# Towards Completeness via Proof Search in the Linear Time $\mu$-calculus

## The case of Büchi inclusions

Amina Doumane
PPS, IRIF, Université Paris Diderot
doumane@pps.univ-paris-diderot.fr

David Baelde, Lucca Hirschi
LSV, ENS Cachan & CNRS,
Université Paris-Saclay
{baelde,hirschi}@lsv.ens-cachan.fr

Alexis Saurin
PPS, IRIF, CNRS &
Université Paris Diderot
saurin@pps.univ-paris-diderot.fr

## Abstract

Modal $\mu$-calculus is one of the central languages of logic and verification, whose study involves notoriously complex objects: automata over infinite structures on the model-theoretical side; infinite proofs and proofs by (co)induction on the proof-theoretical side. Nevertheless, axiomatizations have been given for both linear and branching time $\mu$-calculi, with quite involved completeness arguments. We come back to this central problem, considering it from a proof search viewpoint, and provide some new completeness arguments in the linear time $\mu$-calculus. Our results only deal with restricted classes of formulas that closely correspond to (non-alternating) $\omega$-automata but, compared to earlier proofs, our completeness arguments are direct and constructive. We first consider a natural circular proof system based on sequent calculus, and show that it is complete for inclusions of parity automata directly expressed as formulas, making use of Safra's construction directly in proof search. We then consider the corresponding finitary proof system, featuring (co)induction rules, and provide a partial translation result from circular to finitary proofs. This yields completeness of the finitary proof system for inclusions of sufficiently deterministic parity automata, and finally for arbitrary Büchi automata.

## 1. Introduction

Modal $\mu$-calculus, *i.e.*, modal logic with least and greatest fixed point operators, is a central logic in verification, from both theoretical and practical perspectives. Developing a better understanding of its models and deductions has since long been the focus of numerous works considering their importance to understand and put this logic in practice.

The linear time $\mu$-calculus has infinite words as models, and may be used to express trace properties of reactive systems seen as sets of models. The branching time $\mu$-calculus has infinite trees as models, and allows higher expressiveness since its formulas directly express properties of a system's execution tree. Both logics enjoy close relationships with automata theory. Any formula $F$ can be compiled into an automaton $\mathcal{A}_F$ which accepts exactly the models of the formula. If the reactive system $S$ to be verified can itself be described by means of an automaton $\mathcal{A}_S$, the logical model-checking problem can be expressed purely in automata-theoretic terms: is the automaton of the system included in that of the formula, *i.e.*, $\mathcal{L}(\mathcal{A}_S) \subseteq \mathcal{L}(\mathcal{A}_F)$? In fact, this approach to model-checking also gives a way to decide logical entailment in modal $\mu$-calculi.

Even when decidability of the logic is known, designing a complete proof system is still of interest. For instance, it provides compositionality and permits to use an interactive prover and not only a model-checker for verifying properties of the system. Another perspective is given through the problematic of *proof certificates*. The goal here is to provide independently checkable artifacts justifying the answer of verification tools, in the form of proofs of a standard, well-established proof-theoretical langage. This puts an emphasis and interest in the question of designing complete proof systems, with a specific focus on *constructivity*.

Since Kozen's seminal paper (Kozen 1983), several works focused on designing finitary and complete proof systems for the modal $\mu$-calculus (Walukiewicz 1993, 1995; Kaivola 1995; Walukiewicz 2000). While completeness for the $\mu$-calculus is a notoriously difficult problem, we argue that when considering completeness and its potential applications, the way the result is established is almost as important as the result itself. Indeed, completeness only provides a theoretical way to obtain proofs from validity. The proof of completeness may indeed involve complex, non-constructive arguments yielding no reasonable method for actually constructing a proof. On the contrary, a constructive proof of completeness (for instance, specifying a proof search method) readily provides a realistic algorithm. In the following, we will refer to those completeness results proved constructively as *constructive completeness* results.

When considering the history of completeness proofs for the $\mu$-calculus, there seems to be a tension between two requirements: having a completeness result for the full logic and obtaining constructive completeness. In his seminal paper, Kozen could not establish completeness for the full logic: he targeted a fragment, *aconjunctive formulas*, providing a constructive completeness proof for this fragment. Later developments either kept the constructivity property but fell out of Kozen's axiomatization (Walukiewicz' first result of completeness (Walukiewicz 1993), is constructive but for a deductive system which is stronger than Kozen's axiomatization) or stepped back on constructive completeness: Walukiewicz' result for branching time (Walukiewicz 1995, 2000) and Kaivola's result for linear time $\mu$-calculus (Kaivola 1995) are not constructive.

Constructive completeness results for the $\mu$-calculus have been given for various *infinitary* systems. Various tableaux systems have

been devised for satisfiability (Kozen 1983; Streett and Emerson 1989; Janin and Walukiewicz 1995; Kaivola 1995) and model-checking (Stirling and Walker 1991; Bradfield et al. 1996). All these systems are essentially infinite derivation trees, with validity criteria over infinite branches. These criteria are the main difference between all these systems, and resemble more or less closely acceptance conditions of automata over infinite structures. More recently, and considering validity rather than satisfiability, (Dax et al. 2006) have proposed a system of infinite sequent calculus proofs for linear time $\mu$-calculus, and used it as a basis for validity checking algorithms. The above works provide sound and complete derivation systems, sometimes highly amenable to efficient satisfiability checking. However, from a proof-theoretical viewpoint, these inference systems are not as simple as one might hope[1] and are often not structured enough to stand as proof objects.

***Our problem.*** In the present paper, we aim at developing constructive completeness results for fragments of the linear $\mu$-calculus. More precisely, we consider fragments corresponding to inclusions of $\omega$-automata, motivated by the question of producing proof-theoretical certificates for this important model-checking problem. To this end, we consider both finitary and infinitary proof systems for which we establish several constructive completeness results.

Proof certificates as advocated by Miller *et al* (Miller 2011; Chihani et al. 2013) aim at formulating verification certificates in a common language of proof theory bringing new benefits, such as the ability to *run* certificates (*e.g.*, to extract examples or algorithms from them) which is naturally supported by the computational interpretation of cut elimination. Targeting proof certificates, we impose special requirements on our proof objects. For instance, we need our proof system to be structured enough to support properties such as cut elimination or focalizationn and their computational interpretation. For instance, we will require sequents to be sets of *occurrences* of formulas which is necessary to provide proofs with a Curry-Howard interpretation. Indeed, the following proofs

$$\dfrac{\dfrac{}{A^x \vdash A}\ (\mathsf{Ax})}{A^x, A^y \vdash A}\ (\mathsf{W_l}) \qquad \dfrac{\dfrac{}{A^y \vdash A}\ (\mathsf{Ax})}{A^x, A^y \vdash A}\ (\mathsf{W_l})$$

have different computational interpretations, exactly as $\lambda x.\lambda y.x$ and $\lambda x.\lambda y.y$ do. To make this explicit, we have distinguished the two occurrences of the left formulas by labelling them with variables $x$ and $y$. When sequents are modelled as sets or multisets of formulas, as usual in the studies on $\mu$-calculus, those two proofs are equated. We will work with sequents containing occurrences of formulas in the rest of the paper but we will not be so pedantic: we will keep the labelling of occurrences implicit.

Until now, infinitary derivation systems have mostly been used as an intermediate step in a series of transformations to obtain completeness for a finitary proof systems. We wish to go further, and focus on these infinitary systems as proof-theoretical systems in their own right. This desire to study infinitary proof systems more thoroughly is notably motivated by the simplicity of their formulation of (co)induction rules. Among other reasons, this explains a sustained interest in infinitary proof systems: Santocanale recently made new progress on cut-elimination in his framework (Santocanale 2002; Fortier and Santocanale 2013); circular proofs are being used in richer logics than $\mu$-calculus, for instance, Brotherston designed a complete infinitary proof system for Peano arithmetic (Brotherston and Simpson 2011), or to infer program invariants in separation logic (Brotherston and Gorogiannis 2014); etc.

[1] Although one only ever needs to consider regular derivation trees, checking the validity condition on these derivations remains highly non-trivial — for instance, validity is reduced to an inclusion of Büchi automata in (Dax et al. 2006).

***Contributions and organization of the paper.*** We consider linear time $\mu$-calculus and two proof systems for it: the infinitary system $\mu\mathsf{LK}^\infty$, its regular fragment $\mu\mathsf{LK}^\omega$ and the finitary sequent calculus $\mu\mathsf{LK}$ featuring (co)induction rules in the style of Kozen; they are presented in Section 2. We provide completeness arguments for classes of formulas corresponding to automata inclusions; those classes are introduced in Section 3, motivated by the encodings of $\omega$-automata into $\mu$-calculus. As explained above, these specific statements are central in model-checking, and building proof certificates for them is thus particularly interesting. Our constructive completeness argument is based on a new translation result that does not impose a syntactic condition on formulas, but a weaker geometric condition on proofs. This result makes crucial use of the precise structure of $\mu\mathsf{LK}^\omega$ proofs, which are more constraining than the ones usually found in the litterature, notably (Dax et al. 2006). As a result, the completeness of our circular proofs is already not obvious. First, we show in Section 4 that our circular proof system is complete for inclusions of parity automata. To obtain this result, we make use of Safra's determinization construction to build circular proofs. In other words, we view Safra's construction as a proof search method; we claim that the scope of this idea goes beyond the particular setting considered here. Second, we prove our translation result in Subsection 5.1 for finitizing circular proofs. Third, we show in Subsection 5.2 that the proofs constructed in the first step satisfy the translatability criterion when they are deterministic enough. This yields completeness of the finitary proof system for inclusions of sufficiently deterministic parity automata. We finally obtain Theorem 36, establishing completeness for inclusions of non-deterministic Büchi automata, by gradually diminishing their "level of non-determinism" using some specific proof construction in $\mu\mathsf{LK}$. In Section 6, we conclude and discuss our contributions: although we only provide partial completeness results, we argue that we tackle the main difficulty, that is non-determinism, while remaining constructive.

## 2. Proof systems for the linear-time $\mu$-calculus

In this section we introduce two proof systems for the linear-time $\mu$-calculus. The first one can be viewed as a refinement of the system of (Dax et al. 2006) system while the second is a classical and modal version of $\mu\mathsf{MALL}$ (Baelde 2012).

### 2.1 Linear-time $\mu$-calculus

**Definition 1.** *Let $\mathcal{V} = \{X, Y, \dots\}$ be a set of variables and $\mathcal{P} = \{\mathfrak{p}, \mathfrak{q}, \dots\}$ a set of atoms. Linear-time $\mu$-calculus* **formulas** *$F, G, H, \dots$ are given by:*

$$F ::= \top \mid \bot \mid \mathfrak{p} \mid F^\perp \mid F \vee F \mid F \wedge F \mid \bigcirc F \mid \mu X.F \mid \nu X.F \mid X$$

*In formulas $\mu X.F$ and $\nu X.F$, the variable $X$ must occur only positively in $F$,* i.e.*, under an even number of negations ($\bullet^\perp$). Bound and free variables are defined as usual and substitution is capture-avoiding. The subformula ordering is denoted $\leq$ and $\mathsf{fv}(\bullet)$ denotes free variables.*

**Definition 2.** *The* **Fischer-Ladner closure** *of a formula $F$, denoted by $\mathsf{FL}(F)$, is the least set of formulas such that $F \in \mathsf{FL}(F)$ and, whenever $G \in \mathsf{FL}(F)$,*

- *$G_1, G_2 \in \mathsf{FL}(F)$ if $G = G_1 \vee G_2$ or $G = G_1 \wedge G_2$;*
- *$G_1 \in \mathsf{FL}(F)$ if $G = \bigcirc G_1$ or $G = G_1^\perp$;*
- *$B[G/X] \in \mathsf{FL}(F)$ if $G = \sigma X.B$ for $\sigma \in \{\nu, \mu\}$.*

Formulas of $\mathsf{FL}(F)$ are induced by traversals of $F$ with the possibility of jumping from a variable to the fixed point combinator that introduced it. Due to this ability to cycle, there are infinitely many such traversals. However, cycling in a traversal induces the

$$\frac{}{F \vdash F}\ (\mathsf{Ax}) \qquad \frac{\Gamma, F \vdash \Delta \qquad \Gamma \vdash F, \Delta}{\Gamma \vdash \Delta}\ (\mathsf{Cut}) \qquad \frac{\Sigma \vdash \Theta}{\bigcirc\Sigma \vdash \bigcirc\Theta}\ (\bigcirc)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ (\mathsf{W_l}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ (\mathsf{W_r}) \qquad \frac{\Gamma \vdash F, \Delta}{\Gamma, F^\perp \vdash \Delta}\ (\mathsf{Neg_l}) \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma \vdash F^\perp, \Delta}\ (\mathsf{Neg_r})$$

$$\frac{\Gamma, F \vdash \Delta \qquad \Gamma, G \vdash \Delta}{\Gamma, F \vee G \vdash \Delta}\ (\vee_\mathsf{l}) \qquad \frac{\Gamma \vdash F, G, \Delta}{\Gamma \vdash F \vee G, \Delta}\ (\vee_\mathsf{r}) \qquad \frac{}{\Gamma, \bot \vdash \Delta}\ (\bot)$$

$$\frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \wedge G \vdash \Delta}\ (\wedge_\mathsf{l}) \qquad \frac{\Gamma \vdash F, \Delta \qquad \Gamma \vdash G, \Delta}{\Gamma \vdash F \wedge G, \Delta}\ (\wedge_\mathsf{r}) \qquad \frac{}{\Gamma \vdash \top, \Delta}\ (\top)$$

Figure 1: Inference rules for propositional connectives.

$$\frac{\Gamma, F[\sigma X.F/X] \vdash \Delta}{\Gamma, \sigma X.F \vdash \Delta}\ (\sigma_\mathsf{l}) \qquad \frac{\Gamma \vdash G[\sigma X.G/X], \Delta}{\Gamma \vdash \sigma X.G, \Delta}\ (\sigma_\mathsf{r})$$

Figure 2: Fixed point rules for the $\mu\mathsf{LK}^\infty$ proof system.

same formula as is obtained from the traversal before the cycle. Thus all formulas of $\mathsf{FL}(F)$ can be obtained from acyclic traversals, and there are finitely many.

**Definition 3.** *The* **semantics** $[\![F]\!]^u_\rho$ *of a formula $F$ with respect to an $\omega$-word $u$ over $\Sigma = 2^\mathcal{P}$ (*i.e.*, $u \in \Sigma^\omega$) and a valuation $\rho : \mathcal{V} \mapsto 2^\omega$ is a subset of natural numbers inductively defined as follows:*

$[\![\top]\!]^u_\rho = \omega \quad [\![\bot]\!]^u_\rho = \emptyset \quad [\![F^\perp]\!]^u_\rho = \omega \setminus [\![F]\!]^u_\rho \quad [\![X]\!]^u_\rho = \rho(X)$

$[\![\mathfrak{p}]\!]^u_\rho = \{i \in \omega \mid \mathfrak{p} \in u_i\} \qquad [\![\bigcirc F]\!]^u_\rho = \{i \in \omega \mid i+1 \in [\![F]\!]^u_\rho\}$

$[\![F \vee G]\!]^u_\rho = [\![F]\!]^u_\rho \cup [\![G]\!]^u_\rho \quad [\![F \wedge G]\!]^u_\rho = [\![F]\!]^u_\rho \cap [\![G]\!]^u_\rho$

$[\![\nu X.F]\!]^u_\rho = \bigcup\{\ W \subseteq \omega \mid W \subseteq [\![F]\!]^u_{\rho[X \leftarrow W]}\ \}$

$[\![\mu X.F]\!]^u_\rho = \bigcap\{\ W \subseteq \omega \mid [\![F]\!]^u_{\rho[X \leftarrow W]} \subseteq W\ \}$

*The set of models of $F$ (wrt. $\rho$) is defined to be $\mathcal{M}(F)_\rho = \{u \in \Sigma^\omega \mid 0 \in [\![F]\!]^u_\rho\}$.*

### 2.2 Infinitary proof system $\mu\mathsf{LK}^\infty$

Our proof systems are based on sequent calculus. Depending on requirements, sequents are sometimes viewed as sets of formulas, sometimes as multisets. Neither viewpoint is satisfying when the computational contents of proofs matters; one needs to go further and distinguish between different *occurrences* of a formula. We follow this approach here, viewing sequents $\Gamma \vdash \Delta$ as pairs of sets of occurrences of formulas. This is for two reasons. First, we ultimately seek to develop a full proof-theoretical account of $\mu$-calculus. Second and more concretely, our completeness argument involves a translation result that crucially relies on the precise tracking of formulas through deduction steps.

**Definition 4.** *A* **pre-proof** *of $\mu\mathsf{LK}^\infty$ is a possibly infinite tree, coinductively generated by the rules of Figures 1 and 2.*

It is easy to see that pre-proofs are not sound: one may for instance derive $\vdash \mu X.X$. To obtain proper proofs from pre-proofs, we add a validity condition which (unlike the rules of Figure 2) reflects the different meaning of $\mu$ and $\nu$.

**Definition 5.** *Let $r$ be a rule of conclusion $c$ and let $p$ be one of its premises. If $F$ is a formula occurrence in $c$, and $G$ is one in $p$, we write $F \xrightarrow{c,r,p} G$ if (i) $F$ is not principal in $r$ and $F = G$ or (ii) $F$ is principal in $r$ and $G$ results from the application of $r$ to $F$.*

**Example 6.** *We have $F \wedge G \xrightarrow{c,\wedge_r,p} G$ when $c = \Gamma \vdash F \wedge G, \Delta$ and $p = \Gamma \vdash G, \Delta$.*

**Definition 7.** *Let $\gamma = (s_i)_{i\in\omega}$ be an infinite branch in a pre-proof of $\mu\mathsf{LK}^\infty$, and let $(r_i)_{i\in\omega}$ be the corresponding instances of inference rules. A* **thread** *$t$ in $\gamma$ is a sequence of formula occurrences $(F_i)_{i\in\omega}$ such that $F_i \xrightarrow{s_i,r_i,s_{i+1}} F_{i+1}$. The set of formulas that occur infinitely often in $(F_i)_{i\in\omega}$ admits a minimum wrt. the subformula ordering, we denote it by $\mathsf{min}(t)$. A thread $t$ is* **valid** *if there are infinitely many indices $i$ such that $F_i$ is principal in $r_i$, and $\mathsf{min}(t)$ is a $\nu$ formula (resp. $\mu$ formula) occurring infinitely often on the right-hand side (resp. left-hand side) along the thread.*

$$\frac{\Gamma, S \vdash \Delta \qquad F[S/X] \vdash S}{\Gamma, \mu X.F \vdash \Delta}\ (\mu_\mathsf{l}) \qquad \frac{\Gamma \vdash F[\mu X.F/X], \Delta}{\Gamma \vdash \mu X.F, \Delta}\ (\mu_\mathsf{r})$$

$$\frac{\Gamma, F[\nu X.F/X] \vdash \Delta}{\Gamma, \nu X.F \vdash \Delta}\ (\nu_\mathsf{l}) \qquad \frac{\Gamma \vdash S, \Delta \qquad S \vdash F[S/X]}{\Gamma \vdash \nu X.F, \Delta}\ (\nu_\mathsf{r})$$

Figure 3: Fixed point rules for the $\mu\mathsf{LK}$ proof system.

**Definition 8.** *The* **proofs** *of $\mu\mathsf{LK}^\infty$ are those pre-proofs in which every infinite branch admits a valid thread. We call $\mu\mathsf{LK}^\omega$ the subsystem of $\mu\mathsf{LK}^\infty$ where derivations are regular,* i.e.*, possibly infinite but with only finitely many subderivations.*

The weakening rules $(\mathsf{W_l})$ and $(\mathsf{W_r})$ are notably useful to obtain the following two derived rules:

**Proposition 9.** *The following rules are derivable in $\mu\mathsf{LK}^\infty$:*

$$\frac{}{\Gamma, F \vdash F, \Delta}\ (\mathsf{Ax}) \qquad \frac{\Sigma \vdash \Theta}{\Gamma, \bigcirc\Sigma \vdash \bigcirc\Theta, \Delta}\ (\bigcirc)$$

Classic tableaux arguments can be adapted to show that the infinitary proof system is sound and complete for the linear-time $\mu$-calculus. Roughly, a valid sequent admits a pre-proof that can be constructed by applying rules in an invertible fashion, and infinite branches of that pre-proof must be valid, since we can extract counter-models from invalid infinite branches. Conversely, the existence of a counter-model allows one to find an invalid infinite branch in any pre-proof of an invalid sequent.

**Remark 10.** *The system $\mu\mathsf{LK}^\omega$ is very close to the one presented by Dax, Hofmann and Lange (Dax et al. 2006), which we shall call $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ in the following. The essential difference is that sequents of $\mu\mathsf{LK}^\omega$ are sets of occurrences (roughly, multisets of formulas) while those of $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ are sets of formulas.*

*If this difference may look minor at first glance, it is not trivial how one could transform a $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ derivation into one of $\mu\mathsf{LK}^\omega$. Indeed, if we choose to keep all formula occurrences in sequents, we may get a non-regular proof, and weakening some occurrences away may a priori break validity, as one can see in the example of Appendix B. We will come back to this issue in Section 4.*

### 2.3 Finitary proof system $\mu\mathsf{LK}$

Unlike the previous system, which requires a non-trivial, global validity criterion, the validity of a $\mu\mathsf{LK}$ proof is easily established by checking local inference rules.

**Definition 11.** *The proofs of $\mu\mathsf{LK}$ are finite trees inductively generated from the rules of Figures 1 and 3.*

The fixed point rules of Figure 3 express that $\mu X.F$ is the least pre-fixed point of $X \mapsto F$, and dually for $\nu X.F$. From this it can be shown that $\mu$ and $\nu$ respectively form least and greatest fixed points: $(\sigma_\mathsf{l})$ and $(\sigma_\mathsf{r})$ are derivable in $\mu\mathsf{LK}$. The particular formulation of the rules is such that the system admits cut elimination, a result that can be shown by adapting arguments from (Baelde 2012). It is also easy to see that this proof system is sound and complete for linear-time $\mu$-calculus, since it is equivalent (in terms of provability) to Kaivola's axiomatization (Kaivola 1995). In order to get a feel for our two proof systems, and their relationships, an

example is developed in Appendix A. We conclude this section by introducing a useful construction in $\mu\mathsf{LK}$, called functoriality:

**Proposition 12.** *Let $B$ be a formula with $\mathsf{fv}(B) = X$, $X$ occurring positively in $B$ and let $P_1$, $P_2$ be two closed formulas. The following* functoriality *rule is derivable in $\mu\mathsf{LK}$.*

$$\frac{P_1 \vdash P_2}{B[P_1/X] \vdash B[P_2/X]}\ (\mathsf{Functo})$$

## 3. Encoding $\omega$-automata in linear-time $\mu$-calculus

In this section, we briefly recall the definitions of Büchi and parity automata and define our encoding of these automata into linear-time $\mu$-calculus formulas. For a fixed finite set of atoms $\mathcal{P}$, we consider automata over the alphabet $\Sigma = 2^{\mathcal{P}}$, whose elements will simply be denoted by $a$, $b$, etc. We shall translate these subsets of atoms as formulas by setting $[a] := (\wedge_{\mathfrak{p}\in a}\mathfrak{p}) \wedge (\wedge_{\mathfrak{q}\notin a}\mathfrak{q}^{\perp})$. We eventually simply write $a$ for $[a]$.

**Definition 13.** *A* **parity automaton** *is a tuple $\mathcal{A} = (Q, q_I, \delta, c)$, where $Q$ is a finite set of states, $q_I \in Q$ is the initial state, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ is the transition relation and $c : Q \to \omega$ assigns a* **priority** *to each state. A* **run** *of $\mathcal{A}$ on a word $\alpha = (a_i)_{i\in\omega} \in \Sigma^\omega$ is a sequence $\rho = (q_i)_{i\in\omega}$ such that $q_0 = q_I$ and $q_{i+1} \in \delta(q_i, a_i)$ for $i \geq 0$. $\mathsf{Run}_{\mathcal{A}}(\alpha)$ denotes the set of runs of $\mathcal{A}$ on $\alpha$. For an infinite sequence $\rho$, we denote by $\mathsf{Inf}(\rho)$ the set of symbols that occur infinitely often in $\rho$. We say that $\mathcal{A}$* **accepts** *$\alpha$ if there exists a $\rho \in \mathsf{Run}_{\mathcal{A}}(\alpha)$ such that $\min\{\, c(q) \mid q \in \mathsf{Inf}(\rho)\,\}$ is even. The* **language** *of $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) = \{\, \alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\,\}$.*

**Definition 14.** *A* **Büchi automaton** *is a parity automaton $(Q, q_I, \delta, c)$ such that the image of $c$ is restricted to $\{\,0, 1\,\}$. In that context, we say that a state $q$ is* **accepting** *when $c(q) = 0$.*

**Definition 15** (Encoding of parity automata)**.** *Let $\mathcal{A} = (Q, q_I, \delta, c)$ be a parity automaton. We assume a collection of variables $(X_q)_{q\in Q}$. We define the formula $[q]^\Gamma$ encoding the state $q \in Q$ under the environment $\Gamma$ consisting of a list of states, as follows:*

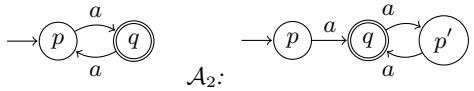
$$[q]^\Gamma = X_q \quad \text{if} \begin{cases} \Gamma = q_1, \ldots, q_n, q, \Gamma' \text{ and} \\ q_i \neq q, c(q_i) \geq c(q) \text{ for all } 1 \leq i \leq n, \end{cases}$$

$$[q]^\Gamma = \sigma X_q.\ \textstyle\bigvee_{a\in\Sigma, q'\in\delta(q,a)} [a] \wedge \bigcirc [q']^{q,\Gamma} \quad \text{otherwise, with } \sigma = \nu \text{ iff } c(q) \text{ is even.}$$

*When unspecified, $\Gamma$ is taken to be empty. We finally set $[\mathcal{A}] = [q_I]$.*

The encoding builds on the well-known correspondence between the $\mu$-calculus and parity games. The key ingredient is the side condition of the first case of the definition, relying on $\Gamma$. The aim here is to bridge the gap between the acceptance condition on runs of the automaton and the validity condition on threads. The latter is almost a parity condition, but with a parity ordering corresponding to the subformula ordering. To obtain a match between the two orderings, we need to control the formation of cycles.

**Example 16.** *Consider the following Büchi automata, where even states are double-circled:*




$\mathcal{A}_1$: $\mathcal{A}_2$:

*A naive encoding of $\mathcal{A}_1$ would be $\mu X_p.(a \wedge \bigcirc(\nu X_q.a \wedge \bigcirc X_p))$. It is incorrect since that formula is equivalent to $\perp$. Syntactically, the problem is that the infinite traversal of the formula corresponds to a cycle on $X_p$, with a regeneration of $X_q$ at each step. In a sense, $X_q$ is hidden by $X_p$ in this encoding. Our encoding of $\mathcal{A}_1$ is the same as for $\mathcal{A}_2$, which is an unfolding of $\mathcal{A}_1$. More generally, the encoding can be seen as duplicating states so as to avoid that an accepting state is hidden by a non-accepting one:*

$$[\mathcal{A}_1] = [\mathcal{A}_2] = \mu X_p.\ a \wedge \bigcirc(\nu X_q.\ a \wedge \bigcirc(\mu X_{p'}.\ a \wedge \bigcirc X_q)))$$

**Proposition 17.** *For any parity automaton $\mathcal{A}$, $\mathcal{M}([\mathcal{A}]) = \mathcal{L}(\mathcal{A})$.*

We now transpose automata-theoretic notions to an appropriate class of formulas that contains the image of our encoding. This will be useful since most of our work is done directly on formulas, and it allows us to state simpler and slightly more general results.

**Definition 18. Disjunctive formulas** *are defined as follows:*

$$F ::= X \mid \sigma X.\ \vee_{i\in I}\,(a_i \wedge \bigcirc F_i) \qquad \text{with} \qquad X \in \mathcal{X},\ a_i \in \Sigma.$$

*The set of disjunctive formulas is denoted by $\mathcal{F}_\vee$. A* **Büchi formula** *is any $F \in \mathcal{F}_\vee$ such that, for all $\mu X.G \leq F$, there is no $\nu Y.H \leq G$ with $X \in \mathsf{fv}(\nu Y.H)$.*

Our disjunctive formulas are similar to the ones of (Walukiewicz 2000). They present the form of non-determinism that makes it difficult to translate circular proofs into finite proofs: their negations are not aconjunctive in the sense of (Kozen 1983).

**Definition 19.** *Let $F$ be a disjunctive formula. The* **states** *of $F$ are the formulas of $Q(F) := \mathsf{FL}(F) \cap \mathcal{F}_\vee$. For $G, H \in Q(F)$ we write $G \xrightarrow{a} H$ if $G = \sigma X.\ \vee_{i\in I} F_i$ and $F_i = (a \wedge \bigcirc H')$ for some $i \in I$ such that $H = H'[G/X]$. We set $a^{-1}G := \{H : G \xrightarrow{a} H\}$, and define $\delta(F)$ to be $(G, a) \mapsto a^{-1}G$. A parity automaton $\mathcal{A}_F$* **is associated to** *$F$ if it is of the form $(Q(F), F, \delta(F), c)$ where $c$ is any priority assignment such that (i) $G$ is a $\nu$-formula iff $c(G)$ is even and (ii) if $G_1$ and $G_2$ are co-accessible (*i.e.*, $G_i \in FL(G_{3-i}), i \in \{1, 2\}$), $G_1 \leq G_2$ iff $c(G_1) \leq c(G_2)$.*

**Proposition 20.** *For any closed disjunctive formula $F$, and any associated parity automaton $\mathcal{A}_F$, one has $\mathcal{M}(F) = \mathcal{L}(\mathcal{A}_F)$.*

Note that Definition 19 may be understood as dealing with formulas or with occurrences of formulas. The second, more precise viewpoint should be taken in the rest of the paper, as it is the one that makes sense proof theoretically. For instance, in the next proposition, $a^{-1}G$ (where $G$ is a formula occurrence) really extracts a set of sub-occurrences of (the unfolding of) $G$.

**Proposition 21.** *For any closed disjunctive formulas $F$, $G_1$, …, $G_n$, the following rule is derivable in $\mu\mathsf{LK}^\omega$:*

$$\frac{\{F_a \vdash a^{-1}G_1, \ldots, a^{-1}G_n\}_{a\in\Sigma, F\xrightarrow{a}F_a}}{F \vdash G_1, \ldots, G_n}\ (\to)$$

*Proof.* We restrict to $n = 1$ for clarity, but the general case is similar by applying the right-hand side rules successively on all $G_i$ formulas. In that case, assuming $F = \sigma X.\bigvee_{i\in I}(a_i \wedge \bigcirc F_i)$, we derive $(\to)$ as follows:

$$\frac{\dfrac{\pi_1}{a_1, \bigcirc F_1[F/X] \vdash G_1} \quad \ldots \quad \dfrac{\pi_n}{a_n, \bigcirc F_n[F/X] \vdash G_1}}{\sigma X.\ \bigvee_{i\in I}(a_i \wedge \bigcirc F_i) \vdash G_1}\ (\sigma_\mathsf{l}),(\vee_\mathsf{l}),(\wedge_\mathsf{l})$$

with $\pi_i$ being defined as:

$$\cfrac{\cfrac{\cfrac{F_i[F/X] \vdash a_i^{-1}G_1}{a_i, \bigcirc F_i[F/X] \vdash \{\, \bigcirc G_k^i \,\}_{G_1 \xrightarrow{a_i} G_k^i}}\ (\bigcirc)}{\cfrac{}{a_i, \bigcirc F_i[F/X] \vdash \{\, a_i \wedge \bigcirc G_k^i \,\}_{G_1 \xrightarrow{a_i} G_k}}\ (\wedge_\mathsf{r}),(\mathsf{Ax}),(\mathsf{W}_\mathsf{l})}}{a_i, \bigcirc F_i[F/X] \vdash G_1}\ (\sigma_\mathsf{r}),(\vee_\mathsf{r}),(\mathsf{W}_\mathsf{r})$$

Note that threads of this derived inference rule do not encounter fixed-point formulas except the ones visible at the premise and conclusion of the rule. Therefore, one can ignore the internal construction of the rule when checking the validity of pre-proofs. □

## 4. Parity automata inclusions in $\mu\mathsf{LK}^\omega$

We shall establish the completeness of $\mu\mathsf{LK}^\omega$ for automata inclusions. Here we can actually work with the full class of parity automata, which we exploit later.

**Theorem 22.** *For any disjunctive formulas $F$ and $G$ one has:*

$$\mathcal{L}(F) \subseteq \mathcal{L}(G) \quad \textit{iff} \quad F \vdash G \textit{ is derivable in } \mu\mathsf{LK}^\omega.$$

The soundness of $\mu\mathsf{LK}^\infty$ implies one direction of the result. For the other direction, the completeness of $\mu\mathsf{LK}^\infty$ is not enough to conclude. Proposition 21 indicates that we can embed the powerset construction in the logic, suggesting a natural strategy for building proofs of language inclusions. But we face a problem here due to the fact that our sequents are not made of sets of formulas, but keep track of distinct occurrences of formulas. This means that we are in fact considering a "powermultiset" construction, accounting for all possible runs rather than reachable states only. If we apply this naive construction, keeping all the copies of a state coming from different states, we are certainly going to get a $\mu\mathsf{LK}^\infty$ proof, but this proof has no chance of being regular in general, since sequents will become larger and larger.

In order to recover regularity, we have to weaken some occurrences. The challenge is to do so whilst preserving validity: for each word accepted by $F$, we need to preserve at least one valid thread among the ones witnessing the fact that $G$ also accepts the word. In other words, we need to select a bounded subset of the possible runs of $G$. The recipe for such a selection can be found in the Safra trees used to determinize $\omega$-word automata. In order to translate nondeterministic parity automata into deterministic Rabin automata, two distinct problems are solved in Safra's construction. The first problem is to refine the powerset construction in a way that bounds the set of alternative runs to consider. Safra trees achieve this, keeping only a *single* run per reachable state. Once this is achieved, determinization is realized in an abstract sense, but a second problem remains: that of designing a Rabin acceptance condition for the refined powerset (the Safra tree) that matches the acceptance conditions of the individual threads that it represents. While the first problem is precisely the one that we are facing in order to obtain regular proofs, the second one is not immediately relevant to our task. In the simple case of Büchi automata, this remark could motivate the use of simpler tools than Safra trees, *e.g.*, the reduced trees of (Kähler and Wilke 2008). In the case of parity automata, it seems difficult to find such a clear cut decomposition, and so we directly use Safra trees for parity automata. Anyway, we shall see that the structure of the trees pertaining to the acceptance condition is a useful device for reasoning about our proofs.

We specialize Safra's construction for Streett automata (Safra 1992; Grädel et al. 2002) to parity automata, in order to obtain a more direct presentation. This is important since the precise construction will impact the structure of $\mu\mathsf{LK}^\omega$ derivations.

We assume a fixed formula $G$ of interest, and an associated automaton $\mathcal{A}_G = (Q(G), G, \delta(G), c)$. We consider without loss of generality that the priorities of $\mathcal{A}_G$ are in $P = \{0, 1, \ldots, 2k, 2k+1\}$.

**Definition 23.** *A* **Safra tree** *is a finitely branching tree whose leaves are labelled with non-empty subsets of $Q(G)$ and whose edges are labelled by priorities from $P$. Descendants of a node are ordered, from left (younger) to right (older). Leaves' labels are pairwise disjoint. Every internal node has at least one outgoing edge with an odd label. On every branch, edge labels appear in increasing order starting from the root. Further, only even labels can appear more than once on a branch and if two labels occur on a branch then odd labels in between must occur too. Each node is identified by a name in a finite set $N$; we say that $v \in N$ is a node in $T$ if $T$ contains a node of name $v$. We denote by $e_T(v)$ the union of the labels of all leaves of the subtree of $T$ rooted in node $v$.*

Note that the conditions on leaves and edges alone imply that only finitely many trees exist, so that the finiteness condition on $N$ is not constraining. The role of node names is to be able to track nodes through the modifications of the trees corresponding to automata transitions. Given a Safra tree $T$ and a letter $a$, we now describe the transition function $\Delta(T, a)$ by the following procedure. A detailed example of that construction is given in Appendix C. Whenever we create a new node in the procedure, we assume that it is given a fresh name, *i.e.*, one that does not yet occur in the whole tree.

(1) For any leaf node $v_0$ of $T$, let $2m+1$ be the least odd label that does not occur from the root to $v_0$ (skip this step if all odd labels occur). Create nodes $v_1, \ldots, v_{k-m+1}$ such that for each $i = 0, \ldots, k-m$, $v_{i+1}$ is the son of $v_i$, with an edge $(v_i, v_{i+1})$ labelled $2(m+i)+1$. The leaf $v_{k-m+1}$ takes the label of $v_0$.

(2) Replace each leaf label $S$ by $a^{-1}S$.

(3) For each leaf $v$ and state $q$ appearing in it, let $(w, w')$ be the edge appearing in the path from the root to $v$, that is labelled by $c(q)$ if $c(q)$ is odd, $c(q)+1$ otherwise. Remove $q$ from the label of $v$ and append to $w$ a new child leaf $v'$, to the right, labelled $\{q\}$, with the edge $(w, v')$ labelled $c(q)$.

(4) Whenever a state $q$ belongs to the labels of two distinct leaves $w_1$ and $w_2$, let $b_1, b_2$ be two branches starting from the root and ending in $w_1, w_2$ respectively. Let $v$ be the node on which these branches fork, and $v_i$ be the child of $v$ in $b_i$, $i \in \{1, 2\}$. If the label of the edge $(v, v_1)$ is strictly greater than that of $(v, v_2)$ then remove $q$ from $w_1$ and conversely. If they have the same label, remove $q$ from the rightmost $w_i$.

(5) Remove any subtree whose leaves all have empty labels.

(6) If after the previous steps all edges going from a node $v$ to its children are labelled by the same even priority, make $v$ a leaf and label it $e_T(v)$.

Safra's construction yields a deterministic Rabin automaton equivalent to the original parity automaton. We only need the completeness direction of that equivalence, established next.

**Remark 24.** *The following properties are invariants of our Safra construction,* i.e.*, they are satisfied by the initial singleton trees and are preserved by $\Delta$:*

- *If $l$ is a leaf of a Safra tree $T$ and $q$ appears in the label of $l$, then $c(q) \geq p$, where $p$ is any edge label from the root to $l$.*
- *If $v$ is an internal node in a Safra tree $T$, then there is an even priority $p$ such that all the outgoing edges from $v$ in $T$ are either labelled $p$ or $p+1$. In the following, this even priority will be denoted by $p_T(v)$, or simply $p(v)$ when $T$ is obvious or irrelevant.*
- *Moreover, if $T' = \Delta(T, a)$ and $v$ appears in both $T$ and $T'$ as an internal node, then $p_T(v) = p_{T'}(v)$.*

**Proposition 25.** *Let $\alpha := (a_i)_{i\in\omega} \in \Sigma^\omega$ and $\rho := (G_i)_{i\in\omega}$ be a run of $\mathcal{A}_G$ over $\alpha$. Let $T_0$ be the Safra tree with a single leaf node labelled $\{G_0\}$, and $T_{i+1} := \Delta(T_i, a_i)$ for all $i \in \omega$. If $\rho$ is accepting then there is a node $v \in N$ and an index $j$ such that for all $i > j$, $v$ is a node in $T_i$ and $v$ becomes a leaf infinitely often.*

*Proof.* Let us first prove the following assertion:

**Assertion:** Let $v \in N$. If there is $i_v \in \omega$ such that for all $i > i_v$, $v$ is a node of $T_i$ and $G_i \in e_{T_i}(v)$, then there are two possibilities:

1. The node $v$ becomes a leaf infinitely often.

2. There is $w \in N$ and $i_w \in \omega$ such that for all $i > i_w$, $w$ is a son of $v$ in $T_i$ and $G_i \in e_{T_i}(w)$.

**Proof of the assertion:** Let $m$ be the minimal priority that appears infinitely often in $\rho$. Suppose that $v$ does not become a leaf infinitely often: let $k$ be an index such that for all $i > k$, the node $v$ is never a leaf in $T_i$ and $c(G_i) \geq m$. By Remark 24, there is an even priority $p$ such that $p = p_{T_i}(v)$ for all $i > i_v$. Since $G_i \in e_{T_i}(v)$ for all $i > k$, and again by Remark 24, we have that $p \leq m$. For all $i > k$, $G_i \in e_{T_i}(w_i)$ for some child $w_i$ of $v$; we seek to establish that $(w_i)_{i>k}$ is eventually constant. We distinguish two cases:

**(i) If $\mathbf{p < m}$,** we actually have $p + 1 < m$. Let $T'_i$ be the tree obtained after steps (1–3) of the construction of $T_{i+1} = \Delta(T_i, a_i)$. Because $c(G_{i+1}) > p + 1$ we still have $G_{i+1} \in e_{T'_i}(w_i)$. We now consider the rest of the construction, from $T'_i$ to $T_{i+1}$. Step (4) may remove the occurrence of $G_{i+1} \in e_{T'_i}(w_i)$ in favor of another one in $eT'_i(w_{i+1})$ for $w_{i+1} \neq w_i$. However, this can only happen if (a) the edge $(v, w_{i+1})$ is labelled $p$ while $(v, w_i)$ is labelled $p + 1$, or (b) the edge labels are the same but the rank of $w_{i+1}$ among the children of $v$ is less than that of $w_i$. Thus, this can happen only finitely many times, and the sequence $(w_i)_i$ is eventually constant. (Obviously, Step (5) is irrelevant in this argument, and Step (6) cannot happen because $v$ never becomes a leaf by assumption.)

**(ii) Otherwise, $\mathbf{p = m}$.** We first show that there is some $i$ such that $(v, w_i)$ is labelled $p$. Consider any position $j > k$ such that $c(G_i) = m$. If $(v, w_i) = p + 1$ then $G_{i+1}$ will be moved by step (3) of the construction of $T_{i+1}$ to a new child $w'$ of $v$, with an edge $(v, w')$ labelled $p$. Then step (4) may not keep the occurrence of $G_{i+1}$ in $w'$, but it will preserve an occurrence in a child $w''$ with $(v, w'')$ labelled $p$ too. Next we observe that, once $(v, w_i)$ is labelled $p$, we have $(v, w_j)$ labelled $p$ as well for all $j \geq i$. This follows from a simple inspection of the procedure, arguing as above. From this point on, we can only have $w_j \neq w_{j+1}$ because of step (4) but, as observed before, this can only decrease the rank of $w_{j+1}$ among the children of $v$. This can only happen finitely often, which concludes the proof.

To prove the initial proposition, we iterate the assertion starting from the root node, which satisfies trivially the hypothesis of the lemma, and iterate it as many times as necessary to find the wanted node. This iteration is bound by the maximal height of Safra trees. □

For a Safra tree $T$, we define $\Phi(T)$ to be the union of the labels of the leaves of $T$ (which contain formula occurrences), forgetting all the tree structure. In the following, when unambiguous, we will sometimes write $T$ instead of $\Phi(T)$. The Safra tree structure corresponds to a refined powerset construction, in the sense that, if $T$ is a Safra tree and $a \in \Sigma$, $\Phi(\Delta(T, a)) \subseteq a^{-1}(\Phi(T))$. We now observe that Safra's construction can be immediately adapted to proof theory, in a way that yields proofs of inclusions for disjunctive formulas.

**Proposition 26.** *Given a disjunctive formula $F$ and a Safra tree $T$, the following rule is derivable:*

$$\frac{\{\ F_a \vdash \Phi(\Delta(T, a))\ \}_{a \in \Sigma, F \xrightarrow{a} F_a}}{F \vdash \Phi(T)}\ (\mathsf{S})$$

*Proof.* Since $\Phi(\Delta(T, a)) \subseteq a^{-1}(\Phi(T))$, rule (S) can be obtained by applying rule $(\rightarrow)$ of Proposition 21 and some weakenings. It is however crucial to perform those weakenings in a way that selects appropriate threads in the powerset construction, to be able to replay Safra's argument in the rest of the development. This is done in the obvious manner, once one understands that Safra's construction is selecting runs, and that threads are runs. □

**Definition 27** ($\Pi(F, T), \Pi(F, G)$)**.** *Given a disjunctive formula $F$ and a Safra tree $T$, we coinductively define $\Pi(F, T)$ to be the following $\mu\mathsf{LK}^\omega$ pre-proof:*

$$\frac{\left\{\dfrac{\Pi(F_a, \Delta(T, a))}{F_a \vdash \Phi(\Delta(T, a))}\right\}_{a \in \Sigma, F \xrightarrow{a} F_a}}{F \vdash \Phi(T)}\ (\mathsf{S})$$

*In other words, it is the infinite derivation obtained by repeatedly applying rule* (S)*. It is obviously regular, as only finitely many $F$ and $T$ may be encountered. Note that $\Pi(F, T)$ is a standard $\mu\mathsf{LK}^\omega$ pre-proof, whose sequents are sets of occurrences, but those sequents have been obtained by forgetting the structure of Safra trees which form the blueprint of the construction. If $G$ is a disjunctive formula, we set $\Pi(F, G) := \Pi(F, T)$ where $T$ is the single-node Safra tree with label $\{G\}$.*

**Proposition 28.** *Let $F, G \in \mathcal{F}_\vee$. If $\mathcal{L}(F) \subseteq \mathcal{L}(G)$, then $\Pi(F, G)$ is a $\mu\mathsf{LK}^\omega$ proof.*

*Proof.* Let us establish the validity of each infinite branch $\gamma = (s_i)_{i \in \omega}$ of $\Pi(F, G)$. For each such branch, there is an $\omega$-word $\alpha = (a_i)_{i \in \omega}$ such that, for all $i$, $s_i = F_i \vdash \Phi(T_i)$ where $F_0 = F$, $T_0$ is the tree with a single node labeled by $\{G\}$, $F_i \xrightarrow{a_i} F_{i+1}$ and $\Delta(T_i, a_i) = T_{i+1}$. Hence $\rho_F := (F_i)_{i \in \omega}$ corresponds to a run of $F$ and $\rho_G := (T_i)_{i \in \omega}$ corresponds to a run of the Rabin automaton obtained by Safra's construction on $\mathcal{A}_G$.

If the run $\rho_F$ is not accepting, we have that $\mathsf{min}(\mathsf{Inf}(\rho_F))$ is a $\mu$-formula and thus $\rho_F$, seen as a thread, validates the branch $\gamma$.

Otherwise, the run $\rho_F$ is accepting, thus $\alpha \in \mathcal{L}(F)$, and $\alpha \in \mathcal{L}(G)$ by hypothesis. By Proposition 25, there is a node $v$ and an index $j$ such that $\forall i \geq j$, $v$ is a node of $T_i$ and $v$ appears as a leaf infinitely often. Now, let us extract a valid thread from this node. To do so, we prove the following assertion:

**Assertion 1:** Let $m > n \geq j$ such that the node $v$ is a leaf in $T_n$ and $T_m$, and it is not a leaf in $T_i$ for all $n < i < m$. For every $K \in e_{T_m}(v)$, there is a sequence of formulas $F_n, \ldots, F_m$ satisfying the following conditions:

- $F_m = K$;
- for all $i = n, \ldots, m-1$, $F_i \overset{s_i, (\mathsf{S}), s_{i+1}}{\longrightarrow} F_{i+1}$
- $\mathsf{min}(F_n, \ldots, F_m)$ is a $\nu$-formula.

We first show that the assertion allows us to conclude the main proof. Let $j < i_0 < i_1 < \ldots$ such that $v$ is a leaf in $T_i$ where $i > j$ iff $i = i_n$ for some $n \in \omega$. We define a tree $\mathcal{T}$ whose set of nodes $N$ is

$$N := \{r\} \cup \{\ (F, n) \mid n \in \omega, F \in e_{T_{i_n}}(v)\ \},$$

where $r$ is a distinguished root node, with an edge from $r$ to any node of the form $(F, 0)$, and an edge from $(H, n)$ to $(K, n+1)$ iff there is a thread $F_{i_n} F_{i_n+1} \ldots F_{i_{n+1}}$ such that $H = F_{i_n}$, $K = F_{i_{n+1}}$ and $\mathsf{min}(F_{i_n}, F_{i_n+1}, \ldots, F_{i_{n+1}})$ is a $\nu$-formula. By the previous assertion, every node except $r$ have at least a parent. The tree $\mathcal{T}$ is thus an infinite tree which is finitely branching, hence by König's lemma it has an infinite branch. It is easy to see that this branch corresponds to a valid thread.

Before proving Assertion 1, we will first enunciate and prove a stronger version of it. By Remark 24, there exists an even priority $p$ such that for all $i \geq j$, the outgoing edges from $v$ are either labelled $p$ or $p + 1$. We actually prove, by induction on $i = n, \ldots, m-1$ the following:

**Assertion 2:** Let $w$ be a child of $v$ in $T_i$ and $K \in e_{T_i}(w)$, then there are formulas $F_k, \ldots, F_i$ satisfying the following conditions:

- $F_i = K$;
- For all $l = n, \ldots, i-1$, $F_l \overset{s_l,(\mathsf{S}),s_{l+1}}{\longrightarrow} F_{l+1}$;
- For all $l = n, \ldots, i$, $F_l \in e_{T_l}(v)$;
- If $c$ is the label of the edge $(v, w)$, then $\mathsf{min}(c(F_{n+1}), \ldots, c(F_i)) = c$.

Let us prove Assertion 2. When $i = n$, the result is obvious, notably because $v$ is a leaf in $T_n$. Now, suppose that the result is true for $i$ and let us prove it for $i+1$. Let $w$ be a child of $v$ in $T_{i+1}$ and $K \in_{T_{i+1}} (w)$. The edge $(v, w)$ is labelled $c$. Notice first that $c(K) \geq c$, otherwise it would be absorbed by a parent of $v$.

There exists $L \in e_{T_i}(v)$ such that $L \overset{s_i,(\mathsf{S}),s_{i+1}}{\longrightarrow} K$. Actually, $L$ should belong to one of the children of $v$ in $T_i$; call it $w'$ and let $c'$ be the label of the edge connecting $v$ and $w'$ in $T_i$. We distinguish two cases:

1. If $w = w'$ we conclude by applying the induction hypothesis.
2. If $w \neq w'$, that is $w$ is a new child in $T_{i+1}$, created by Step 3, then we distinguish two cases: if $c = c'$, here again we apply directly the induction hypothesis. Otherwise, the node $w$ has been created because $c(F_{i+1}) = p$ and $c' = p + 1$. We then have that $c = p$. Here again we apply induction hypothesis to conclude.

We finally prove Assertion 1. Let $K \in e_{T_m}(v)$ and let $w$ be a child of $v$ in $T_{m-1}$ and $L$ a formula such that $L \in e_{T_{m-1}}(w)$ and $L \overset{s_{m-1},(\mathsf{S}),s_m}{\longrightarrow} K$. By the previous result instanciated for $i = m-1$, we have a thread $F_n, \ldots F_{m-1}$ such that $F_{m-1} = L$ and $\mathsf{min}(c(F_n), \ldots, c(F_{m-1})) = c$ where $c \in \{p, p+1\}$ is the label of the edge $(v, w)$. As $v$ is a leaf in $T_m$, it has been obtained after Step 6, which means that before Step 6, the tree $T_{m-1}$ has only ongoing edges labelled $p$. Let $w'$ be the child of $T_{m-1}$ just before Step 6 such that $K$ is in the label of this subtree rooted in $w'$. There are two possibilities:

- If $w' = w$, we have $c = p$ and, since $c(F_m) \geq p$, then $\mathsf{min}(c(F_n), \ldots, c(F_{m-1})) = p$.
- Otherwise, $w'$ has been created in Step 3, and $c(F_m) = p$. Hence we also have $\mathsf{min}(c(F_n), \ldots, c(F_m)) = p$.

Anyway, the thread $F_n, \ldots F_m$ satisfies $\mathsf{min}(c(F_n), \ldots, c(F_m)) = p$. Since $p$ is even, $\mathsf{min}(F_n, \ldots, F_m)$ is a $\nu$-formula. □

## 5. Büchi automata inclusions in $\mu\mathsf{LK}$

The problem of relating finite and infinite proof systems for fixed point logics is still insufficiently understood. In the case of modal $\mu$-calculi, some partial connections have been identified and exploited as part of completeness arguments, but no care has been taken to make these observations generic in proof-theoretical terms. Santocanale (Santocanale 2001, 2002) provided a technique to give a semantics to his circular proof system with fixed points in $\mu$-bicomplete categories. This technique can be used to translate purely additive circular proofs into (the purely additive fragment of) $\mu\mathsf{LK}$. Dealing with with multiplicative connectives or structural rules remains a challenge. We shall extend this technique to obtain a new partial translation result, that applies to Safra proofs, allowing us to finally obtain $\mu\mathsf{LK}$ proofs of Büchi inclusions. The translation is based on a combinatorial condition on threads, which makes it quite generic and directly applicable to other circular proof systems than $\mu\mathsf{LK}^\omega$.

### 5.1 A translation result from $\mu\mathsf{LK}^\omega$ to $\mu\mathsf{LK}$

We give a sufficient condition on $\mu\mathsf{LK}^\omega$ proofs which guarantees that they can be translated into $\mu\mathsf{LK}$ proofs, using an adaptation of Santocanale's procedure. The translation procedure works on any finite representation of regular derivations. To get a handle on this notion, we shall work with *annotated* $\mu\mathsf{LK}^\omega$ *derivations* throughout the whole section. Such derivations are simply $\mu\mathsf{LK}^\omega$ derivations where each sequent is given an annotation in a finite set, in such a way that regularity is not lost and if two sequents of a $\mu\mathsf{LK}^\omega$ derivation have the same annotation, they also have the same derivation. Later on, annotations will be used to keep track of the Safra trees that generated sequents in inclusion proofs, giving a formal way to distinguish identical sequents that originate from distinct Safra trees.

**Definition 29** (Translatable proofs)**.** *Let $\pi$ be an annotated $\mu\mathsf{LK}^\omega$ proof and $\gamma = (s_i)_{i\in\omega}$ an infinite branch of $\pi$. A thread $t = (F_i)_{i\in\omega}$ in $\gamma$ is said to be* **strongly valid** *if $t$ is valid and there is an annotated sequent $s_k$ such that, for all $i \geq k$ such that $s_i$ and $s_k$ correspond to the same annotated sequent, $F_i = \mathsf{min}(t)$ and $F_i$ is active in the rule of conclusion $s_i$. A proof is said to be* **translatable** *if every infinite branch contains a strongly valid thread.*

**Proposition 30.** *Let $\pi$ be a $\mu\mathsf{LK}^\omega$ proof of a sequent $s$. If $\pi$ is translatable then there is a proof of $s$ in $\mu\mathsf{LK}$.*

To prove this, we extend both $\mu\mathsf{LK}^\omega$ and $\mu\mathsf{LK}$ with an assumption rule:

$$\frac{}{\Gamma \vdash \Delta}\ (\mathsf{A})$$

We call the resulting systems $\mu\mathsf{LK}^{\omega\star}$ and $\mu\mathsf{LK}^\star$. The validity condition remains the same, as it is about infinite branches only.

**Definition 31.** *Let $\pi$ be an annotated proof in $\mu\mathsf{LK}^{\omega\star}$ or in $\mu\mathsf{LK}^\star$. We define $S_\pi$ as the set of annotated sequents appearing in $\pi$. We define the* **assumptions** *of $\pi$, written $\mathcal{A}(\pi)$, as the set of annotated conclusion sequents of* (A)*-rules in $\pi$. Finally, we define $C_\pi := S_\pi \setminus \mathcal{A}(\pi)$ and the* **complexity** *of $\pi$ as $\#\pi := \mathsf{card}(C_\pi)$.*

**Definition 32.** *Let $\pi$ be a proof in $\mu\mathsf{LK}^{\omega\star}$ or in $\mu\mathsf{LK}^\star$, and let $s$ be an annotated sequent. The proof $\pi^s$ is obtained from $\pi$ by replacing all subtrees rooted in $s$ by an assumption on $s$.*

*Proof of Proposition 30.* We actually establish that if $\Pi$ is a translatable $\mu\mathsf{LK}^{\omega\star}$ proof of a sequent $s$, there is a $\mu\mathsf{LK}^\star$ proof $\pi$ of $s$ such that $\mathcal{A}(\pi) \subseteq \mathcal{A}(\Pi)$. The proof is by induction on $\#\Pi$. When $\#\Pi = 0$, the derivation is reduced to an assumption on the conclusion sequent $s$, and the result obviously holds by using rule (A) on $s$ in $\mu\mathsf{LK}^\star$. Otherwise, we distinguish two cases.

***The proof is not "strongly connected".*** If there exist $s_1, s_2 \in C_\Pi$ such that no occurrence of sequent $s_2$ appears above an occurrence of $s_1$ in $\Pi$, let $\Pi_1$ be a sub-tree of $\Pi$ rooted in $s_1$ and $\Pi_2 = \Pi^{s_1}$. Both $\Pi_1$ and $\Pi_2$ are translatable and have strictly smaller complexity than $\Pi$. By induction hypothesis we obtain $\mu\mathsf{LK}^\star$ proofs $\pi_1$ of $s_1$ and $\pi_2$ of $s$, such that $\mathcal{A}(\pi_1) \subseteq \mathcal{A}(\Pi)$ and $\mathcal{A}(\pi_2) \subseteq \mathcal{A}(\Pi) \cup \{s_1\}$, which we plug together at the level of $s_1$ to get a proof $\pi$ of $s$ satisfying $\mathcal{A}(\pi) \subseteq \mathcal{A}(\Pi)$.

***The proof is "strongly connected".*** Otherwise, we can find an infinite branch $\gamma = (s_i)_{i\in\omega}$ containing all sequents of $C_\Pi$. Further, we choose this branch such that it contains any finite branch that connects two consecutive occurrences of a sequent; the reasons for this condition will become clear next.

By hypothesis, this branch admits a strongly valid thread $t = (F_i)_{i\in\omega}$. We assume that the minimum of the formulas occurring infinitely often in $t$ is a $\mu$ formula occurring on the left-hand sides of sequents; the case of a $\nu$ on the right is similar. Let $F := \mathsf{min}(t) = \mu X.B$. By strong validity, there is an annotated sequent $s' \in C_\Pi$ such that if $s_i = s'$ then $F_i = F$ and $F_{i+1} = B[F/X]$. We assume, without loss of generality, that $\Pi$ is rooted in $s'$, *i.e.*, $s = s'$.

The sequent $s$ is of the form $\Gamma, F \vdash \Delta$; this allows us to define the *environment* of $F$ in $s$:

$$\mathsf{env}(s) := (\bigvee_{G\in\Gamma} G^{\perp}) \vee (\bigvee_{H\in\Delta} H).$$

Note that $\Gamma, \mathrm{env}(s) \vdash \Delta$ is then obviously derivable. We finally consider the formula $I$, which will be shown to be an invariant of $F$:

$$I := \mu X.(B \wedge \mathsf{env}(s))$$

Observe that $I \vdash F$ is easily derivable using rule $(\mu_{\mathsf{l}})$ with $F$ itself as the invariant.

Let $\Pi_u$ be the immediate subderivation of $\Pi$; the conclusion of $\Pi_u$ is thus the premise of the $(\mu_{\mathsf{l}})$ rule applied to $F$ in $s$. Consider now the derivation $\Pi_u^s$. Its conclusion sequent contains an occurrence $B[F/X]$. We construct $\Theta$ from $\Pi_u^s$ by replacing that occurrence by $B[I/X]$ in the conclusion sequent, and simply propagating that substitution, unfolding $I$ when the corresponding $F$ is unfolded. Note that assumptions may be affected by the substitution, so we do not have $\mathcal{A}(\Theta) = \mathcal{A}(\Pi_u^s)$ in general. Still, one has $\#\Theta < \#\Pi$ since $C_\Theta$ does not contain the sequent $s$ and, if two annotated sequents were equal in $C_\Pi$, they are similarly impacted by the substitution and remain equal in $C_\Theta$. Moreover, $\Theta$ is still translatable. Thus, we obtain by induction hypothesis a $\mu\mathsf{LK}$ proof $\theta$ satisfying $\mathcal{A}(\theta) \subseteq \mathcal{A}(\Theta)$.

We now seek to adapt $\theta$ in order to obtain a $\mu\mathsf{LK}$ derivation $\beta$ such that $\mathcal{A}(\beta) \subseteq \mathcal{A}(\Pi)$. Every assumption $h$ of $\theta$ is an assumption of $\Theta$, which has been obtained from an assumption of $\Pi_u^s$. That original assumption is either an assumption of $\Pi$, or $s$. In each case, we modify the assumption $h$ of $\theta$ into a $\mu\mathsf{LK}$ derivation with assumptions in $\mathcal{A}(\Pi)$:

- If $h$ originates from an assumption $h_0$ of $\Pi$, some occurrences of $F$ in $h_0$ have been substituted for $I$ during the construction of $\Theta$. In other words we have $h_0 = (\Gamma[F/X] \vdash \Delta[F/X])$ and $h = (\Gamma[I/X] \vdash \Delta[I/X])$. Note that, by positivity of $F$, all substituted occurrences of $X$ must be positive in $\Gamma$ (resp. negative in $\Delta$). As observed before, $I \vdash F$ is derivable. Thus we can derive $h$ from $h_0$, by repeatedly applying functoriality and cut.
- Otherwise, $h$ originated from an assumption on $s$ in $\Pi_u^s$, which has also been affected by the substitution of $F$ by $I$ during the construction of $\Theta$. By construction of the infinite branch $\gamma$, and because $s$ is the sequent associated to the thread that strongly validates $\gamma$, we have that the occurrence of $F$ in the assumption $s = (\Gamma, F \vdash \Delta)$ in $\Pi_u^s$ must be traced back to $B[F/X]$ in the conclusion of that derivation. Thus, that toplevel occurrence of $F$ in $s$ has been impacted by the substitution in the construction of $\Theta$, and it becomes $I$ in $h$. More precisely, $h$ is of the form $(\Gamma', I \vdash \Delta')$ where $\Gamma'$ (resp. $\Delta'$) is obtained from $\Gamma$ (resp. $\Delta$) by substituting some positive (resp. negative) instances of $F$ by $I$. We can conclude by completely eliminating that assumption, deriving instead $h$ from $\Gamma, I \vdash \Delta$, which itself can be derived from $\Gamma, \mathrm{env}(s) \vdash \Delta$, which is easily derivable as observed before.

We have thus obtained a derivation $\beta$ of $\Gamma, B[I/X] \vdash \Delta$ such that $\mathcal{A}(\beta) \subseteq \mathcal{A}(\Pi)$. We are now ready to conclude by constructing a $\mu\mathsf{LK}$ derivation $\pi$ of $s = (\Gamma, F \vdash \Delta)$ such that $\mathcal{A}(\pi) \subseteq \mathcal{A}(\Pi)$. The derivation starts with a $(\mu_{\mathsf{l}})$ rule on $F$, using $I$ as invariant. As the second subderivation of that inference, we derive $\Gamma, I \vdash \Delta$ from $\Gamma, \mathsf{env}(s) \vdash \Delta$, which itself is easily derived. For the first subderivation, we proceed as follows:

$$\dfrac{\dfrac{\dfrac{}{B[I/X] \vdash B[I/X]}\ (\mathsf{Ax}) \qquad \dfrac{\dfrac{\beta}{\vdots}}{B[I/X] \vdash \mathrm{env}(s)}}{B[I/X] \vdash B[I/X] \wedge \mathrm{env}(s)}\ (\wedge_{\mathsf{r}})}{B[I/X] \vdash I}\ (\mu_{\mathsf{r}})$$

□

Note that this development makes crucial use of the difference between $\mu\mathsf{LK}^\omega$ and $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$: our translatability criterion and translation procedure rely in a fundamental way on the precise thread structure of $\mu\mathsf{LK}^\omega$ proofs.

### 5.2 Weakly deterministic parity inclusions and Büchi inclusions in $\mu\mathsf{LK}$

We show that inclusion proofs are translatable for a particular class of disjunctive formulas called *weakly deterministic*, and use this as a stepping stone for constructing $\mu\mathsf{LK}$ proofs for all Büchi inclusions.

**Definition 33.** *We say that a disjunctive formula $F$ is* **deterministic** *iff for any $G \in Q(F)$ and $a \in \Sigma$, $a^{-1}G$ contains at most one formula. The* **level of non-determinism** $\mathsf{lvl}(F)$ *of $F$ is the number of its even states that are not deterministic:*

$$\mathsf{lvl}(F) = \#\{\ G \in Q(F)\ \mid\ c(G) \text{ is even},\ G \text{ is not deterministic}\ \}.$$

*Finally, $F$ is said to be* **weakly deterministic** *iff* $\mathsf{lvl}(F) = 0$.

**Proposition 34.** *Let $F$ be a disjunctive formula and $G$ a weakly deterministic disjunctive formula. If $\mathcal{L}(F) \subseteq \mathcal{L}(G)$ then $\Pi(F, G)$ is translatable.*

*Proof.* Since $G$ is weakly deterministic, it has the form $G = N[D_i/X_i]_{i=1\dots n}$ where $D_k$ are deterministic formulas and $N$ is a formula that does not contain the $\nu$ connective. Let us denote the set of states of $D_k$ by $Q_k$ and those of $N$ by $Q_N$.

The construction of $\Pi(F, G)$ depends on the choice of automaton associated to $G$. We choose it here with a priority assignment $c$ for the states of the formula $G$ such that $c(K) = 1$ for all $K \in Q_N$, and $c(K) > 1$ for all $K \in Q_i$, $1 \leq i \leq n$. Such an assignment exists since $Q_N$ contains only $\mu$-formulas and, for all $1 \leq i \leq n$, the states in $Q_N$ and $Q_i$ are not coaccessible.

Let us first make some observation about Safra's construction over weakly-deterministic formulas, with the chosen priority assignment. Consider a run $(T_i)_{i\in\omega}$ on some word $(a_i)_{i\in\omega}$, as in the proof of Proposition 28.

**Assertion:** Let $i \in \omega$ and $v$ be a child of the root $r$ in $T_i$. One has:

- the edge $(r, v)$ is labelled by 1;
- either $e_{T_i}(v) = \{K\}$ where $K \in Q_N$ and in this case $v$ is a leaf (call it child of Type 1); or $e_{T_i}(v) = \{K_1, \dots, K_p\}$ such that for all $k \in [1, p]$ there exists $j_k \in [1, n]$ such that $K_k \in Q_{j_k}$, and $k \neq k'$ implies $j_k \neq j_{k'}$ (call it child of Type 2);
- if $v$ is also a node in $T_{i+1}$, then $v$ is of Type 2 in $T_{i+1}$.

We prove this assertion by induction on $i$. When $i = 0$ it is trivial since the root has no child. Suppose that the assertion is true for some $i$ and let us prove it for $i + 1$. Let $v$ be a child of the root in $T_{i+1}$. The edge $(r, v)$ is labelled by 1: indeed the outgoing edges from the root can only be 1 or 0. But by definition of the priority assignment we gave above, no state has priority 0. On the other hand, the only way to create an edge of label 0 is by Step 3 when a formula has a priority 0. Hence the outgoing edges from $r$ are labelled by 1. To prove the second item, we separate two cases:

- If $v$ does not exist in $T_i$ this means that $w$ has been created in Step 3, hence it is a leaf labelled by a singleton $\{K\}$. We have that $K \in Q_N$ since only formulas from $Q_N$ have priority 1.
- Otherwise, $v$ is a child of $r$ in $T_i$. By induction hypothesis, the node $v$ is of Type 2 in $T_{i+1}$.

Suppose that $v$ is also a child of $r$ in $T_{i+2}$. There are two cases:

- If $v$ is of Type 1 in $T_{i+1}$, let $e_{T_{i+1}}(v) = \{K\}$ and $\delta(K, a_{i+1}) = \{\vec{M}, \vec{K}\}$, $\vec{M} \subseteq Q_N$ and $\vec{K} \subseteq \cup_{1 \leq i \leq n} Q_i$. The tree structure of $G$ implies that $\vec{K}$ is of the form $\vec{K} := \{K_1, \ldots, K_n\}$ with at most one $K_k$ per $Q_j$, and none in $Q_N$. After Step 3, all formulas of $\vec{M}$ will be in new child of the root in $T_{i+2}$. Hence, $v$ will be of Type 2 in $T_{i+2}$.
- If $v$ is of Type 2 in $T_{i+1}$, it is also of Type 2 in $T_{i+2}$ since $D_k$ are deterministic.

Having proved our assertion, we now establish the result. Let $v$ be a node and $j$ an index such that for all $i \geq j$, $v$ is a node in $T_i$ and $v$ is a leaf infinitely often. As the outgoing edges of the root all have label 1, $v$ cannot be the root. Indeed, to become a leaf infinitley often, a node has to collapse in Step 6, and this is possible only when the outgoing edges are of even priority. Hence $v$ appears in the subtree of a child $w$ of the root in $T_i$, for all $i > j$. Since $v$ is persistent, $w$ is also persistent, and by the above assertion, the node $w$ is of Type 2 in $T_i$, for all $i > j$. Hence $v$ is also of Type 2, that is, $e_{T_i}(v)$ has at most one formula from each $Q_k$ and no formulas from $Q_N$.

We finally come back to the proof of Proposition 28, and show that each time we exhibited a valid thread $t$ in that proof, $t$ was actually strongly valid. This follows from the following remarks:

(i) When the sequents of a branch contain only one formula in their left-hand side, any valid thread visiting only left-hand side formulas is strongly valid. Since all the sequents in $\Pi(F, G)$ have only one formula in their left-hand side, the left-hand side thread exhibited in the first case of the proof of Proposition 28 is actually strongly valid.

(ii) Consider the right-hand side thread $t$ that we extracted from the persistent node $v$ in the second case of the proof of Proposition 28. Let $F_r := \mathsf{min}(t)$, there is some $k$ such that $F_r \in Q_k$. Let $s$ be a sequent appearing infinitely often in $\gamma$ such that *(i)* the thread meets $s$ at the level of the formula $F_r$ *(ii)* $s$ is the conclusion of the unfolding of $F_r$. We prove that $s$ satisfies the property in the definition of strong validity. Suppose that there is some $i$ such that $s_i = s$ and $F_i \neq F_r$. Since $F_i$ is in the thread, we have that $F_i \in Q_k$. But $s_i = s$ hence $e_{T_i}(v)$ contains the formula $F_r$ also, which means that $e_{T_i}(v)$ contains two distinct elements from $Q_k$ contradicting the last observation above. □

Note that the previous argument easily yields a slightly more general result:

**Proposition 35.** *Let $F$ be a disjunctive formula and $(G_j)_{1 \leq j \leq n}$ be weakly deterministic ones. If $\mathcal{L}(F) \subseteq \bigcup_j \mathcal{L}(G_j)$ then there is a translatable proof of $F \vdash G_1, \ldots G_n$.*

**Theorem 36.** *Let $F$ and $G$ be two Büchi formulas. We have $\mathcal{L}(F) \subseteq \mathcal{L}(G)$ iff there is a $\mu\mathsf{LK}$ proof of $F \vdash G$.*

*Proof.* Let us consider the following more general result:

> *Let $F$ be a disjunctive formula and $(G_j)_{1 \leq j \leq n}$ be disjunctive formulas of the form $B_j[P_j^i/X^i]_{1 \leq i \leq k}$ where $B_j$ is a Büchi formula and the $P_j^i$ are deterministic, disjunctive $\nu$-formulas. If $\mathcal{L}(F) \subseteq \cup_j \mathcal{L}(G_j)$ then there is a $\mu\mathsf{LK}$ proof of $F \vdash G_1, \ldots, G_n$.*

We prove the latter by induction on $\mathsf{max}\{\mathsf{lvl}(G_j)\}_j$.

**Base case:** When $\mathsf{lvl}(G_j) = 0$ for all $j$, the formulas $G_j$ are obviously weakly deterministic. We thus apply Proposition 35 and Proposition 30 to get a $\mu\mathsf{LK}$ proof of $F \vdash G_1, \ldots, G_n$.

**Inductive case:** For the sake of readability we assume $n = 1$, *i.e.*, we consider only one $G = G_1 = B[P_1/X_1, \ldots, P_k/X_k]$. Notice first that since $B$ is a Büchi formula, it can be written as $B = B_\mu[Q_1/Y_1, \ldots, Q_l/Y_l]$ where $B_\mu$ does not contain any $\nu$-formulas and, for all $1 \leq i \leq l$, $Q_i = \nu Z_i.B_i$, with $\mathsf{fv}(Q_i) \subseteq \{X_1, \ldots, X_k\}$ and variables $(Y_i)_i$ and $(X_j)_j$ being pairwise disjoint. We have that $G = B_\mu[R_i/Y_i]_i$ where $R_i = Q_i[P_j/X_j]_j$.

Since $R_i$ is a closed disjunctive formula, there exists a parity automaton $\mathcal{A}_i$ associated to it. Let $\mathcal{A}'_i$ be any deterministic parity automaton accepting the same language as $\mathcal{A}_i$, and let $R'_i$ be the disjunctive formula encoding $\mathcal{A}'_i$. We thus have $\mathcal{L}(R_i) = \mathcal{L}(R'_i)$. Further, $\mathcal{L}(G) = \mathcal{L}(B_\mu[R'_i/Y_i]_i)$ since any accepting run of $G$ starts by a finite run in $B_\mu$ and continues with an accepting run of some $R_i$.

Consider the following derivation:

$$\dfrac{\dfrac{\pi_0}{F \vdash B_\mu[R'_i/Y_i]_{1 \leq i \leq l}} \qquad \dfrac{\dfrac{\pi_1}{R'_1 \vdash R_1} \quad \ldots \quad \dfrac{\pi_l}{R'_l \vdash R_l}}{B_\mu[R'_i/Y_i]_{1 \leq i \leq l} \vdash B_\mu[R_i/Y_i]_{1 \leq i \leq l}}\ (\mathsf{Functo})}{F \vdash G}\ (\mathsf{Cut})$$

The sub-proof $\pi_0$ can be obtained by applying the base case since $\mathcal{L}(F) \subseteq \mathcal{L}(B_\mu[R'_i/Y_i]_i)$, $\mathsf{lvl}(B_\mu[R'_i/Y_i]_i) = 0$, and the formula on the right is of the expected form.

For each $1 \leq i \leq l$, the sub-proof $\pi_i$ is obtained by applying the coinduction rule $(\nu_\mathsf{r})$ with coinvariant $R'_i$ for $R_i$. The non-trivial premise of that rule is then $R'_i \vdash B_i[P_j/X_j]_j[R'_i/Z_i]$. We derive it using rule $(\rightarrow)$, slightly modified because the formula on the right is not a fixed point formula. We are left to construct, for each $R'_i \xrightarrow{a} R'_{i,a}$, a derivation $\pi_{i,a}$ of $R'_{i,a} \vdash \Delta_{i,a}$ where $\Delta_{i,a}$ is (with a slight abuse of notation) $a^{-1}(B_i[P_j/X_j]_j[R'_i/Z_i])$.

Let us show that each $\pi_{i,a}$ can be obtained by induction hypothesis. We obviously have $\mathcal{L}(R'_{i,a}) \subseteq \mathcal{L}(\Delta_{i,a})$. Moreover, any formula $H \in \Delta_{i,a}$ is of the expected form. It suffices to establish that it is the case for $B_i[P^j/X^j]_j[R'_i/Z_i]$, which follows from the fact that $B[P^j/X^j]_j = B_\mu[(\nu Z_i.B_i[P_j/X_j]_j)/Y_i]_i$: indeed, if a $\nu$-formula contains a variable of a greater (outermost) $\mu$ formula in $B_i$ then this is also the case in $B$ contradicting the hypothesis that $B$ is a Büchi formula. Finally, $\mathsf{lvl}(B_i[P_j/X_j]_j[R'_i/Z_i]) = \mathsf{lvl}(R_i) - 1$ and thus $\mathsf{lvl}(H) < \mathsf{lvl}(G)$ for all $H \in \Delta_{i,a}$. □

Underlying this final result, we have successfully given an algorithm that checks for Büchi inclusions and yields proofs in case of success. At the heart of the procedure, given two automata encoded as $F$ and $G$, we first build a circular proof by means of Safra's construction. If we were interested in a $\mu\mathsf{LK}^\omega$ certificate, we should check for the proof's validity at this point, which can be done using Ramsey-based techniques. But if the target is $\mu\mathsf{LK}$, there is no need to do so: we can simply run the translation algorithm underlying the proof of Proposition 30 specialized for Safra proofs. In the case of a strongly connected proof the algorithm searches, along some branch that visits all sequents, for a persistent node that becomes a leaf infinitely often in Safra trees. If this is found, the translation can continue. Otherwise, it actually means that the branch is invalid, and it immediately yields a word accepted by $F$ but not by $G$. Thus our proof-producing verification algorithm can also produce counterexamples.

## 6. Conclusion

***Contributions.*** We have given a new completeness argument in the linear time $\mu$-calculus, for sequents corresponding directly to

inclusions of Büchi automata. We have done so in $\mu\mathsf{LK}$, a cousin of the well understood proof system $\mu\mathsf{MALL}$. This proof system has a complete, well understood proof theory. It notably enjoys cut elimination, and its proofs are checkable in polynomial time. Although our result does not imply completeness for the full calculus[2], the result is non-trivial, dealing with one of the main problems in such proofs, namely the non-determinism induced by disjunctions appearing on cycles over $\nu$ formulas — this is the exact problem avoided by Kozen with aconjunctivity when translating *refutations* rather than proofs. On the way to this final result we have also obtained an intermediate result for inclusions of sufficiently deterministic parity automata; here, non-determinism is avoided to some extent, but fixed point alternations are arbitrary.

Unlike the full proof of completeness for linear time $\mu$-calculus (Kaivola 1995), our argument is constructive and gives a central role to circular proofs. As by-products of this result, we have defined $\mu\mathsf{LK}^\omega$, a new circular proof system for linear $\mu$-calculus, and given a strong translation result from $\mu\mathsf{LK}^\omega$ to $\mu\mathsf{LK}$. This result does not rely on syntactic conditions on formulas, but on a geometric condition on circular proofs. By its nature, the result is quite generic; it obviously carries to similar circular proof for other $\mu$-calculi, *e.g.*, $\mu\mathsf{MALL}$. The translation exploits a key difference between $\mu\mathsf{LK}^\omega$ and previous circular deductive systems for linear time $\mu$-calculus : in $\mu\mathsf{LK}^\omega$, we distinguish *occurrences* of formulas, which induces a very structured notion of thread. Because of this structure, completeness of $\mu\mathsf{LK}^\omega$ itself is not obvious. In fact, we had to tackle non-determinism already when building circular proofs for automata inclusions. We did so by exploiting determinization techniques for $\omega$-automata to guide a complete proof search strategy, thus giving a logical meaning to Safra's construction.

***Future work.*** We are considering several directions for further investigation. An obvious one is to extend our argument to obtain completeness for inclusions of parity automata, and then for the full calculus. It would also be interesting to investigate the efficiency of the algorithm underlying our argument. There might be room for improvements, for instance by attempting to build more compact circular proofs by improving the proof search technique used in Theorem 22. Here, other determinization techniques than the Safra trees used in this paper (*e.g.*, Piterman trees (Piterman 2007)) could be leveraged, if they can be shown to yield translatable circular proofs. We are also interested in giving a proper proof-theoretical status to $\mu\mathsf{LK}^\omega$. We believe that the precise structure of this proof system makes it a good candidate to investigate important properties such as cut elimination and focalisation; cut elimination for circular proof systems is an open question beyond the purely additive calculus of (Fortier and Santocanale 2013). This would make it possible to propose circular proofs as proof certificates themselves; although non-trivial to check, such proofs can be simpler to produce and smaller, as seen in this study.

## References


D. Baelde. Least and greatest fixed points in linear logic. *ACM Transactions on Computational Logic (TOCL)*, 13(1):2, 2012.

J. C. Bradfield, J. Esparza, and A. Mader. An effective tableau system for the linear time $\mu$-calculus. In F. M. auf der Heide and B. Monien, editors, *ICALP96, Paderborn, Germany, 8-12 July 1996*, volume 1099 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 1996. ISBN 3-540-61440-0. doi: 10.1007/3-540-61440-0_120.

J. Brotherston and N. Gorogiannis. Cyclic abduction of inductively defined safety and termination preconditions. In M. Müller-Olm and H. Seidl, editors, *SAS 2014, Germany, September 2014. Proceedings*, volume 8723 of *Lecture Notes in Computer Science*, pages 68–84. Springer, 2014. ISBN 978-3-319-10935-0.

J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. 21(6):1177–1216, Dec. 2011.

Z. Chihani, D. Miller, and F. Renaud. Foundational proof certificates in first-order logic. In *Automated Deduction–CADE-24*, pages 162–177. Springer, 2013.

C. Dax, M. Hofmann, and M. Lange. A proof system for the linear time $\mu$-calculus. In S. Arun-Kumar and N. Garg, editors, *FSTTCS 2006, Kolkata, India, December 13-15, 2006*, volume 4337 of *Lecture Notes in Computer Science*, pages 273–284. Springer, 2006. ISBN 3-540-49994-6. doi: 10.1007/11944836_26.

J. Fortier and L. Santocanale. Cuts for circular proofs: semantics and cut-elimination. In S. R. D. Rocca, editor, *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPIcs*, pages 248–262. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. ISBN 978-3-939897-60-6.

E. Grädel, W. Thomas, and T. Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag New York, Inc., New York, NY, USA, 2002. ISBN 3-540-00388-6.

D. Janin and I. Walukiewicz. Automata for the modal mu-calculus and related results. In J. Wiedermann and P. Hájek, editors, *MFCS'95, Prague, Czech Republic, September 1995*, volume 969 of *Lecture Notes in Computer Science*, pages 552–562. Springer, 1995. ISBN 3-540-60246-1. doi: 10.1007/3-540-60246-1_160.

D. Kähler and T. Wilke. Complementation, disambiguation, and determinization of büchi automata unified. In *Automata, Languages and Programming*, pages 724–735. Springer, 2008.

R. Kaivola. Axiomatising linear time mu-calculus. In I. Lee and S. A. Smolka, editors, *CONCUR '95, Philadelphia, PA, USA, August 21-24, 1995, Proceedings*, volume 962 of *Lecture Notes in Computer Science*, pages 423–437. Springer, 1995. ISBN 3-540-60218-6. doi: 10.1007/3-540-60218-6_32.

D. Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983. doi: 10.1016/0304-3975(82)90125-6.

D. Miller. A proposal for broad spectrum proof certificates. In *Certified Programs and Proofs*, pages 54–69. Springer, 2011.

N. Piterman. From nondeterministic büchi and streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3(3), 2007. doi: 10.2168/LMCS-3(3:5)2007.

S. Safra. Exponential determinization for $\omega$-automata with strong-fairness acceptance condition (extended abstract). In *STOC'92*, pages 275–282, New York, NY, USA, 1992. ACM. ISBN 0-89791-511-9. doi: 10.1145/129712.129739.

L. Santocanale. A calculus of circular proofs and its categorical semantics. Technical Report RS-01-15, BRICS, Department of Computer Science, University of Aarhus, May 2001.

L. Santocanale. A calculus of circular proofs and its categorical semantics. In M. Nielsen and U. Engberg, editors, *Foundations of Software Science and Computation Structures*, volume 2303 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 2002. ISBN 3-540-43366-X.

C. Stirling and D. Walker. Local model checking in the modal mu-calculus. *Theor. Comput. Sci.*, 89(1):161–177, 1991. doi: 10.1016/0304-3975(90)90110-4.

R. S. Streett and E. A. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inf. Comput.*, 81(3):249–264, 1989. doi: 10.1016/0890-5401(89)90031-X.

I. Walukiewicz. On completeness of the mu-calculus. In *(LICS '93), Montreal, Canada, June 19-23, 1993*, pages 136–146, 1993. doi: 10.1109/LICS.1993.287593.

I. Walukiewicz. Completeness of kozen's axiomatisation of the propositional mu-calculus. In *LICS'95, USA, June 26-29, 1995*, pages 14–24. IEEE Computer Society, 1995. ISBN 0-8186-7050-9.

I. Walukiewicz. Completeness of kozen's axiomatisation of the propositional mu-calculus. *Inf. Comput.*, 157(1-2):142–182, 2000. doi: 10.1006/inco.1999.2836.


[2] Any linear time $\mu$-calculus formula is equivalent to a Büchi formula, but it is far from obvious to prove that equivalence in, say, $\mu\mathsf{LK}$.

$$\dfrac{\dfrac{}{a,\ldots\vdash a}\,(\mathsf{Ax}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b,\bigcirc\Phi\vdash b}\,(\mathsf{Ax}) \qquad \dfrac{\vdots}{\Phi\vdash\Psi}}{b,\bigcirc\Phi\vdash b\wedge\bigcirc\Psi}\,(\mathsf{W_l}),(\bigcirc)}{b,\bigcirc\Phi\vdash (a\wedge\bigcirc\Psi')\vee(b\wedge\bigcirc\Psi)}\,(\wedge_\mathsf{r})}{b\wedge\bigcirc\Phi\vdash\Psi'}\,(\vee_\mathsf{r}),(\mathsf{W_r})}{\bigcirc(b\wedge\bigcirc\Phi)\vdash\bigcirc\Psi'}\,(\wedge_\mathsf{l}),(\sigma_\mathsf{r})}{a,\bigcirc(b\wedge\bigcirc\Phi)\vdash\bigcirc\Psi'}\,(\bigcirc)}{a,\bigcirc(b\wedge\bigcirc\Phi)\vdash a\wedge\bigcirc\Psi'}\,(\mathsf{W_l})}{a\wedge\bigcirc(b\wedge\bigcirc\Phi)\vdash(a\wedge\bigcirc\Psi')\vee(b\wedge\bigcirc\Psi)}\,(\wedge_\mathsf{l}),(\vee_\mathsf{r}),(\mathsf{W_r})}{\Phi\vdash\Psi}\,(\sigma_\mathsf{l}),(\sigma_\mathsf{r})$$

Figure 4: Regular proof of $\Phi \vdash \Psi$ in $\mu\mathsf{LK}^\infty$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b,\bigcirc\Phi\vdash b}\,(\mathsf{Ax}) \qquad \dfrac{\vdots}{\Phi\vdash\Phi}\,(\mathsf{Ax})}{b,\bigcirc\Phi\vdash b\wedge\bigcirc\Phi}\,(\mathsf{W_l}),(\bigcirc)}{b,\bigcirc\Phi\vdash (a\wedge\bigcirc\Psi'')\vee(b\wedge\bigcirc\Psi)}\,(\wedge_\mathsf{r})}{b\wedge\bigcirc\Phi\vdash\Psi''}\,(\vee_\mathsf{r}),(\mathsf{W_r})}{\bigcirc(b\wedge\bigcirc\Phi)\vdash\bigcirc\Psi''}\,(\wedge_\mathsf{l}),(\mu_\mathsf{r})}{a,\bigcirc(b\wedge\bigcirc\Phi)\vdash\bigcirc\Psi''}\,(\bigcirc)}{a,\bigcirc(b\wedge\bigcirc\Phi)\vdash a\wedge\bigcirc\Psi''}\,(\mathsf{W_l})}{a\wedge\bigcirc(b\wedge\bigcirc\Phi)\vdash(a\wedge\bigcirc\Psi'')\vee(b\wedge\bigcirc\Phi)}\,(\wedge_\mathsf{l}),(\vee_\mathsf{r}),(\mathsf{W_r})}{\Phi\vdash\Psi''}\,(\nu_\mathsf{l}),(\mu_\mathsf{r})$$

Figure 5: Finite proof of $\Phi \vdash \Psi''$ in $\mu\mathsf{LK}$

## A. Example proofs in $\mu\mathsf{LK}^\omega$ and $\mu\mathsf{LK}$

We consider proving a simple implication which, informally, expresses that $(ab)^\omega$ is included in $(a^*b)^\omega$. More precisely, we will prove $\Phi \vdash \Psi$ where:

$$\begin{aligned}\Phi &:= \nu X.\, a \wedge \bigcirc(b \wedge \bigcirc X)\\ \Psi &:= \nu X.\mu Y.\, (a \wedge \bigcirc Y) \vee (b \wedge \bigcirc X)\end{aligned}$$

We show in Figure 4 a $\mu\mathsf{LK}^\omega$ pre-proof of $\Phi \vdash \Psi$, where the vertical dots denote a repetition of the tree, and $\Psi'$ stands for $\mu Y.\, (a\wedge\bigcirc Y)\vee(b\wedge\bigcirc\Psi)$ — this formula is obtained by unfolding the two fixed point expressions of $\Psi$ and accessing the first subformula under $\bigcirc$. The threads in this pre-proof are unambiguous, as there is at most one occurrence of a formula in a given sequent. It is easy to see that the pre-proof is valid, and is thus a proof in $\mu\mathsf{LK}^\omega$: indeed, it has a single infinite branch, which admits a single thread on the right-hand side of sequents, and that thread unfolds $\Psi$ infinitely often, and is thus valid.

We prove the same sequent in $\mu\mathsf{LK}$. Instead of cycling in a way that unfolds $\Psi$ infinitely often on the right-hand side, we show that $\Phi$ is a coinvariant of $\Psi$. Specifically, we apply rule $\nu_r$ of Figure 3 with $S := \Phi$. The first premise is an instance of the axiom. The second one is $\Phi \vdash \Psi''$ where $\Psi''$ stands for $\mu Y.\, (a \wedge \bigcirc Y) \vee (b \wedge \bigcirc\Phi)$; its derivation is shown in Figure 5.

Our two derivations are obviously related. The most part of the $\mu\mathsf{LK}$ derivation has the same structure as the $\mu\mathsf{LK}^\omega$ derivation, with $\Phi$ replacing $\Psi$ (and, consequently, $\Psi''$ replacing $\Psi'$) so that the cycle can be replaced by an axiom. In fact, the $\mu\mathsf{LK}$ derivation would be obtained (modulo minor adjustments) from the translation result of Section 5. This simple example illustrates the obvious fact that cycles can be translated by means of (co)induction, but not the difficulty of doing so in a way that terminates (in the case of nested cycles, translating one may expand the other, so that a naive translation process would not terminate).

## B. Comparison with $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$: an example

We give here a proof of inclusion in $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ that cannot be directly translated to $\mu\mathsf{LK}^\omega$, for the reasons explained in Section 2. Consider the automata $\mathcal{B}_1$ and $\mathcal{B}_2$ depicted in Figures 6 and 7. The first one has an empty language, but this is irrelevant to our example, which illustrates a problem on the traces induced by $\mathcal{B}_2$.

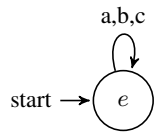



Figure 6: Automaton $\mathcal{B}_1$

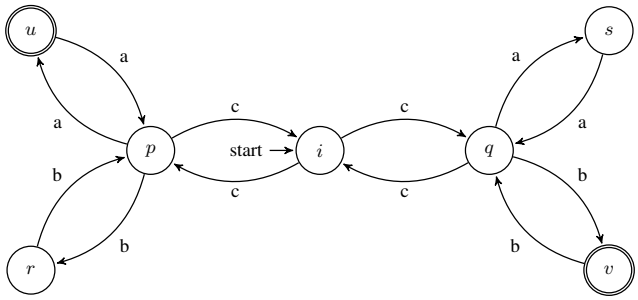



Figure 7: Automaton $\mathcal{B}_2$

The sequent $e \vdash i$ is provable in $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$. The proof has the following shape where $(\dagger)$ and $(\star)$ rules denote "back edges" closing cycles:

$$\dfrac{\dfrac{\dfrac{\vdots\ \dfrac{}{e\vdash p,q}\,(\star)}{e\vdash u,s} \qquad \dfrac{\vdots\ \dfrac{}{e\vdash p,q}\,(\star)}{e\vdash r,v} \qquad \dfrac{}{e\vdash i}\,(\dagger)}{e\vdash p,q \quad (\star)}}{e\vdash i \quad (\dagger)}$$

In the sequent $e \vdash i$ on the top right, the formula $i$ comes from both $p$ and $q$; there are two threads leading to it. In $\mu\mathsf{LK}^\omega$, this is not allowed. To keep a sequent $e \vdash i$ and close the cycle in the same way, we would need to choose one of the two threads. However that is not possible, since any choice would break the validity condition. The Safra construction used in the proof of Theorem 22 proceeds differently. It corresponds to choosing one of the two threads, but the resulting Safra tree will be different from the one used at the root. Therefore the cycle is not closed there, but later in the proof construction, which does result in a valid proof.

## C. Example of Safra construction

We develop in this section an example of Safra's construction illustrating the different steps of the construction and Proposition 25. Let us consider the parity automaton $\mathcal{A} = (\{q_1, q_2\}, q_1, \delta, c)$ depicted in Figure 8 where priorities are such that $c(q_i) = i$. We thus consider the following set of priorities $P = \{0, 1, 2, 3\}$.

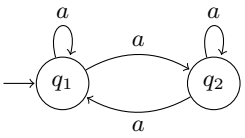



Figure 8: Automaton $\mathcal{A}$

We now consider a run of Safra trees starting with $T_1$ (*i.e.*, a single node of name $n_0$, labelled by $\{q_1\}$) on the word $w = a^\omega$. This run is given in Figure 9, *i.e.*, $T_1.(T_2.T_3)^\omega$. We denote node names by using brackets (*e.g.*, $[n_0]$) and we indicate the leaves' labels to the right of their names (*e.g.*, $[n_0]\ \{q_1\}$). We simply write $T \rightsquigarrow T'$ when $T' = \Delta(T, a)$. We also detail the construction of $T_3 = \Delta(T_2, a)$ in Figure 10, where $\overset{i}{\rightsquigarrow}$ denotes the application of Step $(i)$ of Safra's construction as defined in Section 4.

In the stationary part of the run, *i.e.*, $(T_2.T_3)^\omega$, the node of name $[n_1]$ is persistent (*i.e.*, it is never removed from the trees) while the ones named $[n_2]$ and $[n_3]$ are not. This reflects the fact that all (non-accepting) runs visiting both states $q_1$ and $q_2$ infinitely often are removed by our Safra's construction. Indeed, Safra trees remove some runs when performing Step $(4)$ of the construction. Concretely, the non-accepting run $(q_1.q_2)^\omega$ is discarded in the Safra run $(T_i)_{i\in\omega}$ when the node $[n_8]\ \{q_1\}$ is removed by Step $(4)$ in Figure 10 — we assume here that the state $q_1$ in node $n_3$, which we decided to keep, comes from node $n_5$. In this simple example, the only run that is never removed is $\rho = q_1.q_2^\omega$; it is associated to the persistent node $[n_1]$.

More generally, a persistent node $v$ whose incoming edge is labelled $p$ intuitively selects all runs for which the minimum priority $p'$ visited infinitely often is even and equal to either $p$ or $p + 1$, depending on the priority of $p$. If at some point, a run visits a state of priority $p_1 < p'$ then the node would discard it when applying Step $(3)$. Moreover, it may be the case that this run will not be selected by another node and will be removed at Step $(4)$. This is happening in our example (cf. Figure 10) for nodes $n_1$ and $n_2$ and states $q_1$ (in $n_6$ and $n_5$). Further, if this node becomes a leaf infinitely often, it means that Step $(6)$ is applied infinitely often and thus all runs it has selected have visited infinitely many times a state of priority $p'$. This is happening in our example for the node $n_1$ when performing Step $(6)$. Altogether, we can obtain that the node only selects runs that are accepting for the priority $p'$.

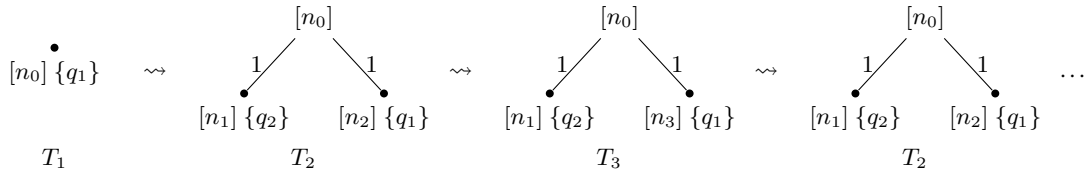



Figure 9: The run of Safra trees associated to $\mathcal{A}$ on $w = a^\omega$

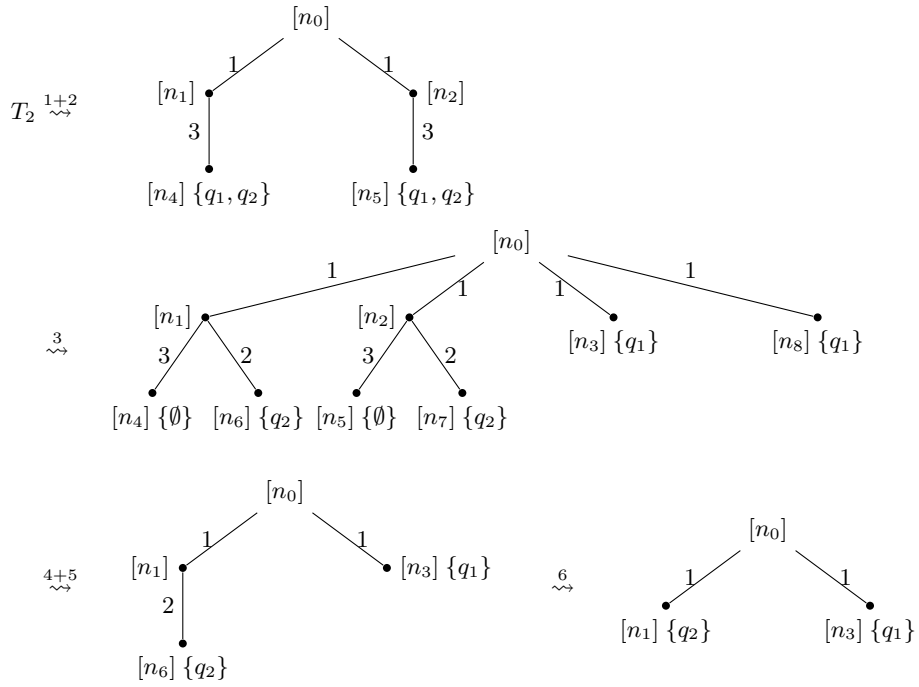



Figure 10: Internal steps of $\Delta(T_2, a) = T_3$