Constructive completeness for the linear-time μ-calculus

Amina Doumane

Constructive completeness for the linear-time μ-calculus

Amina Doumane
IRIF, Université Paris Diderot & CNRS
doumane@irif.fr

***Abstract*—Modal μ-calculus is one of the central logics for verification. In his seminal paper, Kozen proposed an axiomatization for this logic, which was proved to be complete, 13 years later, by Kaivola for the linear-time case and by Walukiewicz for the branching-time one. These proofs are based on complex, non-constructive arguments, yielding no reasonable algorithm to construct proofs for valid formulas. The problematic of constructiveness becomes central when we consider proofs as certificates, supporting the answers of verification tools. In our paper, we provide a new completeness argument for the linear-time μ-calculus which is constructive, i.e. it builds a proof for every valid formula. To achieve this, we decompose this difficult problem into several easier ones, taking advantage of the correspondence between the μ-calculus and automata theory. More precisely, we lift the well-known automata transformations (non-determinization for instance) to the logical level. To solve each of these smaller problems, we perform first a proof-search in a circular proof system, then we transform the obtained circular proofs into proofs of Kozen's axiomatization.**

I. Introduction

The linear-time μ-calculus [1] is a temporal logic that extends Pnueli's *Linear Temporal Logic* (LTL) [2] with least and greatest fixed points. This increases considerably its expressive power while keeping the decidability properties of LTL, which makes it a very suitable logic for verification. The linear-time μ-calculus has infinite words as models, thus it can be used to express trace properties of reactive systems. There exist, among others, two approaches to verification using temporal logics [3]. The first one, called "model theoretic", describes both the system S and the property P to check as formulas φ_S and φ_P; then verifying whether S satisfies P is reduced to checking the validity of the formula $\varphi_S \to \varphi_P$. The other approach, called "proof theoretic" reduces the verification problem to the provability of the formula $\varphi_S \to \varphi_P$. The advantage of the second approach is that it gives, besides the boolean answer to the verification problem, a *certificate* that supports the decision of the verification tool, which is the proof of the formula $\varphi_S \to \varphi_P$. To make this approach work with the linear-time μ-calculus, two conditions should be satisfied: the first is the existence of a sound and complete deductive system for the linear-time μ-calculus; the second is the existence of algorithms that produce proofs for valid formulas. The first condition is satisfied, since the linear-time μ-calculus enjoys a deductive system, that we call $\mu\mathsf{LK}$, which is the restriction of Kozen's axiomatization for the modal μ-calculus to the linear time. This system was proved to be sound and complete by Kaivola ([4]). But the second condition is not really met, since the only existing algorithm is the naive one, that enumerates all $\mu\mathsf{LK}$ proofs.

A proof of completeness is a mathematical argument showing that every valid formula is provable, but it is not always possible to extract from such argument an algorithm that produces proofs for valid formulas. Indeed, completeness proofs may involve complex, non-constructive arguments yielding no method for actually constructing a proof. On the contrary, a constructive proof of completeness, specifying a proof search method, readily provides a "realistic" algorithm.

None of the existing proofs of completeness for the μ-calculus *w.r.t.* Kozen's axiomatization ([4], [5]) is constructive in this sense. All the attempts to get constructive proofs were either partial (Kozen provided a constructive completeness proof for the fragment of *aconjunctive* formulas [6]); or fall out of Kozen's axiomatization. Indeed, in his first completeness result [7], Walukiewicz had to modify Kozen's axiomatization to get a constructive proof for an ad-hoc proof system.

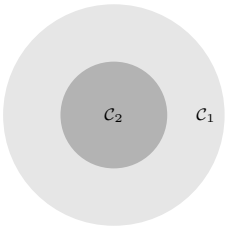
In this paper, we provide a constructive proof for the full linear-time μ-calculus *w.r.t.* $\mu\mathsf{LK}$. To do so, we go back to the earlier proofs of completeness, and try to understand where constructiveness is lost, to better solve this problem.

Earlier proofs of completeness for the μ-calculus rely schematically on the following idea. Find a subset $\mathcal{C}_2$ of the set of μ-calculus formulas $\mathcal{C}_1$ such that:

1) For every valid formula φ_1 in $\mathcal{C}_1$, there is a valid formula φ_2 in $\mathcal{C}_2$ such that $\varphi_2 \vdash \varphi_1$ is provable.
2) Every valid formula of $\mathcal{C}_2$ is provable. This is the completeness result restricted to $\mathcal{C}_2$.

Completeness is proved by combining 1) and 2) via a cut rule:

$$\dfrac{\dfrac{2)}{\vdash \varphi_2} \quad \dfrac{1)}{\varphi_2 \vdash \varphi_1}}{\vdash \varphi_1}\ (\mathsf{Cut})$$



The complexity of problems 1) and 2) depends on the class $\mathcal{C}_2$: the larger it is, the more difficult problem 2) becomes, since it gets close to the original completeness problem. On the contrary, when $\mathcal{C}_2$ gets smaller, the problem 1) becomes difficult. Kaivola's proofs uses the class of *banan form formulas*; and Walukiewicz' one uses the class of *disjunctive formulas*

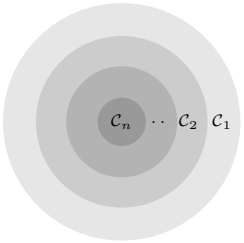
negations. These classes are very small and problem 2) is easy to prove, but problem 1) is much more involved, and this is where constructiveness is lost in both proofs.

Instead of splitting the difficulty in two by introducing one intermediate class, we introduce several classes $\mathcal{C}_n \subseteq \cdots \subseteq \mathcal{C}_1$ and generalize the proof scheme used earlier:

1) For all $i \in [1, n[$ and for every valid formula $\varphi_i \in \mathcal{C}_i$, there is a valid formula $\varphi_{i+1} \in \mathcal{C}_{i+1}$ such that $\varphi_{i+1} \vdash_{\mu\mathsf{LK}} \varphi_i$.
2) Every valid formula of $\mathcal{C}_n$ is provable.

As before, we combine these results to get completeness. The interest of this approach is to split the difficult problem of completeness into several easier problems, for which we can hope to construct effectively a proof.

$$\frac{\vdash \varphi_n \quad \varphi_n \vdash \varphi_{n-1} \quad \cdots \quad \varphi_2 \vdash \varphi_1}{\vdash \varphi_1}\ (\mathsf{Cut})$$



Now the question is how to find these classes. For that, we identified three sources of complexity that make a valid formula hard to prove: i) The alternation of disjunctions and conjunctions, ii) The interleaving of least and greatest fixed points, iii) The presence of disjunctions. In automata theory, these sources of complexity also exist with different names: i) Alternation (of universal and existential non-determinism), ii) The use of parity conditions, and iii) Non-determinism. In automata over infinite words, all these difficulties can be reduced through effective algorithms, transforming automata with one of these difficulties into others without. For example, one has algorithms to eliminate alternation, to reduce the number of priorities for a parity condition or to get rid of non-determinism. The correspondence between linear-time μ-calculus formulas and alternating parity word automata (APW) over infinite words is now very well established. This is fortunate since our idea was to import these techniques from the automata side to the logical one. Concretely, it is known that we can encode every APW $\mathcal{A}$ by a formula $[\mathcal{A}]$ such that the language of $\mathcal{A}$ equals the set of models of $[\mathcal{A}]$. The intermediate classes we will use are the following: The largest class, denoted [APW], is the image of APW by this encoding; this class embodies all the difficulties indicated above. The next class is [NPW], the image of non-deterministic parity automata (NPW) by this encoding. The formulas of this class do not contain the first level of complexity which is the alternation $\vee, \wedge$. The third class is [NBW], the encoding of non-deterministic Büchi automata (NBW). Büchi automata are particular cases of parity automata where only the two priorities 0 and 1 are allowed. We can say then that in this class we simplified the two difficulties i) and ii). The smallest class is [DBW], the image of deterministic Büchi automata, where the three difficulties are eliminated. The proof will be carried out in the following 5 steps:

I	$\forall \varphi \in \mathcal{C}_0, \exists \mathcal{A} \in$ APW such that: $\mathcal{L}(\mathcal{A}) = \mathcal{M}(\varphi)$ and $[\mathcal{A}] \vdash_{\mu\mathsf{LK}} \varphi$.
II	$\forall \mathcal{A} \in$ APW, $\exists \mathcal{P} \in$ NPW such that: $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A})$ and $[\mathcal{P}] \vdash_{\mu\mathsf{LK}} [\mathcal{A}]$.
III	$\forall \mathcal{P} \in$ NPW, $\exists \mathcal{B} \in$ NBW such that: $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{P})$ and $[\mathcal{B}] \vdash_{\mu\mathsf{LK}} [\mathcal{P}]$.
IV	$\forall \mathcal{B} \in$ NBW, if $\mathcal{L}(\mathcal{B}) = \Sigma^\omega$ then $\exists \mathcal{D} \in$ DBW *s.t.*: $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$ and $[\mathcal{D}] \vdash_{\mu\mathsf{LK}} [\mathcal{B}]$.
V	$\forall \mathcal{D} \in$ DBW, if $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$ then $\vdash_{\mu\mathsf{LK}} [\mathcal{D}]$.

Step IV is a bit special because in general NBW cannot be determinized into DBW. But if a NBW $\mathcal{B}$ recognizes the universal language Σ^ω, there is obviously a DBW $\mathcal{D}$ with the same language: the complete Büchi automaton with exactly one (accepting) state for instance. This is enough for our needs, since we start in the proof of completeness from a valid formula φ (*i.e.*, $\mathcal{M}(\varphi) = \Sigma^\omega$), hence the automata $\mathcal{A}, \mathcal{P}$ and $\mathcal{B}$ constructed in steps I-III all recognize the language Σ^ω. To show that $[\mathcal{D}] \vdash_{\mu\mathsf{LK}} [\mathcal{B}]$ in step IV, we use a more general result from [8], which shows that for every Büchi automata $\mathcal{B}_1, \mathcal{B}_2$ such that $\mathcal{L}(\mathcal{B}_2) \subseteq \mathcal{L}(\mathcal{B}_1)$ one has $[\mathcal{B}_2] \vdash_{\mu\mathsf{LK}} [\mathcal{B}_1]$.

We now give an idea of how to prove the sequents of the other steps. Actually, what makes the proof search difficult in $\mu\mathsf{LK}$, is the rule (ν) shown below, where S should be guessed.

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

To circumvent this problem, we go through an intermediate proof system where the rule (ν) just unfolds the ν-formula:

$$\frac{\Gamma \vdash \Delta, F[\nu X.F/X]}{\Gamma \vdash \Delta, \nu X.F}\ (\nu)$$

Two examples of such proof systems are the one introduced in [9] which we call $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$, and the one introduced in [8], called $\mu\mathsf{LK}^\omega$. The proofs of $\mu\mathsf{LK}^\omega$ and $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$, which have the shape of graphs, are called *circular* proofs. The idea is to find a circular proof for the sequent to prove, then to transform this circular proof into a $\mu\mathsf{LK}$ one. The advantage of $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ is that it is competely invertible and the proof search is a trivial task. However, the algorithms known to transform effectively $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ proofs into $\mu\mathsf{LK}$ ones are very restrictive. In contrast, we have given a strong translation result for $\mu\mathsf{LK}^\omega$ ([8]), based on a general geometric condition on proofs. Building on this, we shall work with $\mu\mathsf{LK}^\omega$. To get this stronger translatability criterion, $\mu\mathsf{LK}^\omega$ uses sequents of a particular shape. Indeed, sequents are not sets of formulas, as it is the case for $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$; but are rather sets of *formula occurrences*. The difficulty of using such sequents is that the proof system is not invertible and proving the sequents of steps I-V in $\mu\mathsf{LK}^\omega$ is not immediate.

Let us finally emphasize that the implications appearing in steps I-V are well known at the semantical level, but lifting them to the provability level is not immediate and strongly depends on the encoding $[\_]$ and the shape of automata obtained

by the different automata transformations. To illustrate this by an extremal example, any valid formula φ is semantically equivalent to $\top$ and to itself, but proving $\top \vdash \varphi$ is as difficult as proving $\vdash \varphi$, while proving $\varphi \vdash \varphi$ is immediate. In general, given a formula ψ semantically equivalent to φ, closer ψ is to φ, the easier $\psi \vdash \varphi$ will be to prove. That is why we will provide for our development an encoding of automata that follows closely their structure; and automata transformations that do not change brutally the input automaton (or the input formula for step I). That is also why we cannot treat these transformations as black boxes and will recall them in detail.

Organization of the paper In Section II we introduce the linear-time μ-calculus and its semantics together with $\mu\mathsf{LK}$ and $\mu\mathsf{LK}^\omega$. Then we state a sufficient condition that ensures the translatability of $\mu\mathsf{LK}^\omega$ proofs into $\mu\mathsf{LK}$ ones. In Section III, we present the model of APW and their encoding $[\_]$ in the linear-time μ-calculus. Conversely, we give a way to build for every μ-calculus formula φ an APW $\mathcal{A}_\varphi$ that recognizes the set of its models. The main result of this section is $[\mathcal{A}_\varphi] \vdash_{\mu\mathsf{LK}} \varphi$. In Section IV, we recall the automata transformations that turn an APW $\mathcal{A}$ into an NPW $\mathcal{P}$, and $\mathcal{P}$ into an NBW $\mathcal{B}$, all having the same language. The main results of this section are $[\mathcal{B}] \vdash_{\mu\mathsf{LK}} [\mathcal{P}]$ and $[\mathcal{P}] \vdash_{\mu\mathsf{LK}} [\mathcal{A}]$. In Section V, we show that for every NBW $\mathcal{B}$ recognizing the language Σ^ω, there is a DBW $\mathcal{D}$ recrognizing also Σ^ω, such that $[\mathcal{D}] \vdash_{\mu\mathsf{LK}} [\mathcal{B}]$ and $\vdash_{\mu\mathsf{LK}} [\mathcal{D}]$. We finally bring these pieces together to get a constructive proof of completeness.

II. Linear-time μ-calculus and its proof systems

In this section we introduce the linear-time μ-calculus and its semantics, together with two proof systems. The first is $\mu\mathsf{LK}$, the target of our completeness result. The second is $\mu\mathsf{LK}^\omega$, which will serve as an intermediate proof system. At the end of this section, we show a sufficient condition that ensures the translatabilty of $\mu\mathsf{LK}^\omega$ proofs into $\mu\mathsf{LK}$ ones.

A. Syntax and semantics

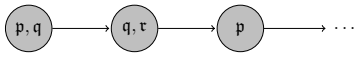
Definition 1. Let $\mathcal{V} = \{X, Y, \dots\}$ be a set of variables and $\mathcal{P} = \{\mathfrak{p}, \mathfrak{q}, \dots\}$ a set of atoms. The linear-time μ-calculus ***formulas*** $\varphi, \psi, \dots$, called simply formulas, are given by:

$$\varphi ::= \mathfrak{p} \mid \neg\mathfrak{p} \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \odot\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

The connectives μ and ν bind the variable X in φ. From there, bound variables, free variables and capture-avoiding substitution are defined in a standard way. The subformula ordering is denoted $\leq$ and $\mathsf{fv}(\bullet)$ denotes free variables. Atoms and their negations are called **litterals**. We shall use σ to denote either μ or ν.

Note that we do not allow negations on variables. This is not a restriction since we are mostly interested in closed formulas. All the results presented here extend to the general case, where negations are allowed under a positivity condition on bound variables. We do not consider the boolean constants $\top, \bot$ as they can be encoded by $\top := \nu X. \odot X$ and $\bot := \mu X. \odot X$.

The models of our formulas are the ω-words over the alphabet $\Sigma := 2^{\mathcal{P}}$. Intuitively, every position of such a word corresponds to an instant of time, and a letter at some position represents the set of atoms true at the corresponding instant of time. In the example below, atoms $\mathfrak{p}, \mathfrak{q}$ are true at instant 0, atoms $\mathfrak{q}, \mathfrak{r}$ are true at instant 1, etc.



We define the semantics of a formula *w.r.t.* a model to be the set of instants of time where the formula holds in this model.

Definition 2. The ***semantics*** $\|\varphi\|^u_\rho$ of a formula F *w.r.t.* $u \in \Sigma^\omega$ and a valuation $\rho : \mathcal{V} \mapsto 2^\omega$ is a subset of natural numbers inductively defined as follows:

$$\|\mathfrak{p}\|^u_\rho = \{i \in \omega \mid \mathfrak{p} \in u_i\} \qquad \|\neg\mathfrak{p}\|^u_\rho = \{i \in \omega \mid \mathfrak{p} \notin u_i\}$$
$$\|X\|^u_\rho = \rho(X) \qquad \|\odot\varphi\|^u_\rho = \{i \in \omega \mid i+1 \in \|\varphi\|^u_\rho\}$$
$$\|\varphi \vee \psi\|^u_\rho = \|\varphi\|^u_\rho \cup \|\psi\|^u_\rho \qquad \|\varphi \wedge \psi\|^u_\rho = \|\varphi\|^u_\rho \cap \|\psi\|^u_\rho$$
$$\|\nu X.\varphi\|^u_\rho = \bigcup\{\, W \subseteq \omega \mid W \subseteq \|\varphi\|^u_{\rho[X \leftarrow W]} \,\}$$
$$\|\mu X.\varphi\|^u_\rho = \bigcap\{\, W \subseteq \omega \mid \|\varphi\|^u_{\rho[X \leftarrow W]} \subseteq W \,\}$$

Suppose that φ is a closed formula. We write $\|\varphi\|^u$ instead of $\|\varphi\|^u_\rho$, since the semantics of φ do not depend on ρ. We say that φ is ***true*** in u, and we write $u \models \varphi$, if $0 \in \|\varphi\|^u$. The set of ***models of*** φ is defined by $\mathcal{M}(\varphi) = \{u \in \Sigma^\omega \mid u \models \varphi\}$. A formula is ***valid*** if it is true in every model, ie. $\mathcal{M}(\varphi) = \Sigma^\omega$.

B. Proof systems for linear-time μ-calculus

There are many possible presentations of sequent calculus, which differ by the way sequents are defined. Sometimes sequents are presented as sets or multisets of formulas, but most proof-theoretical observations, in particular the proofs-as-programs correspondence, hold in a setting where sequents are sets of formula *occurrences*. We choose to work with the latter presentation because this viewpoint was necessary in a previous work [8], which is an essential building block for our completeness proof, and for technical reasons that will be clear later in the paper. We present below the notion of occurrence and use it to build $\mu\mathsf{LK}$ and $\mu\mathsf{LK}^\omega$.

1) Occurrences:

Definition 3. An ***address*** is a word over $\{l, r, i\}$, which stands for left, right and inside. We denote by ε the empty address. We say that α' is a ***sub-address*** of α when α is a prefix of α', written $\alpha \sqsubseteq \alpha'$. We say that α and β are ***disjoint*** when α and β have no upper bound *w.r.t.* $\sqsubseteq$.

Definition 4. An ***occurrence*** (denoted by F, G, H) is given by a formula φ and an address α, and written φ_α. We say that two occurrences are ***disjoint*** when their addresses are. Let $F = \varphi_\alpha$ be an occurrence and β an address. We define F_β to be φ_β . We say that we ***relocated F in*** β. We define $\overline{F}$ to be φ. We write $F \equiv G$ if $\overline{F} = \overline{G}$, we say that F and G are equal up to renaming. Operations on formulas are extended to occurrences as follows: for any $\star \in \{\vee, \wedge\}$, $F \star G = (\varphi \star \psi)_\alpha$ if $F = \varphi_{\alpha l}$ and $G = \psi_{\alpha r}$; $\sigma X.F = (\sigma X.\varphi)_\alpha$ and $\odot F = (\odot\varphi)_\alpha$ if $F = \varphi_{\alpha i}$; we also allow ourselves to write litterals as occurrences without specifying their address. ***Substitution of occurrences*** is defined by $(\varphi_\alpha)[\psi_\beta/X] = (\varphi[\psi/X])_\alpha$.

Formulas with fixed points support two notions of subformula. The first is the usual one: $\psi \leq \varphi$ if the syntactic tree of ψ is a sub-tree of the syntactic tree of φ. In general, the subformula of a closed formula may contain free variables. The second one is specific to formulas with fixed points, it is a sort of subformula up to unfolding, so that the subformulas of a closed formula *w.r.t.* this notion are also closed; we call it *Fischer-Ladner* subformula and introduce it below.

Definition 5. We define the relation $\to$ on occurrences as follows, where $\star \in \{\vee, \wedge\}$:

$$(\varphi \star \psi)_\alpha \ \to \ \varphi_{\alpha l} \quad (\varphi \star \psi)_\alpha \ \to \ \psi_{\alpha r}$$
$$(\odot\varphi)_\alpha \ \to \ \varphi_{\alpha i} \qquad (\sigma X.\varphi)_\alpha \ \to \ (\varphi[\sigma X.\varphi/X])_{\alpha i}$$

A FL-***suboccurrence*** of F is any G such that $F \to^* G$, where $\to^*$ is the reflexive transitive closure of $\to$. The FL-***subformulas*** of F are obtained by forgetting the adresses of its FL-suboccurrences. The ***Fischer-Ladner closure*** of a formula occurrence, denoted $\mathsf{FL}(F)$, is the set of its FL-subformulas.

Example 1. Let $\Phi = \mu X.\nu Y. \odot X \wedge \odot Y$ and $\Psi = \nu Y. \odot \Phi \vee \odot Y$. We have for instance:

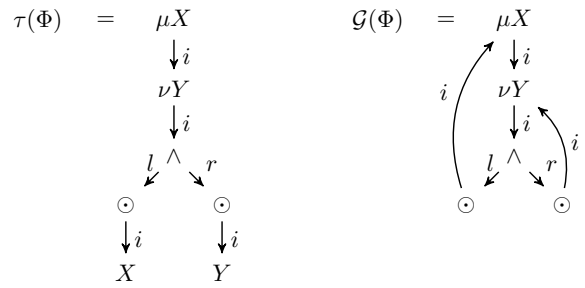
$$\Phi_\varepsilon \to \Psi_i \to (\odot\Phi \vee \odot\Psi)_{ii} \to (\odot\Phi)_{iil} \to \Phi_{iili} \to \Psi_{iilii}$$

We have $\mathsf{FL}(\Phi_\varepsilon) = \{\Phi, \Psi, \odot\Phi \vee \odot\Psi, \odot\Psi, \odot\Phi\}$.

It is well-known that $\mathsf{FL}(F)$ is finite. The FL-suboccurrences of F are induced by traversals of the *graph* of F, *i.e.*, the graph obtained from the tree of F, by adding the possibility of jumping from a variable to the fixed-point combinator introducing it. This observation is made precise in the following.

Definition 6. The ***tree of a formula*** φ, denoted $\tau(\varphi)$, is obtained from the syntactic tree of φ by labelling every edge e as follows: if e is the right (resp. left) outgoing edge of a binary connective, then it is labelled r (resp. l); otherwise it is labelled i. The ***graph of a formula*** φ, denoted $\mathcal{G}(\varphi)$, is the rooted graph obtained from $\tau(\varphi)$ by identifying the nodes of bound variables with their binders.

Example 2. Let Φ be the formula of Example 1. The tree and the graph of Φ are the following:



Proposition 1. *Let $F = \varphi_\alpha$ be an occurrence. If ψ_β is a* FL*-suboccurrence of F, then $\beta = \alpha.p$, where p is a path of $\mathcal{G}(\varphi)$ from the root to some node n, denoted $\mathcal{N}_F(\psi_\beta)$.*

While defining the circular proof system, we will deal with sequences of occurrences related by $\to$, that we call *threads*.

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

$$\frac{F \equiv G}{F \vdash G}\ (\mathsf{Ax}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ (\mathsf{W_l}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ (\mathsf{W_r})$$

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

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

Fig. 1: Inference rules for propositional connectives.

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

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

Definition 7. A ***thread*** of F is a sequence $t = (F_i)_{i \in o}$, where $o \in \omega + 1$ *s.t.* $F_0 = F$ and $\forall i \in o,\ F_i \to F_{i+1}$ or $F_i = F_{i+1}$. Let $\bar{t}$ be the sequence $(\overline{F_i})_{i \in o}$ *i.e.*, the sequence obtained by forgetting the adresses of the formula occurrences of t. We denote by $\mathsf{Inf}(t)$ the elements of $\bar{t}$ that appear infinitely often in $\bar{t}$.

A thread t starting from F can be seen as a path in the graph $\mathcal{G}(F)$. Since $\bar{t} \subseteq \mathsf{FL}(F)$, $\mathsf{Inf}(\bar{t})$ is finite. The following proposition shows that $\mathsf{Inf}(\bar{t})$ admits a minimum.

Proposition 2. *Let $t = (F_i)_{i \in \omega}$ be a thread of F. The set $\mathsf{Inf}(\bar{t})$ admits a minimum* w.r.t. $\leq$*, we denote it* $\mathsf{min}(\mathsf{Inf}(\bar{t}))$.

Example 3. Let t be the thread of Φ_ε (from Example 1) that goes to the right: $t = \Phi_\varepsilon \to \Psi_i \to^\star \Psi_{iiri} \to^\star \Psi_{iiriiri} \dots$ We have $\mathsf{Inf}(\bar{t}) = \{\Psi, \odot\Phi \wedge \odot\Psi, \odot\Psi\}$ and $\mathsf{min}(\mathsf{Inf}(\bar{t})) = \Psi$.

We are now ready to introduce our sequent calculus.

2) Circular proof system:

Definition 8. A ***sequent***, written $\Delta \vdash \Gamma$, is pair of two finite sets of pairwise disjoint occurrences. A ***pre-proof*** of $\mu\mathsf{LK}^\infty$ is a possibly infinite tree, coinductively generated by the rules of Figures 1 and 2.

The disjointness condition on sequents ensures that two occurrences from the same side of a given sequent will never engender a common sub-occurrence. Note that if the disjointness condition is satisfied for the conclusion sequent of a pre-proof, then all its sequents satisfy it, as soon as occurrences of cut-formulas are appropriately chosen [11].

We sometimes write sequents of the form $\varphi_\varepsilon \vdash \psi_\varepsilon$ as $\varphi \vdash \psi$.

Pre-proofs are unsound: it is easy to derive the formula $\mu X. \odot X$, which is semantically equivalent to $\bot$, by applying coinductively (μ_r) rule followed by $(\odot)$ rule. In order to obtain proper proofs from pre-proofs, we add a validity condition that reflects the nature of our two fixed points.

Definition 9. A thread t is said to be a μ-***thread*** (resp. ν-***thread***) if $\mathsf{min}(\mathsf{Inf}(\bar{t}))$ is a μ (resp. ν) formula. Let $\gamma = (\Delta_i \vdash \Gamma_i)_{i \in \omega}$ be an infinite branch in a pre-proof of $\mu\mathsf{LK}^\infty$. A ***thread***

$t = (F_i)_{i\in\omega}$ is a right (resp. left) thread of γ if $\forall i \in \omega,\ \ F_i \in \Gamma_i$ (resp. $F_i \in \Delta_i$). The branch γ is said ***valid*** if it contains a right ν-thread or a left μ-thread which is not stationary.

Definition 10. The ***proofs*** of $\mu\mathsf{LK}^\infty$ are those pre-proofs in which every infinite branch is valid.

This validity condition has its roots in parity games and is very natural for infinitary proof systems with fixed points. It is commonly found in deductive systems for modal μ-calculi: see [9] for a closely related presentation, which yields a sound and complete sequent calculus for linear-time μ-calculus.

In this work, we will be interested in a subsystem of $\mu\mathsf{LK}^\infty$ whose proofs are *finitely* representable, we call it $\mu\mathsf{LK}^\omega$.

Definition 11. Two $\mu\mathsf{LK}^\infty$ proofs Π, Θ are said to be ***equal up to renaming***, if there is a bijection b on adresses such that the proof obtained from Π by replacing every occurrence φ_α by $\varphi_{b(\alpha)}$ is Θ. A $\mu\mathsf{LK}^\omega$ proof is said to be ***circular*** if it has only finitely many sub-derivations up to renaming.

Definition 12. The ***circular proof system*** $\mu\mathsf{LK}^\omega$ is the restriction of $\mu\mathsf{LK}^\infty$ to circular proofs.

The proofs of $\mu\mathsf{LK}^\omega$ can be represented by trees with loops. For instance, the $\mu\mathsf{LK}^\omega$ proof of $\top = \nu X.\odot X$ is below, where we indicated the loop using a symbol $(\star)$:

$$\dfrac{\dfrac{\dfrac{(\star)}{\vdash \top_{ii}}}{\vdash (\odot\top)_i}\ (\odot)}{\vdash \top_\varepsilon \quad (\star)}\ (\nu_{\mathsf{r}})$$

The proof system $\mu\mathsf{LK}^\omega$ is very close to the one in [9], that we call $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$. The sequents of $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ are sets of formulas, thus the proof search is trivial. This is not the case for $\mu\mathsf{LK}^\omega$. For instance, let $\varphi = \mu X.\nu Y.X \vee Y$ and $\psi = \nu Y.\varphi \vee Y$ its unfolding. The $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$ proof π of $\varphi \vee \psi$ (Fig. 3) is obtained by applying bottom up all the possible logical rules. If we apply the same rules in $\mu\mathsf{LK}^\omega$, we get the $\mu\mathsf{LK}^\infty$ proof θ (Fig. 3), which is not circular since the size of its sequents is unbounded. To get a circular proof, we have to apply some weakenings. But the choice of which formula to weaken is crucial, since a bad choice may lead to a non valid proof. For instance, if we weaken the formula ψ_r, the obtained derivation θ_1 (Fig. 3) is circular but does not satisfy the validity condition. The good choice of weakening is the one that fires φ_l, yielding the proof θ_2 (Fig. 3). Proving a valid sequent in $\mu\mathsf{LK}^\omega$ is not trivial, since we have to do some clever choices to get derivations which are circular and valid.

3) Finitary proof system: We present now $\mu\mathsf{LK}$, the restriction of Kozen's axiomatization for the modal μ-calculus to the linear time, written in a sequent calculus fashion.

Definition 13. The proofs of $\mu\mathsf{LK}$ are finite trees inductively generated from the rules of Figures 1 and 4.

In $\mu\mathsf{LK}^\omega$, μ and ν have the same rules, but their difference in nature is reflected by the validity condition. In $\mu\mathsf{LK}$, the rules for μ and ν are distinct; they are derived from the Knaster-

$$\pi = \dfrac{\dfrac{\dfrac{(\star)}{\vdash \varphi \vee \psi}}{\vdash \varphi, \psi}\ (\mu),(\nu),(\nu)}{\vdash \varphi \vee \psi \quad (\star)}\ (\vee) \qquad \theta = \dfrac{\dfrac{\vdots}{\vdash (\varphi \vee \psi)_{lii}, (\varphi \vee \psi)_{ri}}}{\vdash \varphi_l, \psi_r}\ (\mu),(\nu),(\nu)\ (\vee)$$

$$\theta_1 = \dfrac{\dfrac{\dfrac{\dfrac{(\star)}{\vdash (\varphi \vee \psi)_{lii}}}{\vdash \varphi_l}\ (\mu),(\nu)}{\vdash \varphi_l, \psi_r}\ (\mathsf{W})}{\vdash (\varphi \vee \psi)_\varepsilon\ (\star)}\ (\vee) \qquad \theta_2 = \dfrac{\dfrac{\dfrac{\dfrac{(\star)}{\vdash (\varphi \vee \psi)_{ri}}}{\vdash \psi_r}\ (\nu)}{\vdash \varphi_l, \psi_r}\ (\mathsf{W})}{\vdash (\varphi \vee \psi)_\varepsilon\ (\star)}\ (\vee)$$

Fig. 3: $\mu\mathsf{LK}^\omega_{\mathsf{DHL}}$, $\mu\mathsf{LK}^\infty$ and $\mu\mathsf{LK}^\omega$ derivations of $\varphi \vee \psi$

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

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

Fig. 4: Fixed point rules for the $\mu\mathsf{LK}$ proof system.

Tarski characterization of $\mu X.F$ as the least pre-fixed point of $X \mapsto F$, and dually for $\nu X.F$.

Definition 14. A formula F is ***guarded*** if every bound variable of F appear under the scope of a $\odot$ connective.

In [6], it is shown that every formula if provably equivalent to a guarded formula in $\mu\mathsf{LK}^\omega$, and this proof is constructive.

Proviso: If not otherwise stated all formulas are assumed to be closed, guarded, $\top$ and $\bot$ free. By earlier observations, this is not a restriction.

4) Relating the infinitary and the finitary proof systems: Finitary proofs can be easily transformed into circular ones, but giving an effective transformation from circular to finitary proofs is still an open problem. However, in [8] a condition on $\mu\mathsf{LK}^\omega$ proofs is given, which is sufficient to translate them effectively into $\mu\mathsf{LK}$ ones. This condition is much involved and we do not need it here in all its generality. A weaker condition, presented below, will be used instead.

Definition 15. A $\mu\mathsf{LK}^\omega$ derivation is ***thin*** if all the occurrences of (σ_{r}) and (σ_{l}) are of the following form, *i.e.*, there is no context around the unfolded formulas:

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

Thin derivations are very close to thin refutations [6].

Proposition 3. *If π is a thin proof of a sequent s, then it can be transformed effectively into a $\mu\mathsf{LK}$ proof of s.*

Proof. It suffices to notice that in a thin derivation every valid branch is strongly valid (Definition 29 of [8]). Hence every thin proof is translatable (Definition 29 of [8]), by Proposition 30 of [8], it can be transformed effectively into a $\mu\mathsf{LK}$ proof of the same sequent. □

III. Alternating parity automata and μ-calculus

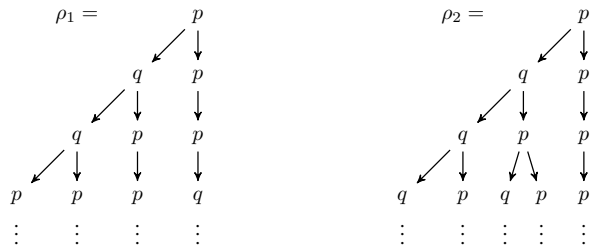
There is a fruitful relationship between μ-calculus and automata theory, at the core of which lies the equivalence between linear-time μ-calculus formulas and alternating parity automata (APW). This equivalence is always shown at a semantical level: every APW $\mathcal{A}$ can be encoded by a formula $[\mathcal{A}]$ such that $\mathcal{M}([\mathcal{A}]) = \mathcal{L}(\mathcal{A})$, and conversely for every formula φ, there is an APW $\mathcal{A}_\varphi$ such that $\mathcal{M}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$. We show in this section that beyond this semantical equivalence, there is also an equivalence at the level of provability, that is, φ and $[\mathcal{A}_\varphi]$ are provably equivalent in μLK. Lifting this semantical equivalence to the provability level relies very precisely on the encoding and the shape of the automaton $\mathcal{A}_\varphi$. That is why we introduce our own encoding of automata in the μ-calculus, that respects the shape of the input automaton; and our construction of $\mathcal{A}_\varphi$, which yields an automaton that sticks the most to the structure of φ. This section is organized as follows: In III-A, we recall the model of APW and define in III-B our encoding of APW int the μ-calculus. Conversely, we construct in III-D for every μ-calculus formula an APW, and show that the encoding of this automaton is provably equivalent to the original formula in μLK. To do so, we need a technical tool, which is an alternative definition of μ-calculus semantics introduced in [12], that we recall in III-C.

A. Alternating parity word automata

Alternating parity automata over words (APW) are finite-state machines designed to accept or reject infinite words. The computation of an APW over an infinite word proceeds in rounds. At the beginning of every round, there are several copies of the APW in some position of the word, each of them in its own state. During a round, each copy splits up in several new copies, which are sent to the successor of the current position, and change their states, according to the transition function. Initially, there is only one copy of the APW, which is in the initial state; and which resides in the first position of the word. Every computation induces a tree, labelled with the states through which the copies of the automaton went during the computation; this tree is called a run and it witnesses the causality between a copy of the automaton and the new copies its yields. The acceptance or rejectance of a computation is defined via path conditions for the infinite branches of the run. Namely, every state is assigned a priority and an infinite branch of the run is accepting if the minimal priority occurring infinitely often is even; a run is accepting if all its infinite branches are accepting. We can see then the branching of runs as a universal non-determinism,; but APW support also existential non-determinism, in the sens that an automaton may have many possible computations over a single word. Formally APW are defined as follows.

Definition 16. An ***APW*** $\mathcal{A}$ is a tuple $(\Sigma, Q, \Delta, q_I, c)$, where Σ is an alphabet, Q is a finite set of states, $\Delta \subseteq Q \times \Sigma \times 2^Q$ is the transition relation, $q_I \in Q$ is the initial state and $c : Q \to \omega$ is the priority function, which assigns a priority to each state. A ***run*** of $\mathcal{A}$ on a word $u = (a_i)_{i\in\omega} \in \Sigma^\omega$ is a labelled tree such that: the label of the root is q_I; and for every node v at level n, if q is the label of v and if E is the set of labels of the sons of v, then $(q, a_n, E) \in \Delta$.

Example 4. Let $\mathcal{A} = (\{a\}, Q, \Delta, p, c)$ be the APW where $Q = \{p, q\}$, $\Delta = \{t_1, t_2, t_3\}$ where $t_1 = (p, a, \{p\})$, $t_2 = (p, a, \{p, q\})$, $t_3 = (q, a, \{p, q\})$ and $c(p) = 1, c(q) = 2$. ρ_1 and ρ_2 are the beginning of two runs of $\mathcal{A}$ over a^ω.



Definition 17. Let $\mathcal{A} = (\Sigma, Q, \Delta, q_I, c)$ and ρ be an infinite word over Q. The word ρ is ***accepting*** if $\min \{ c(q) \mid q \in \mathrm{Inf}(\rho) \}$ is even. A run of $\mathcal{A}$ is ***accepting*** if all its branches (seen as words over Q) are. We say that $\mathcal{A}$ ***accepts*** the word u if there exists an accepting run of $\mathcal{A}$ on u. The ***language*** of $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) = \{ u \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } u \}$. A language $\mathcal{L}$ is ***recognized by*** $\mathcal{A}$ if $\mathcal{L} = \mathcal{L}(\mathcal{A})$.

Non-deterministic automata are particular cases of alternating automata where the existential non-determinism is allowed but not the universal one. We present below two classes of non-deterministic automata which are subclasses of APW: non-deterministic parity automata (NPW) and non-deterministic Büchi automata (NBW).

Definition 18. A ***NPW*** is an APW where the elements of the transition relation are of the form $(p, a, \{q\})$. A ***NBW*** is a NPW $(\Sigma, Q, \Delta, q_I, c)$ where the domain of c is $\{0, 1\}$. A state q of a NBW is said to be ***accepting*** if $c(q) = 0$. We present sometimes NBW as tuples of the form $(\Sigma, Q, \Delta, q_I, F)$ where F is the set of accepting states.

For a non-deterministic automaton $\mathcal{A} = (\Sigma, Q, \Delta, q_I, c)$ we can consider that Δ is a subset of $Q \times \Sigma \times Q$ and that the runs are infinite words over Q.

B. Encoding APW in the μ-calculus

The linear-time μ-calculus contains all the ingredients to encode APW : disjunction and conjunction to simulate existential and universal non-determinism; least and greatest fixed points to encode odd and even states. To get a match between the language of an automaton and the models of the formula encoding it, we will consider automata over the alphabet $\Sigma = 2^{\mathcal{P}}$ where $\mathcal{P}$ is the set of atoms, whose elements will simply be denoted by a, b, etc.

1) The encoding: We first show our encoding of the letters of Σ in the μ-calculus.

Definition 19. Let $a \in \Sigma$. The encoding of a, denoted $[a]$, is the formula $[a] := (\wedge_{\mathfrak{p}\in a}\mathfrak{p}) \wedge (\wedge_{\mathfrak{q}\notin a}\mathfrak{q}^\perp)$.

Definition 20. Let $\mathcal{A} = (\Sigma, Q, \Delta, q_I, c)$ be an APW . A ***run-section*** is a sequence $\Gamma = ((q_i, t_i))_{0 \leq i \leq n}$ of pairs in $Q \times \Delta$ *s.t.* $q_0 = q_I$ and $\forall i \leq n,\ \exists a_i, E_i$ st. $t_i = (q_i, a_i, E_i)$ and $\forall i < n,\ q_{i+1} \in E_i$. We say that Γ ***enables*** a state q if $q \in E_n$.

Definition 21. Let $\mathcal{A} = (Q, q_I, \Delta, c)$ be an APW . 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 run-section Γ, as follows:

$$[q]^\Gamma = X_q \quad \text{if} \begin{cases} (1)\ \Gamma = \Gamma', (q,t), (q_1,t_1), \ldots, (q_n,t_n) \text{ and} \\ \\ (2)\ q_i \neq q, c(q_i) \geq c(q) \text{ for all } 1 \leq i \leq n, \end{cases}$$

$$[q]^\Gamma = \sigma X_q. \bigvee_{a \in \Sigma, t=(q,a,E) \in \Delta} [a] \wedge \bigwedge_{p \in E} \odot [p]^{\Gamma,(q,t)} \text{ otherwise}$$

with $\sigma = \nu$ iff $c(q)$ is even.

We finally set $[\mathcal{A}] = [q_I]^\emptyset$.

The encoding starts from the initial state and traverses the automaton, encoding every state q by a fixed point μX_q, if q is a state with an odd priority and νX_q otherwise. The environment Γ remembers the states that have been visited from the initial state to the current state. If the current state q has been seen before (and if the states seen since the last time q was visited have bigger priorities) then we encode it by its corresponding variable X_q, which ensures that the algorithm of encoding will halt at some point.

Example 5. The encoding of $\mathcal{A}$ of Example 4 is the formula:

$$[\mathcal{A}] = \mu X_p.([a] \wedge \odot X_p) \vee ([a] \wedge \odot(\nu X_q.[a] \wedge \odot X_q \wedge \odot X_p) \wedge \odot X_p)$$

A key ingredient in this encoding is the side condition of the first case of the definition. The aim of this condition 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 (see Example 16 in [8]).

Although not necessary for the completeness proof, we prove the following proposition, which shows that the language of an automaton and the models of its encoding are equal.

Proposition 4. *For any APW $\mathcal{A}$, $\mathcal{M}([\mathcal{A}]) = \mathcal{L}(\mathcal{A})$*

Proof. For this proof we use definitions and result from Section III-C. We show that the automaton $\mathcal{A}$ and the formula $[\mathcal{A}]$ have the same language ie. $\mathcal{L}(\mathcal{A}) = \mathcal{L}([\mathcal{A}])$.

We show first that $\mathcal{L}([\mathcal{A}]) \subseteq \mathcal{L}(\mathcal{A})$. Let $u \in \mathcal{L}([\mathcal{A}])$ and let β be an accepting branch of $\Pi([\mathcal{A}])$ that induces u. Let $s_1 s_2 \ldots$ be the sequence of conclusions of $\odot$ rules in β. Every s_i is of the from $C_i, \odot\{[\![q_k]\!]^{\Gamma_k}\}_{i_0 \leq k \leq i_{l_i}}$ where $\overline{C_i} = C(u_i)$. This branch induces easily a run ρ of $\mathcal{A}$ over u, where at every level i of ρ, the set of nodes labels is $\{q_k\}_{i_0 \leq k \leq i_{l_i}}$. Every branch of ρ is a run-branch of a thread t of β. Since β is accepting, every thread t of β is a ν-thread, thus by Proposition 7 every branch of ρ is valid, hence ρ is valid.

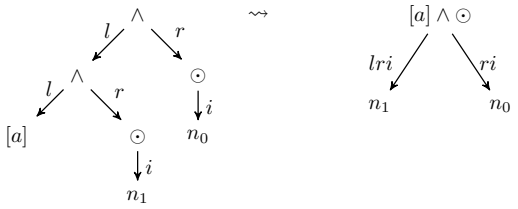
We show the other direction $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}([\mathcal{A}])$ by using a similar argument: to every valid run of $\mathcal{A}_\varphi$ over a word u, one can find an accepting branch of $\Pi(\varphi)$ that induces the word u.

By Proposition 8, $\mathcal{L}(\varphi) = \mathcal{M}(\varphi)$, which concludes the proof. □

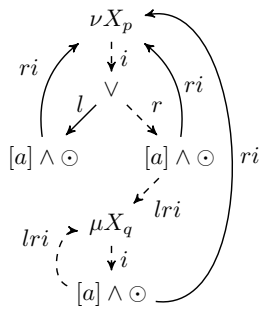
2) FL-suboccurrences of the encoding: The $\mu\mathsf{LK}^\omega$ proofs involving a formula $[\mathcal{A}]$, that we will deal with later, will decompose $[\mathcal{A}]$ into its FL-suboccurrences. To simplify the manipulation of these FL-suboccurrences, we introduce in this section same handy notation, then we relate the threads of $[\mathcal{A}]$, which are sequences of FL-suboccurrences of $[\mathcal{A}]$, to the runs of the automaton $\mathcal{A}$.

Definition 22. Let $\Gamma = ((q_i, t_i))_{0 \leq i \leq n}$ be a run-section of the APW $\mathcal{A}$, and q a state enabled by Γ. Let p be the path in the graph of $[\mathcal{A}]$ that visits successively the nodes labelled σX_{q_i}, according to the transitions t_i, and ends with the node labelled σX_q. The ***address of*** (Γ, q), denoted $\alpha_{\Gamma,q}$, is the label of p.

Example 6. Let $\mathcal{A}$ be the APW of Example 4. To simplify the presentation of the formula graphs, we merge the chain of nodes of the following shape into one node labelled $[a] \wedge \odot$:



The graph of the encoding of $\mathcal{A}$ is the following:



The run-section $\Gamma = (p, t_2), (q, t_3)$ enables the state q, its path is the dashed one. The address $\alpha_{\Gamma,q}$ is $irlriilri$.

Definition 23. Let Γ be a run-section of $\mathcal{A}$, and q a state enabled by Γ. We denote by $[\![q]\!]^\Gamma$ the FL-suboccurrence of $[\mathcal{A}]$ at the address $\alpha_{\Gamma,q}$, *i.e.*, the occurrence $\varphi_{\alpha_{\Gamma,q}}$ such that: $[\mathcal{A}]_\varepsilon \rightarrow^\star \varphi_{\alpha_{\Gamma,q}}$.

Example 7. Let Γ be the run-section of Example 6. We have $[\![q]\!]^\Gamma = (\mu X_q.[b] \wedge \odot X_q \wedge \odot[\mathcal{A}])_{\alpha_{\Gamma,q}}$.

Remark 1. We have chosen the notation $[\![q]\!]^\Gamma$ because of the proximity between these FL-suboccurrences and the formulas

$[q]^\Gamma$. Indeed, one can show that $[\![q]\!]^\Gamma$ is obtained by substituting iteratively the free variables of $[q]^\Gamma$ by their binders in $[\mathcal{A}]$.

Remark 2. The formulas $[\![q]\!]^\Gamma$ are fixed point formulas. Conversely, all the fixed point FL-subformulas of $[\mathcal{A}]$ are of the form $[\![q]\!]^\Gamma$ where Γ, q is a run-section.

The following proposition will be very useful for further proofs. It shows that contrarily to $[q]^\Gamma$, no case analysis on Γ and q is required to figure out the shape of $[\![q]\!]^\Gamma$.

Proposition 5. *Let $\mathcal{A}$ be an APW and $[\![q]\!]^\Gamma$ a FL-suboccurrence of $[\mathcal{A}]$. The top-level connective of $[\![q]\!]^\Gamma$ is σX_q. and we have:*

$$[\![q]\!]^\Gamma \to \bigvee_{t=(q,a,E)\in\Delta} [a] \wedge \bigwedge_{p\in E} \odot\, [\![p]\!]^{\Gamma,(q,t)}$$

Using Proposition 5, we can derive the following rule, which mimics automata transitions in the proof system $\mu\mathsf{LK}^\infty$.

Proposition 6. *Let $\mathcal{A}$ be an APW of transition relation Δ, q be state of $\mathcal{A}$ and Γ a run-section. For every set of occurrences Θ, the following rule is derivable using a thin derivation:*

$$\frac{\{[a], \odot\{[\![p]\!]^{\Gamma,(q,t)}\}_{p\in E} \vdash \Theta\}_{t=(q,a,E)\in\Delta}}{[\![q]\!]^\Gamma \vdash \Theta}$$

Proof. Since $[\![q]\!]^\Gamma \to \bigvee_{t=(q,a,E)\in\Delta} [a] \wedge \bigwedge_{p\in E} \odot\, [\![p]\!]^{\Gamma,(q,t)}$, one can derive the following:

$$\dfrac{\left\{\dfrac{\dfrac{[a], \{\odot[\![p]\!]^{\Gamma,(q,t)}\}_{p\in E} \vdash \Theta}{[a] \wedge \bigwedge_{p\in E} \odot\, [\![p]\!]^{\Gamma,(q,t)} \vdash \Theta}\,(\wedge_{\mathsf{l}})}\right\}_{t=(q,a,E)\in\Delta}}{\dfrac{\bigvee_{t=(q,a,E)\in\Delta} [a] \wedge \bigwedge_{p\in E} \odot\, [\![p]\!]^{\Gamma,(q,t)} \vdash \Theta}{[\![q]\!]^\Gamma \vdash \Theta}\,(\sigma_{\mathsf{l}})}\,(\vee_{\mathsf{l}})$$

Notice that this is a thin derivation. □

Now we relate the threads of the formula encoding an automaton $\mathcal{A}$ to the run branches of $\mathcal{A}$. For that, we define the run-branch corresponding to a thread of $[A]$ as follows:

Definition 24. Let $\mathcal{A}$ be an APW and t be a thread of $[\mathcal{A}]$. Let $([\![q_i]\!]^{\Gamma_i})_{i\in\omega}$ be the sequence of fixed point occurrences appearing in t. The ***run-branch*** of t is $\rho(t) := (q_i)_{i\in\omega}$.

It is easy to see that the run-branch of a thread is indeed a branch in a run of $\mathcal{A}$.

Proposition 7. *Let $\mathcal{A}$ be an APW , c be its priority function. A thread t of $[\mathcal{A}]$ is a ν-thread iff $\rho(t)$ is a valid.*

C. Operational semantics

Our goal now is to extract an APW from every μ-calculus formula. In [12], an alternative definition of formulas semantics is presented, equivalent to Definition 2, but with the advantage that it makes formulas look like automata, checking whether words over Σ are models or not. This is exactly what we need to achieve our goal. Their definition being introduced for the modal μ-calculus, we specialize it to the linear time case.

Definition 25. We call ***constraint*** any set of literals. The ***constraint of a letter*** $a \in \Sigma$, denoted $c(a)$, is the constraint $a \cup \{p^\perp \mid p \notin a\}$. A letter a ***satisfies a constraint*** c, and we write $a \models c$, iff $c \subseteq c(a)$.

Definition 26. Let F be an occurrence. The ***F-derivation*** is a $\mu\mathsf{LK}^\infty$ derivation of conclusion $F \vdash$ obtained by applying all the possible logical rules $(\mu, \nu, \vee, \wedge)$, except for the $\odot$ rule, to F and its sub-occurrences, until reaching sequents of the form $C, \odot\Delta \vdash$, where the set $\overline{C}$ obtained by forgetting the addresses of C, is a constraint. The set of ***successors of*** F, denoted $S(F)$, is the set of premises of the F-derivation.

Notice that the F-derivation is finite, because F is guarded.

Remark 3. There are many possible F-derivations for a given occurrence F, differing only by the order of application of the logical rules. They all share the same set of premises and the same structure of threads. We choose any representative of this class of derivations to be ***the*** F-derivation.

Example 8. Let $F = \nu X.\mathfrak{p} \wedge ((\odot X)_\alpha \vee G) \wedge (\odot X)_\beta$, where $G = \mu Y.\mathfrak{q} \wedge (\odot Y)_\gamma$. The F-derivation is the following:

$$\dfrac{\dfrac{\dfrac{\mathfrak{p}, (\odot F)_\alpha, (\odot F)_\beta \vdash}{\mathfrak{p}, (\odot F)_\alpha \vee G, (\odot F)_\beta \vdash}\,(\vee) \qquad \dfrac{\mathfrak{p}, \mathfrak{q}, (\odot G)_\gamma, (\odot F)_\beta \vdash}{\mathfrak{p}, G, (\odot F)_\beta \vdash}\,(\mu),(\wedge_{\mathsf{l}})}{F \vdash}\,(\nu),(\wedge),(\wedge)$$

We sometimes write an F-derivation as a rule named F:

$$\frac{\mathfrak{p}, (\odot F)_\alpha, (\odot F)_\beta \vdash \qquad \mathfrak{p}, \mathfrak{q}, (\odot G)_\gamma, (\odot F)_\beta \vdash}{F \vdash}\,(F)$$

Recall that if F is an occurrence and β an address, F_β is the relocation of F in β (Definition 4). Recall also that we made the choice not write explicitly the addresses of atoms.

Definition 27. Let $\Delta := \{F_i\}_{1\le i\le n}$ be a set of occurrences. The ***Δ-derivation*** is a $\mu\mathsf{LK}^\infty$ open derivation of conclusion $\Delta \vdash$ obtained by applying successively, and for all $1 \le i \le n$, the F_i-derivations until reaching sequents of the form $C, \odot\Delta \vdash$ where $\overline{C}$ is a constraint. The set of ***successors of*** Δ, denoted $S(\Delta)$, is the set of premises of the Δ-derivation.

Remark 4. Here again, there are many possible Δ-derivations but they differ only by the order where F_i -derivations are applied. They all share the same structure of threads and the same set of premises.

Example 9. Let F be the occurrence of Example 8 and $H = \nu X.\mathfrak{r} \wedge (\odot X)_\delta$. The $\{F, H\}$-derivation is the following:

$$\dfrac{\dfrac{\mathfrak{p}, \mathfrak{r}, (\odot F)_\alpha, (\odot F)_\beta, (\odot H)_\delta \vdash}{\mathfrak{p}, (\odot F)_\alpha, (\odot F)_\beta, H \vdash}\,(H) \qquad \dfrac{\mathfrak{p}, \mathfrak{q}, \mathfrak{r}, (\odot G)_\gamma, (\odot F)_\beta, (\odot H)_\delta \vdash}{\mathfrak{p}, \mathfrak{q}, (\odot G)_\gamma, (\odot F)_\beta, H \vdash}\,(H)}{F, H \vdash}\,(F)$$

We sometimes write a Δ-derivation as a rule named Δ:

$$\frac{\mathfrak{p}, \mathfrak{r}, (\odot F)_\alpha, (\odot F)_\beta, (\odot H)_\delta \vdash \quad \mathfrak{p}, \mathfrak{q}, \mathfrak{r}, (\odot G)_\gamma, (\odot F)_\beta, (\odot H)_\delta \vdash}{F, H \vdash} \; (\{F, H\})$$

Definition 28. We define $\Pi(\Delta)$ to be the $\mu\mathsf{LK}^\infty$ pre-proof of conclusion $\Delta \vdash$, obtained by applying coinductively the following scheme:

$$\Pi(\Delta) \quad = \quad \dfrac{\dfrac{\Pi(\Delta_1)}{C_1, \odot\Delta_1 \vdash}\,(\odot) \quad \ldots \quad \dfrac{\Pi(\Delta_n)}{C_n, \odot\Delta_n \vdash}\,(\odot)}{\Delta \vdash}\,(\Delta)$$

Example 10. Let $\varphi = \mu X.\nu Y.\, \odot\,((\mathfrak{p} \wedge Y) \vee X)$, $\psi = \nu Y.\, \odot\,((\mathfrak{p} \wedge Y) \vee \varphi)$ and $\delta = (\mathfrak{p} \wedge \psi) \vee \varphi$. The derivation $\Pi(\vdash \delta)$ is:

$$\dfrac{\dfrac{\dfrac{(\dagger)}{\delta_{lii} \vdash}}{\mathfrak{p}, (\odot\delta)_{li} \vdash}\,(\odot) \quad \dfrac{\dfrac{(\dagger)}{\delta_{riii} \vdash}}{(\odot\delta)_{rii} \vdash}\,(\odot)}{(\dagger)\;\; \delta_\varepsilon \vdash}\,(\delta_\varepsilon)$$

Notice that $\Pi(\vdash F)$ is not a $\mu\mathsf{LK}^\infty$ proof in general, because it dos not always satisfy the validity condition. This is not a problem, since the goal is to use it only as a support to compute the set of models of F.

Definition 29. Let β be a branch of $\Pi(\vdash\ F)$ and $((C_i, \odot\Delta_i))_{i\in\omega}$ be the sequence of the conclusions of the $\odot$ rules in β. A word $u = (a_i)_{i\in\omega}$ is said to be ***induced*** by β iff $\forall i,\, a_i \models \overline{C_i}$. The branch β is said to be ***accepting*** if it contains no μ-thread. The ***language of*** F, denoted $\mathcal{L}(F)$, is the set of words induced by the accepting branches of $\Pi(\vdash F)$.

Example 11. If $\mathcal{P} = \{\mathfrak{p}\}$ then $\Sigma = \{a, b\}$ where $a = \emptyset$ and $b = \{\mathfrak{p}\}$. The language of δ from Example 10 is $\{a, b\}^\star.b^\omega$.

The use of automata-theoretic vocabulary is due to the fact that we can see an occurrence F as a sort of APW. Indeed, when we put F in the left hand-side of the sequent, the left disjunction rules become branching, and we can see each branch as a non-deterministic choice of a run, while the conjuction rule is non branching and we can see the application of conjuction rule as constructing an alternating run. Hence, every infinite branch of $\Pi(\vdash F)$ can be seen as a run of an alternating automaton. Moreover, the acceptance condition of a branch of $\Pi(\vdash F)$ recalls the parity condition for APW. We will make this remark more precise in the next sections.

Proposition 8 ([12])**.** *For every formula F, $\mathcal{L}(F) = \mathcal{M}(F)$.*

D. From Formulas to APW

In this section, we exploit the intuitions and tools developed in Section III-C to extract from every μ-calculus formula φ an APW $\mathcal{A}_\varphi$. We show not only that φ and $[\mathcal{A}_\varphi]$ are semantically equivalent, but also that $[\mathcal{A}_\varphi] \vdash \varphi$ is provable in $\mu\mathsf{LK}$.

1) APW of a formula: Let us give an outline of the construction of the APW $\mathcal{A}_\varphi$ corresponding to a formula φ. Definition 29 suggests that we can see the FL-suboccurrences of φ as the states of an APW, whose transitions are given by the F-derivations. For instance, the derivation $\Pi(\delta)$ of Example 10 suggests that the set of states of $\mathcal{A}_\delta$ is $\{\delta\}$ and its transition relation is $\{(\delta, a, \delta), (\delta, b, \delta)\}$. To assign a priority to the state of $\mathcal{A}_\delta$, we have to look at the formulas hidden in the δ-derivation. Indeed, what makes the leftmost branch of $\Pi(\delta)$ accepting is the formula ψ hidden in the δ-derivation. Hence we want the priority of δ to be that of ψ, thus to be even. But if we do so, the word a^ω, induced by the rightmost branch of $\Pi(\delta)$, will be accepted also by $\mathcal{A}_\delta$. The rightmost branch of $\Pi(\delta)$ is not accepting because the minimum of its thread is φ. To get an automaton that accepts exactly the models of δ, one should have two copies of the state δ, one with an odd priority corresponding to φ, we denote it δ^φ and the other with an even priority corresponding to ψ, we denote it δ^ψ. More generally, the states of the alternating automaton corresponding to a formula will be pairs of formulas, written as δ^ψ, where δ is morally the state and ψ is a formula containing the priority information.

To generalize this idea, we have to ensure that in an F-derivation, every thread linking F to an occurrence appearing in a premise of this F-derivation admits a minimal formula:

Proposition 9. *Let s be a successor of F and $G \in s$. The thread of the F-derivation, linking F to G, has a minimal formula ψ wrt. the subformula ordering.*

We write $F \overset{s,\psi}{\to} G$ as a shortcut for: "The minimum of the thread linking F to G in the F-derivation is ψ".

To show this result, we need the notion of *straight threads*:

Definition 30. Let F be an occurrence and $t = (F_i)_{1\le i\le k}$ be a thread. For all $1 \le i \le k$, let $n_i = \mathcal{N}_F(F_i)$ (Proposition 1). Recall that n_i are also nodes of $\tau(F)$. The thread t is said to be a ***straight thread*** if the node n_{i+1} is the son of n_i in $\tau(F)$.

It is not difficult to show, by induction on k the following:

Proposition 10. *If t is a straight thread then $\bar{t}$ admits a minimum* w.r.t. $\le$.

Proof of Proposition 9. It suffices to notice that the thread t linking F to G is a straight thread. By Proposition 10 it admits a minimal formula wrt. the subformula ordering. □

The following proposition is traightforward. It shows that the minimum of the thread linking F to a formula occurrence G appearing in one of its successors does not change if we relocate F.

Proposition 11. *Let φ_α be an occurrence and $s \in S(\varphi_\alpha)$. We have:*

- *The sequent s is of the form $s = \{(\varphi_i)_{\alpha.\alpha_i}\}_{0\le i\le n}$.*
- *For every address β, $s' := \{(\varphi_i)_{\beta.\alpha_i}\}_{0\le i\le n} \in S(\varphi_\beta)$,*
- *and $\forall i \le n$ if $\varphi_\alpha \overset{s,\psi}{\to} (\varphi_i)_{\alpha.\alpha_i}$ then $\varphi_\beta \overset{s,\psi}{\to} (\varphi_i)_{\beta.\alpha_i}$.*

We define in the following, for every formula φ, a function p_φ that assigns to every FL-subformula of φ an integer, in a way that respects the subformula ordering and that is compatible with the nature of fixed-point formulas.

Definition 31. Let φ be a formula and let $p_\varphi : FL(\varphi) \to \omega$ be a function such that:

- ψ is a ν-formula if and only if $p_\varphi(\psi)$ is even.

- If $\psi \leq \delta$ then $p_\varphi(\psi) \leq p_\varphi(\delta)$.

The function p_φ is not unique but we can fix one arbitrarily.

The function p_φ will be used to assign priorities to the states of the APW corresponding to the formula φ.

Definition 32. The APW associated to φ, denoted $\mathcal{A}_\varphi$, is the tuple $(\Sigma, Q, q_I, \Delta_\varphi, c)$ where the set of states is $Q = \{\delta^\gamma \mid \delta, \gamma \in FL(\varphi)\}$, the initial state is $q_I = \varphi^\varphi$, the priority function is defined by $c(\delta^\gamma) = p_\varphi(\gamma)$ and the transition relation Δ_φ is defined as follows: $(\psi^\delta, a, E) \in \Delta_\varphi$ iff there is $s = (C, \odot\Delta) \in S(\psi_\varepsilon)$ such that $a \models \overline{C}$ and:

$$\gamma^\sigma \in E \qquad \leftrightarrow \qquad \exists(\odot\gamma_\beta) \in s \text{ such that } \psi_\varepsilon \overset{s,\sigma}{\to} \odot\gamma_\beta$$

Notice that the choice of q_I as φ^φ is arbitrary, one could choose any state of the form φ^δ.

Proposition 12. *We have* $\mathcal{L}(\mathcal{A}_\varphi) = \mathcal{M}(\varphi)$.

We show that the automaton $\mathcal{A}_\varphi$ and the formula φ have the same language ie. $\mathcal{L}(\mathcal{A}_\varphi) = \mathcal{L}(\varphi)$.

We first show that $\mathcal{L}(\varphi) \subseteq \mathcal{L}(\mathcal{A}_\varphi)$. Let $u \in \mathcal{L}(\varphi)$ and let β be an accepting branch of $\Pi(\vdash \varphi)$ that induces u. Let $(s_i)_{i\in\omega}$ be the sequence of conclusions of $\odot$ rules in β. For every $i \in \omega$, s_i is of the from $C_i, \odot\Delta_i$ where $\overline{C_i}$ is a constraint. We construct in the following a run ρ of $\mathcal{A}_\varphi$ over u, level by level, starting from the root. In this construction, we keep the following invariant: for every $i \in \omega$, if $\Delta_i = \{(\delta_j)_{\alpha_j}\}_{1\leq j\leq n}$, then there exists a set of formula $\{\psi_j\}_{1\leq j\leq n}$ such that the set of nodes labels at level i in ρ is $\{\delta_j^{\psi_j}\}_{1\leq j\leq n}$.

Level 1: We label the root of ρ by φ^φ. The sequent Δ_1 is of the form $\{(\delta_j)_{\alpha_j}\}_{1\leq j\leq n}$. For every $j \leq n$, there exists ψ_j such that $\varphi_\varepsilon \overset{s_1,\psi_j}{\to} (\odot\delta_j)_{\alpha_j}$, we add then to the root a son labelled $\delta_j^{\psi_j}$. Our invariant is satisfied for level 1.

Level i $\to$ Level i+1: If $\Delta_i = ((\delta_k)_{\alpha_k})_{1\leq k\leq n}$ then s_{i+1} is of the form $s_{i+1} = \bigcup_{1\leq k\leq n} v_k$, where $v_k \in S((\delta_k)_{\alpha_k})$, for every $k \leq n$. Let w be a node of level i, labelled $\delta_k^{\psi_k}$. For every occurrence $(\odot\gamma)_\alpha$ in v_k, there exists a formula ψ such that $(\delta_k)_{\alpha_k} \overset{v_k,\psi}{\to} (\odot\gamma)_\alpha$, we add to w a son labelled γ^ψ. Here again, our invariant is satisfied for level $i+1$.

The run ρ is valid. Indeed, let $b = (\delta_i^{\psi_i})_{i\in\omega}$ be a branch of ρ. We will show that b is valid using the validity of the branch β of $\Pi(\vdash \varphi)$. By construction one has that $\delta_0^{\psi_0} = \varphi^\varphi$ and there exists a sequence of addresses $(\alpha_i)_{i\in\omega}$ such that for every $i \in \omega$ we have $(\delta_i)_{\alpha_i} \in \Delta_i$. Moreover, the thread linking $(\delta_i)_{\alpha_i}$ to $(\delta_{i+1})_{\alpha_{i+1}}$ in $\Pi(\vdash \varphi)$, that we denote t_i, satisfies the following property: $min(\overline{t_i}) = \psi_{i+1}$. Let $t = t_0 t_1 \dots$ be the thread of $\Pi(\vdash \varphi)$ obtained by the concatenation of the threads $\{t_i\}_{i\in\omega}$. Let $k, l \in \omega$ such that the subthread $t_k \dots t_l$ of t contains all the elements of $Inf(\bar{t})$. One has also that the subsequence $\delta_k^{\psi_k}, \dots, \delta_{l+1}^{\psi_{l+1}}$ of b contains all the elements of $Inf(b)$. One has:

$$\begin{array}{rcl} min(Inf(\bar{t})) & = & min(\overline{t_k} \dots \overline{t_l}) \\ & = & min(min(\overline{t_i}))_{k\leq i\leq l} \\ & = & min(\psi_i)_{k\leq i\leq l} \end{array}$$

Since the branch β is accepting, $min(\psi_i)_{k\leq i\leq l}$ is a ν-formula. Moreover, if c is the priority function of $\mathcal{A}_\varphi$ and p_φ the auxiliary function used to define it (Definition 31), then we have:

$$\begin{array}{rcl} min(c(Inf(b))) & = & min(c(\delta_k^{\psi_k}), \dots, c(\delta_{l+1}^{\psi_{l+1}})) \\ & = & min(p_\varphi(\psi_k), \dots, p_\varphi(\psi_{l+1})) \\ & \overset{\dagger}{=} & p_\varphi(min(\psi_i)_{k\leq i\leq l}) \end{array}$$

($\dagger$) Since the function p_φ is monotonic

Since $min(\psi_i)_{k\leq i\leq l}$ is a ν-formula, one has that $p_\varphi(min(\psi_i)_{k\leq i\leq l})$ is even, hence the branch b is valid. We conclude that ρ is valid and $u \in \mathcal{L}(\mathcal{A}_\varphi)$.

We show the other direction $\mathcal{L}(\mathcal{A}_\varphi) \subseteq \mathcal{L}(\varphi)$ using a similar argument: to every valid run of $\mathcal{A}_\varphi$ over a word u, we can find an accepting branch of $\Pi(\varphi)$ that induces u.

By proposition 8, $\mathcal{L}(\varphi) = \mathcal{M}(\varphi)$, which concludes the proof.

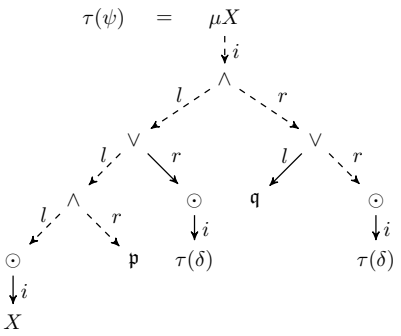
2) $[\mathcal{A}_\varphi] \vdash \varphi$ in μLK: We show that there is a proof of $[\mathcal{A}_\varphi] \vdash \varphi$ in μLK. The idea is to construct a thin proof of this sequent in $\mu\mathsf{LK}^\omega$, and using Proposition 3, to transform it into a proof in μLK. Since φ^φ is the initial state of $\mathcal{A}_\varphi$, we have that $[\mathcal{A}_\varphi]_\varepsilon = [\![\varphi^\varphi]\!]^\emptyset$, hence our goal is to show that the sequent $[\![\varphi^\varphi]\!]^\emptyset \vdash \varphi_\varepsilon$ has a thin $\mu\mathsf{LK}^\omega$ proof. To get this result, we need to generalize our statement and to look for proofs of sequents having the form $[\![\psi^\delta]\!]^\Theta \vdash \psi_\alpha$, where ψ^δ is a state of $\mathcal{A}_\varphi$, Θ a run-section and α an address. Using Proposition 6, we decompose the left-hand side of such sequents and get the following derivation:

$$\frac{\{[a], \{\odot[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)}\}_{\gamma^\sigma\in E} \vdash \psi_\alpha\}_{t=(\psi^\delta,a,E)\in\Delta_\varphi}}{[\![\psi^\delta]\!]^\Theta \vdash \psi_\alpha}$$

Each premise of this derivation corresponds to a transition in Δ_φ, and by construction of Δ_φ, every such transition corresponds to a successor $s \in S(\psi_\varepsilon)$. To get a proof of such premise which is thin, we have to apply weakenings at some points, so that when unfolding the fixed points, their context will be empty. The choice of these weakenings will be guided by this successor s in a sense that we will clarify in the following. But first, let us look closely into the notion of the successor of a formula.

If s is the successor of ψ_ε, then for every $F \in s$, the address of F induces a path in the tree of ψ. When we collect all the paths induced by the formula occurrences of s, we get a subtree $\tau(s)$ of $\tau(\psi)$, as illustrated by the following example.

Example 12. Let $\psi = \mu X.((\odot X \wedge \mathfrak{p}) \vee \odot\delta) \wedge (\mathfrak{q} \vee \odot\delta)$ where $\delta = \nu Y. \odot Y$. Let $s = \mathfrak{p}, (\odot\psi)_\alpha, (\odot\delta)_\beta$ be the successor of ψ_ε where $\alpha = illl$ and $\beta = irr$. The paths of $\tau(\psi)$ induced by the formula occurrences of s form a tree $\tau(s)$ indicated by the dashed lines below:



Any node n of $\tau(s)$ satisfies the following properties:

- If n is labelled $\odot$, the son of n does not belong to $\tau(s)$.
- If n is labelled $\wedge$, the two sons of n belong also to $\tau(s)$.
- If n is labelled $\vee$, exactly one son of n belongs to $\tau(s)$.

A successor $s \in S(F)$ induces a function that chooses, for every suboccurrence of F which is a disjunction, one of its disjuncts; we call this function C_s.

Definition 33. Let $s = \{(\delta_i)_{\alpha_i}\}_{1 \leq i \leq n}$ be the successor of an occurrence $F = \varphi_\alpha$. For all α_i, there is p_i st. $\alpha_i = \alpha.p_i$.

The ***choice*** of s is a partial function on occurrences defined as follows. For all i, and for every address $a \sqsubseteq p_i$ such that the node of $\tau(\varphi)$ at the address a is labelled $\vee$, we set:

$$\begin{array}{rcll} C_s((\varphi \vee \psi)_a) & = & \varphi_{a.l} & \text{if } a.l \sqsubseteq p_i, \\ & = & \psi_{a.r} & \text{otherwise.} \end{array}$$

Example 13. In Example 12, we have: $C_s((\odot\psi \wedge \mathfrak{p}) \vee \odot\delta)_{il}) = (\odot X \wedge \mathfrak{p})_{ill}$ and $C_s((\mathfrak{q} \vee \odot\delta)_{ir}) = (\odot\delta)_{irr}$.

Using this function, we define the derivation $\Pi_s(\Gamma \vdash F)$. In this derivation of conclusion $\Gamma \vdash F$, we apply only right rules, and whenever we meet a right formula which is a disjunction, we apply $(\vee_\mathsf{r})$ rule followed immediately by a weakening, that keeps only the formula chosen by C_s. Doing so, we obtain a derivation which is thin, since we keep the invariant that we have exactly one right formula along the derivation.

Definition 34. Let F be an occurrence, $s \in S(F)$ and Γ be a set of occurrences. The derivation $\Pi_s(\Gamma \vdash F)$ is the $\mu\mathsf{LK}^\omega$ derivation of conclusion $\Gamma \vdash F$ obtained by applying only the following rules:

$$\frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G}\ (\wedge_\mathsf{r}) \qquad \frac{\Gamma \vdash F[\sigma X.F/X]}{\Gamma \vdash \sigma X.F}\ (\sigma_\mathsf{r}) \qquad \frac{\Gamma \vdash C_s(F \vee G)}{\dfrac{\Gamma \vdash F, G}{\Gamma \vdash F \vee G}\ (\vee_\mathsf{r})}\ (\mathsf{W}_\mathsf{r})$$

Example 14. We show in the following the derivation $\Pi_s(\vdash \psi_\varepsilon)$, where s and ψ are defined in Example 12.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash (\odot\psi)_{ill} \quad \vdash \mathfrak{p}}{\vdash (\odot\psi \wedge \mathfrak{p})_{ill}}\ (\wedge_\mathsf{r})}{\vdash ((\odot\psi \wedge \mathfrak{p}) \vee \odot\delta))_{il}}\ (\vee_\mathsf{r}),(\mathsf{W}_\mathsf{r}) \qquad \dfrac{\dfrac{\vdash (\odot\delta)_{irr}}{\vdash (\mathfrak{q} \vee \odot\delta)_{ir}}\ (\vee_\mathsf{r}),(\mathsf{W}_\mathsf{r})}{\vdash ((\odot\psi \wedge \mathfrak{p}) \vee \odot\delta) \wedge (\mathfrak{q} \vee \odot\delta))_i}\ (\wedge_\mathsf{r})}{\vdash \psi_\varepsilon}\ (\mu_\mathsf{r})$$

The following proposition is traightforward.

Proposition 13. *Let F be an occurrence, $s \in S(F)$ and Γ be a set of occurrences.*

- *$\Pi_s(\Gamma \vdash F)$ is a thin derivation.*
- *The premises of $\Pi_s(\Gamma \vdash F)$ are $\{\Gamma \vdash G \mid G \in s\}$.*
- *If $\Gamma \vdash G$ is a premise of $\Pi_s(\Gamma \vdash F)$ and if $F \overset{s,\psi}{\to} G$ then the minimum of the thread linking F to G in $\Pi_s(\Gamma \vdash F)$ is ψ.*

Proposition 14. *Let φ be a formula and ψ^δ be a state of $\mathcal{A}_\varphi$. The following rule is derivable in $\mu\mathsf{LK}^\omega$ using a thin derivation:*

$$\frac{\{[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)} \vdash \gamma_{\alpha.\beta}\}_{s \in S(\psi_\varepsilon), \gamma_\beta \in s, \psi_\varepsilon \overset{s,\sigma}{\to} \odot(\gamma)_\beta}}{[\![\psi^\delta]\!]^\Theta \vdash \psi_\alpha}\ (\circ)$$

And the minimum of the thread linking ψ_α to $\gamma_{\alpha.\beta}$ is σ.

Proof. We already noticed that the following rule is derivable:

$$\frac{\{[a], \{\odot[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)}\}_{\gamma^\sigma \in E} \vdash \psi_\alpha\}_{t=(\psi^\delta,a,E) \in \Delta_\varphi}}{[\![\psi^\delta]\!]^\Theta \vdash \psi_\alpha}$$

By definition of Δ_φ, we have that $(\psi^\delta, a, E) \in \Delta_\varphi$ iff there is $s = (C, \odot\Delta) \in S(\psi_\varepsilon)$ s.t. $a \models \overline{C}$ and if the following holds:

$$\gamma^\sigma \in E \quad \leftrightarrow \quad \exists(\odot\gamma_\beta) \in s \text{ such that } \psi_\varepsilon \overset{s,\sigma}{\to} \odot\gamma_\beta$$

To justify a premise corresponding to a transition $t = (\psi^\delta, a, E)$, we set $\Gamma = [a], \{\odot[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)}\}_{\gamma^\sigma \in E}$ and apply the derivation $\Pi_s(\Gamma \vdash \psi_\alpha)$:

$$\frac{\{\Gamma \vdash \mathfrak{p}_\eta\}_{\mathfrak{p}_\eta \in C} \qquad \{\Gamma \vdash \odot(\gamma)_{\alpha.\beta}\}_{\odot(\gamma)_\beta \in s}}{\Gamma \vdash \psi_\alpha}\ (\Pi_s(\Gamma \vdash \psi_\alpha))$$

To justify a premise $\Gamma \vdash \mathfrak{p}_\eta$, we notice that since $a \models \overline{C}$, we have $\overline{C} \subseteq a$, thus $[a]$ is of the form $[a] = \mathfrak{p} \wedge G$. We then apply the following derivation:

$$\dfrac{\dfrac{}{[a] \vdash F}\ (\wedge_\mathsf{l}),\ (\mathsf{Ax})}{\Gamma \vdash F}\ (\mathsf{W}_\mathsf{l})$$

Now we want to justify a premise $\Gamma \vdash \odot(\gamma)_{\alpha.\beta}$. If $\psi_\varepsilon \overset{s,\sigma}{\to} \gamma_\beta$, then $\gamma^\sigma \in E$, we can thus apply the following:

$$\dfrac{\dfrac{[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)} \vdash \gamma_{\alpha.\beta}}{\odot[\![\gamma^\sigma]\!]^{\Theta,(\psi^\delta,t)} \vdash \odot(\gamma)_{\alpha.\beta}}\ (\odot)}{\Gamma \vdash \odot(\gamma)_{\alpha.\beta}}\ (\mathsf{W}_\mathsf{l})$$

The obtained derivation is thin and by Proposition 13, the minimum of the thread linking ψ_α to $\gamma_{\alpha.\beta}$ is σ. □

The premises of the derived rule $(\circ)$ have the same shape as its conclusion. If we iterate this derivation starting from $[\mathcal{A}_\varphi] \vdash \varphi$ (which is $[\![\varphi^\varphi]\!]^\emptyset \vdash \varphi_\varepsilon$) we obtain a thin $\mu\mathsf{LK}^\omega$ pre-proof.

We show that this pre-proof is actually a proof. Using Proposition 3, we can transform this thin proof into a $\mu\mathsf{LK}$ proof of $[\mathcal{A}_\varphi] \vdash \varphi$.

Theorem 1. *We can construct a $\mu\mathsf{LK}$ proof of $[\mathcal{A}_\varphi] \vdash \varphi$.*

Proof. Let c be the priority function of $\mathcal{A}_\varphi$ and p_φ (Definition 31) the function used to define c. Recall that $[\mathcal{A}_\varphi]_\varepsilon =$

$[\![\varphi^{\varphi}]\!]^{\emptyset}$. Let π be the $\mu\mathsf{LK}^{\infty}$ pre-proof obtained by applying coinductively the derivation $(\circ)$ of Proposition 14. It is not difficult to see that π is a thin $\mu\mathsf{LK}^{\omega}$ pre-proof. We show now that π is $\mu\mathsf{LK}^{\omega}$ proof, *i.e.*, that is satisfies the validity condition. Let γ be an infinite branch of π and $([\![\psi_i^{\delta_i}]\!]^{\Theta_i} \vdash (\psi_i)_{\alpha_i})_{i\in\omega}$ be the sequence made of the conclusions of the derivation $\circ$. In γ, there is exactly one right thread t_r and one left thread t_l. The thread t_l is a thread of $[\mathcal{A}]$. Let $\rho(t_l)$ be the run-branch of t_l (Definition 24). Let $v = (\delta_i)_{i\in\omega}$. we have that $\psi^{\delta} \in Inf(\rho(t_l)) \Leftrightarrow \delta \in Inf(v)$. Hence we have that:

$$\begin{array}{rcl} & & min(\{c(\psi^{\delta}) \mid \psi^{\delta} \in Inf(\rho(t_l))\}) \\ (\star) & = & min(\{p_{\varphi}(\delta) \mid \psi^{\delta} \in Inf(\rho(t_l))\}) \\ & = & min(\{p_{\varphi}(\delta) \mid \delta \in Inf(v)\}) \end{array}$$

In another hand, since the minimum of the thread linking ψ_i to ψ_{i+1} in γ is δ_i, we have:

$$\begin{array}{rcl} & & min(\{\psi \mid \psi \in Inf(\overline{t_r})\}) \\ (\dagger) & = & min(\{\delta \mid \delta \in Inf(v)\}) \end{array}$$

There is two possible cases: either t_l is a μ-thread, thus γ is valid; or t_l is a ν-thread then by Proposition 7 the run-branch $\rho(t_l)$ is accepting, then, by $(\star)$, $min(\{p_{\varphi}(\delta) \mid \delta \in Inf(v)\})$ is even. By definition of p_{φ}, $min(\{\delta \mid \delta \in Inf(v)\})$ is a ν-formula, thus by $(\dagger)$, t_r is a ν-thread, which means that γ is valid. □

Theorem 1 is the step I of our proof of completeness. Actually, we can show also that $\varphi \vdash_{\mu\mathsf{LK}} [\mathcal{A}_{\varphi}]$ in the same way.

IV. Reflecting automata equivalences in the logic

It is well known that APW, NPW and NBW are three equivalent models of automata This result is usually shown by exhibiting automata transformations that convert an automaton of a given model into an automaton in the other models, by preserving the language. In this section, we recall the automata transformations that will be used for the proof of completeness, and show that the transformations provide, via the encoding, provably equivalent formulas.

A. From APW to NPW

We recall the construction that transforms an APW $\mathcal{A}$ into an NPW $\mathcal{P}$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A})$. In a second step, we show that the sequent $[\mathcal{P}] \vdash [\mathcal{A}]$ in provable $\mu\mathsf{LK}$.

1) Non-determinization: We recall in this section the operation of *non-determinization*, that transforms an APW into a NPW with the same language. In [13] this construction is presented for automata over trees, we specialize it here for automata over words. We fix an APW $\mathcal{A} = (\Sigma, Q, \Delta, q_I, c)$.

The key ingredient to non-determinize APW is to simplify their runs. Runs of APW may be *non-uniform* in the sense that two nodes at the same level and labelled with the same state, may have different sets of successors. For instance, in the run ρ_2 of Example 4, the two nodes at level 2 and labelled p have as sets of successors $\{p\}$ and $\{p, q\}$ respectively. On the contrary, we call *uniform* a run such that: all nodes at the same level and with the same labels have the same set of successors. Uniform runs are sufficient to compute the languages of APW [13]: $u \in \mathcal{L}(A)$ iff there is a valid uniform run of $\mathcal{A}$ over u.

In a uniform run, every level can be described by a function that maps a state to the set of its successors: uniform runs can be described by sequences of such functions. For instance, the run ρ_1 of Example 4 can be seen as a sequence $f_0 f_1 f_2 \ldots$ where $f_0(p) = \{p, q\}$, $f_1(p) = \{p, q\}, f_1(q) = q$, $f_2(p) = \{p, q\}, f_2(q) = q$, etc. We call these functions ***choice functions***. The interest of using uniform runs instead of arbitrary runs is that we went from runs having a tree structure to runs with a structure of words (sequences of choice functions), that we will be able to synchronize with Σ-words.

Definition 35. A ***choice function*** is a function $\sigma : Q \to 2^Q$. The set of choice functions is denoted $\mathcal{S}$. Notice that $\mathcal{S}$ is finite.The ***auxiliary alphabet*** of $\mathcal{A}$ is $\Sigma^{aux} = \{(a, \sigma) \in \Sigma \times \mathcal{S} \mid (p, a, \sigma(p)) \in \Delta\}$.

Two sequences can be extracted from a word U over Σ^{aux}: the first is a word u over Σ obtained by projection on the first components of U, and the second is a sequence of choice functions obtained by projection on the second components of U. This latter sequence describes a uniform run of $\mathcal{A}$ over u.

Definition 36. Let $U = ((a_i, \sigma_i))_{i\in\omega} \in (\Sigma^{aux})^{\omega}$. The ***run associated*** to U is a labelled tree such that: the root is labelled q_I; and for every node v at level n, if q is the label of v then each son of v is labelled with one of the sates of $\sigma_n(q)$. Notice that τ is a uniform run of U over the word $(a_i)_{i\in\omega}$.

Definition 37. The language $L \subseteq (\Sigma^{aux})^{\omega}$ is the ***auxiliary language*** of $\mathcal{A}$ iff $\forall U \in L$, the run associated to U is accepting (*w.r.t.* the acceptance condition of $\mathcal{A}$). We denoted it by $\mathcal{L}^{aux}(\mathcal{A})$.

The idea of the non-determinization is to exhibit a deterministic parity automaton (DPW) recognizing the auxiliary language of $\mathcal{A}$. Intuitively, this is possible since all the information of alternation is explicit in the Σ^{aux}-words, hence we do not need an alternating automaton to recognize it. One this is done, we erase the choice functions from the transition structure of this automaton, to get an NPW recognizing the language of $\mathcal{A}$.

Proposition 15. *For very APW we can construct effectively a NPW recognizing the same language.*

Proof sketch. Let $\mathcal{A} = (\Sigma, Q, \Delta, q_I, c)$ be an APW. We construct a NPW recognizing the language of $\mathcal{A}$ in two steps.

I. Construct a DPW $\mathcal{D} = (\Sigma^{aux}, Q^d, q_I^d, \Delta^d, c^d)$ such that $\mathcal{L}(\mathcal{D}) = \mathcal{L}^{aux}(\mathcal{A})$. This is done in 4 steps:
a) Construct an APW recognizing $\mathcal{L}^{aux}(\mathcal{A})$: Let $\mathcal{A}_1 = (\Gamma, Q, \Delta_1, q_I, c)$ be the APW where:

$$\Delta_1 = \{(p, (a, \sigma), E) \mid (p, a, E) \in \Delta \text{ and } \sigma(p) = E\}$$

The language of $\mathcal{A}_1$ is $\mathcal{L}^{aux}(\mathcal{A})$. Notice that for every Σ^{aux}-word, there is only one possible run over $\mathcal{A}_1$.

b) Construct a NPW recognizing the complement of $\mathcal{L}^{aux}(\mathcal{A})$: Let $\mathcal{A}_2 = (\Gamma, Q, \Delta_2, q_I, c_2)$ be the APW where $c_2(q) = c(q) + 1$ and:

$$\Delta_2 = \{(p, A, q) \mid \exists E, (p, A, E \cup \{q\}) \in \Delta_1\}.$$

The language of $\mathcal{A}_2$ is the complement of $\mathcal{L}^{aux}(\mathcal{A})$. Indeed, R is a run of $\mathcal{A}_2$ over U iff R is a path in the unique run of $\mathcal{A}_1$ over U. Since we shifted the priorities in $\mathcal{A}_2$, R is a valid run of $\mathcal{A}_2$ over U iff R seen as a path in the run of $\mathcal{A}_1$ over U is not valid. Thus, $U \in \mathcal{L}(\mathcal{A}_2)$ iff $U \notin \mathcal{L}(\mathcal{A}_1)$.

c) A DPW recognizing the complement of $\mathcal{L}^{aux}(\mathcal{A})$: Determinize $\mathcal{A}_2$ to get a DPW $\mathcal{A}_3 = (\Gamma, Q_3, \Delta_3, p_I, c_3)$, using a determinisation technique (Safra construction for example).

d) A DPW recognizing $\mathcal{L}^{aux}(\mathcal{A})$: Let $\mathcal{A}_4 = (\Gamma, Q_3, \Delta_3, p_I, c_4)$ where $c_4(q) = c_3(q) + 1$. One has $\mathcal{L}(\mathcal{A}_4) = \mathcal{L}^{aux}(\mathcal{A})$.

II. Let $\mathcal{P} = (\Sigma, Q^d, q_I^d, \Delta', c^d)$ where $\Delta' = \{(d, a, d') \mid \exists \sigma, (d, (a, \sigma), d') \in \Delta^d\}$. One has $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A})$.

□

2) $[\mathcal{P}] \vdash_{\mu\mathsf{LK}} [\mathcal{A}]$: Let $\mathcal{A}$ be an APW automaton and $\mathcal{P}$ be the NPW obtained in the proof of Proposition 15. In this section we show that $[\mathcal{P}] \vdash [\mathcal{A}]$ has a proof in $\mu\mathsf{LK}$. For that purpose, we build a thin $\mu\mathsf{LK}^\omega$ proof of this sequent, which gives us a $\mu\mathsf{LK}$ proof by Proposition 3. We use in this section the notations of the proof sketch of Proposition 15. Since q_I and q_I^d are the initial states of $\mathcal{A}$ and $\mathcal{P}$ respectively, we have that $[\mathcal{A}]_\varepsilon = [\![q_I]\!]^\emptyset$ and $[\mathcal{P}]_\varepsilon = [\![q_I^d]\!]^\emptyset$. Our goal is then to show that $[\![q_I^d]\!]^\emptyset \vdash [\![q_I]\!]^\emptyset$ is provable with a thin $\mu\mathsf{LK}^\omega$ proof. To do so, we need to generalize this statement, and to look for proofs of sequents having the form $[\![d]\!]^\Theta \vdash [\![q]\!]^\Upsilon$, where d and Θ (resp. q and Υ) are a state and a run-section for $\mathcal{P}$ (resp. $\mathcal{A}$). We show the following lemma:

Lemma 1. *The following rule is derivable in $\mu\mathsf{LK}^\omega$ using a thin derivation:*

$$\frac{\{[\![d']\!]^{\Theta,(d,t)} \vdash [\![q']\!]^{\Upsilon,(q,T)}\}_{(d,(a,\sigma),d') \in \Delta^d, q' \in \sigma(q)}}{[\![d]\!]^\Theta \vdash [\![q]\!]^\Upsilon} \ (\rightarrow)$$

Where $t = (d, a, d')$ and $T = (q, a, \sigma(q))$.

Proof. We have $t = (d, a, d') \in \Delta'$ iff $\exists (a, \sigma) \in \Sigma^{aux}$ such that $(d, (a, \sigma), d') \in \Delta^d$. Using Proposition 6, we can derive the following, where $t = (d, a, d')$:

$$\frac{\{[a], \odot[\![d']\!]^{\Theta,(d,t)} \vdash [\![q]\!]^\Upsilon\}_{(q,(a,\sigma),d') \in \Delta^d}}{[\![d]\!]^\Theta \vdash [\![q]\!]^\Upsilon}$$

Now we have to justify every premise of this derivation corresponding to a transition $(q, (a, \sigma), d') \in \Delta^d$. By Proposition 5:

$$[\![q]\!]^\Upsilon \quad \rightarrow \bigvee_{a \in \Sigma, (q,a,E) \in \Delta} [a] \wedge \bigwedge_{q' \in E} \odot [\![q']\!]^{\Upsilon,q}$$

Since $(a, \sigma) \in \Sigma^{aux}$, then $T := (q, a, \sigma(q)) \in \Delta$. If we set $\Gamma := [a], \odot[\![d']\!]^{\Theta,(d,t)}$, we can the derive the following:

$$\dfrac{\left\{\dfrac{[\![d']\!]^{\Theta,(d,t)} \vdash [\![q']\!]^{\Upsilon,(q,T)}}{\Gamma \vdash \odot[\![q']\!]^{\Upsilon,(q,T)}}\ (\odot)\right\}_{q' \in \sigma(q)} \quad \dfrac{}{\Gamma \vdash [a]}\ (\mathsf{Ax})}{\dfrac{\Gamma \vdash [a] \wedge \bigwedge\limits_{q' \in \sigma(q)} \odot [\![q']\!]^{\Upsilon,(q,T)}}{\Gamma \vdash [\![q]\!]^\Upsilon}\ (\sigma_r),\ (\vee_r),(\mathsf{W}_r)}\ (\wedge_r)$$

□

The premisses of the derivation $(\rightarrow)$ have the same shape as its conclusion. If we iterate coinductively $(\rightarrow)$ starting from $[\mathcal{P}] \vdash [\mathcal{A}]$, which is $[\![q_I^d]\!]^\emptyset \vdash [\![q_I]\!]^\emptyset$, we get a thin $\mu\mathsf{LK}^\omega$ pre-proof. We show that this pre-proof is actually a $\mu\mathsf{LK}^\omega$ proof.

Proposition 16. *Let π be the $\mu\mathsf{LK}^\omega$ pre-proof of conclusion $[\mathcal{P}] \vdash [\mathcal{A}]$ obtained by applying coinductively the rule $(\rightarrow)$. We have that π is a $\mu\mathsf{LK}^\omega$ proof.*

Proof. Let β be a branch of π and $([\![d_i]\!]^{\Theta_i} \vdash [\![q_i]\!]^{\Upsilon_i})_{i \in \omega}$ be the sequence made of the conclusions of the derivation $(\rightarrow)$ in β. The branch β has only one infinite right thread t_r and one infinite left thread t_l. The run-branch of t_l (Definition 24) is $\rho_l = (d_i)_{i \in \omega}$ and the run-branch of t_r is $\rho_r = (q_i)_{i \in \omega}$. For all $i \in \omega$, $\exists (a_i, \sigma_i) \in \Sigma^{aux}$ such that $(d_i, (a_i, \sigma_i), d_{i+1}) \in \Delta^d$, we set $U = ((a_i, \sigma_i))_{i \in \omega}$ and $u = (a_i)_{i \in \omega}$. The sequence ρ_l is a run of $\mathcal{P}$ over u; and it is also the run of the DPW $\mathcal{D}$ (proof of Proposition 15) over U, since $\mathcal{P}$ and $\mathcal{D}$ have the same set of states. Let R be the run of $\mathcal{A}$ associated to the Σ^{aux} word U. The sequence ρ_r is a branch of R. There are two possible cases:

- If $U \notin \mathcal{L}^{aux}(\mathcal{A})$. Then the run ρ_l of $\mathcal{D}$ is not accepting. Hence ρ_l seen as a run of $\mathcal{P}$ is also not accepting, since $\mathcal{P}$ and $\mathcal{D}$ have the same priority function. By Proposition 7, t_l is a μ-thread, hence β is valid.
- If $U \in \mathcal{L}^{aux}(\mathcal{A})$. Then the run R associated to U is accepting. In particular, its branch ρ_r is accepting. By Proposition 7, t_r is a ν-thread, hence β is valid. □

Using Proposition 3, we get the following theorem, which is step II of our completeness proof. $[\mathcal{A}] \vdash_{\mu\mathsf{LK}} [\mathcal{P}]$ can be shown in the same way.

Theorem 2. *One can construct a $\mu\mathsf{LK}$ proof of $[\mathcal{P}] \vdash [\mathcal{A}]$.*

B. From NPW to NBW

We recall the construction that transforms an NPW $\mathcal{P}$ into an NBW $\mathcal{B}$ such that $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{P})$. In a second step, we show that the sequent $[\mathcal{B}] \vdash [\mathcal{P}]$ is provable in $\mu\mathsf{LK}$.

1) Parity simplification:

Proposition 17. *For very NPW we can construct effectively a NBW recognizing the same language.*

Proof. Let $\mathcal{P} = (\Sigma, Q, \Delta, q_I, c)$ be a NPW and let Q_{ev} be the set of its even states. The idea is to create for every even state p, a copy of the automaton $\mathcal{P}$ where the state p will be accepting and where all the states with smaller priority will be dropped. We keep also a copy of $\mathcal{P}$ where no state is accepting. A run will stay in this copy for some time and will choose one of the copies where an even state is accepting. Formally, let

$\mathcal{B} = (\Sigma, Q_\bot \cup \bigcup_{p\in Q_{ev}} Q_p, (q_I,\bot), \Delta_\bot \cup \Delta_t \cup \bigcup_{p\in Q_{ev}} \Delta_p, F)$ be the NBW such that $\forall i \in \{\bot\} \cup Q_{ev}$, $Q_i = \{(q,i) \mid q \in Q\}$. The relations $\Delta_\bot, \Delta_t$ and Δ_p where $p \in Q_{ev}$ are defined by:

$$\begin{aligned}
\Delta_\bot &= \{((q,\bot),a,(r,\bot)) \mid (q,a,r) \in \Delta\}\\
\Delta_t &= \{((q,\bot),a,(r,r)) \mid (q,a,r) \in \Delta \text{ and } r \in Q_{ev}\}\\
\Delta_p &= \{((q,p),a,(r,p)) \mid (q,a,r) \in \Delta \text{ and } c(q),c(r) \geq c(p)\}
\end{aligned}$$

The set of accepting states is $F = \{(p,p) \mid p \in Q_{ev}\}$

The states $Q_\bot$ and the relation $\Delta_\bot$ correspond to the copy of $\mathcal{P}$ without any accepting state, and for every $p \in Q_{ev}$, the states Q_p and the relation Δ_p correspond to the copy where p is accepting. The relation Δ_t serves as a transition between the non-accepting copy and the other copies. It is easy to show that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{B})$. □

2) $[\mathcal{B}] \vdash_{\mu\mathsf{LK}} [\mathcal{P}]$*:* Let $\mathcal{P}$ be an NPW and $\mathcal{B}$ be the NBW obtained in the proof of Proposition 17. We show that $[\mathcal{B}] \vdash [\mathcal{P}]$ has a proof in $\mu\mathsf{LK}$. For that purpose, we build a thin $\mu\mathsf{LK}^\omega$ proof of this sequent, which gives us a $\mu\mathsf{LK}$ proof by Proposition 3. We use the notations introduced in the proof of Proposition 17. Since q_I and $(q_I,\bot)$ are the initial states of $\mathcal{P}$ and $\mathcal{B}$ respectively, we have that $[\mathcal{P}] = [\![q_I]\!]^\emptyset$ and $[\mathcal{B}] = [\![(q_I,\bot)]\!]^\emptyset$. Our goal is then to show that $[\![(q_I,\bot)]\!]^\emptyset \vdash [\![q_I]\!]^\emptyset$ has a thin $\mu\mathsf{LK}^\omega$ proof. To do so, we need to generalize this statement, and to look for proofs of sequents having the form $[\![(q,r)^\Theta]\!] \vdash [\![q]\!]^\Upsilon$ where $q \in Q$, $r \in Q_{ev} \cup \{\bot\}$ and Θ, Υ are environments of $\mathcal{B}$ and $\mathcal{P}$ respectively. We show in the following lemma.

Lemma 2. *The following rule is derivable in* $\mu\mathsf{LK}^\omega$ *using a thin derivation, where* $t = (q,a,q')$*:*

$$\frac{\{[\![(q',r')]\!]^{\Theta,((q,r),T)} \vdash [\![q']\!]^{\Upsilon,(q,t)}\}_{T=((q,r),a,(q',r'))\in\Delta_\mathcal{B}}}{[\![(q,r)]\!]^\Theta \vdash [\![q]\!]^\Upsilon}\ (\hookrightarrow)$$

Proof. We start by decomposing the left-hand side of the sequent using Proposition 6:

$$\frac{\{[a], \odot[\![(q',r')]\!]^{\Theta,((q,r),T)} \vdash [\![q]\!]^\Upsilon\}_{T=((q,r),a,(q',r'))\in\Delta_\mathcal{B}}}{[\![(q,r)]\!]^\Theta \vdash [\![q]\!]^\Upsilon}$$

Now we have to justify every premise of this derivation. Let $T = ((q,r),a,(q',r')) \in \Delta_\mathcal{B}$. By construction of $\Delta_\mathcal{B}$, we have that $t = (q,a,q') \in \Delta$. We have also that:

$$[\![q]\!]^\Upsilon \to \bigvee_{t=(q,a,q')\in\Delta} [a] \wedge \odot[\![q']\!]^{\Upsilon,(q,t)}$$

We set $\Gamma = [a], \odot[\![(q',r')]\!]^{\Theta,((q,r),T)}$.

$$\dfrac{\dfrac{}{\Gamma \vdash [a]}\ (\mathsf{Ax}) \qquad \dfrac{[\![(q',r')]\!]^{\Theta,((q,r),T)} \vdash [\![q']\!]^{\Upsilon,(q,t)}}{\Gamma \vdash \odot[\![q']\!]^{\Upsilon,(q,t)}}\ (\odot)}{\dfrac{\Gamma \vdash [a] \wedge \odot[\![q']\!]^{\Upsilon,(q,t)}}{\Gamma \vdash [\![q]\!]^\Upsilon}\ (\sigma_r),\ (\vee_r),\ (\mathsf{W}_r)}\ (\wedge_r)$$

□

The premises of the derivation $\hookrightarrow$ have the same shape as its conclusion. If we iterate coinductively $\hookrightarrow$ starting from $[\mathcal{B}] \vdash [\mathcal{P}]$, which is $[\![(q_I,\bot)]\!]^\emptyset \vdash [\![q_I]\!]^\emptyset$, we get a thin $\mu\mathsf{LK}^\omega$ pre-proof. We show that this pre-proof is a $\mu\mathsf{LK}^\omega$ proof.

Proposition 18. *Let* θ *be the* $\mu\mathsf{LK}^\omega$ *pre-proof obtained by applying coinductively* $\hookrightarrow$*. One has that* θ *is* $\mu\mathsf{LK}^\omega$ *proof.*

Proof. Let β be a branch of θ and $([\![(q_i,r_i)]\!]^{\Theta_i} \vdash [\![q_i]\!]^{\Upsilon_i})_{i\in\omega}$ be the sequence of conclusions of the derivation $\hookrightarrow$ in β. We have that $\forall i \in \omega$, $\exists a_i$ such that $(q_i,a_i,q_{i+1}) \in \Delta$. Let $u = (a_i)_{i\in\omega}$. The sequence $\rho_\mathcal{B} = ((q_i,r_i))_{i\in\omega}$ is a run of $\mathcal{B}$ over u and $\rho_\mathcal{P} = (q_i)_{i\in\omega}$ is a run of $\mathcal{P}$ over u. There are two possible cases:

- The run $\rho_\mathcal{B}$ is not accepting. Then by Proposition 7 the left thread of β is a μ-thread, hence β is a valid branch.
- The run $\rho_\mathcal{B}$ is accepting. This means that there is $r \in Q_{ev}$ and $j \in \omega$ such that $\forall i \geq j$ we have $r_i = r$ and $c(q_i) \geq c(r)$ and $q_i = r$ infinitely many times. Hence the run $\rho_\mathcal{P}$ is also accepting. By Proposition 7, the right thread of β is a ν-thread, hence β is a valid branch. □

Using Proposition 3, we get the following theorem, which is step III of our completeness proof. The other implication can be show in the same way.

Theorem 3. *One can construct a* $\mu\mathsf{LK}$ *proof of* $[\mathcal{B}] \vdash [\mathcal{P}]$*.*

V. Constructive completeness

In the previous sections, we have shown steps I-III of our proof, as announced in the introduction. We are left with the following two theorems, corresponding to steps IV and V:

Theorem 4. *If* $\mathcal{B}$ *is a NBW such that* $\mathcal{L}(\mathcal{B}) = \Sigma^\omega$ *then there is a DBW* $\mathcal{D}$ *such that* $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$ *and one can construct a* $\mu\mathsf{LK}$ *proof of* $[\mathcal{D}] \vdash [\mathcal{B}]$

Theorem 4 follows from a more general result ([8]): if $\mathcal{B}_1, \mathcal{B}_2$ are NBW such that $\mathcal{L}(\mathcal{B}_1) \subseteq \mathcal{L}(\mathcal{B}_2)$, then one can construct a $\mu\mathsf{LK}$ proof of $[\mathcal{B}_1] \vdash [\mathcal{B}_2]$. If we set $\mathcal{B}_2 = \mathcal{B}$ and $\mathcal{B}_1$ to be the DBW $\mathcal{D} = \{\Sigma, \{q\}, q, \Delta, \{q\}\}$ where $\Delta = \{(q,a,q) | a \in \Sigma\}$, we get our result.

Remark 5. The result of [8] uses, as Steps II-III, an automata transformation which is determinization of NBW into deterministic Rabin automata, via Safra's algorithm. But contrarily to Steps II-III where automata transformations are used to build an automaton, Safra's determinization is used as a proof-search algorithm.

Theorem 5. *For every DBW* $\mathcal{D}$ *such that* $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$*, one can construct a* $\mu\mathsf{LK}$ *proof of* $\vdash [\mathcal{D}]$*.*

We first show the following lemma:

Lemma 3. *Let* Δ *be the transition relation of* $\mathcal{D}$*,* p *be a state and* Θ *be an environment for* $\mathcal{D}$*. The following rule is derivable in* $\mu\mathsf{LK}^\omega$ *using a thin derivation:*

$$\frac{\{\vdash [\![q]\!]^{\Theta,(p,t)}\}_{t=(p,a,q)\in\Delta}}{\vdash [\![p]\!]^\Theta}\ (\to)$$

Proof. Since $\mathcal{D}$ is deterministic, for every $a \in \Sigma$ there is a unique state q_a such that $t_a := (p, a, q_a) \in \Delta$. We have then:

$$[\![p]\!]^\Theta \to \bigvee_{a\in\Sigma} [a] \wedge [\![q_a]\!]^{\Theta,(p,t_a)}$$

hence we have the following derivation:

$$\dfrac{\dfrac{\{\vdash [a] \wedge [\![q_a]\!]^{\Theta,(p,t_a)}\}_{a\in\Sigma}}{\vdash \bigvee_{a\in\Sigma} [a] \wedge [\![q_a]\!]^{\Theta,(p,t_a)}}\ (\vee_r)}{\vdash [\![p]\!]^\Theta}\ (\sigma_r)$$

To justify this premisse, we apply the following derivation whenever it applies:

$$\dfrac{\vdash [a], \Gamma \qquad \dfrac{\vdash [\![q_a]\!]^{\Theta,(p,t_a)}}{\vdash [\![q]\!]^{\Theta,(p,t_a)}, \Gamma}\ (\mathsf{W}_r)}{\vdash [a] \wedge [\![q_a]\!]^{\Theta,(p,t_a)}, \Gamma}\ (\wedge_r)$$

Doing so, we get the following rule:

$$\dfrac{\vdash \{[a]\}_{a\in\Sigma} \qquad \{\vdash [\![q_a]\!]^{\Theta,(q_a,t)}\}_{a\in\Sigma}}{\vdash [\![p]\!]^\Theta}\ (\to)$$

Now we show that $\vdash \{[a]\}_{a\in\Sigma}$ is provable. For that we notice that $\{[a]\}_{a\in\Sigma} = \{p \wedge [a]\}_{a\in\Sigma_p} \cup \{p^\perp \wedge [a]\}_{a\in\Sigma_p}$ where $\Sigma_p = 2^{\mathcal{P}\setminus\{p\}}$. We apply the following scheme recursively, to obtain a finite proof of $\vdash \{[a]\}_{a\in\Sigma}$:

$$\dfrac{\dfrac{}{\vdash p, p^\perp}\ (\mathsf{Ax}) \qquad \vdash \{[a]\}_{a\in\Sigma_p}}{\vdash \{[a]\}_{a\in\Sigma}}\ (\wedge_r),\ (\mathsf{W}_r)$$

Which concludes the proof. □

Proof of Theorem 5. Let π be the proof obtained by applying coinductively the derivation of Lemma 3. This derivation is a thin $\mu\mathsf{LK}^\omega$ pre-proof. To show that it is actually a proof, notice that every infinite branch β of π contains exactly one right thread t. The run-branch of t, $\rho(t)$ is an accepting run, since every run of $\mathcal{D}$ is accepting. By Proposition 7, t is a ν-thread. Thus π is a proof. □

We can show now our completeness result.

Theorem 6. *If $\models \varphi$, we can construct a $\mu\mathsf{LK}$ proof of $\vdash \varphi$.*

Proof. Let φ be a valid formula. Let $\mathcal{A}_\varphi$ the APW associated to φ (Definition 32). By Proposition 12, one has $\mathcal{M}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$, thus $\mathcal{L}(\mathcal{A}_\varphi) = \Sigma^\omega$. By Theorem 1, one can construct a $\mu\mathsf{LK}$ proof π_1 of $[\mathcal{A}_\varphi] \vdash \varphi$. Let $\mathcal{P}$ be the NPW constructed from $\mathcal{A}_\varphi$ using the algorithm of the proof of Proposition 15. One has $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A}) = \Sigma^\omega$. By Theorem 2, one can construct a $\mu\mathsf{LK}$ proof π_2 of $[\mathcal{P}] \vdash [\mathcal{A}_\varphi]$. Let $\mathcal{B}$ be the NBW constructed from $\mathcal{P}$ using the algorithm of the proof of Proposition 17. One has $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{P}) = \Sigma^\omega$. By Theorem 3, one can construct a $\mu\mathsf{LK}$ proof π_3 of $[\mathcal{B}] \vdash [\mathcal{P}]$. Since $\mathcal{L}(\mathcal{B}) = \Sigma^\omega$ and using Theorem 4, there is DBW $\mathcal{D}$ such that $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$ and such that we can build a proof π_4 of $[\mathcal{D}] \vdash [\mathcal{B}]$. Finally by Proposition 5, one can construct a proof π_5 of $\vdash [\mathcal{D}]$. We gather all these pieces together to build a $\mu\mathsf{LK}$ proof of $\vdash \varphi$ using several cuts:

$$\dfrac{\dfrac{\pi_5}{\vdash [\mathcal{D}]} \quad \dfrac{\pi_4}{[\mathcal{D}] \vdash [\mathcal{B}]} \quad \dfrac{\pi_3}{[\mathcal{B}] \vdash [\mathcal{P}]} \quad \dfrac{\pi_2}{[\mathcal{P}] \vdash [\mathcal{A}_\varphi]} \quad \dfrac{\pi_1}{[\mathcal{A}_\varphi] \vdash \varphi}}{\vdash \varphi}\ (\mathsf{Cut})$$ □

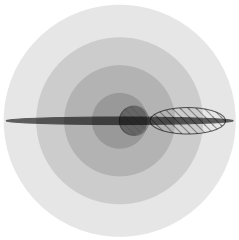


Fig. 5: Comparing our classes with Kaivola's and Walukiewicz' ones

The proof of Theorem 6 describes an algorithm that outputs a $\mu\mathsf{LK}$ proof for every valid formula. We can adapt it to get an algorithm that takes an arbitrary formula φ and outputs either a proof of φ if φ is valid; or a word $u \notin \mathcal{M}(\varphi)$. For that we start by building the automaton $\mathcal{B}$ in the proof of Theorem 6 and a DBW $\mathcal{D}$ such that $\mathcal{L}(\mathcal{D}) = \Sigma^\omega$. Then we run the algorithm of [8] for $\mathcal{D}$ and $\mathcal{B}$. This algorithm has the ability to output either a proof of $[\mathcal{D}] \vdash [\mathcal{B}]$ if $\mathcal{L}(\mathcal{D}) \subseteq \mathcal{L}(\mathcal{B})$; or a word u such that $u \in \mathcal{L}(\mathcal{D})$ and $u \notin \mathcal{L}(\mathcal{B})$ otherwise. If we get a proof, then φ is valid and we carry on with the steps I-III and V. Otherwise, the output word u satisfy $u \notin \mathcal{M}(\varphi)$ since $\mathcal{M}(\varphi) = \mathcal{L}(\mathcal{B})$.

VI. Conclusion

Contributions. We have given a new completeness argument for the full linear-time μ-calculus. Unlike earlier proofs, our argument is constructive, extending the work started in [8] for Büchi inclusions. To achieve this, we have combined techniques and tools coming from automata theory and proof theory, two domains that have been groing apart. One the one hand, we used the fine-grained formalism of occurrences which is widely used in proof theory but far removed from the automata side. On the other hand, we showed that the automata transformations can be reflected in the proof-theoretic side.

Related work. Our proof generalizes a previously used idea, which consists in introducing an intermediate class of formulas to solve the completeness problem. Kaivola's proof uses the class of Banan form formulas, which are formulas without fixed points interleaving and with a very restrictive use of conjunctions and disjunctions(the horizontal black class in Figure 5). Walukiewicz proof uses negations of disjunctive formulas, which corresponds roughly to the encoding of deterministic alternating parity automata(the dashed region in Figure 5). These two classes are "orthogonal" to the ones we have chosen.

Future work. The first direction for future work is to characterize and improve the complexity of the algorithm underlying our argument. Another direction is to obtain a constructive completeness for the branching-time μ-calculus. The latter

has infinite trees as models, and its corresponding model of automata is alternating parity automata over infinite trees. It seems difficult to lift our proof technique to the branching-time case. For instance, there is no algorithm to transform non-deterministic parity tree automata into non-deterministic Büchi tree automata. We hope though that the tools developed in the present work may help to solve this problem.

REFERENCES

[1] M. Y. Vardi, "A temporal fixpoint calculus," in *POPL California, USA, January 1988*.

[2] H. Barringer, R. Kuiper, and A. Pnueli, "A really abstract concurrent model and its temporal logic," in *POPL Florida, USA, January 1986*.

[3] I. Walukiewicz, "A complete deductive system for the mu-calculus," Ph.D. dissertation, Warsaw University, 1994.

[4] R. Kaivola, "Axiomatising linear time mu-calculus," in *CONCUR '95: Concurrency Theory, 6th International Conference, Philadelphia, PA, USA, August 21-24, 1995, Proceedings*, 1995, pp. 423–437.

[5] I. Walukiewicz, "Completeness of kozen's axiomatisation of the propositional mu-calculus," in *LICS 95, San Diego, California, USA, June 26-29, 1995*, 1995, pp. 14–24.

[6] D. Kozen, "Results on the propositional mu-calculus," *Theor. Comput. Sci.*, vol. 27, pp. 333–354, 1983. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(82)90125-6

[7] I. Walukiewicz, "On completeness of the mu-calculus," in *LICS 93, Montreal, Canada, June 19-23, 1993*, pp. 136–146.

[8] A. Doumane, D. Baelde, L. Hirschi, and A. Saurin, "Towards completeness via proof search in the linear time μ-calculus: The case of büchi inclusions," in *Proceedings of LICS '16, New York, USA, July 5-8, 2016*.

[9] C. Dax, M. Hofmann, and M. Lange, "A proof system for the linear time μ-calculus," in *FSTTCS 2006, Kolkata, India, December 13-15, 2006, Proceedings*, pp. 273–284.

[10] A. Doumane, "Constructive completeness for the linear-time mu-calculus," Tech. Rep. https://www.irif.fr/$\sim$doumane/MAIN.pdf, 2017.

[11] D. Baelde, A. Doumane, and A. Saurin, "Infinitary proof theory: the multiplicative additive case," in *CSL 2016, August 29 - September 1, 2016, Marseille, France*.

[12] D. Janin and I. Walukiewicz, "Automata for the modal mu-calculus and related results," in *MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995*, pp. 552–562.

[13] W. Thomas, "Handbook of formal languages, vol. 3," G. Rozenberg and A. Salomaa, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 1997, ch. Languages, Automata, and Logic, pp. 389–455.