# Completeness of Kozen's Axiomatization for the Modal $\mu$-Calculus: A Simple Proof

Kuniaki Tamura
15-9-103, Takasago 3-chome, Katsushika, Tokyo 125-0054, Japan
E-mail: kuniaki.tamura@gmail.com

October 20, 2020



**Abstract**

The modal $\mu$-calculus, introduced by Dexter Kozen, is an extension of modal logic with fixpoint operators. Its axiomatization, Koz, was introduced at the same time and is an extension of the minimal modal logic K with the so-called Park fixpoint induction principle. It took more than a decade for the completeness of Koz to be proven, finally achieved by Igor Walukiewicz. However, his proof is fairly involved.

In this article, we present an improved proof for the completeness of Koz which, although similar to the original, is simpler and easier to understand.

**Keywords:** The modal $\mu$-calculus, completeness, $\omega$-automata.

## 1 Introduction

The *modal $\mu$-calculus* originated with Scott and De Bakker [11] and was further developed by Dexter Kozen [5] into the main version currently used. It is used to describe and verify properties of labeled transition systems (Kripke models). Many modal and temporal logics can be encoded into the modal $\mu$-calculus, including $\mathsf{CTL}^*$ and its widely used fragments – the linear temporal logic LTL and the computational tree logic CTL. The modal $\mu$-calculus also provides one of the strongest examples of the connections between modal and temporal logics, automata theory and game theory (for example, see [6]). As such, the modal $\mu$-calculus is a very active research area in both theoretical and practical computer science. We refer the reader to Bradfield and Stirling's tutorial article [9] for a thorough introduction to this formal system.

The difference between the modal $\mu$-calculus and modal logic is that the former has the *least fixpoint operator* $\mu$ and the *greatest fixpoint operator* $\nu$ which represent the least and greatest fixpoint solution to the equation $\alpha(x) = x$, where $\alpha(x)$ is a monotonic function mapping some power set of possible worlds into itself.[1] In Kozen's initial work [5], he proposed an axiomatization Koz, which was an extension of the minimal modal logic K with a further axiom and inference rule – the so-called Park fixpoint induction principle:

$$\frac{}{\alpha(\mu x.\alpha(x)) \vdash \mu x.\alpha(x)}\ (\mathsf{Prefix}) \qquad \frac{\alpha(\beta) \vdash \beta}{\mu x.\alpha(x) \vdash \beta}\ (\mathsf{Ind})$$

The system Koz is very simple and natural; nevertheless, Kozen himself could not prove completeness for the full language, but only for the negations of formulas of a special kind called the *aconjunctive formula*. Completeness for the full language turned out to be a knotty problem and remained open for more than a decade. Finally, Walukiewicz [8] solved this problem positively, but his proof is quite involved.[2] The aim of this article is to provide an improved proof that is easier to understand. First, we outline Walukiewicz's proof and explain its difficulties, and then present our improvement.

The completeness theorem considered here is sometimes called weak completeness and requires that the validity follows the provability; that is:

[1] In the modal $\mu$-calculus, the term *state* is preferred to *possible world* since it originated in the area of verification of computer systems. However, we do not use this terminology since it is reserved for *state of automata* in this article.

[2] The difficulties of the proof have been pointed out, e.g., see [2, 7, 9, 12, 13]

**(a)** For any formula $\varphi$, if $\varphi$ is not satisfiable, then $\sim\varphi$ is provable in $\mathsf{Koz}$.

Here, $\sim\varphi$ denotes the negation of $\varphi$. Note that strong completeness cannot be applied to the modal $\mu$-calculus since it lacks compactness. The first step of the proof is based on the results of Janin and Walukiewicz [4], in which they introduced the class of formulas called *automaton normal form*,[3] and showed the following two theorems:

**(b)** For any formula $\varphi$, we can construct an automaton normal form $\mathsf{anf}(\varphi)$ which is semantically equivalent to $\varphi$.

**(c)** For any automaton normal form $\widehat{\varphi}$, if $\widehat{\varphi}$ is not satisfiable, then $\sim\widehat{\varphi}$ is provable in $\mathsf{Koz}$; that is, $\mathsf{Koz}$ is complete for the negations of the automaton normal form.

The above theorems lead to the following Claim (d) for proving:

**(d)** For any formula $\varphi$, there exists a semantically equivalent automaton normal form $\widehat{\varphi}$ such that $\varphi \to \widehat{\varphi}$ is provable in $\mathsf{Koz}$.

Indeed, for any unsatisfiable formula $\varphi$, Claim (d) tells us that $\sim\widehat{\varphi} \to \sim\varphi$ is provable; on the other hand, from Theorem (c) we obtain that $\sim\widehat{\varphi}$ is provable; therefore $\sim\varphi$ is provable in $\mathsf{Koz}$ as required. Hence, our target (a) is reduced to Claim (d).

Another important tool is the concept of a *tableau*, which is a tree structure that is labeled by some subformulas of the primary formula $\varphi$ and is related to the satisfiability problem for $\varphi$. Niwinski and Walukiewicz [3] introduced a game played by two adversaries on a tableau and, by analyzing these games, showed that:

**(e)** For any unsatisfiable formula $\varphi$, there exists a structure called the *refutation* for $\varphi$ which is a substructure of tableau.

Importantly, a refutation for $\varphi$ is very similar to a proof diagram for $\varphi$; roughly speaking, the difference between them is that the former can have infinite branches while the latter can not. Walukiewicz shows that if the refutation for $\varphi$ satisfies a special *thin* condition, it can be transformed into a proof diagram for $\varphi$. In other words,

**(f)** For any unsatisfiable formula $\varphi$ such that there exists a thin refutation for $\varphi$, $\sim\varphi$ is provable in $\mathsf{Koz}$.

Note that Claim (f) is a slight generalization of the completeness for the negations of the aconjunctive formula in the sense that the refutation for an unsatisfiable aconjunctive formula is always thin, and Claim (f) can be shown by the same method as Kozen's original argument.

The proof is based on confirming Claim (d) by induction on the length of $\varphi$, using (b) and (f). The hardest step of induction is the case $\varphi = \mu x.\alpha(x)$. Suppose $\varphi = \mu x.\alpha(x)$ and that we could assume, by inductive hypothesis, $\alpha(x) \to \widehat{\alpha}(x)$ is provable in $\mathsf{Koz}$ where $\widehat{\alpha}(x)$ is an automaton normal form equivalent to $\alpha(x)$. For the inductive step, we want to discover an automaton normal form $\widehat{\varphi}$ equivalent to $\mu x.\alpha(x)$ such that $\mu x.\alpha(x) \to \widehat{\varphi}$ is provable. Note that since $\alpha(x) \to \widehat{\alpha}(x)$ is provable, $\mu x.\alpha(x) \to \mu x.\widehat{\alpha}(x)$ is also provable. Furthermore, $\mu x.\alpha(x)$ and $\mu x.\widehat{\alpha}(x)$ are equivalent to each other. Set $\widehat{\varphi} := \mathsf{anf}(\mu x.\widehat{\alpha}(x))$. Then, it is sufficient to show that $\mu x.\widehat{\alpha}(x) \to \widehat{\varphi}$ is provable, and thus, from the induction rule ($\mathsf{Ind}$), $\widehat{\alpha}(\widehat{\varphi}) \to \widehat{\varphi}$ is provable. To show this, Walukiewicz developed a new utility called *tableau consequence*, which is a binary relation on the tableau and is characterized using game theoretical notations. The following two facts were then shown:

**(g)** Let $\widehat{\alpha}(x)$ and $\widehat{\varphi}$ be formulas denoted above. Then the tableau for $\widehat{\varphi}$ is a consequence of the tableau for $\widehat{\alpha}(\widehat{\varphi})$.

**(h)** For any automaton normal forms $\widehat{\beta}(y)$ and $\widehat{\psi}$, if the tableau for $\widehat{\psi}$ is a consequence of the tableau for $\widehat{\beta}(\widehat{\psi})$, then we can construct a thin refutation for $\sim(\widehat{\beta}(\widehat{\psi}) \to \widehat{\psi})$.[4]

[3] In the original article [4], this class of formulas was called the *disjunctive formula*; however, the term *automaton normal form* is the currently used terminology, to the author's knowledge.

[4] More precisely, this assertion must be stated more generally to be applicable in other cases of an inductive step, see Lemma 5.13.

The real difficulty appeared when proving Claim (g). To establish this claim, Walukiewicz introduced complicated functions across some tableaux and analyzed the properties of these functions very carefully. Finally, Claims (f), (g) and (h) together immediately establish that $\widehat{\alpha}(\widehat{\varphi}) \to \widehat{\varphi}$ is provable in $\mathsf{Koz}$. Thus, he obtained a proof for Claim (d), confirming completeness. The following figure summarizes the Walukiewicz's proof strategy described above.

$$
\begin{tikzcd}
(b) \arrow[r, Rightarrow] & (g) \arrow[dr, Rightarrow] & (c) \arrow[dr, Rightarrow] & \\
& (h) \arrow[r, Rightarrow] & (d) \arrow[r, Rightarrow] & (a) \\
(e) \arrow[r, Rightarrow] & (f) \arrow[ur, Rightarrow] & &
\end{tikzcd}
$$

Figure 1: The outline of the Walukiewicz's proof.

This article's main contribution is the simplification of the proof of Claim (g) and (f). For this purpose, we will apply the $\omega$-automaton conversion method introduced by Safra [14] and Křetìnsỳ et al. [10]. It is shown that the proof of claim (g) and (f) are much more visible by using the mechanism called *index appearance record* provided by those automata. In addition, we make improvements to some terms and concepts. For example, the concept of the tableau consequence will be redefined as a concept similar to the concept of *bisimulation* (instead of the game theoretical notations), which is one of the most fundamental and standard notions in the model theory of modal and its extensional logics. As a result, the proof of (h) is a little easier to understand. Consequently, although our proof of completeness does not include any innovative concepts, it is far more concise than the original proof.

The author hopes that the method given in this article may assist investigation of the modal $\mu$-calculus and related topics.

## 1.1 Outline of the article

The remainder of this article is organized as follows: in the following subsection 1.2, we will define some terminologies used within the article. Section 2 gives basic definitions of the syntax and semantics of the modal $\mu$-calculus. Section 3 introduce well known results concerning $\omega$-automata. The automaton mechanism used in the main proof will be introduced in this section. Section 4 is an application of Section 3. We will prove claims (b) and (f) using the theory of $\omega$-automata. Section 5 is the final section and contains the principle part of this article – the proof of Claim (g). Finaly, we prove the completeness of $\mathsf{Koz}$ by showing Claim (d).

## 1.2 Notation

**Sets:** Let $X$ be an arbitrary set. The *cardinality of* $X$ is denoted $|X|$. The *power set of* $X$ is denoted $\mathcal{P}(X)$. $\omega$ denotes the set of natural numbers.

**Sequences:** A finite sequence over some set $X$ is a function $\pi : \{1, \ldots, n\} \to X$ where $1 \leq n$. An infinite sequence over $X$ is a function $\pi : \omega \setminus \{0\} \to X$. Here, a sequence can refer to either a finite or infinite sequence. The length of a sequence $\pi$ is denoted $|\pi|$. Let $\pi$ be a sequence over $X$. The set of $x \in X$ which appears infinitely often in $\pi$ is denoted $\mathsf{Infinite}(\pi)$. We denote the $n$-th element in $\pi$ by $\pi[n]$ and the fragment of $\pi$ from the $n$-th element to the $m$-th element by $\pi[n, m]$. For example, if $\pi = \mathsf{aabbcddd}$, then $\pi[5] = \mathsf{c}$ and $\pi[2, 6] = \mathsf{abbcd}$. Note that when $\pi$ is a finite non-empty sequence, $\pi[|\pi|]$ denotes the tail of $\pi$.

**Alphabets:** Suppose that $\Sigma$ is a non-empty finite set. Then we may call $\Sigma$ an *alphabet* and its element $a \in \Sigma$ a *letter*. We denote the set of finite sequences over $\Sigma$ by $\Sigma^*$, the set of non-empty finite sequences over $\Sigma$ by $\Sigma^+$, and the set of infinite sequences over $\Sigma$ by $\Sigma^\omega$. As usual, we call an element of $\Sigma^*$ a *word*, an element of $\Sigma^\omega$ an $\omega$-*word*, a set of finite words $\mathcal{L} \subseteq \Sigma^*$ a *language* and, a set of $\omega$-words $\mathcal{L}' \subseteq \Sigma^\omega$ an $\omega$-*language*. The notion of the *factor* on words is defined as usual: for two words $u, v \in \Sigma^* \cup \Sigma^\omega$, $u$ is a factor of $v$ if $v = xuy$ for some $x, y \in \Sigma^* \cup \Sigma^\omega$.

**Graphs:** In this article, the term *graph* refers to a directed graph. That is, a graph is a pair $\mathcal{G} = (V, E)$ where $V$ is an arbitrary set of *vertices* and $E$ is an arbitrary binary relation over $V$, i.e., $E \subseteq V \times V$. A vertex $u$ is said to be an $E$-successor (or simply a successor) of a vertex $v$ in $\mathcal{G}$ if $(v, u) \in E$. For any vertex $v$, we denote the set of all $E$-successors of $v$ by $E(v)$. The sequence $\pi \in V^* \cup V^\omega$ is called an *$E$-sequence* if $\pi[n+1] \in E(\pi[n])$ for any $n < |\pi|$. $E^*$ denotes the reflexive transitive closure of $E$ and $E^+$ denotes the transitive closure of $E$.

**Trees:** The term *tree* is used to mean a *rooted direct tree*. More precisely, a tree is a triple $\mathcal{T} = (T, C, r)$ where $T$ is a set of *nodes*, $r \in T$ is a *root* of the tree and, $C$ is a *child relation*, i.e., $C \subseteq T \times T$ such that for any $t \in T \setminus \{r\}$, there is exactly one $C$-sequence starting at $r$ and ending at $t$. An unique $C$-sequence that starts at $r$ and ends at $t$ is denoted by $\vec{rt}$. As usual, we say that $u$ is a child of $t$ (or $t$ is a parent of $u$) if $(t, u) \in C$. A node $t \in T$ is a *leaf* if $C(t) = \emptyset$. A *branch* of $\mathcal{T}$ is either a finite $C$-sequence starting at $r$ and ending at a leaf or an infinite $C$-sequence starting at $r$.

**Unwinding:** Let $\mathcal{G} = (V, E)$ be a graph. An *unwinding* of $\mathcal{G}$ on $v \in V$ is the tree structure $\mathsf{UNW}_v(\mathcal{G}) = (T, C, r)$ where:

- $T$ consists of all finite non-empty $E$-sequences that start at $v$,
- $(\pi, \pi') \in C$ if and only if; $|\pi| + 1 = |\pi'|$, $\pi = \pi'[1, |\pi|]$ and $(\pi[|\pi|], \pi'[|\pi'|]) \in E$, and
- $r := v$.

This concept can be extended naturally into a graph with some additional relations or functions. For example, let $\mathcal{S} = (V, E, f)$ be a structure where $\mathcal{G} = (V, E)$ is a graph and $f$ is a function with domain $V$. Then we define $\mathsf{UNW}_v(\mathcal{S}) := (\mathsf{UNW}_v(\mathcal{G}), f')$ as $f'(\pi) := f(\pi[|\pi|])$ for any $\pi \in V^+$. Note that we use the same symbol $f$ instead of $f'$ in $\mathsf{UNW}_v(\mathcal{S})$ if there is no danger of confusion.

**Functions:** Let $f$ be a function from some set $X$ to some set $Y$. We define the new function $\vec{f}$ from $X^+ \cup X^\omega$ to $Y^+ \cup Y^\omega$ as:

$$\vec{f}(\pi) := f(\pi[1])f(\pi[2]) \cdots$$

where $\pi \in X^+ \cup X^\omega$. It is obvious that for any $\pi \in X^+ \cup X^\omega$, we have $|\pi| = |\vec{f}(\pi)|$.

# 2 The modal $\mu$-calculus

We will now introduce the syntax, semantics and axiomatization $\mathsf{Koz}$ of the modal $\mu$-calculus, and then present some additional concepts and results for use in the following sections.

## 2.1 Syntax

**Definition 2.1 (Formula).** Let $\mathsf{Prop} = \{p, q, r, x, y, z, \dots\}$ be an infinite countable set of *propositional variables*. Then the collection of the *modal $\mu$-formulas* is defined as follows:

$$\varphi ::= (\top), (\bot), (p) \mid (\neg p) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\Diamond \varphi) \mid (\Box \varphi) \mid (\mu x.\varphi) \mid (\nu x.\varphi)$$

where $p, x \in \mathsf{Prop}$. Moreover, for formulas of the form $(\eta x.\varphi)$ with $\eta \in \{\mu, \nu\}$, we require that each occurrence of $x$ in $\varphi$ is positive; that is, $\neg x$ is not a subformula of $\varphi$. Henceforth in this article, we will use $\eta$ to denote $\mu$ or $\nu$. A formula of the form $p$ or $\neg p$ for $p \in \mathsf{Prop}$, $\top$ and $\bot$ is called *literal*. We use the term $\mathsf{Lit}$ to refer to the set of all literals, i.e., $\mathsf{Lit} := \{p, \neg p, \bot, \top \mid p \in \mathsf{Prop}\}$. We call $\mu$ and $\nu$ *the least fixpoint operator* and *the greatest fixpoint operator*, respectively.

**Remark 2.2.** In Definition 2.1, we confined the formula to a *negation normal form*; that is, the negation symbol may only be applied to propositional variables. However, this restriction can be inconvenient, and so we extend the concept of the negation to an arbitrary formula $\varphi$ (denoted by $\sim\varphi$) inductively as follows:

- $\sim\top := \bot$, $\sim\bot := \top$.
- $\sim p := \neg p$, $\sim\neg p := p$ for $p \in \mathsf{Prop}$.
- $\sim(\varphi \vee \psi) := ((\sim\varphi) \wedge (\sim\psi))$, $\sim(\varphi \wedge \psi) := ((\sim\varphi) \vee (\sim\psi))$.

- $\sim(\Diamond\varphi) := (\Box(\sim\varphi))$, $\sim(\Box\varphi) := (\Diamond(\sim\varphi))$.
- $\sim(\mu x.\varphi(x)) := (\nu x.(\sim\varphi(\neg x)))$, $\sim(\nu x.\varphi(x)) := (\mu x.(\sim\varphi(\neg x)))$.

We introduce *implication* $(\varphi \to \psi)$ as $((\sim\varphi) \vee \psi)$ and *equivalence* $(\varphi \leftrightarrow \psi)$ as $((\varphi \to \psi) \wedge (\psi \to \varphi))$ as per the usual notation. To minimize the use of parentheses, we assume the following precedence of operators from highest to lowest: $\neg$, $\sim$, $\Diamond$, $\Box$, $\eta x$, $\vee$, $\wedge$, $\to$ and $\leftrightarrow$. Moreover, we often abbreviate the outermost parentheses. For example, we write $\Diamond p \to q$ for $((\Diamond p) \to q)$ but not for $(\Diamond(p \to q))$.

As fixpoint operators $\mu$ and $\nu$ can be viewed as quantifiers, we use the standard terminology and notations for quantifiers. We denote the set of all propositional variables appearing free in $\varphi$ by $\mathsf{Free}(\varphi)$, and those appearing bound by $\mathsf{Bound}(\varphi)$. If $\psi$ is a subformula of $\varphi$, we write $\psi \leq \varphi$. We write $\psi < \varphi$ when $\psi$ is a proper subformula. $\mathsf{Sub}(\varphi)$ is the set of all subformulas of $\varphi$ and $\mathsf{Lit}(\varphi)$ denotes the set of all literals which are subformulas of $\varphi$. Let $\varphi(x)$ and $\psi$ be two formulas. The *substitution* of all free appearances of $x$ with $\psi$ into $\varphi$ is denoted $\varphi(x)[x/\psi]$ or sometimes simply $\varphi(\psi)$. As with predicate logic, we prohibit substitution when a new binding relation will occur by that substitution.

The following two definitions regarding formulas will be used frequently in the remainder of the article.

**Definition 2.3 (Well-named formula).** The set of *well-named formulas* $\mathsf{WNF}$ is defined inductively as follows:

1. $\mathsf{Lit} \subseteq \mathsf{WNF}$.
2. Let $\alpha, \beta \in \mathsf{WNF}$ where $\mathsf{Bound}(\alpha) \cap \mathsf{Free}(\beta) = \emptyset$ and $\mathsf{Free}(\alpha) \cap \mathsf{Bound}(\beta) = \emptyset$. Then $\alpha \vee \beta, \alpha \wedge \beta \in \mathsf{WNF}$.
3. Let $\alpha \in \mathsf{WNF}$. Then $\Diamond\alpha, \Box\alpha \in \mathsf{WNF}$.
4. Let $\alpha(x) \in \mathsf{WNF}$ where $x \in \mathsf{Free}(\alpha(x))$ occurs at once, positively, moreover, $x$ is in the scope of some modal operators. Then $\eta x.\alpha(x) \in \mathsf{WNF}$.

If $\varphi$ is well-named and $x$ is bounded in $\varphi$, then there is exactly one subformula which binds $x$; this formula is denoted $\eta_x x.\varphi_x(x)$.

**Definition 2.4 (Alternation depth).** Given a formula $\varphi$,

1. Let $\preceq^-_\varphi$ be a binary relation on $\mathsf{Bound}(\varphi)$ such that $x \preceq^-_\varphi y$ if and only if $x \in \mathsf{Free}(\varphi_y(y))$. The *dependency order* $\preceq_\varphi$ is defined as the transitive closure of $\preceq^-_\varphi$.
2. A sequence $\langle x_1, x_2, \dots, x_K\rangle \in \mathsf{Bound}(\varphi)^+$ is said to be an alternating chain if:
$$x_1 \preceq^-_\varphi x_2 \preceq^-_\varphi \cdots \preceq^-_\varphi x_K$$
and $\eta_{x_k} \neq \eta_{x_{k+1}}$ for every $k \in \omega$ such that $1 \leq k \leq K-1$. The *alternation depth* of $\alpha$ (denoted $\mathsf{alt}(\alpha)$) is the maximal length of alternating chains such that $x_1 \leq \alpha$. That is, the alternation depth of $\alpha$ is the maximal number of alternations between $\mu$- and $\nu$-operators in $\alpha$.
3. A *priority function* $\Omega_\varphi : \mathsf{Sub}(\varphi) \to \omega$ is defined as follows:
$$\Omega_\varphi(\psi) := \begin{cases} \mathsf{alt}(x) & \text{if } \psi = x,\ \eta_x = \mu \text{ and } \mathsf{alt}(x) \equiv 0 \pmod 2, \\ \mathsf{alt}(x) - 1 & \text{if } \psi = x,\ \eta_x = \mu \text{ and } \mathsf{alt}(x) \equiv 1 \pmod 2, \\ \mathsf{alt}(x) & \text{if } \psi = x,\ \eta_x = \nu \text{ and } \mathsf{alt}(x) \equiv 1 \pmod 2, \\ \mathsf{alt}(x) - 1 & \text{if } \psi = x,\ \eta_x = \nu \text{ and } \mathsf{alt}(x) \equiv 0 \pmod 2, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$
The number $\Omega_\varphi(\psi)$ is called the priority of $\psi$.

**Example 2.5.** For a formula $\varphi = \mu x.\nu y.(\Diamond x \vee (\mu z.(\Diamond z \wedge \Box y)))$, we have $\mathsf{alt}(\varphi) = 3$ since $x \preceq^-_\varphi y \preceq^-_\varphi z$ with $\eta_x \neq \eta_y$ and $\eta_y \neq \eta_z$. Note that although $x \notin \mathsf{Free}(\varphi_z(z))$, we have $x \preceq_\varphi z$.

## 2.2 Semantics

**Definition 2.6 (Kripke model).** A *Kripke model* for the modal $\mu$-calculus is a structure $\mathcal{S} = (S, R, \lambda)$ such that:

- $S = \{s, t, u, \dots\}$ is a non-empty set of *possible worlds.*
- $R$ is a binary relation over $S$ called the *accessibility relation.*
- $\lambda : \mathsf{Prop} \to \mathcal{P}(S)$ is a *valuation.*

**Definition 2.7 (Denotation).** Let $\mathcal{S} = (S, R, \lambda)$ be a Kripke model and let $x$ be a propositional variable. Then for any set of possible worlds $T \in \mathcal{P}(S)$, we can define a new valuation $\lambda[x \mapsto T]$ on $S$ as follows:

$$\lambda[x \mapsto T](p) := \begin{cases} T & \text{if } p = x, \\ \lambda(p) & \text{otherwise.} \end{cases}$$

Moreover, $\mathcal{S}[x \mapsto T]$ denotes the Kripke model $(S, R, \lambda[x \mapsto T])$. A *denotation* $[\![\varphi]\!]_{\mathcal{S}} \in \mathcal{P}(S)$ of a formula $\varphi$ on $\mathcal{S}$ is defined inductively on the structure of $\varphi$ as follows:

- $[\![\bot]\!]_{\mathcal{S}} := \emptyset$ and $[\![\top]\!]_{\mathcal{S}} := S$.
- $[\![p]\!]_{\mathcal{S}} := \lambda(p)$ and $[\![\neg p]\!]_{\mathcal{S}} := S \setminus \lambda(p)$ for any $p \in \mathsf{Prop}$.
- $[\![\varphi \vee \psi]\!]_{\mathcal{S}} := [\![\varphi]\!]_{\mathcal{S}} \cup [\![\psi]\!]_{\mathcal{S}}$ and $[\![\varphi \wedge \psi]\!]_{\mathcal{S}} := [\![\varphi]\!]_{\mathcal{S}} \cap [\![\psi]\!]_{\mathcal{S}}$.
- $[\![\Diamond\varphi]\!]_{\mathcal{S}} := \{s \mid \exists t \in S, (s,t) \in R \wedge t \in [\![\varphi]\!]_{\mathcal{S}}\}$.
- $[\![\Box\varphi]\!]_{\mathcal{S}} := \{s \mid \forall t \in S, (s,t) \in R \Longrightarrow t \in [\![\varphi]\!]_{\mathcal{S}}\}$.
- $[\![\mu x.\varphi(x)]\!]_{\mathcal{S}} := \bigcap\{T \in \mathcal{P}(S) \mid [\![\varphi(x)]\!]_{\mathcal{S}[x \mapsto T]} \subseteq T\}$.
- $[\![\nu x.\varphi(x)]\!]_{\mathcal{S}} := \bigcup\{T \in \mathcal{P}(S) \mid T \subseteq [\![\varphi(x)]\!]_{\mathcal{S}[x \mapsto T]}\}$.

In accordance with the usual terminology, we say that a formula $\varphi$ is *true* or *satisfied* at a possible world $s \in S$ (denoted $\mathcal{S}, s \models \varphi$) if $s \in [\![\varphi]\!]_{\mathcal{S}}$. A formula $\varphi$ is *valid* (denoted $\models \varphi$) if $\varphi$ is true at every world in any model.

**Example 2.8.** Let $\mathcal{S} = (S, R, \lambda)$ be a Kripke model. A formula $\varphi(x)$ such that $x \in \mathsf{Free}(\varphi(x))$ can be naturally seen as the following function:

$$\begin{array}{ccc} \mathcal{P}(S) & \longrightarrow & \mathcal{P}(S) \\ \in & & \in \\ T & \longmapsto & [\![\varphi(x)]\!]_{\mathcal{S}[x \mapsto T]}. \end{array}$$

This function is *monotone* if $x$ is positive in $\varphi(x)$. Thus, by the Knaster-Tarski Theorem [1], $[\![\mu x.\varphi(x)]\!]_{\mathcal{S}}$ and $[\![\nu x.\varphi(x)]\!]_{\mathcal{S}}$ are the least and greatest fixpoint of the function $\varphi(x)$, respectively.

Under this characterization of fixpoint operators, we find that many interesting properties of the Kripke model can be represented by modal $\mu$-formulas. For example, consider the formula $\varphi_1 = \mu x.(\Diamond x \vee p)$. For every Kripke model $\mathcal{S}$ and its possible world $s$, we have $\mathcal{S}, s \models \varphi_1$ if and only if there is some possible world reachable from $s$ in which $p$ is true. Consider the formula $\varphi_2 = \nu y.\mu x.((\Diamond y \wedge p) \vee (\Diamond x \wedge \neg p))$. Then $\mathcal{S}, s \models \varphi_2$ if and only if there is some path from $s$ on which $p$ is true infinitely often.

## 2.3 Axiomatization

We give the Kozen's axiomatization $\mathsf{Koz}$ for the modal $\mu$-calculus in the Tait-style calculus.[5] Hereafter, we will write $\Gamma$, $\Gamma'$, ... for a finite set of formulas. Moreover, the standard abbreviation in the Tait-style calculus are used. That is, we write $\alpha, \Gamma$ for $\{\alpha\} \cup \Gamma$; $\Gamma, \Gamma'$ for $\Gamma \cup \Gamma'$; and $\sim\Gamma$ for $\{\sim\gamma \mid \gamma \in \Gamma\}$ and so forth.

[5] In Kozen's original article [5], the system $\mathsf{Koz}$ was defined as the axiomatization of the equational theory. Nevertheless we present $\mathsf{Koz}$ as an equivalent Tait-style calculus due to the calculus' affinity with the tableaux discussed in the sequel.

**Axioms** Koz contains basic tautologies of classical propositional calculus and the *pre-fixpoint axioms*:

$$\frac{}{\bot \vdash}\ (\mathsf{Bot}) \qquad \frac{}{\varphi, \sim\varphi \vdash}\ (\mathsf{Tau}) \qquad \frac{}{\alpha(\mu x.\alpha(x)), \sim\mu x.\alpha(x) \vdash}\ (\mathsf{Prefix})$$

**Inference Rules** In addition to the classical inference rules from propositional modal logic, for any formula $\varphi(x)$ such that $x$ appears only positively, we have the *induction rule* (Ind) to handle fixpoints:

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

$$\frac{\Gamma \vdash}{\alpha, \Gamma \vdash}\ (\mathsf{Weak}) \qquad \frac{\psi, \{\alpha \mid \Box\alpha \in \Gamma\} \vdash}{\Diamond\psi, \Gamma \vdash}\ (\Diamond)$$

$$\frac{\Gamma, \sim\alpha \vdash \quad \alpha, \Gamma' \vdash}{\Gamma, \Gamma' \vdash}\ (\mathsf{Cut}) \qquad \frac{\varphi(\psi), \sim\psi \vdash}{\mu x.\varphi(x), \sim\psi \vdash}\ (\mathsf{Ind})$$

Of course, the condition of substitution is satisfied in the (Ind)-rule; namely, no new binding relation occurs by applying the substitution $\varphi(\psi)$. As usual, we say that a formula $\sim \bigwedge \Gamma$ is *provable* in Koz (denoted $\Gamma \vdash$) if there exists a proof diagram of $\Gamma$. We frequently use notation such as $\Gamma \vdash \Gamma'$ to mean $\Gamma, \sim\Gamma' \vdash$.

The following two lemmas state some basic properties of Koz. We leave the proofs of these statement as an exercise to the reader.

**Lemma 2.9.** *Let $\varphi$ be a modal $\mu$-formula and let $\alpha(x)$ and $\beta(x,x)$ be modal $\mu$-formulas where $x$ appears only positively. Then, the following holds:*

1. $\vdash \eta x.\alpha(x) \leftrightarrow \eta y.\alpha(y)$ *where* $y \notin \mathsf{Free}(\alpha(x))$.
2. $\vdash \eta x.\beta(x,x) \leftrightarrow \eta x.\eta y.\beta(x,y)$ *where* $y \notin \mathsf{Free}(\beta(x,x))$.
3. $\vdash \mu x.\alpha(x) \leftrightarrow \alpha(\bot)$, *if no appearances of $x$ are in the scope of any modal operators.*
4. $\vdash \nu x.\alpha(x) \leftrightarrow \alpha(\top)$, *if no appearances of $x$ are in the scope of any modal operators.*
5. *We can construct a well-named formula* $\mathsf{wnf}(\varphi) \in \mathsf{WNF}$ *such that* $\vdash \varphi \leftrightarrow \mathsf{wnf}(\varphi)$.

**Lemma 2.10.** *Let $\alpha$, $\beta$, $\varphi(x)$, $\psi(x)$, $\chi_1(x)$ and $\chi_2(x)$ be modal $\mu$-formulas where $x$ appears only positively in $\varphi(x)$ and $\psi(x)$. Further, suppose that $\chi_1(\alpha)$, $\chi_1(\beta)$ and $\chi_2(\alpha)$ are legal substitution; namely, a new binding relation does not occur by such substitutions. Then, the following holds:*

1. *If* $\vdash \varphi(x) \to \psi(x)$ *then* $\vdash \eta x.\varphi(x) \to \eta x.\psi(x)$.
2. *If* $\vdash \alpha \leftrightarrow \beta$ *then* $\vdash \chi_1(\alpha) \leftrightarrow \chi_1(\beta)$.
3. *If* $\vdash \chi_1(x) \leftrightarrow \chi_2(x)$ *then* $\vdash \chi_1(\alpha) \leftrightarrow \chi_2(\alpha)$.

The following lemma is essentially used when proving claim (f).

**Lemma 2.11.** *Let $\varphi$ and $\alpha(x)$ be modal $\mu$-formulas with $x$ appearing only positively in $\alpha(x)$ and $x \notin \mathsf{Free}(\varphi)$. Then we have that if $\alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \vdash \varphi$ then $\mu x.\alpha(x) \vdash \varphi$.*

*Proof.* Suppose that

$$\alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \vdash \varphi$$

By propositional principal we have

$$\alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \vdash \varphi \wedge \alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \tag{2}$$

On the other hand, by (Prefix) rule we have

$$\varphi \wedge \alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \vdash \mu x.(\varphi \wedge \alpha(x)) \tag{3}$$

By combining Statements (2) and (3) we get

$$\alpha\big(\mu x.(\varphi \wedge \alpha(x))\big) \vdash \mu x.(\varphi \wedge \alpha(x)) \tag{4}$$

Therefore by applying ($\mathsf{Ind}$) to (4) we have

$$\mu x.\alpha(x) \vdash \mu x.(\varphi \wedge \alpha(x)) \tag{5}$$

The following statement (6) is easily provable in $\mathsf{Koz}$:

$$\mu x.(\varphi \wedge \alpha(x)) \vdash \varphi \tag{6}$$

Finally apply ($\mathsf{Cut}$) rule to Statement (5) and (6), then we have

$$\mu x.\alpha(x) \vdash \varphi$$

This completes the proof. □

**Remark 2.12.** Let $\varphi(x)$ be a formula where $x$ appears only positively in $\varphi(x)$. By Lemma 2.10 and 2.11, we can assume that $\mathsf{Koz}$ can simulate the following two inference rules:

$$\frac{\alpha\big(\mu x.(\sim\bigwedge\Gamma \wedge \alpha(x))\big), \Gamma \vdash}{\mu x.\alpha(x), \Gamma \vdash}\ (\mathsf{Record}) \qquad \frac{\chi\big(\mu x.\alpha(x)\big), \Gamma' \vdash}{\chi\big(\mu x.(\sim\bigwedge\Gamma \wedge \alpha(x)\big), \Gamma' \vdash}\ (\mathsf{Forget})$$

where $x \notin \mathsf{Free}(\Gamma)$. In the following, we will discuss the $\mathsf{Koz}$ as having the above inference rules from the beginning.

## 2.4 Tableau

**Definition 2.13 (Cover modality).** Let $\Phi$ be a finite set of formulas. Then $\nabla\Phi$ denotes an abbreviation of the following formula:

$$(\bigwedge \Diamond\Phi) \wedge (\Box \bigvee \Phi).$$

Here, $\Diamond\Phi$ denotes the set $\{\Diamond\varphi \mid \varphi \in \Phi\}$, and as always, we use the convention that $\bigvee\emptyset := \bot$ and $\bigwedge\emptyset := \top$. The symbol $\nabla$ is called the *cover modality*.

**Remark 2.14.** Note that the both the ordinary diamond $\Diamond$ and the ordinary box $\Box$ can be expressed in term of cover modality and the disjunction:

$$\Diamond\varphi \equiv \nabla\{\varphi, \top\},$$
$$\Box\varphi \equiv \nabla\emptyset \vee \nabla\{\varphi\}.$$

Therefore, without loss of generality we restrict ourselves to using only $\nabla$ instead of $\Diamond$ and $\Box$. Hereafter, we exclusively use cover modality notation instead of ordinal modal notation; thus *if not otherwise mentioned, all formulas are assumed to be using this new constructor.* Moreover, syntactic concepts such as the well-named formula and the alternation depth extend to formulas using this modality.

**Definition 2.15.** Let $\Gamma$ be a set of formulas. We will say that $\Gamma$ is *locally consistent* if $\Gamma$ does not contain $\bot$ nor any propositional variable $p$ and its negation $\neg p$ simultaneously. On the other hand, $\Gamma$ is said to be *modal* (under $\varphi$) if $\Gamma$ does not contain formulas of the forms $\alpha \vee \beta$, $\alpha \wedge \beta$, $\eta x.\alpha(x)$, or $x \in \mathsf{Bound}(\varphi)$. In other words, if $\Gamma$ is modal, then $\Gamma$ can possess only literals and formulas of the form $\nabla\Phi$.

**Definition 2.16 (Tableau).** Let $\varphi$ be a well-named formula. A set of *tableau rules* for $\varphi$ is defined as follows:

$$\frac{\alpha, \Gamma \mid \beta, \Gamma}{\alpha \vee \beta, \Gamma}\ (\vee) \qquad \frac{\alpha, \beta, \Gamma}{\alpha \wedge \beta, \Gamma}\ (\wedge)$$

$$\frac{\varphi_x(x), \Gamma}{\eta_x x.\varphi_x(x), \Gamma}\ (\eta) \qquad \frac{\varphi_x(x), \Gamma}{x, \Gamma}\ (\mathsf{Regeneration})$$

$$\frac{\{\psi_k\} \cup \{\bigvee\Psi_n \mid n \in N_{\psi_k}\} \mid \text{For every } k \in \omega \text{ with } 1 \leq k \leq i \text{ and } \psi_k \in \Psi_k.}{\nabla\Psi_1, \ldots, \nabla\Psi_i, l_1, \ldots, l_j}\ (\nabla)$$

where in the $(\nabla)$-rule, $l_1, \ldots, l_j \in \mathsf{Lit}(\varphi)$ and $N_{\psi_k} := \{n \in \omega \mid 1 \leq n \leq i,\ n \neq k\}$. Therefore, the premises of a $(\nabla)$-rule is equal to $\sum_{1\leq k\leq i} |\Psi_k|$.

A *tableau* for $\varphi$ is a structure $\mathcal{T}_\varphi = (T, C, r, L)$ where $(T, C, r)$ is a tree structure and $L : T \to \mathcal{P}(\mathsf{Sub}(\varphi))$ is a *label function* satisfying the following clauses:

1. $L(r) = \{\varphi\}$.

2. Let $t \in T$. If $L(t)$ is modal and inconsistent then $t$ has no child. Otherwise, if $t$ is labeled by a set of formulas which fulfills the form of the conclusion of some tableau rules, then $t$ has children which are labeled by the sets of formulas of premises of one of those tableau rules, e.g., if $L(t) = \{\alpha \vee \beta\}$, then $t$ must have two children $u$ and $v$ with $L(u) = \{\alpha\}$ and $L(v) = \{\beta\}$.

3. The rule $(\triangledown)$ can be applied in $t$ only if $L(t)$ is modal.

We call a node $t$ a $(\triangledown)$-node if the rule $(\triangledown)$ is applied between $t$ and its children. The notions of $(\vee)$-, $(\wedge)$-, $(\eta)$- and $(\mathsf{Regeneration})$-node are defined similarly.

**Definition 2.17 (Modal and choice nodes).** Leaves and $(\triangledown)$-nodes are called *modal nodes*. The root of the tableau and children of modal nodes are called *choice nodes*. We say that a modal node $t$ and choice node $u$ are *near* to each other if $t$ is a descendant of $u$ and between the $C$-sequence from $u$ to $t$, there is no node in which the rule $(\triangledown)$ is applied. Similarly, we say that a modal node $t'$ is a *next modal node* of a modal node $t$ if $t'$ is a descendant of $t$ and between the $C$-sequence from $t$ to $t'$, rule $(\triangledown)$ is applied exactly once between $t$ and its child.

**Definition 2.18 (Trace).** Let $\Gamma$ and $\Gamma'$ are finite sets of formulas. We define the *trace function* $\mathsf{TR}_{\Gamma,\Gamma'} : \Gamma \to \mathcal{P}(\Gamma')$ as follows:

- If $\Gamma$ and $\Gamma'$ can be the lower label and one of the upper label of a tableau inference rule respectively, then $\mathsf{TR}_{\Gamma,\Gamma'}$ is a function which outputs set of formulas of the result of reduction of $\gamma$ where $\gamma \in \Gamma$ as input. For instance, if $\Gamma = \{\alpha\wedge\beta, \gamma\}$ and $\Gamma' = \{\alpha, \beta, \gamma\}$, then these are the labels of the following $(\wedge)$-rule:
$$\frac{\alpha, \beta, \gamma}{\alpha \wedge \beta, \gamma}\ (\wedge)$$
Hence, we have $\mathsf{TR}_{\Gamma,\Gamma'}(\alpha \wedge \beta) := \{\alpha, \beta\}$ and $\mathsf{TR}_{\Gamma,\Gamma'}(\gamma) := \{\gamma\}$.

- Otherwise, we set $\mathsf{TR}_{\Gamma,\Gamma'}(\gamma) := \emptyset$ for every $\gamma \in \Gamma$.

Take a finite or infinite sequence $\vec{\Gamma} = \Gamma_1\Gamma_2\ldots$ of finite sets of formulas. A *trace* $\mathsf{tr}$ on $\vec{\Gamma}$ is a finite or infinite sequence of formulas satisfying the following two conditions;

- $\mathsf{tr}[1] = \Gamma_1$.

- For any $n \in \omega \setminus \{0\}$, if $\mathsf{tr}[n]$ is defined and satisfies $\mathsf{TR}_{\Gamma_n,\Gamma_{n+1}}(\mathsf{tr}[n]) \neq \emptyset$, then $\mathsf{tr}[n+1]$ is also defined and satisfies $\mathsf{tr}[n+1] \in \mathsf{TR}_{\Gamma,\Gamma'}(\mathsf{tr}[n])$.

The infinite trace $\mathsf{tr}$ is said to be *even* if
$$\max \mathsf{Infinite}(\vec{\Omega}_\varphi(\mathsf{tr})) = 0 \pmod 2.$$
$\vec{\Gamma}$ is said to be even if there exists an even trace on $\vec{\Gamma}$.

Let $\mathcal{T}_\varphi = (T, C, r, L)$ be a tableau for a well-named formula $\varphi$. Let $\xi$ be an infinite branch of $\mathcal{T}_\varphi$. Then we say $\xi$ is even if $\vec{L}(\xi)$ is even.

**Definition 2.19 ($\mu$-trace).** Let $\mathsf{tr}$ be an infinite trace. Then we call $\mathsf{tr}$ *$\mu$-trace* if the smallest variable (with respect to dependency order $\preceq_\varphi$) regenerated infinitely often is $\mu$-variable. Similarly, We call a trace $\mathsf{tr}$ a *$\nu$-trace* if the smallest variable regenerated infinitely often is $\nu$-variable. Note that every infinite trace $\mathsf{tr}$ is either a $\mu$-trace or $\nu$-trace since all the rules except $(\mathsf{regeneration})$-rule decrease the size of formulas and formulas are eventually reduced since every bound variable is in the scope of some modal operator.

*Based on the above definition, the fact that $\vec{\Gamma}$ is even can be rephrased that $\vec{\Gamma}$ contains a $\mu$-trace.*

## 2.5 Refutation

**Definition 2.20 (Refutation).** A well-named formula $\varphi$ is given. *Refutation rules* for $\varphi$ are defined as the rules of tableau, but this time, we modify the set of rules by adding an explicit weakening rule:
$$\frac{\Gamma}{\alpha, \Gamma}\ (\mathsf{Weak})$$

and, instead of the ($\triangledown$)-rule, we take the following ($\triangledown_r$)-rule:

$$\frac{\{\psi_k\} \cup \{\bigvee \Psi_n \mid n \in N_{\psi_k}\}}{\triangledown\Psi_1, \ldots, \triangledown\Psi_i, l_1, \ldots, l_j} \ (\triangledown_r)$$

where in the ($\vee_r$)-rule, we have $1 \leq k \leq i$, $\psi_k \in \Psi_k$, $N_{\psi_k} = \{n \in \omega \mid 1 \leq n \leq i,\ n \neq k\}$ and $l_1, \ldots, l_j \in \mathsf{Lit}(\varphi)$. Therefore the ($\triangledown_r$)-rule has one premise.

A *refutation* for $\varphi$ is a structure $\mathcal{R}_\varphi = (T, C, r, L)$ where $(T, C, r)$ is a tree structure and $L : T \to \mathcal{P}(\mathsf{Sub}(\varphi))$ is a *label function* satisfying the following clauses:

1. $L(r) = \{\varphi\}$.
2. Every leaf is labeled by some inconsistent set of formulas.
3. Let $t \in T$. If $L(t)$ is modal and inconsistent, then $t$ has no child. Otherwise, if $t$ is labeled by the set of formulas which fulfils the form of the conclusion of some refutation rules, then $t$ has children which are labeled by the sets of formulas of premises of those refutation rules.
4. The rule ($\triangledown_r$) can be applied to $t$ only if $L(t)$ is modal.
5. For any infinite branch $\xi$, $\xi$ is even in the sense of Definition 2.18. In other words, $\vec{L}(\xi)$ contains some $\mu$-traces.

The following theorem is proved by Niwinski and Walukiewicz [3] (see also the literature [6]).

**Theorem 2.21.** *Let $\varphi$ be a well-named formula. If $\varphi$ is not satisfiable, then there exists a refutation for $\varphi$.*

**Remark 2.22.** The refutation is very similar to the proof diagram. Indeed, it is easy to see that among the rules of refutation, ($\vee$), ($\wedge$), and ($\mathsf{Weak}$) are the same as the rules of $\mathsf{Koz}$, and ($\triangledown_r$) corresponds to ($\Diamond$). The remarkable difference is that the proof diagram is a finite tree, whereas the refutation may contain infinite branches. When trying to convert a refutation to a proof diagram, the whole problem lies in "cutting" these infinite branches.

The condition that the infinite branch contains a $\mu$-trace and inference rule ($\mathsf{Record}$) are the keys to solving this cutting problem. Consider an infinite branch $\xi$ of refutation, as shown on the left side of Figure 2. Since $\xi$ contains a $\mu$-trace, there exists $\mu$-variables $x$ that will be regenerated infinitely often.

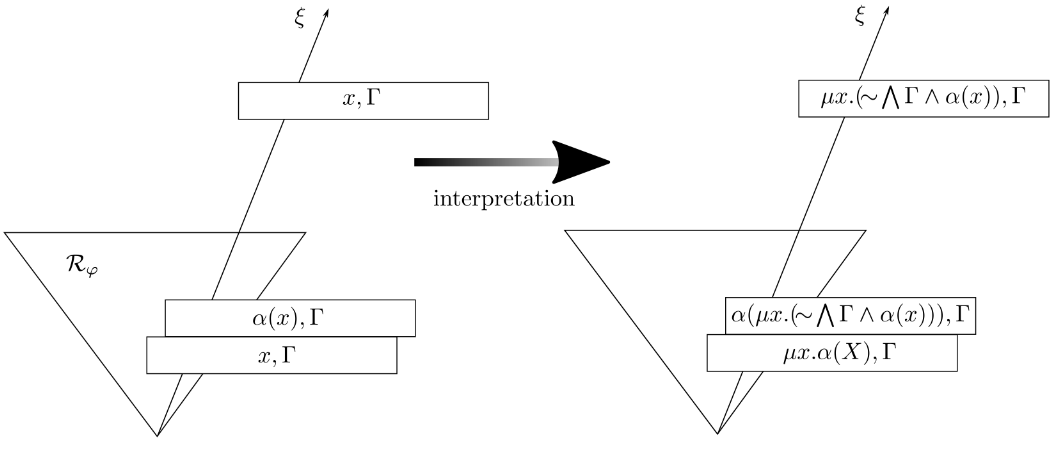


Figure 2: The conversion to proof diagram.

The right side of Figure 2 is constructed so that the interpretation of $x$ matches the inference rule ($\mathsf{record}$); that is, it is interpreted as $x = \mu x.(\sim \bigwedge \Gamma \wedge \alpha(x))$. Since $\{\mu x.(\sim \bigwedge \Gamma \wedge \alpha(x))\} \cup \Gamma$ is provable in $\mathsf{Koz}$, we can obtain the proof diagram we seek. However, in reality, the branch of refutation is branched, so inference rules ($\mathsf{Record}$) and ($\mathsf{Forget}$) must be applied very carefully so that the above argument holds on *all branches*. The strategies for applying these inference rules when refutation satisfies a special *thin* condition will be discussed in detail in the proof of Theorem 4.6.

# 3 Automata

The purpose of this section is to define the terminology of $\omega$-automata theory and to prepare the necessary tools to prove the conpleteness of $\mathsf{Koz}$. Specifically, we will introduce two important concepts, the *Safra's construction* [14] and the *index appearence record* defined by Křetìnsỳ et al. [10].

## 3.1 $\omega$-automata

$\omega$-automata are finite automata that are interpreted over infinite words and recognise $\omega$-regular languages $\mathcal{L} \subseteq \Sigma^\omega$. There are several variations of $\omega$-automata, depending on their acceptance conditions. Among them, we deal with *Büchi automata*, *Rabin automata*, and *parity automata*. Firstly, we define the Büchi automata.

**Definition 3.1 (Büchi automata).** A *Büchi automaton* is a quintuple $\mathcal{BA} = \langle Q, \Sigma, q_0, \Delta, F\rangle$ where:

- $Q$ is a finite set of *states* of the automaton,
- $\Sigma$ is an *alphabet*,
- $q_0 \in Q$ is a state called the *initial state*,
- $\Delta : Q \times \Sigma \to \mathcal{P}(Q)$ is a *transition function*, and
- $F \subseteq Q$ is a set of *final states.*

Using the usual definitions, we say that $\mathcal{BA}$ is *deterministic* if $|\Delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. Let $\mathcal{BA} = (Q, \Sigma, q_0, \Delta, F)$ be a Büchi automaton. A *run* of $\mathcal{BA}$ on an $\omega$-word $\sigma \in \Sigma^\omega$ is an infinite sequence $\rho \in Q^\omega$ of a state where $\rho[1] = q_0$ and $\rho[n+1] \in \Delta(\rho[n], \sigma[n])$ for any $n \geq 1$. An $\omega$-word $\sigma \in \Sigma^\omega$ is *accepted* by $\mathcal{BA}$ if there is a run $\rho$ of $\mathcal{BA}$ on $\sigma$ satisfying the following condition:

$$\mathsf{Infinite}(\rho) \cap F \neq \emptyset.$$

The $\omega$-language of all $\omega$-words accepted by $\mathcal{BA}$ is denoted by $\mathcal{L}(\mathcal{BA})$.

**Remark 3.2.** Consider the problem of converting nondeterministic automaton $\mathcal{BA}$ to its equivalent deterministic automaton $\mathcal{BA}'$, as in the case of finite automata theory. Note that the usual powerset construction which convert of nondeterministic finite automata to deterministic finite automata does not work for $\omega$-automata. For example, consider an automaton $\mathcal{BA}$ and an automation $\mathcal{BA}'$ constructed

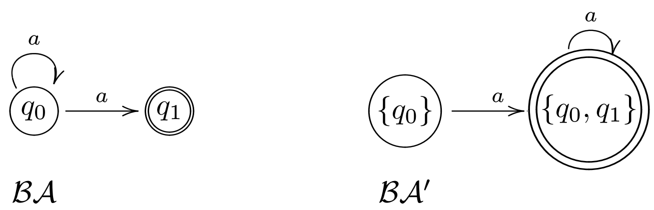


Figure 3: A counterexample of powerset construction.

by the powerset construction, shown in Figure 3. The two automata are not equivalent, $\mathcal{L}(\mathcal{BA}) = \emptyset \neq \{a^\omega\} = \mathcal{L}(\mathcal{BA}')$. The problem is that the fact that the final state $\{q_0, q_1\}$ of the powerset automaton $\mathcal{BA}'$ occurs infinitely often on a run does not guarantee that the automaton $\mathcal{BA}$ has a run on which its final state $q_1$ occurs infinitely often.

Secondly, we define the Rabin automata.

**Definition 3.3 (Rabin automata).** A *Rabin automaton* is a quintuple $\mathcal{RA} = \langle Q, \Sigma, q_0, \Delta, \{(A_j, R_j) \mid j \in J\}\rangle$ where:

- The definition of $Q$, $\Sigma$, $q_0$, and $\Delta$ are exactly the same as for the Büchi automaton.
- $J$ is a finite set of subscripts (index), and for each $j \in J$, $A_j, R_j \subseteq Q$. $(A_j, R_j)$ is called a Rabin's pair. Incidentally, $A_j$ is an acronym for "Accept", $R_j$ is an acronym for "Reject", respectively.

An $\omega$-word $\sigma \in \Sigma^\omega$ is accepted by $\mathcal{RA}$ if there is a run $\rho$ of $\mathcal{RA}$ on $\sigma$ and index $j \in J$ such that:

$$A_j \cap \mathsf{Infinite}(\rho) \neq \emptyset = R_j \cap \mathsf{Infinite}(\rho).$$

Finally, we define the parity automata.

**Definition 3.4** (**Parity automata).** A *parity automaton* is a quintuple $\mathcal{PA} = \langle Q, \Sigma, q_0, \Delta, \mathsf{pri}\rangle$ where:

- The definition of $Q$, $\Sigma$, $q_0$, and $\Delta$ are exactly the same as for the Büchi automaton.
- $\mathsf{pri} : Q \to \omega$ is called the *priority function.*

An $\omega$-word $\sigma \in \Sigma^\omega$ is accepted by $\mathcal{PA}$ if there is a run $\rho$ of $\mathcal{PA}$ on $\sigma$ such that:

$$\max\{\mathsf{pri}(q) \mid q \in \mathsf{Infinite}(\rho)\} \equiv 0 \mod 2.$$

The class of $\omega$-language characterized by the deterministic Büchi automata is denoted by $\mathsf{DBA}$. The class of $\omega$-language characterized by nondeterministic Büchi automata is denoted by $\mathsf{NDBA}$. Similarly, $\mathsf{DRA}$, $\mathsf{NDRA}$, $\mathsf{DPA}$, and $\mathsf{NDPA}$ are classes of $\omega$-language characterized by deterministic Rabin automata, nondeterministic Rabin automata, deterministic parity automata, and nondeterministic parity automata, respectively. Then, it is widely known that the following inclusion holds (see, e.g. the Literature [6]):

$$\mathsf{DBA} \subsetneq \mathsf{NDBA} = \mathsf{DRA} = \mathsf{NDRA} = \mathsf{DPA} = \mathsf{NDPA}$$

In the next subsection, we will prove $\mathsf{NDBA} \subseteq \mathsf{DRA}$ which is the part of the above theorem, by using a method called *Safra's construction.*

## 3.2 Safra's construction

In this subsection, the non deterministic Büchi automaton $\mathcal{BA} = \langle Q, \Sigma, q_0, \Delta, F\rangle$ is fixed and discussed. The subject of this subsection is to specifically construct a deterministic Rabin automaton equivalent to $\mathcal{BA}$. For a given $n \in \omega$, we use $\Pi^n$ to denote the set of all permutations of $N := \{1, \dots, n\}$, i.e., the set of all bijective function $\pi : N \to N$. We identify $\pi$ with its canonical representation as a vector $(\pi(1), \dots, \pi(n))$. In the following, we will often say "the position of $j \in N$ in $\pi$" or similar to refer to $\pi^{-1}(i)$.

**Definition 3.5** (**Safra's tree).** The structure $\mathsf{s} = \langle J, C, 1, l\rangle$ is called *Safra's tree* for $\mathcal{BA}$ when:

1. $J \subseteq \{1, 2, \dots, (|Q| + 1)^2\}$ [6] is the set of vertices[7], where $Q$ is the set of states of $\mathcal{BA}$.
2. $C$ is a childhood relation over $J$.[8]
3. 1 is a root of safra's tree.
4. $l : J \to \mathcal{P}(Q)$ is a labeling function satisfying the following conditions:
   (a) For any $j \in J$, $l(j) \neq \emptyset$.
   (b) For any $j \in J$, $l(j) \supseteq \bigcup_{k\in C(j)} l(k)$. In particular, if $j$ is not the root (i.e. $j \neq 1$), then $l(j) \supsetneq \bigcup_{k\in C(j)} l(k)$.[9]
   (c) For any $j_1, j_2 \in J$, if $j_1$ and $j_2$ are siblings, then $l(j_1) \cap l(j_2) = \emptyset$.

**Remark 3.6.** Let $\mathsf{s} = \langle J, C, r, l\rangle$ be a safra's tree. Consider the assignment $Q \to J \setminus \{1\}$ which assign $j \in J$ for given $q \in Q$, where $q \in l(j)$ and $q \notin \bigcup_{k\in C(j)} l(k)$. According to condition (a) and (b) in part 4 of Definition 3.5, it can be said that the assignment is surjective, and thus $|Q| \geq |J| - 1$ holds. In other words, the number of vertices in the safra's tree is at most $|Q| + 1$.

[6]The number of pool for vertice is $(|Q| + 1)^2$, where the reason why the upper limit is $(|Q| + 1)^2$, will be described later in the Remark 3.9.

[7]The tree structures mentioned in this article are tableau and safra's tree. We use the term vertex for a vertex of safra's tree, and use the term node for a vertex of tableau. We use these terms strictly to prevent confusion.

[8]In the definition of the safra tree, it is common to specify priorities (so-called older-younger relationships) between siblings. however, in this article, the older-younger relationship is defined by a *index appearence record*, so it is not defined here.

[9]Again, it is more general to assume that $l(1) \supsetneq \bigcup_{k\in C(1)} l(k)$. In this article, however, it is intentionally allowed to be $l(1) = \bigcup_{k\in C(1)} l(k)$ so that it is convenient to prove the completeness of the modal $\mu$-calculation later.

**Definition 3.7 (Index appearence record [10]).** A duplex $\langle \pi, \mathsf{col} \rangle$ is an *index appearance record* for $\mathcal{BA}$ if:

1. $\pi \in \Pi^{(|Q|+1)^2}$ is a permutation; that is, $\pi = (\pi(1), \dots, \pi((|Q|+1)^2))$.
2. $\mathsf{col} : \{1, \dots, (|Q|+1)^2\} \to \{\text{green, red, white, black}\}$ is a colouring function. For a vertice $j \in \{1, \dots, (|Q|+1)^2\}$, we say that "$j$ is colored red" or similar if $\mathsf{col}(j) = \text{red}$; and the same applies to other colors.

Take an index appearence record $\langle \pi, \mathsf{col} \rangle$. For any $n, m \in \{1, \dots, (|Q|+1)^2\}$, we say that $n$ is older than $m$ if $n$ is to the right of $m$ (i.e., $\pi^{-1}(m) < \pi^{-1}(n)$).

Let $\mathsf{s} = \langle J, C, r, l \rangle$ be a safra's tree for $\mathcal{BA}$, and $\langle \pi, \mathsf{col} \rangle$ be an index appearence record for $\mathcal{BA}$. We say that $j_1$ is $j_2$'s older brother if $j_1, j_2 \in J$ are siblings, and $j_1$ is older than $j_2$ in $\pi$.

From now on, we will construct a deterministic Rabin automaton $\mathcal{RA} = \langle Q', \Sigma, q'_0, \Delta', \{(A_j, R_j) \mid j \in J\} \rangle$ equivalent to the nondeterministic Büchi automaton $\mathcal{BA}$.

1. Let $\mathsf{S}$ be the set of all safra's trees for $\mathcal{BA}$, and let $\mathsf{IAR}$ be the set of index appearence record for $\mathcal{BA}$. Then, set $Q' := \mathsf{S} \times \mathsf{IAR}$.
2. Set $q'_0 := \langle \{1\}, \emptyset, l_0, \pi_0, \mathsf{col}_0 \rangle$; where $l_0(1) := \{q_0\}$, $\pi_0 := ((|Q|+1)^2, \dots, 3, 2, 1)$, moreover, we set $\mathsf{col}_0(1) = \text{white}$ and $\mathsf{col}_0(k) = \text{black}$ $(k \geq 2)$.
3. The transition function $\Delta'$ will be described later.
4. $J := \{1, \dots, (|Q|+1)^2\}$; that is, the set of indices is the pool of vertices of safra's tree. For any $j \in J$, $A_j$ is the set of states in which $j$ is shining green, and $R_j$ is the set of states in which $j$ is shining red.

**Remark 3.8 (Intuitive meaning of the index appearence record).** Before defining the transition function $\Delta'$, let us explain the intuitive meaning of the index appearence record. Suppose $\mathcal{RA}$ is in state $\langle \mathsf{s}, \pi, \mathsf{col} \rangle$, reads an alphabet, and transitions to state $\langle \mathsf{s}', \pi', \mathsf{col}' \rangle$. In this situation, $\mathsf{s}'$ is generated by adding new vertices to $\mathsf{s}$ and removing unnecessary vertices. The coloring function $\mathsf{col}'$ records the usage of each vertex $j$ in the transition from $\mathsf{s}$ to $\mathsf{s}'$, and intuitively has the meanings shown in the table 1. $\pi'$ represent not only the most recent transition, but also the seniority-based relationships; that is, the farther to the right, the longer it has been used as the vertex of the safra's tree. In particular, since the root 1 is always the oldest, position of 1 is always on the far right side of $\pi$.

Table 1: Intuitive meaning of colors.

| Coloring | Intuitive meaning |
|---|---|
| $\mathsf{col}'(j) = \text{white}$ | $j$ is used for the vertex of $\mathsf{s}'$. |
| $\mathsf{col}'(j) = \text{green}$ | $j$ is used for the vertex of $\mathsf{s}'$; moreover, it is related to an acceptance condition. |
| $\mathsf{col}'(j) = \text{black}$ | $j$ is not used for the vertex of $\mathsf{s}'$, and is waiting for reuse in the pool. |
| $\mathsf{col}'(j) = \text{red}$ | $j$ is not used for the vertex of $\mathsf{s}'$, and was deleted in the latest transition. |

Now let's define the transition $\Delta'$ of $\mathcal{RA}$. Suppose $\mathcal{RA}$ is in the state $\langle J, C, 1, l, \pi, \mathsf{col} \rangle$ and the alphabet $a$ is readed. The next state is generated in the following 7 steps:

Step 1: **Initialize index appearence record.** Let $\pi^{(1)}$ be a permutation obtained from $\pi$ by moving all indices shining in red to the front. For instance, suppose that $\mathsf{col}(\pi[3]) = \mathsf{col}(\pi[5]) = \text{red}$, then $\pi^{(1)}$ is a permutation shown in Figure 4;

Also, the coloring function $\mathsf{col}^{(1)}$ is defined as follows:

$$\mathsf{col}^{(1)}(j) := \begin{cases} \text{white} & (j \in J) \\ \text{black} & (j \notin J) \end{cases}$$

Then, update the state of $\mathcal{RA}$ to $\langle J, C, 1, l, \pi^{(1)}, \mathsf{col}^{(1)} \rangle$.

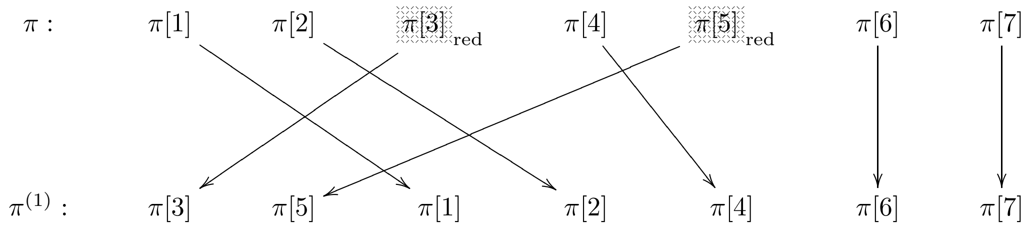


Figure 4: An example of initialize index appearence record.

Step 2: **Update of vertice labels.** Each vertex $j \in J$ is labeled with $l(j) \subseteq Q$. New labeling $l^{(1)} : J \to \mathcal{P}(Q)$ is defined below:

$$l^{(1)}(j) := \Delta(l(j), a)$$

That is, the label of each vertex is updated according to the transition function $\Delta$ of the original Büchi automaton $\mathcal{BA}$. Then, update the state of $\mathcal{RA}$ to $\langle J, C, 1, l^{(1)}, \pi^{(1)}, \mathsf{col}^{(1)} \rangle$.

Step 3: **Add new children.** For each $j \in J$ and $q \in F$, add a new child $k$ of $j$ if $q \in l^{(1)}(j)$ where $k$ is the rightmost vertice black-colored in $\langle \pi^{(1)}, \mathsf{col}^{(1)} \rangle$. In this way, we extend $J$ to $J^{(1)}$ and $C$ to $C^{(1)}$. A label of new child $k$ is $\{q\}$, which extends $l^{(1)}$ to $l^{(2)}$. Also, let $\mathsf{col}^{(2)}$ be the coloring function that changed the color of the newly added $k$ from black to white. Then, update the state of $\mathcal{RA}$ to $\langle J^{(1)}, C^{(1)}, 1, l^{(2)}, \pi^{(1)}, \mathsf{col}^{(2)} \rangle$.

Step 4: **Horizontal pruning.** We obtain a labeling $l^{(3)}$ from $l^{(2)}$ by removing, for every vertex $j \in J^{(1)}$ with label $l^{(2)}(j)$ and all states $q \in l^{(2)}(j)$, $q$ from the labels of all younger siblings of $j$ and all of their descendants. Then, update the state of $\mathcal{RA}$ to $\langle J^{(1)}, C^{(1)}, 1, l^{(3)}, \pi^{(1)}, \mathsf{col}^{(2)} \rangle$. Figure 5 is an example of the horizontal pruning.

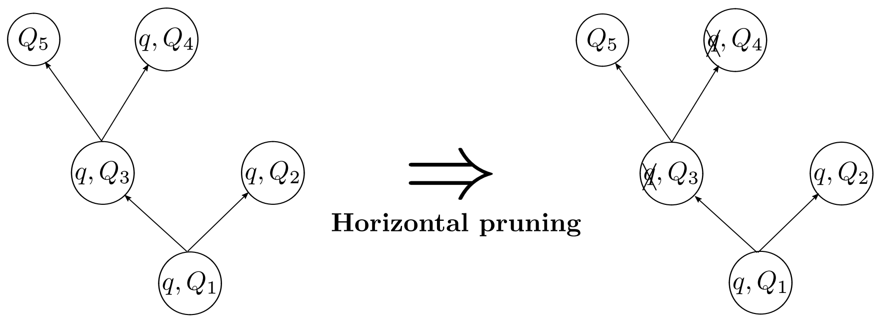


Figure 5: An example of horizontal pruning.

Step 5: **Vertical pruning.** For every $j \in J^{(1)} \setminus \{1\}$, if $l^{(3)}(j) = \bigcup_{k \in C(j)} l^{(3)}(k)$, then remove all descendants of $j$ from the Safra's tree. In this way, $J^{(1)}$ is reduced to $J^{(2)}$. Similarly, $C^{(1)}$ is reduced to $C^{(2)}$, $l^{(3)}$ is reduced to $l^{(4)}$. In addition, we changes the color of $j$ from white to green and the color of the deleted descendant $k$ from white to red. This coloring function is denoted by $\mathsf{col}^{(3)}$. Then, update the state of $\mathcal{RA}$ to $\langle J^{(2)}, C^{(2)}, 1, l^{(4)}, \pi^{(1)}, \mathsf{col}^{(3)} \rangle$.

Step 6: **Removing vertices with empty label.** For each $j \in J^{(2)}$, if $l^{(3)}(j) = \emptyset$, then remove $j$ from the safra's tree. Thus, $J^{(2)}$ is reduce to $J^{(3)}$. Similarly, $C^{(2)}$ is reduced to $C^{(3)}$, $l^{(4)}$ is reduced to $l^{(5)}$. In addition, we change the color of the deleted vertex $j$ from white to red. Let this coloring function be $\mathsf{col}^{(4)}$. Then, update the state of $\mathcal{RA}$ to $\langle J^{(3)}, C^{(3)}, 1, l^{(5)}, \pi^{(1)}, \mathsf{col}^{(4)} \rangle$.

Step 7: **We have a new state.** The state obtained through the above operation is the next states of $\mathcal{RA}$. In other words, $\Delta'$ is defined as,

$$\Delta'(\langle J, C, 1, l, \pi, \mathsf{col} \rangle, a) := \langle J^{(3)}, C^{(3)}, 1, l^{(5)}, \pi^{(1)}, \mathsf{col}^{(4)} \rangle.$$

**Remark 3.9.** From the definition of $\Delta'$, it becomes clear why the number of elements of the pool for the vertex of the safra's tree is $(|Q|+1)^2$. As is mentioned in Remark 3.6, the number of vertices in the safra's tree is at most $|Q|+1$. **Add new children** may add a new vertex $k$ for every $j \in J$ and $q \in F$, that is, up to $(|Q|+1) \cdot |Q| (\geq (|Q|+1) \cdot |F|)$ vertices $k$ may be added. That is, to implement **add new children**, temporarily at the maximum, $|Q|+1+(|Q|+1) \cdot |Q| = (|Q|+1)^2$ vertices need to be prepared.

We prove that $\mathcal{RA}$ is equivalent to $\mathcal{BA}$ by the following two lemmas.

**Lemma 3.10.** *For any $\omega$-word $\sigma \in \Sigma^\omega$, if $\sigma \in \mathcal{L}(\mathcal{BA})$ then $\sigma \in \mathcal{L}(\mathcal{RA})$.*

*Proof.* Suppose that $\omega$-word $\sigma \in \Sigma^\omega$ is accepted by $\mathcal{BA}$; therefore there exists a run $\rho$ of $\mathcal{BA}$ on $\sigma$, and $\rho$ satisfies Büchi's acceptance condition. That is, a certain final state $q \in F$ exists such that $q \in \mathsf{Infinite}(\rho)$. Let $\xi$ be the run of $\mathcal{RA}$ on $\sigma$. For each $n \geq 1$, set $\xi[n] := \langle J_n, C_n, 1, l_n, \pi_n, \mathsf{col_n} \rangle$. In this situation, the definition of Safra's construction shows that for any $n \geq 1$, it becomes $\rho[n] \in l_n(1)$. For a $n \geq 2$ and a vertex $j \in \{1, \ldots, (|Q|+1)^2\}$, when $\rho[n-1] \in l_{n-1}(j)$ and $\rho[n] \notin l_n(j)$, we say "$\rho$ disappears from vertex $j$ on $n$-th transition". Note that from the definition of the transition function $\Delta'$, we can assume that $\rho$ disappears from the vertex $j$ in the $n$-th transition only if either of the following two holds;

**(Case 1)** In the $n$-th run, $\rho[n]$ moved (joined) to $j$'s older brother by **horizontal pruning**.

**(Case 2)** In the $n$-th run, $j$ itself was deleted by **vertical pruning** (an ancestor of $j$ glowed green).

Note that a certain moment $n_0$ exists and that $\rho$ always appears in one of the children of route 1 in the $n$-th runs with $\forall n \geq n_0$; because 1 does not glow green, thus (case 1) cannot occur, and a final states $q$ appears infinitely often in $\rho$. From that moment, $\rho$ can only finitely many times move to older brother by **horizontal pruning**. Let $j_1 \in C_{n_0}(1)$ be the destination where $\rho$ finally moved by **horizontal pruning**. If $j_1$ lights green infinitelly often, then we are done. Otherwise, a moment $n_1$ exists and that $\rho$ always appears in one of the children of $j_1$ in the $n$-th runs with $\forall n \geq n_1$. From that moment, $\rho$ can only finitely many times move to older brother by **horizontal pruning**. Let $j_2 \in C_{n_0}(j_1)$ be the destination where $\rho$ finally moved by **horizontal pruning**. If $j_2$ lights green infinitely often, then we are done. Otherwise, we repeat the reasoning and find a son of $j_1$, $j_2$ an so on. Observe that we cannot go this way forever because safra's tree has a bounded size. Therefore, there exists a vertex $j_i$ $(i \geq 1)$ such that $j_i$ is deleted only a finite times and shines green infinitely often. ☐

**Lemma 3.11.** *For any $\omega$-word $\sigma \in \Sigma^\omega$, if $\sigma \in \mathcal{L}(\mathcal{RA})$ then $\sigma \in \mathcal{L}(\mathcal{BA})$.*

*Proof.* Let $\sigma \in \Sigma^\omega$. Let $\xi$ be the run of $\mathcal{RA}$ on $\sigma$, and set $\xi[n] = \langle J_n, C_n, 1, l_n, \pi_n, \mathsf{col}_n \rangle$. First, suppose that $1 < N < M$ and the vertex $j \in \{1, \ldots, (|Q|+1)^2\}$ satisfy the following two conditions.

1. $j$ is always used as a vertex between $\xi[N]$ and $\xi[M]$. In other words, $N \leq \forall k \leq M$, $\mathsf{col}_k(j) \neq$ red.
2. $\mathsf{col}_N(j) = \mathsf{col}_M(j) =$ green, and $N < \forall k < M$, $\mathsf{col}_k(j) \neq$ green.

From the definition of $\mathcal{RA}$, for any $q_M \in l_M(j)$, there exist $q_N \in l_N(j)$ and a sequence $q_N q_{N+1} \ldots q_M$ such that:

$$N \leq \forall k \leq M-1,\ q_{k+1} \in \Delta(q_k, \sigma[k]).$$

Such a sequence $q_N q_{N+1} \ldots q_M$ is called "$\Delta$-path from $q_N$ to $q_M$". Then, the following claim holds.

(†) For any $q_N \in l_N(j)$ and $q_M \in l_M(j)$, if there is a $\Delta$-path from $q_N$ to $q_M$, then at least one of them (let's denote this $q_N q_{N+1} \ldots q_M$) intersects $F$. That is, there exists a $L$ with $N \leq L \leq M$ such that $q_L \in F$.

The above claim is proved by induction, but we leave it as an exercise for the reader.

Now suppose $\sigma$ is accepted by $\mathcal{RA}$. Thus, there is a run $\xi$ of $\mathcal{RA}$ on $\sigma$, and $\xi$ satisfies Rabin's acceptance condition; that is, there is a $j \in \{1, \ldots, (|Q|+1)^2\}$ which meets the following conditions:

- We can take $N_0 > 1$, and $j$ is always used in the transitions after the $N_0$-th transition.
- There exist $N_0 < N_1 < \cdots < N_k < \cdots$ with $(k \in \omega)$ such that $\mathsf{col}_{N_k}(j) = \mathsf{col}_{N_{k+1}}(j) =$ green, and $N_i < \forall k < N_{i+1}$, $\mathsf{col}_k(j) \neq$ green.

Let $\rho$ be a run of $\mathcal{BA}$ on $\sigma$ such that $\forall n > N_0$, $\rho[n] \in l_n(j)$. In general, $\rho$ does not always satisfy the Büchi's acceptance condition, but from Claim (†), the following holds:

**(‡)** There is a $\Delta$-path from $\rho[N_i]$ to $\rho[N_{i+1}]$ which intersects $F$ (let's denote this $q_{N_i} q_{N_i+1} \dots q_{N_{i+1}}$).

Let $\rho'$ be the $\Delta$-sequence in which a part $\rho[N_i]$ to $\rho[N_{i+1}]$ of sequence $\rho$ is replaced with another sequence $q_{N_i} q_{N_i+1} \dots q_{N_{i+1}}$ for every $i \geq 1$. Then, from Claim (‡), $\rho'$ satisfy the Büchi's acceptance condition, therefore $\sigma \in \mathcal{L}(\mathcal{BA})$. $\square$

From Lemma 3.10 and Lemma 3.11, we obtain the following theorem.

**Theorem 3.12 (Safra's construction [14]).** *For any nondeterministic Büchi automaton $\mathcal{BA}$, the deterministic Rabin automaton $\mathcal{RA}$ generated by Safra's construction satisfies $\mathcal{L}(\mathcal{BA}) = \mathcal{L}(\mathcal{RA})$.*

## 3.3 Conversion to parity automaton

Suppose that an arbitrarily nondeterministic Büchi automaton $\mathcal{BA}$ is given. In this subsection, we will construct an equivalent deterministic parity automaton $\mathcal{PA}$. In fact, most of the content to be discussed is completed in the previous subsection 3.2.

Let $\langle \pi, \mathsf{col} \rangle$ be an index appearence record for $\mathcal{BA}$. Set

$$\mathsf{maxind}(\pi, \mathsf{col}) := \max\{\pi^{-1}(j) \mid \mathsf{col}(j) \in \{\text{green}, \text{red}\}\} \cup \{0\}.$$

In other words, among the vertices colored in either green or red in $\pi$, the position of the vertex on the far right of these is $\mathsf{maxind}(\pi, \mathsf{col})$. From this, we will concretely build the deterministic Parity automaton $\mathcal{PA} = \langle Q', \Sigma, q_0', \Delta', \mathsf{pri} \rangle$ as follows:

1. $Q'$, $q_0'$, and $\Delta'$ are exactly the same as the Rabin automaton defined in Safra's construction.
2. The priority function is defined as follows:

$$\mathsf{pri}\big(\langle \mathsf{s}, \pi, \mathsf{col} \rangle\big) := \begin{cases} 1 & \big(\mathsf{maxind}(\pi, \mathsf{col}) = 0\big) \\ 2 \cdot \mathsf{maxind}(\pi, \mathsf{col}) & \big(\mathsf{col}(\pi[\mathsf{maxind}(\pi, \mathsf{col})]) = \text{green}\big) \\ 2 \cdot \mathsf{maxind}(\pi, \mathsf{col}) + 1 & \big(\mathsf{col}(\pi[\mathsf{maxind}(\pi, \mathsf{col})]) = \text{red}\big) \end{cases}$$

**Theorem 3.13 (Legitimacy of $\mathcal{PA}$ [10]).** *For any nondeterministic Büchi automaton $\mathcal{BA}$, the deterministic parity automaton $\mathcal{PA}$ generated by construction shown above satisfies $\mathcal{L}(\mathcal{BA}) = \mathcal{L}(\mathcal{PA})$.*

*Proof.* Let $\mathcal{RA}$ be the Rabin automaton constructed by Safra's construction. From Theorem 3.12, it is enough to show that $\mathcal{L}(\mathcal{RA}) = \mathcal{L}(\mathcal{PA})$. Take an $\omega$-word $\sigma \in \Sigma^\omega$ arbitrarily. Let $\xi$ be the run of $\mathcal{PA}$ on $\sigma$. Note that $\xi$ is also a run of $\mathcal{RA}$.

First, note that the position of any vertex $j \in \{1, \dots, (|Q|+1)^2\}$ only changes in two different ways:

- $j$ itself is removed from the safra's tree and driven to the far left in **Initialize index appearence record**. In this case, we say that $j$ was *demoted* in the transition.
- Some $k \in \{1, \dots, (|Q|+1)^2\}$ with a position older than $j$ has been removed (demoted), increasing the position of $j$. In this case, we say that $j$ was *promoted* in the transition.

Set $\xi[n] = \langle J_n, C_n, 1, l_n, \pi_n, \mathsf{col_n} \rangle$ for $n \geq 1$. Suppose that a vertex $j \in \{1, \dots, (|Q|+1)^2\}$ and a natural number $N \geq 1$ satisfy $\forall n \geq N$, $j \in J_n$. In this situation, $j$ will not be demoted in the $N$-th and subsequent transitions, and promotion can be done only finitely many times, so if a sufficiently large $M > N$ is taken, then $j$ will not be demoted nor promoted in the $M$-th and subsequent transitions. The position of $j$ when it is no longer demoted nor promoted $\pi_M^{-1}(j)$ is called the *stable position* of $j$ in the run $\xi$ (notated as $\mathsf{stable}_\xi(j)$).

It is obvious from the definition of the priority function $\mathsf{pri}$ that if $\xi$ satisfies the parity condition, then $\xi$ also satisfies the Rabin's acceptance condition. On the contrary, if $\xi$ satisfies Rabin's acceptance condition, then there exists some $j \in \{1, \dots, (|Q|+1)^2\}$ such that

$$A_j \cap \mathsf{Infinite}(\xi) \neq \emptyset = R_j \cap \mathsf{Infinite}(\xi).$$

Let $k$ be the vertex with the largest stable position among such $j$s. In the transition well ahead, $k$ is in a stable position, colored green infinitely often, and the elders of $k$ are not colored red ($\because$ if the elders of $k$ are removed, $k$ is promoted). Therefore, we have

$$\max\{\mathsf{pri}(q) \mid q \in \mathsf{Infinite}(\xi)\} = 2 \cdot \mathsf{stable}_\xi(k),$$

that is, $\xi$ satisfy the parity condition. Hence, $\mathcal{L}(\mathcal{RA}) = \mathcal{L}(\mathcal{PA})$ holds. $\square$

# 4 Application of automata to the modal $\mu$-calculous

In this section, we apply the results of Section 3 to the modal $\mu$-calculous to prove two important results. First, in Subsection 4.1, we give an automaton that determines the parity of the tableau branch. In the following Subsection 4.2, this automaton is used to prove completeness of $\mathsf{Koz}$ for the thin refutation; which is Claim (f) mentioned in Section 1. In the last Subsection 4.3, the proof of the existence of the automaton normal form (Claim (b)) is proved along with the concrete construction method.

## 4.1 Automata that determines the parity of tableau branches

**Definition 4.1 (Activeness).** Let $\varphi$ be a well-named formula, and $\preceq_\varphi$ be its dependency order (recall Definition 2.4). Then, For any $\psi \in \mathsf{Sub}(\varphi)$ and $x \in \mathsf{Bound}(\varphi)$, we say $x$ is *active* in $\psi$ if there exists $y \in \mathsf{Sub}(\psi) \cap \mathsf{Bound}(\varphi)$ such that $x \preceq_\varphi y$.

Suppose that a well-named formula $\varphi$ is arbitrarily given. From now on, we will construct a nondeterministic Büchi automaton $\mathcal{BA}_\varphi = \langle Q, \Sigma, q_0, \Delta, F\rangle$ which determine the parity of the tableau branch for $\varphi$. The letter handled by the automaton $\mathcal{BA}_\varphi$ is subset $\Gamma \subseteq \mathsf{Sub}(\varphi)$, therefore $\Sigma := \mathcal{P}(\mathsf{Sub}(\varphi))$. The state $q \in Q$ is of the form $q = (\Gamma, \gamma) \in \mathcal{P}(\mathsf{Sub}(\varphi)) \times \mathsf{Sub}(\varphi)$ or $q = (\Gamma, \gamma, x) \in \mathcal{P}(\mathsf{Sub}(\varphi)) \times \mathsf{Sub}(\varphi) \times \mathsf{Bound}(\varphi)$ which satisfies the following three conditions:

1. $\gamma \in \Gamma$
2. $x$ is active in $\gamma$.
3. $x$ is $\mu$-variable in $\varphi$. That is, $\mu x.\varphi_x(x) \in \mathsf{Sub}(\varphi)$.

The initial state is $(\{\varphi\}, \varphi)$. The transition function $\Delta$ is defined as follows:

$$\Delta\big((\Gamma,\gamma),\Gamma'\big) := \{(\Gamma',\gamma'), (\Gamma',\gamma',x) \mid \gamma' \in \mathsf{TR}_{\Gamma,\Gamma'}(\gamma),\ x \in \mathsf{Bound}(\varphi)\},$$
$$\Delta\big((\Gamma,\gamma,x),\Gamma'\big) := \{(\Gamma',\gamma',x) \mid \gamma' \in \mathsf{TR}_{\Gamma,\Gamma'}(\gamma)\}.$$

Finally, the final state is defined as $F := \{(\Gamma, x, x) \in Q \mid x \in \mathsf{Bound}(\varphi)\}$. $\mathcal{BA}_\varphi$ embodies a naive way to seek $\mu$-trace non-deterministically. Indeed, let

$$\xi = (\Gamma_1,\gamma_1)\dots(\Gamma_k,\gamma_k)(\Gamma_{k+1},\gamma_{k+1},x)(\Gamma_{k+2},\gamma_{k+2},x)\dots$$

be a run of $\mathcal{BA}_\varphi$ on $\vec{L}(\rho)$ where $\rho$ is an infinite branch of a tableau $\mathcal{T}_\varphi$. Then, from the definition of $\mathcal{BA}_\varphi$, it can be seen that $\vec{L}(\rho) = \Gamma_1\Gamma_2\dots$ and that $\gamma_1\gamma_2\dots$ is a trace on $\vec{L}(\rho)$. In short, a run picked a specific trace $\gamma_1\gamma_2\dots$ from multiple traces on $\rho$. The intuitive meaning of transitioning from $(\Gamma_k,\gamma_k)$ to $(\Gamma_{k+1},\gamma_{k+1},x)$ is,

($\star$) $x$ is a variable such that the value of $\Omega_\varphi(x)$ is maximized in the $(k+1)$-th and subsequent transitions; and that regenerated infinitely often.

Indeed, for any $y \in \mathsf{Bound}(\varphi)$, if $\Omega_\varphi(y) > \Omega_\varphi(x)$, then $(\Gamma, y, x)$ cannot be a states of automaton because $y$ is not active in $x$. Therefore, $y$, which has a higher priority than $x$, does not appear in the traces $\gamma_{k+1}\gamma_{k+2}\dots$. In addition, if $\rho$ is accepted, a states in the form of $(\Gamma, x, x)$ must appear infinitely often in $\rho$. This means that $x$ will be regenerated infinitely often in the trace $\gamma_1\gamma_2\dots$. Therefore, Claim ($\star$) agrees that the automaton accepts $\xi$. From the above, $\mathcal{BA}_\varphi$ certainly determines the parity of tableau branches.

Let $\mathcal{PA}_\varphi$ be the parity automaton which is converted from $\mathcal{BA}_\varphi$ by the method introduced in Subsection 3.3. Then $\mathcal{PA}_\varphi$ becomes a deterministic parity automaton that determines the parity of tableau branches. Hereinafter, We denote $N_\varphi := (|Q|+1)^2$; where $Q$ is a set of states of $\mathcal{BA}_\varphi$.

**Remark 4.2.** Let $\mathcal{PA}_\varphi = \langle Q, \Sigma, q_0, \Delta, \mathsf{pri}\rangle$ be the parity automaton given above. Let $\mathcal{T}_\varphi = (T, C, r, L)$ be a tableau for $\varphi$. For any node $t \in T$, set

$$\Delta(\vec{L}(\vec{rt}), q_0) = \langle J_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t\rangle.$$

Then, if $(\Gamma,\gamma) \in l_t(1)$ or $(\Gamma,\gamma,x) \in l_t(1)$, then $\Gamma = L(t)$ holds. Moreover, $\{\gamma \mid (\Gamma,\gamma) \in l_t(1)\} = L(t)$ holds. What this means is that there is duplication of information in the first and second quadrants of label elements of the safra's tree . With this in mind, we can omitt the first quadrant, hence each vertex of the safra's tree is labeld by $\mathsf{Sub}(\varphi) \cup (\mathsf{Sub}(\varphi) \times \mathsf{Bound}(\varphi))$. In this article, we will think so in the following. In other words, the label of the safra's tree is considered to be in the shape of

$$\{\gamma_1, \dots, \gamma_j\} \cup \{(\gamma'_1, x_1), \dots, (\gamma'_k, x_k)\} \quad (j, k \geq 0).$$

## 4.2 Completeness for thin refutation

In this subsection, the completeness of $\mathsf{Koz}$ will be proven when $\varphi$ has thin refutation. The parity automaton $\mathcal{PA}_\varphi$ created in Subsection 4.1 is used for the proof. First, we will define the concept of the thin refutation.

**Definition 4.3 (Thin refutation).** Let $\mathcal{R}_\varphi$ be a refutation for some well-named formula $\varphi$. We say that $\mathcal{R}_\varphi$ is *thin* if, whenever a formula of the form $\alpha \wedge \beta$ is reduced, some node of the refutation and some variable is active in $\alpha$ as well as $\beta$, then at least one of $\alpha$ and $\beta$ is immediately discarded by using the $(\mathsf{Weak})$-rule.

**Remark 4.4.** Let $\mathcal{R}_\varphi = (T, C, r, L)$ be a thin refutation. Let $\mathcal{PA}_\varphi$ be a parity automaton that determines the parity of tableau branches. For any node $t \in T$, set $\Delta(\vec{L}(\vec{rt}), q_0) = \langle J_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t \rangle$; then it has the following distinctive characteristics:

- For non-root vertices $k \in \{2, 3, \dots, N_\varphi\}$, $k$ is labeled with elements in the form of $(\gamma, x)$.
- For any $k \in \{2, 3, \dots, N_\varphi\}$, $l_t(k)$ consists of at most two elements.
- For any $k \in \{2, 3, \dots, N_\varphi\}$, if $l_t(k)$ consists of two elements, then $t$ is a $(\mathsf{Weak})$-node, and one of them is discarded by $(\mathsf{Weak})$-rule in the transition between $t$ and its child.

In short, if $\mathcal{PA}_\varphi$ load $\vec{L}(\xi)$ where $\xi$ is a branch of thin refutation, each vertex of the safra's tree (except the root) will be labeled with a single element in almost all cases.

**Definition 4.5 (Definition list).** Let $\varphi$ be a well-named formula. The sequence $(x_1, x_2, \dots x_N)$ is a linear ordering of all bound variables of $\varphi$ which is compatible with dependency order, i.e., if $x_i \preceq_\varphi x_j$ then $i \leq j$. A definition list $\mathcal{D}_\varphi$ is a following format:

$$\mathcal{D}_\varphi := \big(x_1 = \eta_1 x_1.\alpha_1(x_1), \dots, x_N = \eta_N x_N.\alpha_N(x_N)\big);$$

where for any $k \leq N$, $\eta_k x_k.\alpha_k(x_k) = \eta_{x_k} x_k.\varphi_{x_k}(x_k)$, or $\eta_k x_k.\alpha_k(x_k) = \eta_{x_k} x_k.(\beta \wedge \varphi_{x_k}(x_k))$ holds with $\beta$ is arbitrary formula such that $x_k \notin \mathsf{Free}(\beta)$. Especially, we call

$$\big(x_1 = \eta_1 x_1.\varphi_{x_1}(x_1), \dots, x_N = \eta_N x_N.\varphi_{x_N}(x_N)\big)$$

a plain definition list. For any $\beta \in \mathsf{Sub}(\varphi)$, we define a expantion $\langle\!\langle \beta \rangle\!\rangle_{\mathcal{D}_\varphi}$ of $\beta$ by $\mathcal{D}_\varphi$ as follows:

$$\langle\!\langle \beta \rangle\!\rangle_{\mathcal{D}_\varphi} := \beta[x_N/\eta_N x_N.\alpha_N(x_N)] \dots [x_1/\eta_1 x_1.\alpha_1(x_1)].$$

In addition, for $\Gamma \subseteq \mathsf{Sub}(\varphi)$, we set $\langle\!\langle \Gamma \rangle\!\rangle_{\mathcal{D}_\varphi} := \{\langle\!\langle \gamma \rangle\!\rangle_{\mathcal{D}_\varphi} \mid \gamma \in \Gamma\}$. In the following, if the $\varphi$ being discussed is clear from the context, it may be represented by $\mathcal{D}$, omitting the subscript of $\mathcal{D}_\varphi$.

**Theorem 4.6 (Completeness for formulas in which the thin refutation exists).** *Let $\varphi$ be a well-named formula. If there exists a thin refutation for $\varphi$, then $\sim\varphi$ is probable in* $\mathsf{Koz}$.

*Proof.* Let $\mathcal{R}_\varphi = (T, C, r, L)$ be a thin refutation for $\varphi$. Our goal is to show that there exists label $f : T \to \mathcal{P}(\mathsf{Form})$ satisfies the following conditions:

(C1) $f(r) = \{\varphi\}$.

(C2) For any $t \in T$ and its children $u_1, \dots, u_k \in C(t)$,

$$\frac{f(u_1) \ \mid \cdots \mid \ f(u_k)}{f(t)}$$

can be simulated with axiomatic system $\mathsf{Koz}$. That is, if $f(u_i) \vdash_{\mathsf{Koz}}$ for every $i$ $(1 \leq i \leq k)$, then $f(t) \vdash_{\mathsf{Koz}}$.

(C3) For any leaf $t$ of $\mathcal{R}_\varphi$, $f(t)$ is inconsistent; thus $f(t) \vdash_{\mathsf{Koz}}$ holds.

(C4) For any infinite branch $\xi$ of $\mathcal{R}_\varphi$, there exists a node $t$ on $\xi$ such that $f(t) = \{\Gamma,\ \mu x.\big(\bigwedge \sim\Gamma \wedge \alpha(x)\big)\}$ for some $\alpha(x), \beta \in \mathsf{Form}$. Thus $f(t) \vdash_{\mathsf{Koz}}$ holds.

It is clear that the theorem holds if the above $f$ can be defined. As is mentioned in Remark 2.22, in constructing $f$, we must carefully apply inference rules ($\mathsf{Record}$) and ($\mathsf{Forget}$). Figure 6 illustrates the idea of how to construct the function $f$. Let $j$ be a vertex where a formula belonging to $j$ is reduced

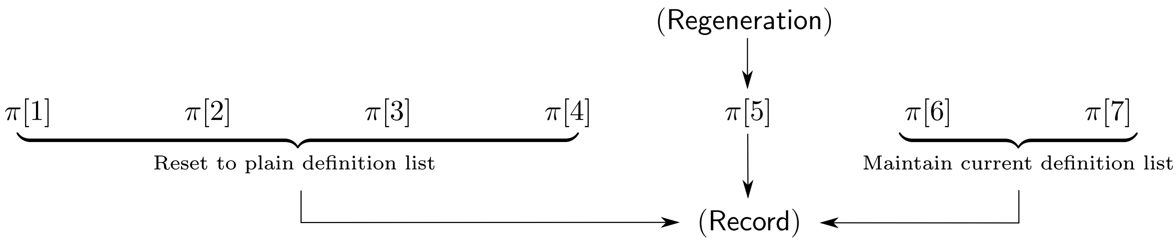


Figure 6: The idea of construction of function $f$.

by ($\mathsf{Regeneration}$)-rule. The basic strategy is to apply ($\mathsf{Forget}$) at the vertex on the left side of $j$, and to apply ($\mathsf{Record}$) at $j$.

For any node $t \in T$, set

$$\Delta_\varphi(\vec{L}(\vec{rt}), q_0) = q_t = \langle J_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t \rangle.$$

From now, for each node $t \in T$ and vertex $j \in \{1, \ldots, N_\varphi\}$, the definition list $\mathcal{D}_{t,j}$ is inductively defined from the root of the tree toward the leaves. First, set $\mathcal{D}_{r,j}$ $(j \geq 1)$ be a plain definition list. Next, assuming that $\mathcal{D}_{t,j}$ $(t \in T, j \leq N_\varphi)$ is already defined, for each $t' \in C(t)$ and $j' \in \{1, \ldots, N_\varphi\}$, $\mathcal{D}_{t',j'}$ is defined by case as follows:

**(Case 1)** $t$ is not a ($\mathsf{Regeneration}$)-node: For the vertex $j'$ that is deleted in the transition from $q_t$ to $q_{t'}$, $\mathcal{D}_{t',j'}$ is a plain definition list. For other $j'$, set $\mathcal{D}_{t',j'} := \mathcal{D}_{t,j'}$; that is, it inherits the same definition list from the same vertex of the parent node.

**(Case 2)** $t$ is a ($\mathsf{Regeneration}$)-node: Suppose the following inferences are made between $t$ and $t'$:

$$\frac{\varphi_x(x), \Gamma}{x, \Gamma}\ (\mathsf{Regeneration})$$

If $x$ is a $\nu$-variable, then $\mathcal{D}_{t',k'}$ is similar to (Case 1). If $x$ is a $\mu$-variable, and there does *not* exist the vertex $k \in \{1, \ldots, N_\varphi\}$ such that $l_t(k) = \{(x, x)\}$, then $\mathcal{D}_{t',k'}$ is also similar to (Case 1). If $x$ is a $\mu$-variable, and there exists the vertex $k \in \{1, \ldots, N_\varphi\}$ such that $l_t(k) = \{(x, x)\}$, then

- Let $\mathcal{D}_{t',k'} := \mathcal{D}_{t,k'}$ for $k'$ on the right side of $k$ in $\pi$. That is, for $k'$ older than $k$, the same definition list is inherited from the same vertex of the parent node.
- For $k'$ on the left side of $k$ in $\pi$, $\mathcal{D}_{t',k'}$ is a plain definition list.
- The definition list $\mathcal{D}_{t',k}$ is obtained by replacing the definition of $x$ in definition list $\mathcal{D}_{t,k}$ with $x = \mu x.\big(\sim \langle\!\langle \Gamma \rangle\!\rangle \wedge \varphi_x(x)\big)$; where $\langle\!\langle \Gamma \rangle\!\rangle := \{\langle\!\langle \gamma \rangle\!\rangle_{\mathcal{D}_{t',k'}} \mid \gamma \in l_{t'}(k'),\ k' \in J_{t'} \setminus \{k\}.\}$.

Set $f(t) := \bigcup_{j \in J_t} \langle\!\langle l_t(j) \rangle\!\rangle_{\mathcal{D}_{t,j}}$, then $f$ is the function we seek. Indeed, it is obvious that $f$ satisfies (C1), (C2), and (C3). For (C4), take an infinite branch $\xi$ arbitrarily and consider the run $\vec{q} := q_{\xi[1]} q_{\xi[2]} \ldots$. Since $\vec{L}(\xi)$ contains $\mu$-trace, $\vec{q}$ satisfies the parity condition; therefore we can find a vertex $k \in \{1, \ldots, N_\varphi\}$ and $M > N > 1$ such that

- $\max\{\mathsf{pri}(q) \mid q \in \mathsf{Infinite}(\vec{q})\} = 2 \cdot \mathsf{stable}_{\vec{q}}(k) \equiv 0 \mod 2$.
- $\pi^{-1}_{\xi[N]}(k) = \pi^{-1}_{\xi[M]}(k) = \mathsf{stable}_{\vec{q}}(k)$.
- $\xi[N]$ and $\xi[M]$ are both ($\mathsf{Regeneration}$)-nodes and regenerated $x$ in them where vertex $k$ is labeled by $\{(x, x)\}$ in $\xi[N]$ and $\xi[M]$.

- For any $L$ ($N < L < M$) and $k$'s older brother $j$ in $\xi[L]$, the label for $j$ is not regenerated. In other words, the definition list of $k$'s older brother does not change between $\xi[N]$ and $\xi[M]$.

From the above conditions, $\mathcal{D}_{\xi[N],j} = \mathcal{D}_{\xi[M],j}$ for any $j \leq N_\varphi \setminus \{k\}$ and thus $f(\xi[M]) = \{\langle\!\langle \Gamma \rangle\!\rangle, \mu x.(\sim \bigwedge \langle\!\langle \Gamma \rangle\!\rangle \wedge \varphi_x(x))\}$ with $\langle\!\langle \Gamma \rangle\!\rangle = \{\langle\!\langle \gamma \rangle\!\rangle_{\mathcal{D}_{\xi[M],k'}} \mid \gamma \in l_{\xi[M]}(k'),\ k' \in J_{\xi[M]} \setminus \{k\}.\}$. Therefore, it certainly satisfies (C4). □

### 4.3 Automaton normal form

**Definition 4.7 (Automaton normal form).** The set of an *automaton normal form* $\mathsf{ANF}$ is the smallest set of formulas defined by the following clauses:

1. If $l_1, \ldots, l_i \in \mathsf{Lit}$, then $\bigwedge_{1 \leq j \leq i} l_j \in \mathsf{ANF}$.
2. If $\alpha \vee \beta \in \mathsf{ANF}$, $\mathsf{Bound}(\alpha) \cap \mathsf{Free}(\beta) = \emptyset$ and $\mathsf{Free}(\alpha) \cap \mathsf{Bound}(\beta) = \emptyset$, then $\alpha \vee \beta \in \mathsf{ANF}$.
3. If $\alpha(x) \in \mathsf{ANF}$ where $x$ occurs only positively in the scope of some modal operator (cover modality), occurs at once, and $\mathsf{Sub}(\alpha(x))$ does not contain a formula of the form $x \wedge \beta$ where $\beta \neq \top$. Then, $\eta x.\alpha(x) \in \mathsf{ANF}$.
4. If $\Phi \subseteq \mathsf{ANF}$ is a finite set such that for any $\varphi_1, \varphi_2 \in \Phi$, we have $\mathsf{Bound}(\varphi_1) \cap \mathsf{Free}(\varphi_2) = \emptyset$, then $(\nabla \Phi) \wedge (\bigwedge_{1 \leq i \leq j} l_i) \in \mathsf{ANF}$ where $l_1, \ldots, l_j \in \mathsf{Lit} \setminus \bigcup_{\varphi \in \Phi} \mathsf{Bound}(\varphi)$ with $0 \leq j$.
5. If $\alpha \in \mathsf{ANF}$ then $\alpha \wedge \top \in \mathsf{ANF}$.

Note that the above clauses imply $\mathsf{ANF} \subseteq \mathsf{WNF}$.

**Remark 4.8.** For any automaton normal form $\widehat{\varphi}$, a tableau $\mathcal{T}_{\widehat{\varphi}} = (T, C, r, L)$ for $\widehat{\varphi}$ forms very simple shapes. Indeed, for any node $t \in T$, there exists at most one formula $\widehat{\alpha} \in L(t)$ which includes some bound variables. Note that for any infinite trace $\mathsf{tr}$, $\mathsf{tr}[n]$ must include some bound variables. Consequently, for any infinite branch of the tableau for an automaton normal form, there exists a unique trace on it.

**Definition 4.9 (Tableau bisimulation).** Let $\mathcal{T}_\alpha = (T, C, r, L)$ and $\mathcal{T}_\beta = (T', C', r', L')$ be two tableaux for some well-named formulas $\alpha$ and $\beta$. Let $T_m$ and $T'_m$ be sets of modal nodes of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, respectively, and let $T_c$ and $T'_c$ be a set of choice nodes of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, respectively. Then $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ are said to be *tableau bisimilar* (notation: $\mathcal{T}_\alpha \rightleftharpoons \mathcal{T}_\beta$) if there exists a binary relation $Z \subseteq (T_m \times T'_m) \cup (T_c \times T'_c)$ satisfying the following seven conditions:

**Root condition:** $(r, r') \in Z$.

**Prop condition:** For any $t \in T_m$ and $t' \in T'_m$, if $(t, t') \in Z$, then

$$L(t) \cap \mathsf{Lit}(\alpha) = L'(t') \cap \mathsf{Lit}(\beta).$$

Consequently $L(t)$ is consistent if and only if $L'(t')$ is consistent.

**Forth condition on modal nodes:** Take $t \in T_m$, $u \in T_c$ and $t' \in T'_m$ arbitrarily. If $(t, t') \in Z$ and $u \in C(t)$, then there exists $u' \in C'(t')$ such that $(u, u') \in Z$ (see Figure 7).

**Back condition on modal nodes:** The converse of the forth condition on modal nodes: Take $t \in T_m$, $t' \in T'_m$ and $u' \in T'_c$ arbitrarily. If $(t, t') \in Z$ and $u' \in C'(t')$, then there exists $u \in C(t)$ such that $(u, u') \in Z$.

**Forth condition on choice nodes:** Take $u \in T_c$, $t \in T_m$ and $u' \in T'_c$ arbitrarily. If $(u, u') \in Z$ and $t$ is near $u$, then there exists $t' \in T'_m$ such that $(t, t') \in Z$ and $t'$ is near $u'$ (see Figure 7).

**Back condition on choice nodes:** The converse of the forth condition on choice nodes: Take $u \in T_c$, $u' \in T'_c$ and $t' \in T'_m$ arbitrarily. If $(u, u') \in Z$ and $t'$ is near $u'$, then there exists $t \in T_m$ such that $(t, t') \in Z$ and $t$ is near $u$.

**Parity condition:** Let $\xi$ and $\xi'$ be infinite branches of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, respectively. We say that $\xi$ and $\xi'$ are *associated* with each other if the $k$-th modal nodes $\xi[i_k]$ and $\xi'[i'_k]$ satisfy $(\xi[j_k], \xi'[j'_k]) \in Z$ for any $k \in \omega \setminus \{0\}$. For any $\xi$ and $\xi'$ which are associated with each other, we have $\xi$ is even if and only if $\xi'$ is even.

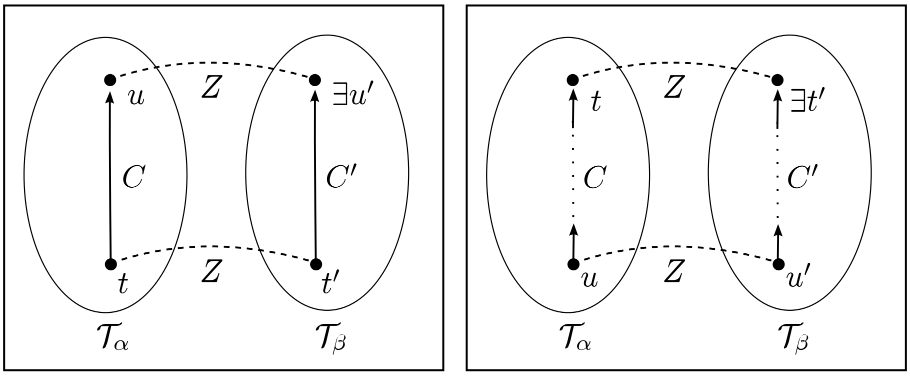


Figure 7: The forth conditions.

If $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ are tableau bisimilar with $Z$, then $Z$ is called a *tableau bisimulation* from $\mathcal{T}_\alpha$ to $\mathcal{T}_\beta$.

We accept the following Lemma 4.10 without proof (see, e.g., the literature [6]).

**Lemma 4.10.** *Let $\alpha$, $\beta$ be well-named formulas. If $\mathcal{T}_\alpha \rightleftharpoons \mathcal{T}_\beta$, then $\models \alpha \leftrightarrow \beta$.*

**Theorem 4.11 (Janin and Walukiewicz [4]).** *For any well-named formula $\alpha$, we can construct an automaton normal form $\mathsf{anf}(\alpha)$ such that $\mathcal{T}_\alpha \rightleftharpoons \mathcal{T}_{\mathsf{anf}(\alpha)}$ for some tableau $\mathcal{T}_\alpha$ for $\alpha$.*[10]

*Proof.* Let $\mathcal{T}'_\alpha = (T, C, r, L)$ be a tableau for a given formula $\alpha$, let $\mathcal{PA}_\alpha = (Q, \Sigma, \Delta, q_0, \mathsf{pri})$ be a parity automaton that is given in Subsection 3.3. For any node $t \in T$, set

$$\Delta(\vec{L}(\vec{rt}), q_0) = q_t = \langle J_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t \rangle.$$

First, we construct a tableau-like structure $\mathcal{TB}_\alpha = (T_b, C_b, r_b, L_b, B_b)$ called a *tableau with back edge* from $\mathcal{T}'_\alpha$ as follows:

- The node $t \in T$ is called a *loop node* if;

  (♠) There is a proper ancestor $t'$ such that $q_t = q_{t'}$, and

  (♡) for any $u \in T$ such that $u \in C^*(t')$ and $t \in C^*(u)$, we have $\mathsf{pri}(q_u) \leq \mathsf{pri}(q_t)(= \mathsf{pri}(q_{t'}))$.

  In this situation, the node $t'$ is called a *return node* of $t$. Note that for any infinite branch $\xi$ of $\mathcal{T}_\alpha$, there exists a loop node on $\xi$ since $Q$ is finite. We define the set $T_b$ of nodes as follows:

  $$T_b := \{t \in T \mid \text{for any proper ancestor } t' \text{ of } t,\ t' \text{ is } \textit{not} \text{ a loop node}\}$$

  Intuitively speaking, we trace the nodes on each branch from the root and as soon as we arrive at a return node, we cut off the former branch from the tableau.

- Set $C_b := C|_{T_b \times T_b}$, $r_b := r$ and $L_b := L|_{T_b}$.

- $B_b := \{(t, t') \in T_b \times T_b \mid t \text{ is a loop node and } t' \text{ is a return node of } t\}$. An element of $B_b$ is called *back edge.*

By König's lemma, we can assume that $\mathcal{TB}_\alpha$ is a finite structure because it has no infinite branches. The tableau with back edge is very similar to the basic tableau. In fact, the unwinding $\mathsf{UNW}_{r_b}(\mathcal{TB}_\alpha)$ is a tableau for $\alpha$. Therefore, we use the terminology and concepts of the tableau, such as the concept of the parity of the sequence of nodes. From the definition of loop and return nodes (particularly Condition (♡)), we can assume that

> (†): Let $\xi$ be an infinite $(C_b \cup B_b)$-sequence and let $t \in T_b$ be the return node which appears infinitely often in $\xi$ and is nearest to the root of all such return nodes. Then, $\xi$ is even if and only if $\mathsf{pri}(q_t)$ is even.

[10] Note that the tableau of $\mathsf{anf}(\alpha)$ is uniquely determined.

Next, we assign an automaton normal form $\mathsf{anf}(t)$ to each node $t \in T_b$ by using top-down fashion:

**Base step:** Let $t \in T_b$ be a leaf. If $t$ is not a loop node, then $t$ must be a modal node with an inconsistent label or contain no formula of the form $\nabla\Phi$. In both cases, we assign $\mathsf{anf}(t) := \bigwedge_{1\leq k\leq i} l_k$ where $\{l_1, \dots, l_i\} = L_b(t) \cap \mathsf{Lit}(\alpha)$. If $t$ is a loop node, we take $x_t \in \mathsf{Prop} \setminus \mathsf{Sub}(\varphi)$ uniquely for each such leaf and we set $\mathsf{anf}(t) := x_t$.

**Inductive step I:** Suppose $t \in T_b$ is a $(\nabla)$-node where $t$ is labeled by $\{\nabla\Psi_1, \dots, \nabla\Psi_i, l_1, \dots, l_j\}$ with $l_1, \dots, l_j \in \mathsf{Lit}(\alpha)$, and we have already assigned the automaton normal form $\mathsf{anf}(u)$ for each child $u \in C_b(t)$. In this situation, we first assign $\mathsf{anf}^-(t)$ to $t$ as follows:

$$\begin{aligned}\mathsf{anf}^-(t) &:= \nabla\{\mathsf{anf}(u) \mid u \in C_b(t)\} \wedge \left(\bigwedge_{1\leq k\leq j} l_k\right) \\ &= \left(\bigwedge_{u\in C_b(t)} \Diamond\mathsf{anf}(u)\right) \wedge \Box\left(\bigvee_{1\leq k\leq i}\left(\bigvee_{u\in C_b^{(k)}(t)} \mathsf{anf}(u)\right)\right) \wedge \left(\bigwedge_{1\leq k\leq j} l_k\right) \qquad (7)\end{aligned}$$

where $C_b^{(k)}(t)$ denotes the set of all children $u \in C_b(t)$ such that $\nabla\Psi_k$ is reduced to some $\psi_k \in \Psi_k$ between $t$ and $u$. That is, we designate the order of disjunction in $\mathsf{anf}^-(t)$ for technical reasons (see Remark 4.12). If $t$ is not a return node, then we set $\mathsf{anf}(t) := \mathsf{anf}^-(t)$. Alternatively, if $t$ is a return node, then let $t_1, \dots, t_n$ be all the loop nodes such that $(t_k, t) \in B_b$ $(1 \leq k \leq n)$. We set

$$\eta_t := \begin{cases} \mu & \text{If } \mathsf{pri}(t)(= \mathsf{pri}(t_1) = \cdots = \mathsf{pri}(t_n)) \equiv 1 \pmod 2 \\ \nu & \text{If } \mathsf{pri}(t)(= \mathsf{pri}(t_1) = \cdots = \mathsf{pri}(t_n)) \equiv 0 \pmod 2 \end{cases} \qquad (8)$$

In this case we define $\mathsf{anf}(t)$ as $\mathsf{anf}(t) := \eta_t x_{t_1}. \dots \eta_t x_{t_n}.\mathsf{anf}^-(t)$.

**Inductive step II:** Suppose $t \in T_b$ is a $(\vee)$-node where, for both children $u_1, u_2 \in C_b(t)$, we have already assigned the automaton normal forms $\mathsf{anf}(u_1)$ and $\mathsf{anf}(u_2)$, respectively. If $t$ is not a return node, then we set $\mathsf{anf}(t) := \mathsf{anf}(u_1) \vee \mathsf{anf}(u_2)$. Suppose $t$ is a return node. Let $t_1, \dots, t_n$ be all the loop nodes such that $(t_k, t) \in B_b$ $(1 \leq k \leq n)$. In this case, $\eta_t$ is defined in the same way as (8) and we define $\mathsf{anf}(t)$ as $\mathsf{anf}(t) := \eta_t x_{t_1}. \dots \eta_t x_{t_n}.\big(\mathsf{anf}(u_1) \vee \mathsf{anf}(u_2)\big)$.

**Inductive step III:** Suppose $t \in T_b$ is a $(\wedge)$-, $(\eta)$- or $(\mathsf{Regeneration})$-node where we have already assigned the automaton normal form $\mathsf{anf}(u)$ for the child $u \in C_b(t)$. If $t$ is not a return node, then we assign $\mathsf{anf}(t) := \mathsf{anf}(u) \wedge \top$. If $t$ is a return node and $t_1, \dots, t_n$ are all the loop nodes such that $(t_k, t) \in B_b$ $(1 \leq k \leq n)$, then, $\eta_t$ is defined in the same way as (8), and we define $\mathsf{anf}(t)$ as $\mathsf{anf}(t) := \eta_t x_{t_1}. \dots \eta_t x_{t_n}.\, \mathsf{anf}(u)$.

We take $\mathsf{anf}(\alpha) := \mathsf{anf}(r_b)$.

Consider the structure $(T_b, C_b, r_b, \mathsf{anf}, B_b)$. We intuit that this structure is almost a tableau with back edge for $\mathsf{anf}(\alpha)$. To clarify this intuition, we give a structure $\mathcal{TB}_{\mathsf{anf}(\alpha)} = (\widehat{T}, \widehat{C}, \widehat{r}, \widehat{L}, \widehat{B})$ by applying the following four steps of procedure re-formatting $(T_b, C_b, r_b, \mathsf{anf}, B_b)$ so that $\mathcal{TB}_{\mathsf{anf}(\alpha)}$ can be seen as a proper tableau with back edge. At the same time, we define the relation $Z^+ \subseteq T_b \times \widehat{T}$.

**Step I (insert $(\eta)$-nodes)** Initially, we set $(\widehat{T}, \widehat{C}, \widehat{r}, \widehat{L}, \widehat{B}) := (T_b, C_b, r_b, \widehat{L}, B_b)$ where $\widehat{L}(t) := \{\mathsf{anf}(t)\}$, and set $Z^+ := \{(t,t) \mid t \in T_b\}$. Let $t \in \widehat{T}$ be a return node where $t_1, \dots, t_n$ are all the loop nodes such that $(t_k, t) \in \widehat{B}$ $(1 \leq k \leq n)$. Then, we insert the $(\eta)$-nodes $u_1, \dots, u_n$ between $t$ and its children in such a way that

$$\mathsf{anf}(t) = \eta_t x_{t_1}.\eta_t x_{t_2}. \dots \eta_t x_{t_n}.\beta(x_{t_1}, \dots, x_{t_n})$$

is reduced to $\beta(x_{t_1}, \dots, x_{t_n})$ from $u_1$ to $u_n$.[11] Moreover, we expand the relation $Z^+$ by adding $\{(t, u_k) \mid 1 \leq k \leq n\}$. For example, if $t$ is a $(\vee)$-node in $\mathcal{TB}_\alpha$ such that $\{v_1, v_2\} = C_b(t)$, then our

[11] In other words, we add $u_1, \dots, u_n$ into $\widehat{T}$, add $(t, u_1), (u_1, u_2), \dots, (u_{n-1}, u_n)$ and $\{(u_n, u) \mid u \in \widehat{C}(t)\}$ into $\widehat{C}$, discard $\{(t, u) \mid u \in \widehat{C}(t)\}$ from $\widehat{C}$, and expand $\widehat{L}$ to $u_1, \dots, u_n$ appropriately.

procedure would be as follows:

$$\frac{\mathsf{anf}(v_1) \ \mid \ \mathsf{anf}(v_2)}{\eta_t x_{t_1}.\eta_t x_{t_2}.\ldots\eta_t x_{t_n}.(\mathsf{anf}(v_1)\vee\mathsf{anf}(v_2))} \quad\Rightarrow\quad \dfrac{\begin{array}{c}\dfrac{\mathsf{anf}(v_1) \ \mid \ \mathsf{anf}(v_2)}{\mathsf{anf}(v_1)\vee\mathsf{anf}(v_2)}\,(\vee)\\ \vdots\ (\eta)\\ \eta_t x_{t_2}.\ldots\eta_t x_{t_n}.(\mathsf{anf}(v_1)\vee\mathsf{anf}(v_2))\end{array}}{\eta_t x_{t_1}.\eta_t x_{t_2}.\ldots\eta_t x_{t_n}.(\mathsf{anf}(v_1)\vee\mathsf{anf}(v_2))}\,(\eta)$$

**Step II (insert** $(\wedge)$**-nodes)** Let $t\in\widehat{T}$ be a node which is labeled by;

$$\nabla\{\mathsf{anf}(u)\mid u\in\widehat{C}(t)\}\wedge\left(\bigwedge_{1\le k\le j} l_k\right).$$

Then, we insert the $(\wedge)$-nodes $u_0,\ldots,u_i$ between $t'$ and its children (i.e., the nodes of $\widehat{C}(t)$) and label such $u_1,\ldots,u_j$ as below:

$$\frac{\mathsf{anf}(u) \ \mid \ u\in\widehat{C}(t)}{\nabla\{\mathsf{anf}(u)\mid u\in\widehat{C}(t)\}\wedge\left(\bigwedge_{1\le k\le j} l_k\right)} \quad\Rightarrow\quad \dfrac{\begin{array}{c}\dfrac{\mathsf{anf}(u) \ \mid \ u\in\widehat{C}(t)}{\nabla\{\mathsf{anf}(u)\mid u\in\widehat{C}(t)\}, l_1,\ldots,l_j}\,(\nabla)\\ \vdots\ (\wedge)\\ \nabla\{\mathsf{anf}(u)\mid u\in\widehat{C}(t)\},\left(\bigwedge_{1\le k\le j} l_k\right)\end{array}}{\nabla\{\mathsf{anf}(u)\mid u\in\widehat{C}(t)\}\wedge\left(\bigwedge_{1\le k\le j} l_k\right)}\,(\wedge)$$

Further, we expand the relation $Z^+$ by adding $\{(t,u_k)\mid 1\le k\le j\}$.

**Step III (revise the back edges)** Let $t_k$ with $1\le k\le n$ be the loop node, and $t$ be the return node of $t_k$ such that

$$\mathsf{anf}(t_k)=x_{t_k}$$
$$\mathsf{anf}(t)=\eta_t x_{t_1}.\eta_t x_{t_2}.\ldots\eta_t x_{t_n}.\beta(x_{t_1},\ldots,x_{t_n}).$$

If $2\le k$, then we delete $(t_k,t)$ from $\widehat{B}$ and add $(t_k,u_k)$ into $\widehat{B}$ where $u_k$ is the unique nodes satisfying;

$$\widehat{L}(u_k)=\{\eta_t x_{t_k}.\ldots\eta_t x_{t_n}.\beta(x_{t_1},\ldots,x_{t_n})\}.$$

By this revising procedure, for any loop node $t$ and its return node $u$, $\widehat{L}(t)$ and $\widehat{L}(u)$ form the (Regeneration)-rule of $\mathsf{anf}(\alpha)$.

**Step IV (add top to label)** Suppose $t\in\widehat{T}$ and its child $u$ are labeled as follows;

$$\frac{\mathsf{anf}(u)}{\mathsf{anf}(u)\wedge\top}$$

Then, we add $\top$ to $\widehat{L}(v)$ where $v\in(\widehat{C}\cup\widehat{B})^+(t)$ such that, between the $(\widehat{C}\cup\widehat{B})$-path from $t$ to $v$, there does not exist a $(\nabla)$-node. By this adding procedure, such a $t$ becomes a proper $(\wedge)$-node.

The structure $\mathcal{TB}_{\mathsf{anf}(\alpha)}=(\widehat{T},\widehat{C},\widehat{r},\widehat{L},\widehat{B})$ repaired by the above four procedures can be seen as a tableau with back edge for $\mathsf{anf}(\alpha)$ in the sense that the following two assertions hold:

($\clubsuit$) The unwinding $\mathsf{UNW}_{\widehat{r}}(\mathcal{TB}_{\mathsf{anf}(\alpha)})$ is a tableau of $\mathsf{anf}(\alpha)$.

($\diamondsuit$) Let $\widehat{\xi}$ be an infinite $(\widehat{C}\cup\widehat{B})$-sequence and let $\widehat{t}\in\widehat{T}$ be the return node which appears infinitely often in $\widehat{\xi}$ and is nearest to the root of all such return nodes. Then $\widehat{\xi}$ is even if and only if $\widehat{L}(\widehat{t})$ includes a $\mu$-formula.

Set $Z := Z^+|_{((T_b)_m \times \widehat{T}_m) \cup ((T_b)_c \times \widehat{T}_c)}$. If we extend the relation $Z$ to the pair of nodes of $\mathsf{UNW}_r(\mathcal{TB}_\alpha)$ and $\mathsf{UNW}_{\widehat{r}}(\mathcal{TB}_{\mathsf{anf}(\alpha)})$, then $Z$ clearly satisfies the root condition, prop condition, back conditions and forth conditions. Moreover, from (†) and ($\diamondsuit$), we can assume that $Z$ satisfies the Parity condition. Therefore, we have $\mathsf{UNW}_r(\mathcal{TB}_\alpha) \rightleftharpoons \mathsf{UNW}_{\widehat{r}}(\mathcal{TB}_{\mathsf{anf}(\alpha)})$, and so $\mathcal{T}_\alpha := \mathsf{UNW}_r(\mathcal{TB}_\alpha)$ and $\mathsf{anf}(\alpha)$ satisfy the required condition. □

**Remark 4.12.** Let $\mathsf{Sub}'(\mathsf{anf}(\alpha))$ be the set of subformulas of $\mathsf{anf}(\alpha)$ which contains some bound variables. From the relation $Z^+$ constructed in the proof of Theorem 4.11, we can construct a function $f$ from $\mathsf{Sub}'(\mathsf{anf}(\alpha))$ to $\mathcal{P}(\mathsf{Sub}(\alpha))$ naturally because of the following:

- for any $\widehat{\beta} \in \mathsf{Sub}'(\mathsf{anf}(\alpha))$, there exists a unique $\widehat{t} \in \widehat{T}$ such that $\widehat{\beta} \in \widehat{L}(\widehat{t})$; and
- for any $\widehat{t} \in \widehat{T}$ there exists a unique $t \in T_b$ such that $(t, \widehat{t}) \in Z^+$.

Therefore, if we define $f(\widehat{\beta}) := L(t)$ where $\widehat{\beta} \in \widehat{L}(\widehat{t})$ and $(t, \widehat{t}) \in Z^+$, then the function $f$ is well-defined. Moreover, let $t \in T_b$ be a ($\nabla$)-node such that $L_b(t) = \{\nabla\Psi_1, \dots, \nabla\Psi_i, l_1, \dots, l_j\}$. Then, we expand $f$ to the formula $\chi_1$ and $\chi_2$ such that

$$\mathsf{anf}(u) \leq \chi_1 \leq \bigvee_{u \in C_b^{(k)}(t)} \mathsf{anf}(u) \leq \chi_2 \leq \left( \bigvee_{1 \leq k \leq i} \left( \bigvee_{u \in C_b^{(k)}(t)} \mathsf{anf}(u) \right) \right),$$

for every $k$ where $1 \leq k \leq i$ and for every $u \in C_b^{(k)}(t)$. Now, we define $f(\chi_2)$ as

$$f(\chi_2) := \left\{ \bigvee \Psi_n \mid 1 \leq n \leq i \right\}.$$

Next, we note that for any $u \in C_b^{(k)}(t)$ there is a unique $\psi_k \in \Psi_k$ such that $\nabla\Psi_k$ is reduced to $\psi_k$. We denote such a $\psi_k$ by $\mathsf{cor}(u)$. Suppose $\chi_1 = \bigvee_{u \in X^{(k)}} \mathsf{anf}(u)$ where $X^{(k)} \subseteq C_b^{(k)}(t)$. Then we define $f(\chi_1)$ as;

$$f(\chi_1) := \left\{ \bigvee \Psi_n \mid 1 \leq n \leq i,\ n \neq k \right\} \cup \left\{ \bigvee_{u \in X^{(k)}} \mathsf{cor}(u) \right\}.$$

Recalling Equation (7), the reason we designated the order of disjunction in $\mathsf{anf}(t)$ is that, in conjunction with above definition of $f$, we obtain the following useful property:

**(Corresponding Property):** Consider the section of the tableau which has the root labeled by

$$\left( \bigvee_{1 \leq k \leq i} \left( \bigvee_{u \in C_b^{(k)}(t)} \mathsf{anf}(u) \right) \right),$$

and every leaf labeled by some $\mathsf{anf}(u)$. Then, for any node $u$ and its children $v_1$ and $v_2$ we have (i) $f(L(u)) = f(L(v_1)) = f(L(v_2))$ or, (ii) $f(L(u))$, $f(L(v_1))$ and $f(L(v_2))$ forming a ($\vee$)-rule.

Let us confirm the above property by observing a concrete example as depicted in Figure 8. In this example, the root and its children satisfy (i), and the child of the root and its children form a ($\vee$)-rule. Thus, (ii) is satisfied.

The function $f$ will be used in the proof of Part 4 of Lemma 5.7.

**Corollary 4.13.** *For any well-named formula $\alpha$, we can construct an automaton normal form* $\mathsf{anf}(\alpha)$ *which is semantically equivalent to $\alpha$. Moreover, for any $x \in \mathsf{Free}(\alpha)$ which occurs only positively in $\alpha$, it holds that $x \in \mathsf{Free}(\mathsf{anf}(\alpha))$ and $x$ occurs only positively in* $\mathsf{anf}(\alpha)$.

*Proof.* This is an immediate consequence of Lemma 4.10 and Theorem 4.11. □

# 5 Completeness

This section is the final section of this article and includes the main part. In Subsection 5.1, we give the concept of *tableau consequence* and show Claim (g); that may be the most difficult to understand in Walukiewicz [8]. In Subsection 5.2, we prove the completeness of $\mathsf{Koz}$ by proving Claim (h) and (d), in that order.

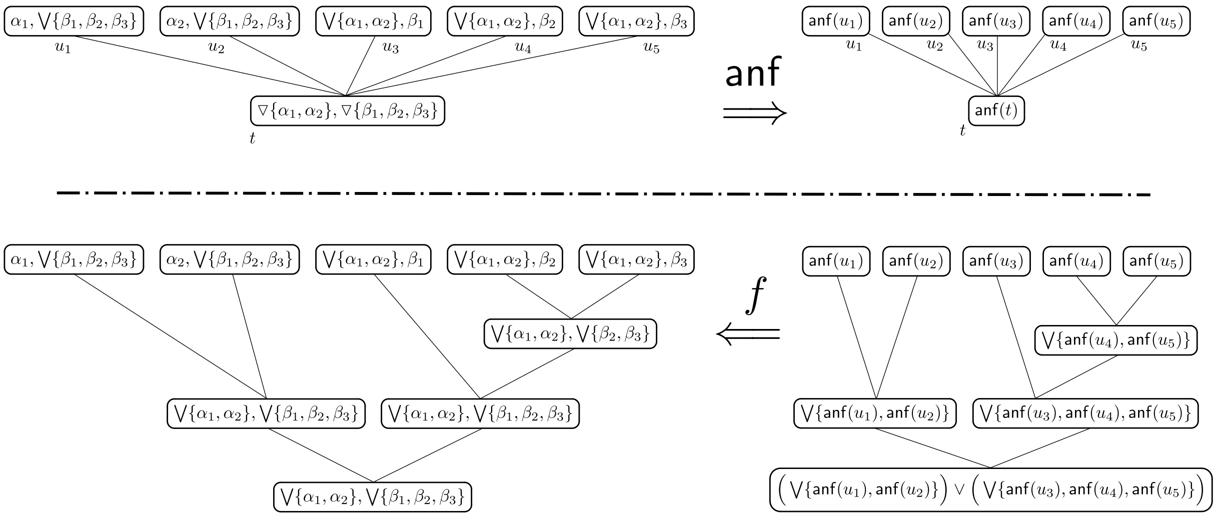


Figure 8: An example of the corresponding property.

## 5.1 Tableau consequence

First, we extend the definition of tableau for technical reasons.

**Definition 5.1 (An extension of tableau).** Let $\varphi$ be a well-named formula. The rule of a *extended tableau* for $\varphi$ is obtained by adding the following three rules to the rule of tableau:

$$\frac{\Gamma}{\Gamma}\ (\epsilon_1) \qquad \frac{\Gamma \ \mid \ \Gamma}{\Gamma}\ (\epsilon_2)$$

$$\frac{\bigvee\Psi_1,\dots,\bigvee\Psi_i \mid \cdots \mid \bigvee\Psi_1,\dots,\bigvee\Psi_i}{\nabla\Psi_1,\dots,\nabla\Psi_i, l_1,\dots,l_j}\ (\nabla_e)$$

where in the $(\nabla_e)$-rule, $l_1,\dots,l_j \in \mathsf{Lit}(\varphi)$ and, the label of premises are all the same (i.e., $\{\bigvee\Psi_1,\dots,\bigvee\Psi_i\}$), and the number of premises is an arbitrary finite number.

An *extended tableau* for $\varphi$ is the structure defined as a tableau for $\varphi$, but satisfying the following additional clause:

4. For any infinite branch $\xi$ of an extended tableau $\mathcal{T}_\varphi$, $\{n \in \omega \mid \xi[n]$ is $(\nabla)$-node or $(\nabla_e)$-node$\}$ is an infinite set.

Clause 4 restrains a branch that does not reach any modal node eternally by infinitely applying $(\epsilon_1)$ and $(\epsilon_2)$.

**Remark 5.2.** A tableau can be considered a special case of an extended tableau, in which the extended rules are not used. Various concepts for tableau, such as trace, parity and tableau bisimulation, can be introduced into this extended tableau as well. Thus, we apply these concepts and results freely to this new structure.

**Definition 5.3 (Tableau consequence).** Let $\mathcal{T}_\alpha = (T, C, r, L)$ and $\mathcal{T}_\beta = (T', C', r', L')$ be two extended tableaux for some well-named formula $\alpha$ and $\beta$. Let $T_m$ and $T'_m$ be the set of modal nodes of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, and let $T_c$ and $T'_c$ be the set of choice nodes of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, respectively. Then $\mathcal{T}_\beta$ is called a *tableau consequence* of $\mathcal{T}_\alpha$ (notation: $\mathcal{T}_\alpha \rightharpoonup \mathcal{T}_\beta$) if there exists a binary relation $Z \subseteq (T_m \times T'_m) \cup (T_c \times T'_c)$ satisfying the following six conditions (here, the condition of the tableau consequence is similar to the condition of tableau bisimulation so we have illustrated the differences between these two conditions using underlines):

**Root condition:** $(r, r') \in Z$.

**Prop condition:** For any $t \in T_m$ and $t' \in T'_m$, if $(t, t') \in Z$, then

$$L(t) \cap \mathsf{Lit}(\alpha) \supseteq L'(t') \cap \mathsf{Lit}(\beta).$$

Consequently, $L(t)$ is consistent only if $L'(t')$ is consistent.

**Forth condition on modal nodes:** Take $t, u \in T_m$ and $t' \in T'_m$ arbitrarily. If $(t, t') \in Z$ and $u$ is a next modal node of $t$, then $C'(t') = \emptyset$ or there exists $u' \in T'_m$ which is a next modal node of $t'$ such that $(u, u') \in Z$.

**Back condition on modal nodes:** Take $t \in T_m$, $t' \in T'_m$ and $u' \in T'_c$ arbitrarily. If $(t, t') \in Z$ and $u' \in C'(t')$, then $C(t) = \emptyset$ or there exists $u \in C(t)$ such that $(u, u') \in Z$.

**Forth condition on choice nodes:** Take $u \in T_c$, $t \in T_m$ and $u' \in T'_c$ arbitrarily. If $(u, u') \in Z$ and $t$ is near $u$, then there exists $t' \in T'_m$ such that $(t, t') \in Z$ and $t'$ is near $u'$.

**Back condition on choice nodes:** No condition.

**Parity condition:** Let $\xi$ and $\xi'$ be infinite branches of $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ respectively. If $\xi$ and $\xi'$ are associated with each other, then $\xi$ is even if $\xi'$ is even.

A relation $Z$ which satisfies the above six conditions called *tableau consequence relation* from $\mathcal{T}_\alpha$ to $\mathcal{T}_\beta$.

**Remark 5.4.** As will be shown in Lemma 4.10, if $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ are tableau bisimilar, then, $\alpha$ and $\beta$ are semantically equivalent. However, the reverse is not applied. For example, consider the following two tableaux, say $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$\dfrac{\dfrac{\dfrac{p,q \quad \mid \quad p,r}{p,q\vee r}\,(\vee)}{p\wedge(q\vee r),q\vee r}\,(\wedge)}{(p\wedge(q\vee r))\wedge(q\vee r)}\,(\wedge) \qquad \dfrac{\dfrac{\dfrac{\dfrac{p,q \quad \mid \quad p,q,r}{p,q\vee r,q}\,(\vee)}{p\wedge(q\vee r),q}\,(\wedge) \quad \mid \quad \dfrac{\dfrac{p,q,r \quad \mid \quad p,r}{p,q\vee r,r}\,(\vee)}{p\wedge(q\vee r),r}\,(\wedge)}{p\wedge(q\vee r),q\vee r}\,(\vee)}{(p\wedge(q\vee r))\wedge(q\vee r)}\,(\wedge)$$

In this example, even $\mathcal{T}_1$ and $\mathcal{T}_2$ are tableaux for the same formula $(p \wedge (q \vee r)) \wedge (q \vee r)$, there does not exist a tableau bisimulation between them. Because, $\mathcal{T}_2$ has leaves labeled by $\{p, q, r\}$ but $\mathcal{T}_1$ does not.

On the other hand, we can assume that $\mathcal{T}_2 \rightharpoonup \mathcal{T}_1$. Suppose $t$ is a node of some tableau labeled by $\{\gamma\} \cup \Gamma$ and, $u$ is a its child labeled by $\{\gamma'\} \cup \Gamma$. Then, there exists two possibilities; $\gamma' \in \Gamma$ or $\gamma' \notin \Gamma$. We say a collision occurred between $t$ and $u$ if $\gamma' \in \Gamma$. In the above example, we can find collisions in $\mathcal{T}_1$ but cannot in $\mathcal{T}_2$. In general, if we construct a tableau $\mathcal{ST}_\varphi$ for a given formula $\varphi$ so that collisions occur as many as possible, then, we have $\mathcal{T}_\varphi \rightharpoonup \mathcal{ST}_\varphi$ for any tableau $\mathcal{T}_\varphi$ for $\varphi$. To denote this fact correctly, we introduce the following definition and lemma.

**Definition 5.5** (**Small tableau).** A well-named formula $\varphi$ and a set $\Gamma \subseteq \mathsf{Sub}(\varphi)$ are given. For a formula $\gamma \in \Gamma$, a closure of $\gamma$ (denotation: $\mathsf{cl}(\gamma)$) is defined as follows:

- $\gamma \in \mathsf{cl}(\gamma)$.
- If $\alpha \circ \beta \in \mathsf{cl}(\gamma)$, then $\alpha, \beta \in \mathsf{cl}(\gamma)$ where $\circ \in \{\vee, \wedge\}$.
- If $\eta_x x.\varphi_x(x) \in \mathsf{cl}(\gamma)$, then $\varphi_x(x) \in \mathsf{cl}(\gamma)$.
- If $x \in \mathsf{cl}(\gamma) \cap \mathsf{Bound}(\varphi)$, then $\varphi_x(x) \in \mathsf{cl}(\gamma)$.

In other words, $\mathsf{cl}(\gamma)$ is a set of all formulas $\delta$ such that for any tableau $\mathcal{T}_\varphi = (T, C, r, L)$ and its node $t \in T$, if $\gamma \in L(t)$, then, there is a descendant $u \in C^*(t)$ near $t$ and a trace $\mathsf{tr}$ on the $C$-sequence from $t$ to $u$ where $\mathsf{tr}[1] = \gamma$ and $\mathsf{tr}[|\mathsf{tr}|] = \delta$. We say $\gamma$ is *reducible* in $\Gamma$ if, for any $\gamma' \in \Gamma \setminus \{\gamma\}$, we have $\gamma \notin \mathsf{cl}(\gamma')$. A tableau $\mathcal{ST}_\varphi = (T, C, r, L)$ is said *small* if for any node $t \in T$ which is not modal, the reduced formula $\gamma \in L(t)$ between $t$ and its children is reducible in $L(t)$.

**Lemma 5.6.** *For any well-named formula $\varphi$, we can construct a small tableau $\mathcal{ST}_\varphi$ for $\varphi$. Moreover, for any extended tableau $\mathcal{T}_\varphi$ for $\varphi$, we have $\mathcal{T}_\varphi \rightharpoonup \mathcal{ST}_\varphi$.*

*Proof.* Let $\varphi$ be a well-named formula. Then, it is enough to show that for any $\Gamma \subseteq \mathsf{Sub}(\varphi)$ which is not modal, there exists a reducible formula $\gamma \in \Gamma$. Suppose, moving toward a contradiction, that there exists $\Gamma \subseteq \mathsf{Sub}(\varphi)$ which is not modal and does not include any reducible formula. Take a formula $\gamma_1 \in \Gamma$ such that $\mathsf{cl}(\gamma_1) \supsetneq \{\gamma_1\}$. Since $\gamma_1$ is not reducible in $\Gamma$, there exists $\gamma_2 \in \Gamma \setminus \{\gamma_1\}$ such that $\gamma_1 \in \mathsf{cl}(\gamma_2)$. Since $\gamma_2$ is not reducible in $\Gamma$, there exists $\gamma_3 \in \Gamma \setminus \{\gamma_2\}$ such that $\gamma_2 \in \mathsf{cl}(\gamma_3)$. And so forth, we obtain the sequence $\langle \gamma_n \mid n \in \omega \setminus \{0\} \rangle$ such that $\gamma_{n+1} \in \Gamma \setminus \{\gamma_n\}$ and $\gamma_n \in \mathsf{cl}(\gamma_{n+1})$ for any $n \in \omega \setminus \{0\}$. Since $|\Gamma|$ is finite, there exists $i, j \in \omega$ such that $1 \leq i < j$ and $\gamma_i = \gamma_j$. Consider the tableau $\mathcal{T}_\varphi = (T, C, r, L)$ and its node $t \in T$ such that $\gamma_j \in L(t)$. Then, from the definition of the closure $\mathsf{cl}$, there exists a trace $\mathsf{tr}$ on $\pi$ such that:

($\heartsuit$) $\pi$ is a finite $C$-sequence starting at $t$ where $(\triangledown)$-rule does not applied between $\pi$.

($\clubsuit$) $\mathsf{tr}[1] = \mathsf{tr}[|\mathsf{tr}|] = \gamma_j$.

On the other hand, since $\varphi$ is well-named, for any bound variable $x \in \mathsf{Bound}(\varphi)$, $x$ is in the scope of some modal operator (cover modality) in $\varphi_x(x)$. Thus we have:

($\spadesuit$) For any trace $\mathsf{tr}$ on $\pi$, if ($\clubsuit$) is satisfied, then $\pi$ includes a $(\triangledown)$-node or $(\triangledown_w)$-node.

($\heartsuit$) and ($\spadesuit$) contradict each other. The proof of the second half of the lemma is left as a reader's exercise. □

The next lemma states some important properties of the tableau consequence; where the proof of the lemma is easier to understand than Walukiewicz's proof, and is the main contribution of this article.

**Lemma 5.7.** *Let $\alpha$, $\beta$, $\gamma$ and $\varphi(x)$ be well-named formulas where $x$ appears only positively and in the scope of some modality in $\varphi(x)$. Then, we have:*

1. *If $\mathcal{T}_\alpha \rightleftharpoons \mathcal{T}_\beta$, then $\mathcal{T}_\alpha \rightharpoonup \mathcal{T}_\beta$, for any extended tableaux $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$.*
2. *If $\mathcal{T}_\alpha \rightharpoonup \mathcal{T}_\beta$ and $\mathcal{T}_\beta \rightharpoonup \mathcal{T}_\gamma$, then $\mathcal{T}_\alpha \rightharpoonup \mathcal{T}_\gamma$, for any extended tableaux $\mathcal{T}_\alpha$, $\mathcal{T}_\beta$ and $\mathcal{T}_\gamma$.*
3. *For any extended tableau $\mathcal{T}_{\varphi(\mu\vec{x}.\varphi(\vec{x}))}$, there exists an extended tableau $\mathcal{T}_{\mu\vec{x}.\varphi(\vec{x})}$ such that $\mathcal{T}_{\varphi(\mu\vec{x}.\varphi(\vec{x}))} \rightharpoonup \mathcal{T}_{\mu\vec{x}.\varphi(\vec{x})}$.*
4. *For any extended tableau $\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))}$, there exists an extended tableau $\mathcal{T}_{\varphi(\alpha)}$ such that $\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))} \rightharpoonup \mathcal{T}_{\varphi(\alpha)}$.*

*Proof.* Part 1 and Part 2 are obvious from the definition.

$\boxed{\text{Part3}}$ First, we divide $\mathsf{Sub}(\varphi(\mu x.\varphi(x)))$ into two disjoint sets:

$$\mathsf{Sub}_1 := \{\alpha(\mu x.\varphi(x)) \mid \alpha(x) \in \mathsf{Sub}(\varphi(x))\} \setminus \{\mu x.\varphi(x)\}$$
$$\mathsf{Sub}_2 := \mathsf{Sub}(\varphi(\mu x.\varphi(x))) \setminus \mathsf{Sub}_1 = \mathsf{Sub}(\mu x.\varphi(x))$$

A function $f : \mathsf{Sub}(\varphi(\mu x.\varphi(x))) \to \mathsf{Sub}(\mu x.\varphi(x))$ is defined as follows:

$$f(\psi) := \begin{cases} \alpha(x) & \psi = \alpha(\mu x.\varphi(x)) \in \mathsf{Sub}_1, \\ \psi & \text{otherwise.} \end{cases}$$

Take an extended tableau $\mathcal{T}_{\varphi(\mu x.\varphi(x))} = (T, C, r, L)$ arbitrarily. Set $Z := \{(t,t) \mid t \in T\}$. If

$$\mathcal{T}_{\varphi(\mu x.\varphi(x))} \rightharpoonup (T, C, r, f \circ L) \tag{9}$$

with $Z$, then we are done. However unfortunately (9) is generally incorrect. $Z$ generally does not satisfy the parity condition among the requests for tableu consequences. Let's explain that with a concrete example. Suppose that $\xi$ is an infinite branch of $\mathcal{T}_{\varphi(\mu x.\varphi(x))}$ where there are only two traces, $\mathsf{tr}_1$ and $\mathsf{tr}_2$ on it. Moreover, suppose that $\mathsf{tr}_1$ is a trace on $\mathsf{Sub}_1$ (i.e., $\mathsf{Infinite}(\mathsf{tr}_1) \subseteq \mathsf{Sub}_1$), and $\mathsf{tr}_2$ is a trace on $\mathsf{Sub}_2$ (i.e., $\mathsf{Infinite}(\mathsf{tr}_2) \subseteq \mathsf{Sub}_2$). Note that

($\spadesuit$) $\mathsf{tr}_i$ is even (i.e., $\mathsf{tr}_i$ is a $\mu$-trace) $\Leftrightarrow$ $\vec{f}(\mathsf{tr}_i)$ is even (i.e., $\vec{f}(\mathsf{tr}_i)$ is a $\mu$-trace) $(i = 1, 2)$

holds from the definition of $f$. Suppose that $\vec{f}(\mathsf{tr}_1)$ and $\vec{f}(\mathsf{tr}_2)$ repeat merging and branching as shown in Figure 9. Suppose $\Omega_{\mu x.\varphi(x)}(\alpha_i) = i$ $(i = 1, 2, 3)$. Then, we have

$$\max \Omega_{\mu x.\varphi(x)}\big(\mathsf{Infinite}(\vec{f}(\mathsf{tr}_1))\big) = \max \Omega_{\mu x.\varphi(x)}\big(\mathsf{Infinite}(\vec{f}(\mathsf{tr}_2))\big) = 3.$$

Therefore, from ($\spadesuit$), it turns out that $\mathsf{tr}_1$ and $\mathsf{tr}_2$ are odd. Thus, $\vec{L}(\xi)$ is also odd. On the other hand, set $\mathsf{tr}_3 = \vec{f}(\mathsf{tr}_1[1, k-1])(\alpha_2\alpha_1)^\omega$ (i.e., $\mathsf{tr}_3$ is the trace represented by $\Rightarrow$ in Figure 9). Since $\max \Omega_{\mu x.\varphi(x)}\big(\mathsf{Infinite}(\vec{f}(\mathsf{tr}_3))\big) = 2$, $\overrightarrow{f \circ L}(\xi)$ is even. This means that $Z$ does not satisfy the parity condition.

It turns out that simply compositing $f$ and the label of $\mathcal{T}_{\varphi(\mu x.\varphi(x))}$ didn't work. The problem is that $\mathsf{tr}_1$ and $\mathsf{tr}_2$ may exist such that $\vec{f}(\mathsf{tr}_1)$ and $\vec{f}(\mathsf{tr}_2)$ repeat branching and merging infinitely often, and these may break the parity condition guaranteed by ($\spadesuit$). Therefore, we overcome this obstacle by using the **horizontal prunning** technique shown in Safra's construction. We will construct an extended tableau $\mathcal{T}_{\mu x.\varphi(x)}$ where $\mathcal{T}_{\varphi(\mu x.\varphi(x))} \rightharpoonup \mathcal{T}_{\mu x.\varphi(x)}$ holds by $Z$ in the following 5 steps:

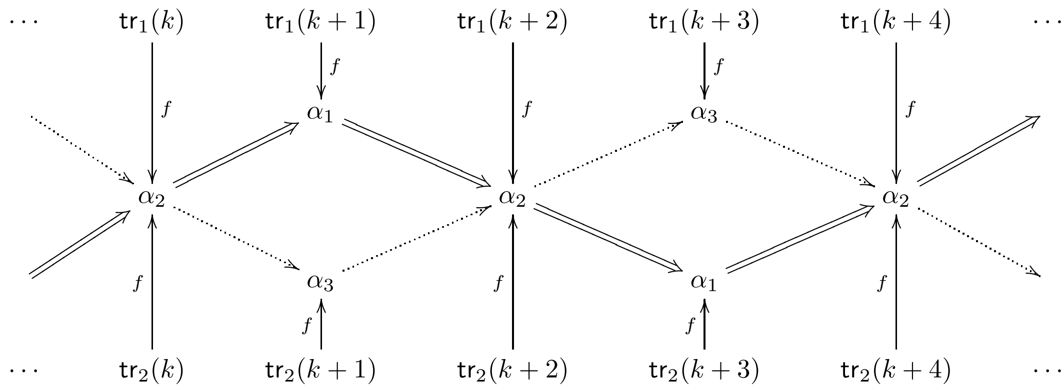


Figure 9: An example that does not sarisfy the parity condition.

Step 1: We define the Büchi automaton $\mathcal{BA}_{\varphi(\mu x.\varphi(x))} := \langle Q, \Sigma, q_0, \Delta, F \rangle$ as follows:

- $Q := \{(\Gamma, \gamma) \mid \Gamma \subseteq \mathsf{Sub}(\varphi(\mu x.\varphi(x))),\ \gamma \in \Gamma\}$.
- $\Sigma := \mathcal{P}(\mathsf{Sub}(\varphi(\mu x.\varphi(x))))$.
- $q_0 := \big(\{\varphi(\mu x.\varphi(x))\}, \varphi(\mu x.\varphi(x))\big)$.
- $\Delta(\Gamma', (\Gamma, \gamma)) := \big\{(\Gamma', \gamma') \mid \gamma' \in \mathsf{TR}_{\Gamma,\Gamma'}(\gamma)\big\}$.
- $F := \big\{q_0\big\} \bigcup \big\{(\Gamma, \mu x.\varphi(x)) \mid (\Gamma, \mu x.\varphi(x)) \in Q\big\}$.

Note that we are not interested in $\mathcal{L}(\mathcal{BA}_{\varphi(\mu x.\varphi(x))})$. $\mathcal{BA}_{\varphi(\mu x.\varphi(x))}$ is constructed only for the use of **horizontal prunning** in the Rabin automaton that will be constructed later.

Step 2: Convert nondeterministic Büchi automaton $\mathcal{BA}_{\varphi(\mu x.\varphi(x))}$ to deterministic Rabin automaton $\mathcal{RA}_{\varphi(\mu x.\varphi(x))}$ using Safra's construction. However, the following two points are changed from the construction described in Subsection 3.2:

- The automaton $\mathcal{BA}_{\varphi(\mu x.\varphi(x))}$ reads the alphabet $\{\varphi(\mu x.\varphi(x))\}$ in the initial state
$$q_0 = \big(\{\varphi(\mu x.\varphi(x))\}, \varphi(\mu x.\varphi(x))\big)$$
and transitions to the next state $q_0$ (as a result, the state does not change). Since $q_0 \in F$ and $\pi_1 = \big(N_{\varphi(\mu x.\varphi(x))}, \ldots, 3, 2, 1\big)$, normally, by **add new children**, we add 2 as a new child. Now change the child to be added from 2 to $N_{\varphi(\mu x.\varphi(x))}$.
- In the **initialize index appearence record**, abolish driving $j$ painted in red to the left end. Instead, change it so that it is driven to the left end excluding $N_{\varphi(\mu x.\varphi(x))}$ (see Figure 10):

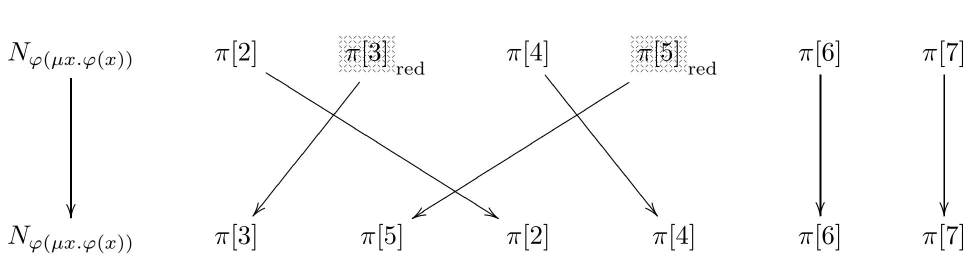


Figure 10: A change of initialize index appearence record.

Step 3: Let the automaton defined above be $\mathcal{RA}_{\varphi(\mu x.\varphi(x))} = \langle Q', \Sigma, q_0, \Delta', \{(A_j, R_j) \mid j \in J\}\rangle$. For a tableau node $t$, set $\Delta'(\vec{L}(\vec{rt}), q_0) := \langle S_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t\rangle$. In this situation, Safra's tree $\langle S_t, C_t, 1, l_t, \pi_t, \mathsf{col}_t\rangle$

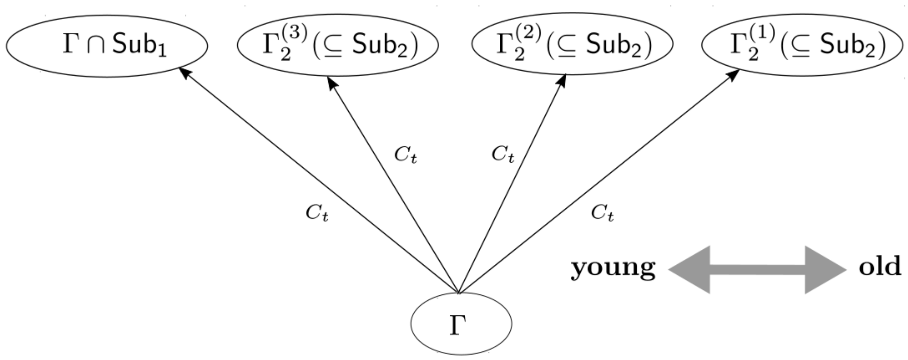


Figure 11: A state of automaton $\mathcal{RA}_{\varphi(\mu x.\varphi(x))}$.

looks like Figure 11.[12] That is, the youngest child of the root 1 is $N_{\varphi(\mu x.\varphi(x))}$, labeled with a subset of $\mathsf{Sub}_1$. The other children of the root 1 are labeled with a subset of $\mathsf{Sub}_2$.

Step 4: For each node $t$, we will define a labeled tree $\langle J_t, C_t, 1, l'_t, \pi_t, \mathsf{col}_t\rangle$ inductively from the root to the leaf; where $l'_t : J_t \to \mathcal{P}(\mathsf{Sub}(\mu x.\varphi(x)))$.

**The basis of induction:** $l'_r(r) := \{\mu x.\varphi(x)\} = f(\{\varphi(\mu x.\varphi(x))\})$.

**The step of induction:** Suppose $t, u \in T$ fills $u \in C(t)$ and $l'_t$ is already determined. Then, for each $j \in S_u$, set

$$l_u^{(1)}(j) := \begin{cases} \mathsf{TR}_{f(l(t)),f(l(u))}(l'_t(j)) & \text{if } j \in J_t, \\ l_u(j)(= \{\mu x.\varphi(x)\}) & \text{otherwise.} \end{cases}$$

Next, suppose $j_1$ and $j_2$ are siblings and $j_1$ is older. Then for every $\beta \in l_u^{(1)}(j_1) \cap l_u^{(1)}(j_2)$, remove $\beta$ from the labels of $j_2$ and its descendants. That is, execute **horizontal pruning**. In this way, the reduced label of $l_u^{(1)}$ is $l'_u$.

Step 5: The new label $L' : T \to \mathcal{P}(\mathsf{Sub}(\mu x.\varphi(x)))$ is defined as follows:

$$L'(t) := \bigcup_{j \in C_t(1)} l'_t(j)$$

From the above, we have completed the definition of $\mathcal{T}_{\mu x.\varphi(x)} := (T, C, r, L')$.

The extended tableau $\mathcal{T}_{\mu x.\varphi(x)}$ is what we want; that is, $\mathcal{T}_{\varphi(\mu x.\varphi(x))} \rightharpoonup \mathcal{T}_{\mu x.\varphi(x)}$ holds by $Z$. To show this, let's make sure that these satisfy the parity condition. Take any even infinite branch $\xi$ of $\mathcal{T}_{\mu x.\varphi(x)}$. Then, there exists an even trace $\mathsf{tr}'$ of $\mathcal{T}_{\mu x.\varphi(x)}$. If $\mathsf{tr}'$ stays at vertex $N_{\varphi(\mu x.\varphi(x))}$ consecutively (i.e., $\mathsf{tr}'[n] \in l'_n(N_{\varphi(\mu x.\varphi(x))})$ for every $n \geq 1$), then, from (♠), we can find even trace $\mathsf{tr}$ of $\mathcal{T}_{\varphi(\mu x.\varphi(x))}$ which stays at vertex $N_{\varphi(\mu x.\varphi(x))}$ consecutively. Similarly, If $\mathsf{tr}'$ is a trace that stays at vertex $j \neq N_{\varphi(\mu x.\varphi(x))}$ consecutively (i.e., $\{n \in \omega \mid \mathsf{tr}'[n] \in l'_n(j)\}$ is an infinite set), then, again from (♠), we can find even trace $\mathsf{tr}$ of $\mathcal{T}_{\varphi(\mu x.\varphi(x))}$ which stays at vertex $j$ consecutively. Therefore, $Z$ certainly satisfies the parity condition.

$\boxed{\text{Part 4}}$ First, we divide $\mathsf{Sub}(\varphi(\mathsf{anf}(\alpha)))$ into two disjoint sets:

$$\mathsf{Sub}_1 := \{\beta(\mathsf{anf}(\alpha)) \mid \beta(x) \in \mathsf{Sub}(\varphi(x))\} \setminus \{\mathsf{anf}(\alpha)\}$$
$$\mathsf{Sub}_2 := \mathsf{Sub}(\varphi(\mathsf{anf}(\alpha))) \setminus \mathsf{Sub}_1 (= \mathsf{Sub}(\mathsf{anf}(\alpha)))$$

A function $g : \mathsf{Sub}(\varphi(\mathsf{anf}(\alpha))) \to \mathcal{P}(\mathsf{Sub}(\varphi(\alpha)))$ is defined as follows:

$$g(\psi) := \begin{cases} \{\beta(\alpha)\} & \text{if } \psi = \beta(\mathsf{anf}(\alpha)) \in \mathsf{Sub}_1, \\ f(\psi) & \text{otherwise.} \end{cases}$$

Here, $f$ is the function mentioned in Remark 4.12. Note that

[12] Here, in the same way as Remark 4.2, instead of thinking that each vertex $j$ is labeled with a set of elements in the shape of $(\Gamma, \gamma)$, it is simply labeled with a set of formulas.

($\clubsuit$) $\mathsf{tr}$ is even ($\mu$-trace) $\Leftrightarrow$ $\vec{g}(\mathsf{tr})$ is even (i.e., include $\mu$-trace)

holds from the definition of $g$. Take an extended tableau $\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))} = (T, C, r, L)$ arbitrarily. Set $Z := \{(t,t) \mid t \in T\}$. If

$$\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))} \rightharpoonup (T, C, r, g \circ L) \tag{10}$$

with $Z$, then we are done. However unfortunately (10) is generally incorrect for the same reasons as mentioned in Part 3. We will construct an extended tableau $\mathcal{T}_{\varphi(\alpha)}$ where $\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))} \rightharpoonup \mathcal{T}_{\varphi(\alpha)}$ holds by $Z$ in the following 5 steps:

Step 1: We define the Büchi automaton $\mathcal{BA}_{\varphi(\mathsf{anf}(\alpha))} := \langle Q, \Sigma, q_0, \Delta, F\rangle$ as follows:

- $Q := \{(\Gamma, \gamma) \mid \Gamma \subseteq \mathsf{Sub}(\varphi(\mathsf{anf}(\alpha))),\ \gamma \in \Gamma\}$.
- $\Sigma := \mathcal{P}(\mathsf{Sub}(\varphi(\mathsf{anf}(\alpha)))$.
- $q_0 := \big(\{\varphi(\mathsf{anf}(\alpha))\}, \varphi(\mathsf{anf}(\alpha))\big)$.
- $\Delta(\Gamma', (\Gamma, \gamma)) := \big\{(\Gamma', \gamma') \mid \gamma' \in \mathsf{TR}_{\Gamma,\Gamma'}(\gamma)\big\}$.
- $F := \big\{q_0\big\} \bigcup \big\{(\Gamma, \mathsf{anf}(\alpha)) \mid (\Gamma, \mathsf{anf}(\alpha)) \in Q\big\}$.

Step 2: Convert nondeterministic Büchi automaton $\mathcal{BA}_{\varphi(\mathsf{anf}(\alpha))}$ to deterministic Rabin automaton $\mathcal{RA}_{\varphi(\mathsf{anf}(\alpha))}$ using Safra's construction. However, the following two points are changed from the conversion described in Subsection 3.2:

- In the **add new children**, change the child added in the first transition from 2 to $N_{\varphi(\mathsf{anf}(\alpha))}$, similar to the method described in Part 3.
- In the **initialize index appearence record**, abolish driving $j$ painted in red to the left end. Instead, change it so that it is driven to the left end excluding $N_{\varphi(\mathsf{anf}(\alpha))}$, similar to the method described in Part 3.

Step 3: Let the automaton defined above be $\mathcal{RA}_{\varphi(\mathsf{anf}(\alpha))} = \langle Q', \Sigma, q_0, \Delta', \{(A_j, R_j) \mid j \in J\}\rangle$. Then, note that the youngest child of the root 1 is $N_{\varphi(\mathsf{anf}(\alpha))}$, labeled with a subset of $\mathsf{Sub}_1$. The other children of the root 1 are labeled with a subset of $\mathsf{Sub}_2$.

Step 4: For each node $t$, we will define a labeled tree $\langle J_t, C_t, 1, l'_t, \pi_t, \mathsf{col}_t\rangle$ inductively from the root to the leaf; where $l'_t : S_t \to \mathcal{P}\big(\mathsf{Sub}(\varphi(\alpha))\big)$.

**The basis of induction:** $l'_r(r) := \{\varphi(\alpha)\} = g(\{\varphi(\mathsf{anf}(\alpha))\})$.

**The step of induction:** Suppose $t, u \in T$ fills $u \in C(t)$ and $l'_t$ is already determined. Then, for each $j \in S_u$, set

$$l^{(1)}_u(j) := \begin{cases} \mathsf{TR}_{g(l(t)),g(l(u))}(l'_t(j)) & \text{If } j \in J_t, \\ l_u(j)(=\{\mathsf{anf}(\alpha)\}) & \text{Otherwise.} \end{cases}$$

Next, suppose $j_1$ and $j_2$ are siblings and $j_1$ is older. Then for every $\beta \in l^{(1)}_u(j_1) \cap l^{(1)}_u(j_2)$, remove $\beta$ from the labels of $j_2$ and its descendants. That is, execute **horizontal pruning**. In this way, the reduced label of $l^{(1)}_u$ is $l'_u$.

Step 5: The new label $L' : T \to \mathcal{P}(\mathsf{Sub}(\varphi(\alpha))$ is defined as follows:

$$L'(t) := \bigcup_{j \in C_t(1)} l'_t(j)$$

From the above, we have completed the definition of $\mathcal{T}_{\varphi(\alpha)} := (T, C, r, L')$.

The extended tableau $\mathcal{T}_{\varphi(\alpha)}$ is what we want; that is, $\mathcal{T}_{\varphi(\mathsf{anf}(\alpha))} \rightharpoonup \mathcal{T}_{\varphi(\alpha)}$ holds by $Z$. Indeed, from ($\clubsuit$), we can show that $Z$ satisfies the parity condition, just as we did in Part 3. □

**Corollary 5.8.** *Let $\widehat{\alpha}(x)$ be an automaton normal form in which $x \in \mathsf{Free}(\widehat{\alpha}(x))$ occurs at once, positively, moreover, $x$ is in the scope of some modal operators. Set $\widehat{\varphi} := \mathsf{anf}(\mu x.\widehat{\alpha}(x))$. Then there exist tableaux $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})}$ and $\mathcal{T}_{\widehat{\varphi}}$ such that $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})} \rightharpoonup \mathcal{T}_{\widehat{\varphi}}$.*

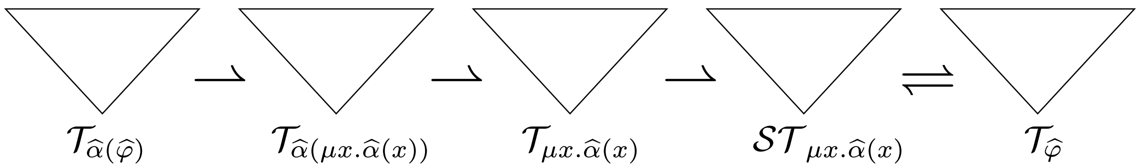


Figure 12: The plan for the proof of the corollary.

*Proof.* This corollary is proved using three tableaux; Figure 12 depicts the plan of the proof.

Let $\mathcal{ST}'_{\mu x.\widehat{\alpha}(x)}$ be a small tableau for $\varphi$ whose existence is guaranteed by Lemma 5.6. Let $\widehat{\varphi} = \mathsf{anf}(\mu x.\widehat{\alpha}(x))$ be an automaton normal form generated from $\mathcal{ST}'_{\mu x.\widehat{\alpha}(x)}$. Let $\mathcal{TB}_{\mu x.\widehat{\alpha}(x)}$ be a tableau with back edge generated from $\mathcal{ST}'_{\mu x.\widehat{\alpha}(x)}$ in the process of creating $\mathsf{anf}(\mu x.\widehat{\alpha}(x))$. Set $\mathcal{ST}_{\mu x.\widehat{\alpha}(x)} = \mathsf{UNW}_r(\mathcal{TB}_{\mu x.\widehat{\alpha}(x)})$. Note that $\mathcal{ST}_{\mu x.\widehat{\alpha}(x)}$ is also a small tableau.

First, we have $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})} \rightharpoonup \mathcal{T}_{\widehat{\alpha}(\mu x.\widehat{\alpha}(x))}$ for some extended tableau $\mathcal{T}_{\widehat{\alpha}(\mu x.\widehat{\alpha}(x))}$; from Part 4 of Lemma 5.7. Second, we have $\mathcal{T}_{\widehat{\alpha}(\mu x.\widehat{\alpha}(x))} \rightharpoonup \mathcal{T}_{\mu x.\widehat{\alpha}(x)}$ for some extended tableau $\mathcal{T}_{\mu x.\widehat{\alpha}(x)}$; from Part 3 of Lemma 5.7. Third, $\mathcal{T}_{\mu x.\widehat{\alpha}(x)} \rightharpoonup \mathcal{ST}_{\mu x.\widehat{\alpha}(x)}$ from Lemma 5.6. Fourth, since $\mathsf{anf}(\mu x.\widehat{\alpha}(x))$ is generated from $\mathcal{ST}_{\mu x.\widehat{\alpha}(x)}$, $\mathcal{ST}_{\mu x.\widehat{\alpha}(x)} \rightleftharpoons \mathcal{T}_{\widehat{\varphi}}$. Finally, by applying Part 1 and 2 of Lemma 5.7 repeatedly, we obtain $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})} \rightharpoonup \mathcal{T}_{\widehat{\varphi}}$. ☐

## 5.2 Proof of completeness

**Definition 5.9** (**Aconjunctive formula).** Let $\varphi$ be a well-named formula, and $\preceq_\varphi$ be its dependency order (recall Definition 2.4). Then, A variable $x \in \mathsf{Bound}(\varphi)$ is called *aconjunctive* if, for any $\alpha \wedge \beta \in \mathsf{Sub}(\varphi_x(x))$, $x$ is active in at most one of $\alpha$ or $\beta$. $\varphi$ is called *aconjunctive* if every $x \in \mathsf{Bound}(\varphi)$ such that $\eta_x = \mu$ is aconjunctive.

**Corollary 5.10.** *Let $\widehat{\varphi}$ be an automaton normal form. Then, we have*

1. *$\widehat{\varphi}$ is aconjunctive.*

2. *If $\widehat{\varphi}$ is not satisfiable, then $\widehat{\varphi} \vdash$.*

*Proof.* The first assertion of the Corollary is obvious from the observation of Remark 4.8. For the second assertion, suppose that $\widehat{\varphi}$ is not satisfiable. Note that, from the definition, a refutation for a aconjunctive formula is always thin. Then, from Lemma 2.21, there exists a thin refutation for $\widehat{\varphi}$. From Theorem 4.6, we obtain $\widehat{\varphi} \vdash$. ☐

In the next Lemma, we confirm that some compositions preserve aconjunctiveness.

**Lemma 5.11** (**Composition).** *Let $\varphi$, $\psi$ and $\alpha(x)$ be aconjunctive formulas where $x \in \mathsf{Prop}$ appears only positively in $\alpha(x)$. Then $\varphi \wedge \psi$, $\alpha(\varphi)$ and $\nu\vec{x}.\alpha(\vec{x})$ are also aconjunctive.*

*Proof.* We leave the proofs of these statement as an exercise to the reader. ☐

Next, in preparation for proving claim (h), we extend the definition of the trace given in Definition 2.18.

**Definition 5.12** (**An extension of trace).** Let $\mathcal{T}_\varphi = (T, C, r, L)$ be a tableau for some well-named formula $\varphi$. Let $\xi$ be a finite or infinite branch of $\mathcal{T}_\varphi$ and let $\mathsf{tr}$ be a trace on $\xi$. The set of all traces on $\xi$ is denoted by $\mathsf{TR}(\xi)$. $\mathsf{TR}(\xi[n,m])$ denotes the set $\{\mathsf{tr}[n,m] \mid \mathsf{tr} \in \mathsf{TR}(\xi)\}$ and may also be written $\mathsf{TR}(\xi[n], \xi[m])$. For any two factors $\mathsf{tr}[n,m]$ and $\mathsf{tr}'[n',m']$, we say $\mathsf{tr}[n,m]$ and $\mathsf{tr}'[n',m']$ are *equivalent* (denoted $\mathsf{tr}[n,m] \equiv \mathsf{tr}'[n',m']$) if, by ignoring invariant portions of the traces, they can be seen as the same sequence. For example, let;

$$\begin{array}{rlllll} \mathsf{tr}[n,n+3] = & \langle(\alpha\wedge\beta)\vee\gamma, & (\alpha\wedge\beta)\vee\gamma, & \alpha\wedge\beta, & \beta\rangle & \\ \mathsf{tr}'[n',n'+4] = & \langle(\alpha\wedge\beta)\vee\gamma, & \alpha\wedge\beta, & \alpha\wedge\beta, & \alpha\wedge\beta, & \beta\rangle \end{array}$$

then $\mathsf{tr}[n,n+3]$ and $\mathsf{tr}'[n',n'+4]$ are equivalent to each other. Let $X$ and $Y$ be the set of some factors of some traces. Then we write $X \Subset Y$ if for any $\mathsf{tr}[n,m] \in X$ there exists $\mathsf{tr}'[n',m'] \in Y$ such that $\mathsf{tr}[n,m] \equiv \mathsf{tr}'[n',m']$; and write $X \equiv Y$ if $X \Subset Y$ and $X \Supset Y$.

For technical reasons, we will need an *extended trace* (denotation: $\mathsf{tr}^+$) for each trace $\mathsf{tr}$ which is constructed by the following procedure (†) (see also Figure 13);

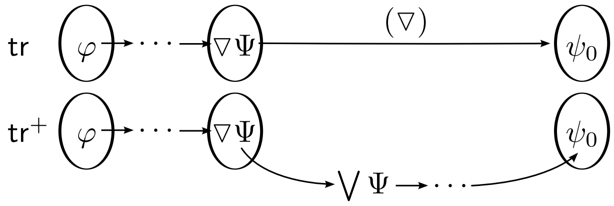


Figure 13: An extended trace.

(†): Suppose $\Psi = \{\psi_0, \psi_1, \ldots, \psi_k\}$ and that $\xi[n]$ is a ($\nabla$)-node in which $\mathsf{tr}[n] = \nabla\Psi$ is reduced into $\mathsf{tr}[n+1] = \psi_0$. Then, we insert the sequence

$$\langle \bigvee \Psi, \bigvee(\Psi \setminus \{\ \psi_1\}), \bigvee(\Psi \setminus \{\ \psi_1, \psi_2\}), \ldots, \bigvee\{\psi_0, \psi_{k-1}, \psi_k\}, \bigvee\{\psi_0, \psi_k\}\rangle$$

between $\mathsf{tr}[n]$ and $\mathsf{tr}[n+1]$.

Note that $\mathsf{tr}$ is even if and only if $\mathsf{tr}^+$ is even because inserted formulas are all $\vee$-formulas and, thus, the priorities of these formulas are equal to 0 (recall Equation (1)). The set of extended traces $\mathsf{TR}^+(\xi)$ and the set of factors of extended traces $\mathsf{TR}^+(\pi[n,m])$ or $\mathsf{TR}^+(\pi[n], \pi[m])$ are defined similarly.

The next lemma is the claim (h) mentioned in Section 1. The proof is long, but if you look closely, you can see that it is a natural proof.

**Lemma 5.13.** *Let $\alpha$ be an aconjunctive formula, and $\widehat{\varphi}$ be an automaton normal form. A tableau $\mathcal{T}_\alpha = (T_\alpha, C_\alpha, r_\alpha, L_\alpha)$ for $\alpha$ and a tableau $\mathcal{T}_{\widehat{\varphi}} = (T_{\widehat{\varphi}}, C_{\widehat{\varphi}}, r_{\widehat{\varphi}}, L_{\widehat{\varphi}})$ for $\widehat{\varphi}$ are given. If $\mathcal{T}_{\widehat{\varphi}}$ is a tableau consequence of $\mathcal{T}_\alpha$, then we can construct a thin refutation $\mathcal{R}$ for $\alpha \wedge \sim\widehat{\varphi}$ ($\equiv \sim(\alpha \to \widehat{\varphi})$).*

*Proof.* Let $\mathcal{T}_\alpha$ and $\mathcal{T}_{\widehat{\varphi}}$ be the tableaux satisfying the condition of the Lemma. Then, there exists a tableau consequence relation $Z$ from $\mathcal{T}_\alpha$ to $\mathcal{T}_{\widehat{\varphi}}$. Now, we will construct a thin refutation $\mathcal{R} = (T, C, r, L)$ for $\alpha \wedge \sim\widehat{\varphi}$ inductively. To facilitate the construction, we define two correspondence functions $\mathsf{Cor}_\alpha : T \to T_\alpha$ and $\mathsf{Cor}_{\widehat{\varphi}} : T \to T_{\widehat{\varphi}}$. These functions are partial and, in every considered node $t$ of $\mathcal{R}$, the following conditions are satisfied:

$$L(t) = L_\alpha(\mathsf{Cor}_\alpha(t)) \cup \left\{\sim \bigvee L_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t))\right\} \tag{11}$$

$$(\mathsf{Cor}_\alpha(t), \mathsf{Cor}_{\widehat{\varphi}}(t)) \in Z \tag{12}$$

Of course, the root of $\mathcal{R}$ is labeled by $\{\alpha \wedge \sim\widehat{\varphi}\}$ and its child, say $t_0$, is labeled by $\{\alpha, \sim\widehat{\varphi}\}$. For the base step, set $\mathsf{Cor}_\alpha(t_0) := r_\alpha$ and $\mathsf{Cor}_{\widehat{\varphi}}(t_0) := r_{\widehat{\varphi}}$. Then, the Condition (11) and (12) are indeed satisfied. The remaining construction is divided into two cases; the second of which will be further divided into four cases.

**Inductive step I** Suppose we have already constructed $\mathcal{R}$ up to a node $t$ where $\mathsf{Cor}_\alpha(t)$ and $\mathsf{Cor}_{\widehat{\varphi}}(t)$ are choice nodes of appropriate tableaux and satisfy Conditions (11) and (12). In this case, we prolong $\mathcal{R}$ up to $u$ so that:

1. $\mathsf{Cor}_\alpha(u)$ is a modal node of $\mathcal{T}_\alpha$ near $\mathsf{Cor}_\alpha(t)$.
2. $\mathsf{Cor}_{\widehat{\varphi}}(u)$ is a modal node of $\mathcal{T}_{\widehat{\varphi}}$ near $\mathsf{Cor}_{\widehat{\varphi}}(t)$.
3. Conditions (11) and (12) are satisfied in $u$.
4. $\mathsf{TR}[t,u] \equiv \mathsf{TR}[\mathsf{Cor}_\alpha(t), \mathsf{Cor}_\alpha(u)] \cup \{\langle\sim\bigvee L_{\widehat{\varphi}}(t_1), \cdots, \sim\bigvee L_{\widehat{\varphi}}(t_k)\rangle\}$ where $t_1 \cdots t_k \in T^+_{\widehat{\varphi}}$ is the $C_{\widehat{\varphi}}$-sequence starting at $\mathsf{Cor}_{\widehat{\varphi}}(t)$ and ending at $\mathsf{Cor}_{\widehat{\varphi}}(u)$.

The idea of the prolonging procedure is represented in Figure 14. From $t$, we first apply the tableau rules to the formulas of $\mathsf{Sub}(L_\alpha(\mathsf{Cor}_\alpha(t)))$ in the same order as they were applied from $\mathsf{Cor}_\alpha(t)$ and its nearest modal nodes. Then, we obtain a finite tree rooted in $t$ which is isomorphic to the section of $\mathcal{T}_\alpha$ between $\mathsf{Cor}_\alpha(t)$ and its nearest modal nodes. Therefore, for each leaf $t'$ of this section of $\mathcal{R}$, we can take unique modal node $t'_\alpha$ of $\mathcal{T}_\alpha$ that is isomorphic to $t'$. Note that $L(t') = L_\alpha(t'_\alpha) \cup \{\sim\bigvee L_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t))\}$. Now, the forth condition on the choice node of $Z$ is used.

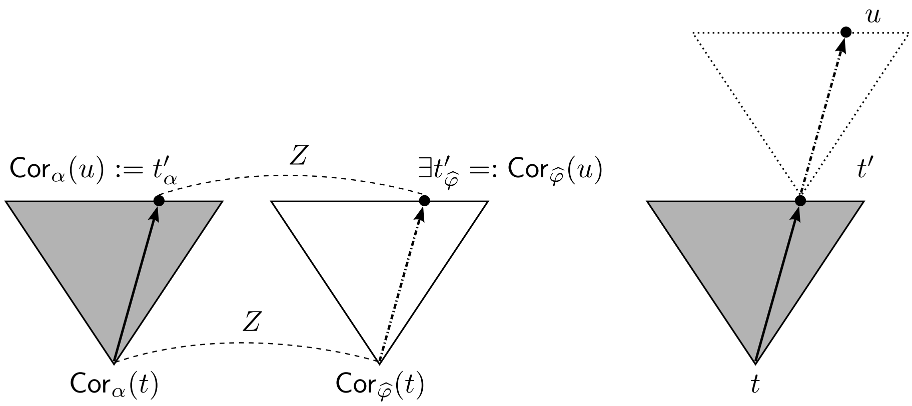


Figure 14: The prolonging procedure for Inductive step I.

From (12), we can find $t'_{\widehat{\varphi}} \in T_{\widehat{\varphi}}$ which is near $\mathsf{Cor}_{\widehat{\varphi}}(t)$ and satisfies $(t'_\alpha, t'_{\widehat{\varphi}}) \in Z$. Let us look at the path from $\mathsf{Cor}_{\widehat{\varphi}}(t)$ to $t'_{\widehat{\varphi}}$ in $\mathcal{T}_{\widehat{\varphi}}$. Since $\widehat{\varphi}$ is an automaton normal form on this path only the $(\vee)$-, $(\eta)$- and ($\mathsf{Regeneration}$)-rules, and $(\wedge)$-rules reducing $\widehat{\psi} \wedge \top$ to $\{\widehat{\psi}, \top\}$ may be applied first. Then, we have zero or more applications of the $(\wedge)$-rule. Let us apply dual rules to $\sim \bigvee L_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t))$ (note that ($\mathsf{Regeneration}$) and $(\eta)$ are self-dual).

For an application of the $(\vee)$-rule in $\mathcal{T}_{\widehat{\varphi}}$, we apply the $(\wedge)$-rule followed by the ($\mathsf{Weak}$)-rule to leave only the conjunct which appears on the path to $t'_{\widehat{\varphi}}$. In this way, we ensure the resulting path of $\mathcal{R}$ will be thin.

For an application of the $(\wedge)$-rule reducing $\widehat{\psi} \wedge \top$ to $\{\widehat{\psi}, \top\}$ in $\mathcal{T}_{\widehat{\varphi}}$, we apply the $(\vee)$-rule in $\mathcal{R}$. Then, we have two children, say $v_1$ and $v_2$ such that $L(v_1)$ includes $\sim \widehat{\psi}$ and $L(v_2)$ includes $\sim \top = \bot$. Since $L(v_2)$ is inconsistent, if we further prolong $\mathcal{R}$ from $v_2$ to its nearest modal nodes, such modal nodes also labeled inconsistent set. This means that the modal nodes can be leaves of a refutation. We therefore stop the prolonging procedure on such modal nodes.

After these reductions, we get a node $u$ which is labeled by $L_\alpha(t'_\alpha) \cup \{\sim \bigvee L_{\widehat{\varphi}}(t'_{\widehat{\varphi}})\}$. Setting $\mathsf{Cor}_\alpha(u) := t'_\alpha$ and $\mathsf{Cor}_{\widehat{\varphi}}(u) := t'_{\widehat{\varphi}}$ establishes Conditions (11) and (12). Conditions 1 through 4 follow directly from the construction.

**Inductive step II** Suppose we have already constructed $\mathcal{R}$ up to a node $t$ where $\mathsf{Cor}_\alpha(t)$ and $\mathsf{Cor}_{\widehat{\varphi}}(t)$ are modal nodes of appropriate tableaux and satisfy Conditions (11) and (12). Note that, since $\widehat{\varphi}$ is an automaton normal form, we can put $L_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t)) = \{\nabla\Psi, l_1, \ldots, l_i\}$ or $L_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t)) = \{l_1, \ldots, l_i\}$ where $l_1, \ldots, l_i \in \mathsf{Lit}(\widehat{\varphi})$. Moreover, observe that

$$\begin{aligned}
\sim\left(\nabla\Psi \wedge \bigwedge_{1\le k\le i} l_k\right) &\equiv \sim\nabla\Psi \vee \left(\bigvee_{1\le k\le i} \sim l_k\right) \\
&\equiv \sim\left(\left(\bigwedge \Diamond\Psi\right) \wedge \Box\left(\bigvee \Psi\right)\right) \vee \left(\bigvee_{1\le k\le i} \sim l_k\right) \\
&\equiv \left(\bigvee_{\psi\in\Psi} \Box \sim\psi\right) \vee \Diamond\left(\bigwedge \sim\Psi\right) \vee \left(\bigvee_{1\le k\le i} \sim l_k\right) \\
&\equiv \left(\bigvee_{\psi\in\Psi} (\nabla\{\sim\psi\} \vee \nabla\emptyset)\right) \vee \nabla\left\{\left(\bigwedge \sim\Psi\right), \top\right\} \vee \left(\bigvee_{1\le k\le i} \sim l_k\right).
\end{aligned}$$

Therefore, if we prolong $\mathcal{R}$ from $t$ up to its nearest modal nodes $u$ by applying the $(\vee)$-rule repeatedly, the label of $u$ can be categorized as one of following four cases:

**(Case 1):** $L(u) = L_\alpha(\mathsf{Cor}_\alpha(t)) \cup \{\sim l_k\}$ for some $k$ such that $1 \le k \le i$.

**(Case 2):** $L(u) = L_\alpha(\mathsf{Cor}_\alpha(t)) \cup \{\nabla\emptyset\}$.

**(Case 3):** $L(u) = L_\alpha(\mathsf{Cor}_\alpha(t)) \cup \{\nabla\{\sim\psi\}\}$ for some $\psi \in \Psi$.

**(Case 4):** $L(u) = L_\alpha(\mathsf{Cor}_\alpha(t)) \cup \{\nabla\{(\bigwedge\sim\Psi), \top\}\}$.

In every cases, it is possible that $L_\alpha(\mathsf{Cor}_\alpha(t))$ is inconsistent and, thus, $L(u)$ is also inconsistent. If this is so, all $u$ can be a leaf of a refutation. Therefore, we stop the prolonging procedure on $u$ in this case. Now, we consider the case where $L_\alpha(\mathsf{Cor}_\alpha(t))$ is consistent.

In Case 1, the prop condition is used; by Condition (12), we have $l_k \in L_\alpha(\mathsf{Cor}_\alpha(t))$. Thus, $L(u)$ includes $l_k$ and $\sim l_k$. This means that $L(u)$ is inconsistent and so $u$ can be a leaf of a refutation. We therefore stop the prolonging procedure on $u$ in this case.

In Case 2, the back condition on modal nodes is used. Since $C_{\widehat{\varphi}}(\mathsf{Cor}_{\widehat{\varphi}}(t)) \neq \emptyset$, it must hold that $C_\alpha(\mathsf{Cor}_\alpha(t)) \neq \emptyset$. Take $v_\alpha \in C_\alpha(\mathsf{Cor}_\alpha(t))$ arbitrarily. We prolong $\mathcal{R}$ from $u$ to $v \in C(u)$ in such a way that $L(v) = L_\alpha(v_\alpha) \cup \{\bigvee\emptyset(\equiv \bot)\}$. Since $L(v)$ is inconsistent, if we further prolong $\mathcal{R}$ from $v$ to its nearest modal nodes, such modal nodes are also inconsistent. This means that the modal nodes can be a leaves of a refutation. We therefore stop the prolonging procedure on such modal nodes in this case.

In Case 3, the back condition on modal nodes is used. Let $v_{\widehat{\varphi}}$ be a child of $\mathsf{Cor}_{\widehat{\varphi}}(t)$ such that $L_{\widehat{\varphi}}(v_{\widehat{\varphi}}) = \{\psi\}$. Then, by Condition (12), we can find $v_\alpha \in C_\alpha(\mathsf{Cor}_\alpha(t))$ such that $(v_\alpha, v_{\widehat{\varphi}}) \in Z$. We create a new child $v$ of $u$ which is labeled by $L_\alpha(\mathsf{Cor}_\alpha(v_\alpha)) \cup \{\sim\psi\}$. Moreover, we set $\mathsf{Cor}_\alpha(v) := v_\alpha$ and $\mathsf{Cor}_{\widehat{\varphi}}(v) := v_{\widehat{\varphi}}$. This prolonging procedure preserves Conditions (11) and (12). Note that, in this case, $\mathsf{Cor}_\alpha(v)$ and $\mathsf{Cor}_{\widehat{\varphi}}(v)$ are choice nodes of appropriate tableaux.

In Case 4, the forth condition on modal nodes is used. The idea of the prolonging procedure is represented in Figure 15. Let $L_\alpha(\mathsf{Cor}_\alpha(t)) = \{\nabla\Delta_1, \dots, \nabla\Delta_i, l_1, \dots, l_j\}$. In this case, we first create

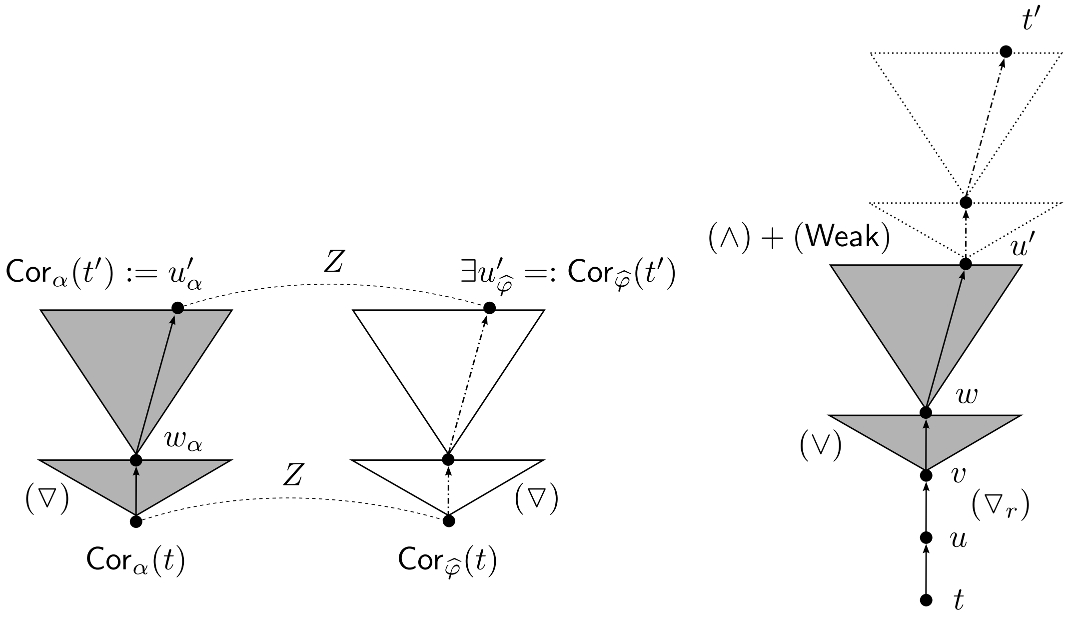


Figure 15: The prolonging procedure for Case 4.

a new child $v$ of $u$ such that

$$L(v) = \left\{\bigvee\Delta_1, \dots, \bigvee\Delta_i\right\} \cup \left\{\bigwedge\sim\Psi\right\}.$$

From the choice node $v$, we further prolong $\mathcal{R}$ up to its nearest modal nodes $t'$ so that

5. $\mathsf{Cor}_\alpha(t')$ is a next modal node of $\mathsf{Cor}_\alpha(t)$.
6. $\mathsf{Cor}_{\widehat{\varphi}}(t')$ is a next modal node of $\mathsf{Cor}_{\widehat{\varphi}}(t)$.
7. Condition (11) and (12) are satisfied in $t'$.

8. $\mathsf{TR}[u,t'] \equiv \mathsf{TR}^{+}[\mathsf{Cor}_\alpha(t), \mathsf{Cor}_\alpha(t')] \cup \{\langle \nabla \{(\bigwedge \sim \Psi), \top\}, \bigwedge \sim \Psi, \dots, \sim \psi =\sim \bigvee L_{\widehat{\varphi}}(t_1), \cdots, \sim \bigvee L_{\widehat{\varphi}}(t_k)\rangle\}$ where $t_1 \cdots t_k \in T^{+}_{\widehat{\varphi}}$ is the $C_{\widehat{\varphi}}$-sequence starting at the child of $\mathsf{Cor}_{\widehat{\varphi}}(t)$ labeled by $\{\psi\}$ and ending at $\mathsf{Cor}_{\widehat{\varphi}}(t')$.

Next, we apply $(\vee)$-rules to $\bigvee \Delta_1$ repeatedly until we arrive at the node $w$ such that

$$L(w) = \{\delta_1\} \cup \left\{\bigvee \Delta_2, \dots, \bigvee \Delta_i\right\} \cup \left\{\bigwedge \sim \Psi\right\}$$

where $\delta_1 \in \Delta_1$. Note that there exists $w_\alpha \in C_\alpha(\mathsf{Cor}_\alpha(t))$ such that

$$L_\alpha(w_\alpha) = \{\delta_1\} \cup \left\{\bigvee \Delta_2, \dots, \bigvee \Delta_i\right\}$$

From $w$, we apply the tableau rules to formulas of $\mathsf{Sub}(L_\alpha(w_\alpha))$ in the same order as they were applied from $w_\alpha$ and its nearest modal nodes. Then, we obtain a finite tree rooted in $w$ which is isomorphic to the section of $\mathcal{T}_\alpha$ between $w_\alpha$ and nearest modal nodes. Therefore, for each leaf $u'$ of this section of $\mathcal{R}$, we can take a unique modal node $u'_\alpha$ of $\mathcal{T}_\alpha$ which is isomorphic to $u'$. Note that $L(u') = L_\alpha(u'_\alpha) \cup \{\bigwedge \sim \Psi\}$. Since $u'_\alpha$ is a next modal node of $\mathsf{Cor}_\alpha(t)$, from Condition (12) and the forth condition on modal nodes, we can assume that there exists $u'_{\widehat{\varphi}}$ which is a next modal node of $\mathsf{Cor}_{\widehat{\varphi}}(t)$ and satisfies $(u'_\alpha, u'_{\widehat{\varphi}}) \in Z$. We will now look at the path from $\mathsf{Cor}_{\widehat{\varphi}}(t)$ to $t'_{\widehat{\varphi}}$ in $\mathcal{T}_{\widehat{\varphi}}$ and exploit $(\wedge)$-rules and $(\mathsf{Weak})$-rules so that the trace $\mathsf{tr}$ on this path satisfies Condition 8. Finally, we get a node $t'$ which is labeled by $L_\alpha(u'_\alpha) \cup \{\sim \bigvee L_{\widehat{\varphi}}(u'_{\widehat{\varphi}})\}$. Setting $\mathsf{Cor}_\alpha(t') := u'_\alpha$ and $\mathsf{Cor}_{\widehat{\varphi}}(t') := u'_{\widehat{\varphi}}$ establishes Conditions (11) and (12). Then, Conditions 5 through 8 follow directly from the construction.

The above two procedures completely describe $\mathcal{R}$. All the leaves are labeled by an inconsistent set. Moreover, take an infinite branch $\xi$ of $\mathcal{R}$ arbitrarily. Let $\xi_\alpha$ be the branch of $\mathcal{T}_\alpha$ such that $\{n \in \omega \mid \mathsf{Cor}_\alpha(\xi) = \xi_\alpha[n]\}$ is an infinite set. Let $\xi_{\widehat{\varphi}}$ be the branch of $\mathcal{T}_{\widehat{\varphi}}$ such that $\{n \in \omega \mid \mathsf{Cor}_{\widehat{\varphi}}(\xi) = \xi_{\widehat{\varphi}}[n]\}$ is an infinite set. For any trace $\mathsf{tr} \in \mathsf{TR}(\xi)$, we have $\mathsf{tr}[1] = \alpha \wedge \sim \widehat{\varphi}$ and, $\mathsf{tr}[2] = \alpha$ or $\mathsf{tr}[2] =\sim \widehat{\varphi}$. $\mathsf{TR}_1(\xi)$ denotes the set of all the trace $\mathsf{tr} \in \mathsf{TR}(\xi)$ such that $\mathsf{tr}[2] = \alpha$. $\mathsf{tr}_2 \in \mathsf{TR}(\xi)$ denotes the trace such that $\mathsf{tr}_2[2] =\sim \widehat{\varphi}$. Then, from the construction of $\mathcal{R}$, we have;

**(T1)** $\mathsf{TR}(\xi) = \mathsf{TR}_1(\xi) \cup \{\mathsf{tr}_2\}$.

**(T2)** $\mathsf{TR}_1^{+}(\xi) \equiv \mathsf{TR}^{+}(\xi_\alpha)$.

**(T3)** $\mathsf{tr}_2$ is even if and only if $\xi_{\widehat{\varphi}}$ is odd.

**(T4)** $\xi_\alpha$ and $\xi_{\widehat{\varphi}}$ are associated with each other.

Above conditions imply that $\xi$ is odd. Indeed, if $\xi_\alpha$ is odd, then, from **(T2)**, $\xi$ is also odd. If $\xi_\alpha$ is even, then, from **(T4)**, $\xi_{\widehat{\varphi}}$ is also even. Therefore, from **(T3)**, $\mathsf{tr}_2$ is odd. From **(T1)**, we can assume that $\xi$ is odd. $\mathcal{R}$ is also thin because $\alpha$ is aconjunctive and whenever we reduce a $\wedge$-formula originated from $\sim \widehat{\varphi}$, we leave only one conjunction and discard the other by applying $(\mathsf{Weak})$-rule. Therefore, $\mathcal{R}$ is a thin refutation as required. ☐

**Lemma 5.14 (Main lemma).** *For any well-named formula $\varphi$, there exists a semantically equivalent automaton normal form $\widehat{\varphi}$ such that $\varphi \to \widehat{\varphi}$ is provable in* $\mathsf{Koz}$. *Moreover, for any $x \in \mathsf{Free}(\varphi)$ which occurs only positively in $\varphi$, it hold that $x \in \mathsf{Free}(\widehat{\varphi})$ and $x$ occurs only positively in $\widehat{\varphi}$.*

*Proof.* We prove the lemma by the induction on the structure of $\varphi$.

**Case:** $\varphi \in \mathsf{Lit}$**.** In this case, $\widehat{\varphi}$ is just $\varphi$.

**Case:** $\varphi = \alpha \vee \beta$**.** By the induction assumption, there exist automaton normal forms $\widehat{\alpha}$ and $\widehat{\beta}$ which are equivalent to $\alpha$ and $\beta$, respectively, such that $\vdash \alpha \to \widehat{\alpha}$ and $\vdash \beta \to \widehat{\beta}$. Set $\widehat{\varphi} := \widehat{\alpha} \vee \widehat{\beta}$. Then, we have $\vdash \alpha \vee \beta \to \widehat{\varphi}$.

**Case:** $\varphi = \nabla \Psi$**.** This case is very similar to the previous one.

**Case:** $\varphi = \alpha \wedge \beta$**.** By the induction assumption, there exist automaton normal forms $\widehat{\alpha}$ and $\widehat{\beta}$ which are equivalent to $\alpha$ and $\beta$ respectively, such that $\vdash \alpha \to \widehat{\alpha}$ and $\vdash \beta \to \widehat{\beta}$; thus, we have $\vdash \alpha \wedge \beta \to \widehat{\alpha} \wedge \widehat{\beta}$. Set $\widehat{\varphi} := \mathsf{anf}(\widehat{\alpha} \wedge \widehat{\beta})$. Then, from Theorem 4.11, we have $\mathcal{T}_{\widehat{\alpha} \wedge \widehat{\beta}} \rightleftharpoons \mathcal{T}_{\widehat{\varphi}}$ for some $\mathcal{T}_{\widehat{\alpha} \wedge \widehat{\beta}}$ and, thus, $\mathcal{T}_{\widehat{\alpha} \wedge \widehat{\beta}} \rightharpoonup \mathcal{T}_{\widehat{\varphi}}$. On the other hand, by Lemma 5.11, we can assume that $\widehat{\alpha} \wedge \widehat{\beta}$ is aconjunctive. From Lemma 5.13 and Theorem 4.6, we have $\vdash \widehat{\alpha} \wedge \widehat{\beta} \to \widehat{\varphi}$. Therefore, we have $\vdash \alpha \wedge \beta \to \widehat{\varphi}$.

**Case:** $\varphi = \nu x.\alpha(x)$**.** By the induction assumption, we have an equivalent automaton normal form $\widehat{\alpha}(x)$ of $\alpha(x)$ such that $\vdash \alpha(x) \to \widehat{\alpha}(x)$. Therefore, $\vdash \nu x.\alpha(x) \to \nu x.\widehat{\alpha}(x)$. Set $\widehat{\varphi} := \mathsf{anf}(\nu x.\widehat{\alpha}(x))$. Then, from Theorem 4.11, we have $\mathcal{T}_{\nu x.\widehat{\alpha}(x)} \rightleftharpoons \mathcal{T}_{\widehat{\varphi}}$ for some $\mathcal{T}_{\nu x.\widehat{\alpha}(x)}$ and, thus, $\mathcal{T}_{\nu x.\widehat{\alpha}(x)} \rightharpoonup \mathcal{T}_{\widehat{\varphi}}$. On the other hand, by Lemma 5.11, we can assume that $\nu x.\widehat{\alpha}(x)$ is aconjunctive. From Lemma 5.13 and Theorem 4.6, we have $\vdash \nu x.\widehat{\alpha}(x) \to \widehat{\varphi}$. Therefore, $\vdash \nu x.\alpha(x) \to \widehat{\varphi}$.

**Case:** $\varphi = \mu x.\alpha(x)$**.** By the induction assumption, we have an equivalent automaton normal form $\widehat{\alpha}(x)$ of $\alpha(x)$ such that $\vdash \alpha(x) \to \widehat{\alpha}(x)$. Therefore, $\vdash \mu x.\alpha(x) \to \mu x.\widehat{\alpha}(x)$. Set $\widehat{\varphi} := \mathsf{anf}(\mu x.\widehat{\alpha}(x))$. Then, from Corollary 5.8, we have $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})} \rightharpoonup \mathcal{T}_{\widehat{\varphi}}$ for some $\mathcal{T}_{\widehat{\alpha}(\widehat{\varphi})}$. On the other hand, by Lemma 5.11, we can assume that $\widehat{\alpha}(\widehat{\varphi})$ is aconjunctive. From Lemma 5.13 and Theorem 4.6, $\vdash \widehat{\alpha}(\widehat{\varphi}) \to \widehat{\varphi}$. By applying the $(\mathsf{Ind})$-rule, we obtain $\vdash \mu x.\widehat{\alpha}(x) \to \widehat{\varphi}$. Thus, $\vdash \mu x.\alpha(x) \to \widehat{\varphi}$.

Hence, we have proved the Lemma for all cases. □

**Theorem 5.15 (Completeness).** *For any formula $\varphi$, if $\varphi$ is not satisfiable, then $\sim\varphi$ is provable in* Koz*.*

*Proof.* Let $\varphi$ be an unsatisfiable formula. By Part 5 of Lemma 2.9, we can construct a well-named formula $\mathsf{wnf}(\varphi)$ such that

$$\vdash \varphi \leftrightarrow \mathsf{wnf}(\varphi) \tag{13}$$

On the other hand, from Lemma 5.14, there exists an automaton normal form $(\mathsf{wnf}(\varphi))\hat{}$ which is semantically equivalent to $\mathsf{wnf}(\varphi)$ and thus to $\varphi$ such that

$$\vdash \mathsf{wnf}(\varphi) \to (\mathsf{wnf}(\varphi))\hat{} \tag{14}$$

Since $(\mathsf{wnf}(\varphi))\hat{}$ is not satisfiable, by Corollary 5.10 we have

$$\vdash (\mathsf{wnf}(\varphi))\hat{} \to \bot \tag{15}$$

Finally by combining Equations (13) through (15) we obtain $\vdash \varphi \to \bot$ as required. □


## Acknowledgements

The author gives great thanks to the anonymous referees for their valuable comments; in particular, they pointed out some mathematical errors (especially in Lemma 5.7) in a preliminary version of this paper.


# References


[1] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics, 5(2):285–309, 1955.

[2] Balder ten Cate and Gaëlle Fontaine. An easy completeness proof for the modal mu-calculus on finite trees. In FOSSACS, pages 161–175, 2010.

[3] Damian Niwinski and Igor Walukiewicz. Games for the $\mu$-calculus. Theor. Comput. Sci., 163(1–2):99-116, 1996.

[4] David Janin and Igor Walukiewicz. Automata for the modal mu-calculus and related results. In Jirí Wiedermann and Petr Hájek, editors, MFCS, volume 969 of Lecture Notes in Computer Science, pages 552–562. Springer, 1995.

[5] Dexter Kozen. Results on the propositional mu-calculus. Theor. Comput. Sci., 27:333–354, 1983.

[6] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001], volume 2500 of Lecture Notes in Computer Science. Springer, 2002.

[7] G. Lenzi. The modal $\mu$-calculus: a survey. TASK Quarterly, 9(3):293–316, 2005.

[8] Igor Walukiewicz. Completeness of kozen's axiomatisation of the propositional $\mu$-calculus. Information and Computation, 157(1–2):142 – 182, 2000.

[9] Julian Bradfield and Colin Stirling. Modal mu-calculi. In HANDBOOK OF MODAL LOGIC, pages 721–756. Elsevier, 2007.

[10] J. Křetìnsỳ, T. Meggendorfer, C. Waldmann, and M. Weininger. Index appearance record for transforming rabin automata into parity automata. In TACAS 2017, pages 443–460, 2017.

[11] J.W. de Bakker and D.S. Scott. A theory of programs. Unpublished Manuscript, IBM, Vienna, 1969.

[12] Luca Alberucci. Sequent calculi for the modal $\mu$-calculus over $\mathsf{S5}$. J. Log. Comput., 19(6):971–985, 2009.

[13] Nick Bezhanishvili and Ian Hodkinson. Sahlqvist theorem for modal fixed point logic. Theor. Comput. Sci., 424(0):1 – 19, 2012.

[14] S. Safra. On the complexity of $\omega$-automata. 29th IEEE FOCS, 1988.