



**HAL**  
open science

# Circular representations of infinite proofs for fixed-points logics : expressiveness and complexity

Rémi Nollet

► **To cite this version:**

Rémi Nollet. Circular representations of infinite proofs for fixed-points logics : expressiveness and complexity. General Mathematics [math.GM]. Université Paris Cité, 2021. English. NNT : 2021UNIP7100 . tel-03692893

**HAL Id: tel-03692893**

**<https://theses.hal.science/tel-03692893>**

Submitted on 10 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Université de Paris

1

2

Université de Paris

3

École doctorale de sciences mathématiques de Paris centre — ED 386

4

Institut de recherche en informatique fondamentale

5

Too complicated to have simple labels

6

Circular representations of infinite proofs for fixed-points logics: expressiveness and  
complexity

7

8

Par Rémi Nollet

9

Thèse de doctorat d'informatique fondamentale

10

Dirigée par Christine Tasson

11

Et par Alexis Saurin

12

Présentée et soutenue publiquement le 29 juin 2021

13

Devant un jury composé de :

14

Laurent Regnier, professeur des Universités, Université d'Aix-Marseille, rapporteur

15

Damien Pous, directeur de recherche, C. N. R. S. et É. N. S. de Lyon, rapporteur

16

Delia Kesner, professeure des Universités, Université de Paris, examinatrice

17

Bahareh Afshari, *associate professor*, University of Amsterdam et Göteborgs Universitet,  
examinatrice

18

1 Christine Tasson, professeure des Universités, Sorbonne Université, directrice

2 Alexis Saurin, chargé de recherche, C. N. R. S. et Université de Paris, directeur

3 This work is licensed under CC BY 4.0. To view a copy of this license, visit

4 <http://creativecommons.org/licenses/by/4.0/>

1 Titre : Représentations circulaires de preuves infinies pour les logiques à points fixes :  
2 expressivité et complexité.

3 Résumé : Dans le cadre des logiques adaptées aux plus petits et plus grands points fixes  
4 de formules, les preuves circulaires ont été proposées comme alternative aux principes  
5 d'induction et coinduction à invariants explicites. Ces preuves circulaires ont reçu un  
6 intérêt grandissant ces dernières années avec le développement simultané de leurs appli-  
7 cations et de leur méta-théorie. Les preuves infinies sont maintenant bien implantées  
8 dans plusieurs sujets de la théorie de la démonstration tels que les prédicats inductifs à  
9 la Martin-Löf, la logique linéaire étendue avec des points fixes, etc.

10 Dans le cadre des preuves circulaires ou infinies, non bien-fondées, un critère de validité  
11 est nécessaire pour rétablir la cohérence logique. Ce critère sert à distinguer, parmi  
12 l'ensemble des prépreuves infinies ou circulaires, celles qui sont des preuves valides. Une  
13 approche maintenant standard est de considérer qu'une prépreuve est valide si chacune  
14 de ses branches infinies est justifiée par un thread qui progresse infiniment souvent.

15 Cette thèse étudie ces problèmes dans le cadre plus spécifique de MALL étendue avec des  
16 points fixes. On sait que, étant donnée une représentation circulaire finie d'une prépreuve  
17 non bien-fondée, il est possible de décider en espace polynomial si cette prépreuve est  
18 valide vis-à-vis du critère de threads. La première contribution de cette thèse est de  
19 démontrer que ce problème de décider le critère de threads pour  $\mu\text{MALL}^\omega$  est en fait  
20 PSPACE-complet. Notre démonstration est fondée sur une étude approfondie des liens  
21 entre le critère de threads et le principe de size-change termination, habituellement utilisé  
22 pour garantir la terminaison de programmes.

23 La deuxième contribution de cette thèse est de décrire un nouveau fragment de l'ensemble  
24 des représentations circulaires valides de prépreuves, fondé sur un critère de validité plus  
25 contraignant. Ce nouveau critère s'appuie sur un étiquetage des formules et des preuves, et  
26 sa validité est purement locale. Il permet à ce fragment d'être plus simple à manipuler, tout  
27 en restant suffisamment expressif pour contenir l'image des preuves finitaires du système  
28  $\mu\text{MALL}$  de David Baelde, qui fonctionnent avec des invariants explicites d'induction  
29 et de coinduction. En particulier, il est possible de décider efficacement la validité  
30 d'une prépreuve circulaire vis-à-vis de ce nouveau critère, et de calculer efficacement un  
31 étiquetage qui certifie cette validité.

32 De plus, et ceci constitue la troisième contribution de cette thèse, la conjecture de  
33 Brotherston-Simpson est vraie pour ce fragment : nous décrivons une méthode qui permet  
34 de transformer toute preuve circulaire étiquetée de notre fragment en une preuve finitaire,  
35 avec des invariants d'induction et de coinduction explicites. Finalement, nous explorons  
36 des manières d'étendre ces résultats de validité et de finitisation à un plus grand fragment,  
37 en relâchant les contraintes de l'étiquetage tout en conservant (1) la possibilité de certifier  
38 localement la validité des preuves et (2) dans une certaine mesure, la possibilité de  
39 finitiser les preuves circulaires ainsi étiquetées.

1 Mots clefs : Size-change, PSPACE-complet, complexité, calcul des séquents, preuves  
2 non bien-fondées, preuves circulaires, induction, coinduction, points fixes, recherche de  
3 preuves, logique linéaire, muMALL, finitisation.

4 Title: Circular representations of infinite proofs for fixed-points logics: expressiveness and  
5 complexity.

6 Abstract: In the context of logics with least and greatest fixed points, circular, i./,e.  
7 non wellfounded but regular proofs have been proposed as an alternative to induction  
8 and coinduction with explicit invariants. Those circular proofs have received increasing  
9 interest in recent years with the simultaneous development of their applications and  
10 metatheory: infinitary proof theory is now well-established in several proof-theoretical  
11 frameworks such as Martin-Löf's inductive predicates, linear logic with fixed points,  
12 etc.

13 In the setting of non wellfounded proofs, in order to recover logical consistency, it is  
14 necessary to consider a validity criterion which differentiates, among all infinite or circular  
15 derivation trees, a. k. a. preproofs, those which are logically valid proofs. A standard  
16 approach is to consider a preproof to be valid if every infinite branch is supported by an  
17 infinitely progressing thread.

18 This thesis focuses on circular proofs for MALL with fixed points. It is known that given  
19 a finite circular representation of a non-wellfounded preproof, one can decide in PSPACE  
20 whether this preproof is valid with respect to the thread criterion. The first contribution  
21 of this thesis is to prove that the problem of deciding thread-validity for  $\mu\text{MALL}^\omega$  is in  
22 fact PSPACE-complete. Our proof is based on a deeper exploration of the connection  
23 between thread-validity and the size-change termination principle, which is usually used  
24 to ensure program termination.

25 The second contribution of this thesis is to describe a new fragment of all valid circular  
26 representations of preproofs, based on a stronger validity criterion. This new criterion is  
27 based on a labeling of formulas and proofs, whose validity is purely local. This allows  
28 this fragment to be easily handled, while being expressive enough to still contain all  
29 circular embeddings of Baelde's  $\mu\text{MALL}$  finite proofs with inductive and coinductive  
30 invariants. In particular, deciding validity and computing a certifying labelling can be  
31 done efficiently.

32 Moreover, and this is the third contribution of this thesis, the Brotherston-Simpson  
33 conjecture holds for this fragment: every labeled representation of a circular proof in the  
34 fragment is translated into a standard finitary proof. Finally, we explore how to extend  
35 these results to a bigger fragment, by relaxing the labeling discipline while retaining (1)  
36 the ability to locally certify the validity and (2) to some extent, the ability to finitize  
37 circular proofs.

<sup>1</sup> Keywords: Size-change, PSPACE-complete, Complexity, sequent calculus, non-wellfounded  
<sup>2</sup> proofs, circular proofs, induction, coinduction, fixed points, proof-search, linear logic,  
<sup>3</sup> muMALL, finitization.



# Contents

2	<b>1 Introduction</b>	<b>9</b>
3	1.1 Generalities on least and greatest fixed points . . . . .	9
4	1.1.1 Basic definitions . . . . .	9
5	1.1.2 Proving their existence . . . . .	9
6	1.1.3 Expressivity in logic . . . . .	11
7	1.1.4 Expressivity in programs . . . . .	13
8	1.1.5 Usage in logic . . . . .	14
9	1.2 Least and greatest fixed points in proof theory . . . . .	14
10	1.2.1 Proof theory . . . . .	15
11	1.2.2 Usage of fixed points in proof theory . . . . .	15
12	1.2.3 Finite proof systems for least and greatest fixed points . . . . .	16
13	1.2.4 Infinite proof systems for least and greatest fixed points . . . . .	17
14	1.2.5 Finite representations and circular proof systems for least and 15 greatest fixed points . . . . .	18
16	1.3 In this thesis . . . . .	20
17	1.3.1 Fixed points in linear logic . . . . .	20
18	1.3.2 Addressed questions, proposed answers . . . . .	21
19	1.3.3 Outline of this thesis . . . . .	23
20	<b>2 Technical background</b>	<b>27</b>
21	2.1 Proof theory for least and greatest fixed points . . . . .	27
22	2.1.1 Usual proof theory and MALL . . . . .	27
23	2.1.2 Finite proof trees: $\mu$ MALL . . . . .	34
24	2.1.3 Infinite proof trees: $\mu$ MALL <sup>∞</sup> . . . . .	37
25	2.1.4 Finite proof trees with back edges: $\mu$ MALL <sup>ω</sup> . . . . .	39
26	2.2 Complexity theory . . . . .	49
27	2.2.1 Generalities on complexity and PSPACE . . . . .	49
28	2.2.2 Some PSPACE-complete problems . . . . .	52
29	2.3 Parity automata . . . . .	52
30	<b>3 PSPACE-completeness of the thread criterion</b>	<b>55</b>
31	3.1 Deciding thread validity in PSPACE . . . . .	56
32	3.2 PSPACE-completeness . . . . .	60
33	3.2.1 Outline of the PSPACE-completeness proof . . . . .	60
34	3.2.2 Definition of the reduction . . . . .	60

1	3.2.3	Main theorem . . . . .	62
2	3.2.4	Generalisation to other systems . . . . .	69
3	3.3	Comments on our approach and discussion of related works . . . . .	71
4	3.4	Conclusion . . . . .	72
5	<b>4</b>	<b>A polynomial sub-criterion</b>	<b>81</b>
6	4.1	$\mathcal{L}$ -proofs: labelling as validity . . . . .	83
7	4.2	Finite representations of circular $\mathcal{L}$ -proofs. . . . .	85
8	4.3	Two alternative characterizations of $\mu\text{MALL}^{\curvearrowright}$ . . . . .	89
9	4.4	Conclusion . . . . .	96
10	<b>5</b>	<b>Finitization</b>	<b>99</b>
11	5.1	On Brotherston-Simpson’s conjecture: finitizing circular proofs . . . . .	100
12	5.2	Relaxing the labelling of proofs . . . . .	103
13	5.3	Extending finitization . . . . .	108
14	5.4	An <i>ad hoc</i> finitization of $\pi_\infty$ . . . . .	109
15	5.5	Related works on Chapters 4 and 5 . . . . .	118
16	5.6	Conclusion . . . . .	118
17	<b>6</b>	<b>Conclusion</b>	<b>121</b>
18	6.1	General conclusion . . . . .	121
19	6.2	Future works . . . . .	122

# 1 Introduction

## 1.1 Generalities on least and greatest fixed points

### 1.1.1 Basic definitions

The general subject of this thesis is the study of least and greatest fixed points in logic.

The basic notions of least and greatest fixed points are easily described in the setting of ordered sets.

If  $(E, \leq)$  is an order and  $f: E \rightarrow E$  is a non-decreasing endofunction, an element  $x \in E$  is a *fixed point* of  $f$  when  $f(x) = x$ .

More than general fixed points, what interests us are *pre-* and *post-fixed points*. An element  $x \in E$  is a *pre-fixed point* of  $f$  when  $f(x) \leq x$ , and it is a *post-fixed point* of  $f$  when  $x \leq f(x)$ .

Even more precisely, what we want to talk about are *least pre-fixed points* and *greatest post-fixed points*. An element  $x \in E$  is a *least pre-fixed point* of  $f$  when  $f(x) \leq x$  and  $\forall y \in E, f(y) \leq y \Rightarrow x \leq y$ . And  $x$  is a *greatest post-fixed point* of  $f$  when  $x \leq f(x)$  and  $\forall y \in E, y \leq f(y) \Rightarrow y \leq x$ . There are usual notations for that: the *least pre-fixed point* of a non-decreasing function  $f$  is denoted, when it exists, by  $\mu x f(x)$ , and similarly its *greatest post-fixed point* is denoted by  $\nu x f(x)$ .

Note that the least pre-fixed point of a non-decreasing function is always a fixed point, and therefore it is its least fixed point.

### 1.1.2 Proving their existence

The main theorems on the existence of least and greatest fixed points are the following:

## 1 Introduction

1 **Theorem 1.** *Let  $(E, \leq)$  be an ordered set.*

2 1. *If*  
3 *every well-ordered subset of  $E$  has a least upper bound* (1.1)

4 *then*  
5 *every non-decreasing endofunction on  $E$  has a least pre-fixed point* (1.2)

6 2. *Conversely, if*  
7 *every non-decreasing endofunction on  $E$  has a least fixed point* (1.3)

8 *then*  
9  *$E$  has a minimum and every directed subset of  $E$  has a least upper bound* (1.4)

10 *in which we recall that a subset  $A$  of  $E$  is said to be directed when every finite subset of*  
11  *$A$  has an upper bound in  $A$ .*

12 *Proof elements and references.*

- 13 • First remark that, as already mentioned above, a least pre-fixed point is always a  
14 fixed point and, therefore, (1.2) is stronger than (1.3). Similarly, every well-ordered  
15 subset of an order is either empty or directed. Hence (1.4) is stronger than (1.1).  
16 That is why the two parts 1 and 2 of that theorem are indeed converse to each  
17 other.
- 18 • The direction 1 seems to be a folklore theorem which is difficult to trace back to a  
19 unique original author. Yet the first version of it seems to be that of Abian and  
20 Brown [1961], although it is very close to a similar theorem by Bourbaki [1949].  
21 See [Lassez, Nguyen, and Sonenberg, 1982, Section 4] for a bibliographic study of  
22 its origins. See Davey and Priestley [2002, Exercise 8.19, p. 198] for a quick proof  
23 using ordinal induction.
- 24 • The direction 2 is more recent and has been proved by Markowsky [1976].

25 To the reader interested by those questions, we recommend the book by Davey and  
26 Priestley [2002, in particular ch. 8, p. 175, and the bibliographic discussion p. 285], which  
27 contains more details and references about fixed-point theorems.  $\square$

28 That theorem characterizes, in a sense, the preorders on which it makes sense to study  
29 the least fixed points of non-decreasing endofunctions. And it says that in these preorders,

1 those least fixed points are in fact least pre-fixed points. Those orders are called complete  
2 partial orders, or CPO.

3 Because of that theorem, we will commonly say “least fixed point” when talking about  
4 least pre-fixed points, and similarly for greatest fixed points.

### 5 1.1.3 Expressivity in logic

6 We will now motivate the will to add a connective to logic that has the behavior of  
7 forming a least fixed point by a first example, coming from temporal logic. We will see  
8 that least pre-fixed points and greatest post-fixed points give the possibility to define  
9 new formulas, and therefore add expressivity to the language. Such ideas are the basis of  
10 modal  $\mu$ -calculus [Kozen, 1983].

11 Let us consider a logic like LTL [Pnueli, 1977], in which the truth of a proposition  $A$  may  
12 depend on an instant  $t \in \mathbf{N}$ . The boolean denotation of a formula  $A$  is a  $\llbracket A \rrbracket: \mathbf{N} \rightarrow \{0, 1\}$   
13 and if  $t \in \mathbf{N}$ , we denote by  $\llbracket A \rrbracket_t \in \{0, 1\}$  the truth value of  $A$  at  $t$ . As an alternative,  
14 we may write  $t \models A$  for  $\llbracket A \rrbracket_t = 1$ . We use one logical connective and two temporal  
15 connectives. For each of them, we give two equivalent definitions:

$$\begin{array}{lll}
 \text{“or”} & \llbracket A \vee B \rrbracket_t = \max\{\llbracket A \rrbracket_t, \llbracket B \rrbracket_t\} & t \models A \vee B \Leftrightarrow t \models A \text{ or } t \models B \\
 \text{“next”} & \llbracket \bigcirc A \rrbracket_t = \llbracket A \rrbracket_{t+1} & t \models \bigcirc A \Leftrightarrow t+1 \models A \\
 \text{“eventually”} & \llbracket \diamond A \rrbracket_t = \max\{\llbracket A \rrbracket_s \mid s \geq t\} & t \models \diamond A \Leftrightarrow \exists s \geq t, s \models A
 \end{array}$$

16 We abuse the notation  $\models$  by using  $A \models B$  to denote the fact that  $\forall t \in \mathbf{N}, \llbracket A \rrbracket_t \leq \llbracket B \rrbracket_t$ ,  
17 that is, equivalently:  $\forall t \in \mathbf{N}$ , if  $t \models A$  then  $t \models B$ . Now we say that

18 **Proposition 1.** *For every formulas  $A$  and  $B$ , the following are true:*

$$\begin{array}{ll}
 19 \quad A \vee \bigcirc \diamond A \models \diamond A & (1.5) \\
 \frac{A \vee \bigcirc B \models B}{\diamond A \models B} & (1.6)
 \end{array}$$

20 *Proof.*

21 1. Proof of (1.5)

## 1 Introduction

1 Assume that  $t \in \mathbf{N}$  and  $t \models A \vee \bigcirc \diamond A$ . There are two cases:

2 1.1.  $t \models A$

3 In that case  $t \models \diamond A$ .

4 1.2.  $t \models \bigcirc \diamond A$

5 In that case  $t + 1 \models \diamond A$ , that is  $\exists s \geq t + 1, s \models A$ , so  $t \models \diamond A$ .

6 Hence, in any case,  $t \models \diamond A$ .

7 2. Proof of (1.6)

8 Let us assume that

$$9 \qquad A \vee \bigcirc B \models B \qquad (1.7)$$

10 and show that  $\diamond A \models B$ .

11 2.1. We first prove the following lemma:  $\forall u \in \mathbf{N}, \forall t \in \mathbf{N}, t + u \models A \Rightarrow t \models B$ .

12 PROOF: By induction on  $u$ .

13 2.1.1. Case  $u = 0$ :

14 In that case we know that  $t \models A$ , hence  $t \models A \vee \bigcirc B$ . By assumption (1.7)  
15 we get  $t \models B$ .

16 2.1.2. Case  $u = v + 1$ :

17 In that case we know that  $t + 1 + v \models A$ . By induction hypothesis on  $v$ ,  
18 we get  $\forall t' \in \mathbf{N}, t' + v \models A \Rightarrow t' \models B$ , which we apply to  $t' = t + 1$  to get  
19  $t + 1 \models B$ , that is  $t \models \bigcirc B$ . Hence  $t \models A \vee \bigcirc B$ . By assumption (1.7) we  
20 deduce  $t \models B$ .

21 2.2. From this, we immediately deduce the following:

$$22 \qquad \forall s \in \mathbf{N}, s \models A \Rightarrow \forall t \leq s, t \models B \qquad (1.8)$$

23 2.3. We now prove that  $\diamond A \models B$ .

24 Assume a  $t \in \mathbf{N}$  and  $t \models \diamond A$ , that is assume a  $s \in \mathbf{N}, s \geq t$  and  $s \models A$ . It remains  
25 to show that  $t \models B$ . This is exactly given by (1.8) above.

26 □

27 This relates to our previous description of **least/greatest pre-/post-fixed points** in the  
28 following way. If we consider  $\models$  as our order, we just proved that  $\diamond A$  is a least pre-fixed  
29 point of  $F: X \mapsto A \vee \bigcirc X$ , because we proved that  $F(\diamond A) \leq \diamond A$  and  $\forall B, F(B) \leq B \Rightarrow$   
30  $\diamond A \leq B$ , with  $\leq$  being  $\models$ .

31 That means that if we are allowed to use **least pre-fixed points** in the construction  
32 of the formulas, we do not need  $\diamond$  to be given as a primitive connective of the logic;  
33 we could *define* it as  $\diamond A := \mu X(A \vee \bigcirc X)$ . Dually, you can check that the “always”  
34 operator, denoted by  $\square A$  and defined equivalently by  $\llbracket \square A \rrbracket_t = \min\{\llbracket A \rrbracket_s \mid s \geq t\}$  or  
35  $t \models \square A \Leftrightarrow \forall s \geq t, s \models A$  could alternatively be defined as the **greatest post-fixed point**  
36  $\square A := \nu X(A \wedge \bigcirc X)$ .

### 1 1.1.4 Expressivity in programs

2 Least pre-fixed points and greatest post-fixed points can also be used to model inductive  
3 and coinductive datatypes in programming languages. We will try to show that on a  
4 second example.

5 In this example, we will use an OCaml-like syntax, although we are not writing strictly  
6 legit OCaml. We are trying to convey a general intuition, which probably apply *mutatis*  
7 *mutandis* to any programming language in which one can define inductive algebraic  
8 datatypes.

9 Let us consider the type of integer lists, which can be described as:

```
type int list = [] | ( :: ) of int * int list
```

10 And let us consider the following type `t`, parameterized over a type `'a`:

```
type 'a t = Nil | Cons of int * 'a
```

11 Remark that the type `t` is conceptually simpler than `int list`, because it is not a  
12 recursive type. If you are familiar with algebraic datatypes, you may see that this type  
13 could be written as

$$'a\ t = 1 + \text{int} * 'a$$

14 where we use `1` to denote the singleton type:

```
type 1 = ()
```

15 Now we can define two functions

```
inj  : int list t -> int list
fold : ('a t -> 'a) -> int list -> 'a
```

16 as follows:

```
let inj = function
  | Nil      -> []
  | Cons (x, xs) -> x :: xs

let rec fold f = function
  | []      -> f Nil
  | x :: xs -> f (Cons (x, fold f xs))
```

1

\*

2 In this example, for every two types  $\alpha$  and  $\beta$ , let us say that  $\alpha \leq \beta$  when the type  $\alpha \rightarrow \beta$   
 3 is inhabited. Then if we take  $F(\alpha)$  to be the type  $\alpha \mathbf{t}$ , we have  $F(\mathbf{int\ list}) \leq \mathbf{int\ list}$   
 4 because of `inj`. And each time we have a type  $\alpha$  such that  $F(\alpha) \leq \alpha$ , we also have  
 5  $\mathbf{int\ list} \leq \alpha$  thanks to `fold`. So it makes sense to say that  $\mathbf{int\ list}$  is a least pre-fixed  
 6 point of  $\mathbf{t}$ .

7 Another way of saying it is that  $\mathbf{int\ list} = \mu\alpha(1 + \mathbf{int} * \alpha)$ . And the general idea is  
 8 that inductive datatypes correspond to least pre-fixed points of functors. And, dually,  
 9 coinductive datatypes correspond to greatest post-fixed points of functors. For instance,  
 10 we could define a type of integer streams as  $\mathbf{int\ stream} = \nu\alpha(\mathbf{int} * \alpha)$ .

11 *Remark 1.* Actually the previous example is not totally complete. That example charac-  
 12 terises  $\mathbf{int\ list}$  as the least pre-fixed point of  $\mathbf{t}$ , but it happens that the singleton type  
 13 `1` could fit this description as well. In fact, the description of least and greatest fixed  
 14 points in terms of orders or preorders does not give enough precision. To have a complete  
 15 characterisation of  $\mathbf{int\ list}$  as a least fixed point, we would need to talk about the  
 16 computational behaviour of the functions, and we would need to shift from the setting  
 17 of orders and preorders to the setting of categories, which is out of the scope of this  
 18 introduction.

### 19 1.1.5 Usage in logic

20 There has been a lot of logics allowing the use of least and greatest fixed-point con-  
 21 structions in their formulas. Let us simply mention here some of the most noticeable.  
 22 de Bakker and de Roever [1972] define a logic with least fixed points to prove properties  
 23 of recursively defined programs. Aczel [1977] study the extension of recursion theory with  
 24 inductive definitions. Aho and Ullman [1979] study the relational calculus, which may be  
 25 understood as first-order logic, and suggest extending it by adding a least fixed-point  
 26 operator, in order to make it more expressive. Gurevich and Shelah [1986] study the  
 27 expressive power of different extensions of first-order logic with fixed point induction.  
 28 Dawar and Gurevich [2002] study the expressive power of fixed point logics.

## 29 1.2 Least and greatest fixed points in proof theory

30 Least and greatest fixed point of formulas may be seen as a way to formulate definitions by  
 31 induction, or by coinduction. We will study those least and greatest fixed points from the

1 point of view of proof theory. We believe that, through the proof-program correspondence,  
 2 the analysis we make on proofs could also be applied to programs. The advantage of  
 3 proofs over programs for our study is that they provide a system which makes things  
 4 visible, decomposing things in simple elements: subformula property, cut-elimination,  
 5 formulas occurrences, . . .

### 6 **1.2.1 Proof theory**

7 We do here a short recall of a few elements that made the success of modern proof theory  
 8 and sequent calculus.

9 In usual sequent calculus, a proof is defined to be a finite proof tree.

10 The **subformula property** says that if you can prove a given conclusion then you can do  
 11 it by using only subformulas of this conclusion. This is a fundamental property, saying,  
 12 intuitively, that if I ask you a question (“Is this formula true ?”), you can answer it using  
 13 only the concepts that are already present in the question, you do not need to introduce  
 14 any new concept. You may still do it, because it may ease the task, but you do not need  
 15 it.

16 The **cut rule** is the rule that allows you to do some deductive reasoning. It is a generali-  
 17 sation of two logical principles, *modus ponens*, which says that from a proof of  $A$  and a  
 18 proof of  $A \Rightarrow B$ , you can build a proof of  $B$ , and the transitivity of implication, which  
 19 says that from a proof of  $A \Rightarrow B$  and a proof of  $B \Rightarrow C$  you can build a proof of  $A \Rightarrow C$ .  
 20 One of the ideas that made sequent calculus so successful is that all inferences preserve  
 21 the subformula property, except for the cut rule. When following this discipline, the  
 22 subformula property amounts to saying that we can eliminate cuts from any proof.

### 23 **1.2.2 Usage of fixed points in proof theory**

24 Various logical settings have been introduced to reason about inductive and coinductive  
 25 statements, both at the level of the logical languages modelling (co)induction (Martin  
 26 L of’s inductive predicates vs. fixed-point logics, that is  $\mu$ -calculi) and at the level of the  
 27 proof-theoretical framework considered (finite proofs with (co)induction *à la* Park [1969]  
 28 vs. infinite proofs with fixed-point/inductive predicate unfoldings) [Brotherston, 2006,  
 29 Brotherston and Simpson, 2007, 2011, Baelde and Miller, 2007, Baelde, 2009, 2012].

30 Moreover, such proof systems have been considered over classical logic [Brotherston, 2006,  
 31 Brotherston and Simpson, 2011], intuitionistic logic [Clairambault, 2009], linear-time or  
 32 branching-time temporal logic [Kozen, 1983, Kaivola, 1995b, Walukiewicz, 1993, 1995,

1 Dax, Hofmann, and Lange, 2006, Doumane, 2017b, Doumane, Baelde, Hirschi, and Saurin,  
2 2016] or linear logic [Santocanale, 2002, Fortier and Santocanale, 2013, Baelde and Miller,  
3 2007, Baelde, Doumane, and Saurin, 2016, Doumane, 2017b].

4 In all those proof systems, the treatment of inductive and coinductive reasoning brings  
5 some highly complex proof objects.

### 6 **1.2.3 Finite proof systems for least and greatest fixed points**

7 The first systems designed to reason about fixed points of formulas were systems with  
8 finite proofs, in which induction and coinduction principles are used in order to provide  
9 a finite proof theory for reasoning on formulas with least or greatest fixed points [Kozen,  
10 1981, 1983, Baelde, 2012].

11 But finite proof systems for least and greatest fixed points have some drawbacks.

- 12 • To introduce a greatest fixed point formula or to eliminate a least fixed point  
13 formula, you have to use a rule of induction or coinduction, and you have to provide  
14 an explicit invariant, which has to be a formula of the same system. This is a  
15 restriction because the language of formulas may not be expressive enough to  
16 express all the invariants you may need to prove formulas that should intuitively  
17 be true, as was shown, for instance, by Berardi and Tatsuta [2017b, 2019] in the  
18 context of first-order classical logic with inductive definitions.
- 19 • It breaks the subformula property.
- 20 • Here, in the finitary setting for least and greatest fixed points of formulas, the  
21 failure of the subformula property can therefore be restated by saying that we  
22 cannot eliminate all cuts. Nevertheless, it makes sense to define a procedure of “cut  
23 elimination” for this setting. This procedure will not eliminate all cuts, but it will  
24 simplify them and eliminate some of them and it is still a terminating procedure,  
25 which leaves the proof in a normal form in which the only remaining cuts are of  
26 a very specific form, and are cuts that cannot be reduced anymore. The point is  
27 that in this setting, besides the fact that we cannot completely eliminate cuts, this  
28 procedure of cut elimination turns out to be quite complex.

29 For instance, in proof systems using (co)induction rules *à la* Park, the rules allowing to  
30 derive a coinductive property (or dually to use an inductive hypothesis) have a complex  
31 inference of the form of Figure 1.1 (when presented in the setting of fixed-point logic –  
32 here we follow the one-sided sequent tradition of MALL that we will adopt in the rest of  
33 the thesis).

$$\frac{\vdash \Gamma, S \vdash S^\perp, F[S/X]}{\vdash \Gamma, \nu X F} (\nu_{\text{inv}})$$

Figure 1.1. Coinduction rule *à la* Park

1 It is not only difficult to figure out intuitively what is the meaning of this inference,  
 2 but it is also problematic for at least two additional and more technical reasons: (i) it  
 3 is hiding a cut rule that *cannot* be eliminated, which is problematic for extending the  
 4 proof-program correspondence to fixed-points logics, and (ii) it breaks the subformula  
 5 property, which is problematic for proof search: at each coinduction rule, one has to guess  
 6 an invariant (in the same way as one has to guess an appropriate induction hypothesis in  
 7 usual mathematical proofs).

## 8 1.2.4 Infinite proof systems for least and greatest fixed points

9 Infinite proofs have been introduced to reason on (least and greatest) fixed points of  
 10 formulas.

11 For all these reasons, infinite (non-wellfounded) proofs, which are infinite proofs satisfying  
 12 a validity criterion, have been proposed, in recent years, as an alternative to induction and  
 13 coinduction with explicit invariants [Brotherston, 2006, Brotherston and Simpson, 2007,  
 14 2011]. By replacing the coinduction rule with simple fixed-point unfoldings and allowing  
 15 for non-wellfounded branches, those proof systems address the problem of the subformula  
 16 property for the cut-free systems. The cut-elimination dynamics for inductive-coinductive  
 17 rules is also much simpler. In particular, Baelde, Doumane, and Saurin [2016], inspired  
 18 notably by Dax, Hofmann, and Lange [2006], proposed a very successful system of infinite  
 19 proofs, called  $\mu\text{MALL}^\infty$ , to reason on least and greatest fixed points of formulas. Actually,  
 20 this is for the propositional setting, but such ideas already existed to reason on inductive  
 21 and coinductive predicates.

22 Infinite, non well-founded, proofs present the advantage over explicit induction or coinduc-  
 23 tion to offer a framework in which it is possible to recover the good structural properties  
 24 of sequent calculus, such as cut-elimination, subformula property and focusing, making  
 25 them a more suitable tool to automated proof search. Indeed, cut-elimination and focusing  
 26 have recently been extended to non well-founded proofs for  $\mu\text{MALL}$  by Baelde, Doumane,  
 27 and Saurin [2016], Doumane [2017b].

28 Now the problem is that if we take this approach too naively, by simply allowing proof  
 29 trees to be infinite, then two bad things happen:

- 30 1. the system becomes inconsistent, meaning that every formula becomes provable;

$$\frac{\frac{\vdots}{\vdash \mu XX, \Gamma}^{(\mu)} \quad \frac{\vdots}{\vdash \nu XX, \Gamma}^{(\nu)}}{\frac{\vdash \mu XX, \Gamma}^{(\mu)} \quad \vdash \nu XX, \Gamma}^{(\nu)}}{\vdash \Gamma}^{(\text{cut})}$$

Figure 1.2

- 1     2. the cut-elimination procedure becomes a non-terminating, non-converging one.
- 2     The price to pay is that the consistency of the logical system is broken and that a validity  
3     criterion has to be added in order to ensure consistency.
- 4     More precisely, in those proof systems when considering all possible infinite, non-  
5     wellfounded derivations, or **preproofs**, it is straightforward to derive any sequent  $\Gamma$   
6     (see Figure 1.2).
- 7     Such preproofs are therefore unsound and one needs to impose a validity criterion to  
8     distinguish, among all preproofs, those which are logically valid proofs from the unsound  
9     ones.
- 10    A solution to that is to say that all these infinite proof trees are only preproofs, and  
11    that only some of these preproofs are valid proofs. This means that we need a criterion  
12    to distinguish valid (pre)proofs from invalid preproofs, and to reflect the inductive and  
13    coinductive nature of our fixed-point connectives.
- 14    A standard approach [Brotherston, 2006, Brotherston and Simpson, 2007, 2011, Santo-  
15    canale, 2002, Baelde, Doumane, and Saurin, 2016] considers a preproof to be valid if  
16    every infinite branch is supported by an infinitely progressing thread. This is called the  
17    **thread criterion**.
- 18    However, doing so, the logical correctness of circular proofs becomes a non-local property,  
19    much in the spirit of proof nets correctness criteria [Girard, 1987, Danos and Regnier,  
20    1989].

## 21    1.2.5 Finite representations and circular proof systems for least and 22    greatest fixed points

- 23    The problem with infinite proofs, on the other hand, is mainly that they are infinite,  
24    which has two major drawbacks:

- 1 1. The first one is epistemic: we would like a proof to be a finite object, which I can  
2 communicate to you in finite time, and which you can check in finite time. The  
3 fact that a proof may be an infinite object means that I may not be able to give  
4 you a proof, but I will in fact give you the finite description of an infinite proof,  
5 formulated in the meta-theory.
- 6 2. The second one is practical: if we want our system to be used in an automated  
7 prover or in a proof assistant, we need those proofs to be finitely representable.  
8 It means that whatever system of representation you chose, you will not be able  
9 to represent all infinite proofs, because there is an uncountable number of infinite  
10 proofs and your system of finite representations will only be able to represent a  
11 countable number of them.

12 From that it may seem that going from finite to infinite proofs was a mistake. But infinite  
13 proofs give great insights on least and greatest fixed points of formulas, on their different  
14 proof systems and on the dynamic of cut-elimination in those systems, including the  
15 finitary ones. In particular they provide us with a new way to understand the finitary  
16 systems. Each finitary proof can be translated into an infinitary one. This means that  
17 finitary proofs may be seen as one particular way to represent some of those infinitary  
18 proofs. So now we may transform the question of designing a finitary proof system  
19 for least and greatest fixed points of formulas into the question of identifying suitable  
20 fragments of the infinitary system, which have some nice-behaved representations.

21 Among those non-wellfounded proofs, circular proofs, that have infinite but regular  
22 derivation trees, have attracted a lot of attention for retaining the simplicity of the  
23 inferences of non-wellfounded proof systems but being amenable to a simple finite  
24 representation making it possible to have an algorithmic treatment of those proof  
25 objects.

26 In this context, a very natural way of representing some infinite proofs is to use finite  
27 proof trees with back edges. This means that instead of constructing explicitly an infinite  
28 branch, we are allowed to stop at some point and point out some previous step of the  
29 construction and say: from there, we start again at that point. Such a cyclic representation  
30 has a canonical unfolding into an infinite proof tree. This system of representations allows  
31 us to represent some of the infinite preproofs. Those preproofs that can be represented  
32 by such a proof tree with back edges are called circular preproofs. Those representations  
33 are called circular representations.

34 Despite the need for a validity condition, circular, *i. e.* non-wellfounded but regular proofs  
35 have received increasing interest in recent years with the simultaneous development of  
36 their applications and meta-theory: infinitary proof theory is now well-established in  
37 several proof-theoretical frameworks such as Martin L of's inductive predicates, linear  
38 logic with fixed points, *etc.*

## 1 1.3 In this thesis

2 **Statement of the contribution** On the question of how much time and space it takes  
3 to check the **thread criterion** on a **circular representation** of an **infinite preproof**, it  
4 was known how to do it in **PSPACE**, and we prove in this thesis that this problem is  
5 **PSPACE-complete**, which means that we cannot do substantially better.

6 As we would like to be able to check our proofs in polynomial time, we provide a new  
7 **validity criterion**, which is stronger than the **thread criterion** while accepting at least the  
8 translations of **finitary proofs**, and which can be checked in quadratic time.

9 We also would like a proof system in which there is no **global criterion** to check, in  
10 which a proof is correct as soon as each inference used in it is correct. This is made  
11 possible. We provide such a proof system, obtained by adding some labelling to the  
12 **circular representations** accepted by my new quadratic criterion.

13 Finally, this new proof system proves at least as much propositions than the **finitary**  
14 **setting** and it is legit to ask how much more it does prove exactly. The answer is: No more.  
15 This result may look disappointing but what it says is that every **circular representation**  
16 of an **infinite preproof** that is valid for my quadratic criterion and every proof in my  
17 labelled system can be turned into a **finitary proof** with the same conclusion. And we  
18 provide an effective method to do it. In other words, we provide a method to **synthesize**  
19 **some induction and coinduction invariants**, **finitize** some circular proofs and partially  
20 answer the **Brotherston-Simpson conjecture**.

### 21 1.3.1 Fixed points in linear logic

22 This thesis focuses on circular proofs for MALL with fixed points. Least and greatest  
23 fixed point were already well-established in linear logic [Baelde and Miller, 2007, Baelde,  
24 2012, Baelde, Doumane, and Saurin, 2016, Doumane, 2017b] and we build on this basis  
25 to contribute to several directions in the field of circular proofs:

- 26 1. the relationship between finite and circular proofs (at the level of provability and  
27 at the level of proofs themselves) and
- 28 2. the certification of circular proofs, that is the production of fast and/or small pieces  
29 of evidence to support validity of a circular preproof.

## 1.3.2 Addressed questions, proposed answers

**Proof-checking** Among all circular preproofs, some are valid proofs and some are not. The thread criterion gives a non-ambiguous mathematical definition of whether a circular representation represents a valid (pre)proof or an invalid preproof. The first interesting question is whether there exist a method to tell whether the circular representation of a preproof is valid or not. In other words: is the thread criterion on circular preproofs decidable? Given a finite circular representation of a non-wellfounded preproof, can one decide whether this preproof is valid with respect to the thread criterion?

The answer to that is yes: there is an algorithm which is able to tell the difference between a valid circular representation of a preproof and an invalid one. In fact several such algorithms are known.

The next question would be: how difficult is this problem? What is its computational complexity? When we started to look at this problem, the state of the art was the following: all known algorithms for this problem ran in exponential time and ran or could be made to run in polynomial space. But no subexponential algorithm was known, there was no lower bound on its complexity and the exact complexity of checking the thread criterion was still unknown. Recall that  $P \subseteq NP \subseteq PSPACE \subseteq EXP$ . The first contribution of this thesis is to prove that this problem is in fact PSPACE-complete. This implies in particular that this problem is probably not in NP and that there is probably no subexponential algorithm to solve it.

Our proof is based on a deeper exploration of the connection between thread-validity and the size-change termination principle [Lee, Jones, and Ben-Amram, 2001], which is usually used to ensure program termination.

Circular proofs have already proved useful in implementing efficient automatic provers, *e. g.* the Cyclist prover [Gorogiannis and Rowe, 2014]. The complexity avoided in the search, thanks to the fact that we need not to guess invariants, is counterbalanced by the complexity of the validity criterion at the time of proof checking. It seems however that this approach gives good performances in practice.

**A new validity criterion** Among all representations of valid circular proofs, a new fragment is described in Chapter 4, based on a stronger validity criterion. This new criterion is based on a labelling of formulas and proofs, whose validity is purely local.

This allows this fragment to be easily handled, while being expressive enough to still contain all circular embeddings of Baelde's  $\mu$ MALL finite proofs with (co)inductive invariants: in particular deciding validity and computing a certifying labelling can be done efficiently.

1 **Finitization** Comparing finite and infinite proofs is very natural. Informally, it amounts  
 2 to considering the relative strength of inductive reasoning versus infinite descent: while  
 3 infinite descent is a very old form of mathematical reasoning which appeared already  
 4 in Euclid’s *Elements* and was systematically investigated by Fermat, making precise  
 5 its relationship with mathematical induction is still an open question for many proof  
 6 formalisms. Their equivalence is known as the Brotherston–Simpson conjecture. While it  
 7 is fairly straightforward to check that infinite descent (circular proofs) proves at least as  
 8 many statements as inductive reasoning, the converse is complex and remains largely  
 9 open. A few years ago, Simpson [2017], on the one hand, and Berardi and Tatsuta  
 10 [2017b,a], on the other hand, made progress on this question but only in the framework  
 11 of Martin L of’s inductive definitions, not in the setting of  $\mu$ -calculi circular proofs in  
 12 which invariant extraction is highly complex and known only for some fragments.

13 We will show that the Brotherston-Simpson conjecture holds for the fragment we present:  
 14 every labelled representation of a circular proof in the fragment is translated into a  
 15 standard finitary proof.

16 **Propositions of extensions** Finally we explore how to extend these results to a bigger  
 17 fragment, by relaxing the labelling discipline while retaining (i) the ability to locally  
 18 certify the validity and (ii) to some extent, the ability to finitize circular proofs.

**A complex example** We conclude this introduction by considering a typical example  
 of a circular proof with a complex validating thread structure: while this infinite proof  
 has a regular derivation tree, its branches and threads have a complex geometry. The  
 circular proof of Figure 1.3 derives the sequent  $\vdash F, G, H, I, J$  where

$$F = \mu X((X \wp G) \& (X \wp H)) \quad G = \nu X(X \oplus \perp) \quad H = \nu X(\perp \oplus X)$$

$$I = \mu Z((Z \wp J) \oplus \perp) \quad J = \mu X((K \wp X) \oplus \perp) \quad K = \nu Y \mu Z((Z \wp \mu X(Y \wp X) \oplus \perp) \oplus \perp)$$

19 This example of a circular derivation happens to be valid (it is a  $\mu\text{MALL}^\omega$  proof) but the  
 20 description of its validating threads is quite complex. Indeed, each infinite branch  $\beta$  is  
 21 validated by exactly one thread (see next section for detailed definitions) going through  
 22 either  $G$ ,  $H$  or  $K$  depending on the shape of the branch *at the limit* (infinite branches of  
 23 this derivations can be described as  $\omega$ -words on  $A = \{l, r\}$  depending on whether the left  
 24 or right back edge is taken):

- 25 (i) if  $\beta$  *ultimately* follows always the left cycle ( $A^* \cdot l^\omega$ ), the unfolding of  $H$  validates  $\beta$ ;
- 26 (ii) if  $\beta$  *ultimately* follows always the right cycle ( $A^* \cdot r^\omega$ ), the unfolding of  $G$  validates  $\beta$ ;
- 27 (iii) if  $\beta$  *endlessly switches* between left and right cycles ( $A^* \cdot (r^+ \cdot l^+)^\omega$ ),  $K$  validates  $\beta$ .

28 The description of the thread validating this proof is thus complex. This is reflected in  
 29 the difficulty to provide a local way to validate this proof and in the lack of a general



## 1 Introduction

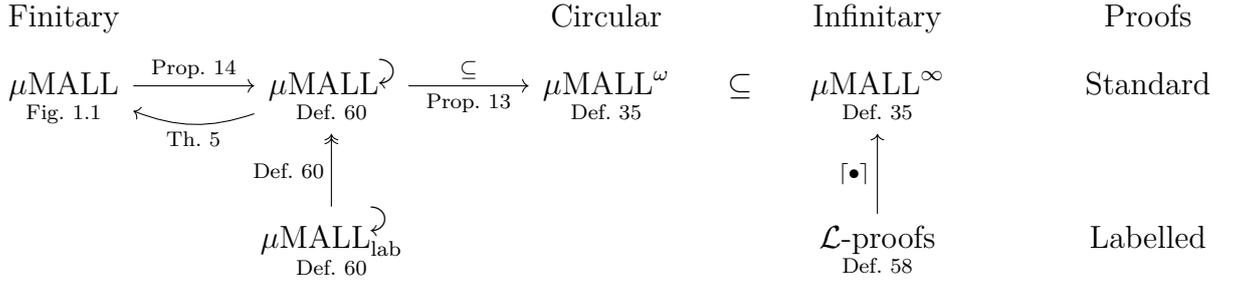


Figure 1.4. Relations between the different systems used in this thesis.

1 which are the true subject of this thesis (Section 2.1.4). We also recall some basic elements  
 2 of the theories of algorithmic complexity (Section 2.2) and of  $\omega$ -automata (Section 2.3).

3 Chapter 3 is concerned with the algorithmic complexity of the thread criterion. We start  
 4 by recalling in Section 3.1 that the thread criterion is effectively decidable in PSPACE.  
 5 The main section of the first technical chapter is Section 3.2, in which we show the  
 6 PSPACE-completeness of the thread criterion for  $\mu\text{MALL}^{\omega}$ , in Theorem 4. Section 3.3  
 7 is devoted to a discussion of our approach and a comparison with related works. We  
 8 conclude this chapter in Section 3.4.

9 Chapter 4 studies an approach to circular proofs based on labellings of greatest fixed  
 10 points. We first motivate in Section 4.1 such labellings as an alternative way to express the  
 11 validating threads. Then, in Section 4.2 we introduce finite representations of preproofs  
 12 and use such labellings in order to locally certify their validity. Finally, in Section 4.3, we  
 13 turn to alternative characterizations of those circular proofs which can be labelled. The  
 14 fragment of labellable proofs, while quite constrained (for instance, it does not include  
 15 the example of Figure 1.3), is already enough to capture the circular proofs obtained by  
 16 translation of  $\mu\text{MALL}$  proofs.

17 In Chapter 5 and Section 5.1, we address the converse: for any labelled derivation tree  
 18 with back edges, we provide a corresponding  $\mu\text{MALL}$  proof by generating a (co)inductive  
 19 invariant based on an inspection of the labelling structure. Therefore, we answer the  
 20 Brotherston–Simpson conjecture in a restricted fragment. In Section 5.2, we introduce  
 21 a more permissive labelling strategy that allows to label more proofs (in particular by  
 22 allowing to loop not only on  $(\nu)$  rules but on any rule) and that still ensures validity of  
 23 the labellable derivations. For this relaxed labelling, we label the example of Figure 1.3  
 24 and show how to finitize it by adapting the method of Section 5.1. Nevertheless, there is  
 25 not yet a general method applicable to the complete extended labelling fragment. We end  
 26 this chapter by a discussion, in Section 5.5, of related works to both Chapters 4 and 5.  
 27 Relations between the various systems considered in these two chapters are summarized  
 28 in Figure 1.4.





# 2 Technical background

## 2.1 Proof theory for least and greatest fixed points

### 2.1.1 Usual proof theory and MALL

Logic is generally built around two main concepts: formulas and proofs.

**Formulas** or *propositions*, model some statements that you may say, and which may, intuitively, be true or false such as “The cat is black” or “Earth is flat”. But logic and proof theory are more interested in understanding the nature of the logical connectives and their dynamic than in what happens to be true in our world. That is why a formal language of formulas is generally built out of some propositional variables and some connectives. For instance a language of formulas for the classical logic LK may be defined by the following grammar:

$$A, B ::= X \mid \bar{X} \mid A \wedge B \mid A \vee B \mid \top \mid \perp$$

where  $X$  ranges over a given set of propositional variables, which we denote by  $X, Y, Z, \dots$

$\wedge$  denotes conjunction, “and”, while  $\vee$  denotes disjunction, “or”. The constants  $\top$  and  $\perp$  represent their respective neutral elements, that is respectively a true statement and a false statement. The variables  $X, Y, Z, \dots$  represent some undetermined statements. That means that, for instance,  $(\bar{X} \vee Y) \wedge (\bar{Y} \vee Z)$  is a formula of LK, which you may read as “not  $X$  or  $Y$ , and not  $Y$  or  $Z$ ”.

The main logic we will use to apply our methods in this thesis is the logic MALL, composed of the Multiplicative and Additive fragments of Linear Logic. Linear logic was designed by Girard [1987] in a successful attempt to decompose the well-known connectives of classical and intuitionistic logics into more elementary components. For a detailed introduction to linear logic, see [Girard, 1987], [Girard, Taylor, and Lafont, 1989] or [Girard, 2011].

## 2 Technical background

1 The formulas of MALL are defined by the following grammar:

**Definition 1** (MALL formulas).

$$A, B ::= X \mid \bar{X} \mid A \otimes B \mid A \wp B \mid \mathbf{1} \mid \perp \mid A \oplus B \mid A \& B \mid \mathbf{0} \mid \top$$

2 where  $X$  ranges over a given set of propositional variables.

3 The set of all MALL formulas is denoted by  $\mathbf{Fml}_{\text{MALL}}$ .

4  $\otimes$  and  $\&$  are conjunctions of different natures and  $\mathbf{1}$  and  $\top$  are their respective neutral  
5 elements.  $\wp$  and  $\oplus$  are disjunctions of different natures and  $\perp$  and  $\mathbf{0}$  are their respective  
6 neutral elements.

7 *Remark 2* (Precedence rules). The usage is that  $\otimes$  and  $\wp$ , the “multiplicative” connectives,  
8 have a higher precedence than  $\oplus$  and  $\&$ , the “additive” connectives. For instance  $\bar{X} \wp$   
9  $\bar{Y} \& \bar{X} \wp \bar{Y}$  should be read as  $(\bar{X} \wp \bar{Y}) \& (\bar{X} \wp \bar{Y})$ .

10 With such a syntax of formulas come a syntax for the positions of subformulas inside  
11 formulas. To that effect we use the following alphabet

12 Let  $A_{\mathbf{Fml}}$  be the following alphabet:

$$\mathcal{P}_{\text{MALL}} = \{\otimes_0, \otimes_1, \wp_0, \wp_1, \oplus_0, \oplus_1, \&_0, \&_1\}$$

13 We denote by  $\mathcal{P}_{\text{MALL}}^*$  the free monoid of finite words on  $\mathcal{P}_{\text{MALL}}$ . Those words are used to  
14 denote the position of a subformula in a formula.

15 **Definition 2** (Positions of subformulas in a formula). The set of positions of subformulas  
16 of a given formula  $A$  is defined by induction on  $A$ :

$$\begin{aligned} \text{SubPos}(X) &= \{\epsilon\} & \text{SubPos}(\bar{X}) &= \{\epsilon\} & \text{SubPos}(\mathbf{1}) &= \{\epsilon\} & \text{SubPos}(\mathbf{0}) &= \{\epsilon\} \\ \text{SubPos}(\perp) &= \{\epsilon\} & \text{SubPos}(\top) &= \{\epsilon\} \\ \text{SubPos}(A \otimes B) &= \{\epsilon\} \cup \otimes_0 \cdot \text{SubPos}(A) \cup \otimes_1 \cdot \text{SubPos}(B) \\ \text{SubPos}(A \wp B) &= \{\epsilon\} \cup \wp_0 \cdot \text{SubPos}(A) \cup \wp_1 \cdot \text{SubPos}(B) \\ \text{SubPos}(A \oplus B) &= \{\epsilon\} \cup \oplus_0 \cdot \text{SubPos}(A) \cup \oplus_1 \cdot \text{SubPos}(B) \\ \text{SubPos}(A \& B) &= \{\epsilon\} \cup \&_0 \cdot \text{SubPos}(A) \cup \&_1 \cdot \text{SubPos}(B) \end{aligned}$$

1 **Example 1.** The set of positions of subformulas of  $X \otimes (\perp \oplus Y)$  is  $\{\epsilon, \otimes_0, \otimes_1 \oplus_0, \otimes_1 \oplus_1\}$ .

2 Those formulas are equipped with an involutive negation, denoted by  $\cdot^\perp$  and defined  
3 inductively as follows:

**Definition 3.**

$$\begin{aligned} X^\perp &= \bar{X} & \bar{X}^\perp &= X & (A \otimes B)^\perp &= A^\perp \wp B^\perp & (A \wp B)^\perp &= A^\perp \otimes B^\perp & \mathbf{1}^\perp &= \perp \\ \perp^\perp &= \mathbf{1} & (A \oplus B)^\perp &= A^\perp \& B^\perp & (A \& B)^\perp &= A^\perp \oplus B^\perp & \mathbf{0}^\perp &= \top & \top^\perp &= \mathbf{0} \end{aligned}$$

4 **Proofs** are certificates used to attest the truth of a formula. In this thesis, proofs are  
5 built in sequent calculus, following the usage of modern proof theory, initiated by Gentzen  
6 [1935a,b, 1969].

7 In sequent calculus, a proof is a finite tree built out of some given deduction rules. We  
8 recall the basics of that setting. We will exemplify it first on the particular case of MALL,  
9 then on MALL extended with least and greatest fixed point in the system  $\mu$ MALL. For  
10 a more detailed introduction to sequent calculus, you may look at [Girard, Taylor, and  
11 Lafont, 1989] or [Girard, 1991].

12 **Definition 4** (Sequent). Given a set of formulas **Fml**, a *sequent* is defined to be a finite  
13 list of formulas. If  $\Gamma$  is a sequent of size  $n$ , we use the notation  $|\Gamma|$  to denote either  $n$ , the  
14 size of the list, or the set  $\{0, \dots, n-1\}$ , the set of indices of this list.<sup>1</sup>

15 A sequent is denoted by the symbol  $\vdash$  followed by its formulas. A sequent should be  
16 understood, intuitively, as the disjunction, in the  $\wp$  sense, of its formulas.

17 A proof is a tree made out of some logical inferences.

18 **Definition 5** (Inference). An *inference* consists of :

- 19 • A finite number of sequents,  $\Gamma_0, \dots, \Gamma_{n-1}$ , which are called the *premises* of the  
20 inference
- 21 • A sequent  $\Gamma$ , which is called the *conclusion* of the inference

---

<sup>1</sup>This small abuse of notation is nothing more than the usual set-theoretic convention of identifying an integer (or an ordinal) with the set of integers (or ordinals) that are smaller than it:  $n = \{m \mid m < n\}$

## 2 Technical background

$$\begin{array}{cccc}
\frac{}{\vdash A, A^\perp} \text{(id)} & & \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \text{(cut)} & \\
\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \text{(\otimes)} & \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} \text{(\wp)} & \frac{}{\vdash \mathbf{1}} \text{(1)} & \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \text{(\perp)} \\
\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \text{(\oplus_0)} & \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \text{(\oplus_1)} & \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \text{(\&)} & \frac{}{\vdash \Gamma, \top} \text{(\top)}
\end{array}$$

Figure 2.1. MALL inference rules

- A partial function

$$\sigma: |\Gamma_0| + \dots + |\Gamma_{n-1}| \rightarrow |\Gamma| \times \mathcal{P}^*$$

where  $|\Gamma_0| + \dots + |\Gamma_{n-1}|$  denotes the usual set-theoretic sum

$$\{0\} \times |\Gamma_0| \cup \dots \cup \{n-1\} \times |\Gamma_{n-1}|.$$

This function is called the *threading function*. It can equivalently be described by  $n$  partial functions

$$\sigma_0: |\Gamma_0| \rightarrow |\Gamma| \times \mathcal{P}^* \quad \dots \quad \sigma_{n-1}: |\Gamma_{n-1}| \rightarrow |\Gamma| \times \mathcal{P}^*$$

This inference is written

$$\frac{\Gamma_0 \quad \dots \quad \Gamma_{n-1}}{\Gamma}$$

- 1 with the *threading function*  $\sigma$  generally left implicit.
- 2 The inference rules of MALL are given on Figure 2.1.
- 3 For instance, the rule

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \text{(\otimes)}$$

- 4 should be understood, intuitively, as saying that for every lists of formulas  $\Gamma, \Delta$  and
- 5 for every formulas  $A, B$ , whenever the sequents  $\vdash \Gamma, A$  and  $\vdash \Delta, B$  are true, so is
- 6  $\vdash \Gamma, \Delta, A \otimes B$ .

Formally, it should also be understood by this rule that in a  $(\otimes)$  inference, the function  $\sigma$  sends any position of formula in  $\Gamma$  or  $\Delta$  in the premises to the corresponding position in the conclusion and the empty path, sends the position of  $A$  in the first premise on the position of  $A \otimes B$  in the conclusion and the path  $\otimes_0$  and sends the position of  $B$  in the second premise on the position of  $A \otimes B$  in the conclusion and the path  $\otimes_1$ . Meanwhile, the order of the formulas in the sequents is not imposed by the rule. For instance, this is a correct  $(\otimes)$  inference:

$$\frac{\vdash X \wp \overline{X}, X \quad \vdash X, Y, Z}{\vdash X \otimes Y, X \wp \overline{X}, X, Z} (\otimes)$$

with the implicit function

$$\begin{aligned} \sigma: \{0\} \times \{0, 1\} \cup \{1\} \times \{0, 1, 2\} &\rightarrow \{0, 1, 2, 3\} \times A_{\mathbf{Fml}}^* \\ (0, 0) &\mapsto (1, \epsilon) \\ (0, 1) &\mapsto (0, \otimes_0) \\ (1, 0) &\mapsto (2, \epsilon) \\ (1, 1) &\mapsto (0, \otimes_1) \\ (1, 2) &\mapsto (3, \epsilon) \end{aligned}$$

<sup>1</sup> Here this function happens to be total. The possibility to be partial is used in the  $(\text{cut})$   
<sup>2</sup> rule.

<sup>3</sup> **Definition 6** (Proof tree). A *proof* for MALL is any finite proof tree built with these  
<sup>4</sup> inference rules.

<sup>5</sup> Here is an example:

**Example 2.**

$$\frac{\frac{\frac{\overline{\vdash \overline{X}, X}}{\vdash \overline{X}, X} (\text{id}) \quad \frac{\overline{\vdash \overline{Y}, Y}}{\vdash \overline{Y}, Y \oplus Z} (\oplus_0)}{\vdash \overline{X}, \overline{Y}, X \otimes (Y \oplus Z)} (\otimes)}{\vdash \overline{X} \wp \overline{Y}, X \otimes (Y \oplus Z)} (\wp) \quad \frac{\frac{\overline{\vdash \overline{Z}, Z}}{\vdash \overline{Z}, Z} (\text{id}) \quad \frac{\overline{\vdash \overline{Z}, Z}}{\vdash \overline{Z}, Y \oplus Z} (\oplus_1)}{\vdash \overline{X}, \overline{Z}, X \otimes (Y \oplus Z)} (\otimes)}{\vdash \overline{X} \wp \overline{Z}, X \otimes (Y \oplus Z)} (\wp)}{\vdash \overline{X} \wp \overline{Y} \& \overline{X} \wp \overline{Z}, X \otimes (Y \oplus Z)} (\&)$$

<sup>6</sup> One of the interests of the *threading functions* is that they can be composed.

## 2 Technical background

**Definition 7** (composition of threading functions). If  $A, B, C$  are three sets and  $\sigma: A \rightarrow B \times A_{\mathbf{Fml}}^*$  and  $\rho: B \rightarrow C \times A_{\mathbf{Fml}}^*$  are two partial functions, we define their composite

$$\begin{aligned} \rho \circ \sigma: A &\rightarrow C \times A_{\mathbf{Fml}}^* \\ a &\mapsto (c, v \cdot u) \quad \text{where } (b, u) = \sigma(a) \text{ and } (c, v) = \rho(b) \end{aligned}$$

<sup>1</sup> *Remark 3.* The interested reader may recognize the Kleisli category of the monad  $\_ \times A_{\mathbf{Fml}}^*$   
<sup>2</sup> on  $\text{Set}$ , whose multiplication and unit are induced by the monoidal structure of  $A_{\mathbf{Fml}}^*$ .

<sup>3</sup> This fact allows us to extend this definition in order to relate any two sequents which are  
<sup>4</sup> connected by a chain of successive inferences:

**Definition 8** (Threading function associated to a path in a proof tree). Suppose a proof tree  $\pi$ . Suppose a sequence  $u = s_0, s_1, \dots, s_n$  of positions of sequents in  $\pi$  and a sequence  $r_0, r_1, \dots, r_{n-1}$  of inferences in  $\pi$  such that  $\forall i \in \llbracket 0, n-1 \rrbracket$ ,  $s_i$  is the conclusion of  $r_i$  and  $s_{i+1}$  is a premise of  $r_i$ .  $\forall i$ , let  $\Gamma_i$  be the sequent at position  $s_i$  in  $\pi$ . Then, by composing the threading functions of  $r_0, r_1, \dots, r_{n-1}$ , we get a function

$$\sigma(u): |\Gamma_n| \rightarrow |\Gamma_0| \times \mathcal{P}^*$$

**Example 3.** For instance, if we denote by  $u$  the following sequence of sequents and inferences, taken from the right of Example 2:

$$\frac{\frac{\frac{\frac{\frac{\vdash \bar{Z}, Y}{\vdash \bar{Z}, Y \oplus Z}^{(\oplus_1)}}{\vdash \bar{X}, \bar{Z}, X \otimes (Y \oplus Z)}^{(\otimes)}}{\vdash \bar{X} \wp \bar{Z}, X \otimes (Y \oplus Z)}^{(\wp)}}{\vdash \bar{X} \wp \bar{Y} \& \bar{X} \wp \bar{Z}, X \otimes (Y \oplus Z)}^{(\&)}}{\vdash \bar{X}, Y}^{(\oplus_1)}$$

Then we have

$$\begin{aligned} \sigma(u): |\bar{Z}, Y| &\rightarrow |\bar{X} \wp \bar{Y} \& \bar{X} \wp \bar{Z}, X \otimes (Y \oplus Z)| \\ 0 &\mapsto (0, \&_1 \wp_1) \\ 1 &\mapsto (1, \oplus_1 \oplus_1) \end{aligned}$$

**Definition 9** (Image of a subformula in an inference). Suppose a proof tree  $\pi$  and an inference  $r = \frac{\Gamma_0 \dots \Gamma_{n-1}}{\Gamma}$  in  $\pi$ . Let us denote by  $\sigma(r)$  the threading function of  $r$ . If  $\Gamma_i$  is one of its premises, if  $A$  is a formula in  $\Gamma_i$ , at position  $f \in |\Gamma_i|$ , if  $p \in \text{SubPos}(A)$  is

a position of subformula in  $A$ , then to this position  $f \cdot p$  of subformula in  $\Gamma_i$  is associated, when possible, a position of subformula in  $\Gamma$ , denoted by  $\mathfrak{t}(r)(f \cdot p)$  and defined as

$$\mathfrak{t}(r)(f \cdot p) := g \cdot q \cdot p \quad \text{where } (g, q) := \sigma(r)$$

1 This makes  $\mathfrak{t}(r)$  a partial function from positions of subformulas in the premises of  $r$  to  
2 positions of subformulas in the conclusion of  $r$ .

3 Similarly to what we did for threading functions in Definition 8, we can extend this  
4 definition of  $\mathfrak{t}$  from rules to paths in a proof tree:

5 **Definition 10** (Image of a subformula through a path in a proof tree). Suppose a  
6 proof tree  $\pi$ . Suppose a sequence  $u = (s_0, s_1, \dots, s_n)$  of positions of sequents in  $\pi$  and a  
7 sequence  $r_0, r_1, \dots, r_{n-1}$  of inferences in  $\pi$  such that  $\forall i \in \llbracket 0, n-1 \rrbracket$ ,  $s_i$  is the conclusion  
8 of  $r_i$  and  $s_{i+1}$  is a premise of  $r_i$ .  $\forall i$ , let  $\Gamma_i$  be the sequent at position  $s_i$  in  $\pi$ . Then, by  
9 composing the  $\mathfrak{t}(r_i)$ , we get a partial function from the position of subformulas in  $\Gamma_n$  to  
10 the positions of subformulas in  $\Gamma_0$ , which we denote by  $\mathfrak{t}(u)$ .

11 **Proposition 2.** *With the notations of the previous definition, for every formula  $A$  in*  
12  *$\Gamma_n$ , at position  $f \in |\Gamma_n|$  and every position of of subformula  $p \in \text{SubPos}(A)$ , if  $\mathfrak{t}(u)(f \cdot p)$*   
13 *is defined, then the subformula at position  $\mathfrak{t}(u)(f \cdot p)$  in  $\Gamma_0$  is identical to the subformula*  
14 *at position  $f \cdot p$  in  $\Gamma_n$ .*

15 *Proof.* Immediate by induction on the length of  $u$  and examination of each possible  
16 rule. The threading functions have been chosen precisely to witness the subformula  
17 property.  $\square$

**Definition 11** (Path of subformula images). Suppose a proof tree  $\pi$ . Suppose a sequence  
 $u = (s_0, s_1, \dots, s_n)$  of positions of sequents in  $\pi$  and a sequence  $r_0, r_1, \dots, r_{n-1}$  of inferences  
in  $\pi$  such that  $\forall i \in \llbracket 0, n-1 \rrbracket$ ,  $s_i$  is the conclusion of  $r_i$  and  $s_{i+1}$  is a premise of  $r_i$ .  $\forall i$ , let  
 $\Gamma_i$  be the sequent at position  $s_i$  in  $\pi$  and let  $p$  be a subformula in one of the formulas of  
 $\Gamma_n$ . We define a sequence of positions of subformulas  $p_{i_0}, \dots, p_n$ , such that  $\forall i \geq i_0$ ,  $p_i$  is  
a subformula of a formula of  $\Gamma_i$ . This sequence is defined as follows:

$$\begin{aligned} p_n &:= p \\ \forall i, p_{i-1} &:= \mathfrak{t}(r_{i-1})(p_i) \end{aligned}$$

or, equivalently:

$$\forall i, p_i := \mathfrak{t}(s_i, \dots, s_n)(p)$$

The last defined index of this sequence,  $i_0$ , is either 0 or the first  $i$  such that  $\mathfrak{t}(r_{i-1})(p_i)$  is  
not defined, which happens if  $r_{i-1}$  is a cut and  $p_i$  is in a cut formula. This sequence is  
denoted by

$$\mathfrak{T}(u)(p) := p_{i_0}, \dots, p_n$$

## 1 2.1.2 Finite proof trees: $\mu$ MALL

2 In this section, we introduce the logic  $\mu$ MALL [Baelde, 2012], its formulas and proofs,  
3 which are finite sequent calculus proofs.

4  **$\mu$ MALL formulas** The grammar of formulas is obtained by extending the formulas of  
5 MALL with two fixed-point connectives,  $\mu$  and  $\nu$ , denoting respectively least and greatest  
6 fixed points of formulas.

7 Formulas of  $\mu$ MALL are selected among a set of preformulas. Preformulas of  $\mu$ MALL  
8 are obtained by taking the usual formulas of MALL and adding two monadic second  
9 order binders,  $\mu$  and  $\nu$ :

**Definition 12** ( *$\mu$ MALL preformulas*).

$$A, B ::= X \mid A \otimes B \mid A \wp B \mid \mathbf{1} \mid \perp \mid A \oplus B \mid A \& B \mid \mathbf{0} \mid \top \mid \mu X A \mid \nu X A$$

10 where  $X$  ranges over an infinite set of propositional variables.

11 The connectives  $\mu$  and  $\nu$  are binders and, as usual, *preformulas* are considered modulo  
12 renaming of bound variables. For instance,  $\nu X(X \otimes X)$  and  $\nu Y(Y \otimes Y)$  denote the same  
13 *preformula*.

14 **Definition 13** ( *$\mu$ MALL formulas*). A *formula* is a closed *preformula*. We denote by  
15 **Fml** the set of all formulas and we denote by  $\leq$  the usual subformula ordering between  
16 *formulas* and *preformulas*.

17 **Definition 14** ( *$\mu$ MALL negation*). An involutive negation  $\cdot^\perp$  is defined on every  $\mu$ MALL  
18 *preformula*, inductively specified by:

$$\begin{array}{lll} (A \otimes B)^\perp = A^\perp \wp B^\perp & \mathbf{1}^\perp = \perp & X^\perp = X \\ (A \oplus B)^\perp = A^\perp \& B^\perp & \mathbf{0}^\perp = \top \\ & & (\mu X A)^\perp = \nu X A^\perp \end{array}$$

19 **Example 4.** If  $A$  is any *formula* and  $F = \nu X(\mu Y((A \otimes X) \wp Y))$  then  $F^\perp = \mu X(\nu Y((A^\perp \wp$   
20  $X) \otimes Y))$ .

21 *Remark 4.* It may be counterintuitive that  $X^\perp = X$ . Yet, in practice negation will only  
22 be applied to *formulas*, which are *closed preformulas*. This simple hack allows us to avoid

$$\frac{\vdash \Gamma, A[\mu X A[X]]}{\vdash \Gamma, \mu X A[X]}^{(\mu)} \qquad \frac{\vdash \Gamma, B \vdash B^\perp, A[B]}{\vdash \Gamma, \nu X A[X]}^{(\nu_{\text{inv}})}$$

 Figure 2.2.  $\mu$ MALL rules for  $\mu$  and  $\nu$ 

$$\frac{\vdash \Gamma, A[\mu X A[X]]}{\vdash \Gamma, \mu X A[X]}^{(\mu)} \qquad \frac{\vdash \Gamma, B \vdash B^\perp, A[B]}{\vdash \Gamma, \nu X A[X]}^{(\nu_{\text{inv}})}$$

 Figure 2.3. Threading functions for the  $(\mu)$  and  $(\nu_{\text{inv}})$  rules of  $\mu$ MALL

1 the use of negative atoms  $\bar{X}, \bar{Y}, \dots$ . The fact that we have only positive atoms guarantees  
 2 in turn that bound variables can only appear in covariant position, thus avoiding the  
 3 need for a positivity condition when forming a fixed point formula.

4 In the presence of negative atoms, negation should be defined by:

$$\begin{aligned} X^\perp &= \bar{X} & \bar{X}^\perp &= X \\ (\mu X A[X])^\perp &= \nu \bar{X} A[X]^\perp & (\nu X A[X])^\perp &= \mu \bar{X} A[X]^\perp \end{aligned}$$

5 which does not change its definition on closed formulas. For instance, in Example 4 we  
 6 would still have:

$$\begin{aligned} F^\perp &= (\nu X (\mu Y ((A \otimes X) \wp Y)))^\perp \\ &= \mu \bar{X} (\mu Y ((A \otimes X) \wp Y))^\perp \\ &= \mu \bar{X} (\nu \bar{Y} ((A \otimes X) \wp Y)^\perp) \\ &= \mu \bar{X} (\nu \bar{Y} ((A^\perp \wp \bar{X}) \otimes \bar{Y})) \\ &= \mu X (\nu Y ((A^\perp \wp X) \otimes Y)) \end{aligned}$$

7  **$\mu$ MALL inferences and proofs** The proofs for  $\mu$ MALL are standard sequent calculus  
 8 proofs, as described in Section 2.1.1. They are obtained by extending the inference  
 9 system for MALL, in Figure 2.1 with two rules. Those rules reflect the definitions of  
 10 least and greatest fixed points of non-decreasing function on ordered sets that we gave in  
 11 Section 1.1.1.

12 **Definition 15.** The set of rules of the sequent calculus  $\mu$ MALL is the union of the rules  
 13 for MALL, given in Figure 2.1 and of the two rules given in Figures 2.2 and 2.3.

## 2 Technical background

1 *Remark 5.* As it was mentioned in the informal introduction, in Section 1.2.3, the  $(\nu_{\text{inv}})$   
 2 rule of  $\mu\text{MALL}$  sequent calculus breaks the subformula property. Indeed, the  $B$  and  $B^\perp$   
 3 that appears in the premises of the rule, on Figure 2.2, do not appear in its conclusion.  
 4 This means that anyone wanting to prove a  $\nu$  formula by applying that rule has to find  
 5 an appropriate  $B$ , which will act as an induction or coinduction invariant.

6 We recall two properties of  $\mu\text{MALL}$ , that will be used in Chapter 5. The first one is  
 7 called *functoriality*:

**Proposition 3.** *For every variable  $X$  and every formulas  $A[X], B, C$  of  $\mu\text{MALL}$ , the following rule is derivable in  $\mu\text{MALL}$ :*

$$\frac{\vdash B, C}{\vdash A^\perp[B], A[C]} [A]$$

8 *Proof.* By induction on  $A$ . See [Baelde, 2012, Section 2.4] for a full proof.  $\square$

9 The second one asserts the two following equivalences:

$$\nu X A[B[X]] \circ\!\circ\! A[\nu X B[A[X]]] \qquad \mu X A[B[X]] \circ\!\circ\! A[\mu X B[A[X]]]$$

10 **Proposition 4.** *For every two formulas  $A[X], B[X]$  of  $\mu\text{MALL}$ , the following sequents*  
 11 *are derivable in  $\mu\text{MALL}$ :*

$$\vdash \nu X A^\perp[B^\perp[X]], A[\mu X B[A[X]]] \qquad \vdash \mu X A^\perp[B^\perp[X]], A[\nu X B[A[X]]]$$

*Proof.*

$$\Pi_0(A, B) = \frac{\frac{\frac{\vdash B^\perp[A^\perp[\nu X B^\perp[A^\perp[X]]]], B[A[\mu X B[A[X]]]]}{\vdash B^\perp[A^\perp[\nu X B^\perp[A^\perp[X]]]], \mu X B[A[X]]} (\mu)}{\vdash A^\perp[B^\perp[A^\perp[\nu X B^\perp[A^\perp[X]]]], A[\mu X B[A[X]]]} [A]} (\nu_{\text{inv}}^0)$$

$$\frac{\vdash \Gamma, A[\mu X A[X]]}{\vdash \Gamma, \mu X A[X]} (\mu) \qquad \frac{\vdash \Gamma, A[\nu X A[X]]}{\vdash \Gamma, \nu X A[X]} (\nu)$$

 Figure 2.4.  $\mu\text{MALL}^\infty$  rules for  $\mu$  and  $\nu$ 

$$\frac{\vdash \Gamma, A[\mu X A[X]]}{\vdash \Gamma, \mu X A[X]} (\mu) \qquad \frac{\vdash \Gamma, A[\nu X A[X]]}{\vdash \Gamma, \nu X A[X]} (\nu)$$

 Figure 2.5. Threading functions for the  $(\mu)$  and  $(\nu)$  rules of  $\mu\text{MALL}^\infty$ 

$$\Pi_1(A, B) = \frac{\begin{array}{c} \vdots \Pi_0(B^\perp, A^\perp) \\ \vdash B^\perp[\mu X A^\perp[B^\perp[X]]], \nu X B[A[X]] \end{array}}{\frac{\vdash A^\perp[B^\perp[\mu X A^\perp[B^\perp[X]]]], A[\nu X B[A[X]]]}{\vdash \mu X A^\perp[B^\perp[X]], A[\nu X B[A[X]]]} (\mu)} [A]$$

1

□

### 2.1.3 Infinite proof trees: $\mu\text{MALL}^\infty$

$\mu\text{MALL}^\infty$  [Baelde, Doumane, and Saurin, 2016, Doumane, 2017b] is a non well-founded proof system for an extension of MALL with least and greatest fixed points operators. It was designed to fix the defects of the  $(\nu_{\text{inv}})$  rule of  $\mu\text{MALL}$ .

**Definition 16.** The formulas of  $\mu\text{MALL}^\infty$  are the same as the formulas of  $\mu\text{MALL}$ , in the previous section.

**Inference rules for  $\mu\text{MALL}^\infty$ .** The proofs for  $\mu\text{MALL}^\infty$  will be defined as infinite, non well-founded proof trees. They are built out of a set of inference rules, which is given in the following definition.

**Definition 17** ( $\mu\text{MALL}^\infty$  inference rules). The set of inference rules for  $\mu\text{MALL}^\infty$  is the union of the rules for MALL, given in Figure 2.1 and of the two rules given in Figures 2.4 and 2.5.



1 **Proofs and validity for  $\mu\text{MALL}^\infty$**  The validity criterion used to distinguish proofs  
 2 among preproofs will be given in Definition 35 and can be stated as: “every infinite  
 3 branch must contain a valid thread”. To make this formal, we will first define how a  
 4 preproof induces two graphs and then define the “branches” and “threads” of a preproof  
 5 as infinite paths in these graphs.

6 **Definition 19** (thread of an infinite branch of a preproof). Given an infinite branch  $\beta$   
 7 of a preproof  $\pi$ , we define the set of threads of  $\beta$  as follows. A thread in  $\beta$  is a sequence  
 8  $(s_n)_{n \in \mathbf{N}}$  of positions of formulas in  $\pi$  such that:

- 9 • each  $s_n$  belongs to a sequent of  $\beta$
- 10 •  $\forall n \in \mathbf{N}$ ,  $s_n$  is the immediate descendant of  $s_{n+1}$  in  $\beta$ , according to the threading  
 11 functions of  $\beta$ .

12 The following lemma is the key to the notion of a valid thread, which is defined right  
 13 after it. If  $s$  is a position of formula in a proof tree, we denote by  $\mathbf{fml}(s) \in \mathbf{Fml}$  the  
 14 associated formula.

15 **Definition 20.** If  $t = (s_n)_{n \in \mathbf{N}}$  is an infinite thread in a preproof, we define  $\text{inf}(t) =$   
 16  $\{A \in \mathbf{Fml} \mid \forall n_0 \in \mathbf{N}, \exists n \geq n_0, s_n \text{ is principal and } \mathbf{fml}(s_n) = A\}$  *i. e.* the set of formulas  
 17 that are infinitely often principal in  $t$ .

18 **Definition 21** (Valid thread). An infinite thread  $t$  is *valid* if  $\text{inf}(t)$  has a minimum, with  
 19 respect to the subformula ordering, and this minimum is a  $\nu$ -formula.

20 **Definition 22** (proofs). We say that an infinite branch  $b$  of a preproof  $\varpi$  is valid if there  
 21 is a valid infinite thread  $t$  on  $b$ .

22 A  $\mu\text{MALL}^\infty$  preproof  $\varpi$  is a proof if all its infinite branches are valid.

### 23 2.1.4 Finite proof trees with back edges: $\mu\text{MALL}^\omega$

24 The aim of  $\mu\text{MALL}^\omega$  is to have a system of finite proofs, as in  $\mu\text{MALL}$ , with only the rules  
 25 of fixed point unfolding instead of a rule with an explicit invariant, as in  $\mu\text{MALL}^\infty$ .

26 As well as in  $\mu\text{MALL}^\infty$ , proofs of  $\mu\text{MALL}^\omega$  are selected among a set of preproofs.  
 27 Preproofs of  $\mu\text{MALL}^\omega$  are circular objects, defined by adding back edges to ordinary  
 28 proof trees.

## 2 Technical background

1 **Definition 23** ( $\mu\text{MALL}^\omega$  formulas). The formulas of  $\mu\text{MALL}^\omega$  are the same as the  
 2 formulas of  $\mu\text{MALL}$  and  $\mu\text{MALL}^\infty$  (Definition 13).

3 **Definition 24** (Proof tree with open sequents). A *proof tree with open sequents* is a  
 4 proof tree in which it is not mandatory for a sequent to be the conclusion of a logical  
 5 inference. Those positions of sequents that are not the conclusion of a logical inference  
 6 are called *positions of open sequents*. The other positions are called *positions of closed*  
 7 *sequents*.

8 **Definition 25** (lower, closer to the root). Whenever  $v, v'$  are two vertices of a tree, we  
 9 say that  $v$  is *lower* than  $v'$ , or that  $v$  is *below*  $v'$ , or that  $v$  is *closer to the root* than  $v'$   
 10 when  $v$  appears on the path from  $v'$  to the root.

11 The following definition aims at defining some representation for the fragment of all  
 12  $\mu\text{MALL}^\infty$  preproofs that are regular, that is those that have only a finite number of  
 13 subtrees up to isomorphism.

14 **Definition 26** ( $\Pi_0(\mu\text{MALL}^\omega)$ :  $\mu\text{MALL}^\omega$  b.e.-tree preproofs). A  *$\mu\text{MALL}^\omega$  b.e.-tree pre-*  
 15 *proof* consists of a finite proof tree with open sequents  $\pi$  (Def. 24), composed using the  
 16 rules of  $\mu\text{MALL}^\infty$  given above in Def. 17, together with a function **back**, which associates  
 17 to each *position  $s$  of an open sequent* in  $\pi$ , a position **back**( $s$ ) of the same sequent in  $\pi$ ,  
 18 such that **back**( $s$ ) is strictly *below*  $s$  in  $\pi$ , *i. e.* between  $s$  and the root. We denote by  
 19  $\Pi_0(\mu\text{MALL}^\omega)$  the set of all  $\mu\text{MALL}^\omega$  b.e.-tree preproofs.

20 We will sometimes say simply preproof instead of b.e.-tree preproof, or circular preproof  
 21 representation.

22 **Example 6.** Let  $\pi$  be the following proof tree, with three *open sequents*, and let us  
 23 denote by  $s_0, \dots, s_8$  the positions of its sequents, as indicated:

$$\begin{array}{c}
 \frac{\frac{\frac{\frac{s_6: \vdash \nu XX, \nu X(X \wp X), \mu XX}{s_4: \vdash \nu X(X \wp X), \mu XX} \quad \frac{s_5: \vdash \nu XX, \nu X(X \wp X), \mu XX}{s_3: \vdash \nu X(X \wp X), \nu X(X \wp X), \mu XX} \quad (\nu)}{s_2: \vdash \nu X(X \wp X) \wp \nu X(X \wp X), \mu XX} \quad (\wp)}{s_1: \vdash \nu X(X \wp X), \mu XX} \quad (\nu)}{s_0: \vdash \nu X(X \wp X)} \quad (\text{cut})}{s_7: \vdash \nu XX} \quad (\nu)}{s_8: \vdash \nu XX} \quad (\nu)}
 \end{array}$$

24 then  $(\pi, \{s_4 \mapsto s_1, s_6 \mapsto s_5, s_8 \mapsto s_7\})$  is a  $\mu\text{MALL}^\omega$  b.e.-tree preproof, that we will more



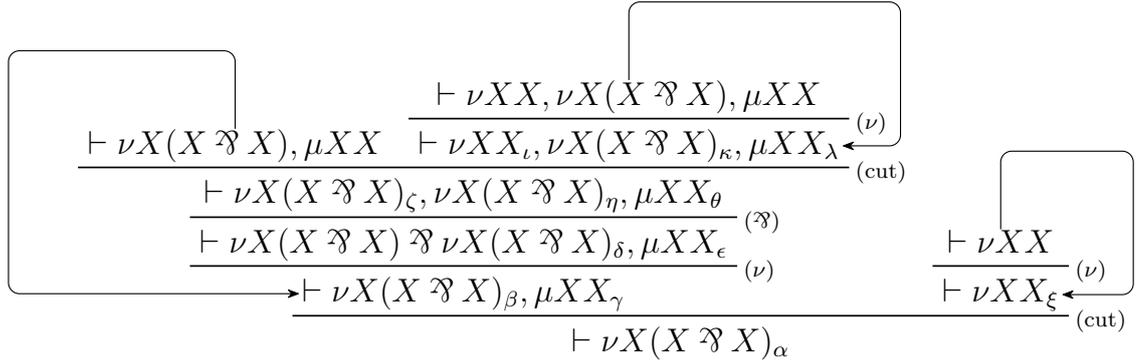
## 2 Technical background

1 Note that, in order to be totally rigorous, we should not only give the vertices of the  
 2 paths but also the edges, *i. e.* when an inference has several premises, indicate explicitly  
 3 which one was chosen. These details are omitted here for concision; they will cause no  
 4 ambiguity on the validity of the preproof of Example 6.

5 **Definition 30** ( $G_{\text{thread}}^{\text{out}}$ , thread graph of a preproof). Let  $(\pi, \mathbf{back})$  be a  $\mu\text{MALL}^\omega$  b.e.-  
 6 tree preproof. Its thread graph is the graph  $G_{\text{thread}}^{\text{out}}$  defined as follows. The vertices of  
 7  $G_{\text{thread}}^{\text{out}}$  are the positions of formulas in the closed sequents of  $\pi$ . For each inference  $I$  with  
 8 conclusion  $s$  in  $\pi$ , for each premise  $s'$  of  $I$  and for each position of formula  $\beta$  in  $s'$  which  
 9 has an immediate descendent  $\alpha$  in  $s$ , there is an edge in  $G_{\text{thread}}^{\text{out}}$ , from  $\alpha$  to  $\beta$  if  $s'$  is a  
 10 closed position of sequent in  $\pi$ , and from  $\alpha$  to the position of the formula corresponding  
 11 to  $\beta$  in  $\mathbf{back}(s')$  if  $s'$  is an open position of sequent in  $\pi$ .

12 **Definition 31** (thread). A *thread* in a b.e.-tree preproof is simply a path (finite or  
 13 infinite) in  $G_{\text{thread}}^{\text{out}}$ .

14 **Example 8.** Let us denote by  $\{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \xi\}$  the vertices of  $G_{\text{thread}}^{\text{out}}$  for the  
 15 preproof shown on Example 6, as indicated here:



16 The maximal threads of this preproof, that is those that cannot be extended in any  
 17 direction, are  $(\xi)^\omega$ ,  $\gamma\epsilon\theta(\lambda)^\omega$ ,  $(\iota)^\omega$ ,  $\alpha(\beta\delta\zeta)^\omega$  and the elements of  $\{\alpha(\beta\delta\zeta)^k\beta\delta\eta(\kappa)^\omega \mid k \in \mathbf{N}\}$ .

18 Once again, in order to be totally rigorous, we should explicitly include the explicit choice  
 19 of a descendent relation, in cases where a position has several immediate ancestors.

20 **Definition 32** ( $\mathbf{U}: G_{\text{thread}}^{\text{out}} \rightarrow G_{\text{branch}}$ ). For any b.e.-tree preproof, there is an obvious  
 21 graph morphism from  $G_{\text{thread}}^{\text{out}}$  to  $G_{\text{branch}}$ , associating to every position of a formula the  
 22 sequent position it belongs to. We denote this graph morphism by  $\mathbf{U}$ . If  $t$  is a path in  
 23  $G_{\text{thread}}^{\text{out}}$  (*i. e.* an outer thread), we will also denote by  $\mathbf{U}(t)$  the corresponding path in  
 24  $G_{\text{branch}}$ .

1 **Example 9.** The images, by the morphism of Definition 32, of the threads of Example 8  
 2 are, with the notations of Example 6:

$$\begin{aligned} \mathbf{U}((\xi)^\omega) &= (s_7)^\omega & \mathbf{U}(\gamma\epsilon\theta(\lambda)^\omega) &= s_1s_2s_3(s_5)^\omega & \mathbf{U}((\iota)^\omega) &= (s_5)^\omega \\ \mathbf{U}(\alpha(\beta\delta\zeta)^\omega) &= s_0(s_1s_2s_3)^\omega & \forall k \in \mathbf{N}, \mathbf{U}(\alpha(\beta\delta\zeta)^k\beta\delta\eta(\kappa)^\omega) &= s_0(s_1s_2s_3)^{k+1}(s_5)^\omega \end{aligned}$$

3 The definition of a valid thread is then the same as in  $\mu\text{MALL}^\infty$ :

4 **Definition 33.** If  $t = (s_n)_{n \in \mathbf{N}}$  is an infinite thread in a b.e.-tree preproof, we define  
 5  $\text{inf}(t) = \{A \in \mathbf{Fml} \mid \forall n_0 \in \mathbf{N}, \exists n \geq n_0, s_n \text{ is principal and } \mathbf{fml}(s_n) = A\}$  *i. e.* the set of  
 6 formulas that are infinitely often principal in  $t$ .

7 **Definition 34** (Valid thread). An infinite thread  $t$  is *valid* if  $\text{inf}(t)$  has a minimum, with  
 8 respect to the subformula ordering, and this minimum is a  $\nu$ -formula.

9 **Example 10.** Among the threads of Example 8:

- 10 •  $(\xi)^\omega$  is valid: its smallest infinitely principal formula is  $\nu XX$ , which is principal at  
 11  $\xi$ ;
- 12 •  $(\iota)^\omega$  is valid: its smallest infinitely principal formula is  $\nu XX$ , which is principal at  
 13  $\iota$ ;
- 14 •  $\alpha(\beta\delta\zeta)^\omega$  is valid: its smallest infinitely principal formula is  $\nu X(X \wp X)$ , which is  
 15 principal at  $\beta$ ;
- 16 •  $\gamma\epsilon\theta(\lambda)^\omega$  is not valid: it has no principal formula;
- 17 •  $\forall k \in \mathbf{N}, \alpha(\beta\delta\zeta)^k\beta\delta\eta(\kappa)^\omega$  is not valid: it has no principal formula after the last  
 18 position of  $\beta$ .

19 **Definition 35** (proofs). We say that an infinite branch  $b$  of a b.e.-tree preproof  $\varpi$  is  
 20 valid if there is a valid infinite thread  $t$  of  $\varpi$  such that  $\mathbf{U}(t)$  is a suffix of  $b$ .

21 A  $\mu\text{MALL}^\omega$  b.e.-tree preproof  $\varpi$  is a *proof* if all its infinite branches are valid.

22 We denote by  $\Pi(\mu\text{MALL}^\omega)$  the set of all  $\mu\text{MALL}^\omega$  b.e.-tree proofs and we denote by  
 23  $\overline{\Pi}(\mu\text{MALL}^\omega)$  its complement in  $\Pi_0(\mu\text{MALL}^\omega)$ , *i. e.* the set of all invalid b.e.-tree preproofs.

24 **Example 11.** The preproof of Example 6 is a proof:

## 2 Technical background

- 1 • the branch  $s_0(s_7)^\omega$  contains the valid thread  $(\xi)^\omega$ ;
- 2 • the branch  $s_0(s_1s_2s_3)^\omega$  contains the valid thread  $(\beta\delta\zeta)^\omega$ ;
- 3 •  $\forall k \in \mathbf{N}$ , the branch  $s_0(s_1s_2s_3)^k(s_5)^\omega$  contains the valid thread  $(\iota)^\omega$ .

### 4 Link between circular representations and infinite preproofs

5 To every  $\mu\text{MALL}^\omega$  b.e.-tree preproof  $\varpi$  is associated a canonical  $\mu\text{MALL}^\infty$  preproof  
 6  $\llbracket \varpi \rrbracket_\infty$ . The finite circular preproof  $\varpi$  may then be seen as a finite representation of the  
 7 infinite preproof  $\llbracket \varpi \rrbracket_\infty$ .

8 Without going into too much details, we give an explanation of how to define the  
 9 transformation  $\llbracket \cdot \rrbracket_\infty$ . We first define a process  $F$  of expansion of a circular preproof.  
 10 Let  $\varpi$  be a circular preproof. We construct  $F(\varpi)$  as follows. We say that a back edge  $b$   
 11 is minimal in  $\varpi$  if the position of its target is minimal, meaning that no other back edge  
 12 has a target strictly closer to the root of  $\varpi$ . For every minimal back edge  $b$  of  $\varpi$ , we  
 13 substitute, in place of the source of  $b$ , the whole subtree of  $\varpi$  rooted at the target of  $b$ ,  
 14 including its back edges. This gives us  $F(\varpi)$ . Remark that the proof tree of  $\varpi$  is a strict  
 15 prefix of the proof tree of  $F(\varpi)$  and that the greatest prefix of  $\varpi$  without any target of  
 16 back edge is, again, strictly smaller than the greatest prefix of  $F(\varpi)$  without any target  
 17 of back edge. Said differently: the lowest target of back edges in  $F(\varpi)$  are strictly higher  
 18 than in  $\varpi$ . By iterating  $F$  on  $\varpi$  we then obtain a strictly increasing sequence of circular  
 19 preproofs, in which back edges targets are higher and higher and in which the prefix of  
 20 the proof tree which is free of all back edge targets is increasing.  $\llbracket \varpi \rrbracket_\infty$  is defined as the  
 21 limit of those prefixes:  $\llbracket \varpi \rrbracket_\infty = \lim_{n \rightarrow \infty} F^n(\varpi)$ , in a sense that can be made formal.

22 **Example 12.** Let us denote  $A = \mu X \nu Y (X \otimes Y)$  and consider the following circular  
 23 preproof:

$$\varpi = \frac{\frac{\frac{\frac{\vdash A \quad \vdash \nu Y (A \otimes Y)}{\vdash A \otimes \nu Y (A \otimes Y)} (\otimes)}{\vdash \nu Y (A \otimes Y)} (\nu)}{\vdash A} (\mu)}{\vdash A}$$

24 then its first step of unfolding is

$$F(\varpi) = \frac{\frac{\frac{\frac{\frac{\vdash A \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)} \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)}$$

1 A second unfolding gives

$$F^2(\varpi) = \frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\vdash A \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)} \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)} \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)} \quad \vdash \nu Y(A \otimes Y)}{\vdash A \otimes \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}}{\vdash A}^{(\mu)}$$

2 and, at the limit, we get the infinite preproof  $\llbracket \varpi \rrbracket_\infty$  which is such that

$$\llbracket \varpi \rrbracket_\infty = \frac{\vdots \pi}{\vdash \nu Y(A \otimes Y)}^{(\mu)} \quad \text{with} \quad \pi = \frac{\frac{\vdots \llbracket \varpi \rrbracket_\infty \quad \vdots \pi}{\vdash A \quad \vdash \nu Y(A \otimes Y)}^{(\otimes)}}{\vdash \nu Y(A \otimes Y)}^{(\nu)}$$

3 The principal remark that can be made about this unfolding is that  $F(\varpi)$ , as a  $\mu\text{MALL}^\infty$   
 4 preproof, has the same branches as  $\varpi$  as a  $\mu\text{MALL}^\omega$  preproof. And each branch of  $F(\varpi)$   
 5 has the same threads as the corresponding branch of  $\varpi$ . Hence,  $F(\varpi)$  is valid as a



1 position of formula it belongs to. We denote this graph morphism by **out**. If  $t$  is a path  
 2 in  $G_{\text{thread}}^{\text{in}}$  (*i. e.* an **inner thread**), we will also denote by **out**( $t$ ) the corresponding path in  
 3  $G_{\text{thread}}^{\text{out}}$  (*i. e.* the corresponding **thread**).

**Example 14.** The images, by the morphism of Definition 38, of the inner threads of  
 Example 13 are, with the notations of Example 8:

$$\begin{aligned} \mathbf{out}((\psi)^\omega) &= (\mu)^\omega & \mathbf{out}((\psi)^\omega) &= (\iota)^\omega & \mathbf{out}(o(\pi\rho\tau)^\omega) &= \alpha(\beta\delta\zeta)^\omega \\ \forall k \in \mathbf{N}, \mathbf{out}(o(\pi\rho\tau)^k \pi\sigma\nu(\chi)^\omega) &= \alpha(\beta\delta\zeta)^k \beta\delta\eta(\kappa)^\omega \end{aligned}$$

4 **Definition 39** (Valid inner thread). An infinite inner thread  $t = (s_n)_{n \in \mathbf{N}}$  is *valid* if  
 5  $\forall n_0 \in \mathbf{N}, \exists n \geq n_0, s_n$  is principal.

6 **Example 15.** Among the inner threads of Example 13:

- 7 •  $(\psi)^\omega$  is valid:  $\psi$  is principal;
- 8 •  $(\psi)^\omega$  is valid:  $\psi$  is principal;
- 9 •  $o(\pi\rho\tau)^\omega$  is valid:  $\pi$  is principal;
- 10 •  $\forall k \in \mathbf{N}, o(\pi\rho\tau)^k \pi\sigma\nu(\chi)^\omega$  is not valid:  $\chi$  is not principal and it has no principal  
 11 vertex after the last  $\pi$ .

12 **Proposition 5** (Equivalence between the two definitions of valid infinite threads).

- 13 1. If  $t$  is a valid infinite inner thread then **out**( $t$ ) is a valid infinite thread.
- 14 2. If  $u$  is a valid infinite thread then there is a unique valid infinite inner thread  $t$   
 15 such that  $u = \mathbf{out}(t)$ .

16 *Proof.*

17  $t$  **valid**  $\Rightarrow$  **out**( $t$ ) **valid** First remark that if  $A$  is a **formula** (*i. e.* closed) then the **images**  
 18 or **preimages** of any of its positions along any **descendent relation** are still positions  
 19 of  $A$ . Now, if  $t = (s_n)_{n \in \mathbf{N}}$  is an **inner thread**, let us denote by **fml**( $t$ ) = **fml**( $s_0$ ).  
 20 Then  $\forall n \in \mathbf{N}, \mathbf{fml}(s_n) = \mathbf{fml}(t)$ . Because  $t$  is valid, this **fml**( $t$ ) is a  $\nu$ -formula.  
 21 Each sequent position at which  $t$  is principal is also a position at which **fml**( $t$ ) is  
 22 principal in **out**( $t$ ), **fml**( $t$ ), hence **fml**( $t$ ) is infinitely principal in **out**( $t$ ). Finally,  
 23 **fml**( $t$ ) is the smallest formula of **out**( $t$ ) because it is a subformula of every formula  
 24 of **out**( $t$ ). Conclusion: **out**( $t$ ) is a valid infinite thread.

## 2 Technical background

1 **Converse** if  $t \in G_{\text{thread}}^{\text{out}}$  is valid, let  $\nu XA$  be its validating formula, let  $t = t_0 t_1 t_2 \dots$ , let  $i_n$   
 2 be the sequence of all  $i$  such that  $t_i$  is a principal position of  $\nu XA$ . If  $i_n < i < i_{n+1}$ ,  
 3 we define  $u_i$  as the image of  $t_{i_{n+1}}$  in  $t_i$ . Then  $u = (u_i)_i$  is a valid inner thread such  
 4 that  $\mathbf{out}(u) = t$ . It is easy to see that it is the only one, thanks to the following  
 5 lemma.

6 □

7 **Lemma 1.** *The positions of  $\sigma XA[X]$  in  $A[\sigma XA[X]]$  are exactly the positions of  $X$  in*  
 8  *$A[X]$ .*

9 *Proof.* Otherwise,  $\sigma XA[X]$  would be a subformula of  $A[X]$ , which is impossible. □

10 **Example 16.** The preproof of Example 6 is a proof:

- 11 • the branch  $s_0(s_7)^\omega$  contains the valid inner thread  $(\psi)^\omega$ ;
- 12 • the branch  $s_0(s_1 s_2 s_3)^\omega$  contains the valid inner thread  $(\pi \rho \tau)^\omega$ ;
- 13 •  $\forall k \in \mathbb{N}$ , the branch  $s_0(s_1 s_2 s_3)^k (s_5)^\omega$  contains the valid inner thread  $(\psi)^\omega$ .

14 Much in the spirit of these **inner threads**, we can generalize Definitions 10 and 11 to the  
 15 case of b.e.-tree-preproofs.

16 **Definition 40** (Generalization of  $\mathfrak{t}$  and  $\mathfrak{T}$ ). The definitions of  $\mathfrak{t}$  (Definition 10) and  
 17  $\mathfrak{T}$  (Definition 11) are extended in the obvious way to the setting of  $\mu\text{MALL}^\omega$  b.e.-tree  
 18 preproof: if  $(\pi, \mathbf{back})$  is a  $\mu\text{MALL}^\omega$  b.e.-tree preproof, if  $u = (s_0, \dots, s_n)$  is a path in its  
 19  $G_{\text{branch}}$  and  $p$  is a position of subformula in the sequent at position  $s_n$ , then:

- 20 •  $\mathfrak{T}(u)(p)$  is the unique path  $p_{i_0}, \dots, p_n$  such that
  - 21 –  $\forall i, p_i$  is a position of subformula in the sequent at position  $s_i$ ,
  - 22 –  $p_n = p$ ,
  - 23 –  $\forall i, i_0 \leq i < n \Rightarrow$  the formula containing  $p_i$  in the sequent at position  $s_i$  is  
 24 the immediate descendant of the formula containing  $p_{i+1}$  in the sequent at  
 25 position  $s_{i+1}$ , and  $p_i$  is the image of  $p_{i+1}$  in that formula,
  - 26 – either  $i_0 = 0$  or  $p_{i_0}$  is in a cut formula,

$$\bullet \mathfrak{t}(u)(p) = \begin{cases} p_0 & \text{if } i_0 = 0 \\ \text{undefined} & \text{if } i_0 > 0 \end{cases}$$

In this last case,  $p_{i_0}$  is in a cut formula.

*Remark 7.* If, in the previous definition, the subformula at position  $p$  is a  $\nu$ -formula, then  $\mathfrak{T}(u)(p)$  is a path in  $G_{\text{thread}}^{\text{in}}$ .

## 2.2 Complexity theory

### 2.2.1 Generalities on complexity and PSPACE

We give here a brief and somewhat informal recall of the essential notions of algorithmic complexity that we will need in the thesis. For an in-depth perspective on that subject, see, for instance, [Arora and Barak, 2009]. If you understand French, you may also read the excellent book by Perifel [2014].

In particular we do not go here into the choice and definition of a precise model of computation. All programs are considered deterministic. We assume that the inputs and outputs of programs contain at least  $\{0, 1\}^*$ , the finite words on  $\{0, 1\}$ , and that there is a pairing operation which allows to combine two entries  $x$  and  $y$  into a single one  $(x, y)$ .

**Definition 41** (Notations). If  $f$  is a program and  $x$  an entry for this program, we denote by  $\|x\|$  the size of the entry  $x$ , we denote by  $\mathbf{time}(f, x)$  the execution time of  $f$  on  $x$ , which may be an integer or  $+\infty$ , and we denote by  $\mathbf{space}(f, x)$  the memory space used by the execution of  $f$  on  $x$ , which may be an integer or  $+\infty$ .

**Definition 42** (Problem, language). A *problem* or *language* is any subset of all possible program entries.

**Definition 43.**

• if  $t: \mathbb{N} \rightarrow \mathbb{N}$  is a function, a language  $L$  is in  $\text{DTIME}(t(n))$  if there is a program  $f$  such that

$$- \forall x \in L, f(x) = 1$$

## 2 Technical background

- 1           –  $\forall x \notin L, f(x) = 0$
- 2           –  $\exists \alpha > 0, \forall x, \mathbf{time}(f, x) \leq \alpha \cdot t(\|x\|)$
- 3       • if  $t: \mathbf{N} \rightarrow \mathbf{N}$  is a function, a language  $L$  is in  $\text{DSPACE}(t(n))$  if there is a program
- 4        $f$  such that
- 5           –  $\forall x \in L, f(x) = 1$
- 6           –  $\forall x \notin L, f(x) = 0$
- 7           –  $\exists \alpha > 0, \forall x, \mathbf{space}(f, x) \leq \alpha \cdot t(\|x\|)$

8 **Definition 44.**

- 9       •  $\mathbf{P} := \bigcup_{k \in \mathbf{N}} \text{DTIME}(n^k)$
- 10      •  $\mathbf{EXP} := \bigcup_{k \in \mathbf{N}} \text{DTIME}(2^{n^k})$
- 11      •  $\mathbf{PSPACE} := \bigcup_{k \in \mathbf{N}} \text{DSPACE}(n^k)$
- 12      •  $\mathbf{NP} := \{A \mid \exists p \text{ polynomial}, \exists B \in \mathbf{P}, \forall x, (x \in A \Leftrightarrow \exists y \in \{0, 1\}^{p(\|x\|)}, (x, y) \in B)\}$

**Proposition 6.**

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP}$$

13 The strictness of each one of these inclusions is an open problem. Experts of the domain  
 14 seem to believe that all these inclusions are in fact strict. We still know for sure that not  
 15 all of them can be equalities, as a consequence of Corollary 1 to Theorem 2 below.

16 **Definition 45.** A function  $t: \mathbf{N} \rightarrow \mathbf{N}$  is *time-constructible* if there is a program  $f$  such  
 17 that

- 18      •  $\forall n \in \mathbf{N}, f(1^n) = 1^{t(n)}$  (where  $1^n$  is the word of length  $n$  made only of 1)
- 19      •  $\exists \alpha > 0, \forall n \in \mathbf{N}, \mathbf{time}(f, 1^n) \leq \alpha \cdot t(n)$

20 **Theorem 2** (Deterministic time hierarchy). *If  $f, g: \mathbf{N} \rightarrow \mathbf{N}$  are two functions such that*

- 1     •  $\forall n \in \mathbf{N}, f(n) > 0$
- 2     •  $g$  is *time-constructible*
- 3     •  $g(n) \gg f(n) \cdot \log(f(n))$
- 4 then  $\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$ .

**Corollary 1.**

$$\text{P} \subsetneq \text{EXP}$$

5 **Definition 46** (Polynomial-time many-one reduction). A *polynomial-time many-one*  
6 *reduction* from a language  $B$  to a language  $A$  is a program  $f$  which runs in polynomial  
7 time and such that  $\forall x, x \in B \Leftrightarrow f(x) \in A$ . If such a program exist, we say that  $B$  *reduces*  
8 *to*  $A$  and we denote it by  $B \leq A$ .

**Definition 47.** A problem  $A$  is *PSPACE-complete* if  $A$  is in PSPACE and every PSPACE  
problem reduces to  $A$ . Equivalently,  $A$  is PSPACE-complete iff.

$$\forall B, B \in \text{PSPACE} \Leftrightarrow B \leq A$$

9 **Proposition 7.**

- 10     • A problem  $A$  is in PSPACE iff. its complement  $\bar{A}$  is in PSPACE.
- 11     • A problem  $A$  is PSPACE-complete iff. its complement  $\bar{A}$  is PSPACE-complete.

12 *Proof.*

- 13     • Every  $\text{DSPACE}(n^k)$  is closed under complement: if  $f$  recognizes  $A$ ,  $\bar{A}$  is recognized  
14     by  $g(x) = 1 - f(x)$ . Hence PSPACE is closed under complement.
- 15     • Suppose  $A$  is PSPACE-complete and  $B$  is PSPACE. We then show that  $B$  reduces to  
16      $\bar{A}$ . As  $B$  is PSPACE, so is  $\bar{B}$ . Hence there is a polynomial-time many-one reduction  
17      $f$  from  $\bar{B}$  to  $A$ . Then we show that  $f$  is also a reduction from  $B$  to  $\bar{A}$ . Indeed,  
18      $\forall x, x \notin B \Leftrightarrow x \in \bar{B} \Leftrightarrow f(x) \in A$ , and then  $\forall x, x \in B \Leftrightarrow f(x) \notin A \Leftrightarrow f(x) \in \bar{A}$ .

19 □

20 *Remark 8.* Although this will not be used in this thesis, we recall to the interested reader  
21 that we also have  $\text{PSPACE} = \text{NPSPACE} = \text{coNPSPACE}$  because of Savitch theorem.

## 1 2.2.2 Some PSPACE-complete problems

**Definition 48** ( $\text{BOOLE}_{false}$  and  $\mathcal{B}_{false}$ ). A program in  $\text{BOOLE}_{false}$  is a sequence of instructions  $b = 1:I_1 \ 2:I_2 \ \dots \ m:I_m$  where an instruction can have one of the two following forms:

$$I ::= X := \neg X \mid \text{if } X \text{ then goto } \ell' \text{ else goto } \ell''$$

2 where  $X$  ranges over a finite set of variable names. The semantic is as expected, with all  
 3 variables being equal to **false** at the beginning of the program, and the program halting  
 4 when reaching instruction  $m + 1$ . Finally it is required of every program  $b \in \text{BOOLE}_{false}$   
 5 that if  $b$  terminates then all variables have value *false* at the end of its execution.

We also denote the set of terminating programs as:

$$\mathcal{B}_{false} := \{b \in \text{BOOLE}_{false} \mid b \text{ terminates}\}$$

6 *Remark 9.* The constraint on the values of the variables at the end of the program  
 7 will be useful when reducing it, in Chapter 3, to the problem of validity of a  $\mu\text{MALL}^\omega$   
 8 b.e.-tree preproof. This preproof will encode the fact that the program  $b$  is terminating  
 9 by connecting the final state to the initial one, hence the necessity that its initial and  
 10 terminal states are the same.

11 **Lemma 2.**  $\mathcal{B}_{false}$  is PSPACE-hard under LOGSPACE-reductions:

$$12 \quad \text{PSPACE} \leq_L \mathcal{B}_{false}$$

13 *Proof.* We reduce from the problem of termination for a more expressive language, which  
 14 has been defined and proved PSPACE-complete by Jones in [Jones, 1997], under the  
 15 name of  $\text{BOOLE}$ . □

## 16 2.3 Parity automata

17 We recall here the basic definitions and property of parity automata. There are several  
 18 definitions of parity automata, all of which are equivalent. We chose a definition that  
 19 best suits our needs. For more informations on the subject of automata for infinite words,  
 20 see, for instance, [Grädel, Thomas, and Wilke, 2002] or [Perrin and Pin, 2004].

21 We denote by  $\omega$  the set of natural integers, and we also denote by  $\omega$  the first infinite  
 22 ordinal. We denote by  $\omega + 1$  the set  $\omega \cup \{\omega\}$ , ordered by setting  $\omega$  to be its maximum.

1 The ordinal  $\omega$  is considered even. All of that is consistent with usual set-theoretic  
2 conventions.

3 Given an alphabet  $A$ , we denote by  $A^\omega$  the set of infinite words on  $A$ , that is the set of  
4 sequences of elements of  $A$ , indexed by the integers.

5 **Definition 49** (Parity automaton). Given an alphabet  $A$ , a *parity automaton* on  $A$   
6 consists of

- 7 • a set  $Q$  of *states*;
- 8 • a subset  $I \subseteq Q$  of *initial states*;
- 9 • a set  $T \subseteq Q \times A \times (\omega + 1) \times Q$ .

10 Such a parity automaton is *finite* if  $A$ ,  $Q$  and  $T$  are finite. In  $T$ , elements of  $\omega + 1$  are  
11 called *priorities*.

**Definition 50.** A *run*, or *path*, in a parity automaton  $\mathcal{A}$  is any infinite sequence

$$q_0 \xrightarrow[p_0]{a_0} q_1 \xrightarrow[p_1]{a_1} q_2 \xrightarrow[p_2]{a_2} q_3 \dots$$

12 such that

- 13 •  $q_0 \in I$
- 14 •  $\forall n \in \omega, (q_n, a_n, p_n, q_{n+1}) \in T$

15 If  $r$  is such a run, we denote by **label**( $r$ ) the infinite word  $a_0 a_1 a_2 \dots \in A^\omega$ .

16 **Definition 51.** Such a run  $r$  is *successful* in the parity automaton if there is a  $p \in \omega + 1$   
17 such that

- 18 •  $p$  is odd (which implies  $p \neq \omega$ )
- 19 • there are only finitely many  $n \in \omega$  such that  $p_n < p$
- 20 • there are infinitely many  $n \in \omega$  such that  $p_n = p$

21 Equivalently :  $p$  is an odd priority and it is the smallest priority appearing infinitely often  
22 in the run  $r$ .

## 2 Technical background

1 **Definition 52** (Language of a parity automaton). The *language* of a parity automaton  
2  $\mathcal{A}$  is the set  $\mathcal{L}(\mathcal{A}) := \{\text{label}(p) \mid p \text{ is a successful run in } \mathcal{A}\}$ .

3 And the main theorem that we will use is the following.

4 **Theorem 3.** *The problem of equality of languages for finite parity automata is decidable*  
5 *in polynomial space.*

6 *Proof.* We do not go into the details of this complex result. It relies on the fact that  
7 every parity automaton can be translated into an equivalent Büchi automaton with a  
8 polynomial increase of size, which is folklore, and on the fact that the problem of  
9 equality of languages for Büchi automata is decidable in polynomial space, which was  
10 proven by Sistla, Vardi, and Wolper [1987]. □

# 3 PSPACE-completeness of the thread criterion

As we explained in the introduction, the subject of this thesis is the study of finite circular representations of infinite proofs and preproofs. In this context, the first question arising is to distinguish, among circular representations of preproofs, those denoting valid preproofs, *i. e.* proofs, from those denoting invalid preproofs.

It has already been shown by Doumane [2017b] that this problem is decidable in PSPACE. In the first section of this chapter, we recall a proof of this result.

The second section of this chapter is devoted to its main result: that the problem of deciding the validity of a circular representation of a  $\mu\text{MALL}^\omega$  preproof is in fact PSPACE-complete. This result is established by a reduction from the problem of deciding the termination of a boolean program.

Our proof takes a lot of inspiration from the proof of PSPACE-completeness of size-change termination by Lee, Jones and Ben Amram [Lee, Jones, and Ben-Amram, 2001]: in order to prove that deciding size-change termination is PSPACE-complete, they define a notion of boolean program and use the fact that the following set is complete in PSPACE:

$$\mathcal{B} = \{b \mid b \text{ is a boolean program and } b \text{ terminates.}\}$$

then they reduce  $\mathcal{B}$  to the problem of size-change termination. We adapt their method by reducing  $\mathcal{B}$  to the problem of thread-validity in circular  $\mu\text{MALL}^\omega$  preproof.

It would be very interesting to get a more precise understanding of the relation between threads in circular proofs and size-change termination.

We show in Section 3.2.4 how our method adapts to other systems such as  $\mu\text{LJ}$  and  $\mu\text{LK}$ .

### 1 3.1 Deciding thread validity in PSPACE

2 In this section, we recall that the problem of deciding whether the **circular representation**  
 3 of a **preproof** is a valid **proof** is in PSPACE. Several algorithms can be used for deciding  
 4 this problem in PSPACE. Here we reduce this problem to the problem of deciding equality  
 5 of languages for **parity  $\omega$ -automata**, which is known to be in PSPACE. See Section 2.3 of  
 6 the technical background for more details on that. More precisely, given a **preproof**  $\varpi$ ,  
 7 we define two **parity automata**: the language of the first one is the set of **infinite branches**  
 8 of  $\varpi$  and the language of the second one is the set of **valid infinite branches** of  $\varpi$ .

9 We will immediately give two examples to show how the algorithm works.

10 Let  $G := \nu Y \mu X (X \wp Y)$ . The following preproof (3.1) is valid. The branches that  
 11 ultimately always take the left back edge are valid because of a thread going through  
 12  $\nu X X$  and the branches that take infinitely often the right back edge are valid because of  
 13 a thread going through  $\mu X (X \wp G)$  and  $G$ .

$$\begin{array}{c}
 \boxed{
 \begin{array}{c}
 \frac{\frac{\vdash \nu X X, \mu X (X \wp G), \mu X X}{\vdash \nu X X, \mu X (X \wp G), \mu X X} (\nu) \quad \frac{\vdash \nu X X, \mu X (X \wp G), \mu X X}{\vdash \nu X X, G, \mu X X} (\nu)}{\frac{\vdash \nu X X, \mu X (X \wp G), G, \mu X X}{\vdash \nu X X, \mu X (X \wp G) \wp G, \mu X X} (\wp)} (\text{cut}) \\
 \frac{\vdash \nu X X, \mu X (X \wp G) \wp G, \mu X X}{\rightarrow \vdash \nu X X, \mu X (X \wp G), \mu X X \leftarrow} (\mu)
 \end{array}
 } \\
 14
 \end{array} \tag{3.1}$$

15 From this preproof, we build a first automaton, represented on Figure 3.1 p. 74, which  
 16 recognizes the language of infinite branches of the preproof. It has one initial state and  
 17 all paths are accepted. It recognizes the language  $(abcd + abef)^\omega$ .

And from the same preproof (3.1) we build a second automaton, represented on Figure 3.2,  
 p. 75, which recognizes the language of valid infinite branches of the preproof (3.1). No  
 priority indicated on an edge implicitly means  $\omega$ , which is the highest non-accepting  
 priority. There are four initial states, at the bottom of the automaton. Three edges have  
 a priority  $< \omega$

$$\nu X X \xrightarrow[1]{d} \nu X X \quad G \xrightarrow[1]{f} \mu X (X \wp G) \quad \mu X (X \wp G) \xrightarrow[2]{a} \mu X (X \wp G) \wp G$$

It recognizes the language

$$\begin{aligned} & (abcd + abef)^*(abcd)^\omega + ((abcd)^*abef)^\omega \\ & = (abcd + abef)^\omega \end{aligned}$$

1 Now let  $F := \mu X \nu Y (X \wp Y)$ . The following preproof (3.2), unlike the preproof (3.1), is  
 2 not valid. The branches taking ultimately only the left back edge are valid because of  
 3 a thread going through  $\nu XX$ , the branches taking ultimately only the right back edge  
 4 are valid because of a thread going through  $\nu Y (F \wp Y)$  but the branches which takes  
 5 infinitely often the left back edge and infinitely often the right back edge are not valid.

$$\begin{array}{c} \boxed{\begin{array}{c} \frac{\vdash \nu XX, \nu Y (F \wp Y), \mu XX}{\vdash \nu XX, F, \mu XX} \text{ } (\mu) \\ \frac{\vdash \nu XX, F, \mu XX}{\vdash \nu XX, F, \mu XX} \text{ } (\nu) \quad \vdash \nu XX, \nu Y (F \wp Y), \mu XX \\ \hline \frac{\vdash \nu XX, F, \nu Y (F \wp Y), \mu XX}{\vdash \nu XX, F \wp \nu Y (F \wp Y), \mu XX} \text{ } (\wp) \\ \frac{\vdash \nu XX, F \wp \nu Y (F \wp Y), \mu XX}{\vdash \nu XX, \nu Y (F \wp Y), \mu XX} \text{ } (\nu) \end{array}} \quad \text{(cut)} \end{array} \quad (3.2)$$

7 The deduced automaton recognizing all infinite branches of preproof (3.2), is represented  
 8 on Figure 3.3, p. 76. It has one initial state. All paths are accepted. It recognizes the  
 9 language  $(abcde + abf)^\omega$ .

And the automaton of valid infinite branches of preproof (3.2) is represented on Figure 3.4,  
 p. 77. It has 4 initial states. The only three transitions with priorities  $< \omega$  are

$$\nu Y (F \wp Y) \xrightarrow{\frac{a}{3}} F \wp \nu Y (F \wp Y) \quad \nu XX \xrightarrow{\frac{d}{1}} \nu XX \quad F \xrightarrow{\frac{e}{2}} \nu Y (F \wp Y)$$

It recognizes the language

$$\begin{aligned} & (abcde + abf)^*(abcde)^\omega + (abcde + abf)^*(abf)^\omega \\ & = (abcde + abf)^\omega \setminus ((abf)^*abcde(abcde)^*abf)^\omega \end{aligned}$$

10 We will now explain the procedure by which, in each case, those two automata were  
 11 obtained.

12 Let  $\varpi$  be a b.e.-tree preproof. Let  $G_{\text{branch}}$  be its branch graph, as defined in Definition 28.

13 Let  $A := E_{\text{branch}}$  be the set of its edges. This will be the input alphabet of our automata.

14 The first  $\omega$ -automaton is  $\mathcal{A}_{\text{branch}} := (Q_{\text{branch}}, I_{\text{branch}}, T_{\text{branch}})$ , where:

### 3 PSPACE-completeness of the thread criterion

- 1 • The set of states is  $Q_{\text{branch}} := V_{\text{branch}}$ , the set of vertices of  $G_{\text{branch}}$ , that is the set  
2 of closed positions of sequents in  $\pi$ .
- 3 • The set of initial states is the singleton  $I_{\text{branch}} := \{i_{\text{branch}}\}$ , where  $i_{\text{branch}}$  is the  
4 position of the root of  $\varpi$ , that is the position of the conclusion sequent of  $\varpi$ .
- 5 • The set of transitions is

$$T_{\text{branch}} := \{ s \xrightarrow[1]{e} s' \mid e \text{ is an edge from } s \text{ to } s' \text{ in } G_{\text{branch}} \}$$

6 This amounts to saying that the accepted paths in  $\mathcal{A}_{\text{branch}}$  are exactly the paths in  $G_{\text{branch}}$   
7 which start at the root of the proof tree. With that definition, the following lemma is  
8 immediate:

9 **Lemma 3.** *The language  $\mathcal{L}(\mathcal{A}_{\text{branch}})$  is the set of infinite branches of  $\varpi$ .*

10 For our second automaton, we need a priority assignment  $\Omega: \mathbf{Fml} \rightarrow \omega$ . It will be used  
11 to reflect the condition of validity of threads in the parity condition of our automata. We  
12 need this priority assignment to have two main properties:

- 13 1. if  $A$  is a subformula of  $B$  then  $\Omega(A) \leq \Omega(B)$ ;
- 14 2.  $\forall A$ ,  $\Omega(\mu X A)$  is even and  $\Omega(\nu X A)$  is odd.

15 Remember that even priorities are rejecting and odd priorities are accepting (Definition 51).  
16 Thus, those two properties reflect the condition of Definition 34.

**Definition 53** ( $\Omega: \mathbf{Fml} \rightarrow \omega$ ). A function  $\Omega$  is defined by induction, which associate a  
priority to every *preformula*:

$$\Omega(\nu X A) := \begin{cases} \Omega(A) & \text{if } \Omega(A) \text{ is odd} \\ \Omega(A) + 1 & \text{if } \Omega(A) \text{ is even} \end{cases}$$

$$\Omega(\mu X A) := \begin{cases} \Omega(A) & \text{if } \Omega(A) \text{ is even} \\ \Omega(A) + 1 & \text{if } \Omega(A) \text{ is odd} \end{cases}$$

$$\Omega(A \odot B) := \max\{\Omega(A), \Omega(B)\}$$

for any binary connective  $\odot$

$$\Omega(\mathbf{c}) := 0$$

for any propositional constant  $\mathbf{c}$

$$\Omega(X) := 0$$

17 The following remarks are immediate:

1 **Lemma 4.** For all formulas  $A$  and  $B$ :

- 2 •  $\Omega(\nu XA)$  is odd and  $\Omega(\mu XA)$  is even.
- 3 • If  $A$  is a subformula of  $B$  then  $\Omega(A) \leq \Omega(B)$ .

4 Our second automaton is a **parity  $\omega$ -automaton**, with priorities in  $\omega + 1 = \omega \cup \{\omega\}$ ,  
 5  $\omega$  being even.<sup>1</sup> This second automaton is defined as  $\mathcal{A}_{\text{thread}} = (Q_{\text{thread}}, I_{\text{thread}}, T_{\text{thread}})$ ,  
 6 where:

- 7 • the set of states is  $Q_{\text{thread}} := V_{\text{thread}} + \{\perp_s \mid s \in V_{\text{branch}}\}$ , *i. e.* the vertices of  $G_{\text{thread}}^{\text{out}}$   
 8 plus one extra vertex for each vertex of  $G_{\text{branch}}$
- 9 • If we keep the notation  $i_{\text{branch}}$  to denote the position of the endsequent of  $\varpi$ , the  
 10 set of initial states  $I_{\text{thread}}$  is the set of all positions of formulas in  $i_{\text{branch}}$ , plus the  
 11 extra vertex  $\perp_{i_{\text{branch}}}$ .
- 12 • the set of transitions is

$$\begin{aligned}
 T_{\text{thread}} := & \{ \perp_s \xrightarrow{\omega} \perp_{s'} \mid e \text{ is an edge from } s \text{ to } s' \text{ in } G_{\text{branch}} \} \\
 & \cup \{ \alpha \xrightarrow[\Omega(\alpha)]{\mathbf{U}(e)} \beta \mid e \text{ is an edge from } \alpha \text{ to } \beta \text{ in } G_{\text{thread}}^{\text{out}} \text{ and } \alpha \text{ is principal} \} \\
 & \cup \{ \alpha \xrightarrow[\omega]{\mathbf{U}(e)} \beta \\
 & \quad \mid e \text{ is an edge from } \alpha \text{ to } \beta \text{ in } G_{\text{thread}}^{\text{out}} \text{ and } \alpha \text{ is not principal} \} \\
 & \cup \{ \perp_s \xrightarrow[\omega]{e} \alpha \\
 & \quad \mid I \text{ is a (cut) inference in } \varpi, \\
 & \quad \quad s \text{ is the position of the conclusion sequent of } I, \\
 & \quad \quad s' \text{ is the position of a premise of } I, \\
 & \quad \quad e \text{ is an edge induced by } I \text{ in } G_{\text{branch}} \text{ from } s \text{ to } s' \\
 & \quad \quad \text{and } \alpha \text{ is the position of the corresponding cut-formula in } s' \}
 \end{aligned}$$

13 where  $q \xrightarrow[i]{e} q'$  denote a transition from state  $q \in Q_{\text{thread}}$  to state  $q' \in Q_{\text{thread}}$  with  
 14 label  $e \in A$  and priority  $i \in \omega + 1$ .

15 The acceptance condition is the one given in our definition of **parity automata** in  
 16 Definition 49: a run is accepted if the smallest priority appearing infinitely often is *odd*  
 17 ( $\omega$  being even).

18 Once again, it should be clear from Definitions 21 and 22 that:

---

<sup>1</sup>We use the usual set-theoretic convention that an ordinal is equal to the set of ordinals strictly below it:  $\alpha = \{\beta \in \text{Ord} \mid \beta < \alpha\}$

1 **Lemma 5.** *The language  $\mathcal{L}(\mathcal{A}_{\text{thread}})$  is the set of valid infinite branches of  $\varpi$ .*

2 From these two lemmas it is immediate that

3 **Proposition 8.** *We have the inclusion  $\mathcal{L}(\mathcal{A}_{\text{thread}}) \subseteq \mathcal{L}(\mathcal{A}_{\text{branch}})$  and the preproof  $\varpi$  is*  
4 *valid iff. this inclusion is an equality.*

5 Deciding this equality can be done in PSPACE (c.f. Theorem 3), and the constructions  
6 of these automata are obviously PSPACE, so:

7 **Proposition 9.** *The problem of deciding whether a  $\mu\text{MALL}^\omega$  b.e.-tree preproof is valid*  
8 *is in PSPACE.*

## 9 3.2 PSPACE-completeness

### 10 3.2.1 Outline of the PSPACE-completeness proof

11 We now aim at proving that the problem of deciding the validity of a  $\mu\text{MALL}^\omega$  b.e.-tree  
12 preproof is PSPACE-complete. As it is already known that this problem is in PSPACE,  
13 it remains to prove that it is PSPACE-hard.

We follow the same methodology as Lee, Jones, and Ben-Amram [2001]: in order to  
prove that deciding size-change termination is PSPACE-complete, they define a notion  
of boolean program (see Definition 48) and use the fact that the following problem is  
PSPACE-complete:

$$\mathcal{B} = \{b \mid b \text{ is a boolean program and } b \text{ terminates.}\}$$

14 then they reduce  $\mathcal{B}$  to the decidability of size-change termination.

15 We try to adapt their method by reducing  $\mathcal{B}$  to  $\Pi(\mu\text{MALL}^\omega)$ .

### 16 3.2.2 Definition of the reduction

17 We use the boolean programs that were defined in Definition 48. The following definition  
18 will be used in the proof of Proposition 10:

1 **Definition 54** (Call graph of a program). Assume a boolean program  $b$  with variables  
 2  $X_1, \dots, X_k$  and instructions  $1 : I_1, \dots, m : I_m$ . Define the call graph of  $b$  to be  $G = (V, E)$   
 3 with

- 4 •  $V = \{0, 1, \dots, m\}$
- 5 •  $E = \{0 \xrightarrow{0} 1\}$   
 $\cup \{\ell \xrightarrow{\ell} ((\ell + 1) \bmod (m + 1)) \mid I_\ell = \text{"X := not X"}\}$   
 $\cup \{\ell \xrightarrow{\ell^+} \ell', \ell \xrightarrow{\ell^-} \ell'' \mid I_\ell = \text{"if X goto } \ell' \text{ else } \ell''"\}$

6 The following definition is the main object of this chapter. It describes a transformation  
 7 that turns every boolean program  $b$  into a  $\mu\text{MALL}^\omega$  b.e.-tree-preproof, with the property  
 8 that this preproof is thread-valid iff. the source program is non-terminating.

**Definition 55** ( $\llbracket \cdot \rrbracket : \text{BOOLE}_{\text{false}} \rightarrow \Pi_0(\mu\text{MALL}^\omega)$ ). For every boolean program  $b \in \text{BOOLE}_{\text{false}}$ , we define a preproof  $\llbracket b \rrbracket \in \Pi_0(\mu\text{MALL}^\omega)$ . Let  $X_1, \dots, X_k$  be the variables of  $b$  and  $1 : I_1, \dots, m : I_m$  its instructions. We first give names to the formulas that will appear in  $\llbracket b \rrbracket$ : we define a unary operation  $\iota$ , three formulas  $A, B, C$ , a family of unary operations  $(\iota_n)$  and two families of formulas  $(D_n), (E_n)$ :

$$A = \iota(\nu X \iota X) \quad B = \nu X(\perp \oplus X) \quad C = \mu X(B \wp X) \quad E_n = \iota_n(\nu X \iota_n X)$$

$$\iota F = \mu X(F \oplus (\perp \oplus (X \wp X))) \quad \iota_n F = \mu X(\perp \oplus (X \wp \underbrace{(F \wp \dots \wp F)}_{n-1}))$$

$$D_n = \mu X(\underbrace{X \& \dots \& X}_n)$$

9 We now define  $\llbracket b \rrbracket$  to be the preproof

$$10 \quad \frac{\begin{array}{c} \vdots \\ \llbracket 0 : \rrbracket \\ \vdots \end{array} \quad \begin{array}{c} \vdots \\ \llbracket 1 : I_1 \rrbracket \\ \vdots \end{array} \quad \dots \quad \begin{array}{c} \vdots \\ \llbracket m : I_m \rrbracket \\ \vdots \end{array}}{\text{(ROOT)} \vdash A^{2k}, B, C, D_2, D_m, E_m^m} \quad \frac{\vdash A^{2k}, B, C, D_2, D_m, E_m^m}{\vdash A^{2k}, B, C, D_2, D_m, E_m^m} \quad (\mu, (\&))^{m-1} \quad (3.3)$$

11 where  $A^{2k}$  is an abbreviation for  $\underbrace{A, \dots, A}_{2k}$  and  $E_m^m$  is an abbreviation for  $\underbrace{E_m, \dots, E_m}_m$   
 12 and the subtrees  $\llbracket 0 : \rrbracket, \llbracket 1 : I_1 \rrbracket, \dots, \llbracket m : I_m \rrbracket$  will be defined below.

13 *Remark 10* (Short notation for long sequents). You just saw on Eq. (3.3) that, for any  
 14 formula  $A$ , we use  $A^n$  as an abbreviation for  $\underbrace{A, \dots, A}_n$ . Similarly, for any formulas  $A$  and  
 15  $B$ , we may use  $(A, B)^n$  as an abbreviation for  $\underbrace{A, B, \dots, A, B}_{2n \text{ formulas}}$ . We do so, for instance, on  
 16 Figure 3.6.

### 3 PSPACE-completeness of the thread criterion

1 The root of the preproof  $\llbracket b \rrbracket$  is constructed by translating each pair  $\ell : \mathbf{I}_\ell$  of a label and an  
 2 instruction into a finite segment of branch of preproof, as defined in Eq. (3.3), with each  
 3 subderivation  $\llbracket \ell : \mathbf{I}_\ell \rrbracket$  defined in Fig. 3.6 and each subderivation  $\llbracket \ell : \text{goto } \ell' \rrbracket$  in Fig. 3.5.

4 *Remark 11* (Implicit threading functions). Notice that in our description of the translation  
 5 in the previous definition, the **threading functions** were left implicit. The only exception  
 6 is in the definition of  $\llbracket \ell : \mathbf{X}_i := \text{not } \mathbf{X}_i \rrbracket$ , on Figure 3.6, in which we use arrows to express  
 7 the exchange of the positions of two occurrences of  $A$ .

8 *Remark 12* (Infinite branches of  $\llbracket b \rrbracket \simeq E^\omega$ ). The preproof  $\llbracket b \rrbracket$  constructed from  $b$  by the  
 9 reduction  $\llbracket \cdot \rrbracket$  of Definition 55 is a finite tree with back edges in which every finite branch  
 10 ends with a back edge to the root. This finite tree has exactly as many branches and  
 11 consequently as many back edges to the root, as the number  $\mathbf{Card} E$  of edges in the  
 12 call-graph of  $b$  (Definition 54). This in turn entails that the set of infinite branches of  
 13 the preproof  $\llbracket b \rrbracket$  is in one-to-one correspondence with the set  $E^\omega$  of infinite words on  $E$ .  
 14 Note however that an infinite word  $\bar{u} \in E^\omega$  has no reason *a priori* to be a path in  $G$ .

15 From now on, **we will refer directly to infinite branches of the preproof by**  
 16 **words  $\bar{u} \in E^\omega$ .**

#### 17 3.2.3 Main theorem

18 We now prove that  $\Pi(\mu\text{MALL}^\omega)$  is PSPACE-complete.

*Remark 13* (Thread groups). We need to be more precise about the occurrences of  
 formulas in the conclusion sequent of preproof  $\llbracket b \rrbracket$ :

$$\underbrace{A, \dots, A}_{2k}, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_m$$

Let us label the occurrences of  $A$  in this sequent as follows:

$$A_1^+, A_1^-, \dots, A_k^+, A_k^-, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_m$$

so that we can talk precisely about them. It can be seen by examining the definition of  
 $\llbracket \cdot \rrbracket$  (def. 55) that a valid thread in the preproof cannot pass through  $D_2$  or  $D_m$ , which  
 contain no  $\nu$ , and that the remaining formulas are divided into  $k + 2$  groups

$$\underbrace{A_1^+, A_1^-}, \dots, \underbrace{A_k^+, A_k^-}, \underbrace{B, C}, \underbrace{E_m, \dots, E_m}$$

19 which cannot thread-interact with each other, in the sense that, for instance, no thread  
 20 can contain a  $B$  and a  $E_m$ , or a  $A_\ell^\epsilon$  and a  $A_{\ell'}^{\epsilon'}$  if  $\ell \neq \ell'$ .

1 **Lemma 6.** *An infinite branch  $\bar{u} \in E^\omega$  in the preproof contains a validating thread*

- 2 • *in the  $E_m$  group iff. no suffix of  $\bar{u}$  is a valid path in  $G$ .*
- 3 • *in the  $B, C$  group iff. 0 occurs only finitely in  $\bar{u}$ .*

4 *Proof.* The proof goes by case on the instructions involved.

5 In order to prove the first part of the statement, that is that an infinite branch  $\bar{u} \in E^\omega$  in  
 6 the preproof contains a validating thread in the  $E_m$  group iff. no suffix of  $\bar{u}$  is a valid path  
 7 in  $G$ , we reason by case on the instructions involved and remark that the  $E_m$  formulas  
 8 are touched only in the  $\llbracket \ell : \text{goto } \ell' \rrbracket$  parts of the preproof.

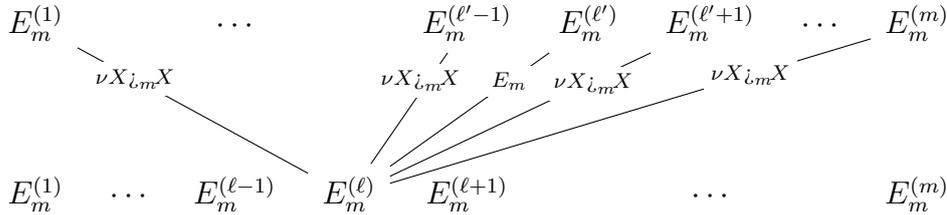
The  $E_m$  formulas are touched only in the  $\llbracket \ell : \text{goto } \ell' \rrbracket$  parts of the preproof. Observe that  
 in  $\llbracket \ell : \text{goto } \ell' \rrbracket$  all  $E_m$  but one are erased, after which the remaining  $E_m$  recreates all of  
 them. As a consequence, every infinite branch  $\bar{u} \in E^\omega$  in the preproof has exactly one  
 thread in the  $E_m$  group. Suppose a factor  $e_1 e_2 \in E^2$  of  $\bar{u}$  and suppose their respective  
 sources and targets in  $G$  to be named  $\ell_1 \xrightarrow{e_1} \ell'_1$  and  $\ell_2 \xrightarrow{e_2} \ell'_2$ . The crucial observation is  
 that the piece of the  $E_m$  thread delimited by  $e_1$  has minimal formula equal to

$$\begin{cases} E_m \text{ (which is a } \mu\text{-formula)} & \text{if } \ell'_1 = \ell_2 \\ \nu X \dot{\iota}_m X & \text{if } \ell'_1 \neq \ell_2 \end{cases}$$

9 the formula  $\nu X \dot{\iota}_m X$  being a subformula of  $E_m$  able to recreate  $E_m$ . As a result:

- 10 • if  $\bar{u}$  has a suffix which is a valid path in  $G$  then the principal formula of the  $E_m$   
 11 thread is  $E_m$ , hence this thread is a  $\mu$ -thread, and there is no valid thread in the  
 12  $E_m$  group.
- 13 • otherwise, the principal formula of the  $E_m$  thread is  $\nu X \dot{\iota}_m X$ , and this thread is a  
 14  $\nu$ -thread, infinitely progressing, hence a validating thread.

This diagram sums up the behavior of the threads of the  $E_m$  group in  $\llbracket \ell : \text{goto } \ell' \rrbracket$ :



15 In order to prove the second part of the statement, that is that an infinite branch  $\bar{u} \in E^\omega$   
 16 in the preproof contains a validating thread in the  $B, C$  group iff. 0 occurs only finitely  
 17 in  $\bar{u}$ , we reason by case on the instructions involved.

### 3 PSPACE-completeness of the thread criterion

1 First note that  $B$  is a  $\nu$ -formula and that it is a subformula of  $C$ , which is a  $\mu$ -formula.  
 2 Observe also that  $B$  can recreate  $B$ , but  $B$  cannot recreate  $C$ . Therefore a  $\{B, C\}$ -thread,  
 3 in order to be validating, must end with an infinite alternation  $B \xrightarrow{(\nu)} (\perp \oplus B) \xrightarrow{(\oplus_1)} B \xrightarrow{(\nu)}$   
 4  $\dots$  Let us now take a closer look at what happens to  $B$  and  $C$  in the different branches  
 5 of the tree with back edges  $p$ . Formulas  $B$  and  $C$  are touched only at the bottom of the  
 6  $\llbracket \ell : \Gamma_\ell \rrbracket$  parts of the preproof  $\llbracket b \rrbracket$ , in the following way:

- 7 • when going through branch 0,  $B$  is erased and  $C$  forks into a new copy of  $B$  and a  
 8 recreated  $C$ .
- 9 • When going through any other branch, two threads are maintained: one at  $B$  and  
 10 one at  $C$ , and the thread at  $B$  progresses.

11 Therefore: if an infinite branch  $\bar{u}$  goes infinitely through 0,  $\bar{u}$  contains no infinite thread  
 12 at  $B$ , only an infinite thread at  $C$ , which is a  $\mu$ -thread, hence  $\bar{u}$  has no validating thread  
 13 in the  $B/C$  part of the root sequent. If, on the contrary,  $\bar{u}$  goes only finitely many  
 14 times through 0, it has then a thread at  $B$ , which is a  $\nu$ -thread, and which is infinitely  
 15 progressing, hence a validating thread.

These two diagrams sum up the behavior of the threads in the  $\{B, C\}$  group in  $\llbracket 0 : \rrbracket$  and  
 in all other  $\llbracket \ell : \Gamma'_\ell \rrbracket$  respectively:

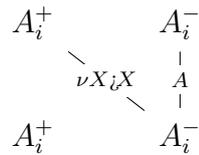


16

□

17 *Remark 14.* Because of lemma 6, the only infinite branches of  $\llbracket b \rrbracket$  whose validity is not  
 18 known in advance are the  $\bar{u} \in E^\omega$  which are valid paths in  $G$  going infinitely many  
 19 times through edge 0, and we know that these infinite branches may have validating  
 20 threads only in one of the  $k$  groups  $\{A_i^+, A_i^-\}_{1 \leq i \leq k}$ . Such an infinite branch can always  
 21 be factorized into  $u_0 0 u_1 0 u_2 0 \dots$  where the  $u_n$  do not contain 0. As the edge  $0 \in E$  has  
 22 source and target  $0 \xrightarrow{0} 1$ , and because of the hypothesis that  $\bar{u}$  is a path in  $G$ , for  $n \geq 1$   
 23 every  $u_n$  has source and target  $1 \xrightarrow{u_n} 0$ .

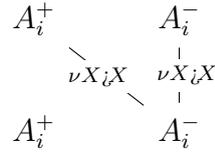
**Lemma 7.** Assume  $1 \xrightarrow{u} 0$   $\ell$ , which does not contain the edge 0. If  $u$  is a prefix of the  
 execution of  $b$  then the threads of  $\{A_i^+, A_i^-\}$  in  $0 \xrightarrow{0u} 0$   $\ell$  are



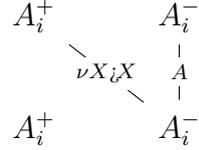


### 3 PSPACE-completeness of the thread criterion

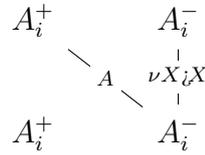
hence, by composition, the threads of  $\{A_i^+, A_i^-\}$  in  $0ve = 0u$  are still



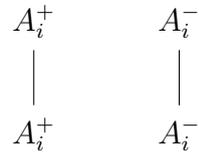
Otherwise  $v$  is a beginning of the execution of  $b$ . Then, by induction hypothesis, the threads of  $\{A_i^+, A_i^-\}$  in  $0 \xrightarrow{0v} \ell$  are



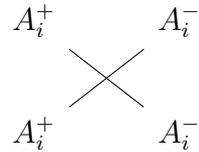
if  $X_i = \text{false}$  at the end of  $v$  and



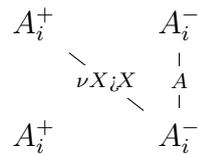
if  $X_i = \text{true}$  at the end of  $v$ . The next instruction executed by  $b$  is either  $I_\ell: X_i := \text{not } X_i$  or  $I_\ell: \text{if } X_i \text{ goto } \ell' \text{ else } \ell''$ . If the next instruction is  $I_\ell: X_i := \text{not } X_i$  then  $e$  has to be  $\ell \xrightarrow{\ell} ((\ell + 1) \bmod (m + 1))$ , so  $0u = 0ve$  is still a beginning of the execution of  $b$ , and then: for every  $j \neq i$ , the threads of  $\{A_j^+, A_j^-\}$  in  $e$  are



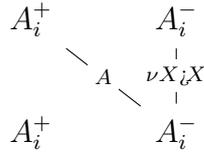
so the threads of  $\{A_j^+, A_j^-\}$  in  $0u$  have the same form as those in  $0v$ , while the value of  $X_j$  has not changed, hence the invariant is still respected for  $j$ . As for the threads in  $\{A_i^+, A_i^-\}$ , in  $e$  they are



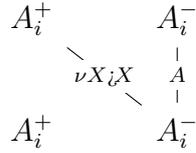
so if the threads of  $\{A_i^+, A_i^-\}$  in  $0v$  are



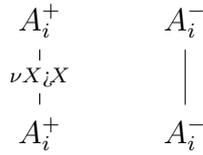
those in  $0u$  are



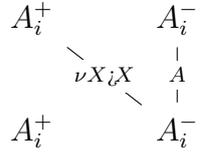
and *vice versa*. As the value of  $X_i$  is changed by this instruction, the invariant is still respected also for  $i$ . The last case we have to treat is when the next instruction executed by  $b$  after  $v$  is  $I_\ell: \text{if } X_i \text{ goto } \ell' \text{ else } \ell''$ . That means that  $v$  ends in vertex  $\ell$ , from which there are two edges:  $\ell \xrightarrow{\ell^+} \ell'$  and  $\ell \xrightarrow{\ell^-} \ell''$ . There are four cases, depending on the value of the variable  $X_i$  after  $v$ , and the choice of  $e \in \{\ell^+, \ell^-\}$ . If  $X_i = \text{false}$  and  $e = \ell^-$  then  $0u = 0ve$  is still a prefix of the execution of  $b$ . The threads of  $\{A_i^+, A_i^-\}$  in  $0v$  where, by induction hypothesis:



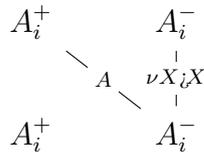
and those in  $e = \ell^-$  are



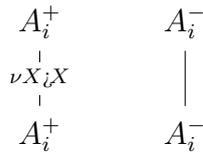
hence, by composition, the threads of  $\{A_i^+, A_i^-\}$  in  $0u$  are still



and the invariant is still respected. The situation is symmetric when  $X_i = \text{true}$  and  $e = \ell^+$ . If  $X_i = \text{true}$  and  $e = \ell^-$  then  $0u = 0ve$  is not a prefix of the execution of  $b$ . The threads of  $\{A_i^+, A_i^-\}$  in  $0v$  where, by induction hypothesis:



and those in  $e = \ell^-$  are



### 3 PSPACE-completeness of the thread criterion

hence, by composition, the threads of  $\{A_i^+, A_i^-\}$  in  $0u$  are

$$\begin{array}{ccc} A_i^+ & & A_i^- \\ & \searrow \nu X_i X & \downarrow \nu X_i X \\ & & A_i^- \\ & & \downarrow \nu X_i X \\ A_i^+ & & A_i^- \end{array}$$

1 (recall that  $\nu X_i X$  is a subformula of  $A$ ). The situation is symmetric when  $X_i = false$  and  
2  $e = l^+$ . So in all of these four cases the invariant is respected.

3

□

4 **Proposition 10.**  $\llbracket \cdot \rrbracket$  is a LOGSPACE reduction from  $\overline{\Pi(\mu\text{MALL}^\omega)}$  to  $\mathcal{B}_{false}$ .

5 *Proof.* For the LOGSPACE character: the only data that need to be remembered while  
6 constructing the preproof are integers like  $k, m, \ell, \ell'$ . Because  $\ell, \ell' \leq m$  and the entry  
7 has size  $\Omega(k + m)$ , this takes a space at most logarithmic in the size of the entry.

8 As for the fact that it is indeed a reduction: let us assume a  $b \in \text{BOOLE}_{false}$  and prove  
9 that  $\llbracket b \rrbracket \notin \Pi(\mu\text{MALL}^\omega) \Leftrightarrow b \in \mathcal{B}_{false}$ . Let  $G = (V, E)$  be the call-graph of  $b$ , as defined in  
10 Definition 54. Following remark 12, we denote by elements of  $E^\omega$  the infinite branches of  
11  $\llbracket b \rrbracket$ . There are two cases: either  $b \in \mathcal{B}_{false}$  and we have to prove that  $p \notin \Pi(\mu\text{MALL}^\omega)$ ,  
12 or  $b \notin \mathcal{B}_{false}$  and we have to prove that  $p \in \Pi(\mu\text{MALL}^\omega)$ . First case: if  $b \in \mathcal{B}_{false}$ :  
13 the execution of  $b$  induces a finite path  $u = 1 \xrightarrow{*} 0$  in  $G$ . This finite path can be  
14 completed into  $v = 0 \xrightarrow{0} 1 \xrightarrow{u} 0$ . Then  $v^\omega$  is an invalid branch of  $\llbracket p \rrbracket_\omega$ . Here we use  
15 the fact that when  $b$  terminates, every variable has value *false*. Second case: if  $b \notin \mathcal{B}_{false}$ :  
16 let  $\mathcal{P}_1 = \{vw_\infty \mid v \in E^*, w_\infty \in E^\omega \text{ and } w_\infty \text{ is a path in } G\}$  and  $\mathcal{P}_2 = \{v_\infty \in \mathcal{P}_1 \mid$   
17  $0 \text{ occurs infinitely in } v_\infty\}$ . By construction,  $\mathcal{P}_2 \subseteq \mathcal{P}_1 \subseteq E^\omega$ . We will prove three facts:  
18 that every branch  $v_\infty \in E^\omega \setminus \mathcal{P}_1$  is thread-valid, that every branch  $v_\infty \in \mathcal{P}_1 \setminus \mathcal{P}_2$  is  
19 thread-valid and that every branch  $v_\infty \in \mathcal{P}_2$  is thread-valid. These three facts, together  
20 with the fact that  $(E^\omega \setminus \mathcal{P}_1) \cup (\mathcal{P}_1 \setminus \mathcal{P}_2) \cup \mathcal{P}_2 = E^\omega$ , are enough to conclude that every  
21 branch  $v_\infty \in E^\omega$  is thread-valid. The first fact, that every branch  $v_\infty \in E^\omega \setminus \mathcal{P}_1$  is  
22 thread-valid, is due to the thread going through the  $E_m$ . The second fact, that every  
23 branch  $v_\infty \in \mathcal{P}_1 \setminus \mathcal{P}_2$  is thread-valid, is due to the thread going through  $B$ . The third fact,  
24 that every branch  $v_\infty \in \mathcal{P}_2$  is thread-valid, is due to the fact that  $b$  is non-terminating  
25 and that, because of that, one of the  $2k$  threads going through the  $A$  is valid. □

**Theorem 4.** *The problem  $\Pi(\mu\text{MALL}^\omega)$  is PSPACE-hard:*

$$\text{PSPACE} \leq \Pi(\mu\text{MALL}^\omega)$$

26 *Proof.* We reduce from  $\mathcal{B}_{false}$ , which is PSPACE-complete by Lemma 2. More precisely,  
27 we reduce  $\mathcal{B}_{false}$  to  $\overline{\Pi(\mu\text{MALL}^\omega)}$ , the complement of  $\Pi(\mu\text{MALL}^\omega)$ . This is enough because

1 PSPACE is closed under complements, in the same way as all deterministic classes. The  
 2 reduction  $[\cdot]: \text{BOOLE}_{false} \rightarrow \Pi_0(\mu\text{MALL}^\omega)$  is defined in Definition 55. It is a LOGSPACE  
 3 reduction, by Proposition 10, hence it is also a polynomial-time reduction.  $\square$

4 *Remark 15.* In fact, since our construction do not use the (cut) rule, the cut-free fragment  
 5 of  $\Pi(\mu\text{MALL}^\omega)$  is already PSPACE-hard.

### 6 3.2.4 Generalisation to other systems

7 Systems of circular proofs with least and greatest fixed points can and have been built for  
 8 other logics, either with different connectives (LK, LJ) or with added modalities (LK $\circ$ ,  
 9 LK $\square\lozenge$ ). Such systems, which can also be seen as variants of  $\mu$ -calculus, are described in  
 10 [Vardi, 1988, Kaivola, 1995a, Dax, Hofmann, and Lange, 2006, Doumane, Baelde, Hirschi,  
 11 and Saurin, 2016, Doumane, 2017a,b]. They all share the fact that MALL can be seen as  
 12 a fragment of them, and therefore it is easy to translate  $\mu\text{MALL}^\omega$  proofs into them and  
 13 transfer to them our result of PSPACE-completeness:

14 *Remark 16.* There is a  $\mu\text{LJ}$  defined by Clairambault [2009, 2010], which is a finitary  
 15 calculus in the spirit of  $\mu\text{MALL}$ , with explicit invariants of induction and coinduction.  
 16 The  $\mu\text{LJ}^\omega$  we are considering here is an infinitary one, in which proofs are distinguished  
 17 from invalid preproofs by a thread criterion, in the spirit of  $\mu\text{MALL}^\omega$ .

18 **Corollary 2** (to Theorem 4). *The decidability of the thread criterion is also PSPACE-*  
 19 *complete in  $\mu\text{LJ}^\omega$ ,  $\mu\text{LK}^\omega$ ,  $\mu\text{LK}\circ^\omega$  and  $\mu\text{LK}\square\lozenge^\omega$ .*

20 *Proof.* It is true for  $\mu\text{LJ}^\omega$  because every preproof  $\pi$  in  $\mu\text{MALL}^\omega$  can be translated to a  
 21 preproof  $\pi^*$  in  $\mu\text{LJ}^\omega$ , such that  $\pi^*$  is thread-valid iff.  $\pi$  is, by translating MALL connectives  
 22 into LJ connectives and translating  $\vdash \Gamma$  into  $|\Gamma^\perp| \vdash \perp$ .

23 It is true for the other mentioned systems because  $\mu\text{LJ}^\omega$  is a fragment of them, therefore  
 24 every  $\mu\text{LJ}^\omega$  preproof is already a preproof in each of them.

25 Details of how the constructions adapt to LK and LJ are given below.  $\square$

26 *Remark 17.* Note that the following adaptations from MALL to LK and LJ might be  
 27 simplified by replacing  $F \vee F$  by  $F$ ,  $X \wedge X$  by  $X$ ,  $\perp \vee X$  by  $X$  and so on. A lot of the  
 28 complexity of these formulas is there to simulate structural rules in the linear world. When  
 29 we are in the classical world, these workaround may not be needed anymore. Anyway,  
 30 the translation we present here is simply to show that even without thinking about that,

### 3 PSPACE-completeness of the thread criterion

- 1 a straightforward translation is possible, which gives the same PSPACE-completeness
- 2 result with LK and LJ and their extensions.

#### 3 Generalisation to $\mu\text{LK}$ , $\mu\text{LK}\circ$ and $\mu\text{LK}\square\lozenge$

$$\begin{aligned}
 A &= \imath(\nu X \imath X) & B &= \nu X(\perp \vee X) & C &= \mu X(B \vee X) & E_n &= \imath_n(\nu X \imath_n X) \\
 \imath F &= \mu X(F \vee (\perp \vee (X \vee X))) & \imath_n F &= \mu X(\perp \vee (X \vee (\underbrace{F \vee \dots \vee F}_{n-1}))) \\
 D_n &= \mu X(\underbrace{X \wedge \dots \wedge X}_n)
 \end{aligned}$$

- 4 We define  $\llbracket b \rrbracket$  to be the preproof

$$\frac{\begin{array}{c} \vdots \llbracket 0 : \cdot \rrbracket \\ \vdots \llbracket 1 : I_1 \rrbracket \\ \dots \\ \vdots \llbracket m : I_m \rrbracket \end{array}}{\text{(ROOT)} \vdash A^{2k}, B, C, D_2, D_m, E_m^m \vdash A^{2k}, B, C, D_2, D_m, E_m^m \vdash A^{2k}, B, C, D_2, D_m, E_m^m}_{(\mu), (\wedge)^{m-1}} \quad (3.4)$$

- 6 See the details of all branches on Figures 3.7 and 3.8.

#### 7 Generalisation to $\mu\text{LJ}$

$$\begin{aligned}
 A &= \imath(\mu X \imath X) & B &= \mu X(\top \wedge X) & C &= \nu X(B \wedge X) & E_n &= \imath_n(\mu X \imath_n X) \\
 \imath F &= \nu X(F \wedge (\top \wedge (X \wedge X))) & \imath_n F &= \nu X(\top \wedge (X \wedge (\underbrace{F \wedge \dots \wedge F}_{n-1}))) \\
 D_n &= \nu X(\underbrace{X \vee \dots \vee X}_n)
 \end{aligned}$$

- 8 We define  $\llbracket b \rrbracket$  to be the preproof

$$\frac{\begin{array}{c} \vdots \llbracket 0 : \cdot \rrbracket \\ \vdots \llbracket 1 : I_1 \rrbracket \\ \dots \\ \vdots \llbracket m : I_m \rrbracket \end{array}}{\text{(ROOT)} A^{2k}, B, C, D_2, D_m, E_m^m \vdash A^{2k}, B, C, D_2, D_m, E_m^m \vdash A^{2k}, B, C, D_2, D_m, E_m^m}_{(\nu^L), (\vee^L)^{m-1}} \quad (3.5)$$

- 10 See the details of all branches on Figures 3.9 and 3.10.

1 *Remark 18.* The interested open question is: does the result still stand in the guarded  
 2 fragments of  $\mu\text{LK}\bigcirc^\omega$  and  $\mu\text{LK}\square\blacklozenge^\omega$ ? We conjecture that it does and that the method  
 3 we illustrate here on  $\mu\text{MALL}$  can apply as well to the guarded cases of  $\mu$ -calculi with  
 4 modalities.

### 5 **3.3 Comments on our approach and discussion of related** 6 **works**

7 Our proof for the PSPACE-completeness of the thread criterion is an encoding and an  
 8 adaptation to our setting of the proof used by Lee, Jones and Ben Amram to prove  
 9 that size-change termination is PSPACE-complete [Lee, Jones, and Ben-Amram, 2001].  
 10 We reduce, as they do, from the problem of termination of boolean programs and the  
 11 thread diagrams that we have used to describe the preproof generated by the reduction  
 12 are very similar to the size-change graphs generated by their reduction; this is in fact  
 13 what has guided the design of this preproof: formula  $A$  mimicks the  $X_i, \bar{X}_i$  part of their  
 14 graphs and formulas  $B$  and  $C$  adapt the  $Z$  part of their graphs. We had to add the  
 15 formulas  $D_2$  and  $D_m$  in order to have branching rules in the preproof. One of the main  
 16 novelties of our reduction, compared to the reduction of Lee, Jones and Ben Amram for  
 17 size-change termination, lies in the  $E_m$  and  $\llbracket \ell : \text{goto } \ell' \rrbracket$  part of the constructed preproof,  
 18 which has no equivalent in the size-change graphs obtained by their reduction. This part  
 19 of our construction allows us to construct a preproof which is a tree with back edges,  
 20 hence proving that the thread criterion is PSPACE-complete even when preproofs are  
 21 represented by trees with back edges. We could in fact drop the  $E_m$  and  $\llbracket \ell : \text{goto } \ell' \rrbracket$   
 22 part of the construction by constructing  $\llbracket b \rrbracket$  as a rooted graph instead of a tree with  
 23 back edges. The constructions proofs are still correct — and shorter. The caveat is that  
 24 it only proves the thread-criterion to be PSPACE-hard in graph-shaped preproofs and  
 25 not in b.e.-tree-shaped preproofs. Furthermore, we could not have filled this gap by  
 26 simply unfolding the graph into a tree with back edges, for it could lead, as shown in the  
 27 following example, to an exponential blow-up in size, which would prevent the reduction  
 28 to be LOGSPACE, or even P. The following boolean program:

```

1:if X then goto 2 else goto 2
2:if X then goto 3 else goto 3
   :
n:if X then goto n + 1 else goto n + 1

```

29 will be translated to a graph-shaped preproof of size  $\Theta(n)$  but the unfolding of this  
 30 preproof into a b.e.-tree-shaped preproof will have size  $\Theta(2^n)$ . Therefore we had to be

### 3 PSPACE-completeness of the thread criterion

1 clever in order to target trees with back edges by *simulating* several vertices with a single  
2 one; this is accomplished by the  $E_m$  and  $[[\ell:\text{goto } \ell']]$ .

3 This improvement of the reduction of Lee, Jones and Ben Amram could in fact be adapted  
4 in the other direction, to show that size-change termination is already PSPACE-complete  
5 even when restricted to programs with only one function (in the terminology of [Lee,  
6 Jones, and Ben-Amram, 2001]), that is when the corresponding call graph / control flow  
7 graph has only one vertex.

8 If, as it is commonly believed,  $\text{NP} \neq \text{PSPACE}$ , our result implies that there is no way to  
9 add a polynomial quantity of information to a preproof so that its thread-validity can  
10 be checked in polynomial time. This can be seen as a problem, both for the complexity  
11 of proof search and proof verification. It suggests trying to find restrictions of the  
12 thread criterion which will be either decidable or certifiable in polynomial time, while  
13 keeping enough expressivity to validate interesting proofs. This will be the subject of the  
14 Chapters 4 and 5 of this thesis.

15 We recalled in section 3.1 that thread validity is decidable in PSPACE, and we did so by  
16 reducing to the problem of language inclusion for  $\omega$ -parity-automata. The original size-  
17 change article [Lee, Jones, and Ben-Amram, 2001] gives two different methods to check  
18 size-change termination, the first one is based on reducing to inclusions of  $\omega$ -languages  
19 defined by finite automata while the second one is a direct, graph-based approach. It is in  
20 fact possible to use this more direct method to decide the thread criterion, and this has  
21 already been done by Dax, Hofmann, and Lange [2006], who remark furthermore that  
22 this method leads to a more efficient implementation than the automata-based one.

## 23 3.4 Conclusion

24 In the present chapter, we analyzed the complexity of deciding the validity of circular  
25 proofs in  $\mu\text{MALL}$  logic: while the problem was already known to be in PSPACE, we  
26 established here its PSPACE-completeness. In doing so, we drew inspiration from the  
27 PSPACE-completeness proof of SCT even though we defer at some crucial points on  
28 order to build our reduction and carry our proof taking into account the specific forms of  
29 circular proofs.

30 Our proof adapt straightforwardly to a number of other circular proof systems based on  
31 sequent calculus such as intuitionistic or classical proof systems in addition to the linear  
32 case on which we focused here.

33 While our result can be seen as negative one for circular proofs, it does not prevent  
34 actual implementations to be tractable and usable on many situations as exemplified by

1 the Cyclist prover for instance. In such systems, validity checking does not seem to be  
2 the bottleneck in circular proof construction as compared with the complexity that is  
3 inherent to exploring and backtracking in the search tree [Brotherston, Gorogiannis, and  
4 Petersen, 2012, Rowe and Brotherston, 2017, Tellez and Brotherston, 2017].

5 Our work suggests deep connections between thread-validity and SCT that we only  
6 touched upon in the previous section. This confirms connections previously hinted by  
7 other authors [Dax, Hofmann, and Lange, 2006, Hyvernath, 2014, 2019, Lepigre and  
8 Raffalli, 2019] that we plan to investigate further in the future.

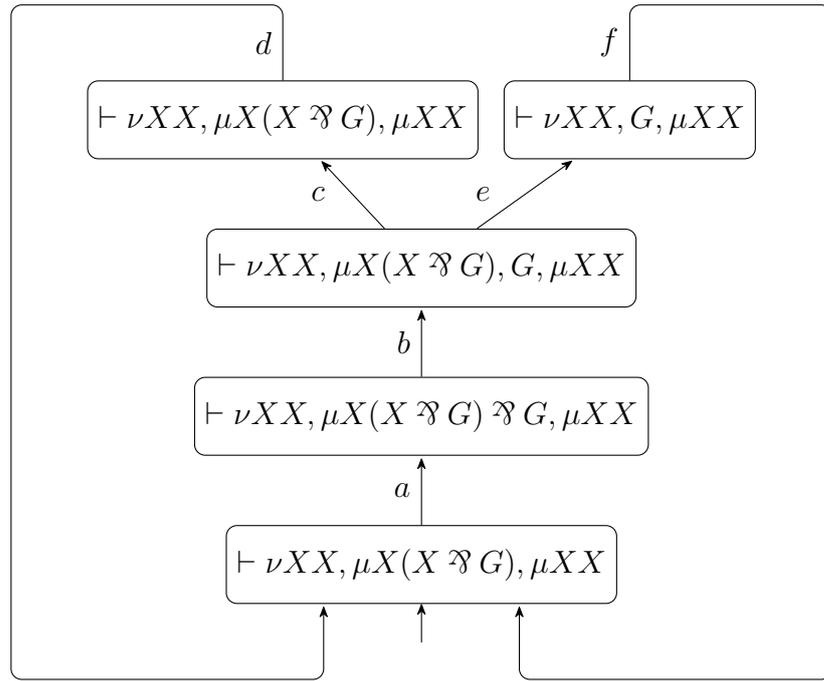


Figure 3.1. Automaton recognizing the branches of the preproof (3.1).

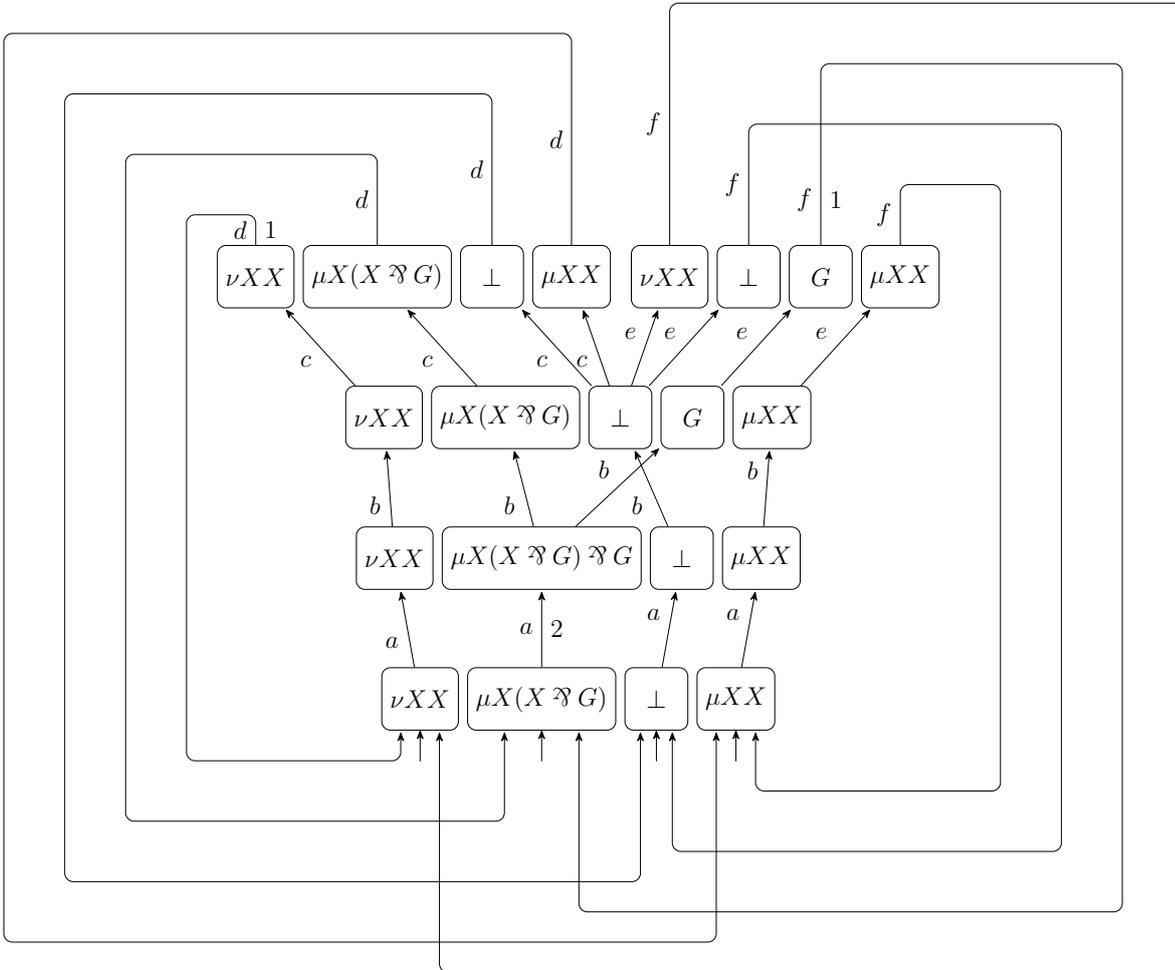


Figure 3.2. Automaton recognizing the valid branches of the preproof (3.1).

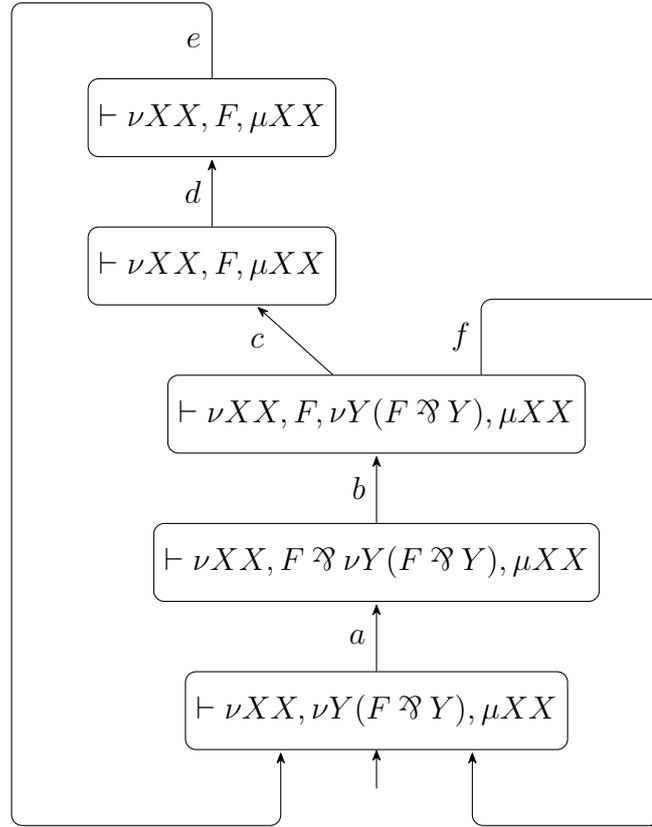


Figure 3.3. Automaton recognizing the branches of the preproof (3.2).

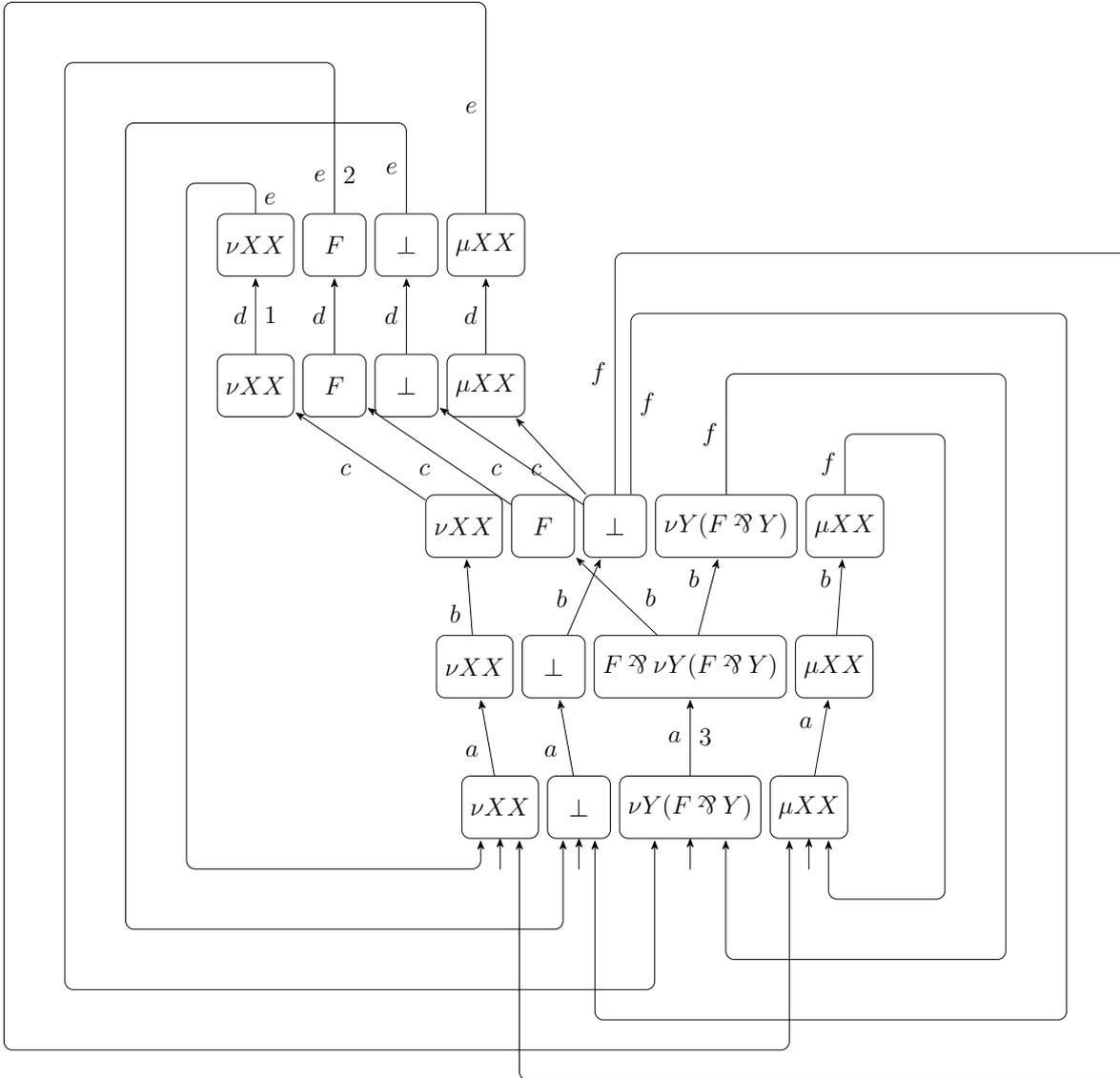


Figure 3.4. Automaton recognizing the valid branches of the preproof (3.2).

$$\begin{array}{c}
 \llbracket \ell : \text{goto } \ell' \rrbracket \quad := \\
 \text{Back edge to (ROOT)} \\
 \frac{\frac{\frac{\frac{\vdash A, \dots, A, B, C, D_2, D_m, E_m, \dots, E_m}{\vdash A, \dots, A, B, C, D_2, D_m, \underbrace{\nu X \dot{\iota}_m X, \dots, \nu X \dot{\iota}_m X}_{\ell'-1}, E_m, \underbrace{\nu X \dot{\iota}_m X, \dots, \nu X \dot{\iota}_m X}_{m-\ell'}}{(\nu)^{m-1}}}{\vdash A, \dots, A, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_{\ell-1}, E_m, \underbrace{E_m, \dots, E_m}_{m-\ell}}}{(\mu, (\oplus_1), (\wp))^{m-1}}}{\vdash \underbrace{A, \dots, A}_{2k}, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_{\ell-1}, E_m, \underbrace{E_m, \dots, E_m}_{m-\ell}}}{((\mu), (\oplus_0), (\perp))^{m-1}}
 \end{array}$$

Figure 3.5. Back edges of the preproof

$$\begin{array}{c}
 \vdots \llbracket 0 : \text{goto } 1 \rrbracket \\
 \frac{\frac{\frac{\vdash (A, A)^k, B, C, D_2, D_m, E_m^m}{\vdash (\nu X \dot{\iota}_m X, A)^k, B, C, D_2, D_m, E_m^m}}{(\nu)^k}}{(\mu), (\oplus_0)}^k \\
 \llbracket 0 : \rrbracket \quad := \quad \frac{\frac{\frac{\frac{\vdash (\underline{A}, A)^k, B, C, D_2, D_m, E_m^m}{\vdash \underline{A}^k, B, C, D_2, D_m, E_m^m}}{(\mu), (\oplus_1), (\oplus_1), (\wp)}^k}}{(\mu), (\oplus_1), (\oplus_0), (\perp)}^k}}{(\mu), (\wp)} \\
 \frac{\frac{\vdash A^{2k}, \underline{C}, D_2, D_m, E_m^m}{\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m}}{(\nu), (\oplus_0), (\perp)}}{(\mu), (\wp)} \\
 \vdots \llbracket \ell : \text{goto } (\ell + 1 \bmod m + 1) \rrbracket \\
 \llbracket \ell : X_i := \text{not } X_i \rrbracket \quad := \quad \frac{\frac{\frac{\frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, A, \nu X \dot{\iota}_m X, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}}{(\nu), (\oplus_1)}}{(\mu), (\oplus_1)}}{(\mu), (\oplus_0)}}{(\mu), (\&)} \\
 \llbracket \ell : \text{if } X_i \text{ then goto } \ell' \text{ else } \ell'' \rrbracket \quad := \\
 \frac{\frac{\frac{\frac{\vdots \llbracket \ell : \text{goto } \ell' \rrbracket}{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}}{(\nu)}}{(\mu), (\oplus_0)}}{(\mu), (\oplus_0)}}{\frac{\frac{\frac{\frac{\vdots \llbracket \ell : \text{goto } \ell'' \rrbracket}{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}}{(\nu)}}{(\mu), (\oplus_0)}}{(\mu), (\oplus_0)}}{(\mu), (\&)}}{\frac{\frac{\vdash A^{2k}, B, C, D_2, D_m, E_m^m}{\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m}}{(\nu), (\oplus_1)}}{(\mu), (\&)}}
 \end{array}$$

 Figure 3.6. Premises  $p_\ell$  of the preproof

$$\begin{array}{c}
\vdots \llbracket 0:\text{goto } 1 \rrbracket \\
\frac{\vdash (A, A)^k, B, C, D_2, D_m, E_m^m}{\vdash (\nu X \dot{\iota} X, A)^k, B, C, D_2, D_m, E_m^m} (\nu)^k \\
\frac{\vdash (\underline{A}, A)^k, B, C, D_2, D_m, E_m^m}{\vdash \underline{A}^k, B, C, D_2, D_m, E_m^m} ((\mu), (\nu_0))^k \\
\llbracket 0: \rrbracket := \frac{\vdash (\underline{A}, A)^k, B, C, D_2, D_m, E_m^m}{\vdash \underline{A}^k, B, C, D_2, D_m, E_m^m} ((\mu), (\nu_1), (\nu_1), (\nu))^k \\
\frac{\vdash (\underline{A}, A)^k, B, C, D_2, D_m, E_m^m}{\vdash A^{2k}, \underline{C}, D_2, D_m, E_m^m} ((\mu), (\nu_1), (\nu_0), (\perp))^k \\
\frac{\vdash A^{2k}, \underline{C}, D_2, D_m, E_m^m}{\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m} (\mu), (\nu) \\
\frac{\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m}{\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m} (\nu), (\nu_0), (\perp) \\
\vdots \llbracket \ell:\text{goto } (\ell + 1 \bmod m + 1) \rrbracket \\
\frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, \underline{A}, A, A^{2(k-i)}, \underline{B}, C, D_2, D_m, E_m^m} (\nu), (\nu_1) \\
\llbracket \ell: X_i := \text{not } X_i \rrbracket := \frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, \underline{A}, A, A^{2(k-i)}, \underline{B}, C, D_2, D_m, E_m^m} (\nu), (\nu_1) \\
\llbracket \ell:\text{if } X_i \text{ then goto } \ell' \text{ else } \ell'' \rrbracket := \\
\frac{\vdots \llbracket \ell:\text{goto } \ell' \rrbracket \quad \vdots \llbracket \ell:\text{goto } \ell'' \rrbracket}{\frac{\frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, A, \nu X(\dot{\iota} X), A^{2(k-i)}, B, C, D_2, D_m, E_m^m} (\nu) \quad \frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, \nu X(\dot{\iota} X), A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m} (\nu)}{\vdash A^{2(i-1)}, A, \underline{A}, A^{2(k-i)}, B, C, D_2, D_m, E_m^m} (\mu), (\nu_0)} \quad \frac{\vdash A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m}{\vdash A^{2(i-1)}, \underline{A}, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m} (\mu), (\nu_0)}{\vdash A^{2k}, B, C, \underline{D}_2, D_m, E_m^m} (\mu), (\wedge)}{\vdash A^{2k}, B, C, \underline{D}_2, D_m, E_m^m} (\nu), (\nu_1) \\
\vdash A^{2k}, \underline{B}, C, D_2, D_m, E_m^m} (\nu), (\nu_1)
\end{array}$$

Figure 3.7. Premises  $p_\ell$  of the preproof for  $\mu\text{LK}^\omega$ 

$$\begin{array}{c}
\llbracket \ell:\text{goto } \ell' \rrbracket := \\
\text{Back edge to (ROOT)} \\
\frac{\vdash A, \dots, A, B, C, D_2, D_m, E_m, \dots, E_m}{\vdash A, \dots, A, B, C, D_2, D_m, \underbrace{\nu X \dot{\iota}_m X, \dots, \nu X \dot{\iota}_m X}_{\ell-1}, E_m, \underbrace{\nu X \dot{\iota}_m X, \dots, \nu X \dot{\iota}_m X}_{m-\ell'}} (\nu)^{m-1} \\
\frac{\vdash A, \dots, A, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_{\ell-1}, E_m, \underbrace{E_m, \dots, E_m}_{m-\ell}}{\vdash A, \dots, A, B, C, D_2, D_m, \underbrace{E_m, \dots, E_m}_{2k}, \underbrace{E_m, \dots, E_m}_{\ell-1}, \underbrace{E_m, \dots, E_m}_{m-\ell}} ((\mu), (\nu_1), (\nu))^{m-1}} ((\mu), (\nu_0), (\perp))^{m-1}
\end{array}$$

Figure 3.8. Back edges of the preproof for  $\mu\text{LK}^\omega$

$$\begin{array}{c}
 \vdots \llbracket 0:\text{goto } 1 \rrbracket \\
 \frac{(A, A)^k, B, C, D_2, D_m, E_m^m \vdash}{(\underline{\mu X} \dot{\downarrow} X, A)^k, B, C, D_2, D_m, E_m^m \vdash} (\mu^L)^k \\
 \llbracket 0: \rrbracket := \frac{\frac{(\underline{A}, A)^k, B, C, D_2, D_m, E_m^m \vdash}{\underline{A}^k, B, C, D_2, D_m, E_m^m \vdash} ((\nu^L), (\wedge_1^L), (\wedge_1^L), (\wedge^L))^k}{(\underline{A}, A)^k, B, C, D_2, D_m, E_m^m \vdash} ((\nu^L), (\wedge_1^L), (\wedge_0^L), (\top^L))^k}{\frac{A^{2k}, \underline{C}, D_2, D_m, E_m^m \vdash}{A^{2k}, \underline{B}, C, D_2, D_m, E_m^m \vdash} (\nu^L), (\wedge^L)}{A^{2k}, \underline{B}, C, D_2, D_m, E_m^m \vdash} (\mu^L), (\wedge_0^L), (\top^L)} \\
 \vdots \llbracket \ell:\text{goto } (\ell + 1 \bmod m + 1) \rrbracket \\
 \llbracket \ell: X_i := \text{not } X_i \rrbracket := \frac{A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash}{A^{2(i-1)}, \underline{A}, A, A^{2(k-i)}, \underline{B}, C, D_2, D_m, E_m^m \vdash} (\mu^L), (\wedge_1^L) \\
 \llbracket \ell:\text{if } X_i \text{ then goto } \ell' \text{ else } \ell'' \rrbracket := \\
 \frac{\frac{\vdots \llbracket \ell:\text{goto } \ell' \rrbracket}{A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash} (\mu^L)}{A^{2(i-1)}, A, \underline{\mu X}(\dot{\downarrow} X), A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash} (\nu^L), (\wedge_0^L)}{\frac{\frac{\vdots \llbracket \ell:\text{goto } \ell'' \rrbracket}{A^{2(i-1)}, A, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash} (\mu^L)}{A^{2(i-1)}, \underline{\mu X}(\dot{\downarrow} X), A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash} (\nu^L), (\wedge_0^L)}{A^{2(i-1)}, \underline{A}, A, A^{2(k-i)}, B, C, D_2, D_m, E_m^m \vdash} (\nu^L), (\wedge^L)}{\frac{A^{2k}, B, C, \underline{D}_2, D_m, E_m^m \vdash}{A^{2k}, \underline{B}, C, D_2, D_m, E_m^m \vdash} (\mu^L), (\wedge_1^L)}{A^{2k}, \underline{B}, C, D_2, D_m, E_m^m \vdash} (\nu^L), (\nu^L)}
 \end{array}$$

 Figure 3.9. Premises  $p_\ell$  of the preproof for  $\mu\text{LJ}^\omega$ 

$$\begin{array}{c}
 \llbracket \ell:\text{goto } \ell' \rrbracket := \\
 \frac{\text{Back edge to (ROOT)}}{A, \dots, A, B, C, D_2, D_m, E_m, \dots, E_m \vdash} (\mu^L)^{m-1} \\
 \frac{A, \dots, A, B, C, D_2, D_m, \underbrace{\mu X \dot{\downarrow} X, \dots, \mu X \dot{\downarrow} X}_{\ell-1}, E_m, \underbrace{\mu X \dot{\downarrow} X, \dots, \mu X \dot{\downarrow} X}_{m-\ell'} \vdash}{\frac{A, \dots, A, B, C, D_2, D_m, \underline{E}_m \vdash}{\underbrace{A, \dots, A}_{2k}, B, C, D_2, D_m, \underbrace{\underline{E}_m, \dots, \underline{E}_m}_{\ell-1}, E_m, \underbrace{\underline{E}_m, \dots, \underline{E}_m}_{m-\ell}} ((\nu^L), (\wedge_0^L), (\top^L))^{m-1}} ((\nu^L), (\wedge_1^L), (\wedge^L))^{m-1}
 \end{array}$$

 Figure 3.10. Back edges of the preproof for  $\mu\text{LJ}^\omega$

## 1 4 A polynomial sub-criterion

2 We have shown in the previous chapter that deciding the validity of the circular rep-  
3 resentation of a preproof, with respect to the thread criterion, is a PSPACE-complete  
4 problem.

5 This implies in particular that there is probably no subexponential algorithm to check  
6 the validity of a circular representation. This also implies that there is probably no way  
7 to certify the validity of a circular preproof, so that it can be checked in polynomial time,  
8 without adding to it an exponential quantity of information.

9 We tackle both these problems by taking a different approach. Instead of trying to handle  
10 the existing criterion, we will look for a new criterion.

11 The thread criterion has very good properties. In particular, Baelde, Doumane, and  
12 Saurin [2016] proved that it guarantees soundness with respect to boolean interpretation  
13 and full cut elimination. Therefore, we would like our new criterion to have the following  
14 properties:

- 15 • that any preproof valid for the new criterion is also valid for the thread criterion,  
16 so that it inherits its soundness properties,
- 17 • that this new criterion is decidable in reasonable time,
- 18 • that it is possible to add a reasonable amount of information to the circular  
19 representation of a preproof so that the validity of this certified preproof may be  
20 checked in linear time,
- 21 • that this new criterion is expressive enough.

22 In this chapter, we provide such a criterion, which we called the **loop criterion**.

23 There is an essential difference between the thread criterion and our loop criterion. In  
24 usual proof theory, a proof is a finite tree, made of logical inferences, and a proof is  
25 correct as soon as each one of its inferences is correct. This is not true anymore with  
26 logics such as  $\mu\text{MALL}^\omega$ , in which the soundness of each inference only ensure that the  
27 tree is a preproof, and a proof also has to pass the thread criterion, which is a global

#### 4 A polynomial sub-criterion

1 criterion. The PSPACE-complete nature of the thread criterion forbids that it could have  
2 a local definition. It is, and has to be, a *global* criterion, which cannot be turned into  
3 a local criterion. In contrast, our **loop criterion** gives rise to a proof system in which  
4 the validity of a proof is a purely local property. More precisely, with our loop criterion  
5 comes an enriched proof system, made of labelled proof trees, with two properties:

- 6 1. if the circular representation of a preproof is valid for our loop criterion, it is  
7 possible to label it so as to obtain a proof in our labelled proof system,
- 8 2. in this labelled proof system, a proof is correct if and only if each one of its  
9 individual inferences is correct.

10 This would not be possible with the thread criterion, or the labeling would probably  
11 need to be at least exponential in the size of the proof.

12 There is another important difference between the usual thread criterion and our new  
13 **loop criterion**. The thread criterion is really a criterion on the infinite preproof, not on  
14 its representation. This means that if you have two different circular representations of  
15 the same infinite preproof, either they are both valid or both invalid. In contrast, our  
16 criterion really depends on the finite, circular representation, and not only on the infinite  
17 preproof which is represented. This means that among the circular representations of a  
18 given infinite preproof, it may happen that some of them are valid for our loop criterion  
19 and some other are not.

20 In this chapter we prove that:

- 21 • If a circular representation of an infinite preproof is valid for our loop criterion  
22 then this preproof is valid for the thread criterion.
- 23 • Our loop criterion can be decided in quadratic time.
- 24 • Every circular representation which is valid for our loop criterion can be labelled so  
25 as to be turned into a proof in our labelled proof system, in which the validity of  
26 the proof only depends on the validity of each individual inference, and is therefore  
27 checkable in linear time.
- 28 • There is a canonical translation of the finitary proofs in the style of Baelde [2012]  
29 into circular representations of infinite proofs, which are valid for the thread criterion.  
30 Those representations are also valid for our more restricted loop criterion. In other  
31 words, our labelling method capture the image of all finitary  $\mu$ MALL proofs.

## 1 4.1 $\mathcal{L}$ -proofs: labelling as validity

2 In this subsection, we briefly mention an alternative approach to ensure validity of  
 3  $\mu\text{MALL}^\infty$  preproofs, aiming at motivating the tools used in the remainder of this chapter.  
 4 The idea is to witness thread progress by adding labels on some formulas, much in the  
 5 spirit of Stirling [2014], Afshari and Leigh [2017].

6 **Definition 56** (Labelled formulas). Let  $\mathcal{L}$  be an infinite countable set of *atoms* and call  
 7 *labels* any finite list of atoms. Let  $\mathcal{F}^\mathcal{L} := \{\sigma^L \mid \sigma \in \{\mu, \nu\}, L \in \text{list}(\mathcal{L})\}$ . *Labelled formulas*,  
 8 or  $\mathcal{L}$ -formulas, are defined as  $\mu\text{MALL}$  formulas, by replacing  $\mathcal{F} = \{\mu, \nu\}$  with  $\mathcal{F}^\mathcal{L}$  in  
 9 the grammar of formulas, in Definition 12. Negation is lifted to labelled formulas, as  
 10  $(\mu^L X A)^\perp = \nu^L X A^\perp$ . We write  $\sigma X A$  for  $\sigma^\emptyset X A$ , and standard, unlabelled formulas can  
 11 thus be seen as labelled formulas where every label is empty. We define a *label-erasing*  
 12 function  $[\bullet]$  that associates to every  $\mathcal{L}$ -formula  $A$  the  $\mu\text{MALL}$ -formula  $[A]$  obtained by  
 13 erasing every label and satisfying  $[\sigma^L X B] = \sigma X [B]$ .

14 The standard  $\mu\text{MALL}^\infty$  proof system is adapted, to handle labels, by updating the (id)  
 15 and  $(\nu)$  rules of Figures 2.1 and 2.4 as

$$\frac{[A] = [B]^\perp}{\vdash A, B} \text{ (id')} \qquad \frac{\vdash A[\nu^{L,a} X A[X]], \Gamma}{\vdash \nu^L X A[X], \Gamma} \text{ } (\nu_b(a))$$

16 where, in  $(\nu_b(a))$ ,  $a$  must be a fresh label name, meaning that  $a$  does not appear free in  
 17 the conclusion sequent of  $(\nu_b(a))$  (in particular,  $a \notin L$ ).

18 Since we are in a one-sided framework, only labels on  $\nu$  operators are relevant. Therefore,  
 19 from now on, formulas have non-empty labels only on  $\nu$  and we require, for the cut  
 20 inference, that all labels of cut formulas are empty.

21 **Definition 57.**  *$\mathcal{L}$ -preproofs* are, as in Definition 18, possibly infinite derivations using  
 22  $\mathcal{L}$ -formulas, with the constraint that for every two inferences  $(\nu_b(a))$  and  $(\nu_b(b))$ , if one is  
 23 above the other, then  $a \neq b$ .

24 And the validity condition is expressed in terms of labels in the following definition:

25 **Definition 58** ( $\mathcal{L}$ -proof). An  *$\mathcal{L}$ -proof* is an  $\mathcal{L}$ -preproof such that for every infinite  
 26 branch  $\gamma = (s_i)_{i \in \omega}$  (seen as a sequence of sequent occurrences), there exists a sequence  
 27  $(\nu^{L_i} X G_i)_{i \in \omega}$  and a strictly increasing function  $\epsilon$  on natural numbers such that for every  
 28  $i \in \omega$ ,

#### 4 A polynomial sub-criterion

- 1 1. the formula  $\nu^{L_i} XG_i$  is principal in  $s_{\epsilon(i)}$
- 2 2.  $[\nu^{L_i} XG_i] = [\nu^{L_{i+1}} XG_{i+1}]$  and
- 3 3.  $\forall i \geq 1, L_i = (L_{i-1}, a_i)$  for some  $a_i \in \mathcal{L}$ .

4 The label-erasing function  $[\bullet]$  is easily lifted to sequents and to  $\mathcal{L}$ -preproofs. We then  
5 have the following property:

6 **Proposition 11.** *If  $\pi$  is an  $\mathcal{L}$ -proof, then  $[\pi]$  is a  $\mu\text{MALL}^\infty$  proof.*

7 *Proof.* First,  $[\pi]$  is a  $\mu\text{MALL}^\infty$  preproof because, as we explained, if we remove the  
8 labels, every rule in a  $\mathcal{L}$ -proof is a  $\mu\text{MALL}^\infty$  proof. Then, as we will show now, the  
9 definition of  $\mathcal{L}$ -proofs precisely reflects the thread criterion.

10 Suppose an infinite branch  $\gamma = (s_i)_{i \in \omega}$ ; we will show that  $[\pi]$  has a **valid infinite inner**  
11 **thread** in  $\gamma$  (Definition 39). There exists a sequence  $(\nu^{L_i} XG_i)_{i \in \omega}$  and a strictly increasing  
12 function  $\epsilon$  on natural numbers as in Definition 58.  $\forall i$ , let  $p_i$  be the position of the  $\nu^{L_i} XG_i$   
13 which is principal in  $s_{\epsilon(i)}$ .

14 Remark that any sequent contains only a finite number of atoms  $a \in \mathcal{L}$  and therefore  
15  $\{i \in \omega \mid a_i \text{ appears in the endsequent of } \pi\}$  is a finite set. Let  $i_0 := 1 + \max\{i \in \omega \mid$   
16  $a_i \text{ appears in the endsequent of } \pi\}$ , so that  $\forall i \geq i_0$ , the symbol  $a_i$ , contained by  $L_i$ , does  
17 not appear in the endsequent of  $\pi$ .

18 First consider any  $i \geq i_0$ , any  $j \in \omega$  and any position  $p$  of a subformula equal to  $\nu^{L_i} XG_i$   
19 in  $s_j$ . The extremity  $\mathfrak{t}(s_0, \dots, s_j)(p)$  of  $\mathfrak{T}(s_0, \dots, s_j)(p)$  is either in a cut formula or in  
20 the endsequent of  $\pi$ . In both cases, it does not contain  $a_i$ , which is present in  $L_i$ . Let  
21 us denote by  $g(p)$  the lowest point of  $\mathfrak{T}(s_0, \dots, s_j)(p)$  which contains  $a_i$ , it has to be a  
22 subformula of the active formula of the premise of a  $(\nu_b(a_i))$  inference. Similarly, let us  
23 denote by  $f(p)$  the highest point of  $\mathfrak{T}(s_0, \dots, s_j)(p)$  which does not contain  $a_i$ , it has to  
24 be the principal formula of this same inference  $(\nu_b(a_i))$ . And, of course, there is an edge  
25  $f(p) \rightarrow g(p)$  in  $\mathfrak{T}(s_0, \dots, s_j)(p)$ , induced by this inference.

26 From the last paragraph, we deduce that  $\forall i \geq i_0$ , there is a  $(\nu_b(a_i))$  inference in  $\gamma$ . We also  
27 deduce that the principal formula of this inference is  $\nu^{L_{i-1}} XG_{i-1}$ . Remark that this is  
28 the only  $(\nu_b(a_i))$  inference in  $\gamma$  because of the constraint of Definition 57. Let  $s_{\phi(i)}$  be the  
29 position of the conclusion of this inference, and let us denote by  $q_i$  the position of the  
30  $\nu^{L_{i-1}} XG_{i-1}$  which is the principal formula of this inference.

31 The union of the  $\mathfrak{T}(s_{\phi(i-1)}, \dots, s_{\phi(i)})(q_i)$ , for  $i \geq i_0 + 1$ , is an infinite path in  $G_{\text{thread}}^{\text{in}}$ ,  
32 which contains an infinity of principal formulas. This is a **valid inner thread** in  $[\pi]$ .  $\square$

## 4.2 Finite representations of circular $\mathcal{L}$ -proofs.

We now turn our attention to finite representations of (circular)  $\mathcal{L}$ -proofs. Immediately a difficulty occurs in comparison to non-labelled proofs: whereas an infinite non-labelled proof may happen to be regular, a valid  $\mathcal{L}$ -proof cannot be circular, because along every infinite branch, the sets of labels will grow endlessly. To form circular proofs with labels, some atoms must be forgotten when going bottom-up.

We introduce two more rules:  $\overleftarrow{\mathcal{Q}}(a)$  and (LWk). The first one allows to forget one atom, just before recreating it by means of a back edge to an already encountered  $\nu$ -rule. The other one allows to forget any atom that will not be used to validate the proof. It is used to synchronise the different labels in a sequent before travelling through a back edge.

- labelled back edge:  $\frac{}{\vdash \nu^{L,a} XA[X], \Gamma} \overleftarrow{\mathcal{Q}}(a)$  with the constraint that it must be the source of a back edge to the conclusion of a  $\frac{\vdash A[\nu^{L,a} XA[X]], \Gamma}{\vdash \nu^L XA[X], \Gamma} (\nu_b(a))$  below  $\overleftarrow{\mathcal{Q}}(a)$ .
- labelled weakening:  $\frac{\vdash \Gamma, B[\nu^L XA], \Delta}{\vdash \Gamma, B[\nu^{L,a} XA], \Delta} (\text{LWk})$

**Definition 59** ( $\mu\text{MALL}_{\text{lab}}^{\overleftarrow{\mathcal{Q}}}$ ).  $\mu\text{MALL}_{\text{lab}}^{\overleftarrow{\mathcal{Q}}}$  denotes the finite derivations of  $\mathcal{L}$ -sequents built from the rules in Definition 17 by replacing  $(\nu)$  with  $(\nu_b(a))$ ,  $\overleftarrow{\mathcal{Q}}(a)$ , (LWk), such that

1. the endsequent of the proof has empty labels and
2. for every two  $(\nu_b(a))$  and  $(\nu_b(b))$  occurring in the derivation,  $a \neq b$ .

Since  $\mu\text{MALL}_{\text{lab}}^{\overleftarrow{\mathcal{Q}}}$  proofs are finite, label-erasing gives rise to  $\mu\text{MALL}^\omega$  b.e.-tree preproofs:

**Proposition 12.** *The label-erasing function  $[\bullet]$  lifts to a translation from  $\mu\text{MALL}_{\text{lab}}^{\overleftarrow{\mathcal{Q}}}$  to  $\mu\text{MALL}^\omega$  b.e.-tree preproofs.*

*Proof.* Every rule of the labelled  $\mu\text{MALL}_{\text{lab}}^{\overleftarrow{\mathcal{Q}}}$  proof is sent by  $[\bullet]$  to a valid rule of unlabelled  $\mu\text{MALL}^\omega$ , except for the (LWk) rule, which can safely be removed:

$$\frac{\vdash \Gamma, B[\nu^L XA], \Delta}{\vdash \Gamma, B[\nu^{L,a} XA], \Delta} (\text{LWk}) \quad \text{becomes} \quad \frac{\vdash [\Gamma], [B][\nu X[A]], [\Delta]}{\vdash [\Gamma], [B][\nu X[A]], [\Delta]}$$

1

□

2 **Definition 60** ( $\mu\text{MALL}^{\curvearrowright}$ ). We denote by  $\mu\text{MALL}^{\curvearrowright}$  the set of b.e.-tree preproofs that  
 3 are obtained from  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  by label-erasing. This is a subset of all  $\mu\text{MALL}^{\omega}$  b.e.-tree  
 4 preproofs.

5 **Proposition 13** ( $\mu\text{MALL}^{\curvearrowright} \subseteq \mu\text{MALL}^{\omega}$ ). *Every preproof of  $\mu\text{MALL}^{\omega}$  that is the image  
 6 of a proof in  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  by label-erasing satisfies thread validity.*

7 We immediately give a proof sketch of this proposition; we then prove the two Lemmas 8  
 8 and 9 before giving a full proof of Proposition 13.

9 *Proof sketch of Proposition 13.* Consider a preproof  $\lceil \pi \rceil$  in  $\mu\text{MALL}^{\curvearrowright}$  which is the image  
 10 of an  $\mathcal{L}$ -proof  $\pi$  in  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  and an infinite branch  $b$  in  $\lceil \pi \rceil$ . We want to prove that  $b$   
 11 contains a **valid thread** (see Definition 21). Let  $b_0$  be the corresponding infinite  $\mathcal{L}$ -branch  
 12 in  $\pi$ . We remark (Lemma 8) that there is a sequent  $S_0$  which is the lowest target of back  
 13 edge crossed infinitely often by  $b_0$ . Besides,  $S_0$  is the conclusion of a  $(\nu_b(a))$  rule, which  
 14 unfolds some  $\nu^L XA$ .

15 We decompose  $b_0$ , with root  $r$ ;  $S_0$  conclusion of  $(\nu_b(a))$  and  $\nu^L XA$  at position  $p_0$  in  $S_0$ ; for  
 16 any  $i \geq 1$ ,  $S_i$  conclusion of a back edge  $(\varrho(a))$  with  $\nu^{L,a} XA$  at position  $p_i$  in  $S_i$ . Then we  
 17 notice that  $\mathfrak{T}(u_i)(p_0)$  (Definition 40) is a thread  $(S_0, p_0) \xrightarrow{*} (S_i, p_i)$  which is progressing,  
 18 as its source is the principal conclusion of the rule  $(\nu_b(a))$ . By concatenating the  $\mathfrak{T}(u_i)(p_0)$   
 19 and then erasing labels, we get a valid thread of  $b$  in  $\lceil \pi \rceil$ .

$$\begin{array}{c}
 \overline{S_i}^{(\varrho(a))} \\
 \uparrow u_i \quad \curvearrowright \\
 \overline{S_0}^{(\nu_b(a))} \\
 \uparrow u_0 \\
 r
 \end{array}$$

20

□

21 **Lemma 8.** *Let  $b$  be an infinite branch in a b.e.-tree preproof, i.e. an infinite ascending  
 22 path in its **branch graph**, starting from the root.*

1 Then there is an occurrence of sequent in this b.e.-tree preproof, i. e. a vertex  $s$  in its  
 2 *branch graph*, which is the lowest one infinitely appearing on  $b$ .

3 Moreover, this occurrence of sequent (this vertex) is the target of a back edge.

4 *Proof.* This is essentially a graph-theoretic property, which does not rely on the logical  
 5 content of the proof.

6 The crucial fact to notice is that, in general, in a tree, if  $S$  is a non empty, finite set  
 7 of vertices that is connected for the relation of comparability, that is if  $\forall v, v' \in S, v \leq$   
 8  $v'$  or  $v' \leq v$ , then  $S$  has a minimum. This is proved by induction on the cardinal of  $S$ .

9 Take then for  $S$  the set of vertices occurring infinitely often on the branch  $b$ . It is a  
 10 connected subgraph of  $G_{\text{branch}}$ . Because of that, it is connected for comparability in the  
 11 tree: if  $s_0 \rightarrow s_1$  in an edge in  $S$  then either it is induced by a logical inference, in which  
 12 case  $s_0$  is below  $s_1$ , or it is induced by a back edge, in which case  $s_1$  is below  $s_0$ .

13 We can then apply the previous remark to  $S$ , and we obtain a vertex  $s$ , which is the  
 14 lowest position of sequent infinitely appearing on  $b$ .

15 In particular, because  $s$  appears infinitely often on  $b$ , there must be another position  
 16  $s_0$  in  $S$  and an edge  $s_0 \rightarrow s$ . Because  $s$  is the lowest position in  $S$ , this edge must be  
 17 induced by a back edge. Hence  $s$  is a target of back edge.  $\square$

18 **Lemma 9** (Follow-up of labels). *If  $u$  is a path in the *branch graph* of a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof,*  
 19 *if  $u$  does not cross the rule  $(\nu_b(a))$ , and if  $p$  is a position in the target sequent of  $u$  (its top*  
 20 *sequent) that is labelled with  $a$ , then  $\mathfrak{t}(u)(p)$  (Definition 40) is defined and is a position*  
 21 *labelled with  $a$  in the source sequent of  $u$  (its bottom sequent).*

22 *Proof.* This is quite straightforward, by induction on the length of  $u$ , and by looking at  
 23 the first (or the last) rule crossed by  $u$ . We use notably the fact that, when the induced  
 24 thread  $\mathfrak{T}(u)(p)$  is followed top-down, the label  $a$  cannot be erased because we do not  
 25 cross  $(\text{Rec}(a))$  and the thread cannot reach a cut-formula because cut-formulas do not  
 26 contain labels.  $\square$

27 *Proof of Proposition 13.* Suppose  $\pi$  is a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof.

- 28 • Let  $\lceil \pi \rceil$  be its erasure.  $\lceil \pi \rceil$  is thus a  $\mu\text{MALL}^\omega$  b.e.-tree preproof.
- 29 • Suppose  $b$  an infinite branch of  $\lceil \pi \rceil$ , that is an infinite ascending path in the b.e.-tree  
 30  $\lceil \pi \rceil$ , starting from the root.

#### 4 A polynomial sub-criterion

- 1 • Let  $b_0$  be the corresponding infinite branch in  $\pi$ .
- 2 • By Lemma 8, let  $S_0$  be the occurrence of sequent in  $\pi$  which is the lowest back  
3 edge target infinitely often crossed by  $b_0$ . Being the target of some back edge(s),  $S_0$   
4 is the conclusion of a  $(\nu_b(a))$  rule, which unfolds some  $\nu XA[X]$ .

- Let us decompose  $b_0$  by making apparent each occurrence of  $S_0$ :

$$b_0 = r \xrightarrow[u_0]{*} S_0 \xrightarrow[u_1]{*} S_1 \xrightarrow[\text{be}]{*} S_0 \xrightarrow[u_2]{*} S_2 \xrightarrow[\text{be}]{*} S_0 \cdots$$

5 where  $r$  is the root of  $\pi$ , “be” denotes a back edge, the  $S_i$  ( $i \geq 1$ ) are the sources of  
6 the back edges to  $S_0$  and  $\forall i, u_i$  does not cross  $S_0$  except at its source.

- 7 • Let  $p_0$  be the position in  $S_0$  of the  $\nu XA[X]$  which is its principal formula.
- 8 • Remark that, because of the existence of back edges from every  $S_{i+1}$  to  $S_0$ , all  $S_i$   
9 are identical sequents, except for the fact that  $a$  does not appear in  $S_0$  while it  
10 appears at the only position  $p_0$  in  $S_{i+1}$ . In particular,  $p_0$  is also the position of a  
11  $\nu XA[X]$  in  $S_{i+1}$ .
- 12 • For  $i \geq 1$ ,  $\mathfrak{T}(u_i)(p_0)$  is a  $\nu$ -thread in  $u_i$ . Its target is the position  $p_0$  in  $S_i$ , which is  
13 the position of a  $\nu XA[X]$  labelled with  $a$ . Its source is a position of  $\nu XA[X]$  in  $S_0$ .  
14 We show that this source position is the same position  $p_0$  of the  $\nu XA[X]$  that is  
15 principal in  $S_0$ .

Let us decompose  $u_i$  as

$$u_i = S_0 \xrightarrow[e_0]{} S'_0 \xrightarrow[u'_i]{*}$$

16 where  $S'_0$  is the position of sequent immediately above  $S_0$ , that is the premise of  
17 the  $(\nu_b(a))$ , and  $e_0$  is the edge induced by the  $(\nu_b(a))$  rule.

18 By definition of  $u_i$ , the path  $u'_i$  does not cross  $(\nu_b(a))$ . By Lemma 9, the source  
19 of  $\mathfrak{T}(u'_i)(p_0)$  is a position labelled with  $a$ , in  $S'_0$ . Hence, this is a position  $p'_0$  of a  
20  $\nu XA[X]$  in the unfolding  $A[\nu XA[X]]$  caused by  $(\nu_b(a))$ . The source of  $\mathfrak{T}(u_i)(p_0)$  is  
21 then the image in  $S_0$  of this  $\nu XA[X]$  at  $p'_0$ , that is the principal formula  $\nu XA[X]$   
22 at position  $p_0$  in  $S_0$ .

- 23 •  $\mathfrak{T}(u_i)(p_0)$  is then a thread  $(S_0, p_0) \xrightarrow[\mathfrak{T}(u_1)(p_1)]{*} (S_1, p_0)$ . It is progressing, because its  
24 source is the principal conclusion of the rule  $(\nu_b(a))$ .

- By glueing the  $\mathfrak{T}(u_i)(p_0)$  together, we get an infinite **inner thread**

$$(S_0, p_0) \xrightarrow[\mathfrak{T}(u_1)(p_0)]{*} (S_1, p_0) \xrightarrow[\text{be}]{*} (S_0, p_0) \xrightarrow[\mathfrak{T}(u_2)(p_0)]{*} (S_2, p_0) \xrightarrow[\text{be}]{*} (S_0, p_0) \cdots$$

### 4.3 Two alternative characterizations of $\mu\text{MALL}^{\curvearrowright}$

This thread is valid because every  $\mathfrak{T}(u_i)(p_0)$  is progressing. And it is indeed a thread of

$$b_0 = r \xrightarrow[u_0]{*} S_0 \xrightarrow[u_1]{*} S_1 \xrightarrow[\text{be}]{\rightarrow} S_0 \xrightarrow[u_2]{*} S_2 \xrightarrow[\text{be}]{\rightarrow} S_0 \cdots$$

- Hence  $b_0$  is valid, which concludes this demonstration.

□

**Proposition 14.**  $\mu\text{MALL}$  proofs can be translated to  $\mu\text{MALL}^{\curvearrowright}$ .

*Proof.* The target of the usual translation  $\mu\text{MALL} \rightarrow \mu\text{MALL}^\omega$  [Doumane, 2017b] is included in  $\mu\text{MALL}^{\curvearrowright}$ . See key case of the translation on Figure 4.1. □

$$\frac{\frac{\vdash A[B], B^\perp \vdash B, \Gamma}{\vdash \nu X A[X], \Gamma} (\nu_{\text{inv}}^0)}{\vdash \nu^a X A[X], B^\perp} (\varrho(a)) \equiv \frac{\frac{\frac{\vdash A[\nu^a X A[X]], A[B]^\perp}{\vdash A[\nu^a X A[X]], B^\perp} (\nu_b(a)) \quad \vdash A[B], B^\perp}{\vdash \nu X A[X], B^\perp} (\text{cut})}{\vdash \nu X A[X], \Gamma} (\text{cut})} (\text{cut})$$

Figure 4.1. translation  $\mu\text{MALL} \rightarrow \mu\text{MALL}_{\text{lab}}^{\curvearrowright}$

*Remark 19.* Observe that a proof in  $\mu\text{MALL}^{\curvearrowright}$  is not, in general, the translation of a  $\mu\text{MALL}$  proof.

## 4.3 Two alternative characterizations of $\mu\text{MALL}^{\curvearrowright}$

In the two following sections, we give two characterizations of  $\mu\text{MALL}^{\curvearrowright}$ , through **validating sets** in Definition 64, and through a threading criterion over back edges in Definition 66. This allows us to give effective algorithms to decide whether a  $\mu\text{MALL}^\omega$  b.e.-tree preproof is in  $\mu\text{MALL}^{\curvearrowright}$  and, in case it is, to produce a labelling witnessing that fact.

**Definition 61.** Given a directed graph  $G = (V, E)$  and a set  $S \subseteq V$ , the set of vertices accessibles from  $S$  is denoted as  $S^\uparrow := \{v \in V \mid \exists s \in S, s \rightarrow^* v\}$ . Similarly  $S^\downarrow$  is the set of vertices from which  $S$  is accessible.

#### 4 A polynomial sub-criterion

1 Given a  $\mu\text{MALL}^\omega$  b.e.-tree preproof, we will define a graph  $G_\pi$  very similar to  $G_{\text{thread}}^{\text{in}}$ . In  
 2  $G_{\text{thread}}^{\text{in}}$ , we consider only positions of subformulas in closed sequents, and the positions  
 3 of subformulas in a source of back edge, which is an open sequent, are identified to those  
 4 in the target of that back edge. In contrast, in  $G_\pi$ , we consider the positions in sources  
 5 of back edges as distinct from the corresponding positions in their targets. The aim of  
 6  $G_\pi$  is precisely to see how we can connect those positions through paths in the graph.

7 **Definition 62** ( $G_\pi$ ). Given a  $\mu\text{MALL}^\omega$  b.e.-tree preproof  $\pi$ , we define  $G_\pi$  to be the  
 8 following graph:

- its *vertices* are all positions of  $\nu$ -formulas in all occurrences of sequents in  $\pi$ , plus the vertex  $\perp$ :

$$V_\pi := \left\{ \begin{array}{l} (i) \ v \text{ position of a sequent } \Gamma \text{ in } \pi \\ (v, i, p) \text{ such that } \\ (ii) \ i \text{ position of a formula } A \text{ in } \Gamma \\ (iii) \ p \text{ position of a } \nu\text{-subformula in } A \end{array} \right\} \uplus \{\perp\};$$

- for its *edges*: for every position  $p$  of a  $\nu$ -subformula, if  $q$  is its image in the sequent just below, as defined by the threading structure of the preproof, then there is an edge  $q \rightarrow p$  in  $G_\pi$ . Moreover, for every position  $p$  of a  $\nu$ -subformula of a cut formula, there is an edge  $\perp \rightarrow p$ .

13 We illustrate the construction of the edges of the graph defined in definition 62 with  
 14 the following examples in which we have indexed the apparent  $\nu$ -formulas by numbers  
 15 representing vertices of the graph:

$$\frac{\vdash \nu_1 XX, \nu_2 XX \quad \vdash \mathbf{1} \oplus \nu_3 XX}{\vdash \nu_4 XX \otimes (\mathbf{1} \oplus \nu_5 XX), \nu_6 XX}^{(\otimes)}$$

16 induces edges  $4 \rightarrow 1, 6 \rightarrow 2$  and  $5 \rightarrow 3$ ,

$$\frac{\vdash \nu_1 XX, (\mathbf{1} \oplus \nu_2 XX), \nu_3 XX}{\vdash \nu_4 XX \wp (\mathbf{1} \oplus \nu_5 XX), \nu_6 XX}^{(\wp)}$$

17 induces edges  $4 \rightarrow 1, 5 \rightarrow 2$  and  $6 \rightarrow 3$ , and

$$\frac{\vdash \nu_4 Y \nu_5 X (\nu_6 Y X \otimes X) \otimes \nu_7 X (\nu_8 Y X \otimes X), \nu_9 XX}{\vdash \nu_1 X (\nu_2 Y X \otimes X), \nu_3 XX}^{(\nu)}$$

### 4.3 Two alternative characterizations of $\mu\text{MALL}^{\curvearrowright}$

1 induces edges  $2 \rightarrow 4, 2 \rightarrow 6, 2 \rightarrow 8, 1 \rightarrow 5, 1 \rightarrow 7$  and  $3 \rightarrow 9$ . Moreover, if the conclusion  
 2 of this last rule is the target of a back edge whose source is  $\vdash \nu_{10}X(\nu_{11}YX \otimes X), \nu_{12}XX$   
 3 then this back edge also induces edges  $10 \rightarrow 1, 11 \rightarrow 2, 12 \rightarrow 3$ .

4 In the case of a cut formula, the formula has no corresponding formula in the conclusion  
 5 sequent and in this case it induces an ingoing edge from the extra vertex  $\perp$ :

$$\frac{\vdash \nu_2XX \quad \vdash \mu XX, \nu_3XX}{\vdash \nu_1XX} \text{ (cut)}$$

6 induces edges  $\perp \rightarrow 2$  and  $1 \rightarrow 3$ .

7 *Remark 20.* Note that a vertex of the above graph may therefore have

- 8 • many ingoing edges: in case it is in a sequent which is the target of some back edges;
- 9 • no ingoing edges: essentially the extra vertex  $\perp$  and the  $\nu$ -formulas in the conclusion  
 10 of the proof if it is not a target of back edge;
- 11 • many outgoing edges: see a  $\nu$ -rule with  $\nu X(X \otimes X)$  or  $\nu X(X \wp X)$  as principal  
 12 conclusion;
- 13 • no outgoing edges: see the  $\nu XX$  in the  $\mathbf{1} \oplus \nu XX$  principal conclusion of a  $(\oplus_1)$ -rule.

14 In the following definition, we define, for each  $(\nu)$  rule  $(r)$  of a b.e.-tree preproof  $\pi$ , a  
 15 subgraph of  $G_\pi$  (Definition 62) called  $G_r$ . The aim of this subgraph is to contain all paths  
 16 in  $G_\pi$  that uses only sequents and back edges above  $(r)$ . We also define two subsets of the  
 17 vertices of  $G_r$ , called  $S_r$  and  $T_r$ . As we will show in Proposition 16,  $\pi$  is *not* in  $\mu\text{MALL}^{\curvearrowright}$   
 18 iff. there is a  $(\nu)$  rule  $(r)$  such that there is a path in  $G_r$  with source in  $S_r$  and target in  
 19  $T_r$ . This gives us an effective way to check membership in  $\mu\text{MALL}^{\curvearrowright}$  (Proposition 17).

20 **Definition 63**  $(G_r, S_r, T_r)$ . Let  $\pi$  be a  $\mu\text{MALL}^\omega$  b.e.-tree preproof and  $(r)$  an occurrence  
 21 of a  $(\nu)$ -rule. We define a subgraph of  $G_\pi$ , denoted by  $G_r = (V_r, E_r)$ , and two subsets of  
 22  $V_r$ , denoted by  $T_r$  and  $S_r$ :

- 23 • the vertices  $v \in V_r$  are the extra vertex  $\perp$  plus all positions that are in the conclusion  
 24 of  $(r)$  and in all above sequents, that is all sequents accessible from the conclusion  
 25 of  $(r)$ , in the sense of Definition 61;
- 26 • the edges  $e \in E_r$  are all edges of  $G_\pi$  between those vertices, minus the edges of  $G_\pi$   
 27 that are induced by the back edges of  $\pi$  targetting the conclusion of  $(r)$ , if there

#### 4 A polynomial sub-criterion

1 are some.

- 2 •  $T_r \subseteq V_r$  is the set of all positions of the principal formulas of the sources sequents
- 3 of the back edges targetting the conclusion of  $(r)$ ;
- 4 •  $S_r \subseteq V_r$  is the set of all positions of all subformulas of the conclusion of  $(r)$  except
- 5 for the position of its principal formula, plus the extra vertex  $\perp$ .

6 **Definition 64.** Let  $(r)$  be an occurrence of a  $(\nu)$  rule in a  $\mu\text{MALL}^\omega$  b.e.-tree preproof  $\pi$ .  
 7 A *validating set* for  $(r)$  is a set  $L \subseteq V_r$  such that  $L = L\downarrow$  and  $T_r \subseteq L \subseteq (V_r \setminus S_r)$ .

8 **Proposition 15.** Let  $(r)$  be an occurrence of a  $(\nu)$  rule of a  $\mu\text{MALL}^\omega$  b.e.-tree preproof  
 9  $\pi$ . The following are equivalent:

- 10 1. There exists a validating set for  $(r)$
- 11 2.  $T_r$  is not accessible from  $S_r$  in  $G_r$
- 12 3.  $T_r\downarrow \subseteq V_r \setminus (S_r\uparrow)$ .

13 And if they are true,  $T_r\downarrow$  is the smallest validating set of  $(r)$  and  $V_r \setminus (S_r\uparrow)$  is the biggest  
 14 one.

15 *Proof.* We first prove that

$$16 \quad \text{If } L \text{ is a validating set, then } T_r\downarrow \subseteq L \subseteq V \setminus (S_r\uparrow) \quad (4.1)$$

If  $L$  is a validating set, then on the one hand we have

$$T_r \subseteq L$$

hence

$$T_r\downarrow \subseteq L\downarrow = L.$$

And on the other hand

$$L \subseteq V \setminus S_r$$

hence

$$V \setminus L \supseteq S_r$$

and, because the complement of a downward-closed set is upward-closed,

$$V \setminus L = (V \setminus L)\uparrow \supseteq S_r\uparrow$$

hence

$$L \subseteq V \setminus (S_r\uparrow).$$

1 Now the implication (1)  $\Rightarrow$  (3) is an immediate consequence of (4.1).

For (3)  $\Rightarrow$  (2): if

$$T_r\downarrow \subseteq V \setminus (S_r\uparrow)$$

then

$$T_r \subseteq T_r\downarrow \subseteq V \setminus (S_r\uparrow)$$

hence

$$S_r\uparrow \cap T_r \subseteq \emptyset$$

2 hence  $T_r$  is not accessible from  $S_r$ .

Finally (2)  $\Rightarrow$  (1): if  $T_r$  is not accessible from  $S_r$  then

$$S_r \cap T_r\downarrow \subseteq \emptyset$$

hence

$$T_r \subseteq T_r\downarrow \subseteq V \setminus S_r$$

3 hence  $T_r\downarrow$  is a validating set.

4 Regarding the second part of the proposition:

- if (3) is true then

$$T_r \subseteq T_r\downarrow \subseteq V_r \setminus (S_r\uparrow) \subseteq V_r \setminus S_r$$

5 hence  $T_r\downarrow$  and  $V_r \setminus (S_r\uparrow)$  are validating sets

- if  $L$  is any other validating set, then by (4.1),  $T_r\downarrow \subseteq L \subseteq V \setminus (S_r\uparrow)$ .

7 □

#### 4 A polynomial sub-criterion

1 The following proposition gives an alternative criterion for  $\mu\text{MALL}^{\curvearrowright}$ .

2 **Proposition 16.** *A finite representation  $\pi$  of a  $\mu\text{MALL}^{\omega}$  preproof is the erasing of a*  
 3  *$\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof iff. every occurrence of a  $(\nu)$  rule of  $\pi$  has a validating set.*

*Proof.* ( $\Leftarrow$ ) Let us assume that every  $(\nu)$  rule of  $\pi$  has a validating set. There is a finite number of  $(\nu)$  rules in  $\pi$ ; we label them with distinct variables  $a_1, \dots, a_n$ , in such a way that if the  $(\nu)$  rule labelled by  $a_i$  is below the rule labelled by  $a_j$  in the representation then  $i \leq j$ . We denote by  $L_i$  a validating set for  $(\nu_{b(a_i)})$ . We then do the following for each  $i$ , going from 1 to  $n$ : for each occurrence of  $\nu$ -formula  $\nu^V XA$  that is at a position belonging to  $L_i$ , add the variable  $a_i$  to  $V$ , that is replace this occurrence of  $\nu^V XA$  with  $\nu^{V, a_i} XA$ . By doing this it may happen that we break the validity of some rules of the representation: because  $L_i$ , although downward closed, is in general not upward closed, so we may end with a situation in which this

$$\frac{\vdash A, C[\nu^V X D] \quad \vdash A, C[\nu^V X D]}{\vdash A \& B, C[\nu^V X D]} \&$$

becomes

$$\frac{\vdash A, C[\nu^{V, a} X D] \quad \vdash B, C[\nu^V X D]}{\vdash A \& B, C[\nu^{V, a} X D]} \&$$

which is not anymore a valid rule. We then patch this by adding as many (LWk) rules as needed on the premises:

$$\frac{\vdash A, C[\nu^{V, a} X D] \quad \frac{\vdash B, C[\nu^V X D]}{\vdash B, C[\nu^{V, a} X D]} \text{(LWk)}}{\vdash A \& B, C[\nu^{V, a} X D]} \&$$

4 Similarly it may happen that the source of a back edge gets a bigger labelling than the  
 5 target of this back edge; we patch this by adding (LWk) rules under the source sequent of  
 6 the back edge. When this operation has been done for every  $i$ , from 1 to  $n$ , we obtain a  
 7 validly labelled proof of  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$ .

8 ( $\Rightarrow$ ) Conversely, let  $\pi_0$  be a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  representation such that  $\pi = [\pi_0]$ . Up to  
 9 renaming, we can assume that all  $(\nu_b)$  rules of  $\pi_0$  are labelled with distinct variables. For  
 10 every occurrence of a  $(\nu)$  rule in  $\pi$ , consider the corresponding  $(\nu_b(a))$  rule in  $\pi_0$  and let  
 11  $L_a$  be the set of all occurrences of  $\nu$ -formulas in  $\pi_0$  that carry the variable  $a$  in their  
 12 labelling. The constraints on the labelling of  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof precisely say that  $L_a$  is a  
 13 validating set for the considered occurrence of  $(\nu_b)$  in  $\pi$ .  $\square$

14 **Proposition 17.**

- 1 • Checking validity of a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  preproof is decidable.
- 2 • Membership in  $\mu\text{MALL}^{\curvearrowright}$  can be decided in a time quadratic in the size of the
- 3 (circular) preproof.

4 *Proof.*

- 5 • For the first point, it is enough to check the validity of each inference of the proof.
- 6 • For the second point, first check that each back edge targets a  $(\nu)$  rule, then use the
- 7 characterization of  $\mu\text{MALL}^{\curvearrowright}$  by validating sets (Proposition 16): for each  $(\nu)$  rule
- 8  $(r)$ , build graph  $G_r$  and sets  $S_r$  and  $T_r$  and checks whether  $T_r$  is accessible from
- 9  $S_r$  in  $G_r$ . Such a test of accessibility in a graph can be done in linear time. As we
- 10 have to do that for each  $(\nu)$  rule, in total it may take a time quadratic in the size of
- 11 the preproof, but no more than that.

12  $\square$

13 **Definition 65.** Let  $\pi$  be a  $\mu\text{MALL}^{\omega}$  b.e.-tree preproof and  $e$  a back edge in  $\pi$ , from a

14 position of sequent  $s_1$  to a position of sequent  $s_0$ . Let  $S$  be the set of all positions of

15 sequents that are strictly above  $s_0$ . We say that this back edge is *strongly valid* under

16 two conditions:

- 17 1. Its target  $s_0$  has to be the conclusion of a  $(\nu)$  rule.
- 18 2. Let us denote by  $p$  the position in  $s_0$  of the  $\nu X A$  which is the principal formula of
- 19 that  $(\nu)$  rule. This is also a position of the same formula in  $s_1$ . The second condition
- 20 is that for every path  $u$  in  $G_{\text{branch}}$ , going from  $s_0$  to  $s_1$ , using only positions of
- 21 sequents in  $S$ ,  $t(u)(p)$  is, again, the position  $p$  in  $s_0$ .

22 **Definition 66.** A  $\mu\text{MALL}^{\omega}$  b.e.-tree preproof is *strongly valid* if all its back edges are

23 *strongly valid*.

24 **Proposition 18.** A  $\mu\text{MALL}^{\omega}$  b.e.-tree preproof  $\pi$  is strongly valid iff. it is the erasing

25 of a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof.

26 *Proof.* We will use our first characterization by validating sets (Proposition 16) and

27 prove that a  $\mu\text{MALL}^{\omega}$  b.e.-tree preproof  $\pi$  is strongly valid iff. every  $\nu$ -rule of  $\pi$  has a

28 validating set.

1 ( $\Leftarrow$ ) Let us assume that  $\pi$  has a validating set. Let us consider one occurrence  
2  $\frac{\vdash A[\nu XA[X]], \Gamma}{\vdash \nu XA[X], \Gamma}$  ( $\nu$ ) of a  $\nu$ -rule in  $\pi$  and a path  $u$  in the subgraph above this  $\nu$ -rule,  
3 going down, from the source of a back edge targetting this  $\nu$ -rule, to the  $\nu$ -rule itself,  
4 ending by this  $\nu$ -rule.  $u$  has then premise and conclusion equal to  $\vdash \nu XA[X], \Gamma$ .

5 Let us denote by  $L$  a validating set of this ( $\nu$ ) rule occurrence, and let us denote by  $t$   
6 the maximal thread going down in  $u$  starting from the main  $\nu XA[X]$  in its premise.  
7 This occurrence of  $\nu XA[X]$  is in  $L$ , because  $L$  is a validating set. Then, because  $L$  is  
8 downward closed, all vertices of  $t$  are in  $L$ . Therefore the lowest vertex of  $t$ , which is  
9 a position in the  $\vdash \nu XA[X], \Gamma$  conclusion of the considered  $\nu$ -rule, or  $\perp$ , is also in  $L$ .  
10 But in this last sequent occurrence, the only position that is in  $L$  is the one of the main  
11  $\nu XA[X]$ , which is consequently the end point of  $t$ .

12 ( $\Rightarrow$ ) Conversely, let us consider an occurrence ( $r$ ) of a ( $\nu$ ) rule in  $\pi$ , whose conclusion has  
13 the form  $\vdash \nu XA[X], \Gamma$ , and let us assume that it has no validating set. By Proposition 15,  
14 there is a path  $v$  in  $G_r$  with source  $s \in S_r$  and target  $t \in T_r$ . Let  $u$  be the corresponding  
15 path in  $G_{\text{branch}}$ . By definition of  $T_r$  (Definition 63),  $t$  is in the source of a back edge  $e$   
16 targetting the conclusion of ( $r$ ). More precisely, the position of  $t$  in the sequent at  $s$  is  
17 the same as the position of the principal formula of ( $r$ ) in the conclusion of ( $r$ ). Because  
18 the definition of  $G_r$  follows the threading structure of the preproof, the source  $s$  of  $v$  is  
19  $\mathbf{t}(u)(t)$ . By definition of  $S_r$ , this position  $s = \mathbf{t}(u)(t)$  is either in a cut formula or in the  
20 conclusion of ( $r$ ), at a position different from the position of its principal formula. This  
21 shows that ( $r$ ) is not strongly valid.  $\square$

22 **Corollary 3.** *The fragments defined in Definitions 60 and 66 coincide.*

## 23 4.4 Conclusion

24 As this chapter and the next one share a lot of ideas, discussion of related works for this  
25 chapter has been placed in Section 5.5.

26 This chapter introduced the loop criterion for circular representations of infinite preproofs  
27 of  $\mu\text{MALL}^\omega$ . This defines a fragment of the circular representations of  $\mu\text{MALL}^\omega$  which  
28 we called  $\mu\text{MALL}^\omega$ .

29 We also defined a new proof system called  $\mu\text{MALL}_{\text{lab}}^\omega$ . The proofs of  $\mu\text{MALL}_{\text{lab}}^\omega$  are  
30 labelings of circular representations of preproofs, valid for the loop criterion. While  
31 circular representation of preproofs make use of a global validity criterion such as the

1 thread criterion or the loop criterion, the validity of a proof in  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  is a purely  
 2 local property, which can be checked in linear time.

3 We proved in Propositions 16 and 18 that the b.e.-tree preproof obtained by erasing  
 4 the labels of a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof is valid for the loop criterion and, conversely, that any  
 5 circular representation that is valid for the loop criterion can be turned into a proof of  
 6  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  by labelling some of its formulas.

7 We made that last property explicit by providing a procedure to check a circular represen-  
 8 tation with respect to the loop criterion and to turn it into a labelled proof of  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$   
 9 when it is valid. This algorithm runs in quadratic time.

10 Now that we have identified a fragment of  $\mu\text{MALL}^{\omega}$  which proves at least all sequents  
 11 provables in  $\mu\text{MALL}$ , it is natural to wonder whether it proves strictly more than  $\mu\text{MALL}$ .  
 12 This question is the subject of the next and final chapter of this thesis.



# 5 Finitization

The first systems designed to handle least and greatest fixed points connectives were finitary systems. Infinitary and circular systems were then proposed as an alternative. In each of these circular extensions of finitary systems, there is a canonical translation from finitary proofs to circular proofs. This implies in particular that any sequent that is provable in the finitary system is also provable in the circular system. The converse is known as the Brotherston-Simpson conjecture and was first stated by Brotherston and Simpson [2007, 2011] in the context of Martin-Löf’s system of inductive definitions for first-order classical logic. It is also called the problem of finitization: can we turn an infinitary proof into a finitary one, and how? In the context of first-order classical logic with inductive definitions, it was proved by Berardi and Tatsuta [2017b, 2019] that the conjecture is false in general, but it was shown by Simpson [2017], Berardi and Tatsuta [2017a, 2018] that it becomes true as soon as you add Peano (or even Heyting) arithmetic to the circular and finitary systems. But the question is still open in the case of  $\mu\text{MALL}$  and  $\mu\text{MALL}^\omega$ , which is a quite different system, more in the family of  $\mu$ -calculi. In that setting, on one hand we do not have first-order predicates and the possibility to encode Peano arithmetic, on the other hand we have access to all possible inductive and coinductive definitions of formulas, including interleaving of least and greatest fixed points.

Here the question of finitization is: if a conclusion is provable by a circular proof, valid for the thread criterion, that is a proof in  $\mu\text{MALL}^\omega$ , is it already provable in the finitary system  $\mu\text{MALL}$ ?

This question is, as of today, an open problem.

Now, in the previous chapter, we provided a new criterion, the loop criterion, which defines a new logic  $\mu\text{MALL}^\curvearrowright$ , which has two properties:

- the canonical translation of any finitary proof is a circular representation which is valid for the loop criterion, so any conclusion which is provable in  $\mu\text{MALL}$  is also provable in  $\mu\text{MALL}^\curvearrowright$
- if the circular representation of a preproof is valid for the loop criterion, then it is also valid for the thread criterion, so any conclusion that is provable in  $\mu\text{MALL}^\curvearrowright$  is

1 also provable in  $\mu\text{MALL}^\omega$ .

2 This allows us to split the problem of finitization into two subproblems:

- 3 • If a conclusion is provable in  $\mu\text{MALL}^\omega$ , is it already provable in  $\mu\text{MALL}^\curvearrowright$ ?
- 4 • If a conclusion is provable in  $\mu\text{MALL}^\curvearrowright$ , is it already provable in  $\mu\text{MALL}$ ?

5 A positive answer to both questions would mean that finitization is possible and the  
6 Brotherston-Simpson conjecture is true for  $\mu\text{MALL}$  and  $\mu\text{MALL}^\omega$ .

7 In this chapter, we answer positively the second question: we prove that any conclusion  
8 provable in  $\mu\text{MALL}^\curvearrowright$  is already provable in  $\mu\text{MALL}$ . We do so by providing an explicit  
9 method which takes as input a circular representation of preproof, valid for the loop  
10 criterion, and output a standard, finitary  $\mu\text{MALL}$  proof.

## 11 5.1 On Brotherston-Simpson's conjecture: finitizing 12 circular proofs

13 The aim of this section is to prove a converse of Proposition 14: *Every provable sequent*  
14 *of  $\mu\text{MALL}^\curvearrowright$  is provable in  $\mu\text{MALL}$ .*

15 This will be proved by defining a translation from every  $\mu\text{MALL}^\curvearrowright$  proof of a sequent  $\vdash \Gamma$   
16 containing no label variable into a  $\mu\text{MALL}$  proof of the same sequent.

17 Let us consider a  $\mu\text{MALL}^\curvearrowright$  proof  $\pi$ . Up to renaming of bound variables, we can assume  
18 that all  $(\nu_b)$  rules are labelled by distinct labels. For every two labels  $a$  and  $b$  occurring in  
19  $\pi$ , we say that  $a \leq b$  whenever  $(\nu_b(a))$  is under  $(\nu_b(b))$ . This order is well-founded because  
20 finite.

21 **Definition 67** (Context associated to a labelling atom). For every rule  $\frac{\vdash A[\nu^{V,a}X.A], \Gamma}{\vdash \nu^V X.A, \Gamma} (\nu_b(a))$   
22 we define  $\Gamma_{(a)}$  to be  $\Gamma$ .

23 We now define

- 24 1. for each atom  $a$  a sequent  $\Gamma_a$  formed of non-labelled formulas;

- 1     2. for each formula  $A$  (with labels) occurring in the proof, a formula  $\llbracket A \rrbracket$  without  
 2       labels:

3 **Definition 68** (Translation of labelled formulas to unlabelled formulas). We define by  
 4 mutual induction:

- 5     1.  $\Gamma_a := \llbracket \Gamma_{((a))} \rrbracket$ .
2.  $H_\emptyset[F] := F$  and  $H_{V,a}[F] := \otimes \Gamma_a^\perp \oplus H_V[F]$ . In other words,

$$H_V[F] \simeq \left( \bigoplus_{a \in V} \otimes \Gamma_a^\perp \right) \oplus F$$

- 6     3. By induction on formula  $A$ ,  $\llbracket A \rrbracket$  is:

- 7       •  $\llbracket \nu^V X.A \rrbracket := \nu X H_V[\llbracket A \rrbracket]$
- 8       • it is homomorphic on other connectives:  $\llbracket X \rrbracket := X$ ,  $\llbracket \mathbf{1} \rrbracket := \mathbf{1}$ ,  $\llbracket \mu X A \rrbracket := \mu X \llbracket A \rrbracket$ ,  
 9        $\llbracket A \otimes B \rrbracket := \llbracket A \rrbracket \otimes \llbracket B \rrbracket$ , *etc.*

- 10    4.  $\llbracket \cdot \rrbracket$  is lifted from formulas to sequences of formulas, pointwise.

11 This is well-founded because, since any two distinct  $\nu_b$  rules wear distinct variables,  
 12 the only  $\Gamma_b$  that are needed in the computation of  $\Gamma_a$  are those with  $b < a$ . Note that  
 13  $\llbracket A \rrbracket = A$  as soon as  $A$  has no label variable. We can now state and prove the finitization  
 14 theorem:

15 **Theorem 5.** *Every provable sequent of  $\mu\text{MALL}^{\curvearrowright}$  is provable in  $\mu\text{MALL}$ .*

16 *Proof.* Let  $\pi$  be a  $\mu\text{MALL}_{\text{lab}}^{\curvearrowright}$  proof and replace, everywhere, each formula  $A$  by  $\llbracket A \rrbracket$ . All  
 17 rules in this (almost) new derivation are now valid instances of  $\mu\text{MALL}$  rules, except  
 18 for  $(\nu_b)$ ,  $(\text{LWk})$  and  $(\curvearrowright)$  rules. Actually, images of these rules by sequent translation  $\llbracket \cdot \rrbracket$  are  
 19 derivable in  $\mu\text{MALL}$  as shown in fig. 5.1 (a), (b) and (c) for  $(\curvearrowright)$ ,  $(\text{LWk})$  and  $(\nu_b)$ , respectively.

20 Replacing each instance of a  $(\nu_b)$ ,  $(\text{LWk})$  or  $(\curvearrowright)$  rule in  $\pi$  by its derived version, we get a  
 21 fully valid proof of  $\mu\text{MALL}$ . If the conclusion of the original  $\mu\text{MALL}^{\curvearrowright}$  proof was  $\vdash \Gamma$  then  
 22 what we get is a proof in  $\mu\text{MALL}$  of  $\vdash \llbracket \Gamma \rrbracket$ , *i. e.* the conclusion of the original  $\mu\text{MALL}^{\curvearrowright}$   
 23 proof, if  $\Gamma$  contains no label variable.  $\square$

$$\begin{array}{c}
 \frac{\frac{\frac{\vdash \llbracket A[\nu^V X.A] \rrbracket, \Gamma}{\vdash H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, \Gamma}}{\vdash \llbracket \nu^V X.A \rrbracket, \Gamma} (\oplus_1)^{|V|}}{\vdash \llbracket \nu^V X.A \rrbracket, \Gamma} (\nu)}{(a)}
 \end{array}
 \qquad
 \frac{\frac{\frac{\frac{\frac{\frac{\vdash \otimes \Gamma_a^\perp, \Gamma_a}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \Gamma_a}}{\vdash \llbracket \nu^{V,a} X.A \rrbracket, \Gamma_a} (\nu)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \Gamma_a} (\oplus_0)}{\vdash \otimes \Gamma_a^\perp, \Gamma_a} (\otimes), (\text{id})}}{\vdash H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket} (\text{id})}}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket} (\oplus_1)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\mu)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\nu_{\text{inv}}^0)}{\vdash \llbracket \nu^{V,a} X.A \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \llbracket B[\nu^V X.A] \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \Gamma} (\text{cut})}
 \end{array}
 \qquad
 \frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\vdash \llbracket B[\nu^V X.A] \rrbracket, \Gamma}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \Gamma} :=}}{\vdash H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket} (\text{id})}}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket} (\oplus_1)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\mu)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^V X.A] \rrbracket \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\nu_{\text{inv}}^0)}{\vdash \llbracket \nu^{V,a} X.A \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \llbracket B[\nu^V X.A] \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \Gamma} (\text{cut})}
 \end{array}
 \qquad
 \frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\vdash \llbracket A[\nu^{V,a} X.A] \rrbracket, \Gamma_a}{\vdash \llbracket \nu^V X.A \rrbracket, \Gamma_a} \llbracket \nu_b(a) \rrbracket :=}}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \Gamma_a} (\oplus_1)^{|V|}}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \wp \Gamma_a} (\wp)}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket} (\text{id})}}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket} (\&)}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \llbracket \nu^{V,a} X.A \rrbracket^\perp} (\mu)}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \llbracket \nu^{V,a} X.A \rrbracket^\perp} (\nu)}{\vdash \llbracket \nu^V X.A \rrbracket, \Gamma_a} (\nu_{\text{inv}})}
 \end{array}
 \qquad
 \frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{\vdash \otimes \Gamma_a^\perp, \Gamma_a}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \Gamma_a} (\oplus_0)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \Gamma_a} (\nu)}{\vdash \otimes \Gamma_a^\perp, \Gamma_a} (\otimes), (\text{id})}}{\vdash H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket} (\text{id})}}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, H_V \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket} (\oplus_1)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \llbracket \nu^{V,a} X.A \rrbracket^\perp} (\mu)}{\vdash H_{V,a} \llbracket \llbracket A[\nu^{V,a} X.A] \rrbracket \rrbracket, \llbracket \nu^{V,a} X.A \rrbracket^\perp} (\nu_{\text{inv}}^0)}{\vdash \llbracket \nu^{V,a} X.A \rrbracket, \llbracket \nu^V X.A \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \llbracket B[\nu^V X.A] \rrbracket^\perp} (\llbracket B \rrbracket)}{\vdash \llbracket B[\nu^{V,a} X.A] \rrbracket, \Gamma} (\text{cut})}
 \end{array}$$

Figure 5.1. Derivability of (a)  $\llbracket (\nu_\uparrow) \rrbracket$  rule ; (b)  $\llbracket (\nu_\downarrow) \rrbracket$  rule ; (c)  $\llbracket (\text{LWk}) \rrbracket$  rule and (d)  $\llbracket (\nu_b) \rrbracket$  rule .





1     2. in our previous setting, back edges must target  $(\nu)$  rules, which is not the case here.

2 Both difficulties have, in fact, the same origin, namely that in our previous setting the  
 3  $(\nu)$  rule has two roles: being the target of a back edge and ensuring thread progression.  
 4 Both difficulties also have the same solution: dissociating these two roles. We therefore  
 5 introduce, in Definition 69, a new rule (*Rec*), whose only effect is to allow its premise to  
 6 be the target of a back edge, and to introduce a new label. Since (*Rec*) is disentangled  
 7 from greatest fixed point unfolding, the labelling must account for the progression of a  
 8 thread. That is why every atomic label is now given in one of two modes: a passive mode  
 9  $(a-)$  and an active one  $(a+)$ . Only an unfolding by a  $(\nu)$  can turn a  $-$  into a  $+$ .

10 Let us now turn back to our introductory example:  $\pi_\infty$  (Figure 5.2). For that example,  
 11 separating the introduction of back edges and the coinductive progress is not enough  
 12 to be able to label  $\pi_\infty$  in a way that certify its thread validity. Indeed, since targets of  
 13 back edges do not require to unfold a  $\nu$ , there is *a priori* no reason to require that the  
 14 sequents contain some  $\nu$ -formula. While this is slightly hidden in the interleaving example  
 15 of Figure 5.3,  $\pi_\infty$  gives a clear example of this phenomenon and suggests that the (*Rec*)  
 16 inference should have the ability to add labels *deeply* in the sequent, that is not only on  
 17 the topmost  $\nu$  fixed-points, but also to greatest fixed points occurring under some other  
 18 connectives. The same remark applies to the back edge rule since its conclusion sequents  
 19 have the same structure as those of (*Rec*).

20 Driven by these observations, we now define a new labelling of circular preproofs and  
 21 prove its correctness with respect to thread-validity.

**Definition 69** (Extended labelling). Labelled formulas are built on the same grammar as in Definition 56, except that labels are lists of *signed variables*, that is of pairs of a variable and a symbol in  $\{+, -\}$ . Derivations are built with  $\mu$ MALL inferences plus the following rules:

$$\frac{\vdash \nu^L X A, \Gamma}{\vdash \nu^{L, a-} X A, \Gamma} (\text{LWk}(a-)) \quad \frac{\vdash \nu^{L, a-, L'} X A, \Gamma}{\vdash \nu^{L, a+, L'} X A, \Gamma} (\text{LWk}(a+)) \quad \frac{\vdash A[\nu^{a_1+, \dots, a_n+} X A], \Gamma}{\vdash \nu^{a_1-, \dots, a_n-} X A, \Gamma} (\nu)$$

$$\frac{\vdash \Gamma[\nu^{L, a-} X A]}{\vdash \Gamma[\nu^L X A]} (\text{Rec}(a)) \quad \frac{}{\vdash \Gamma[\nu^{L, a+} X A]} (\text{Q}(a))$$

22 and the constraints that:

- 23     • a cut-formula cannot contain a non-empty label;
- 24     • all (*Rec*) rules must wear distinct variables;
- 25     • every (*Rec*(*a*)) rule must have at least one occurrence of “*a-*” in its premise;

## 5 Finitization

- 1 • each  $\frac{\vdash \Gamma[\nu^{L,a+} XA]}{\vdash \Gamma[\nu^{L,a-} XA]} \overset{(a)}{\curvearrowright}$  rule is connected to the premise of a  $\frac{\vdash \Gamma[\nu^{L,a-} XA]}{\vdash \Gamma[\nu^L XA]} \overset{(Rec(a))}{\curvearrowright}$   
 2 via a back edge. This implies in particular that this  $\overset{(a)}{\curvearrowright}$  must be above this  $\overset{(Rec(a))}{\curvearrowright}$   
 3 and that the premise of this  $\overset{(Rec(a))}{\curvearrowright}$  must be the same sequent as the conclusion  
 4 of this  $\overset{(a)}{\curvearrowright}$  except for the change of sign of  $a$ , at every of its occurrences in the  
 5 sequent.

6 With the extended labelling of Definition 69, we have the following.

7 **Proposition 19** (Soundness of extended labelling). *If  $\pi$  is an extended labelled circular*  
 8 *representation then  $[\pi]$  is a valid  $\mu\text{MALL}^\omega$  b.e.-tree proof.*

9 *Proof.* First remark that Lemma 9, as it is stated on p. 87, still holds for this extended  
 10 labelling. The proof is the same, keeping in mind to replace every mention of  $(\nu_b(a))$  with  
 11  $(Rec(a))$ . As for Proposition 13, the proof of this proposition crucially relies on it.

12 Suppose  $\pi$  is a labelled circular representation. Let  $[\pi]$  be its erasure.  $[\pi]$  is thus a  
 13 circular representation of a  $\mu\text{MALL}^\omega$  preproof. Suppose  $b$  an infinite branch of  $[\pi]$ , that  
 14 is an infinite ascending path in the b.e.-tree  $[\pi]$ , starting from the root. Let  $b_0$  be the  
 15 corresponding infinite branch in  $\pi$ . Let  $S_0$  be the occurrence of the sequent in  $\pi$  which is  
 16 the lowest back edge target infinitely often crossed by  $b_0$ . Being the target of some back  
 17 edge(s),  $S_0$  is the premise of a  $\overset{(Rec(a))}{\curvearrowright}$  rule, for some variable  $a$ .

18 This implies that  $b_0$  is of the form  $b_0 = r \xrightarrow[u_0]{*} S_0 \xrightarrow[u_1]{*} S_1 \xrightarrow[\text{be}]{\rightarrow} S_0 \xrightarrow[u_2]{*} S_2 \xrightarrow[\text{be}]{\rightarrow} S_0 \cdots$  where  $r$  is the  
 19 root of  $\pi$  and where the  $u_i$  do not cross  $S_0$  except at their sources.

20 Remark that the positions labelled by  $a$  are the same in all  $S_i$ , as there are back edges  
 21 from every  $S_{i+1}$  to  $S_0$ . The difference, however, is that these positions are labelled with  
 22  $a-$  in  $S_0$  and with  $a+$  in every  $S_{i+1}$ . Let  $P_0$  be the set of those positions.  $P_0$  is finite and  
 23 non empty. Now we would like, as in the proof of Proposition 13, to construct an infinite  
 24 thread along  $b_0$ . However, because  $P_0$  may contain more than one element, we cannot  
 25 know by advance, for each  $S_i$ , which  $p \in P_0$  will support an infinite thread. Thus, we  
 26 will use König's lemma to show the existence of such a thread. Let  $T_0$  be the tree whose  
 27 vertices are the pairs  $(i, p)$  where  $1 \leq i < \omega$  and  $p \in P_0$ , whose roots are the vertices of  
 28 the form  $(1, p)$  and where, for  $i > 1$ , the father of  $(i, p)$  is<sup>2</sup>  $(i-1, \mathfrak{t}(u_i)(p))$ . Here we have  
 29 to prove that  $\mathfrak{t}(u_i)(p)$  is defined and that it belongs to  $P_0$  for every  $i$  and  $p \in P_0$ . This is  
 30 ensured by Lemma 9 thanks to the labels.

31 Remark that every edge in  $T_0$  induces a progressing thread. Indeed, for  $i \geq 1$  and  $p \in P_0$ :

<sup>2</sup>Recall that  $\mathfrak{t}(u)$  and  $\mathfrak{T}(u)$  are defined in Definition 40, p. 48.

- 1     •  $\mathfrak{T}(u_i)(p)$  is a  $\nu$ -thread in  $u_i$ ,
- 2     • its target is  $p$  in  $S_i$ , which is labelled with  $a+$ ,
- 3     • and its source is  $p$  in  $S_0$ , which is labelled with  $a-$ .

4 An examination of the rules that may compose  $u_i$  shows that the only way for that to be  
 5 true is that  $\mathfrak{T}(u_i)(p)$  is progressing. Now  $T_0$  is an infinite tree with a finite number of  
 6 roots and an arity bounded by  $\mathbf{Card}(P_0)$ , hence, by Kőnig's lemma, it has an infinite  
 7 branch  $(1, p_1) \leftarrow (2, p_2) \leftarrow (3, p_3) \cdots$ .

8 This infinite branch induces in turn an infinite thread

$$9 \quad (S_0, p_0) \xrightarrow[\mathfrak{T}(u_1)(p_1)]{*} (S_1, p_1) \xrightarrow[\text{be}]{} (S_0, p_1) \xrightarrow[\mathfrak{T}(u_2)(p_2)]{*} (S_2, p_2) \xrightarrow[\text{be}]{} (S_0, p_2) \cdots$$

10 This thread is valid because every  $\mathfrak{T}(u_i)(p_i)$  is progressing. And it is indeed a thread  
 11 of  $b_0 = r \xrightarrow[u_0]{*} S_0 \xrightarrow[u_1]{*} S_1 \xrightarrow[\text{be}]{} S_0 \xrightarrow[u_2]{*} S_2 \xrightarrow[\text{be}]{} S_0 \cdots$ . Hence  $b_0$  is valid, which concludes this  
 12 demonstration.  $\square$

13 We now label our two examples with this new system. We will show that, while it is quite  
 14 straightforward to label the interleaving example (Figure 5.3) with this new system, it  
 15 requires, for  $\pi_\infty$  (Figure 5.2), to unfold one back edge.

16  $\pi_\infty$  is presented, on Figure 5.5, labelled according to the extended labelling of Figure 5.5.  
 17 To be able to label  $\pi_\infty$ , two preliminary steps were necessary:

- To make  $K$  apparent as a subformula of  $I$  and  $J$  respectively by decomposing  $I$  and  $J$  as:

$$I = I'[K] \qquad J = J'[K]$$

where

$$J'[Y] := \mu X((Y \wp X) \oplus \perp) \qquad I'[Y] := \mu Z((Z \wp J'[Y]) \oplus \perp).$$

- 18     • To do one step of unfolding on the rightmost back edge.

19 After that, we can take advantage of the two new facilities of the extended labelling:

- 20     1. to add three (*Rec*) rules, corresponding to the three ways for a branch of  $\pi_\infty$  to be  
 21         valid, as summarized in the following array.

	Shape of the branch	$A^* \cdot l^\omega$	$A^* \cdot r^\omega$	$l^* \cdot (r^+ \cdot l^+)^\omega$
1	Lowest $(Rec)$ visited $\infty^{\text{ly}}$	$b$	$a$	$c$
	Validating $\nu$ -formula	$H$	$G$	$K$

2     2. to label the three formulas  $H$ ,  $G$  and  $K$  at each corresponding  $(Rec)$ , using for  $K$  the  
3         ability to label several occurrences at a time, and to label deeply  $\nu$ -subformulas.

4 This indeed forms a correct labelling of  $\pi_\infty$  according to the extended labelling, hence  
5 ensuring its thread-validity.

### 6     5.3 Extending finitization

7 As we did in Section 5.1 for the labelling defined in Chapter 4, we will rely on the labelled  
8 presentation of the proofs in order to finitize them. More precisely, we will consider a  
9 fragment of the extended labelling of Definition 69, for which we will be able to finitize  
10 proofs.

11 **Definition 70** (Core fragment of the extended labelling). A  $\mu\text{MALL}^\omega$  b.e.-tree preproof  
12 labelled with the extended labelling of Definition 69 is in the *core fragment* of this  
13 extended labelling if:

- 14     1. no  $\nu$ -formula wears more than one variable
- 15     2.  $(Rec)$  is used only in the particular form  $\frac{\vdash \nu^{a^-} XA, \Gamma}{\vdash \nu XA, \Gamma} (Rec'(a))$  in which only one  
16         occurrence of  $\nu XA$  is labelled, and this occurrence is a formula of the sequent and  
17         not a strict subformula.

18 Observe already that both the interleaving example (Figure 5.3) and  $\pi_\infty$  (Figure 5.2)  
19 respect the two constraints of Definition 70, except for the labelling of  $K$  in  $\pi_\infty$ .

20 We show now how to finitize any labelled representation which verify those two restrictions.  
21 As the interleaving example of Figure 5.3 is labelled in this *core fragment*, it gives a  
22 finitization for the preproof of Figure 5.3. We will then show how to extend this method  
23 in an *ad hoc* way to finitize entirely  $\pi_\infty$  (Figure 5.2) from the labelling of Figure 5.5.

**Definition 71.** For any formula  $\nu XA$  and any unlabelled context  $\Gamma$ , we define the following formula translation:

$$\llbracket \nu^{\Gamma^-} XA[X] \rrbracket_e := \nu X \llbracket A \rrbracket_e [\otimes \Gamma^\perp \oplus X] \quad \llbracket \nu^{\Gamma^+} XA[X] \rrbracket_e := \otimes \Gamma^\perp \oplus \llbracket \nu^{\Gamma^-} XA[X] \rrbracket_e$$

Remark 21. With these definitions, we have

$$\llbracket \nu^{\Gamma^-} X A[X] \rrbracket_e \simeq \llbracket A \rrbracket_e [\llbracket \nu^{\Gamma^+} X A[X] \rrbracket_e]$$

1 through a simple  $\nu$ -unfolding.

**Proposition 20.** *With the notations of Definition 71, the following rules are derivable:*

$$\frac{\vdash \llbracket \nu X A \rrbracket_e, \Delta}{\vdash \llbracket \nu^{\Gamma^-} X A \rrbracket_e, \Delta} \llbracket (\text{LWk}(\Gamma^-)) \rrbracket_e \qquad \frac{\vdash \llbracket \nu^{\Gamma^-} X A \rrbracket_e, \Delta}{\vdash \llbracket \nu^{\Gamma^+} X A \rrbracket_e, \Delta} \llbracket (\text{LWk}(\Gamma^+)) \rrbracket_e$$

$$\frac{\vdash \llbracket \nu^{\Gamma^-} X A \rrbracket_e, \Gamma}{\vdash \llbracket \nu X A \rrbracket_e, \Gamma} \llbracket (\text{Rec}'(\Gamma)) \rrbracket_e \qquad \frac{}{\vdash \llbracket \nu^{\Gamma^+} X A \rrbracket_e, \Gamma} \llbracket (\text{Q}(\Gamma)) \rrbracket_e$$

2 *Proof.* See derivations on Figure 5.4. □

3 **Proposition 21.** *If  $\pi$  is a  $\mu\text{MALL}^\omega$  preproof with endsequent  $\Gamma$ , if  $\pi$  is labelled with*  
 4 *the **core fragment of the extended labelling** of Definition 70, then  $\pi$  can be turned into a*  
 5  *$\mu\text{MALL}$  proof of  $\Gamma$ .*

6 *Proof.* As for Theorem 5, it is enough, in order to turn a labelled formula into an  
 7 unlabelled one, to translate the  $\nu$  connectives, leaving all other connectives untouched.  
 8 The translation of Definition 68 is adapted using Definition 71 and picking, for  $\Gamma_{(a)}$ , the  
 9 context of the rule  $(\text{Rec}(a))$ .

10 By Proposition 20, the rules of the **core fragment of the extended labelling** are translated  
 11 to usual  $\mu\text{MALL}^\omega$  rules involving no label.

12 Remark moreover that  $\frac{\vdash \llbracket A[\nu^{\Gamma^+} X A[X]] \rrbracket_e, \Delta}{\vdash \llbracket \nu^{\Gamma^-} X A[X] \rrbracket_e, \Delta} (\nu)$  is the usual  $(\nu)$  rule.

13 These allow to translate any labelled proof verifying the two constraints of Definition 70  
 14 into a  $\mu\text{MALL}$  finitary proof. □

## 15 5.4 An ad hoc finitization of $\pi_\infty$

16 The method of the previous section works almost as well for finitizing  $\pi_\infty$  based on the  
 17 labelling of Figure 5.5: it allows to expand everything concerning the variables  $a$  and  $b$ .



$$\begin{aligned} \Gamma_c &:= F, G, H & \llbracket K^{c-} \rrbracket_e &:= \nu Y \mu_- ((\otimes \Gamma_c^\perp \oplus (I'[Y] \wp J'[Y])) \oplus \perp) \\ I^{c+} &:= \llbracket I_+ \rrbracket_e = \llbracket I'[K^{c+}] \rrbracket_e := \mu_- ((\otimes \Gamma_c^\perp \oplus (I'[\llbracket K^{c-} \rrbracket_e] \wp J'[\llbracket K^{c-} \rrbracket_e]))) \oplus \perp) \\ L^{c+} &:= \llbracket I'[K^{c+}] \wp J'[K^{c+}] \rrbracket_e := \otimes \Gamma_c^\perp \oplus (I'[\llbracket K^{c-} \rrbracket_e] \wp J'[\llbracket K^{c-} \rrbracket_e]) \end{aligned}$$

1 These definitions allow to make finitary the derivation of Figure 5.5, by expanding  
 2 every formula as explained above, and by replacing every rule dealing with labels with  
 3 an appropriate derivation, while leaving untouched the structure of rules not dealing  
 4 with labels. The result is presented on Figure 5.6a, in an abbreviated form, showing  
 5 the similarity with the original proof. The left premise of the bottommost ( $\&$ ) of the  
 6 finitization of  $\pi_\infty$  is fully expanded on Figure 5.7, the two premises of the topmost ( $\&$ )  
 7 are fully expanded, respectively, on Figures 5.8 and 5.9.

8 We now detail the analysis leading to this choice of formulas. To finitize  $\pi_\infty$  we try to  
 9 apply the same method as for the interleaving example of Figure 5.3 p. 104, by expanding  
 10 every labelled formula to a non-labelled one and expanding the rules that need it to  
 11 match these transforms. This works perfectly for  $H$  and  $G$ , which appear respectively as  
 12 formulas of the premises ( $Rec(b)$ ) and ( $Rec(a)$ ). But the situation is more delicate for  $K$   
 13 which we have to face a double difficulty: in the premise of ( $Rec(c)$ ),  $K$  is not a formula of  
 14 the sequent but a subformula, and it appears in two different formulas.

15 Let us try to transform this situation into one that would fit our method. First we would  
 16 like to have only one formula containing  $K$  instead of the two  $I$  and  $J$ . Unfortunately,  
 17 none of them can be unlabelled without breaking the labelling. Fortunately the solution  
 18 to that is easy:  $I, J$  is simply equivalent to  $I \wp J$ .

19 Now we would like  $I \wp J$  to be a  $\nu$ -formula that we could label. We already made use,  
 20 for the interleaving example of Figure 5.3, of the isomorphism

$$21 \quad A[\nu X B[A[X]]] \simeq \nu X A[B[X]] \quad (5.1)$$

22 to turn an almost- $\nu$ -formula into a real one. Let us apply that again.

23 The formula  $I \wp J$  is equal to  $L'[K]$  where  $L'[Y] := I'[Y] \wp J'[Y]$ , that is:  $I \wp J =$   
 24  $L'[\nu Y I'[Y]]$ . In order to apply an isomorphism of the form (5.1) we would like  $I'[Y]$  to  
 25 be of the form  $M'[L'[Y]]$  for a given  $M'$ . This is unfortunately not the case as  $I'[Y]$  is a  
 26 subformula of  $L'[Y]$ . However, a careful examination of the flow of  $I, J$  and  $K$  along the  
 27 loops of  $\pi_\infty$  makes apparent the fact that

$$28 \quad I'[Y] = \mu Z ((Z \wp J'[Y]) \oplus \perp) \simeq \mu_- ((I'[Y] \wp J'[Y]) \oplus \perp) = M'[L'[Y]]$$



$$\begin{array}{c}
\frac{}{\vdash F, G, H, \underline{L}^{c+}} \mathcal{Q}(c) \\
\frac{}{\vdash F, G, \underline{H}, L^{c+}} (\nu)(\oplus_1) \\
\frac{}{\vdash F, G, H, \underline{I}^{c+}} (\mu)(\oplus_0) \\
\frac{}{\vdash F, G, H, I^{c+}, \underline{J}_-} (\mu)(\oplus_1)(\perp) \\
\frac{}{\vdash F \wp G, H, I^{c+}, \underline{J}_-} (\wp) \\
\frac{}{\vdash F \wp G, \underline{G}, H, I^{c+}, \underline{J}_-} (\nu)(\oplus_1)(\perp) \\
\frac{}{\vdash F \wp G, \underline{G}^{a-}, H, I^{c+}, \underline{J}_-} (\text{LWk}(a-)) \\
\frac{}{\vdash F, G, H, \underline{L}^{c+}} \mathcal{Q}(a) \\
\frac{}{\vdash F, \underline{G}^{a-}, H, I^{c+}, \underline{J}_-} (\nu)(\oplus_0) \\
\frac{}{\vdash F, G^{a-}, H, \underline{K}^{c-}, \underline{J}_-} (\nu) \\
\frac{}{\vdash F, G^{a-}, H, \underline{J}_-} (\mu)(\oplus_0)(\wp) \\
\frac{}{\vdash F, G^{a-}, H, \underline{I}^{c+}, \underline{J}_-} (\mu)(\oplus_1)(\perp) \\
\frac{}{\vdash F, \underline{H}, G^{a-}, I^{c+}, \underline{J}_-} (\text{exc}) \\
\frac{}{\vdash F \wp \underline{H}, G^{a-}, I^{c+}, \underline{J}_-} (\wp) \\
\frac{}{\vdash F \wp \underline{H}, G^{a-}, \underline{H}, I^{c+}, \underline{J}_-} (\nu)(\oplus_0)(\perp) \\
\frac{}{\vdash F \wp G, \underline{G}^{a-}, H, I^{c+}, \underline{J}_-} (\mu) \\
\frac{}{\vdash \underline{F}, G^{a-}, H, I^{c+}, \underline{J}_-} (\text{Rec}(a)) \\
\frac{}{\vdash F, \underline{G}, H, I^{c+}, \underline{J}_-} (\nu)(\oplus_0) \\
\frac{}{\vdash F, \underline{G}, H, I^{c+}, \underline{J}_-} (\nu) \\
\frac{}{\vdash F, G, H, \underline{K}^{c-}, \underline{J}_-} (\mu)(\oplus_0)(\wp) \\
\frac{}{\vdash F, G, H, \underline{J}_-} (\mu)(\oplus_1)(\perp) \\
\frac{}{\vdash F, G, H, \underline{I}_-, \underline{J}_-} (\text{exc}) \\
\frac{}{\vdash F, \underline{H}, G, I_-, \underline{J}_-} (\wp) \\
\frac{}{\vdash F \wp \underline{H}, G, I_-, \underline{J}_-} (\nu)(\oplus_0)(\perp) \\
\frac{}{\vdash F \wp H, G, \underline{H}, I_-, \underline{J}_-} (\text{LWk}(b-)) \\
\frac{}{\vdash F \wp H, G, \underline{H}^{b-}, I_-, \underline{J}_-} (\&) \\
\frac{}{\vdash F, G, \underline{H}^{b+}, I_-, \underline{J}_-} \mathcal{Q}(b) \\
\frac{}{\vdash F, G, \underline{H}^{b-}, I_-, \underline{J}_-} (\nu)(\oplus_1) \\
\frac{}{\vdash F, G, H^{b-}, \underline{I}_-} (\mu)(\oplus_0)(\wp) \\
\frac{}{\vdash F, G, H^{b-}, \underline{I}_-} (\mu)(\oplus_1)(\perp) \\
\frac{}{\vdash F, G, H^{b-}, I_-, \underline{J}_-} (\wp) \\
\frac{}{\vdash F \wp \underline{G}, H^{b-}, I_-, \underline{J}_-} (\nu)(\oplus_1)(\perp) \\
\frac{}{\vdash F \wp G, \underline{G}, H^{b-}, I_-, \underline{J}_-} (\mu) \\
\frac{}{\vdash \underline{F}, G, H^{b-}, I_-, \underline{J}_-} (\text{Rec}(b)) \\
\frac{}{\vdash F, G, \underline{H}, I_-, \underline{J}_-} (\text{Rec}(c)) \\
\frac{}{\vdash F, G, H, I'[\underline{K}], J'[\underline{K}]} (\text{Rec}(c))
\end{array}$$

(a) Finitization of  $\pi_\infty$ . Brackets  $\llbracket \bullet \rrbracket_e$  should be put around every formula and rule name. They were omitted only for the sake of readability.

$$\begin{array}{c}
\frac{}{\vdash F, F^\perp} \text{(id)} \quad \frac{}{\vdash G, G^\perp} \text{(id)} \quad \frac{}{\vdash \llbracket I_- \rrbracket_e^\perp, \llbracket I_- \rrbracket_e} \text{(id)} \quad \frac{}{\vdash \llbracket J_- \rrbracket_e^\perp, \llbracket J_- \rrbracket_e} \text{(id)} \\
\hline
\vdash F, G, F^\perp \otimes G^\perp \otimes \llbracket I_- \rrbracket_e^\perp \otimes \llbracket J_- \rrbracket_e^\perp, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\otimes)^3 \\
\hline
\vdash F, G, F^\perp \otimes G^\perp \otimes \llbracket I_- \rrbracket_e^\perp \otimes \llbracket J_- \rrbracket_e^\perp \oplus \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\oplus_0) \\
\hline
\vdash F, G, \perp \oplus (F^\perp \otimes G^\perp \otimes \llbracket I_- \rrbracket_e^\perp \otimes \llbracket J_- \rrbracket_e^\perp \oplus \llbracket H^{b-} \rrbracket_e), \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\oplus_1) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\nu) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e \wp \llbracket J_- \rrbracket_e \quad (\wp) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, (\llbracket I_- \rrbracket_e \wp \llbracket J_- \rrbracket_e) \oplus \perp \quad (\oplus_0) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e \quad (\mu) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e \quad (\perp) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \perp \quad (\oplus_1) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, (\llbracket K^{c-} \rrbracket_e \wp \llbracket J_- \rrbracket_e) \oplus \perp \quad (\mu) \\
\hline
\vdash F, G, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\wp) \\
\hline
\vdash \underline{F} \wp \underline{G}, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\perp) \\
\hline
\vdash \underline{F} \wp \underline{G}, \perp, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\oplus_1) \\
\hline
\vdash \underline{F} \wp \underline{G}, \underline{G} \oplus \perp, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e \quad (\nu) \\
\hline
\vdash \underline{F} \wp \underline{G}, \underline{G}, \llbracket H^{b-} \rrbracket_e, \llbracket I_- \rrbracket_e, \llbracket J_- \rrbracket_e
\end{array}$$

Figure 5.7. Left premise of the bottommost (&) of the finitization of  $\pi_\infty$



$$\begin{array}{c}
\frac{}{\vdash F, F^\perp} \text{(id)} \quad \frac{}{\vdash H^\perp, H} \text{(id)} \quad \frac{}{\vdash (I^{c+})^\perp, I^{c+}} \text{(id)} \quad \frac{}{\vdash \llbracket J_- \rrbracket_e^\perp, \llbracket J_- \rrbracket_e} \text{(id)} \\
\hline
\vdash F, F^\perp \otimes H^\perp \otimes (I^{c+})^\perp \otimes \llbracket J_- \rrbracket_e^\perp, H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\otimes)^3 \\
\hline
\vdash F, F^\perp \otimes H^\perp \otimes (I^{c+})^\perp \otimes \llbracket J_- \rrbracket_e^\perp \oplus \llbracket G^{a-} \rrbracket_e, H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\oplus_0) \\
\hline
\vdash F, (F^\perp \otimes H^\perp \otimes (I^{c+})^\perp \otimes \llbracket J_- \rrbracket_e^\perp \oplus \llbracket G^{a-} \rrbracket_e) \oplus \perp, H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\oplus_0) \\
\hline
\vdash F, \llbracket G^{a-} \rrbracket_e, H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\nu) \\
\hline
\frac{}{\vdash F, \llbracket G^{a-} \rrbracket_e, H, I^{c+}, \llbracket J_- \rrbracket_e} \text{(}\wp\text{)} \\
\frac{}{\vdash F, \llbracket G^{a-} \rrbracket_e, H, \llbracket K^{c-} \rrbracket_e \wp \llbracket J_- \rrbracket_e} \text{(}\wp\text{)} \\
\hline
\vdash F, \llbracket G^{a-} \rrbracket_e, H, (\llbracket K^{c-} \rrbracket_e \wp \llbracket J_- \rrbracket_e) \oplus \perp \quad \oplus_0 \\
\hline
\vdash F, \llbracket G^{a-} \rrbracket_e, H, \llbracket J_- \rrbracket_e \quad (\mu) \\
\hline
\frac{}{\vdash F, \llbracket G^{a-} \rrbracket_e, H, \llbracket J_- \rrbracket_e} \text{(}\perp\text{)} \\
\frac{}{\vdash F, \llbracket G^{a-} \rrbracket_e, H, \perp, \llbracket J_- \rrbracket_e} \text{(}\perp\text{)} \\
\hline
\vdash F, \llbracket G^{a-} \rrbracket_e, H, L^{c+} \oplus \perp, \llbracket J_- \rrbracket_e \quad (\oplus_1) \\
\hline
\vdash F, \llbracket G^{a-} \rrbracket_e, H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\mu) \\
\hline
\vdash F \wp H, \llbracket G^{a-} \rrbracket_e, I^{c+}, \llbracket J_- \rrbracket_e \quad (\wp) \\
\hline
\vdash F \wp H, \llbracket G^{a-} \rrbracket_e, \perp, I^{c+}, \llbracket J_- \rrbracket_e \quad (\perp) \\
\hline
\vdash F \wp H, \llbracket G^{a-} \rrbracket_e, \perp \oplus H, I^{c+}, \llbracket J_- \rrbracket_e \quad (\oplus_0) \\
\hline
\vdash F \wp H, \llbracket G^{a-} \rrbracket_e, \underline{H}, I^{c+}, \llbracket J_- \rrbracket_e \quad (\nu)
\end{array}$$

Figure 5.9. Right premise of the topmost (&) of the finitization of  $\pi_\infty$

1 where

$$M'[Y] := \mu_-(Y \oplus \perp)$$

2 in which we use the notation  $\mu_-A$  to denote a  $\mu XA$  with  $X$  not appearing free in  $A$ .  
 3 This degenerate  $\mu$  binder could be removed to simplify the formulas involved in the  
 4 finitization, but we keep it to stay as close as possible to the original structure of  $I$ ,  
 5 trying to preserve its head connective.

6 When we stick all that together we get

$$I \wp J \simeq L'[\nu Y M'[L'[Y]]] \simeq \nu Y L'[M'[Y]]$$

7 which is a  $\nu$ -formula that we know, when labelled, how to expand into an unlabelled  
 8 formula. If we stopped here our analysis, we would then define:

$$\Gamma_c := F, G, H \quad L^{c-} := \nu Y L'[M'[\otimes \Gamma_c^\perp \oplus Y]] \quad L^{c+} := \otimes \Gamma_c^\perp \oplus L^{c-}.$$

9 However we will do yet a bit more work in order to get the structure of  $L^{c-}$  closer to  
 10 that of  $I \wp J$ .

11 Indeed the isomorphism (5.1) can be used in the other direction:

$$\begin{aligned} \nu Y L'[M'[\otimes \Gamma_c^\perp \oplus Y]] &\simeq L'[\nu Y M'[\otimes \Gamma_c^\perp \oplus L'[Y]]] \\ &= I'[\nu Y M'[\otimes \Gamma_c^\perp \oplus L'[Y]]] \wp J'[\nu Y.M'[\otimes \Gamma_c^\perp \oplus L'[Y]]]. \end{aligned}$$

12 This, finally, leads us to define:

$$\Gamma_c := F, G, H \quad \text{and} \quad K^{c-} := \nu Y M'[\otimes \Gamma_c^\perp \oplus L'[Y]]$$

which allows to expand  $I'[K^{c-}]$  and  $J'[K^{c-}]$ . On the other hand, this is not sufficient to  
 define an expansion of  $K^{c+}$ , and we still need an *ad hoc* treatment for formulas containing  
 it:

$$"I'[K^{c+}]" := I^{c+} := M'[\otimes \Gamma_c^\perp \oplus L'[K^{c-}]] \quad "I'[K^{c+}] \wp J'[K^{c+}]" := L^{c+} := \otimes \Gamma_c^\perp \oplus L'[K^{c-}]$$

1 With these expansions of labelled formulas into unlabelled formulas, we can finitize the  
 2 derivation of Figure 5.5 into the very close derivation of Figure 5.6a, on which the rules  
 3 dealing with labelling can be expanded into  $\mu$ MALL derivations.

## 4 5.5 Related works on Chapters 4 and 5

5 *Labelling and local certification* is the basis of our approach. The idea of labelling  $\mu$ -  
 6 formulas to gather information on fixed-points unfoldings is naturally not new, already to  
 7 be found in fixed-point approximation methods (see [Dax, Hofmann, and Lange, 2006] for  
 8 instance). The closest work in this direction is Stirling [2014]’s annotated proofs and the  
 9 application Afshari and Leigh [2017] made of such proofs in obtaining completeness for  
 10 the modal  $\mu$ -calculus. Our labelling system works quite differently since only fixed-point  
 11 operators are labelled while, in Stirling’s annotated proofs, every formula is labelled  
 12 and labels are transmitted to immediate subformulas with a label extension on greatest  
 13 fixed-points.

14 *Finitization of circular proofs* has been recently a very active topic with much research  
 15 effort on solving Brotherston-Simpson’s conjecture. The following recent contributions  
 16 were made in the setting of Martin-Löf’s inductive definitions: firstly, Berardi and Tatsuta  
 17 [2017b] proved that, in general, the equivalence is false by providing a counter-example  
 18 inspired by the *Hydra* paradox. Secondly, Simpson [2017] on the one hand and Berardi  
 19 and Tatsuta [2017a] on the other hand provided a positive answer in the restricted  
 20 frameworks when the proof system contains arithmetics. While Simpson used tools from  
 21 reverse mathematics and internalized circular proofs in  $\text{ACA}_0$ , a fragment of second-order  
 22 arithmetic with a comprehension axiom on arithmetical statements, Tatsuta and Berardi  
 23 proved an equivalent result by a direct proof translation relying on an arithmetical version  
 24 of the Ramsey and Podelsky-Rybalchenko theorems.

25 *Circular proof search* triggered interest compared to proof system with explicit inductive  
 26 invariants (lacking subformula property). As we already mentionned, this has actually  
 27 been turned to practice by Brotherston, Gorogiannis, and Petersen [2012].

## 28 5.6 Conclusion

29 In this chapter, we contributed to the theory of circular proofs for  $\mu$ MALL in two  
 30 directions: (i) expanding the results of Chapter 4 by identifying larger fragments of  
 31 circular proofs for which local conditions account for the validity of circular proof objects  
 32 (in contrast to the global nature of thread conditions) and (ii) designing methods for

1 translating circular proofs to finitary proofs (with explicit (co)induction rules). To do so,  
2 we introduced and studied several labelling systems, for circular proofs, and made the  
3 following contributions:

- 4 **(i)** First, we investigated how such labellings ensure validity of a labellable proof, turning  
5 a global and complex problem into a local and simpler one. Indeed, validity-checking  
6 is far from trivial in circular proof-theory for fixed-point logics: the best known  
7 bound for this problem is generally **PSPACE** and, in the case of  $\mu$ MALL and similar  
8  $\mu$ -calculi, we proved in Chapter 3 that checking thread-validity is in fact a PSPACE-  
9 complete problem. We extended the results of Chapter 4 by defining a second, more  
10 liberal labelling, which does not force back edges to target ( $\nu$ ) inferences. We proved  
11 that this extended labelling still ensure thread-validity, and we were able to finitize  
12 a fragment of it.
- 13 **(ii)** Second, we provided evidence on the usability of such labellings as a helpful guide in  
14 the generation of (co)inductive invariants which are necessary to translate a circular  
15 proof in a finitary proof system with (co)induction rules *à la* Park. We provided  
16 a full finitization method for the fairly restricted labelling system of Chapter 4,  
17 which contains at least all the translations of  $\mu$ MALL proofs as well as for a core  
18 fragment of the extended labelling of this chapter, which subsumes the labelling of  
19 Chapter 4. However, this fragment is too constrained to treat standard examples  
20 that we discuss in the chapter, and which contain most of the difficulties in finitizing  
21 circular proofs, namely: (i) interleaving of fixed-points and (ii) interleaving of back  
22 edges resulting in various choices of a valid thread to support a branch.



# 6 Conclusion

## 6.1 General conclusion

We started this journey by explaining that the already existing finitary systems for least and greatest fixed points of formulas, with induction and coinduction rules [Baelde, 2012], needing explicit invariants were not satisfactory. We explained that the infinitary setting designed by [Baelde, Doumane, and Saurin, 2016] was proposed as a solution to these issues. We then complained about the fact that these infinite proof trees are infinite and explained that this thesis is only concerned with the circular representations of infinite proofs. After proving that the thread criterion is PSPACE-complete on these circular representations, we argued for the search of a smaller, easier to check criterion, which would still be expressive enough for practical use. We provided such a criterion, the loop criterion, and we finally showed how any circular preproof valid for our loop criterion can be turned into a finitary proof à la Baelde with explicit induction and coinduction invariants.

It may seem that after all this work we finally reached our starting point.

But, as one may expect, we learned some valuable things through this journey.

- The result of PSPACE-completeness of the thread criterion (Theorem 4), which is the main contribution of Chapter 3, is, as far as we know, the first of its kind. Not only does it give a precise characterization of the complexity of the thread criterion, but its proof also made apparent the deep link that seems to exist between this global validity condition and program termination techniques such as size-change termination [Lee, Jones, and Ben-Amram, 2001].
- The fragments of  $\mu\text{MALL}^\omega$  defined in Chapter 4 and in Section 5.2 combine advantages of both  $\mu\text{MALL}$  and  $\mu\text{MALL}^\omega$ . On one side, when they are viewed as unlabelled systems, they are circular systems, in it is not necessary to find and write explicit invariants when introducing a  $\nu$  or eliminating a  $\mu$ . On the other side, when they are viewed as circular systems, they do not necessitate a global validity criterion which may be subtle to understand, but they work with basic sequent calculus constructions in which the validity of the proof, which is finite,

1 comes directly from the validity of each rule. Over all that, going from a labelled  
 2 proof to an unlabelled one is immediate, and checking and labelling an unlabelled  
 3 preproof is doable in quadratic time.

- 4 • The finitization results of Chapter 5 do not only tell that the different fragments of  
 5  $\mu\text{MALL}^\omega$  studied in Chapter 4 and in Section 5.2 do not have more expressivity  
 6 than  $\mu\text{MALL}$ . These are constructive results, providing an explicit method to  
 7 synthesize induction and coinduction invariants and to turn a circular proof into a  
 8 finitary, well-founded one.

## 9 6.2 Future works

10 Our work in Chapter 3 suggests deep connections between thread-validity and size-change  
 11 termination (SCT) [Lee, Jones, and Ben-Amram, 2001], which was already hinted by other  
 12 authors [Dax, Hofmann, and Lange, 2006, Hyvernats, 2014, 2019, Lepigre and Raffalli,  
 13 2019]. More generally, the connection between SCT and thread validity in  $\mu$ -calculi is not  
 14 yet well understood despite those early investigations. More than a connection, this looks  
 15 like an interplay: size-change termination is originally shown decidable by using Büchi  
 16 automata and size-change graphs can be used to show validity of circular proofs [Dax,  
 17 Hofmann, and Lange, 2006]. There seems to be connections with our labelling system  
 18 too, that we plan to investigate further in the future.

19 We already underlined in Section 5.5 the similarities between our system and the systems  
 20 by Dax, Hofmann, and Lange [2006], Stirling [2014], Afshari and Leigh [2017]. Despite  
 21 their differences, the relationships of those systems should be investigated further (in  
 22 particular the role of the annotation restriction rule of Stirling [2014, Definition 4]).

23 In addition to investigating more closely those connections, we have several directions  
 24 for improving our labelled proof system. The first task is to lift the results of Chapter 4  
 25 and Section 5.1 to the extended labelling system of Section 5.2. Indeed, for the more  
 26 restricted fragment and given a circular proof presented as a graph with back edges, we  
 27 provided a method to effectively check that one can assign labels. It is therefore natural  
 28 to expect extending these results to the relaxed framework. Another point we plan to  
 29 investigate is whether every circular  $\mu\text{MALL}$  proof can be labelled. Even though this  
 30 can look paradoxical given the complexity of checking validity of circular proofs, one  
 31 should keep in mind that it might well be the case that, in order to label a circular proof  
 32 presented as a tree with back edges, one has to unfold some of the back edges, or possibly  
 33 pick a different finite representation of the proof which may result in a space blow up.  
 34 Related to this question is the connection of our labelling methods with size-change  
 35 termination methods. Indeed, in designing the extended labelling, one gets closer to the  
 36 kind of constructions one finds in SCT-based approaches: this should be investigated

1 further since it may also be a key for our finitization objective. Note that the previous  
2 two directions would lead to a solution to the Brotherston-Simpson conjecture.

3 Our work in Chapter 5 helped closing the gap between  $\mu\text{MALL}^\omega$  and  $\mu\text{MALL}$  by proposing  
4 a rich fragment of  $\mu\text{MALL}^\omega$  that is equivalent  $\mu\text{MALL}$ . A very natural question for future  
5 work is to extend the still *ad hoc* finitization method presented in the last section to the  
6 whole fragment of relaxed labelled proofs, and maybe to all  $\mu\text{MALL}^\omega$ .

7 We also wish to investigate the potential use of labellings in circular proof-search. Indeed,  
8 there are several different labellings for a given finite derivation with back edges where the  
9 labels are weakened. Proposition 15 characterizes least and greatest validating sets: those  
10 extremal validating sets correspond to different strategies in placing the labels, which  
11 have different properties with respect to the ability to form back edges or to validate the  
12 proof that one may exploit in proof-search.



# 1 Bibliography

2 Smbat Abian and Arthur B Brown. A theorem on partially ordered sets, with applications  
3 to fixed point theorems. *Canadian Journal of Mathematics*, 13:78–82, 1961.

4 Peter Aczel. An introduction to inductive definitions. In *Studies in Logic and the*  
5 *Foundations of Mathematics*, volume 90, pages 739–782. Elsevier, 1977.

6 Bahareh Afshari and Graham E. Leigh. Cut-free completeness for modal mu-calculus.  
7 In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017,*  
8 *Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:  
9 10.1109/LICS.2017.8005088. URL <https://doi.org/10.1109/LICS.2017.8005088>.

10 Alfred V. Aho and Jeffrey D. Ullman. The universality of data retrieval languages. In  
11 Alfred V. Aho, Stephen N. Zilles, and Barry K. Rosen, editors, *Conference Record*  
12 *of the Sixth Annual ACM Symposium on Principles of Programming Languages, San*  
13 *Antonio, Texas, USA, January 1979*, pages 110–120. ACM Press, 1979. doi: 10.1145/  
14 567752.567763. URL <https://doi.org/10.1145/567752.567763>.

15 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cam-  
16 bridge University Press, 2009. ISBN 978-0-521-42426-4. URL <http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264>.

18 David Baelde. On the proof theory of regular fixed points. In Martin Giese and  
19 Arild Waaler, editors, *Automated Reasoning with Analytic Tableaux and Related Meth-*  
20 *ods, 18th International Conference, TABLEAUX 2009, Oslo, Norway, July 6-10,*  
21 *2009. Proceedings*, volume 5607 of *Lecture Notes in Computer Science*, pages 93–107.  
22 Springer, 2009. doi: 10.1007/978-3-642-02716-1\_8. URL [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-642-02716-1_8)  
23 [978-3-642-02716-1\\_8](https://doi.org/10.1007/978-3-642-02716-1_8).

24 David Baelde. Least and greatest fixed points in linear logic. *ACM Trans. Comput.*  
25 *Log.*, 13(1):2:1–2:44, 2012. doi: 10.1145/2071368.2071370. URL [https://doi.org/10.](https://doi.org/10.1145/2071368.2071370)  
26 [1145/2071368.2071370](https://doi.org/10.1145/2071368.2071370).

27 David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In  
28 Nachum Dershowitz and Andrei Voronkov, editors, *Logic for Programming, Artificial*  
29 *Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan,*

## Bibliography

- 1     *Armenia, October 15-19, 2007, Proceedings*, volume 4790 of *Lecture Notes in Computer*  
2     *Science*, pages 92–106. Springer, 2007. doi: 10.1007/978-3-540-75560-9\_9. URL  
3     [https://doi.org/10.1007/978-3-540-75560-9\\_9](https://doi.org/10.1007/978-3-540-75560-9_9).
- 4     David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multi-  
5     multiplicative additive case. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL*  
6     *Annual Conference on Computer Science Logic, CSL 2016, August 29 - September*  
7     *1, 2016, Marseille, France*, volume 62 of *LIPICs*, pages 42:1–42:17. Schloss Dagstuhl  
8     - Leibniz-Zentrum fuer Informatik, 2016. doi: 10.4230/LIPICs.CSL.2016.42. URL  
9     <https://doi.org/10.4230/LIPICs.CSL.2016.42>.
- 10    Stefano Berardi and Makoto Tatsuta. Equivalence of inductive definitions and cyclic  
11    proofs under arithmetic. In *32nd Annual ACM/IEEE Symposium on Logic in Computer*  
12    *Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer  
13    Society, 2017a. doi: 10.1109/LICS.2017.8005114. URL <https://doi.org/10.1109/LICS.2017.8005114>.
- 14    Stefano Berardi and Makoto Tatsuta. Classical system of martin-löf’s inductive defi-  
15    nitions is not equivalent to cyclic proof system. In Javier Esparza and Andrzej S.  
16    Murawski, editors, *Foundations of Software Science and Computation Structures*  
17    - *20th International Conference, FOSSACS 2017, Held as Part of the European*  
18    *Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala,*  
19    *Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Com-*  
20    *puter Science*, pages 301–317, 2017b. doi: 10.1007/978-3-662-54458-7\_18. URL  
21    [https://doi.org/10.1007/978-3-662-54458-7\\_18](https://doi.org/10.1007/978-3-662-54458-7_18).
- 22    Stefano Berardi and Makoto Tatsuta. Intuitionistic podelski-rybalchenko theorem and  
23    equivalence between inductive definitions and cyclic proofs. In Corina Cîrstea, editor,  
24    *Coalgebraic Methods in Computer Science - 14th IFIP WG 1.3 International Workshop,*  
25    *CMCS 2018, Colocated with ETAPS 2018, Thessaloniki, Greece, April 14-15, 2018,*  
26    *Revised Selected Papers*, volume 11202 of *Lecture Notes in Computer Science*, pages  
27    13–33. Springer, 2018. doi: 10.1007/978-3-030-00389-0\_3. URL [https://doi.org/10.1007/978-3-030-00389-0\\_3](https://doi.org/10.1007/978-3-030-00389-0_3).
- 28    Stefano Berardi and Makoto Tatsuta. Classical system of martin-lof’s inductive definitions  
29    is not equivalent to cyclic proofs. *Log. Methods Comput. Sci.*, 15(3), 2019. doi: 10.  
30    23638/LMCS-15(3:10)2019. URL [https://doi.org/10.23638/LMCS-15\(3:10\)2019](https://doi.org/10.23638/LMCS-15(3:10)2019).
- 31    Nicolas Bourbaki. Sur le theoreme de zorn. *Archiv der Mathematik*, 2(6):434–437, 1949.
- 32    James Brotherston. *Sequent calculus proof systems for inductive definitions*. PhD thesis,  
33    University of Edinburgh. College of Science and Engineering. School of . . . , 2006.
- 34    James Brotherston and Alex Simpson. Complete sequent calculi for induction and infinite  
35
- 36

- 1 descent. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12*  
2 *July 2007, Wroclaw, Poland, Proceedings*, pages 51–62. IEEE Computer Society, 2007.  
3 doi: 10.1109/LICS.2007.16. URL <https://doi.org/10.1109/LICS.2007.16>.
- 4 James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent.  
5 *J. Log. Comput.*, 21(6):1177–1216, 2011. doi: 10.1093/logcom/exq052. URL <https://doi.org/10.1093/logcom/exq052>.
- 7 James Brotherston, Nikos Gorgiannis, and Rasmus Lerchedahl Petersen. A generic cyclic  
8 theorem prover. In Ranjit Jhala and Atsushi Igarashi, editors, *Programming Languages*  
9 *and Systems - 10th Asian Symposium, APLAS 2012, Kyoto, Japan, December 11-13,*  
10 *2012. Proceedings*, volume 7705 of *Lecture Notes in Computer Science*, pages 350–  
11 367. Springer, 2012. doi: 10.1007/978-3-642-35182-2\_25. URL [https://doi.org/10.1007/978-3-642-35182-2\\_25](https://doi.org/10.1007/978-3-642-35182-2_25).
- 13 Pierre Clairambault. Least and greatest fixpoints in game semantics. In Luca  
14 de Alfaro, editor, *Foundations of Software Science and Computational Structures,*  
15 *12th International Conference, FOSSACS 2009, Held as Part of the Joint Euro-*  
16 *pean Conferences on Theory and Practice of Software, ETAPS 2009, York, UK,*  
17 *March 22-29, 2009. Proceedings*, volume 5504 of *Lecture Notes in Computer Sci-*  
18 *ence*, pages 16–31. Springer, 2009. doi: 10.1007/978-3-642-00596-1\_3. URL [https://doi.org/10.1007/978-3-642-00596-1\\_3](https://doi.org/10.1007/978-3-642-00596-1_3).
- 20 Pierre Clairambault. *Logique et Interaction : une Étude Sémantique de la Totalité. (Logic*  
21 *and Interaction : a Semantic Study of Totality)*. PhD thesis, Paris Diderot University,  
22 France, 2010. URL <https://tel.archives-ouvertes.fr/tel-00459307>.
- 23 Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Arch. Math. Log.*,  
24 28(3):181–203, 1989. doi: 10.1007/BF01622878. URL <https://doi.org/10.1007/BF01622878>.
- 26 Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order (2. ed.)*.  
27 Cambridge University Press, 2002. ISBN 978-0-521-78451-1.
- 28 Anuj Dawar and Yuri Gurevich. Fixed point logics. *Bull. Symb. Log.*, 8(1):65–88, 2002.  
29 doi: 10.2178/bsl/1182353853. URL <https://doi.org/10.2178/bsl/1182353853>.
- 30 Christian Dax, Martin Hofmann, and Martin Lange. A proof system for the linear time  $\mu$ -  
31 calculus. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS 2006: Foundations of*  
32 *Software Technology and Theoretical Computer Science, 26th International Conference,*  
33 *Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in*  
34 *Computer Science*, pages 273–284. Springer, 2006. doi: 10.1007/11944836\_26. URL  
35 [https://doi.org/10.1007/11944836\\_26](https://doi.org/10.1007/11944836_26).

## Bibliography

- 1 J. W. de Bakker and Willem P. de Roever. A calculus for recursive program schemes. In  
2 Maurice Nivat, editor, *Automata, Languages and Programming, Colloquium, Paris,*  
3 *France, July 3-7, 1972*, pages 167–196. North-Holland, Amsterdam, 1972.
- 4 Amina Doumane. Constructive completeness for the linear-time  $\mu$ -calculus. In *32nd*  
5 *Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik,*  
6 *Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017a. doi: 10.1109/  
7 LICS.2017.8005075. URL <https://doi.org/10.1109/LICS.2017.8005075>.
- 8 Amina Doumane. *On the infinitary proof theory of logics with fixed points. (Théorie de la*  
9 *démonstration infinitaire pour les logiques à points fixes)*. PhD thesis, Paris Diderot Uni-  
10 versity, France, 2017b. URL <https://tel.archives-ouvertes.fr/tel-01676953>.
- 11 Amina Doumane, David Baelde, Lucca Hirschi, and Alexis Saurin. Towards completeness  
12 via proof search in the linear time  $\mu$ -calculus: The case of büchi inclusions. In Martin  
13 Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual*  
14 *ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY,*  
15 *USA, July 5-8, 2016*, pages 377–386. ACM, 2016. doi: 10.1145/2933575.2933598. URL  
16 <https://doi.org/10.1145/2933575.2933598>.
- 17 Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-  
18 elimination. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013*  
19 *(CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPICs*, pages  
20 248–262. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. doi: 10.4230/  
21 LIPICs.CSL.2013.248. URL <https://doi.org/10.4230/LIPICs.CSL.2013.248>.
- 22 Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische*  
23 *zeitschrift*, 39(1):176–210, 1935a.
- 24 Gerhard Gentzen. Untersuchungen über das logische schließen. ii. *Mathematische*  
25 *Zeitschrift*, 39(1):405–431, 1935b.
- 26 Gerhard Gentzen. Investigations into logical deduction. *The Collected Papers of Gerhard*  
27 *Gentzen*, pages 68–131, 1969. URL <https://ci.nii.ac.jp/naid/10007157703/en/>.
- 28 Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987. doi: 10.1016/  
29 0304-3975(87)90045-4. URL [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4).
- 30 Jean-Yves Girard. Proof theory and logical complexity. *Annals of Pure and Applied*  
31 *Logic*, 53(4):197, 1991.
- 32 Jean-Yves Girard. *The Blind Spot: lectures on logic*. European Mathematical Society,  
33 2011.

- 1 Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University  
2 Press, New York, NY, USA, 1989. ISBN 0-521-37181-3.
- 3 Nikos Gorogiannis and Reuben Rowe. The cyclist theorem prover, 2014. URL <http://www.cyclist-prover.org/>.  
4
- 5 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and  
6 Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar,  
7 February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, 2002. Springer.  
8 ISBN 3-540-00388-6. doi: 10.1007/3-540-36387-4. URL [https://doi.org/10.1007/  
9 3-540-36387-4](https://doi.org/10.1007/3-540-36387-4).
- 10 Yuri Gurevich and Saharon Shelah. Fixed-point extensions of first-order logic. *Ann.  
11 Pure Appl. Log.*, 32:265–280, 1986. doi: 10.1016/0168-0072(86)90055-2. URL [https://doi.org/10.1016/0168-0072\(86\)90055-2](https://doi.org/10.1016/0168-0072(86)90055-2).  
12
- 13 Pierre Hyvernat. The size-change termination principle for constructor based languages.  
14 *Logical Methods in Computer Science*, 10(1), 2014. doi: 10.2168/LMCS-10(1:11)2014.  
15 URL [https://doi.org/10.2168/LMCS-10\(1:11\)2014](https://doi.org/10.2168/LMCS-10(1:11)2014).
- 16 Pierre Hyvernat. The size-change principle for mixed inductive and coinductive types.  
17 *CoRR*, abs/1901.07820, 2019. URL <http://arxiv.org/abs/1901.07820>.
- 18 Neil D. Jones. *Computability and complexity - from a programming perspective*. Founda-  
19 tions of computing series. MIT Press, 1997. ISBN 978-0-262-10064-9.
- 20 Roope Kaivola. Axiomatising linear time mu-calculus. In Insup Lee and Scott A.  
21 Smolka, editors, *CONCUR '95: Concurrency Theory, 6th International Conference,  
22 Philadelphia, PA, USA, August 21-24, 1995, Proceedings*, volume 962 of *Lecture Notes  
23 in Computer Science*, pages 423–437. Springer, 1995a. doi: 10.1007/3-540-60218-6\\_32.  
24 URL [https://doi.org/10.1007/3-540-60218-6\\_32](https://doi.org/10.1007/3-540-60218-6_32).
- 25 Roope Kaivola. A simple decision method for the linear time mu-calculus. In *Structures  
26 in Concurrency Theory*, pages 190–204. Springer, 1995b.
- 27 Dexter Kozen. On induction vs. \*-continuity. In Dexter Kozen, editor, *Logics of Programs,  
28 Workshop, Yorktown Heights, New York, USA, May 1981*, volume 131 of *Lecture Notes  
29 in Computer Science*, pages 167–176. Springer, 1981. doi: 10.1007/BFb0025782. URL  
30 <https://doi.org/10.1007/BFb0025782>.
- 31 Dexter Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:  
32 333–354, 1983. doi: 10.1016/0304-3975(82)90125-6. URL [https://doi.org/10.1016/  
33 0304-3975\(82\)90125-6](https://doi.org/10.1016/0304-3975(82)90125-6).

## Bibliography

- 1 Jean-Louis Lassez, V. L. Nguyen, and Liz Sonenberg. Fixed point theorems and semantics:  
2 A folk tale. *Inf. Process. Lett.*, 14(3):112–116, 1982. doi: 10.1016/0020-0190(82)90065-5.  
3 URL [https://doi.org/10.1016/0020-0190\(82\)90065-5](https://doi.org/10.1016/0020-0190(82)90065-5).
- 4 Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle  
5 for program termination. In Chris Hankin and Dave Schmidt, editors, *Conference*  
6 *Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of*  
7 *Programming Languages, London, UK, January 17-19, 2001*, pages 81–92. ACM, 2001.  
8 doi: 10.1145/360204.360210. URL <http://doi.acm.org/10.1145/360204.360210>.
- 9 Rodolphe Lepigre and Christophe Raffalli. Practical subtyping for curry-style languages.  
10 *ACM Trans. Program. Lang. Syst.*, 41(1):5:1–5:58, 2019. doi: 10.1145/3285955. URL  
11 <https://doi.org/10.1145/3285955>.
- 12 George Markowsky. Chain-complete posets and directed sets with applications. *Algebra*  
13 *universalis*, 6(1):53–68, 1976.
- 14 David Park. Fixpoint induction and proofs of program properties. *Machine intelligence*,  
15 5(59-78):5–3, 1969.
- 16 Sylvain Perifel. *Complexité algorithmique*. Ellipses, 2014.
- 17 Dominique Perrin and Jean-Éric Pin. *Infinite words: automata, semigroups, logic and*  
18 *games*, volume 141. Academic Press, 2004.
- 19 Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations*  
20 *of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November*  
21 *1977*, pages 46–57. IEEE Computer Society, 1977. doi: 10.1109/SFCS.1977.32. URL  
22 <https://doi.org/10.1109/SFCS.1977.32>.
- 23 Reuben N. S. Rowe and James Brotherston. Automatic cyclic termination proofs  
24 for recursive procedures in separation logic. In Yves Bertot and Viktor Vafeiadis,  
25 editors, *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and*  
26 *Proofs, CPP 2017, Paris, France, January 16-17, 2017*, pages 53–65. ACM, 2017. doi:  
27 10.1145/3018610.3018623. URL <https://doi.org/10.1145/3018610.3018623>.
- 28 Luigi Santocanale. A calculus of circular proofs and its categorical semantics. In  
29 Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and*  
30 *Computation Structures, 5th International Conference, FOSSACS 2002. Held as Part*  
31 *of the Joint European Conferences on Theory and Practice of Software, ETAPS 2002*  
32 *Grenoble, France, April 8-12, 2002, Proceedings*, volume 2303 of *Lecture Notes in*  
33 *Computer Science*, pages 357–371. Springer, 2002. doi: 10.1007/3-540-45931-6\_25.  
34 URL [https://doi.org/10.1007/3-540-45931-6\\_25](https://doi.org/10.1007/3-540-45931-6_25).

- 1 Alex Simpson. Cyclic arithmetic is equivalent to peano arithmetic. In Javier Esparza  
2 and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation*  
3 *Structures - 20th International Conference, FOSSACS 2017, Held as Part of the*  
4 *European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala,*  
5 *Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer*  
6 *Science*, pages 283–300, 2017. doi: 10.1007/978-3-662-54458-7\_17. URL [https://doi.org/10.1007/978-3-662-54458-7\\_17](https://doi.org/10.1007/978-3-662-54458-7_17).
- 8 A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem  
9 for büchi automata with applications to temporal logic. *Theor. Comput. Sci.*, 49:  
10 217–237, 1987. doi: 10.1016/0304-3975(87)90008-9. URL [https://doi.org/10.1016/](https://doi.org/10.1016/0304-3975(87)90008-9)  
11 [0304-3975\(87\)90008-9](https://doi.org/10.1016/0304-3975(87)90008-9).
- 12 Colin Stirling. A tableau proof system with names for modal mu-calculus. In Andrei  
13 Voronkov and Margarita V. Korovina, editors, *HOWARD-60: A Festschrift on the*  
14 *Occasion of Howard Barringer's 60th Birthday*, volume 42 of *EPiC Series in Computing*,  
15 pages 306–318. EasyChair, 2014. URL [http://www.easychair.org/publications/](http://www.easychair.org/publications/?page=1932281032)  
16 [?page=1932281032](http://www.easychair.org/publications/?page=1932281032).
- 17 Gadi Tellez and James Brotherston. Automatically verifying temporal properties of  
18 pointer programs with cyclic proof. In Leonardo de Moura, editor, *Automated Deduction*  
19 *- CADE 26 - 26th International Conference on Automated Deduction, Gothenburg,*  
20 *Sweden, August 6-11, 2017, Proceedings*, volume 10395 of *Lecture Notes in Computer*  
21 *Science*, pages 491–508. Springer, 2017. doi: 10.1007/978-3-319-63046-5\_30. URL  
22 [https://doi.org/10.1007/978-3-319-63046-5\\_30](https://doi.org/10.1007/978-3-319-63046-5_30).
- 23 Moshe Y. Vardi. A temporal fixpoint calculus. In Jeanne Ferrante and P. Mager,  
24 editors, *Conference Record of the Fifteenth Annual ACM Symposium on Principles*  
25 *of Programming Languages, San Diego, California, USA, January 10-13, 1988*, pages  
26 250–259. ACM Press, 1988. doi: 10.1145/73560.73582. URL [https://doi.org/10.](https://doi.org/10.1145/73560.73582)  
27 [1145/73560.73582](https://doi.org/10.1145/73560.73582).
- 28 Igor Walukiewicz. On completeness of the mu-calculus. In *Proceedings of the Eighth*  
29 *Annual Symposium on Logic in Computer Science (LICS '93), Montreal, Canada, June*  
30 *19-23, 1993*, pages 136–146. IEEE Computer Society, 1993. doi: 10.1109/LICS.1993.  
31 287593. URL <https://doi.org/10.1109/LICS.1993.287593>.
- 32 Igor Walukiewicz. Completeness of kozen's axiomatisation of the propositional mu-  
33 calculus. In *Proceedings, 10th Annual IEEE Symposium on Logic in Computer Science,*  
34 *San Diego, California, USA, June 26-29, 1995*, pages 14–24. IEEE Computer Society,  
35 1995. doi: 10.1109/LICS.1995.523240. URL [https://doi.org/10.1109/LICS.1995.](https://doi.org/10.1109/LICS.1995.523240)  
36 [523240](https://doi.org/10.1109/LICS.1995.523240).