \widetilde{L}(\widetilde{m})$. In the second case clearly $\bigvee\Theta[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in L(\widetilde{n})$ and $p(\widetilde{n})((a \to \Theta)[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]) = \infty$. Hence $I2'$ and $I3'$ are satisfied.

If $q \in \mathcal{N}$ then we let $nd(\widetilde{n})(q)$ to be the $[a]$-son of $nd(\widetilde{m})(q)$. This establishes $I2'$ and $I3'$ if only q is in the image of $p(\widetilde{n})$.

By $I3$ there is $[a]$-son of $\widehat{nd}(\widetilde{m})(q)$. Let us take the unique Ψ such that $(a \to \Psi) \in \widehat{L}(\widehat{nd}(\widetilde{m})(q))$. We know by $I1$ that $\bigvee\Psi \in \widetilde{L}(\widetilde{n})$. If $p(\widetilde{n})(\bigvee\Psi) = q' < q$ then $(a \to \Psi) \in \widehat{L}(\widehat{nd}(\widetilde{m})(q'))$, hence the $[a]$-sons of $\widehat{nd}(\widetilde{m})(q)$ and $\widehat{nd}(\widetilde{m})(q')$ have the same labels. By Observation 9.9.2 $[a]$-sons of $nd(\widetilde{m})(q)$ and $nd(\widetilde{m})(q')$ have the same labels. This means that $(a \to \Theta) \in L(nd(\widetilde{m})(q'))$, contradiction with the choice of q. $\square$

Observation 9.9.5 Assume the game is in the position $(\widetilde{m}, m)$ as in Observation 9.9.3. For any $\langle a\rangle$-son n of m of we can find $\langle a\rangle$-son $\widetilde{n}$ of $\widetilde{m}$ and define $nd(\widetilde{n})$ so that for obtained pair $(\widetilde{n}, n)$ conditions $I1', I2', I3'$ are satisfied.

Proof: Let n be a son of m labeled by

$$\{\xi\} \cup \{\bigvee\Theta : (a \to \Theta) \in L(m), \Theta \neq \Xi\}$$

for some $(a \to \Xi) \in L(m)$ and $\xi \in \Xi$. We know by $I2$ that either there is the smallest d such that $(a \to \Xi) \in L(nd(\widetilde{m})(d))$ or $(a \to \Xi)[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in \widetilde{L}(\widetilde{m})$. In the second case take a son of $\widetilde{m}$ labeled

$$\{\xi[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]\} \cup \{\bigvee\Theta : (a \to \Theta) \in \widetilde{L}(\widetilde{m}), \Theta \neq \Xi[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]\}$$

and we reason as in the observation above.

The case when $d \in \mathcal{N}$ is presented in Figure 11. We used abbreviations

$$\begin{aligned}
\Gamma_m &= \{\bigvee\Theta : (a \to \Theta) \in L(m), \Theta \neq \Xi\}\\
\Gamma_{nd(\widetilde{m})(d)} &= \{\bigvee\Theta : (a \to \Theta) \in L(nd(\widetilde{m})(d)), \Theta \neq \Xi\}\\
\Gamma_{\widetilde{m}} &= \{\bigvee\Theta : (a \to \Theta) \in \widetilde{L}(\widetilde{m}), \Theta \neq \Psi\}
\end{aligned}$$

Let us describe this picture. By $I2$ there is an $\langle a\rangle$-son n' of $nd(\widetilde{m})(d)$ labeled

$$\{\xi\} \cup \{\bigvee\Theta : (a \to \Theta) \in L(nd(\widetilde{m})(d)), \Theta \neq \Xi\}$$

By $I3$ $\mathcal{S}(\widehat{nd}(\widetilde{m})(q), n')$ is defined. Let $(a \to \Psi) \in \widetilde{L}(\widehat{nd}(\widetilde{m})(d))$ and $\psi \in \Psi$ be such that $\mathcal{S}(\widehat{nd}(\widetilde{m})(q), n')$ is labeled by $\{\psi\}$. By $I1$ we can choose a $\langle a\rangle$-son $\widetilde{n}$ of $\widetilde{m}$ labeled

$$\{\psi\} \cup \{\bigvee\Theta : (a \to \Theta) \in \widetilde{L}(\widetilde{m}), \Theta \neq \Psi\}$$

For every priority $q' \in \mathcal{N}$ in the image of $p(\widetilde{n})$ we let $nd(\widetilde{n})(q')$ to be the $[a]$-son of $nd(\widetilde{m})(q')$ if $q' \neq d$ and let $nd(\widetilde{n})(d) = n'$ if d is in the image. This guarantees condition $I3'$.

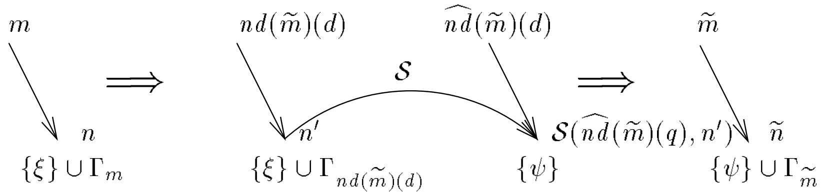

Figure 11: Finding $\widetilde{n}$ for n.

We would like to show condition $I2'$ in particular for $\xi \in L(n)$. Let $q = p(\widetilde{n})(\psi)$. If $q = d$ then $\xi \in L(nd(\widetilde{n})(d))$.

It may happen that $q < d$. This may be because of two reasons, either: (1) $p(\widetilde{m})((a \to \Psi)) = q$ or (2) there is $(a \to \{\psi\}) \in \widetilde{L}(\widetilde{m})$ such that $p(\widetilde{m})((a \to \{\psi\})) = q$ and $\{\psi\} \neq \Psi$.

In the first case we have that $(a \to \Psi) \in \widehat{L}(\widehat{nd}(\widetilde{m})(d))$ and $(a \to \Psi) \in \widehat{L}(\widehat{nd}(\widetilde{m})(q))$. By Observation 9.9.2 this means that $nd(\widetilde{m})(d)$ and $nd(\widetilde{m})(q)$ have the $[a]$-sons with the same labels hence $(a \to \Xi) \in L(nd(\widetilde{m})(q))$, contradiction with the choice of d.

In the second case we know that there is unique $\langle a\rangle$-son of $nd(\widetilde{m})(q)$ and by Observation 9.9.2 it has the same label as n'. In this case it is just $\{\xi\}$ and the $[a]$-son of $nd(\widetilde{m})(q)$ is also labeled by $\{\xi\}$.

This shows that $\xi \in L(nd(\widetilde{n})(q))$ for some q in the image of $p(\widetilde{n})$. The argument for any other $\bigvee \Theta \in L(n)$ is similar as in the previous observation. $\square$

Next observation shows how to define auxiliary functions for a modal node near a choice node.

Observation 9.9.6 Suppose $\widetilde{n}$ is a choice and $p(\widetilde{n})$, $nd(\widetilde{n})$ are defined so that conditions $I1', I3'$ are satisfied. For any modal node $\widetilde{m}_1$ near n we can define $p(\widetilde{m}_1)$, $nd(\widetilde{m}_1)$ and $\widehat{nd}(\widetilde{m}_1)$ so that $I1$ and $I3$ are satisfied and for every $q \in \mathcal{N}$ in the image of $p(\widetilde{m}_1)$, $nd(\widetilde{m}_1)(q) = \mathcal{S}(\widehat{nd}(\widetilde{m}_1)(q), nd(\widetilde{n})(q))$.

Proof: For every formula $\delta \in \widetilde{L}(\widetilde{m}_1)$ there is the smallest priority formula $\gamma \in \widetilde{L}(\widetilde{n})$ from which there is a trace to δ. We let $p(\widetilde{m}_1)(\delta) = p(\widetilde{n})(\gamma)$ with one exception when $p(\widetilde{n})(\gamma) = \infty$ and formula $\widehat{\varphi}$ was reduced on a trace to δ. In this case $p(\widetilde{n})(\delta)$ will be the smallest unused priority q' that is the smallest number not in the image of $p(\widetilde{n})$. This exception is needed to establish $I2$.

Let us define $\widehat{nd}(\widetilde{m}_1)$. If the fresh priority q' was introduced then for $\widehat{nd}(\widetilde{m}_1)(q')$ we take a modal node near the root of $\widehat{\mathcal{W}}$ to which leads the same sequence of reductions as the one applied to $\widehat{\varphi}$ on the trace to δ. For any other priority q in the image of $p(\widetilde{n})$ we know by $I1'$ that there is exactly one formula $\alpha \in \widetilde{L}(\widetilde{n})$ of this priority and $\{\alpha\} = \widehat{L}(\widehat{nd}(\widetilde{n})(q))$. Take a modal node $\widehat{m}_1$ near $\widehat{nd}(\widetilde{n})(q)$ to which leads the same chain of reductions as that applied to α on the path to $\widetilde{m}_1$. Let $\widehat{nd(\widetilde{m}_1)}(q) = \widehat{m}_1$.

We let $nd(\widetilde{m}_1)(q) = \mathcal{S}(\widehat{nd}(\widetilde{m}_1)(q), nd(\widetilde{n})(q))$ for any q for which right hand side is defined. If q' is freshly introduced priority then we let $nd(\widetilde{m}_1)(q') =$

$\mathcal{S}(\widehat{nd}(\widetilde{m}_1)(q'), r)$ where r is the root of $\mathcal{W}$. Clearly this way we satisfy $I3$ and the additional requirement of the observation.

We now check whether condition $I1$ is satisfied. Any formula $\psi \in \widetilde{L}(m_1)$ with priority q has a trace from some formula of priority q in $\widetilde{L}(\widetilde{n})$. Hence $\psi \in \widehat{L}(\widehat{nd}(\widetilde{m}_1)(q))$ by definition of $\widehat{nd}(\widetilde{m}_1)(q)$. For any $\psi \in \widehat{L}(\widehat{nd}(\widetilde{m}_1)(q))$ there is a trace to it from the unique formula in $\widetilde{L}(\widetilde{n})$ with priority q. By definition $\psi \in \widetilde{L}(\widetilde{m}_1)$. It may nevertheless happen that $p(\widetilde{m}_1)(\psi) < q$. $\square$

The last observation considers a situation when we are given a pair of choice nodes $(\widetilde{n}, n)$ so that conditions $I1', I2', I3'$ are satisfied. It will describe a response of player *II* to choosing a modal node near $\widetilde{n}$ by player *I*.

Observation 9.9.7 Let $(\widetilde{n}, n)$ be a pair of choice nodes, let $p(\widetilde{n}), nd(\widetilde{n}), \widehat{nd}(\widetilde{n})$ be defined and let conditions $I1', I2', I3'$ be satisfied. For any modal node $\widetilde{m}_1$ near $\widetilde{n}$ we can find a modal node m_1 near n so that for the pair $(\widetilde{m}_1, m_1)$ conditions $I1, I2, I3$ will be satisfied and traces from $(\widetilde{n}, n)$ to $(\widetilde{m}_1, m_1)$ will be preserved which means that whenever there is a trace from $\alpha \in L(n)$ to $\beta \in L(m_1)$ and X is the smallest variable with respect to $\leq_{\mu X.\widehat{\alpha}(X)}$ ordering which was regenerated on the trace then either:

- $p(n)(\alpha) > p(m_1)(\beta)$ or
- $p(n)(\alpha) = p(m_1)(\beta) = q$ and when $q \in \mathcal{N}$ there is a trace from $\alpha \in L(nd(\widetilde{n})(q))$ to $\beta \in L(nd(\widetilde{m}_1)(q))$ or when $q = \infty$ we have a trace from $\alpha[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in L(\widetilde{n})$ to $\beta[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in L(\widetilde{n}_1)$. In both cases X is the smallest regenerated variable.

Proof: Definitions of $p(\widetilde{m}_1)$, $nd(\widetilde{m}_1)$ and $\widehat{nd}(\widetilde{n})$ are given by Observation 9.9.6. For convenience we extend the definition of $nd(\widetilde{n})$. For the smallest priority q' not in the image of $p(\widetilde{n})$ we let $nd(\widetilde{n})(q')$ to be the root of $\mathcal{W}$

We will construct a path from n to the desired m_1. For n and every considered descendant o of it we will define function $p(o)$. For every $\beta \in L(n)$ condition $I2'$ allows us to define $p(n)(\beta)$ as the smallest priority q for which $\beta \in L(nd(\widetilde{n})(q))$ or ∞ if there is no such q. We will assume that for any considered node o and any $\psi \in L(o)$:

I4' If $p(o)(\psi) = \infty$ then $\psi[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]$ appears on the path from $\widetilde{n}$ to $\widetilde{m}_1$ otherwise $p(o)(\psi) = q \in \mathcal{N}$ and ψ appears on the path from $nd(\widetilde{n})(q)$ to $nd(\widetilde{m}_1)(q)$

Suppose we have constructed the path up to a node o and some formula ψ is reduced in o.

- If ψ is not a disjunction then there is only one son o' of o and let ψ' be the result of reducing ψ. For every formula $\beta \in L(o')$, $\beta \neq \psi'$ we let $p(o')(\beta) = p(o)(\beta)$. If $\psi' \notin L(o)$ then let $p(o')(\psi') = p(o)(\psi)$ otherwise let $p(o')(\psi') = \min\{p(o)(\psi), p(o)(\psi')\}$. One exception to this rule is when $\psi = \mu X.\widehat{\alpha}(X)$ and $p(o)(\psi) = \infty$. In this case the result of reduction is $\widehat{\alpha}(X)$ and we let $p(o)(\widehat{\alpha}(X))$ to be the smallest unused priority q'.

- If $\psi = \alpha \vee \beta$ then o has two sons o_1, o_2 and we have to choose one of them. If $p(o)(\psi) = \infty$ then $\psi[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)]$ is on the path from $\widetilde{n}$ to $\widetilde{m}_1$ otherwise ψ appears on the path from $nd(\widetilde{n})(p(o)(\psi))$ to $nd(\widetilde{m}_1)(p(o)(\psi))$. We choose a son of o with the same disjunct as the one appearing on the appropriate path. For chosen o' we define $p(o')$ as in case of unary rule. It should be easy to check that for so defined o' and $p(o')$ condition $I4'$ holds.

Repeating this procedure we arrive at a modal node m_1 near n. We know that conditions $I1$ and $I3$ hold. Let us now check condition $I2$. Suppose $\psi \in L(m_1)$ and $q = p(m_1)(\psi)$. Because m_1 is a modal node ψ may be reducible only by application of (mod) rule. By $I4'$ if $q = \infty$ then $\psi[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in \widetilde{L}(\widetilde{m}_1)$ otherwise $q \in \mathcal{N}$ and $\psi \in L(nd(\widetilde{m}_1)(q))$. There is one subtle point here. We have defined $nd(\widetilde{m}_1)(q)$ for all priorities in the image of $p(\widetilde{n})$ not paying attention to the fact that there may be no formula of a given priority in $L(\widetilde{m}_1)$. Fortunately by Observation 9.9.2 we may assume that for any priority q in the image of $p(\widetilde{n})$ there is a formula of priority q in $L(\widetilde{m}_1)$

Finally it is easy to see that if in our procedure we choose formulas appropriately then trace preservation will be guaranteed. □

To describe the strategy for the player *II* completely it remains to define auxiliary functions for the root $\widetilde{r}_0$ of $\widetilde{\mathcal{W}}$. Node $\widetilde{r}_0$ is labeled by $\{\widehat{\alpha}(\widehat{\varphi})\}$ and we let $p(\widetilde{r}_0)(\widehat{\alpha}(\widehat{\varphi})) = \infty$. Functions $\widehat{nd}(\widetilde{r}_0)$ and $nd(\widetilde{r}_0)$ are totally undefined.

We must show that we have defined a winning strategy for player *II* in the game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$. Assume conversely that there is a way for *I* to win against our just described strategy. We will show that *I* can win against $\mathcal{S}$ in the game $\mathcal{G}(\widehat{\mathcal{W}}, \mathcal{W})$.

Let us examine the play in the game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ where *II* plays according to the strategy we have just defined and *I* wins. For a position $(\widetilde{m}, m)$ of the play we will have a sub-play of the game $\mathcal{G}(\widehat{\mathcal{W}}, \mathcal{W})$ for every natural number in the image of $p(\widetilde{m})$. For such a number q the corresponding sub-play will be in the position $(\widehat{nd}(\widetilde{n})(q), nd(\widetilde{n})(q))$.

Game $\mathcal{G}(\widetilde{\mathcal{W}}, \mathcal{W})$ starts in the roots of both tableaux and as there is only one formula and it has priority ∞, we have no sub-plays.

Suppose the play is in a position $(\widetilde{m}, m)$ consisting of two modal nodes. Now player *I* chooses a son n of m and *II* responds with the son $\widetilde{n}$ of $\widetilde{m}$ of the same type according to Observations 9.9.4 and 9.9.5. For every priority $q \in \mathcal{N}$ in the image of $p(\widetilde{n})$ we have by $I3'$ that $\widehat{nd}(\widetilde{n})(q) = \mathcal{S}(\widehat{nd}(\widetilde{m})(q), nd(\widetilde{n})(q))$, so each sub-play also advances by one step. No new priorities are introduced. Some sub-plays may be abandoned because corresponding priorities have disappeared.

When the play is in a position $(\widetilde{n}, n)$ consisting of two choice nodes then player *I* chooses some modal node $\widetilde{m}$ near $\widetilde{n}$ and *II* responds with a modal node m near n according to Observation 9.9.7. For every priority $q \in \mathcal{N}$ in the image of $p(\widetilde{m})$ we have by Observation 9.9.6 that $nd(\widetilde{m})(q) = \mathcal{S}(\widehat{nd}(\widetilde{m})(q), nd(\widetilde{n})(q))$. New priority may be introduced at this step and this tigers a new sub-play.

Now suppose that *I* won. The above observations show that player *II* can always respond to a move of player *I* if *II* plays according to the strategy. Hence the play was infinite and we have as the result two paths: $\widetilde{\mathcal{P}}$ of $\widetilde{\mathcal{W}}$ and $\mathcal{P}$ of $\mathcal{W}$.

Because I won there is a μ-trace T on $\mathcal{P}$ and there is no μ-trace on $\widetilde{\mathcal{P}}$.

Condition $I2'$ allows us to define for every choice node n of $\mathcal{P}$ a priority $p(n)(T(n))$. By definition of the strategy this priority can only decrease so from some point, say node n_0, it must be constant, say equal q. Let $n_0, n_1, \ldots$ be successive choice nodes on $\mathcal{P}$ and $\widetilde{n}_0, \widetilde{n}_1, \ldots$ corresponding choice nodes of $\widetilde{\mathcal{P}}$. For any $i \in \mathcal{N}$ we have by $I2'$

$$T(n_i)[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in \widetilde{L}(\widetilde{n}_i), \quad T(n_{i+1})[\widehat{\varphi}/\mu X.\widehat{\alpha}(X)] \in \widetilde{L}(\widetilde{n}_{i+1}) \quad \text{if } q = \infty \tag{2}$$

and

$$T(n_i) \in L(nd(\widetilde{n}_i)(q)), \quad T(n_{i+1}) \in L(nd(\widetilde{n}_{i+1})(q)) \quad \text{if } q \in \mathcal{N} \tag{3}$$

By trace preservation, in both cases the smallest variable with respect to $\leq_{\mu X.\widehat{\alpha}(X)}$ ordering regenerated on the trace is the same as the smallest variable regenerated between $T(n_i) \in L(n_i)$ and $T(n_{i+1}) \in L(n_{i+1})$.

In case $q = \infty$ this considerations show that there is a μ-trace on $\widetilde{\mathcal{P}}$.

In case $q \in \mathcal{N}$ we know by (3) that there is a μ-trace on the path $\mathcal{P}'$ of $\mathcal{W}$ designated by $\{nd(\widetilde{n}_i)(q) : i = 0, 1, \ldots\}$. On the other hand by $I1'$, $\widehat{L}(\widehat{nd}(\widetilde{n}_i)(q)) \subseteq \widetilde{L}(\widetilde{m})$ and as there is no μ-trace on $\widetilde{\mathcal{P}}$ the unique trace on the path $\widehat{\mathcal{P}}$ designated by $\{\widehat{nd}(\widetilde{n}_i)(q) : i \in \mathcal{N}\}$ is not a μ-trace. But a pair of paths $\widehat{\mathcal{P}}$ and $\mathcal{P}'$ is the result of the sub-play for priority q. This is a contradiction because we have assumed that $\mathcal{S}$ is winning.

□

Summarising the case of the proof for $\varphi = \mu X.\alpha(X)$. By induction assumption we have a disjunctive formula $\widehat{\alpha}(X)$ equivalent to $\alpha(X)$ and know that $\alpha(X) \vdash \widehat{\alpha}(X)$ is provable. By Theorem 5.4 we obtain a disjunctive formula $\widehat{\varphi}$ which has a tableau $\widehat{\mathcal{T}}$ equivalent to some regular tableau $\mathcal{T}$ for $\mu X.\widehat{\alpha}(X)$. By Theorem 4.10, formula $\widehat{\varphi}$ is equivalent to φ. By Lemma 9.8 there are wide tableaux: $\mathcal{W}$ for $\mu X.\widehat{\alpha}(X)$ and $\widehat{\mathcal{W}}$ for $\widehat{\varphi}$ such that $\mathcal{W}$ is a consequence of $\widehat{\mathcal{W}}$ and vice versa. By Lemma 9.9 $\mathcal{W}$ is a consequence of $\widetilde{\mathcal{W}}$, a wide tableau for $\widehat{\alpha}(\widehat{\varphi})$. Hence, as the consequence relation is transitive, $\widehat{\mathcal{W}}$ is a consequence of $\widetilde{\mathcal{W}}$. Now by Proposition 8.3 $\widehat{\alpha}(\widehat{\varphi})$ is a weakly aconjunctive formula and $\widehat{\varphi}$ is by definition a disjunctive formula. By Lemma 9.7 the sequent $\widehat{\alpha}(\widehat{\varphi}) \vdash \widehat{\varphi}$ is provable. Then $\mu X.\widehat{\alpha}(X) \vdash \widehat{\varphi}$ is provable by rule (P) and $\varphi \vdash \mu X.\widehat{\alpha}(X)$ is provable by induction assumption, hence the sequent $\varphi \vdash \widehat{\varphi}$ is provable.

This completes the proof of Theorem 9.1 and the proof of the completeness of Kozen's axiomatisation for the μ-calculus.

References

[1] O. Bernholtz and O. Grumberg. Branching time temporal logics and amorphous tree automata. In *Proc. 4th Conference on Concurency Theory*, volume 715 of *LNCS*, pages 262–277. Springer-Verlag, 1993.

[2] E. A. Emerson, C.S. Jutla, and A.P. Sistla. On model-checking for fragments of μ-calculus. In *CAV'93*, volume 697 of *LNCS*, pages 385–396, 1993.

[3] E. Allen Emerson. Automata, tableaux and temporal logic. In *Colledge Conference on Logic of Programs*, volume 193 of *LNCS*. Springer-Verlag, 1985.

[4] 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.

[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] R. Hossley and C. Rackoff. The emptiness problem for automata on infinite trees. In *13th IEEE Symp. on Switching and Automata Theory*, pages 121–124, 1972.

[8] David Janin and Igor Walukiewicz. Automata for the μ-calculus and related results. To appear.

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

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

[11] R. McNaughton. Testing and generating infinte sequences by a finite automaton. *Information & Control*, 9:521–530, 1966.

[12] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automta. In A. Skowron, editor, *Fith Symposium on Computation Theory*, volume 208 of *LNCS*, pages 157–168, 1984.

[13] D.E. Muller and P.E.Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.

[14] D. Niwiński and I. Walukiewicz. Games for μ-calculus. Technical Report TR 94-03(192), Institute of Informatics, Warsaw Univeristy, February 1994. To appear in TCS.

[15] Damian Niwiński. Fixed points vs. infinite generation. In *Proc. 3rd. IEEE LICS*, pages 402–409, 1988.

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

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

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

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

[20] Igor Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. To appear.

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

Recent Publications in the BRICS Notes Series

NS-95-1 **Igor Walukiewicz.** ***Notes on the Propositional μ-calculus: Completeness and Related Results*. February 1995. 54 pp.**

NS-94-6 **Uffe H. Engberg, Kim G. Larsen, and Peter D. Mosses, editors.** ***Proceedings of the 6th Nordic Workshop on Programming Theory* (Aarhus, Denmark, 17–19 October, 1994), December 1994. v+483pp.**

NS-94-5 **Andrew M. Pitts.** ***Some Notes on Inductive and Co-Inductive Techniques in the Semantics of Functional Programs,* DRAFT VERSION. December 1994. vi+135 pp.**

NS-94-4 **Peter D. Mosses, editor.** ***Abstracts of the 6th Nordic Workshop on PROGRAMMING THEORY* (Aarhus, Denmark, 17–19 October, 1994), October 1994. v+52 pp.**

NS-94-3 **Sven Skyum, editor.** ***Complexity Theory: Present and Future* (Aarhus, Denmark, 15–18 August, 1994), September 1994. v+213 pp.**

NS-94-2 **David Basin.** ***Induction Based on Rippling and Proof Planning. Mini-Course*. August 1994. 62 pp.**

NS-94-1 **Peter D. Mosses, editor.** ***Proc. 1st International Workshop on Action Semantics* (Edinburgh, 14 April, 1994), May 1994. 145 pp.**