

Exploring the Relation between Intuitionistic BI and Boolean BI: An unexpected Embedding

DOMINIQUE LARCHEY-WENDLING^b and DIDIER GALMICHE[†]

LORIA – CNRS^b – Université Henri Poincaré[†]
Campus Scientifique, BP 239,
54 506 Vandœuvre-lès-Nancy, France
Email: {larchey,galmiche}@loria.fr

Received 22 July 2008; Revised 1 December 2008

The logic of Bunched Implications, through its intuitionistic version (BI) as well as one of its classical versions called Boolean BI (BBI), serves as a logical basis to spatial or separation logic frameworks. In BI, the logical implication is interpreted intuitionistically whereas it is generally interpreted classically in spatial or separation logics like in BBI. In this paper, we aim at giving some new insights w.r.t. the semantic relations between BI and BBI. Then we propose a sound and complete syntactic constraints based framework for Kripke semantics of both BI and BBI, a sound labelled tableau proof system for BBI, and a representation theorem relating the syntactic models of BI to those of BBI. Finally we deduce, as main and unexpected result, a sound and faithful embedding of BI into BBI.

1. Introduction

Spatial logics for dynamic processes, static spatial logics for trees or processes, context logic, separation logic, resource and processes logic share a common core language for expressing logical properties that is the language of BI, the Logic of Bunched Implications (Pym 1999; Pym 2002) mixing multiplicative connectives $*$, \multimap with additive connectives \wedge , \vee , \rightarrow and its Kripke sharing interpretation of the multiplicative connectives:

$$m \Vdash A * B \quad \text{iff} \quad \text{there exist } a, b \text{ such that } a \circ b \triangleright m \text{ and } a \Vdash A \text{ and } b \Vdash B$$

The ternary relation $-\circ-\triangleright-$ has different interpretations depending on various semantic frameworks: process composition/interaction (for \circ) and structural congruence (for \triangleright) in spatial logics (Cardelli and Gordon 2000; Caires and Lozes 2006), finite edge-labelled tree/process composition and structural equivalence for static spatial logics (Calcagno *et al.* 2005; Lozes 2004), contexts composition and structural equivalence for context logic (Calcagno *et al.* 2007), disjoint heap union and equality (or inclusion) for separation

logic (Ishtiaq and O’Hearn 2001) denoted **SL** here, and the product of resource and process composition for **SCR**P/**MBI** (Pym and Tofts 2006).

Even if they are based on the same language together with the sharing interpretation, these logics differ because the underlying models do not have the same properties: the set of valid formulae differ from one logic to the other, some have decidable model-checking and others not, etc. Moreover, nearly all of these logics use a classically or pointwise defined Kripke semantics for the additive implication \rightarrow , thereby favoring Boolean **BI** (**BBI**) whereas the logic **BI** has an intuitionistically defined additive implication. It should be noted however that there is an attempt to model intuitionistic implication in (Ishtiaq and O’Hearn 2001) but the authors quickly dismiss it as a particular case of the classical implication: the reason for this could be viewed as a rather restrictive choice for the order relation, map inclusion in this case. In this article, we aim at giving some new insights w.r.t. the relation between **BI** and **BBI** in the general case.

The model theoretic properties of these logics, based on **BBI**, have been widely studied: decidability (Calcagno *et al.* 2005) and undecidability (Caires and Lozes 2006), quantifier and adjunct elimination (Lozes 2004), and expressivity (Brochenin *et al.* 2008). However, even if the proof-theory of **BI** has been extensively explored with natural deduction and sequent style proof systems (Pym 2002) and labelled tableau proof systems (Galmiche and Méry 2003; Galmiche *et al.* 2005), the proof-theory of **BBI** is either missing or heavily based on model-checking methods like in (Calcagno *et al.* 2005).

The relations between **BI** and **BBI** are often misunderstood. Whereas classical logic **CL** (the additive fragment of **BBI**) can be faithfully embedded into intuitionistic logic **IL** (the additive fragment of **BI**) by the Gödel translation for example, this result has no known extension when linear operators are added. Moreover, in this article we show that the reverse is true: it is possible to faithfully embed **BI** into **BBI**. This result suggests that proof and counter-model search in **BBI** is certainly not easier than in **BI** and might in fact be much more difficult than in **BI**.

Our approach to **BI** and **BBI** and their relations is not model-oriented. We aim at studying the formulae of **BI**/**BBI** which hold in all of these particular models, propose proof-systems to prove or refute these formulae and compare the provability relation of **BI** and **BBI**. We consider **BI** and **BBI** defined by the abstract Kripke semantics, namely *partially ordered partial monoids* for **BI** and *partial monoids* for **BBI**. We use the models generated by the syntactic constraints occurring in tableau proof-search. Soundness and completeness of our tableau systems ensure that the sub-class of models generated by syntactic constraints is complete w.r.t. the abstract Kripke semantics and thus, these syntactic constraints and their solutions grasp the semantic properties of these logics.

In this article we provide the following results: a sound and complete syntactic constraints based framework for Kripke semantics of both **BI** and **BBI**, a sound labelled tableau proof system for **BBI**, a representation theorem linking the syntactic models of **BI** to those of **BBI** and as a consequence, a faithful embedding of **BI** into **BBI**. We also discuss some expressivity properties of **BI** that can be deduced from our results.

As for the potential consequences and later developments of this work (more fully described in the conclusion of this article), we list the following items: a sound and complete proof and counter-model search method for **BI** based on partial monoidal constraints as

opposed to the existing resource graph method (Galmiche *et al.* 2005); a concrete and complete class of separation logic style models for BI based on the distinction between observable and unobservable resources; hopefully a characterization of the full class of BBI-generated constraints and explicit forms for constraint extensions; more expressivity properties for BBI as well as for BI.

We now detail the contents of sections leading to the result that the function $G \mapsto (I \wedge H) \rightarrow G^\circ$ constitutes a sound and faithful embedding of BI into BBI. Here, the BBI-formula G° is an image of the BI-formula G defined by (linear) structural induction using two spare logical variables L and K . H is some given fixed BBI-formula only depending on the logical variables L and K . I is the multiplicative unit of BBI.

In Section 2, we recall the monoidal Kripke semantics of BI and BBI. We point out different semantic frameworks for interpreting the monoidal relation like spatial logics, separation logics or abstract monoidal Kripke semantics. We also stress the difference between the intuitionistic and classical interpretation of the additive implication \rightarrow , and the properties required for those models.

In Section 3, we describe a common framework for dealing with the Kripke semantics of BI and BBI. This framework is based on particular binary relations between words (which are in fact multisets in this article) expressed by sets of constraints of the form $m \multimap n$ where m and n are two words. An atomic constraint $m \multimap n$ is the syntactic expression of a semantic relation between the words/labels m and n that should hold in all the interpretations of m and n that satisfy the constraint $m \multimap n$. From a finite or infinite set of (atomic) constraints, we generate particular relations by closure. These particular relations are themselves the “least” models of the syntactic constraints from which they originate. These (closed) relations are called *partial monoidal orders* (PMOs) for BI and *partial monoidal equivalences* (PMEs) for BBI. They are characterized as being closed under some particular deduction rules. We introduce a Kripke interpretation within this PMO/PME framework and prove that the corresponding semantics is equivalent to the abstract Kripke semantics of BI and BBI respectively: up to some quotient by an equivalence relation, the PMO \sqsubseteq is in fact a partially ordered partial monoid and the PME \sim , a partial monoid. Then, we show how to build PMOs and PME by closure from arbitrary sets of constraints and derive some properties linking sets of constraints and their closures, like for example a compactness property or how the constraints involving the empty word ϵ behave in the closure.[†]

In Section 4, we present a link between PMOs (models of BI) and PME (models of BBI). Indeed, we describe a map $\sim \mapsto \sqsubseteq_{\sim}^{L,K}$ which associates a PMO $\sqsubseteq_{\sim}^{L,K}$ to any PME \sim , given some alphabets L and K . The idea of the map is that the words of L^* are Kripke-interpreted in both BI and BBI, whereas the words of K^* are only Kripke-interpreted in BBI. Thus we say that the words of K^* are unobservable by BI. The relation $m \sqsubseteq_{\sim}^{L,K} n$ holds whenever $m, n \in L^*$ belong to the observable words of L^* and are equivalent up to some unobservable word $\delta \in K^*$, i.e. $\delta m \sim n$ holds. We show how this idea extends

[†] The empty word ϵ plays a particular role because it is the only word which can be squared or erased a priori.

the intuitionistic interpretation of the implication \rightarrow by heap inclusion in intuitionistic Separation Logic SL (Ishtiaq and O'Hearn 2001). To any formula G of BI, we associate a formula G° of BBI such that the Kripke semantics of G in the model defined by $\sqsubseteq_{\sim}^{L,K}$ is equivalent to the Kripke semantics of G° in the model defined by \sim .

In Section 5, we introduce the labels and constraints based tableau proof system TBI which is sound and complete for BI. We also define a tableau proof system for BBI called TBBI and prove its soundness w.r.t. BBI (completeness also holds but for reasons explained later, this result is not proved in this article). Elementary PMOs are generated from finite sequences of constraints of the form $ab \vdash m$, $am \vdash b$, $m \vdash b$ or $\epsilon \vdash m$ where a and b are new letters and m is already defined by previous constraints. The sequence can be infinite for simple PMOs. We prove that the constraints occurring in TBI-tableaux are elementary PMOs for finite tableaux and simple PMOs for infinite tableaux branches and thus simple PMOs form a complete sub-class of models for BI. Hence every invalid formula of BI has a counter-model which is a simple PMO.

In Section 6, we define the notions of elementary and simple PME as generated by sequences of constraints of the form $ab \vdash m$, $am \vdash b$, $m \vdash b$ or $\epsilon \vdash b$ where a and b are new letters and m is already defined. We show that the map $\sim \mapsto \sqsubseteq_{\sim}^{L,K}$ is surjective on simple PMOs, its “reverse map” being described as an algorithm transforming a simple PMO \sqsubseteq into a simple PME \sim such that $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$. The validity of this algorithm can be considered as the main technical result of the article and is based on the notion of elementary representation which is basically a PMO/PME pair (\sqsubseteq, \sim) that verifies some specific conditions including $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$. The proof of the validity of the algorithm requires to establish some properties about free PME extensions like for example the extension of the PME \sim with the constraint $ab \vdash m$ where a and b are new letters and m is already defined. We also need to prove that simple PMEs have no square, i.e. for a simple PME \sim , the relation $mm \sim mm$ only holds when $m \sim \epsilon$.

In Section 7, we describe a tableau transformation algorithm of a TBI-proof of some BI-formula G into a TBBI-proof of the formula $(I \wedge H) \rightarrow G^\circ$, hence establishing that the map $G \mapsto (I \wedge H) \rightarrow G^\circ$ is a sound embedding of BI into BBI. As tableau proofs proceed by branch expansion, we show how to map any TBI-branch expansion into a combination of TBBI-branch expansions. The soundness of the transformation is based on the properties of elementary representations. Starting from a closed TBI-tableau (i.e. a proof) for G , the obtained TBBI-tableau for $(I \wedge H) \rightarrow G^\circ$ is not necessarily closed, but it is pseudo-closed and we later show that pseudo-closed tableaux can be expanded into closed tableaux in TBBI.

In Section 8, we show that the formula $(I \wedge H) \rightarrow G^\circ$ is BBI-invalid whenever G is BI-invalid by counter-model transformation. The result is based on the fact that if a BI-formula G is invalid then it has a BI-counter-model based on a simple PMO \sqsubseteq , the class of simple PMOs being complete for BI. As \sqsubseteq is a simple PMO, there exists a PME \sim such that $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ and we prove that \sim provides a BBI-counter-model to $(I \wedge H) \rightarrow G^\circ$. We present this counter-model transformation on the example of the intuitionistically invalid formula $X \vee (X \rightarrow \perp)$ (excluded middle).

In Section 9, we introduce some basic applications of our results to the expressivity of BI. For example, the property $\epsilon \sqsubseteq m$ is trivially expressed by the logical constant I . On

the contrary, neither the property $m \sqsubseteq \epsilon$ nor the property $mm \sqsubseteq mm$ can be expressed by formulae of BI.

2. Sharing interpretation and monoidal Kripke semantics for BI and BBI

The logics BI and BBI are syntactically defined by the following grammar (Pym 2002) where Var is a set of propositional variables and X ranges over Var :

$$\begin{aligned} \text{BI} : \quad & A, B ::= X \mid \perp \mid \top \mid A \wedge B \mid A \vee B \mid A \rightarrow B \mid \mathbf{I} \mid A * B \mid A \multimap B \\ \text{BBI} : \quad & A, B ::= X \mid \perp \mid \top \mid A \wedge B \mid A \vee B \mid A \rightarrow B \mid \neg A \mid \mathbf{I} \mid A * B \mid A \multimap B \end{aligned}$$

2.1. The monoidal Kripke semantics of BI and BBI

Before we introduce the Kripke semantics of BI and BBI, we recall the general semantic framework under which the Kripke interpretation is going to be defined.

Definition 2.1. A *partial monoid* is a triple $(\mathcal{M}, \circ, \mathbf{e})$ where $\mathbf{e} \in \mathcal{M}$ and $\circ : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$ is a partial map for which the following conditions hold:

1. $\forall a \in \mathcal{M}, \mathbf{e} \circ a \downarrow \wedge \mathbf{e} \circ a = a$ (identity)
2. $\forall a, b \in \mathcal{M}, a \circ b \downarrow \Rightarrow b \circ a \downarrow \wedge a \circ b = b \circ a$ (commutativity)
3. $\forall a, b, c \in \mathcal{M}, a \circ (b \circ c) \downarrow \Rightarrow (a \circ b) \circ c \downarrow \wedge a \circ (b \circ c) = (a \circ b) \circ c$ (associativity)

We write $x \circ y \downarrow$ when the composition of x and y by \circ is defined. Note that $a \circ (b \circ c)$ can only be defined if $b \circ c$ is itself defined, hence $a \circ (b \circ c) \downarrow$ implies $b \circ c \downarrow$. We also assume the reader does not confuse the meta-logical connectives \Rightarrow , \wedge and \vee with the logical connectives of BI/BBI even though the conjunction and disjunction have the same denotation.

A binary relation $\triangleright \subseteq \mathcal{M} \times \mathcal{M}$ is a *partial order* if it is reflexive, antisymmetric and transitive. To give a Kripke interpretation to the formulae of BI (resp. BBI), we start from a given structure $(\mathcal{M}, \circ, \mathbf{e}, \triangleright)$ where $(\mathcal{M}, \circ, \mathbf{e})$ is a partial monoid of *resources*, \triangleright is a partial order on \mathcal{M} (resp. the identity relation $\triangleright \equiv =$ on \mathcal{M}) such that composition is *monotonic*:

$$\forall k, x, y \in \mathcal{M}, (k \circ y \downarrow \wedge x \triangleright y) \Rightarrow (k \circ x \downarrow \wedge k \circ x \triangleright k \circ y)$$

The structure $(\mathcal{M}, \circ, \mathbf{e}, \triangleright)$ is thus a *partially ordered partial monoid* for the case of BI formulae and a *partial monoid* for the case of BBI formulae. The Kripke interpretation on the set of logical variables Var is given by a *forcing relation* $\Vdash \subseteq \mathcal{M} \times \text{Var}$ which verifies the *monotonicity condition*:

$$\forall X \in \text{Var}, \forall m, n \in \mathcal{M}, (m \triangleright n \wedge m \Vdash X) \Rightarrow n \Vdash X$$

For BBI, both the monotonicity of composition and the monotonicity condition trivially hold because the order \triangleright is the identity $=$.

The Kripke interpretation is inductively extended to the compound formulae of BI (resp. BBI) by the equations of Figure 1. We may write $\Vdash_{\circ, \triangleright}$ to denote this forcing relation extended to the whole BI (resp. BBI).

$m \Vdash \bot$	iff	$e \triangleright m$	$m \Vdash \neg A$	iff	$m \not\Vdash A$
$m \Vdash \perp$	iff	never	$m \Vdash A \vee B$	iff	$m \Vdash A$ or $m \Vdash B$
$m \Vdash \top$	iff	always	$m \Vdash A \wedge B$	iff	$m \Vdash A$ and $m \Vdash B$
$m \Vdash A \rightarrow B$	iff	$\forall x \in \mathcal{M}, (m \triangleright x \text{ and } x \Vdash A) \Rightarrow x \Vdash B$			
$m \Vdash A * B$	iff	$\exists x, y \in \mathcal{M}, x \circ y \downarrow \text{ and } x \circ y \triangleright m \text{ and } x \Vdash A \text{ and } y \Vdash B$			
$m \Vdash A \multimap B$	iff	$\forall x, y \in \mathcal{M}, (x \circ m \downarrow \text{ and } x \circ m \triangleright y \text{ and } x \Vdash A) \Rightarrow y \Vdash B$			

Fig. 1. The Kripke semantics of BI and BBI.

Definition 2.2. Given a Kripke structure $(\mathcal{M}, \circ, e, \triangleright, \Vdash)$, a formula F of BI (resp. BBI) is *valid* in \mathcal{M} if $e \Vdash_{\circ, \triangleright} F$ (resp. $\forall m \in \mathcal{M}, m \Vdash_{\circ, \triangleright} F$) and we say that \mathcal{M} is a *BI-model* (resp. *BBI-model*) of F . If F is not valid, then it is *invalid* and \mathcal{M} is a *BI-counter-model* (resp. *BBI-counter-model*) if $e \not\Vdash_{\circ, \triangleright} F$ (resp. $m \not\Vdash_{\circ, \triangleright} F$ for some $m \in \mathcal{M}$).

Within this framework, it is possible to prove (see (Pym 2002) for example) that the monotonicity condition extends to any formula of BI (resp. BBI) and that the logical rules of the natural deduction proof theory of BI and BBI (Pym 2002) are sound with respect to this Kripke interpretation. Moreover, if \triangleright is symmetric (which is a property fulfilled when \triangleright is the identity relation as required for BBI) it is possible to show that the logical implication \rightarrow is in fact interpreted classically (or pointwise), i.e. $m \Vdash A \rightarrow B$ iff $m \Vdash A \Rightarrow m \Vdash B$. This classical Kripke semantics for the additive implication \rightarrow is one of the reasons why BI is sometimes called *intuitionistic* BI whereas BBI is called *Boolean* BI. Without the linear connectives $*$ and \multimap , the distinction between intuitionistic and classical Kripke interpretation for the connective \rightarrow gives rise to propositional intuitionistic logic IL and propositional classical logic CL which have relations but also have huge differences. On the one hand, it is possible to faithfully embed CL into IL using the well known Gödel translation. On the other hand, the decision of validity in IL is PSPACE-complete (Statman 1979) whereas it is coNP-complete for CL, leading to the very unlikely existence of some (low complexity) reverse faithful embedding of IL into CL. This article aims at establishing a new relation between BI and BBI which may naively be viewed as respectively IL and CL enriched with the linear connectives $*$ and \multimap . We claim that the embedding relation we establish tells us that this naive view is not very accurate.

2.2. Overview of various instances of the monoidal Kripke semantics

To present some of the different existing frameworks for interpretation of BI and BBI, we single out the sharing interpretation of the $*$ operator:

$$m \Vdash A * B \quad \text{iff} \quad \exists x, y \in \mathcal{M}, x \circ y \downarrow \wedge x \circ y \triangleright m \wedge x \Vdash A \wedge y \Vdash B$$

$A * B$ is forced at m if there is composition $a \circ b$ somehow related to m by $a \circ b \triangleright m$ such that A is forced at a and B is forced at b . So the semantics of $A * B$ depends on the particular interpretation we provide for the composition \circ and the relation \triangleright . The

language of BI/BBI gives thus rise to different logics (where universally valid formulae differ) depending on how (\circ, \triangleright) are interpreted:

- what we call BI is the logic defined by abstract Kripke models, i.e. \mathcal{M} can be any partially ordered partial monoid. There exists a proof system for this logic (Galmiche *et al.* 2005). Similarly, what we call the logic BBI is defined by partial monoids without further restrictions and we will present a proof system for it in Section 5;
- if we restrict models so that \mathcal{M} is the set of *heaps* where heaps are partially and finitely defined functions $\text{Loc} \rightarrow_{fn} \text{Val}$ mapping locations to values, and $\circ \equiv (\cdot)$ is the disjoint union of heaps (undefined when domains overlap), we obtain *Separation Logic* SL (Ishtiaq and O’Hearn 2001). The relation $\triangleright \equiv =$ is interpreted as identity in SL giving rise to models of BBI formulae. Starting from SL, if $\triangleright \equiv \subseteq$ is interpreted as partial map inclusion instead of identity, we obtain models of BI-formulae (Ishtiaq and O’Hearn 2001) and the corresponding logic is called intuitionistic SL;
- if \mathcal{M} is the set of *finite unordered resource trees* and \circ is the composition of resource trees, we obtain *resource tree logic* (Biri and Galmiche 2007) which can be viewed as models of BI or BBI.

The logics arising from these different interpretations are not necessarily identical: indeed, weakening $A * B \rightarrow A$ is universally valid in intuitionistic SL whereas it has an obvious counter-model in partially ordered partial monoids thus in the version of BI we favor in this article. The study of the faithful embeddings between some of these logics and some of their sub-logics has already provided results:

- the modal translation of intuitionistic SL into classical SL provides a faithful embedding. As suggested in (Ishtiaq and O’Hearn 2001), this embedding is not so surprising because map inclusion $\triangleright \equiv \subseteq$ is a very restrictive interpretation for the relation \triangleright ;
- a faithful embedding of the modal logic S4 and hence intuitionistic logic IL into BBI has also been established (Galmiche and Larchey 2006) where BBI is given a *non-deterministic* monoidal semantics. The same argument applies *as is* with partial monoidal semantics for BBI (instead of non-deterministic monoidal semantics), so that the embedding of S4 into BBI is also faithful with the (partial and deterministic) interpretation of BBI we favor in this article;
- the well known Gödel translation provides a faithful embedding of classical propositional logic CL into intuitionistic logic IL. However, to the knowledge of the authors, there is no extension of this translation providing a faithful embedding of BBI into BI.

In this article, we will establish the existence of a faithful embedding of BI into BBI in the context of their general abstract semantics, namely partially ordered partial monoids for BI and partial monoids for BBI. By faithful embedding, we mean a map that preserves both validity and invalidity. Even as a faithful embedding from intuitionistic SL into classical SL already exists (Ishtiaq and O’Hearn 2001), that embedding is based on a restrictive interpretation of \triangleright which arguably gives rise to a denatured BI where weakening is valid (i.e. the formula $A * B \rightarrow A$ is valid in intuitionistic SL). Of course, this is not the case in BI. So our faithful embedding of BI into BBI is a much more unexpected result.

3. A complete semantics for BI and BBI based on words and constraints

We will now introduce a framework of labels and constraints to establish an original semantic relation between the abstract models of BI and BBI. This framework is useful because it provides a unifying view of both the models of BI and BBI and the proofs in BI and BBI. The main idea here is that restricting monoidal composition to word combination does not alter validity in the Kripke interpretation of BI (resp. BBI).

3.1. Words, constraints, PMOs and PMEs

Let L be a (potentially infinite) alphabet of letters. We consider the set of words L^* where the order of letters is not taken into account, i.e. we consider *words as finite multisets of letters*. The composition of words is denoted *multiplicatively* and the *empty word* is denoted ϵ .

We denote $x \prec y$ when x is a *subword* of y , i.e. when there exists a word k such that $kx = y$. If $x \prec y$, there is only one k such that $xk = y$ and it is denoted y/x , hence $y = x(y/x)$. The (*carrier*) *alphabet of a word* m is the set of letters of which it is composed: $A_m = \{l \in L \mid l \prec m\}$. We may view the alphabet L or any of its subsets $X \subseteq L$ as a subset $X \subset L^*$, i.e. we identify letters and one-letter words.

Definition 3.1. Let L be an alphabet. A *constraint* is a ordered pair (m, n) of words in $L^* \times L^*$ denoted $m \multimap n$.

We represent *binary relations* $R \subseteq L^* \times L^*$ between words of L^* as *set of constraints* through the logical equivalence: $x R y$ iff $x \multimap y \in R$. We view constraints as syntactic objects whereas relations between words can either be viewed as syntactic or semantic. When $\mathcal{C} = \{\dots, x_i \multimap y_i, \dots\}$ represents a finite or infinite collection of individual constraints, it is viewed as a syntactic notion and we write $x \multimap y \in \mathcal{C}$ for example. When R represents a relation between words, it is viewed as a semantic notion and we rather write $x R y$. But the very nature of \mathcal{C} and R is the same, that of a set of constraints. So we will use both terminologies for the same objects throughout this article depending on whether we interpret them more as syntactic objects or semantic objects. We will consider particular sets of constraints closed under some deduction rules and their corresponding relations. Because closed relations/closed sets of constraints can themselves be viewed as models, they are most of the time considered as semantic objects rather than syntactic ones.

Definition 3.2. Let L be an alphabet. The *language* of a binary relation $R \subseteq L^* \times L^*$ denoted \mathcal{L}^R is defined by $\mathcal{L}^R = \{x \in L^* \mid \exists m, n \in L^* \text{ s.t. } xm R n \vee m R xn\}$. The *carrier alphabet* of R denoted A_R is defined by $A_R = \bigcup \{A_m \cup A_n \mid m R n\}$.

A word $m \in L^*$ is said to be *defined in* R if $m \in \mathcal{L}^R$ and is *undefined* otherwise. A letter $l \in L$ is *new to* R if $l \notin A_R$. The language \mathcal{L}^R is downward closed w.r.t. the subword order \prec . The inclusion $\mathcal{L}^R \subseteq A_R^*$ and the identity $A_R = \mathcal{L}^R \cap L$ hold. If R_1 and R_2 are two relations such that $R_1 \subseteq R_2$ then the inclusions $A_{R_1} \subseteq A_{R_2}$ and $\mathcal{L}^{R_1} \subseteq \mathcal{L}^{R_2}$ hold. Let us define the particular sets of constraints/relations we are interested in.

PMOs	PMEs	PMOs & PME	
$\frac{x \multimap y}{x \multimap x} \langle l \rangle$	$\frac{x \multimap y}{y \multimap x} \langle s \rangle$	$\frac{}{\epsilon \multimap \epsilon} \langle \epsilon \rangle$	$\frac{ky \multimap ky \quad x \multimap y}{kx \multimap ky} \langle c \rangle$
$\frac{x \multimap y}{y \multimap y} \langle r \rangle$		$\frac{xy \multimap xy}{x \multimap x} \langle d \rangle$	$\frac{x \multimap y \quad y \multimap z}{x \multimap z} \langle t \rangle$

Table 1. Rules for the definition of PMOs and PME.

Definition 3.3 (PMO/PME). Consider the rules of Table 1. A *partial monoidal order* (PMO) \sqsubseteq over the alphabet L is a binary relation $\sqsubseteq \subseteq L^* \times L^*$ which is closed under the rules $\langle \epsilon, l, r, d, c, t \rangle$. A *partial monoidal equivalence* (PME) \sim over the alphabet L is a binary relation $\sim \subseteq L^* \times L^*$ which is closed under the rules $\langle \epsilon, s, d, c, t \rangle$.

Proposition 3.4. Rules $\langle l \rangle$ and $\langle r \rangle$ can be derived from rules $\langle s \rangle$ and $\langle t \rangle$. Hence any PME is also a PMO.

Proof. We provide the two following deduction trees:

$$\begin{array}{c}
 \frac{x \multimap y}{y \multimap x} \langle s \rangle \\
 \frac{x \multimap y \quad y \multimap x}{x \multimap x} \langle t \rangle
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{x \multimap y}{y \multimap x} \langle s \rangle \\
 \frac{y \multimap x \quad x \multimap y}{y \multimap y} \langle t \rangle
 \end{array}$$

□

Let us informally discuss the meaning of rules of Table 1. Let R be either a PMO $R = \sqsubseteq$ or a PME $R = \sim$. By upcoming Proposition 3.6, a word m is defined in R if and only if $m R m$ holds. Thus, rule $\langle d \rangle$ ensures that subwords of defined words are defined. Rule $\langle \epsilon \rangle$ ensures that ϵ is always defined (even when nothing else is). Rules $\langle l, r \rangle$ ensure that words that are related to other words by R are defined. Rule $\langle t \rangle$ ensures that R is transitive and rule $\langle s \rangle$ that it is symmetric. Rule $\langle c \rangle$ states that word composition should be monotonic w.r.t. the relation R .

We provide some derived rules which will be more suitable for proving properties of PMOs/PMEs throughout this article.

Proposition 3.5. Rules $\langle p_l, p_r, e_l \rangle$ can be derived from rules $\langle l, r, d, c, t \rangle$ and rule $\langle e_r \rangle$ can be derived from rules $\langle s, d, c, t \rangle$. Hence PMOs and PME are closed under rules $\langle p_l, p_r, e_l \rangle$ and PME are closed under rule $\langle e_r \rangle$.

$$\frac{kx \multimap y}{x \multimap x} \langle p_l \rangle \quad \frac{x \multimap ky}{y \multimap y} \langle p_r \rangle \quad \frac{x \multimap y \quad yk \multimap m}{xk \multimap m} \langle e_l \rangle \quad \frac{x \multimap y \quad m \multimap yk}{m \multimap xk} \langle e_r \rangle$$

Proof. Rule $\langle p_l \rangle$ (resp. $\langle p_r \rangle$) is a trivial combination of rules $\langle l, d \rangle$ (resp. $\langle r, d \rangle$). For

rules $\langle e_l \rangle$ and $\langle e_r \rangle$, we provide the two deduction trees:

$$\begin{array}{c}
 \frac{yk \multimap m}{yk \multimap yk} \langle l \rangle \quad x \multimap y \\
 \hline
 \frac{\quad}{xk \multimap yk} \langle c \rangle \quad yk \multimap m \\
 \hline
 xk \multimap m \quad \langle t \rangle
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{m \multimap yk}{yk \multimap m} \langle s \rangle \\
 \hline
 x \multimap y \quad yk \multimap m \\
 \hline
 xk \multimap m \quad \langle e_l \rangle \\
 \hline
 \frac{\quad}{m \multimap xk} \langle s \rangle
 \end{array}$$

□

Rule $\langle p_l \rangle$ (resp. $\langle p_r \rangle$) is a left (resp. right) projection rule, a kind of generalized version of $\langle l \rangle$ (resp. $\langle r \rangle$). Rules $\langle e_l \rangle$ and $\langle e_r \rangle$ express the capacity to exchange R -related subwords inside R -relations, either on the left (for PMOs and PME) or on the right (only for PME).

Proposition 3.6. Whether R is a PMO ($R = \sqsubseteq$) or a PME ($R = \sim$) over L , the identities $\mathcal{L}^R = \{x \in L^* \mid x R x\}$ and $A_R = \{l \in L \mid l R l\}$ hold.

Proof. We prove the properties for $R = \sqsubseteq$. The same properties will then hold with a PME \sim because any PME is also a PMO. First, it is obvious that $\{x \in L^* \mid x \sqsubseteq x\} \subseteq \mathcal{L}^\sqsubseteq$. For the converse, if $xm \sqsubseteq n$ (resp. $m \sqsubseteq xn$) then $x \sqsubseteq x$ by rule $\langle p_l \rangle$ (resp. rule $\langle p_r \rangle$). Hence, $\mathcal{L}^\sqsubseteq \subseteq \{x \in L^* \mid x \sqsubseteq x\}$. As $A_\sqsubseteq = \mathcal{L}^\sqsubseteq \cap L$, we get $A_\sqsubseteq = \{l \in L \mid l \sqsubseteq l\}$. □

3.2. PMO/PME based Kripke semantics

We introduce a Kripke interpretation of BI and BBI formulae based on PMOs and PME. The framework for BI and BBI is thus common and this facilitates the building of both semantic and proof-theoretic bridges between those logics.

Definition 3.7. A BI-frame (resp. BBI-frame) is a triple (L, R, \Vdash) where L is an alphabet, R is a PMO (resp. PME) over L , and \Vdash is a forcing relation $\Vdash \subseteq \mathcal{L}^R \times \text{Var}$ which verifies the *monotonicity property*:

$$\forall X \in \text{Var}, \forall m, n \in \mathcal{L}^R, (m R n \wedge m \Vdash X) \Rightarrow n \Vdash X$$

We extend the forcing relation to $\Vdash_R \subseteq \mathcal{L}^R \times \text{BI}$ (resp. $\mathcal{L}^R \times \text{BBI}$) by induction on formulae:

$$\begin{array}{llll}
 m \Vdash_R \top & \text{iff} & \epsilon R m & m \Vdash_R \neg A & \text{iff} & m \not\Vdash_R A \\
 m \Vdash_R \perp & \text{iff} & \text{never} & m \Vdash_R A \vee B & \text{iff} & m \Vdash_R A \text{ or } m \Vdash_R B \\
 m \Vdash_R \top & \text{iff} & \text{always} & m \Vdash_R A \wedge B & \text{iff} & m \Vdash_R A \text{ and } m \Vdash_R B \\
 m \Vdash_R A \rightarrow B & \text{iff} & \forall x \in \mathcal{L}^R, (m R x \text{ and } x \Vdash_R A) \Rightarrow x \Vdash_R B & & & \\
 m \Vdash_R A * B & \text{iff} & \exists x, y \in \mathcal{L}^R, xy R m \text{ and } x \Vdash_R A \text{ and } y \Vdash_R B & & & \\
 m \Vdash_R A \multimap B & \text{iff} & \forall x, y \in \mathcal{L}^R, (xm R y \text{ and } x \Vdash_R A) \Rightarrow y \Vdash_R B & & &
 \end{array}$$

We may write \Vdash for \Vdash_R when the relation R is obvious from the context.

Proposition 3.8. If R is a PMO (resp. a PME) then the extended relation $\Vdash_R \subseteq \mathcal{L}^\sqsubseteq \times \text{BI}$ (resp. $\Vdash_R \subseteq \mathcal{L}^\sim \times \text{BBI}$) is monotonic.

Proof. Monotonicity holds when for any $F \in \mathbf{BI}$ (resp. $F \in \mathbf{BBI}$) and any $m, n \in \mathcal{L}^R$, the condition $(m R n \wedge m \Vdash_R F) \Rightarrow n \Vdash_R F$ holds. It is standard to prove monotonicity by induction on the formula F . When F is a logical variable, the monotonicity condition holds as a direct consequence of Definition 3.7. For the additive operators \perp , \top , \vee and \wedge , the induction step is trivial. For operators \mathbf{l} , \rightarrow , and $*$, the induction step involves the use of rule $\langle t \rangle$. For operator $\neg*$, the induction step involves the use of rule $\langle e_l \rangle$. For the Boolean negation \neg , the induction step involves the use of rule $\langle s \rangle$, but as the operator \neg only exists in \mathbf{BBI} , the relation R is thus a PME, hence is closed under rule $\langle s \rangle$. \square

When R is a PME, symmetry (rule $\langle s \rangle$) ensures that the additive implication \rightarrow is interpreted pointwise or classically.

Proposition 3.9. If $R = \sim$ is a PME then for any $m \in \mathcal{L}^\sim$, $m \Vdash_\sim A \rightarrow B$ if and only if $m \Vdash_\sim A \Rightarrow m \Vdash_\sim B$.

Proof. Here we just write \Vdash for \Vdash_\sim . As \sim is a reflexive relation when restricted to \mathcal{L}^\sim , the *only if* part is trivial. For the *if part*, we use monotonicity. Let us suppose $m \Vdash A \Rightarrow m \Vdash B$. Let $m \sim n$ and $n \Vdash A$. Then by rule $\langle s \rangle$, $n \sim m$, hence by monotonicity, $m \Vdash A$ hence $m \Vdash B$. By monotonicity again, $n \Vdash B$. So for any n such that $m \sim n$ we have $n \Vdash A \Rightarrow n \Vdash B$. Thus $m \Vdash A \rightarrow B$. \square

Definition 3.10. A formula $F \in \mathbf{BI}$ is *valid* in the BI-frame (L, \sqsubseteq, \Vdash) if the relation $\epsilon \Vdash_\sqsubseteq F$ holds. A formula $F \in \mathbf{BBI}$ is *valid* in the BBI-frame (L, \sim, \Vdash) if for every $m \in \mathcal{L}^\sim$ the relation $m \Vdash_\sim F$ holds.

As a complement, we briefly state the relation between $\epsilon \Vdash F$ and $\forall m \in \mathcal{L}^\sim m \Vdash F$ in BBI-frames.

Proposition 3.11. For any BBI-frame (L, \sim, \Vdash) and any formula $F \in \mathbf{BBI}$:

1. $\epsilon \Vdash F$ iff $\forall m \in \mathcal{L}^\sim m \Vdash \mathbf{l} \rightarrow F$;
2. $\forall m \in \mathcal{L}^\sim m \Vdash F$ iff $\epsilon \Vdash \top \neg* F$.

Proof. For property 1, the *if part* is trivial since $\epsilon \in \mathcal{L}^\sim$ and $\epsilon\epsilon \sim \epsilon$. For the *only if part*, we use Proposition 3.9. Let $m \in \mathcal{L}^\sim$ such that $m \Vdash \mathbf{l}$. Then $\epsilon \sim m$. As $\epsilon \Vdash F$, by monotonicity we obtain $m \Vdash F$. Hence, $m \Vdash \mathbf{l} \rightarrow F$. For property 2, the *only if part* is trivial whereas for the *if part*, just use the fact that $m\epsilon \sim m$ for any $m \in \mathcal{L}^\sim$. \square

Remarks: property 2 holds also for \mathbf{BI} but not the *only if part* of property 1. Property 2 is used in (Calcagno *et al.* 2005) to establish an equivalence between validity and satisfaction problems in the spatial logic for trees.

Theorem 3.12 (Completeness of PMOs w.r.t. BI). A formula F of \mathbf{BI} is valid in every partially ordered partial monoid Kripke structure if and only if it is valid in every BI-frame.

Proof. This is an obvious but tedious proof based on quotients by equivalence relations. We give a brief sketch of the proof but leave the details to the reader. Let us first prove that if a formula of \mathbf{BI} has a Kripke counter-model of the form of a partially ordered

partial monoid then it has a counter-model of the form of a BI-frame. Then we will prove the converse result.

Let us consider a partially ordered partial monoid Kripke structure $(\mathcal{M}, \circ, e, \triangleright, \Vdash_{\circ, \triangleright})$. We take the elements of \mathcal{M} as letters of the language $L = \mathcal{M}$. A word of k letters $m = m_1 \dots m_k \in L^*$ is (partially) mapped to an element $m_{\circ} = m_1 \circ \dots \circ m_k \in \mathcal{M}$. The fact that m_{\circ} is defined or its value does not depend on the order upon which we perform the compositions of the letters of m because of the associativity and commutativity axioms of partial monoids. By definition, for the empty word, ϵ_{\circ} is defined and $\epsilon_{\circ} = e$. The binary relation \sqsubseteq over L^* defined by $m \sqsubseteq n$ iff $m_{\circ} \downarrow \wedge n_{\circ} \downarrow \wedge m_{\circ} \triangleright n_{\circ}$ is thus a PMO over L . We define the BI-frame $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$ by $m \Vdash_{\sqsubseteq} X$ iff $m_{\circ} \downarrow \wedge m_{\circ} \Vdash_{\circ, \triangleright} X$. Then, by straightforward induction on BI formulae, it is possible to prove that $m \Vdash_{\sqsubseteq} F$ iff $m_{\circ} \downarrow \wedge m_{\circ} \Vdash_{\circ, \triangleright} F$. As $\epsilon_{\circ} = e$, if \mathcal{M} is a Kripke counter-model of the formula F (i.e. $e \not\Vdash_{\circ, \triangleright} F$) then $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$ is also counter-model of the F (i.e. $\epsilon \not\Vdash_{\sqsubseteq} F$).

For the converse, let us consider a BI-frame $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$. We define the partial equivalence relation on L^* by $\sim = \sqsubseteq \cap \sqsubseteq^{-1}$, i.e. $m \sim n$ iff $m \sqsubseteq n \wedge n \sqsubseteq m$. We insist on the fact that \sim is not necessarily reflexive. Then we define \mathcal{M} as the set of partial equivalence classes of L^* . Let $[m] = \{x \in L^* \mid m \sim x\}$ and $\mathcal{M} = \{[m] \mid m \in L^* \wedge [m] \neq \emptyset\}$. The unit e is the class $[\epsilon]$ which is not empty by rule $\langle \epsilon \rangle$. The partial composition \circ is defined by $[m] \circ [n] = [mn]$ and the partial order \triangleright is defined by $[m] \triangleright [n]$ iff $m \sqsubseteq n$. Then $(\mathcal{M}, \circ, e, \triangleright)$ is a partially ordered partial monoid. We define the Kripke structure $(\mathcal{M}, \circ, e, \triangleright, \Vdash_{\circ, \triangleright})$ by $[m] \Vdash_{\circ, \triangleright} X$ iff $m \Vdash_{\sqsubseteq} X$. By straightforward induction on BI formulae, it is possible to show that $[m] \Vdash_{\circ, \triangleright} F$ iff $m \in \mathcal{L}^{\sqsubseteq} \wedge m \Vdash_{\sqsubseteq} F$. Hence, if $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$ is a counter-model of the formula F (i.e. $\epsilon \not\Vdash_{\sqsubseteq} F$) then \mathcal{M} is a counter-model of F (i.e. $[\epsilon] \not\Vdash_{\circ, \triangleright} F$). \square

Theorem 3.13 (Completeness of PME's w.r.t. BBI). A formula F of BBI is valid in every partial monoid Kripke structure if and only if it is valid in every BBI-frame.

Proof. Straightforward adaptation of the previous proof to PME's and BBI. \square

According to these two theorems, we can define *universal validity* and *counter-models*. A BI-counter-model for $F \in \text{BI}$ is a BI-frame in which $\epsilon \not\Vdash_{\sqsubseteq} F$. A BI formula F is *universally valid* (or *BI-valid*) when it has no BI-counter-model. A BBI-counter-model for $F \in \text{BBI}$ is a BBI-frame in which there exists $m \in \mathcal{L}^{\sim}$ such that $m \not\Vdash_{\sim} F$. F is *universally valid* (or *BBI-valid*) when it has no BBI-counter-model.

3.3. Sets of constraints and other properties of PMOs/PME's

Defined by closure under some deduction rules, the classes of PMOs and PME's are thus closed under arbitrary intersection. Thus, given a binary relation R between words described by a set of constraints, there exists a least PMO (resp. PME) containing R . We are especially interested in PMOs/PME's generated by some finite or infinite set of constraints.

Definition 3.14. Let L be an alphabet and \mathcal{C} be a set of constraints over the alphabet L .[‡] The PMO *generated by* \mathcal{C} is the least PMO denoted $\sqsubseteq_{\mathcal{C}}$ such that the inclusion $\mathcal{C} \subseteq \sqsubseteq_{\mathcal{C}}$ holds between those two sets of constraints. We also denote by $\sim_{\mathcal{C}}$ the PME *generated by* \mathcal{C} which is the least PME such that $\mathcal{C} \subseteq \sim_{\mathcal{C}}$.

For example, the PMO \sqsubseteq_0 generated by the singleton constraint $\mathcal{C}_0 = \{\epsilon \dashv a\}$ is $\sqsubseteq_0 = \{\epsilon \dashv \epsilon, \epsilon \dashv a, a \dashv a\}$ whereas the PMO \sqsubseteq_1 generated by the singleton constraint $\mathcal{C}_1 = \{a \dashv \epsilon\}$ is $\sqsubseteq_1 = \{a^i \dashv a^j \mid i \geq j\}$. The PME generated by the singleton constraint $\mathcal{C}_0 = \{\epsilon \dashv a\}$ is $\sim_0 = \{a^i \dashv a^j \mid i, j \in \mathbb{N}\}$. Proofs are left to the reader. By rule $\langle s \rangle$, \sim_0 is also the PME generated by the singleton constraint $\mathcal{C}_1 = \{a \dashv \epsilon\}$. Obviously, $\sqsubseteq_0 \subset \sim_0$ and $\sqsubseteq_1 \subset \sim_0$ and the inclusion is strict.

Considering two sets of constraints $\mathcal{C} \subseteq \mathcal{D}$, we have $\sqsubseteq_{\mathcal{C}} \subseteq \sqsubseteq_{\mathcal{D}}$ and $\sim_{\mathcal{C}} \subseteq \sim_{\mathcal{D}}$. Also $\sqsubseteq_{\mathcal{C}} \subseteq \sim_{\mathcal{C}}$ because $\sim_{\mathcal{C}}$ contains \mathcal{C} as a subset and, being a PME, is also a PMO.

Definition 3.15. Let $R = \sqsubseteq$ (resp. $R = \sim$) be a PMO (resp. PME). Let \mathcal{C} be a set of constraints. We denote by $R + \mathcal{C}$ the *extension* of R by the constraints of \mathcal{C} which is the least PMO (resp. PME) containing $R \cup \mathcal{C}$.

Let us insist on the fact that the meaning of the extension $R + \mathcal{C}$ depends whether R is viewed as a PMO or a PME, especially since PMEs are also PMOs. Let R be a PMO or a PME over the alphabet L and \mathcal{C}_1 and \mathcal{C}_2 be two sets of constraints over L then $(R + \mathcal{C}_1) + \mathcal{C}_2 = (R + \mathcal{C}_2) + \mathcal{C}_1 = R + (\mathcal{C}_1 \cup \mathcal{C}_2)$. These identities hold both for PMO extensions and PME extensions and their proofs are trivial and left to the reader. Moreover, for any $m, n \in L^*$, the relation $m R n$ holds if and only if the identity $R + \{m \dashv n\} = R$ holds; in particular $R + \{\epsilon \dashv \epsilon\} = R$.

Proposition 3.16. If \mathcal{C} is a set of constraints over L then the inclusion $\sqsubseteq_{\mathcal{C}} \subseteq \sim_{\mathcal{C}}$ and the identity $A_{\mathcal{C}} = A_{\sqsubseteq_{\mathcal{C}}} = A_{\sim_{\mathcal{C}}}$ hold.

Proof. For the first property, $\sim_{\mathcal{C}}$ is a PME containing \mathcal{C} hence also a PMO containing \mathcal{C} . For the second property, as $\mathcal{C} \subseteq \sqsubseteq_{\mathcal{C}} \subseteq \sim_{\mathcal{C}}$ as relations, we derive $A_{\mathcal{C}} \subseteq A_{\sqsubseteq_{\mathcal{C}}} \subseteq A_{\sim_{\mathcal{C}}}$. Thus it is sufficient to prove that $A_{\sim_{\mathcal{C}}} \subseteq A_{\mathcal{C}}$. Let \sim be defined by $m \sim n$ iff $m, n \in A_{\mathcal{C}}^*$. Then, $A_{\sim} = A_{\mathcal{C}}$, $\mathcal{C} \subseteq \sim$ and \sim is a PME. Hence $\sim_{\mathcal{C}} \subseteq \sim$ and thus $A_{\sim_{\mathcal{C}}} \subseteq A_{\sim} = A_{\mathcal{C}}$. \square

Proposition 3.17 (Compactness). Let \mathcal{C} be a possibly infinite set of constraints over the alphabet L . Let $m, n \in L^*$ be such that $m \sqsubseteq_{\mathcal{C}} n$ (resp. $m \sim_{\mathcal{C}} n$) holds. There exists a finite subset $\mathcal{C}_f \subseteq \mathcal{C}$ such that $m \sqsubseteq_{\mathcal{C}_f} n$ (resp. $m \sim_{\mathcal{C}_f} n$) holds.

Proof. We do the proof for PMOs. It is similar for PMEs. Let \mathcal{C} be a set of constraints. Let the relation R_f be (the finite approximation of $\sqsubseteq_{\mathcal{C}}$) defined by $m R_f n$ if and only if there exists $\mathcal{D} \subseteq \mathcal{C}$ such that \mathcal{D} is finite and $m \sqsubseteq_{\mathcal{D}} n$. Obviously $\mathcal{C} \subseteq R_f \subseteq \sqsubseteq_{\mathcal{C}}$. If we show that R_f is a PMO then we get $R_f = \sqsubseteq_{\mathcal{C}}$ which proves the proposition.

So let us prove that R_f is a PMO. For rule $\langle \epsilon \rangle$, we have $\epsilon \sqsubseteq_{\emptyset} \epsilon$ and \emptyset is a finite subset of \mathcal{C} , hence $\epsilon R_f \epsilon$. Thus R_f is closed under rule $\langle \epsilon \rangle$. If $x R_f y$ then, for some finite subset

[‡] As \mathcal{C} is also a relation, the alphabet of \mathcal{C} is $A_{\mathcal{C}} = \bigcup \{A_m \cup A_n \mid m \dashv n \in \mathcal{C}\}$, i.e. the set of letters which occur in at least one of the constraints of \mathcal{C} .

\mathcal{D} of \mathcal{C} we have $x \sqsubseteq_{\mathcal{D}} y$. Then, as $\sqsubseteq_{\mathcal{D}}$ is a PMO, we have $x \sqsubseteq_{\mathcal{D}} x$ by rule $\langle l \rangle$ and $y \sqsubseteq_{\mathcal{D}} y$ by rule $\langle r \rangle$. Hence $x R_f x$ and $y R_f y$ and R_f is closed under rules $\langle l \rangle$ and $\langle r \rangle$. The same reasoning applies to the unary rule $\langle d \rangle$. We consider the binary rule $\langle t \rangle$. If $x R_f y$ and $y R_f z$ then there exist \mathcal{D} and \mathcal{E} , two finite subsets of \mathcal{C} such that $x \sqsubseteq_{\mathcal{D}} y$ and $y \sqsubseteq_{\mathcal{E}} z$. Let $\mathcal{F} = \mathcal{D} \cup \mathcal{E}$. Then \mathcal{F} is a finite subset of \mathcal{C} and $x \sqsubseteq_{\mathcal{F}} y$ and $y \sqsubseteq_{\mathcal{F}} z$, because $\sqsubseteq_{\mathcal{D}} \subseteq \sqsubseteq_{\mathcal{F}}$ and $\sqsubseteq_{\mathcal{E}} \subseteq \sqsubseteq_{\mathcal{F}}$. Thus $x \sqsubseteq_{\mathcal{F}} z$ as $\sqsubseteq_{\mathcal{F}}$ is closed under rule $\langle t \rangle$ and then $x R_f z$. Hence the relation R_f is closed under rule $\langle t \rangle$. The same reasoning applies to the other binary rule $\langle c \rangle$. Hence, the relation R_f is indeed a PMO. \square

This *compactness property* is not related to the particular nature of rules defining PMOs or PME's but solely to the fact that these rules only have a *finite number of premises*. Apart generating PMOs/PME's from sets of constraints, we provide another basic way to build them from sub-alphabets and derive a interesting property from it.

Proposition 3.18. Let L be an alphabet and $X \subseteq L$ be a subset of L . Then:

1. \sqsubseteq^X is a PMO over L where \sqsubseteq^X is defined by $m \sqsubseteq^X n$ iff $(n \in X^* \Rightarrow m \in X^*)$;
2. \sim^X is a PME over L where \sim^X is defined by $m \sim^X n$ iff $(n \in X^* \Leftrightarrow m \in X^*)$.

Proof. The relation \sqsubseteq^X is reflexive because the meta-logical implication \Rightarrow of its definition is reflexive. So it is obviously closed under rules $\langle \epsilon, l, r, d \rangle$. It is also transitive because the meta-logical implication is transitive, hence \sqsubseteq^X is closed under rule $\langle t \rangle$.

As $mn \in X^*$ if and only if $m, n \in X^*$, \sqsubseteq^X is closed under rule $\langle c \rangle$: indeed, let $m, n \in L^*$ such that $m \sqsubseteq^X n$. If $kn \in X^*$ then $k, n \in X^*$. As $n \in X^*$ and $m \sqsubseteq^X n$, we deduce $m \in X^*$. Hence, $k, m \in X^*$ and so $km \in X^*$.

The inverse relation $(\sqsubseteq^X)^{-1}$ is also a PMO for identical reasons. Then $\sim^X = \sqsubseteq^X \cap (\sqsubseteq^X)^{-1}$ is a PME as intersection of two PMOs. The relation $R \cap R^{-1}$ is always symmetric hence closed under rule $\langle s \rangle$. So \sim^X is closed under rule $\langle s \rangle$, hence is a PME. \square

Proposition 3.19. Let \mathcal{C} be a set of constraints on the alphabet L :

1. if no constraint of \mathcal{C} is of the form $m \dashv \epsilon$ (with $m \neq \epsilon$), then for any $m \in L^*$, $m \sqsubseteq_{\mathcal{C}} \epsilon$ only if $m = \epsilon$;
2. if no constraint of \mathcal{C} is of the form $m \dashv \epsilon$ or $\epsilon \dashv m$ (with $m \neq \epsilon$), then for any $m \in L^*$, $m \sim_{\mathcal{C}} \epsilon$ only if $m = \epsilon$.

Proof. We use Proposition 3.18 in the particular case where $X = \emptyset$ is the empty sub-alphabet. For property 1, let us first prove that $\mathcal{C} \subseteq \sqsubseteq^{\emptyset}$. Let $m, n \in L^*$ be such that $m \dashv n \in \mathcal{C}$. Let us prove $m \sqsubseteq^{\emptyset} n$. As $\emptyset^* = \{\epsilon\}$, if $n \in \emptyset^*$ then $n = \epsilon$ and hence $m \dashv \epsilon \in \mathcal{C}$. By the hypothesis on \mathcal{C} , we must have $m = \epsilon$ and as a consequence $m \in \emptyset^*$. So $m \sqsubseteq^{\emptyset} n$ and we have proved that $\mathcal{C} \subseteq \sqsubseteq^{\emptyset}$. As $\sqsubseteq_{\mathcal{C}}$ is the least PMO containing \mathcal{C} and \sqsubseteq^{\emptyset} is a PMO by property 1 of Proposition 3.18, we have $\sqsubseteq_{\mathcal{C}} \subseteq \sqsubseteq^{\emptyset}$. Now let us consider $m \in L^*$ such that $m \sqsubseteq_{\mathcal{C}} \epsilon$. We deduce $m \sqsubseteq^{\emptyset} \epsilon$. As $\epsilon \in \emptyset^*$, we must have $m \in \emptyset^*$ by definition of \sqsubseteq^{\emptyset} . Thus we obtain $m = \epsilon$.

For property 2, the reasoning is similar: we prove the inclusion $\mathcal{C} \subseteq \sim^{\emptyset}$ from which we deduce $\sim_{\mathcal{C}} \subseteq \sim^{\emptyset}$. Thus no $m \in L^*$ such that $m \sim_{\mathcal{C}} \epsilon$ can exist unless $m = \epsilon$. \square

For the moment, we stop our investigation on the different methods to build PMOs and PME's. We will come back to it later when we will describe precisely how to compute “freely generated” PME's in Section 6.3. We have enough material to present the first building block of our embedding of BI into BBI.

4. Linking PMOs/PME's and the Kripke semantics of BI and BBI

We define the relation between PMOs and PME's which is at the core of the embedding of BI into BBI. First let us give an intuition of the design of this relation between PMOs and PME's.

The basic idea can be viewed as a variant of the embedding of intuitionistic SL into classical SL. Indeed, in (Ishtiaq and O'Hearn 2001), the order relation chosen to interpret implication intuitionistically is *graph inclusion* \subseteq between *heaps*:

$$m \Vdash A \rightarrow B \quad \text{iff} \quad \forall h' (h \subseteq h' \wedge h' \Vdash A) \Rightarrow h' \Vdash B$$

But as composition is the disjoint union of graphs, we have the following relation:

$$h \subseteq h' \quad \text{iff} \quad \exists g, g \cdot h = h'$$

Thus h is below h' if it is possible to compose h with something (g here), obtaining h' . Or stated otherwise, h' is identical to h up to some unspecified part (g here). We generalize this idea in the following way: we restrict the choice of the missing part g to a space of heaps which might be disjoint from the space of observable heaps, observable meaning observable through the Kripke semantics.

4.1. Building PMOs with PME's

In our words and constraints based semantics, heaps are abstracted by words. To distinguish observable words from potentially unobservable words, we divide the alphabet L' into two sub-alphabets L and K which may be disjoint, L^* representing observable words and K^* unobservable words.

Definition 4.1. Let \sim be a PME over L' and L, K be two subsets of L' , i.e. $L \cup K \subseteq L'$. We define the relation $\sqsubseteq_{\sim}^{L,K} \subseteq L^* \times L^*$ by:

$$m \sqsubseteq_{\sim}^{L,K} n \quad \text{iff} \quad \exists \delta \in K^*, \delta m \sim n$$

Thus, m is below n if m can be completed into n by some unobservable part δ . Then we prove that the relation $\exists \delta \in K^*, \delta m \sim n$ defines a PMO over L provided the relation \sim is a PME over L' . Obviously, if \sim and \sim' are two PME's over L' such that $\sim \subseteq \sim'$ then $\sqsubseteq_{\sim}^{L,K} \subseteq \sqsubseteq_{\sim'}^{L,K}$.

Lemma 4.2. If \sim is a PME over L' then the relation $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ is a PMO over L , and the identities $A_{\sqsubseteq} = A_{\sim} \cap L$ and $\mathcal{L}^{\sqsubseteq} = \mathcal{L}^{\sim} \cap L^*$ hold.

Proof. First, we remark that for any $m \in L^*$, $m \sqsubseteq m$ if and only if $m \sim m$: indeed, $\delta m \sim m$ implies $m \sim m$ by rule $\langle r \rangle$. Thus, by Proposition 3.6, $\mathcal{L}^{\sqsubseteq} = \mathcal{L}^{\sim} \cap L^*$ and $A_{\sqsubseteq} = A_{\sim} \cap L$. Let us prove that \sqsubseteq is a PMO.

As $m \sqsubseteq m$ iff $m \sim m$ and \sim is a PME, \sqsubseteq is obviously closed under rules $\langle \epsilon \rangle$ and $\langle d \rangle$. If $x \sqsubseteq y$ then $\delta x \sim y$ for some $\delta \in K^*$. Then $x \sim x$ by rule $\langle p_l \rangle$, and thus $x \sqsubseteq x$. Hence \sqsubseteq is closed under rule $\langle l \rangle$. With rule $\langle p_r \rangle$ for \sim , we also show that \sqsubseteq is closed under rule $\langle r \rangle$. Let us consider rule $\langle t \rangle$. If $x \sqsubseteq y$ and $y \sqsubseteq z$ then $\delta x \sim y$ and $\delta' y \sim z$ for some $\delta, \delta' \in K^*$. Then $\delta\delta' \in K^*$ and $\delta\delta'x \sim z$ by application of rule $\langle e_l \rangle$, hence $x \sqsubseteq z$. So \sqsubseteq is closed under rule $\langle t \rangle$. Let us consider rule $\langle c \rangle$. If $qy \sqsubseteq qy$ and $x \sqsubseteq y$ then $qy \sim qy$ and $\delta x \sim y$ for some $\delta \in K^*$. By rule $\langle c \rangle$, we get $\delta qx \sim qy$. Then we have $qx \sqsubseteq qy$. So \sqsubseteq is closed under rule $\langle c \rangle$. \square

Since we have a way to build PMOs starting from PMEs, several questions arise. For example, is this construction process general enough to represent any PMO, i.e. is the map $(\sim, L, K) \mapsto \sqsubseteq_{\sim}^{L,K}$ a surjective one? Is it semantically compatible with some embedding of BI into BBI, i.e. does it preserve Kripke semantics? Let us first answer the second question.

In the next two sections, we present a map $F \mapsto F^\circ$ from BI-formulae to BBI-formulae and show that this map preserves the Kripke semantics provided the PMO $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ comes from a PME \sim .

4.2. An intuitive description of the map of BI-formulae to BBI-formulae

Before we introduce the map, we single out two logical variables L and K which behave as the syntactical counterpart of the distinction between observable and unobservable words. We point out that we have intentionally chosen to give the two variables L and K the same name (i.e. same letters) as the sub-alphabets L and K which occur in the definition of $\sqsubseteq_{\sim}^{L,K}$, leaving the distinction in the choice of the font L/L and K/K . The link between the (semantic) set of observable words and the (syntactic) variable L is enforced by choosing \Vdash_{\sim} such that $x \in L^*$ iff $x \Vdash_{\sim} L$. The same holds for K/K : $x \in K^*$ iff $x \Vdash_{\sim} K$ holds for any word x .

Let us informally explain the idea of the encoding of BI into BBI and its link between observable and unobservable words. Let us suppose that the PMO \sqsubseteq is of the form $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ for some PME \sim . The Kripke interpretation of $m \Vdash_{\sqsubseteq} A * B$ is thus

$$\exists a, b \in L^*, ab \sqsubseteq m \wedge a \Vdash_{\sqsubseteq} A \wedge b \Vdash_{\sqsubseteq} B$$

Using $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$, we transform this formula into

$$\exists a, b \in L^*, \exists \delta \in K^*, \delta ab \sim m \wedge a \Vdash_{\sqsubseteq} A \wedge b \Vdash_{\sqsubseteq} B$$

As we have chosen to encode the set L^* with the logical variable L in the Kripke semantics \Vdash_{\sim} , then we can exchange $a \in L^*$ with $a \Vdash_{\sim} L$ (same for $b/L/L$ and $\delta/K/K$) and obtain

$$\exists \delta, a, b, \delta ab \sim m \wedge \delta \Vdash_{\sim} K \wedge a \Vdash_{\sim} L \wedge a \Vdash_{\sqsubseteq} A \wedge b \Vdash_{\sim} L \wedge b \Vdash_{\sqsubseteq} B$$

Now suppose (recursively) that there are two BBI-formulae A° and B° such that for any $x \in \mathcal{L}^\sqsubseteq$, $x \Vdash_{\sqsubseteq} A$ iff $x \Vdash_{\sim} A^\circ$ and $x \Vdash_{\sqsubseteq} B$ iff $x \Vdash_{\sim} B^\circ$. Exchanging $a \Vdash_{\sqsubseteq} A$ by $a \Vdash_{\sim} A^\circ$ and $b \Vdash_{\sqsubseteq} B$ by $b \Vdash_{\sim} B^\circ$ we obtain

$$\exists \delta, a, b, \delta ab \sim m \wedge \delta \Vdash_{\sim} K \wedge a \Vdash_{\sim} L \wedge a \Vdash_{\sim} A^\circ \wedge b \Vdash_{\sim} L \wedge b \Vdash_{\sim} B^\circ$$

or stated otherwise

$$m \Vdash_{\sim} K * (L \wedge A^\circ) * (L \wedge B^\circ)$$

We see that we have to coerce a and b to range over observable words in L^* by stating $a \Vdash_{\sim} L$ and $b \Vdash_{\sim} L$ (L^* are the words on which the BI formula $A*B$ is Kripke-interpreted) whereas we coerce δ to range over unobservable words in K^* by stating $\delta \Vdash_{\sim} K$.

4.3. Formal definition of the embedding map of BI-formulae to BBI-formulae

We formalize these ideas into the recursively defined map $F \mapsto F^\circ$. The formulae on which the map $(\cdot)^\circ$ is applied should not contain occurrences of either L or K to enforce the distinction between observable and unobservable words.

Definition 4.3 (Embedding map). Let L and K be two different spare logical variables in Var . Given $F \in \text{BI}$ containing neither L nor K , we define by induction on F the formula $F^\circ \in \text{BBI}$ as follows:

$$\begin{aligned} X^\circ &= K * X \text{ for } X \in \text{Var} \setminus \{L, K\} & I^\circ &= K * I & \perp^\circ &= \perp & \top^\circ &= \top \\ (A \oplus B)^\circ &= A^\circ \oplus B^\circ \text{ for } \oplus \in \{\wedge, \vee\} & (A * B)^\circ &= K * ((L \wedge A^\circ) * (L \wedge B^\circ)) \\ (A \rightarrow B)^\circ &= K \multimap ((L \wedge A^\circ) \rightarrow B^\circ) & (A \multimap B)^\circ &= (K * (L \wedge A^\circ)) \multimap (L \rightarrow B^\circ) \end{aligned}$$

In fact, the formula F° also belongs to the language of BI but it will be interpreted as a BBI-formula, not as a BI-formula. We now state and prove that the map $(\cdot)^\circ$ preserves the Kripke semantics provided the PMO is of the form $\sqsubseteq_{\sim}^{L,K}$.

Theorem 4.4. Let \sim be a PME over L' and $L \cup K \subseteq L'$. Let $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$ be a BI-frame on $\text{Var} \setminus \{L, K\}$ such that $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$. We define the relation $\Vdash_{\sim} \subseteq \mathcal{L}^{\sim} \times \text{Var}$, by

$$\begin{aligned} m \Vdash_{\sim} X &\text{ iff } \exists l \in L^*, l \sim m \wedge l \Vdash_{\sqsubseteq} X & \text{ for } X \in \text{Var} \setminus \{L, K\} \\ m \Vdash_{\sim} K &\text{ iff } \exists \delta \in K^*, \delta \sim m \\ m \Vdash_{\sim} L &\text{ iff } \exists l \in L^*, l \sim m \end{aligned}$$

Then $(L', \sim, \Vdash_{\sim})$ is a BBI-frame on Var and for any formula F of BI containing neither K nor L and any $m \in \mathcal{L}^{\sqsubseteq}$, the equivalence $m \Vdash_{\sqsubseteq} F$ iff $m \Vdash_{\sim} F^\circ$ holds.

Proof. We remind the reader that $\mathcal{L}^{\sqsubseteq} = \mathcal{L}^{\sim} \cap L^*$ (see Lemma 4.2). Let us first prove that $(L', \sim, \Vdash_{\sim})$ is a BBI-frame, i.e. that \Vdash_{\sim} is monotonic. Let $m, n \in \mathcal{L}^{\sim}$ such that $m \sim n$. If $X \in \text{Var} \setminus \{L, K\}$ and $m \Vdash_{\sim} X$ then there exists $l \in L^*$ such that $l \sim m$ and $l \Vdash_{\sqsubseteq} X$. By rule $\langle t \rangle$, we obtain $l \sim n$, hence $n \Vdash_{\sim} X$. If $m \Vdash_{\sim} K$ then there exists $\delta \in K^*$ such that $\delta \sim m$. Then $\delta \sim n$ hence $n \Vdash_{\sim} K$. If $m \Vdash_{\sim} L$ then there exists $l \in L^*$ such that $l \sim m$. Then $l \sim n$ hence $n \Vdash_{\sim} L$. So \Vdash_{\sim} is indeed monotonic.

We prove the equivalence of semantic interpretations by induction on F . The cases for F of the form \perp , \top , $A \wedge B$ or $A \vee B$ are trivial because the Kripke interpretations are obviously the same. We list the non-trivial cases:

$\boxed{F = X}$ with $X \in \text{Var} \setminus \{L, K\}$. Let $m \in \mathcal{L}^{\sqsubseteq}$. On the one hand, if $m \Vdash_{\sqsubseteq} X$ then, as $m \in L^*$ and $m \sim m$, we have $m \Vdash_{\sim} X$. As $\epsilon \Vdash_{\sim} K$ and $\epsilon m \sim m$, we obtain $m \Vdash_{\sim} K * X$, hence $m \Vdash_{\sim} X^\circ$. On the other hand, if $m \Vdash_{\sim} K * X$ then there exist x, y such that $xy \sim m$, $x \Vdash_{\sim} K$ and $y \Vdash_{\sim} X$. Then there exists $\delta \in K^*$ such that $\delta \sim x$ and there

exists $l \in L^*$ such that $l \sim y$ and $l \Vdash_{\sqsubseteq} X$. Then, by two applications of rule $\langle e_l \rangle$, $\delta l \sim m$, hence $l \sqsubseteq m$. As $l \Vdash_{\sqsubseteq} X$ and \Vdash_{\sqsubseteq} is monotonic, we obtain $m \Vdash_{\sqsubseteq} X$;

$\boxed{F = I}$ We have $m \Vdash_{\sim} K * I$ iff $m \Vdash_{\sim} K$ iff $\exists \delta \in K^* \delta \sim m$ iff $\epsilon \sqsubseteq m$ iff $m \Vdash_{\sqsubseteq} I$;

$\boxed{F = A \rightarrow B}$ Let us suppose $m \Vdash_{\sqsubseteq} A \rightarrow B$ with $m \in \mathcal{L}^{\sqsubseteq}$ and prove $m \Vdash_{\sim} K \multimap ((L \wedge A^\circ) \rightarrow B^\circ)$. Let k, c such that $km \sim c$, $k \Vdash_{\sim} K$ and $c \Vdash_{\sim} L \wedge A^\circ$. Let us prove $c \Vdash_{\sim} B^\circ$. From $k \Vdash_{\sim} K$, we obtain $\delta \in K^*$ such that $\delta \sim k$. From $c \Vdash_{\sim} L$, we get $l \in L^*$ such that $l \sim c$. By monotonicity of \Vdash_{\sim} and rule $\langle s \rangle$, from $c \Vdash_{\sim} A^\circ$, we deduce $l \Vdash_{\sim} A^\circ$, hence $l \Vdash_{\sqsubseteq} A$ by induction hypothesis. As $\delta m \sim l$, hence $m \sqsubseteq l$, and $m \Vdash_{\sqsubseteq} A \rightarrow B$, we deduce $l \Vdash_{\sqsubseteq} B$, hence again by induction hypothesis, $l \Vdash_{\sim} B^\circ$. As $l \sim c$, by monotonicity of \Vdash_{\sim} , we obtain $c \Vdash_{\sim} B^\circ$. So from $m \Vdash_{\sqsubseteq} A \rightarrow B$, we deduce $m \Vdash_{\sim} (A \rightarrow B)^\circ$.

Let us prove the converse implication. Let us suppose $m \Vdash_{\sim} K \multimap ((L \wedge A^\circ) \rightarrow B^\circ)$. Let $x \in \mathcal{L}^{\sqsubseteq}$ such that $m \sqsubseteq x$ and $x \Vdash_{\sqsubseteq} A$. Let us prove $x \Vdash_{\sqsubseteq} B$. From $x \Vdash_{\sqsubseteq} A$, we deduce $x \Vdash_{\sim} A^\circ$ by induction hypothesis. As $x \in L^*$ and $x \sim x$ ($x \in \mathcal{L}^{\sqsubseteq} \subseteq \mathcal{L}^\sim$), we deduce $x \Vdash_{\sim} L \wedge A^\circ$. As $\epsilon \Vdash_{\sim} K$ and $\epsilon x \sim x$, we deduce $x \Vdash_{\sim} B^\circ$, hence $x \Vdash_{\sqsubseteq} B$ by induction hypothesis. Hence $m \Vdash_{\sqsubseteq} A \rightarrow B$;

$\boxed{F = A * B}$ let us suppose $m \Vdash_{\sqsubseteq} A * B$ with $m \in \mathcal{L}^{\sqsubseteq}$. There exist $x, y \in \mathcal{L}^{\sqsubseteq}$ such that $xy \sqsubseteq m$, $x \Vdash_{\sqsubseteq} A$ and $y \Vdash_{\sqsubseteq} B$. As $x, y \in L^*$, by induction hypothesis, we obtain $x \Vdash_{\sim} L \wedge A^\circ$ and $y \Vdash_{\sim} L \wedge B^\circ$. Then, as $xy \sqsubseteq m$, there exists $\delta \in K^*$ such that $\delta xy \sim m$. Then $\delta \Vdash_{\sim} K$, hence $m \Vdash_{\sim} K * ((L \wedge A^\circ) * (L \wedge B^\circ))$.

On the other hand, let us suppose $m \Vdash_{\sim} K * ((L \wedge A^\circ) * (L \wedge B^\circ))$. Then, there exists k, a, b such that $kab \sim m$, $k \Vdash_{\sim} K$, $a \Vdash_{\sim} L \wedge A^\circ$ and $b \Vdash_{\sim} L \wedge B^\circ$. Then $a \Vdash_{\sim} L$ and $b \Vdash_{\sim} L$ and there exists $\delta \in K^*$ and $x, y \in L^*$ such that $\delta \sim k$, $x \sim a$ and $y \sim b$. By three applications of rule $\langle e_l \rangle$, $\delta xy \sim m$, hence $xy \sqsubseteq m$. By monotonicity of \Vdash_{\sim} and rule $\langle s \rangle$, as $a \Vdash_{\sim} A^\circ$ and $b \Vdash_{\sim} B^\circ$, we obtain $x \Vdash_{\sim} A^\circ$ and $y \Vdash_{\sim} B^\circ$, hence by induction hypothesis, $x \Vdash_{\sqsubseteq} A$ and $y \Vdash_{\sqsubseteq} B$. Thus $m \Vdash_{\sqsubseteq} A * B$;

$\boxed{F = A \multimap B}$ let us suppose $m \Vdash_{\sqsubseteq} A \multimap B$ with $m \in \mathcal{L}^{\sqsubseteq}$. Let us prove $m \Vdash_{\sim} (K * (L \wedge A^\circ)) \multimap (L \rightarrow B^\circ)$. So let c, d such that $cm \sim d$, $c \Vdash_{\sim} K * (L \wedge A^\circ)$ and $d \Vdash_{\sim} L$. Let us prove that $d \Vdash_{\sim} B^\circ$. There exist k, a such that $ka \sim c$, $k \Vdash_{\sim} K$, $a \Vdash_{\sim} L$ and $a \Vdash_{\sim} A^\circ$. Then there exists $\delta \in K^*$ and $x \in L^*$ such that $\delta \sim k$ and $x \sim a$. By monotonicity, $x \Vdash_{\sim} A^\circ$, hence by induction hypothesis, we obtain $x \Vdash_{\sqsubseteq} A$. Moreover $\delta xm \sim d$ by three applications of rule $\langle e_l \rangle$. As $d \Vdash_{\sim} L$, let $y \in L^*$ such that $y \sim d$. Then $\delta xm \sim y$ by rule $\langle e_r \rangle$. Thus we have $xm \sqsubseteq y$ and $m \Vdash_{\sqsubseteq} A \multimap B$, and then $y \Vdash_{\sqsubseteq} B$. By induction hypothesis, we obtain $y \Vdash_{\sim} B^\circ$ and by monotonicity $d \Vdash_{\sim} B^\circ$. We have proved that $m \Vdash_{\sim} (K * (L \wedge A^\circ)) \multimap (L \rightarrow B^\circ)$;

On the other hand, let us suppose $m \Vdash_{\sim} (K * (L \wedge A^\circ)) \multimap (L \rightarrow B^\circ)$. Let $x, y \in \mathcal{L}^{\sqsubseteq}$ such that $xm \sqsubseteq y$ and $x \Vdash_{\sqsubseteq} A$. Let us prove $y \Vdash_{\sqsubseteq} B$. There exists $\delta \in K^*$ such that $\delta xm \sim y$. As $x \in L^*$ and $x \Vdash_{\sim} A^\circ$ (by induction hypothesis), we then obtain $x \Vdash_{\sim} L \wedge A^\circ$. As $\delta \Vdash_{\sim} K$ and $\delta x \sim \delta x$ (by rule $\langle p_l \rangle$), we get $\delta x \Vdash_{\sim} K * (L \wedge A^\circ)$. As $\delta xm \sim y$, we derive $y \Vdash_{\sim} L \rightarrow B^\circ$. But $y \in L^*$ thus $y \Vdash_{\sim} L$, hence $y \Vdash_{\sim} B^\circ$, so $y \Vdash_{\sqsubseteq} B$ by induction hypothesis. We have proved that $m \Vdash_{\sqsubseteq} A \multimap B$.

So we have inductively proved that for any $m \in \mathcal{L}^{\sqsubseteq}$ and any formula F of BI containing neither L nor K , $m \Vdash_{\sqsubseteq} F$ if and only if $m \Vdash_{\sim} F^\circ$. \square

With this result, we have established the first step of our embedding of BI into BBI. Indeed, provided a BI-counter-model of F can be chosen of the form $\sqsubseteq_{\sim}^{L,K}$ for some PME \sim , then we automatically obtain a BBI-counter-model of F° . So we are now going to study more precisely the counter-models of BI based on PMOs to show that this condition is not restrictive.

5. Tableau Proof Systems for BI and BBI

Tableau systems are refutation based procedures that produces *statements* like $\mathbb{T}A$ or $\mathbb{F}A$. Sometimes statements may also be written A or $\neg A$ like in the reference textbook (Fitting 1990). The statement $\mathbb{T}A$ expresses the fact that the tableau refutation process tries to build a model of the formula A whereas the statement $\mathbb{F}A$ expresses the fact that the refutation process tries to build a counter-model of A .

Tableaux for a formula G are finite trees indexed with statements obtained by some *branch expansion* process described by expansion rules and starting from the one node tree $\mathbb{F}G$. So a tableau for G contains the trace of a process that tries to refute G . The formula A occurring in the statements $\mathbb{T}A$ or $\mathbb{F}A$ produced by the branch expansion process are usually sub-formulae of the initial formula G , although it is not always the case for some non-classical logics.

The expansion process works as following: to refute a branch $\gamma_0 = [\dots, \mathbb{F}A \vee B, \dots]$ containing a statement $\mathbb{F}A \vee B$, γ_0 is expanded into one branch $[\gamma_0, \mathbb{F}A, \mathbb{F}B]$ whereas to refute a branch $\gamma_1 = [\dots, \mathbb{T}A \vee B, \dots]$ containing a statement $\mathbb{T}A \vee B$, γ_1 is expanded into two branches, $[\gamma_1, \mathbb{T}A]$ and $[\gamma_1, \mathbb{T}B]$. These two instances of the branch expansion process are represented by the two following branch expansion rules, $\mathbb{F}\vee$ on the left hand side and $\mathbb{T}\vee$ on the right hand side:

$$\begin{array}{ccc} \mathbb{F}A \vee B & & \mathbb{T}A \vee B \\ | & & \swarrow \quad \searrow \\ \mathbb{F}A & & \mathbb{T}A \quad \mathbb{T}B \\ \mathbb{F}B & & \end{array}$$

The justification of these rules corresponds to the following semantic arguments: for $A \vee B$ to be invalid, it is required that A and B are invalid whereas for $A \vee B$ to be valid, it is sufficient that either A or B is valid.

The expansion process stops either when branch expansion does not generate new statements, and then the branch is called *saturated*; or when a contradiction occurs like in the branch $[\dots, \mathbb{F}A, \dots, \mathbb{T}A, \dots]$, in which case the branch is called *closed*. It is not possible to refute such a closed branch because the formula A cannot be both valid and invalid. A tableau for G which only has closed branches is called a *closed tableau* and is generally a witness of the universal validity of G , depending of course on the soundness theorem for the tableau method.

When considering non-classical logics, it is sometimes useful to enrich statements with *labels* like $\mathbb{T}A : m$ or $\mathbb{F}A : m$. In $\mathbb{T}A : m$, the label m carries some semantic information about the world in which the Kripke forcing relation $m \Vdash A$ holds. For intuitionistic logic IL for example, labelled statements and *unification of prefixes* are sufficient to provide a

sound and complete proof system (Otten and Kreitz 1996), although the system is based on Wallen's matrix characterization rather than the tableau method.

Unfortunately, labels alone do not carry enough information to provide a sound and complete tableau proof system for BI. The statements also have to be enriched with *constraints* of the form $m \sqsubseteq n$ which are relations that are supposed to hold between labels. A sound and complete labelled tableau proof system was proposed for BI in (Galmiche and Méry 2003) and we briefly recall it in the next section. We also adapt the proof system to BBI and prove its soundness. In fact, we propose a common framework to describe BI- and BBI-tableaux.

5.1. Labelled tableaux with constraints for BI and BBI

We define the notion of tableau with constraints for BI (resp. BBI) providing a proof system called TBI (resp. TBBI). We warn the reader that the following definition refers to *tableaux expansion rules*, rules which are described a bit later and that he should not jump on them before he is actually prepared to read them.

Definition 5.1 (TBI- and TBBI-tableaux). Let L be an alphabet. A TBI- (resp. TBBI-) *tableau with constraints for a formula G* is a finite tree with nodes labelled either by *statements of the form $\mathbb{S}A : m$* where $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$, $A \in \text{BI}$ (resp. $A \in \text{BBI}$) and $m \in L^*$ or by *assertions which are constraints of the form $m \leftarrow n$* where $m, n \in L^*$, built according to the following rules:

- the single node tree $[\mathbb{F}G : \epsilon]$ is a TBI-tableau for G ;
- the two nodes tree $[a \leftarrow b, \mathbb{F}G : a]$ is a TBBI-tableau for G whenever $a \neq b \in L$;
- any (maximal) branch of a tableau for G can be expanded according to the tableau expansion rules of TBI (resp. TBBI).

We might speak of a tableau \mathcal{T} without specifying formula G , simply meaning that \mathcal{T} is a tableau for some formula G (which can be recovered by reading at the root of the tableau tree).

The tableau expansion rules for both TBI and TBBI are described in Tables 2 and 3. The rules in Table 2 are common to both TBI and TBBI ($\{\mathbb{T}, \mathbb{F}\}\{\wedge, \vee, *, \rightarrow\}$ and $\mathbb{T}\mathbb{I}$). Table 3 contains rules either specific to TBI ($\{\mathbb{T}, \mathbb{F}\}\rightarrow$ on the left part) or TBBI ($\{\mathbb{T}, \mathbb{F}\}\{\rightarrow, \neg\}$ on the right part).

In these expansion rules, existing statements (like $\mathbb{T}A * B : m$ in rule $\mathbb{T}*$) are decomposed into new statements (e.g. $\mathbb{T}A : a$ and $\mathbb{T}B : b$) and new assertions (e.g. $\text{ass} : ab \leftarrow m$). These are the products of the decomposition rule. Rules $\mathbb{T}*$ and $\mathbb{F}*$ have a side condition: the letters a and b should be chosen new in the current tableau branch. Rule $\mathbb{F}*$ (resp. $\mathbb{T}*$) has another kind of side condition: the words x and y should be chosen such that the relation $xy R_\gamma m$ (resp. $xm R_\gamma y$) holds with R_γ being either the PMO \sqsubseteq_γ (for TBI) or PME \sim_γ (for TBBI) generated by the assertions of the current tableau branch γ . For the rules which are specific to either TBI or TBBI, the side conditions are: in rule $\mathbb{F}\rightarrow$, the letter b should be new; in rule $\mathbb{T}\rightarrow$, the word x should verify relation $m \sqsubseteq_\gamma x$ where \sqsubseteq_γ is the PMO generated by the current branch γ .

	$\begin{array}{c} \mathbb{T}A \wedge B : m \\ \\ \mathbb{T}A : m \\ \mathbb{T}B : m \end{array}$	$\begin{array}{c} \mathbb{T}A \vee B : m \\ \swarrow \quad \searrow \\ \mathbb{T}A : m \quad \mathbb{T}B : m \end{array}$
	$\begin{array}{c} \mathbb{F}A \wedge B : m \\ \swarrow \quad \searrow \\ \mathbb{F}A : m \quad \mathbb{F}B : m \end{array}$	$\begin{array}{c} \mathbb{F}A \vee B : m \\ \\ \mathbb{F}A : m \\ \mathbb{F}B : m \end{array}$
$\begin{array}{c} \mathbb{T}1 : m \\ \\ \text{ass} : \epsilon \multimap m \end{array}$	$\begin{array}{c} \mathbb{T}A * B : m \\ \\ \text{ass} : ab \multimap m \\ \mathbb{T}A : a \\ \mathbb{T}B : b \end{array}$	$\begin{array}{c} \mathbb{T}A \multimap B : m \\ \\ \text{req} : xm R y \\ \swarrow \quad \searrow \\ \mathbb{F}A : x \quad \mathbb{T}B : y \end{array}$
	$\begin{array}{c} \mathbb{F}A * B : m \\ \\ \text{req} : xy R m \\ \swarrow \quad \searrow \\ \mathbb{F}A : x \quad \mathbb{F}B : y \end{array}$	$\begin{array}{c} \mathbb{F}A \multimap B : m \\ \\ \text{ass} : am \multimap b \\ \mathbb{T}A : a \\ \mathbb{F}B : b \end{array}$

Table 2. *Tableaux expansion rules which are common in TBI and TBBI, for the additives (\wedge, \vee) in the upper part and for the multiplicatives ($1, *, \multimap$) in the lower part.*

$\begin{array}{c} \mathbb{T}A \rightarrow B : m \\ \\ \text{req} : m \sqsubseteq x \\ \swarrow \quad \searrow \\ \mathbb{F}A : x \quad \mathbb{T}B : x \end{array}$	$\begin{array}{c} \mathbb{T}A \rightarrow B : m \\ \swarrow \quad \searrow \\ \mathbb{F}A : m \quad \mathbb{T}B : m \end{array}$	$\begin{array}{c} \mathbb{T}\neg A : m \\ \\ \mathbb{F}A : m \end{array}$
$\begin{array}{c} \mathbb{F}A \rightarrow B : m \\ \\ \text{ass} : m \multimap b \\ \mathbb{T}A : b \\ \mathbb{F}B : b \end{array}$	$\begin{array}{c} \mathbb{F}A \rightarrow B : m \\ \\ \mathbb{T}A : m \\ \mathbb{F}B : m \end{array}$	$\begin{array}{c} \mathbb{F}\neg A : m \\ \\ \mathbb{T}A : m \end{array}$

Table 3. *Tableaux expansion rules specific to TBI (on the left part) and tableaux expansion rules specific to TBBI (on the right part).*

The build process in tableau systems is based on the notion of branch expansion. This process is explained in full details in Section 5.2, as well as how tableaux expansion rules are supposed to be applied. We also explain how to build the $\text{PMO} \sqsubseteq_\gamma$ (resp. $\text{PME} \sim_\gamma$) generated by a branch γ of a tableau. The expansion process may stop when a *closure condition* is fulfilled.

Definition 5.2 (Closure conditions). A branch γ of a TBI- or TBBI-tableau is closed if one of the following condition is fulfilled for some propositional variable $X \in \text{Var}$ and some $m, n \in L^*$:

1. $\mathbb{T}X : m \in \gamma, \mathbb{F}X : n \in \gamma \text{ and } m R_\gamma n$
2. $\mathbb{F}I : m \in \gamma \text{ and } \epsilon R_\gamma m$
3. $\mathbb{T}\perp : m \in \gamma$
4. $\mathbb{F}\top : m \in \gamma$

where R_γ is either \sqsubseteq_γ or \sim_γ depending on whether we consider a TBI- or TBBI-tableau. If a branch is not closed, then it is *open*. A *tableau is closed* if all its branches are closed.

5.2. Explanations on tableau proof rules and the branch expansion process

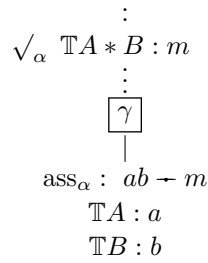
Each branch γ of a tableau tree contains a sequence of assertions and statements. Assertions are constraints and we collect them in a sequence or set of constraints \mathcal{C}_γ (the order or potential duplication of constraints is indifferent). The PMO $\sqsubseteq_\gamma = \sqsubseteq_{\mathcal{C}_\gamma}$ (resp. PME $\sim_\gamma = \sim_{\mathcal{C}_\gamma}$) is associated to the branch γ : hence $\sqsubseteq_\gamma/\sim_\gamma$ is the PMO/PME generated by the assertions of γ (see Section 6.2 for an example). We also denote by A_γ the alphabet A_{\sqsubseteq_γ} (resp. A_{\sim_γ}) of the relation \sqsubseteq_γ (resp. \sim_γ), which is exactly the set of letters that occur in the assertions of \mathcal{C}_γ (see Proposition 3.16).

As explained in the previous section, some of the rules of the two systems are common (see Table 2) while others are different (see Table 3). The rules which differ are $\mathbb{T}\rightarrow$, $\mathbb{F}\rightarrow$,[§] and of course $\mathbb{T}\neg$ and $\mathbb{F}\neg$ because $\neg A$ is not a BI formula. The fact that some rules have the same shape for TBI and TBBI does not imply that they can always be used when the constraints in the branch are identical: the corresponding PMO \sqsubseteq_γ and PME \sim_γ may differ, implying different situations for the side conditions. Indeed, some rules have a *requirement* $x R y$ that imposes the condition $x \sqsubseteq_\gamma y$ ($R = \sqsubseteq_\gamma$) in TBI, whereas the condition is $x \sim_\gamma y$ ($R = \sim_\gamma$) in TBBI. And of course, \sqsubseteq_γ and \sim_γ are generally not identical relations. Let us further comment on the assertions $x \multimap y$ and the requirements $x R y$ that occur in the tableaux expansion rules:

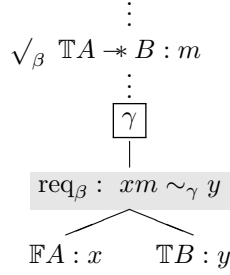
- the former assertion $x \multimap y$ indicates that the constraint $x \multimap y$ should be added to the branch as well. The concerned rules are $\mathbb{T}I : m, \mathbb{T}A * B : m, \mathbb{F}A \multimap B : m$ (for both TBI and TBBI) and also $\mathbb{F}A \rightarrow B : m$ (but just for TBI). Let us consider the example of rule $\mathbb{T}A * B : m$ for TBI. It can be applied to a branch γ of a tableau provided the statement $\mathbb{T}A * B : m$ occurs in γ (not necessarily at the leaf of γ) and, as a side condition, the letters $a \neq b \in L$ are new to γ , i.e. $a, b \notin A_\gamma$. Then γ is expanded into

[§] The TBI-rules $\mathbb{T}\rightarrow$ and $\mathbb{F}\rightarrow$ are also valid and even complete for BBI but they are profitably replaced with simpler rules because $A \rightarrow B$ is logically equivalent to $\neg A \vee B$ in BBI (see Proposition 3.9) whereas this equivalence does not hold in BI.

the one branch $[\gamma, ab \multimap m, \mathbb{T}A : a, \mathbb{T}B : b]$ as displayed below:



- the latter requirement $x R y$ is just a side condition that should be fulfilled so that the expansion rule could be applied. For example in the case of $\mathbb{T}B\mathbb{I}$, to apply rule $\mathbb{T}A \multimap B : m$ to the branch γ , the statement $\mathbb{T}A \multimap B : m$ should occur in γ and the chosen words $x, y \in L^*$ should verify $xm \sim_{\gamma} y$, R being interpreted as $R = \sim_{\gamma}$. Then the branch γ can be expanded into the two branches $[\gamma, \mathbb{F}A : x]$ and $[\gamma, \mathbb{T}B : y]$ as displayed below:



We point out that *we generally tag the tableau trees* with requirements (like req_{β} generally displayed in a grey box) and *history information* (like $\sqrt{\alpha}$, $\sqrt{\beta}$ and α , β in ass_{α} and req_{β}) so that it is easier for the reader to check which rule is applied and why the conditions for its application are fulfilled. Formally, requirements and history are not part of tableau trees. On the other hand, assertions are not boxed but displayed prefixed with ass . Contrary to requirements, assertions are critical bits of semantic information, not just a guideline for checking that the tableau tree is well formed. Finally, statements, which also constitute critical bits of information, are not prefixed because they always start with either \mathbb{T} or \mathbb{F} .

Proposition 5.3. For any branch γ in a $\mathbb{T}B\mathbb{I}$ - (resp. $\mathbb{T}B\mathbb{I}$ -) tableau, if the statement $\mathbb{S}F : m$ occurs in γ then the relation $m \sqsubseteq_{\gamma} m$ (resp. $m \sim_{\gamma} m$) holds.

Proof. By induction on the tableau expansion process using rules $\langle p_l, p_r \rangle$. The only tableaux rules that introduce statements with new labels are:

- $\mathbb{T}*$ (resp. $\mathbb{F}*$) but in this case the labels a and b are defined by the assertion $ab \multimap m$ (resp. $am \multimap b$), i.e. if $R' = R + \{ab \multimap m\}$ (resp. $R' = R + \{am \multimap b\}$) then $a R' a$ and $b R' b$ whether $R \equiv \sqsubseteq$ is a PMO (for $\mathbb{T}B\mathbb{I}$) or $R \equiv \sim$ is a PME (for $\mathbb{T}B\mathbb{I}$). Simply apply the rules $\langle p_l, p_r \rangle$;
- $\mathbb{F}*$ (resp. $\mathbb{T}*$) but in this case the labels x and y must already be defined because

- $xm R y$ (resp. $xm R y$) implies $x R x$ and $y R y$ whether R is a PMO (for TBI) or PME (for TBB);
- $\mathbb{T} \rightarrow$ (for TBI) but in this case the label x must already be defined because $m \sqsubseteq x$ implies $x \sqsubseteq x$;
- $\mathbb{F} \rightarrow$ (for TBI) but in this case the label b is defined by the assertion $m \rightarrow b$, i.e. if $\sqsubseteq' = \sqsubseteq + \{m \rightarrow b\}$ then $b \sqsubseteq' b$.

The other tableau expansion rules introduce statements with labels that must already exist in a previous statement in the branch γ , hence we must already have $m \sqsubseteq_\gamma m$ (resp. $m \sim_\gamma m$) by induction hypothesis. For example, for rule \vee , the statement $\mathbb{T}A \vee B : m$ must already occur in the branch γ before the rule is applied producing the two branches $[\gamma, \mathbb{T}A : m]$ and $[\gamma, \mathbb{T}B : m]$. \square

5.3. Elementary and simple PMOs/BI frames

The TBI-tableau method provides some insights into the semantics of BI. Indeed, the TBI-tableau rules which introduce assertions, namely $\mathbb{F} \rightarrow$, $\mathbb{T}I$, $\mathbb{T}*$ and $\mathbb{F}*$, do not introduce constraints of arbitrary form. We collect all the possible forms into the following definitions and Proposition 5.7 proves that our gathering is adequate.

Definition 5.4. Given a PMO \sqsubseteq over L , a constraint is *BI-elementary w.r.t. \sqsubseteq* when it is of one of the five following forms:

1. $ab \rightarrow m$ with $m \sqsubseteq m$, $m \neq \epsilon$ and $a \neq b \in L \setminus A_\sqsubseteq$;
2. $am \rightarrow b$ with $m \sqsubseteq m$ and $a \neq b \in L \setminus A_\sqsubseteq$;
3. $m \rightarrow b$ with $m \sqsubseteq m$ and $b \in L \setminus A_\sqsubseteq$;
4. $\epsilon \rightarrow m$ with $m \sqsubseteq m$ and $m \neq \epsilon$;
5. $\epsilon \rightarrow \epsilon$.

Let $(x_i \rightarrow y_i)_{i < k}$ be a sequence of constraints with $k \in \mathbb{N} \cup \{\infty\}$ and \mathcal{C}_p be the set of constraints $\mathcal{C}_p = \{x_i \rightarrow y_i \mid i < p\}$ for $p < k$. We suppose that for any $p < k$, the constraint $x_p \rightarrow y_p$ is BI-elementary with respect to $\sqsubseteq_{\mathcal{C}_p}$ (resp. $\sim_{\mathcal{C}_p}$). If $k < \infty$ then the sequence $(x_i \rightarrow y_i)_{i < k}$ is called *BI-elementary*. This definition implies in particular that the empty sequence of constraints is BI-elementary. If $k = \infty$ then the sequence $(x_i \rightarrow y_i)_{i < \infty}$ is called *BI-simple*.

Definition 5.5. A PMO is *BI-elementary* (resp. *BI-simple*) if it is of the form $\sqsubseteq_{\mathcal{C}}$ where $\mathcal{C} = \{x_i \rightarrow y_i \mid i < k\}$ and $(x_i \rightarrow y_i)_{i < k}$ is a BI-elementary (resp. BI-simple) sequence of constraints.

We make the obvious following remark. According to those definitions, if \sqsubseteq is a BI-elementary PMO and the constraint $x \rightarrow y$ is BI-elementary w.r.t. \sqsubseteq then the PMO extension $\sqsubseteq + \{x \rightarrow y\}$ is a BI-elementary PMO.

Using case 5 ($\epsilon \rightarrow \epsilon$) of Definition 5.4, any finite BI-elementary sequence can be completed into a infinite BI-simple sequence by repeated use of the constraint $\epsilon \rightarrow \epsilon$. Since adding this constraint does not change the corresponding PMO (because of rule $\langle \epsilon \rangle$), BI-elementary PMOs are also BI-simple. Of course, the converse is not true. Indeed, the

language of a BI-elementary PMO is always finite whereas the language of a BI-simple PMO can be infinite. So the difference between BI-elementary and BI-simple PMOs is that in the later case, the underlying sequence can be infinite whereas it must be finite for BI-elementary PMOs.

Let us now prove that the PMOs occurring in the branches of TBI-tableau are BI-elementary. But we must first establish that $m \sqsubseteq \epsilon$ never holds for BI-elementary PMOs unless of course $m = \epsilon$.

Proposition 5.6. If \sqsubseteq is a BI-elementary or BI-simple PMO over L , then for any $m \in L^*$, $m \sqsubseteq \epsilon$ only if $m = \epsilon$.

Proof. According to the definition of BI-elementary constraints, they can be of the form $m \multimap \epsilon$ only if $m = \epsilon$ (case 5 of Definition 5.4). So we can apply property 1 of Proposition 3.19. \square

Proposition 5.7. Statements of the form $\mathbb{T}F : \epsilon$ for some $F \in \text{BI}$ never occur in a TBI-tableau and for every branch γ of a TBI-tableau the sequence of assertions occurring in γ is BI-elementary.

Proof. The proof of these two properties is done by mutual induction on the TBI-tableau construction process for G . Of course, the properties are valid for the single node TBI-tableau $\llbracket FG : \epsilon \rrbracket$. Indeed there is only one branch $\gamma = \llbracket FG : \epsilon \rrbracket$ which has no assertion, the empty sequence of constraints is BI-elementary, and $\mathbb{T}F : \epsilon$ does not occur in γ .

Then we consider tableau expansion rules. The only rules that may introduce a statement of the form $\mathbb{T}F : \epsilon$ are $\mathbb{T}\wedge$, $\mathbb{T}\vee$, $\mathbb{T}\rightarrow$ and $\mathbb{T}\multimap$:

- For $\mathbb{T}\wedge$ (resp. $\mathbb{T}\vee$) it would mean that $\mathbb{T}A \wedge B : \epsilon$ (resp. $\mathbb{T}A \vee B : \epsilon$) already occurs in the branch which is false by induction;
- For $\mathbb{T}A \rightarrow B : m$, it would mean that $m \sqsubseteq_{\gamma} \epsilon$ holds. But by Proposition 5.6, since \sqsubseteq_{γ} is BI-elementary by induction hypothesis, we must have $m = \epsilon$, hence $\mathbb{T}A \rightarrow B : \epsilon$ occurs in γ which contradicts the induction hypothesis;
- For $\mathbb{T}A \multimap B : m$, it would mean that $xm \sqsubseteq_{\gamma} \epsilon$ holds and then $xm = \epsilon$ which implies $m = \epsilon$ and thus $\mathbb{T}A \multimap B : \epsilon$ occurs in γ , contradiction.

So we have proved that branch expansion cannot introduce $\mathbb{T}F : \epsilon$ in a branch. We now prove that the sequences of assertions remain BI-elementary in the expanded branches. Four rules introduce new assertions: $\mathbb{T}\ast$, $\mathbb{F}\ast$, $\mathbb{F}\rightarrow$ and $\mathbb{T}\text{I}$. The corresponding constraints are respectively $ab \multimap m$, $am \multimap b$, $m \multimap b$ and $\epsilon \multimap m$. It is not a coincidence that these are exactly the four first constraint types BI-elementary w.r.t. \sqsubseteq_{γ} , see Definition 5.4. In the case of $ab \multimap m$ or $\epsilon \multimap m$, we have $m \sqsubseteq_{\gamma} m$ by Proposition 5.3 and we have $m \neq \epsilon$ because otherwise $\mathbb{T}A \ast B : \epsilon$ or $\mathbb{T}\text{I} : \epsilon$ would occur in γ which contradicts the induction hypothesis. \square

So we have singled out a sub-class of PMOs, namely BI-elementary PMOs and their limits BI-simple PMOs, which are those that occur in the branches of TBI-tableau trees. In the following sections, we will exploit this sub-class and its properties to establish a direct link between the models of BI and those of BBI.

5.4. Soundness and completeness of the TBI-tableau system

We state the completeness of the TBI-tableau method for BI. We also state that the class of BI-simple PMOs is complete for BI and hence, BI-counter-models can always be chosen BI-simple. The proof of the following theorems can be found in (Galmiche *et al.* 2005) and in Daniel Méry's thesis (Méry 2004).

Theorem 5.8 (Soundness and completeness of TBI). Provided the alphabet L is infinite, there exists a closed TBI-tableau for the formula G if and only if G is a BI-valid formula.

Theorem 5.9 (Completeness for simple PMOs). Every invalid formula of BI has a BI-counter-model of the form of the BI-frame (L, \sqsubseteq, \Vdash) where \sqsubseteq is a BI-simple PMO.

The idea is that the counter-model is extracted from an open, saturated and potentially infinite branch of a TBI-tableaux sequence. This counter-model is composed of the infinite sequence of assertions occurring the branch and is thus a BI-simple PMO.

5.5. Soundness of the TBBI-tableau system

Soundness and completeness also hold for the TBBI-tableau system. However, there is no published proof of these results for the moment. In this section, we give a soundness proof for the TBBI-tableau system w.r.t. BBI-frames. We do not provide the proof of completeness for two reasons:

- first reason: we do not need completeness for our purpose of embedding BI into BBI, we only need soundness of the TBBI-tableau system;
- second reason: the completeness proof is much more complicated than the soundness proof. In particular, it involves the manipulation of infinite branches of tableaux. In this article, we have chosen a definition of tableaux as finite trees because the definition suits better for graphical representation as in Section 7.1. For the formalization of the completeness proof, it is much easier to represent tableaux as sets of branches, the branches being finite or infinite sets of statements.

The soundness proof is decomposed as usual into two parts. First, we prove that branch expansion preserves *realizability*. Then we show that closed branches are not realizable.

We consider TBBI-tableaux over the alphabet L and BBI-frames over the alphabet K where L and K are not necessarily identical. So in statements $\mathbb{S}F : m$ or assertions $m \multimap n$, m, n belong to L^* whereas the relation $q \Vdash F$ or $q \nVdash F$ in the frame (K, \sim, \Vdash) involves q belonging to K^* . Given a total map $\rho : L \longrightarrow K^*$, for $m = m_1 \dots m_p \in L^*$, we define $m_\rho = \rho(m_1) \dots \rho(m_p)$ and obtain a morphism of (commutative) monoids $(\cdot)_\rho : L^* \longrightarrow K^*$.

Given $\mathcal{K} = (K, \sim, \Vdash, \rho)$, we say that the statement $\mathbb{T}A : m$ (resp. $\mathbb{F}A : m$) is satisfied in \mathcal{K} if $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \Vdash A$ (resp. $m_\rho \nVdash A$). We say that the assertion $m \multimap n$ is satisfied in \mathcal{K} if $m_\rho \sim n_\rho$.

Definition 5.10. We say that a branch of a tableau is satisfied in $\mathcal{K} = (K, \sim, \Vdash, \rho)$ if all its statements and all its assertions are satisfied in \mathcal{K} . We say a tableau \mathcal{T} for G is *realizable* if there exists \mathcal{K} such that at least one of the branches of \mathcal{T} is satisfied in \mathcal{K} .

Proposition 5.11. If a branch γ of \mathcal{T} is satisfied in $\mathcal{K} = (K, \sim, \Vdash, \rho)$ and $m, n \in L^*$ verify $m \sim_\gamma n$ then $m_\rho \sim n_\rho$.

Proof. As all the assertions of γ are satisfied in \mathcal{K} , the binary relation $\sim' \subseteq L^* \times L^*$ defined by $m \sim' n$ iff $m_\rho \sim n_\rho$ contains all the assertions of γ . Moreover, it is straightforward to prove that the relation \sim' is a PME over L from the fact that \sim is a PME over K and $m \mapsto m_\rho$ is a morphism of monoids. Hence, since \sim_γ is the least PME containing all the assertions of γ , we have $\sim_\gamma \subseteq \sim'$. \square

If a branch γ is satisfied in \mathcal{K} then all of its requirements, i.e. all the constraints in the PME \sim_γ , are also satisfied in \mathcal{K} . Moreover, if $\mathbb{S}A : m \in \gamma$ then, by Proposition 5.3 we have $m \sim_\gamma m$, hence $m_\rho \sim m_\rho$.

Proposition 5.12. Closed TBBI-tableaux are not realizable.

Proof. We prove that a closed branch γ cannot be satisfied in any (K, \sim, \Vdash, ρ) . Let us suppose the contrary and proceed by case analysis on the closure condition:

- if $\mathbb{T}X : m \in \gamma$, $\mathbb{F}X : n \in \gamma$ and $m \sim_\gamma n$ then, as γ is satisfied in \mathcal{K} , we have $m_\rho \sim n_\rho$ by Proposition 5.11. Moreover, both $\mathbb{T}X : m$ and $\mathbb{F}X : n$ are satisfied and thus $m_\rho \Vdash X$ and $n_\rho \nVdash X$. As $m_\rho \sim n_\rho$, we obtain a contradiction by monotonicity of \Vdash ;
- if $\mathbb{F}I : m \in \gamma$ and $\epsilon \sim_\gamma m$ then, as γ is satisfied in \mathcal{K} , we have $\epsilon = \epsilon_\rho \sim m_\rho$ and $m_\rho \nVdash I$. But then, we should have $\epsilon \approx m_\rho$. Thus we obtain a contradiction;
- if $\mathbb{T}\perp : m \in \gamma$ then, as γ is satisfied in \mathcal{K} , we have $m_\rho \sim m_\rho$ and $m_\rho \Vdash \perp$ which is impossible;
- if $\mathbb{F}\top : m \in \gamma$ then, as γ is satisfied in \mathcal{K} , we have $m_\rho \sim m_\rho$ and $m_\rho \nVdash \top$ which is impossible.

So we obtain a contradiction in any case. A closed branch cannot be satisfied. Thus closed TBBI-tableaux are not realizable. \square

Lemma 5.13. TBBI-tableaux expansion rules preserve realizability.

Proof. Let \mathcal{T} be a realizable BBI-tableau and let $\mathcal{K} = (K, \sim, \Vdash, \rho)$ be such that at least one branch of \mathcal{T} is satisfied in \mathcal{K} . We consider the expansion of one of the branches of \mathcal{T} by one of the rules of the TBBI-tableau system. If the expanded branch is not among the satisfied ones then the satisfied branches are unchanged by the application of the rule and the obtained tableau \mathcal{T}' is thus still realizable.

So we consider the case when the branch γ we expand is among the satisfied ones. We proceed by case analysis depending on the rule applied:

- $\boxed{\mathbb{T}\neg A : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \Vdash \neg A$. Thus $m_\rho \nVdash A$ and $\mathbb{F}A : m$ is satisfied in \mathcal{K} . So the new branch $[\gamma, \mathbb{F}A : m]$ of \mathcal{T}' is satisfied in \mathcal{K} ;
- $\boxed{\mathbb{F}\neg A : m}$ similar to case $\mathbb{T}\neg$;
- $\boxed{\mathbb{T}A \wedge B : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \Vdash A \wedge B$. Then $m_\rho \Vdash A$ and $m_\rho \Vdash B$ hence $\mathbb{T}A : m$ and $\mathbb{T}B : m$ are satisfied in \mathcal{K} . So the new branch $[\gamma, \mathbb{T}A : m, \mathbb{T}B : m]$ of \mathcal{T}' is satisfied in \mathcal{K} ;

- $\boxed{\mathbb{F}A \wedge B : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and either $m_\rho \not\models A$ or $m_\rho \not\models B$. Hence either $\mathbb{F}A : m$ or $\mathbb{F}B : m$ is satisfied in \mathcal{K} . So at least one of the two new branches of \mathcal{T}' (namely $[\gamma, \mathbb{F}A : m]$ or $[\gamma, \mathbb{F}B : m]$) is satisfied in \mathcal{K} ;
- $\boxed{\mathbb{T}A \vee B : m}$ similar to case $\mathbb{F}\wedge$;
- $\boxed{\mathbb{F}A \vee B : m}$ similar to case $\mathbb{T}\wedge$;
- $\boxed{\mathbb{T}A \rightarrow B : m}$ similar to case $\mathbb{F}\wedge$;
- $\boxed{\mathbb{F}A \rightarrow B : m}$ similar to case $\mathbb{T}\wedge$;
- $\boxed{\mathbb{T}1 : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \models 1$. Thus $\epsilon \sim m_\rho$. As $\epsilon_\rho = \epsilon$, we obtain $\epsilon_\rho \sim m_\rho$ and thus the assertion $\epsilon \vdash m$ is satisfied in \mathcal{K} . So the new branch $[\gamma, \epsilon \vdash m]$ of \mathcal{T}' is satisfied in \mathcal{K} ;
- $\boxed{\mathbb{T}A * B : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \models A * B$. So there exists $x, y \in \mathcal{L}^\sim$ such that $xy \sim m_\rho$, $x \models A$ and $y \models B$. We define $\rho' = \rho[a \mapsto x, b \mapsto y]$ (possible because $a \neq b$). Then for any $m, n \in L^*$ s.t. $m \sim_\gamma n$ we have $m, n \in A_\gamma^*$ and thus $m_{\rho'} = m_\rho$ and $n_{\rho'} = n_\rho$ (ρ and ρ' are identical maps when restricted to A_γ because $a, b \notin A_\gamma$). Thus γ is satisfied in $\mathcal{K}' = (K, \sim, \models, \rho')$. Moreover, $ab \vdash m$ is satisfied in \mathcal{K}' (because $(ab)_{\rho'} = xy$, $m_{\rho'} = m_\rho$ and $xy \sim m_\rho$), $\mathbb{T}A : a$ is satisfied (because $a_{\rho'} = x$ and $x \models A$), and $\mathbb{T}B : b$ is satisfied (because $b_{\rho'} = y$ and $y \models B$). So the (new) branch $[\gamma, ab \vdash m, \mathbb{T}A : a, \mathbb{T}B : b]$ of \mathcal{T}' is satisfied in \mathcal{K}' ;
- $\boxed{\mathbb{F}A * B : m}$ is satisfied in \mathcal{K} hence $m_\rho \in \mathcal{L}^\sim$ and $m_\rho \not\models A * B$. γ is expanded into two branches $[\gamma, \mathbb{F}A : x]$ and $[\gamma, \mathbb{F}B : y]$ with $xy \sim_\gamma m$. Then $x_\rho y_\rho \sim m_\rho$ and thus $x_\rho, y_\rho \in \mathcal{L}^\sim$. So either $x_\rho \not\models A$ or $y_\rho \not\models B$. Thus at least one of the two new branches of \mathcal{T}' (namely $[\gamma, \mathbb{F}A : x]$ and $[\gamma, \mathbb{F}B : y]$) is satisfied in \mathcal{K} ;
- $\boxed{\mathbb{T}A \multimap B : m}$ similar to case $\mathbb{F}\ast$;
- $\boxed{\mathbb{F}A \multimap B : m}$ similar to case $\mathbb{T}\ast$.

So in any case, there exists a satisfiable branch in \mathcal{T}' and thus \mathcal{T}' is realizable. \square

Theorem 5.14 (Soundness of TBBI). If there exists a closed TBBI-tableau for the formula G then G is a valid BBI formula.

Proof. Let us suppose that G has a counter-model (K, \sim, \models) , i.e. there exists $m \in \mathcal{L}^\sim$ such that $m \not\models G$. Then, for $c_0 \neq d_0 \in L$, the unique branch of the TBBI-tableau $[c_0 \vdash d_0, \mathbb{F}G : c_0]$ is satisfied in (K, \sim, \models, ρ) where $\rho = x \mapsto m$ (in particular $\rho(c_0) = \rho(d_0) = m$). Any initial TBBI-tableau for G is thus realizable. Hence, as branch expansion preserves realizability, all the TBBI-tableaux for G are realizable. Thus G cannot have a closed TBBI-tableau. \square

6. Representing simple PMOs by PME

In Section 4.1, we have presented the map $\sim \mapsto \sqsubseteq_{\sim}^{L, K}$ which transforms a PME into a PMO. We asked ourselves the question of whether this transformation is general enough to produce any PMO. We do not know the answer to this question yet and won't provide an answer in this article. But we have a positive answer in the case of BI-simple PMOs. This is what we develop in this section.

6.1. Elementary and simple PME's

We introduce the notion of BBI-elementary and BBI-simple PME's similarly to the case of PMOs. But there is a major difference between the two cases. BI-elementary (and BI-simple) PMOs were designed to capture those PMOs occurring in TBI-tableau proofs. BBI-elementary (and BBI-simple) PME's are not designed to capture the PME's occurring in TBBI-tableau proofs. They do not fit for such a goal as explained later in Section 6.2. The study of the properties of PME's occurring in TBBI-tableau proofs is way off the scope of this article.

BBI-elementary (and BBI-simple) PME's are in fact specifically designed to represent BI-elementary (and BI-simple) PMOs through the map $\sim \mapsto \sqsubseteq_{\sim}^{L,K}$ and the notion of (L, K, M) elementary representation defined in Section 6.5.

Definition 6.1. Given a PME \sim over L , a constraint is *BBI-elementary w.r.t. \sim* when it is of one of the five following forms:

1. $ab \multimap m$ with $m \sim m$, $m \approx \epsilon$ and $a \neq b \in L \setminus A_{\sim}$;
2. $am \multimap b$ with $m \sim m$ and $a \neq b \in L \setminus A_{\sim}$;
3. $m \multimap b$ with $m \sim m$, $m \approx \epsilon$ and $b \in L \setminus A_{\sim}$;
4. $\epsilon \multimap b$ with $b \in L \setminus A_{\sim}$;
5. $\epsilon \multimap \epsilon$.

Let $(x_i \multimap y_i)_{i < k}$ be a sequence of constraints with $k \in \mathbb{N} \cup \{\infty\}$ and \mathcal{C}_p be the set of constraints $\mathcal{C}_p = \{x_i \multimap y_i \mid i < p\}$ for $p < k$. We suppose that for any $p < k$, the constraint $x_p \multimap y_p$ is BBI-elementary with respect to $\sqsubseteq_{\mathcal{C}_p}$ (resp. $\sim_{\mathcal{C}_p}$). If $k < \infty$ then the sequence $(x_i \multimap y_i)_{i < k}$ is called *BBI-elementary*. If $k = \infty$ then the sequence $(x_i \multimap y_i)_{i < \infty}$ is called *BBI-simple*.

Definition 6.2. A PME is BBI-elementary (resp. BBI-simple) if it is of the form $\sim_{\mathcal{C}}$ where $\mathcal{C} = \{x_i \multimap y_i \mid i < k\}$ and $(x_i \multimap y_i)_{i < k}$ is a BBI-elementary (resp. BBI-simple) sequence of constraints.

We make the obvious following remark. According to those definitions, if \sim be a BI-elementary PME and the constraint $x \multimap y$ is BBI-elementary w.r.t. \sim then the PME extension $\sim + \{x \multimap y\}$ is BBI-elementary. Using case 5 of Definition 6.1, any finite BBI-elementary sequence can be completed into a infinite BBI-simple sequence by repeated use of the constraint $\epsilon \multimap \epsilon$. Hence, BBI-elementary PME's are also BBI-simple.

6.2. A PME occurring in a TBBI-tableau which is not simple

Let us present an example of PME which is not BBI-simple, but nevertheless coming from a branch of a TBBI-tableau: we remind the reader that contrary to BI-elementary and BI-simple PMOs, BBI-elementary and BBI-simple PME's are not designed to capture those PME's generated by TBBI-tableaux.

Consider the set of constraints $\mathcal{C} = \{c_0 \multimap d_0, \epsilon \multimap c_0, ab \multimap c_0\}$. In the given order, it is obvious that this sequence of constraints is not BBI-elementary: $\epsilon \multimap c_0$ is not BBI-elementary w.r.t. $\sim_{\{c_0 \multimap d_0\}} = \{\epsilon \multimap \epsilon, c_0 \multimap c_0, d_0 \multimap d_0, c_0 \multimap d_0\}$ because c_0 is not new. But

this does not prove that the corresponding PME \sim_C is not BBI-elementary or BBI-simple. This sequence of constraints arise as the sequence of assertions of the unique branch of the following TBBI-tableau for $\neg(I \wedge A * B)$:

$$\begin{array}{c}
\text{ass}_0 : c_0 \multimap d_0 \\
\sqrt{1} \text{ F}\neg(I \wedge (A * B)) : c_0 \\
| \\
\sqrt{2} \text{ TI} \wedge (A * B) : c_0 \\
| \\
\sqrt{3} \text{ TI} : c_0 \\
\sqrt{4} \text{ TA} * B : c_0 \\
| \\
\text{ass}_3 : \epsilon \multimap c_0 \\
| \\
\text{ass}_4 : ab \multimap c_0 \\
\text{TA} : a \\
\text{TB} : b
\end{array}$$

It is possible to compute the form of the PME \sim_C explicitly. Indeed, the reader can check the following identity by double inclusion:

$$\sim_C = \{a^{i_0}b^{j_0}x \multimap a^{i_1}b^{j_1}y \mid i_0 + j_1 = i_1 + j_0 \text{ and } x, y \in \{c_0, d_0\}^*\}$$

arguably after a certain amount of work. Then, it is obvious to check that $a^2 \sim_C a^2$ and $a \sim_C \epsilon$. The one letter word a is squarable in \sim_C but nevertheless not equivalent to ϵ . Then, according to Corollary 6.10 (coming later), the PME \sim_C cannot be BBI-simple. It is nevertheless associated to some branch of a TBBI-tableau.

This example points out the conceptual differences between BI-elementary PMOs which capture those PMOs occurring in TBI-tableaux and BBI-elementary PMEs which do not capture those PMEs occurring in TBBI-tableaux. The role played by BBI-elementary PMEs and the justification of the introduction of this concept will become clearer when elementary representations will be introduced in Section 6.5.

6.3. Free PME extensions

To be able to prove further properties of BBI-elementary and BBI-simple PMEs, we now introduce general results which explicitly compute “free” PME extensions like $\sim + \{ab \multimap m\}$ or $\sim + \{am \multimap b\}$ where m is already defined in \sim (i.e. $m \sim m$) and $a \neq b$ are two letters new to \sim (i.e. $a \not\sim a$ and $b \not\sim b$).

The three following results are essential as a basis for reasoning about those “free” PME extensions. The case $\sim + \{\alpha \multimap m\}$ covers both the extension $\sim + \{ab \multimap m\}$ (where a and b are new) and $\sim + \{m \multimap b\}$ (where b is new). Apart for the first one $\sim + \{\epsilon \multimap b\}$, neither the shape of $\sim + \{\alpha \multimap m\}$ nor that of $\sim + \{\alpha m \multimap b\}$ nor the hypotheses on m , α and b are obvious.

Their tedious proofs are done by basic arguments. They are provided to be complete, but are postponed in Appendix A because they are quite long and we feel that they would distract the reader at this point.

Proposition 6.3 ($\sim + \{\epsilon \multimap b\}$). Let \sim be a PME over L and b be new to \sim , i.e. $b \in L \setminus A_\sim$. Then $\sim + \{\epsilon \multimap b\} = \sim'$ with $\sim' = \{b^p x \multimap b^q y \mid x \sim y \wedge p, q \geq 0\}$ and $A_{\sim'} = A_\sim \cup \{b\}$.

Lemma 6.4 ($\sim + \{\alpha \multimap m\}$). Let \sim be a PME over L . Let $m \in L^*$ and $\alpha \in L^*$ such that $m \sim m$, $mm \approx mm$, $\alpha \neq \epsilon$ and $A_\alpha \cap A_\sim = \emptyset$. Then $\sim + \{\alpha \multimap m\} = \sim'$ with

$$\begin{aligned} \sim' = \sim \cup & \{ \delta x \multimap \delta y \mid x \sim y \wedge mx \sim my \wedge \delta \prec \alpha \wedge \delta \notin \{\epsilon, \alpha\} \} \\ & \cup \{ \alpha x \multimap \alpha y \mid mx \sim my \} \\ & \cup \{ \alpha x \multimap y \mid mx \sim y \} \\ & \cup \{ x \multimap \alpha y \mid x \sim my \} \end{aligned}$$

and $A_{\sim'} = A_\sim \cup A_\alpha$.

Lemma 6.5 ($\sim + \{\alpha m \multimap b\}$). Let \sim be a PME over L . Let $m \in L^*$, $\alpha \in L^*$, $b \in L$ such that $m \sim m$, $\alpha \neq \epsilon$, $A_\alpha \cap A_\sim = \emptyset$ and $b \notin A_\sim \cup A_\alpha$. Then $\sim + \{\alpha m \multimap b\} = \sim'$ with

$$\begin{aligned} \sim' = \sim \cup & \{ \delta x \multimap \delta y \mid x \sim y \wedge \epsilon \neq \delta \prec \alpha \wedge \exists k \, xk \sim m \} \\ & \cup \{ \alpha x \multimap jb \mid x \sim jm \wedge \exists k \, jkm \sim m \} \\ & \cup \{ ib \multimap \alpha y \mid y \sim im \wedge \exists k \, ikm \sim m \} \\ & \cup \{ ib \multimap jb \mid \exists k \, (ikm \sim m \wedge jkm \sim m) \} \end{aligned}$$

and $A_{\sim'} = A_\sim \cup A_\alpha \cup \{b\}$.

6.4. No square in simple PMEs

We remark that one of the hypothesis on m in the free PME extension $\sim + \{ab \multimap m\}$ is that the square of m is not defined in \sim , i.e. $mm \approx mm$, see Lemma 6.4 with $\alpha = ab$. To be able to use the equation of this lemma to compute BBI-elementary PMEs, we first establish that they do not contain squares, in a kind of relaxed way.

Definition 6.6. Let \sim be a PME over L . We define $I_\sim = \{i \in L \mid i \sim \epsilon\}$. We say that \sim has no square if for any letter $c \in L$, $cc \sim cc$ only if $c \sim \epsilon$.

This is not exactly the same as stating that no word can be squared unless it is ϵ . But this amounts to stating that no word m can be squared unless it is equivalent to ϵ (i.e. $m \sim \epsilon$). The set I_\sim is the set of letters which are equivalent to ϵ as one letter words.

Proposition 6.7. If the PME \sim over L has no square then the following properties hold:

1. for any $m \in L^*$, we have $mm \sim mm$ iff $m \in I_\sim$ iff $m \sim \epsilon$;
2. for any $i, k, m \in L^*$, if $m \sim m$ then $ikm \sim m$ iff $i \sim \epsilon \wedge k \sim \epsilon$;
3. for any $i, j \in I_\sim$, $m, n \in L^*$, we have $im \sim jn$ iff $m \sim n$.

Proof. Let us define $I = I_\sim$. For property 1, let $m \in L^*$. If $mm \sim mm$, let us consider two cases: $m = \epsilon$ and $m \neq \epsilon$. If $m = \epsilon$ then $m \in I$. If $m \neq \epsilon$, let c be a letter of m . Then $cm' = m$ for some $m' \in L^*$. We get $cm'cm' \sim cm'cm'$, hence $cc \sim cc$ by rule $\langle d \rangle$. So $c \in I$. As $c \in I$ for any letter of m , we deduce $m \in I$. Hence, $mm \sim mm$ only if $m \in I$.

Let us now suppose $m \in I$. We prove that $m \sim \epsilon$ by induction on the length of m . If $m = \epsilon$ then $m \sim \epsilon$ by rule $\langle \epsilon \rangle$. Otherwise $m = cm'$ with $c \in I$ and $m' \in I$. By induction

we have $m' \sim \epsilon$. As $c \in I$, we have $c \sim \epsilon$. By rule $\langle e_l \rangle$, from $\epsilon m' \sim \epsilon$ we deduce $cm' \sim \epsilon$ hence $m \sim \epsilon$. So $m \in I^*$ only if $m \sim \epsilon$.

Let us now suppose $m \sim \epsilon$. Then from $\epsilon m \sim \epsilon$ we deduce $mm \sim \epsilon$ by rule $\langle e_l \rangle$. Hence $mm \sim mm$ by rule $\langle l \rangle$. So we have proved property 1.

For property 2, let $i, k, m \in L^*$ such that $m \sim m$. On the one hand, if $ikm \sim m$ then $ik(ikm) \sim m$ by rule $\langle e_l \rangle$. Hence $(ik)(ik) \sim (ik)(ik)$ by rule $\langle p_l \rangle$. Thus $ik \in I^*$ by property 1. Thus $i, k \in I^*$ and then $i \sim \epsilon$ and $k \sim \epsilon$. On the other hand, if $i \sim \epsilon$ and $k \sim \epsilon$, then from $\epsilon \epsilon m \sim m$, we get $ikm \sim m$ by two applications of rule $\langle e_l \rangle$.

For property 3, we have both $i \sim \epsilon$, $j \sim \epsilon$, $\epsilon \sim i$ and $\epsilon \sim j$ by property 1 and rule $\langle s \rangle$. The equivalence is obtained by application of rules $\langle e_l \rangle$ and $\langle e_r \rangle$. \square

When \sim has no square, the explicit form of the PME extension $\sim + \{am \rightarrow b\}$ can be simplified a bit.

Proposition 6.8. Let \sim be a PME over L , $m \sim m$ and $a \neq b \in L \setminus A_\sim$. If \sim has no square then the following identity holds:

$$\begin{aligned} \sim + \{am \rightarrow b\} = & \sim \cup \{ax \rightarrow ay \mid x \sim y \wedge \exists k \, xk \sim m\} \\ & \cup \{ax \rightarrow jb, jb \rightarrow ax \mid x \sim m \wedge j \sim \epsilon\} \\ & \cup \{ib \rightarrow jb \mid i \sim \epsilon \wedge j \sim \epsilon\} \end{aligned}$$

Proof. Starting from the identity of Lemma 6.5, as \sim has no square, with property 2 of Proposition 6.7, we can simplify the condition $x \sim jm \wedge \exists k \, jkm \sim m$ into the equivalent $x \sim m \wedge j \sim \epsilon$. We can also simplify $\exists k \, (ikm \sim m \wedge jkm \sim m)$ into the equivalent $i \sim \epsilon \wedge j \sim \epsilon$. \square

Now we prove than BBI-elementary extensions preserve the property of “having no square” and compute the sets I_\sim accordingly.

Proposition 6.9. If the PME \sim has no square and $x \rightarrow y$ is BBI-elementary w.r.t. \sim then $\sim' = \sim + \{x \rightarrow y\}$ has no square. Moreover $I_{\sim'} = I_\sim$ in all cases except case 4 ($\sim' = \sim + \{\epsilon \rightarrow b\}$) where $I_{\sim'} = I_\sim \cup \{b\}$.

Proof. Let $A = A_\sim$, $I = I_\sim$, $\sim' = \sim + \{x \rightarrow y\}$, $A' = A_{\sim'}$, $I' = I_{\sim'}$. Then, as $\sim \subseteq \sim'$, we deduce $A \subseteq A'$ and $I \subseteq I'$. We consider each case for $x \rightarrow y$ according to Definition 6.1, using Proposition 6.3, Lemma 6.4 and Proposition 6.8:

$\boxed{ab \rightarrow m}$ where $m \sim m$, $m \not\sim \epsilon$, $a \neq b \in L \setminus A$. As \sim has no square, from $m \not\sim \epsilon$ we deduce $mm \not\sim mm$ (see property 1 of Proposition 6.7) and thus by Lemma 6.4:

$$\begin{aligned} \sim' = \sim + \{ab \rightarrow m\} = & \sim \cup \{ax \rightarrow ay, bx \rightarrow by \mid x \sim y \wedge mx \sim my\} \\ & \cup \{abx \rightarrow aby \mid mx \sim my\} \\ & \cup \{abx \rightarrow y, y \rightarrow abx \mid mx \sim y\} \end{aligned}$$

Let c be a letter such that $cc \sim' cc$. As a and b are two different letters not occurring in $A = A_\sim$, from the form of \sim' it is obvious that the only option is $cc \sim cc$, hence $c \sim \epsilon$ because \sim has no square. Thus $c \sim' \epsilon$. We have proved that \sim' has no square and $I' \subseteq I$ hence $I' = I$;

$\boxed{am \multimap b}$ where $m \sim m$ and $a \neq b \in L \setminus A$. As \sim has no square, by Proposition 6.8, we obtain the identity:

$$\begin{aligned} \sim' &= \sim + \{am \multimap b\} = \sim \cup \{ax \multimap ay \mid x \sim y \wedge \exists k \, xk \sim m\} \\ &\quad \cup \{ax \multimap jb, jb \multimap ax \mid x \sim m \wedge j \sim \epsilon\} \\ &\quad \cup \{ib \multimap jb \mid i \sim \epsilon \wedge j \sim \epsilon\} \end{aligned}$$

Let c be a letter such that $cc \sim' cc$. From the form of \sim' it is obvious that the only option is $cc \sim cc$, hence $c \sim \epsilon$ because \sim has no square. Thus $c \sim' \epsilon$. Hence \sim' has no square and $I' = I$;

$\boxed{m \multimap b}$ where $m \sim m$, $m \approx \epsilon$, $b \in L \setminus A$. As \sim has no square, from $m \approx \epsilon$ we deduce $mm \approx mm$ and thus by Lemma 6.4:

$$\begin{aligned} \sim' &= \sim + \{b \multimap m\} = \sim \cup \{bx \multimap by \mid mx \sim my\} \\ &\quad \cup \{bx \multimap y, y \multimap bx \mid mx \sim y\} \end{aligned}$$

But $\sim' = \sim + \{m \multimap b\} = \sim + \{b \multimap m\}$ by rule $\langle s \rangle$. Let c be a letter such that $cc \sim' cc$. From the form of $\sim' = \sim + \{b \multimap m\}$ it is obvious that the only option is $cc \sim cc$, hence $c \sim \epsilon$ because \sim has no square. Thus $c \sim' \epsilon$. Hence \sim' has no square and $I' = I$;

$\boxed{\epsilon \multimap b}$ with $b \in L \setminus A$. By Proposition 6.3:

$$\sim' = \sim + \{\epsilon \multimap b\} = \{b^p x \multimap b^q y \mid x \sim y \wedge p, q \geq 0\}$$

Let $c \in L$. There are two options for $cc \sim' cc$: either $cc \sim cc$ or $c = b$. In case $cc \sim cc$, we obtain $c \sim \epsilon$ because \sim has no square, hence $c \sim' \epsilon$. In case $c = b$, we get $c = b \sim' \epsilon$ by rule $\langle s \rangle$. Hence \sim' has no square and $I' = I \cup \{b\}$;

$\boxed{\epsilon \multimap \epsilon}$ then $\sim' = \sim + \{\epsilon \multimap \epsilon\} = \sim$ has no square and $I' = I$.

In any case, we have proved that \sim' has no square and computed $I_{\sim'}$ accordingly. \square

Corollary 6.10. BBI-elementary and BBI-simple PME's have no square.

Proof. We prove the result for BBI-simple PME's as BBI-elementary PME's are also BBI-simple PME's. Let \sim be a BBI-simple PME. Then there exists $(x_i \multimap y_i)_{i < \infty}$, a BBI-simple sequence of constraints such that $\sim = \sim_{\mathcal{C}}$ with $\mathcal{C} = \{x_i \multimap y_i \mid i < \infty\}$. Let $\sim_p = \sim_{\mathcal{C}_p}$ with $\mathcal{C}_p = \{x_i \multimap y_i \mid i < p\}$ for $p \in \mathbb{N} \cup \{\infty\}$. Then $\sim = \sim_{\infty}$.

For $p = 0$, we have $\sim_0 = \sim_{\emptyset} = \{\epsilon \multimap \epsilon\}$ which of course has no square. With Proposition 6.9, it is trivial to prove the induction step, i.e. \sim_p has no square implies $\sim_{p+1} = \sim_p + \{x_p \multimap y_p\}$ has no square, because $x_p \multimap y_p$ is BBI-elementary w.r.t. \sim_p . By induction, for any $p < \infty$, the PME \sim_p has no square.

Let $c \in L$ such that $cc \sim_{\infty} cc$. By compactness (see Proposition 3.17), there exists $p < \infty$ such that $cc \sim_p cc$. Hence, as \sim_p has no square, we deduce $c \sim_p \epsilon$. Hence, as $\sim_p \subseteq \sim_{\infty}$, we have $c \sim_{\infty} \epsilon$. Thus $\sim = \sim_{\infty}$ has no square. \square

Let us now briefly discuss the incremental computation of BBI-elementary PME's. This problem consists in the computation of BBI-elementary extensions of \sim where \sim is itself a BBI-elementary PME. Hence, \sim has no square and thus, since cases 1 and 3 of Definition 6.1 contain the condition $m \approx \epsilon$, by Proposition 6.7, we deduce $mm \approx mm$. Hence, it is legitimate to use the equation of Lemma 6.4 to compute the BBI-elementary

extensions $\sim + \{ab \vdash m\}$ and $\sim + \{m \vdash b\}$ as already done in the proof of Proposition 6.9. To complete the description, for case 2, we use Proposition 6.8 and for case 4, we use Proposition 6.3.

6.5. Elementary representations

Having defined BI-elementary PMOs and BBI-elementary PME's, having described how to compute BBI-elementary PME's, we are now in position to state and prove the fundamental lemma of this article. It describes how BI-elementary extensions of BI-elementary PMOs are related to BBI-elementary extensions of BBI-elementary PME's. From this, we deduce a procedure that, given a BI-simple PMO \sqsubseteq over L , computes a BBI-simple PME \sim over $L \cup K \cup M$ such that \sim represents \sqsubseteq , i.e. $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$.

Lemma 6.13 is the fundamental brick on which the adequacy and the faithfulness of our embedding of BI into BBI relies.

Definition 6.11. Let L , K and M be three mutually disjoint alphabets. We say that the pair (\sqsubseteq, \sim) is a (L, K, M) elementary representation if the following properties hold:

1. \sqsubseteq is a BI-elementary PMO over L ;
2. \sim is a BBI-elementary PME over $L \cup K \cup M$;
3. the inclusion $L \sim \subseteq M$ holds;
4. for any $d \in M$, if $d \sim d$ then $x\alpha \sim d$ for some $x \in L^*$ and $\alpha \in K^*$;
5. the identity $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ holds.

Thus, if (\sqsubseteq, \sim) is a (L, K, M) elementary representation and $m \in L^*$ then $m \sqsubseteq m$ if and only if $m \sim m$ (see Lemma 4.2).

Proposition 6.12. If the pair (\sqsubseteq, \sim) is a (L, K, M) elementary representation then $A_{\sqsubseteq} = A_{\sim} \cap L$, $\mathcal{L}^{\sqsubseteq} = L^* \cap \mathcal{L}^{\sim}$ and for any $k \in (L \cup K \cup M)^*$ such that $k \sim k$ there exists $x \in L^*$ and $\alpha \in K^*$ such that $x\alpha \sim k$.

Proof. The identity $A_{\sqsubseteq} = A_{\sim} \cap L$ and $\mathcal{L}^{\sqsubseteq} = L^* \cap \mathcal{L}^{\sim}$ are a direct consequence of Lemma 4.2. Let $k \in (L \cup K \cup M)^*$. There exist $l \in L^*$, $\delta \in K^*$ and $m \in M^*$ such that $l\delta m = k$. Let $m = d_1 \dots d_p$ where $d_1, \dots, d_p \in M$ are the letters of m .

As $k \sim k$, we get $l\delta m \sim l\delta m$ hence $m \sim m$ by rule $\langle d \rangle$. Thus $d_1 \dots d_p \sim d_1 \dots d_p$ and for any $i \in [1, p]$, we have $d_i \sim d_i$. As (\sqsubseteq, \sim) is a (L, K, M) elementary representation, for any $i \in [1, p]$ there exist $x_i \in L^*$ and $\alpha_i \in K^*$ such that $x_i\alpha_i \sim d_i$.

From $l\delta d_1 \dots d_p \sim k$, we get $l\delta(x_1\alpha_1) \dots (x_p\alpha_p) \sim k$ by p applications of rule $\langle e_i \rangle$. Hence we have $x\alpha \sim k$ with $x = lx_1 \dots x_p \in L^*$ and $\alpha = \delta\alpha_1 \dots \alpha_p \in K^*$. \square

Lemma 6.13. Let L , K and M be three mutually disjoint alphabets. Let (\sqsubseteq, \sim) be a (L, K, M) elementary representation. Let m, a, b, δ and c such that $m \sqsubseteq m$, $a \neq b \in L \setminus A_{\sqsubseteq}$, $\delta \in K \setminus A_{\sim}$ and $c \in M \setminus A_{\sim}$. Then in each of the following cases, (\sqsubseteq', \sim') is a (L, K, M)

elementary representation:

1. $\sqsubseteq' = \sqsubseteq + \{ab \multimap m\}$ and $\sim' = \sim + \{\delta c \multimap m, ab \multimap c\}$ when $m \neq \epsilon$
2. $\sqsubseteq' = \sqsubseteq + \{am \multimap b\}$ and $\sim' = \sim + \{cm \multimap b, \delta a \multimap c\}$
3. $\sqsubseteq' = \sqsubseteq + \{m \multimap b\}$ and $\sim' = \sim + \{\delta m \multimap b\}$
- 3'. $\sqsubseteq' = \sqsubseteq + \{m \multimap b\}$ and $\sim' = \sim + \{\delta m \multimap b, \epsilon \multimap \epsilon\}$
4. $\sqsubseteq' = \sqsubseteq + \{\epsilon \multimap m\}$ and $\sim' = \sim + \{\delta c \multimap m, \epsilon \multimap c\}$ when $m \neq \epsilon$
- 4'. $\sqsubseteq' = \sqsubseteq + \{\epsilon \multimap m\}$ and $\sim' = \sim + \{\epsilon \multimap c, m \multimap \delta\}$ when $m \neq \epsilon$

Proof. Here we only provide the proof of case 2 as an illustration of the type of arguments which are involved. The rest of the proof (cases 1, 3, 3', 4 and 4') is postponed in Appendix B because of the overall length of the argument.

In the case of line 2 where $\sqsubseteq' = \sqsubseteq + \{am \multimap b\}$ and $\sim' = \sim + \{cm \multimap b, \delta a \multimap c\}$. First \sqsubseteq' is clearly BI-elementary. The constraint $cm \multimap b$ is obviously BBI-elementary w.r.t. \sim . Then, by Proposition 6.9, $\sim'' = \sim + \{cm \multimap b\}$ has no square and $I_{\sim''} = I_{\sim}$. As $c \notin I_{\sim}$ (because $c \notin A_{\sim}$) then $c \notin I_{\sim''}$, hence $c \sim'' \epsilon$. Thus $\delta a \multimap c$ is BBI-elementary w.r.t. \sim'' . Then $\sim' = \sim'' + \{\delta a \multimap c\}$ is BBI-elementary, has no square and $I_{\sim'} = I_{\sim''} = I_{\sim} \subseteq M$.

We have $A_{\sim'} = A_{\sim} \cup \{a, b, \delta, c\}$. Let $d \in M$ such that $d \sim' d$. Then, either $d \in A_{\sim}$ or $d = c$. On the one hand, if $d \in A_{\sim}$ then $d \sim d$ and let $x \in L^*$ and $\alpha \in K^*$ such that $x\alpha \sim d$. Hence $x\alpha \sim' d$ because $\sim \subseteq \sim'$. On the other hand, if $d = c$ then $a\delta \sim' d$ with $a \in L^*$ and $\delta \in K^*$.

As $cm \sim' b$ and $\delta a \sim' c$, by rule $\langle e_l \rangle$ we obtain $\delta am \sim' b$, hence $am \sqsubseteq_{\sim'}^{L,K} b$. As $\sqsubseteq = \sqsubseteq_{\sim}^{L,K} \subseteq \sqsubseteq_{\sim'}^{L,K}$, we get $\sqsubseteq \cup \{am \multimap b\} \subseteq \sqsubseteq_{\sim'}^{L,K}$ and obtain $\sqsubseteq' \subseteq \sqsubseteq_{\sim'}^{L,K}$.

Let us now consider the converse inclusion $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$ which is the tricky part of the proof. We have the following identities according to Proposition 6.8 and Lemma 6.4 ($c \sim'' \epsilon$ and thus $cc \sim'' cc$):

$$\begin{aligned}
 \sim'' &= \sim + \{cm \multimap b\} = \sim \cup \{cx \multimap cy \mid x \sim y \wedge \exists k \, xk \sim m\} \\
 &\quad \cup \{cx \multimap jb, jb \multimap cx \mid x \sim m \wedge j \sim \epsilon\} \\
 &\quad \cup \{ib \multimap jb \mid i \sim \epsilon \wedge j \sim \epsilon\} \\
 \sim' &= \sim'' + \{\delta a \multimap c\} = \sim'' \cup \{\delta x \multimap \delta y, ax \multimap ay \mid x \sim'' y \wedge cx \sim'' cy\} \\
 &\quad \cup \{\delta ax \multimap \delta ay \mid cx \sim'' cy\} \\
 &\quad \cup \{\delta ax \multimap y, y \multimap \delta ax \mid cx \sim'' y\}
 \end{aligned}$$

Let $\gamma \in K^*$ and $x, y \in L^*$ such that $\gamma x \sim' y$. Let us prove that $x \sqsubseteq' y$. We study each case depending on the form of $(\gamma x, y)$:

- if $\gamma x \sim'' y$ then according to the equations for \sim'' , the only two possibilities are when $\gamma x \sim y$ or $(\gamma x, y) = (ib, jb)$ with $i \sim \epsilon$ and $j \sim \epsilon$ (indeed, otherwise the letter $c \notin L \cup K$ occurs either on the left or on the right). Clearly if $\gamma x \sim y$ then $x \sqsubseteq y$ hence $x \sqsubseteq' y$. Let us study the case where $(\gamma x, y) = (ib, jb)$ with $i \sim \epsilon$ and $j \sim \epsilon$. Then $i, j \in I_{\sim}$. As $I_{\sim} \subseteq M$ and $i \prec \gamma x \in (L \cup K)^*$, we must have $i = \epsilon$. As $j \prec y \in L^*$, we must have $j = \epsilon$. Hence $(\gamma x, y) = (b, b)$. Thus $\gamma = \epsilon$ and $(x, y) = (b, b)$. As $am \sqsubseteq' b$, by rule $\langle r \rangle$ we get $b \sqsubseteq' b$, so $x \sqsubseteq' y$;
- $(\gamma x, y) = (\delta x', \delta y')$ is impossible because $\delta \neq y$ ($\delta \notin L$);
- $(\gamma x, y) = (ax', ay')$ with $x' \sim'' y'$ and $cx' \sim'' cy'$. The only possibility for $cx' \sim'' cy'$ is when $x' \sim y'$ and $x'k \sim m$ for some k . Thus $y'k \sim m$ by rules $\langle s \rangle$ and $\langle e_l \rangle$ and

$k \sim k$ by rule $\langle p_l \rangle$. By Proposition 6.12, there exists $z \in L^*$ and $\alpha \in K^*$ such that $z\alpha \sim k$. Hence, $y'z\alpha \sim m$ by rule $\langle e_l \rangle$. As $y' \prec y \in L^*$, we get $y'z \sqsubseteq m$. As $\gamma x = ax'$ and $a \notin A_\gamma \subseteq K$ then $a \prec x$ and let $ax'' = x$. Hence $x' = \gamma x''$, thus $\gamma x'' \sim y'$ and we obtain $x'' \sqsubseteq y'$. As $\sqsubseteq \subseteq \sqsubseteq'$, we get $y'z \sqsubseteq' m$ and $x'' \sqsubseteq' y'$. Consider the following deduction tree:

$$\frac{\frac{\frac{y'z \sqsubseteq' m \quad am \sqsubseteq' b}{ay'z \sqsubseteq' b} \langle e_l \rangle}{ay' \sqsubseteq' ay'} \langle p_l \rangle \quad x'' \sqsubseteq' y'}{ax'' \sqsubseteq' ay'} \langle c \rangle$$

Hence $x = ax'' \sqsubseteq' ay' = y$;

- $(\gamma x, y) = (\delta ax', \delta ay')$ is impossible because $\delta \not\prec y$;
- $(\gamma x, y) = (\delta ax', y)$ with $cx' \sim'' y$. The only option according to the equations for \sim'' is $(cx', y) = (cx', jb)$ with $x' \sim m$ and $j \sim \epsilon$. Then $j \in I_\sim^*$ hence $j \in M^*$. But $j \prec y \in L^*$. So $j = \epsilon$ and $y = b$. As $\gamma x = \delta ax'$, then $\delta \prec \gamma$ and $a \prec x$. Let $ax'' = x$ and $\delta \gamma' = \gamma$. Then $\delta \gamma' ax'' = \delta ax'$ thus $x' = \gamma' x''$. So $\gamma' x'' \sim m$. As $\gamma' \prec \gamma \in K^*$, $x'' \prec x \in L^*$ and $m \in A_\gamma^* \subseteq L^*$, we deduce $x'' \sqsubseteq m$. Hence $x'' \sqsubseteq' m$. As $am \sqsubseteq' b$, we obtain $ax'' \sqsubseteq' b$ by rule $\langle e_l \rangle$. But $ax'' = x$ and $b = y$. We obtain $x \sqsubseteq' y$;
- $(\gamma x, y) = (\gamma x, \delta ay')$ is impossible because $\delta \not\prec y$;

We have proved that for any $\gamma \in K^*$ and $x, y \in L^*$, if $\gamma x \sim' y$ then $x \sqsubseteq' y$. Thus $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. So, in case of line 2, we have indeed proved that $\sqsubseteq' = \sqsubseteq_{\sim'}^{L,K}$. \square

Remark: even if in case 4, \sim' does not “look” BBI-elementary (because c is not new to $\sim + \{\delta c \prec m\}$), it is in fact BBI-elementary when viewed in the form of case 4' (see proof in Appendix B).

6.6. From simple PMOs to simple PME's

The notion of elementary representation is thus a useful tool to maintain the relation $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$ between the BI-elementary PMO \sqsubseteq and the BBI-elementary \sim when \sqsubseteq is enriched with new BI-elementary constraints. What happens with the limit of increasing sequences of BI-elementary PMOs, i.e. BI-simple PMOs ? The following theorem provides an answer to this question.

Theorem 6.14. Let L be an alphabet and \sqsubseteq be a BI-simple PMO over L . There exist two alphabets K and L' and a BBI-simple PME \sim over L' such that $L \cup K \subseteq L'$ and $\sqsubseteq = \sqsubseteq_{\sim}^{L,K}$.

Proof. Let \sqsubseteq be described by the BI-simple sequence of constraints $(x_i \prec y_i)_{i < \infty}$ over the alphabet L : we have $\sqsubseteq = \sqsubseteq_{\mathcal{C}}$ with $\mathcal{C} = \{x_i \prec y_i \mid i < \infty\}$. Let $\sqsubseteq_p = \sqsubseteq_{\mathcal{C}_p}$ with $\mathcal{C}_p = \{x_i \prec y_i \mid i < p\}$ for $p \in \mathbb{N} \cup \{\infty\}$. Then $\sqsubseteq = \sqsubseteq_\infty$.

Let $K = \{\delta_0, \delta_1, \dots\}$ and $M = \{c_0, c_1, \dots\}$ be two infinite, countable and disjoint sets such that $(K \cup M) \cap L = \emptyset$. Let $L' = L \cup K \cup M$. We build the sequence $(x'_i \prec y'_i)_{i < \infty}$ of constraints over L' according to the following table (the case column refers to the

terminology of Lemma 6.13):

case	$x_i \multimap y_i$	$x'_{2i} \multimap y'_{2i}$	$x'_{2i+1} \multimap y'_{2i+1}$
1	$ab \multimap m$	$\delta_i c_i \multimap m$	$ab \multimap c_i$
2	$am \multimap b$	$c_i m \multimap b$	$\delta_i a \multimap c_i$
3'	$m \multimap b$	$\delta_i m \multimap b$	$\epsilon \multimap \epsilon$
4'	$\epsilon \multimap m$	$\epsilon \multimap c_i$	$m \multimap \delta_i$
	$\epsilon \multimap \epsilon$	$\epsilon \multimap \epsilon$	$\epsilon \multimap \epsilon$

Let $\mathcal{D}_p = \{x'_i \multimap y'_i \mid i < p\}$ with $p \in \mathbb{N} \cup \{\infty\}$ and let $\sim_p = \sim_{\mathcal{D}_p}$. By Lemma 6.13, the reader can check by induction on p that for any $p < \infty$, $(\sqsubseteq_p, \sim_{2p})$ is a (L, K, M) elementary representation. Hence \sim_{2p} is a BBI-elementary PME, $I_{\sim_{2p}} \subseteq M$ and $\sqsubseteq_p = \sqsubseteq_{\sim_{2p}}^{L, K}$.

Let us prove that $\sqsubseteq_\infty = \sqsubseteq_{\sim_\infty}^{L, K}$. If $x \sqsubseteq_\infty y$ then by compactness (see Proposition 3.17), there exists $p < \infty$ such that $x \sqsubseteq_p y$. Then $x \sqsubseteq_{\sim_{2p}}^{L, K} y$. So there exists $\delta \in K^*$ such that $\delta x \sim_{2p} y$. Then $\delta x \sim_\infty y$ as $\sim_{2p} \subseteq \sim_\infty$. Hence $x \sqsubseteq_{\sim_\infty}^{L, K} y$. Conversely, if $x \sqsubseteq_{\sim_\infty}^{L, K} y$ then there exists $\delta \in K^*$ such that $\delta x \sim_\infty y$. By compactness again, there exists $q < \infty$ such that $\delta x \sim_q y$. Then, as $\sim_q \subseteq \sim_{2q}$ (because $\mathcal{D}_q \subseteq \mathcal{D}_{2q}$), we have $\delta x \sim_{2q} y$ hence $x \sqsubseteq_q y$. Thus $x \sqsubseteq_\infty y$ as $\sqsubseteq_q \subseteq \sqsubseteq_\infty$.

So we have proved that $\sqsubseteq = \sqsubseteq_\infty = \sqsubseteq_{\sim_\infty}^{L, K}$ where $\sim_\infty = \sim_{\mathcal{D}_\infty}$ and $\mathcal{D}_\infty = \{x'_i \multimap y'_i \mid i < \infty\}$. The sequence $(x'_i \multimap y'_i)_{i < \infty}$ is a BBI-simple sequence of constraints because for each $p < \infty$, $x_p \multimap y_p$ is BBI-elementary w.r.t. \sim_p (indeed, $p < 2p + 2$ and \sim_{2p+2} is BBI-elementary). Thus \sim_∞ is a BBI-simple PME. \square

7. Soundness of the embedding of BI into BBI

The map $F \mapsto F^\circ$ looks like a good candidate for embedding BI into BBI. Indeed, given an invalid formula F of BI, by Theorem 5.9 it is possible to obtain a counter-model of F of the form of a BI-simple PMO \sqsubseteq over some language L , i.e. $\epsilon \not\models_\sqsubseteq F$. Then by Theorem 6.14, there exists a (BBI-simple) PME \sim such that $\sqsubseteq = \sqsubseteq_{\sim}^{L, K}$. Thus, by Theorem 4.4, we obtain $\epsilon \not\models_\sim F^\circ$, hence we obtain a counter-model of F° .

It may seem that we have our embedding but unfortunately, F° is not necessarily BBI-valid when F is BI-valid. The mapping $F \mapsto F^\circ$ is not exactly the embedding we are looking for. It preserves counter-models but does not preserve provability.

Indeed, nothing in F° captures the special roles played by the two spare variables L and K . We have to incorporate some information on L and K that logically encodes the way they are interpreted in the particular model of Theorem 4.4 where they are forced by words belonging to sub-languages generated by a sub-alphabets, respectively L^* and K^* . So let H be the following formula:

$$H \equiv (L \wedge K) \wedge ((\top \multimap (L * L \rightarrow L)) \wedge (\top \multimap (K * K \rightarrow K)))$$

For example, the sub-formula $\top \multimap (L * L \rightarrow L)$ encodes the property that the decomposition of words forcing L yields words forcing L , a subword property typical of sub-languages generated by sub-alphabets.

We are going to state and prove that $(I \wedge H) \rightarrow G^\circ$ is BBI-valid whenever G is BI-valid. And then we will prove that $(I \wedge H) \rightarrow G^\circ$ is BBI-invalid whenever G is BI-invalid.

7.1. From TBI-tableaux to TBBI-tableaux

In this section we describe how to process a TBI-tableau for G and obtain a corresponding TBBI-tableau for $(I \multimap (I \wedge H)) \rightarrow (I \multimap G^\circ)$. We have chosen this translation instead of simpler $(I \wedge H) \rightarrow G^\circ$ (those two formulae are logically equivalent in BBI, see Proposition 7.5) because for the former one, we can provide a direct tableau translation procedure as described in the following results.

Lemma 7.1. Let $L' = L \cup K \cup M$ be a partition of L' where K and M are two disjoint infinite sets of spare letters. Let \mathcal{T} be a TBI-tableau for $G \in \text{BI}$ over L . There exists a TBBI-tableau \mathcal{T}' for $(I \multimap (I \wedge H)) \rightarrow (I \multimap G^\circ)$ over L' and an injective map φ from the branches of \mathcal{T} to the branches of \mathcal{T}' such that:

1. each branch γ' of \mathcal{T}' either contains $\mathbb{F}I : \epsilon$ or the following set of statements:

$$\begin{aligned} & \{\mathbb{T}L : \epsilon, \mathbb{T}T \multimap (L * L \rightarrow L) : \epsilon\} \cup \{\mathbb{T}K : \epsilon, \mathbb{T}T \multimap (K * K \rightarrow K) : \epsilon\} \\ \cup & \{\mathbb{T}L : a \mid a \in L \cap A_{\gamma'}\} \quad \cup \quad \{\mathbb{T}K : \delta_0 \mid \delta_0 \in K \cap A_{\gamma'}\} \end{aligned}$$

2. for each branch γ of \mathcal{T} , with $\gamma' = \varphi(\gamma)$, the two following conditions hold:

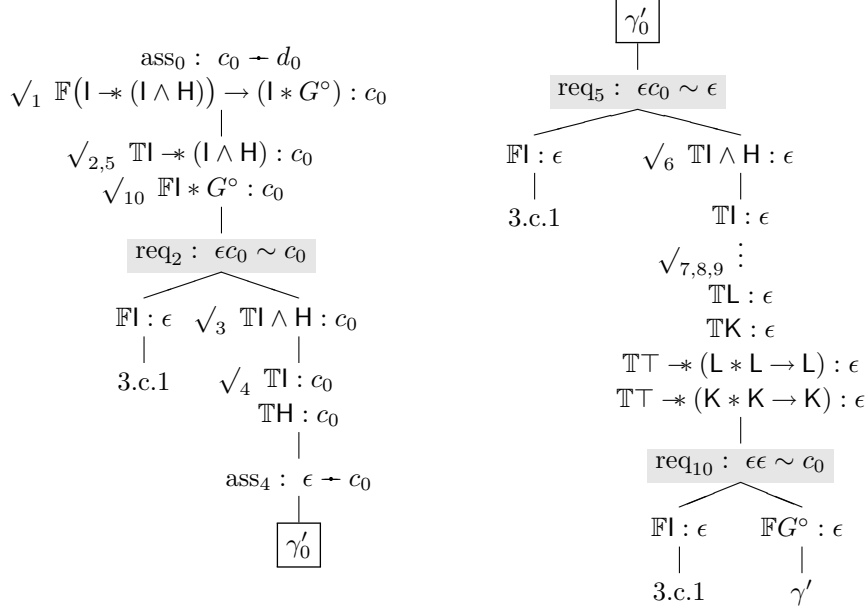
- 2.1. $(\sqsubseteq_\gamma, \sim_{\gamma'})$ is a (L, K, M) elementary representation;
- 2.2. for every statement $\mathbb{S}A : m$ of γ , the statement $\mathbb{S}A^\circ : m$ occurs in γ' ;

3. for each other branch γ' of \mathcal{T}' (i.e. which is not the image $\varphi(\gamma)$ of some branch γ of \mathcal{T}), at least one of the following conditions hold:

- 3.c.1. $\mathbb{F}I : \epsilon \in \gamma'$;
- 3.c.2. $\mathbb{F}L : l \in \gamma'$ for some $l \in L^*$;
- 3.c.3. $\mathbb{F}K : \delta \in \gamma'$ for some $\delta \in K^*$.

Proof. We build \mathcal{T}' and φ by following the construction process of \mathcal{T} , the TBI-tableau of G . Let us consider the initial TBI-tableau $\llbracket FG : \epsilon \rrbracket$. Let us choose two letters $c_0 \neq d_0 \in M$. Here is a TBBI-tableau for $(I \multimap (I \wedge H)) \rightarrow (I \multimap G^\circ)$ which fulfills the conditions 1, 2 and 3. It is split into two parts for a reasonable display and these two parts should be glued at

point γ'_0 by the reader:

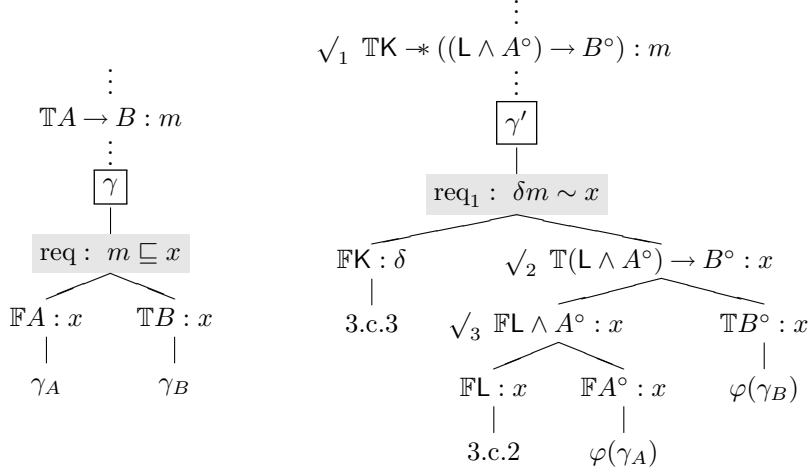


The reader is invited to check that this is indeed a valid TBBI-tableau. We point out that the formula $H \equiv (L \wedge K) \wedge ((\top \multimap (L * L \rightarrow L)) \wedge (\top \multimap (K * K \rightarrow K)))$ is developed at history checkpoints 7, 8 and 9 by three applications of expansion rule $\mathbb{T}\wedge$ (actually not displayed in the TBBI-tableau to shorten its length). This TBBI-tableau has four branches, three of which stopping at $\mathbb{F}I : \epsilon$ (hence fulfilling conditions 1 and 3.c.1), and the last one containing $\mathbb{F}G^o : \epsilon$ as required by condition 2.2. This is the branch $\gamma' = \varphi(\gamma)$ associated through φ to the unique branch γ of the one node TBI-tableau $[\mathbb{F}G : \epsilon]$. γ' obviously fulfills condition 1 because $A_{\gamma'} = \{c_0, d_0\} \subseteq M$ hence $A_{\gamma'} \cap (L \cup K) = \emptyset$. It fulfills condition 2.1 for the following reasons: for $\mathcal{C} = \{\epsilon \multimap c_0, \epsilon \multimap d_0\}$ and $\mathcal{C}_{\gamma'} = \{c_0 \multimap d_0, \epsilon \multimap c_0\}$, we have $\sim_{\mathcal{C}} = \sim_{\mathcal{C}_{\gamma'}}$. Hence, $\sqsubseteq_{\gamma} = \sqsubseteq_{\emptyset} = \{\epsilon \multimap \epsilon\}$ and $\sim_{\gamma'} = \sim_{\mathcal{C}}$ where $\mathcal{C} = \{\epsilon \multimap c_0, \epsilon \multimap d_0\}$ and $c_0, d_0 \in M$. Thus $\sim_{\gamma'} = \{x \multimap y \mid x, y \in \{c_0, d_0\}^*\}$ and we can check that $\sqsubseteq_{\gamma} = \sqsubseteq_{\sim_{\gamma'}}^{L, K}$. So $(\sqsubseteq_{\gamma}, \sim_{\gamma'})$ forms a (L, K, M) elementary representation.

If \mathcal{T} is a TBI-tableau of G obtained by expansion using rule $\mathbb{T}\wedge$, $\mathbb{F}\wedge$, $\mathbb{T}\vee$ or $\mathbb{F}\vee$, we trivially use the same expansion rule for \mathcal{T}' and fix the mapping φ accordingly.

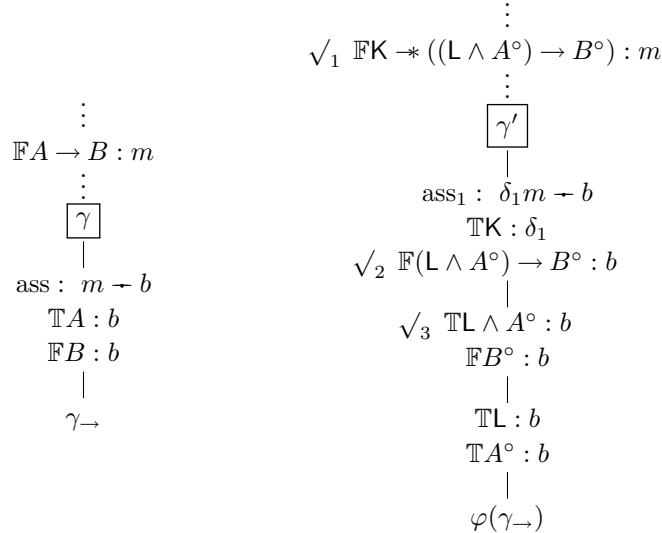
If \mathcal{T} is obtained by expansion of $\mathbb{T}A \rightarrow B : m$ in branch γ , then the requirement $m \sqsubseteq_{\gamma} x$ must hold and then there exists $\delta \in K^*$ such that $\delta m \sim_{\gamma'} x$ holds. We extend γ' the

following way:



and we extend φ so that the branch containing $\mathbb{F}A : x$ (resp. $\mathbb{T}B : x$) in \mathcal{T} corresponds to the branch of $\mathbb{F}A^o : x$ (resp. $\mathbb{T}B^o : x$) in \mathcal{T}' . No assertion is generated so $(\sqsubseteq_\gamma, \sim_{\gamma'})$ does not change on either branches $(\gamma_A, \varphi(\gamma_A))$ or branches $(\gamma_B, \varphi(\gamma_B))$ and is thus still an elementary representation. We also see that the two remaining branches contain respectively $\mathbb{F}K : \delta$ with $\delta \in K^*$ and $\mathbb{F}L : x$ with $x \in L^*$ fulfilling condition 3.c.3 and 3.c.2 respectively.

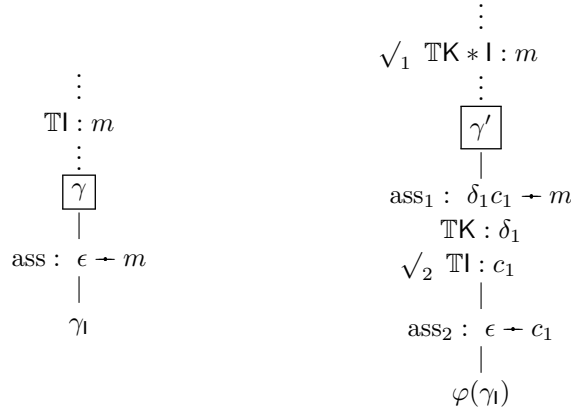
If \mathcal{T} is obtained by expansion of $\mathbb{F}A \rightarrow B : m$ in branch γ , then $b \in L \setminus A_\gamma$. As $A_\gamma = A_{\gamma'} \cap L$ we deduce $b \in L \setminus A_{\gamma'}$. Let us choose $\delta_1 \in K \setminus A_{\gamma'}$ (possible because K is infinite and $A_{\gamma'}$ is finite as $\sim_{\gamma'}$ is BBI-elementary) and apply the following expansion rules to the branch γ' :



Then $(\sqsubseteq_\gamma + \{m \rightarrow b\}, \sim_{\gamma'} + \{\delta_1 m \rightarrow b\})$ is an elementary representation according to

Lemma 6.13 case 3 and φ is extended in the obvious way. We also observe that $\mathbb{TK} : \delta_1$ and $\mathbb{TL} : b$ are introduced fulfilling condition 1.

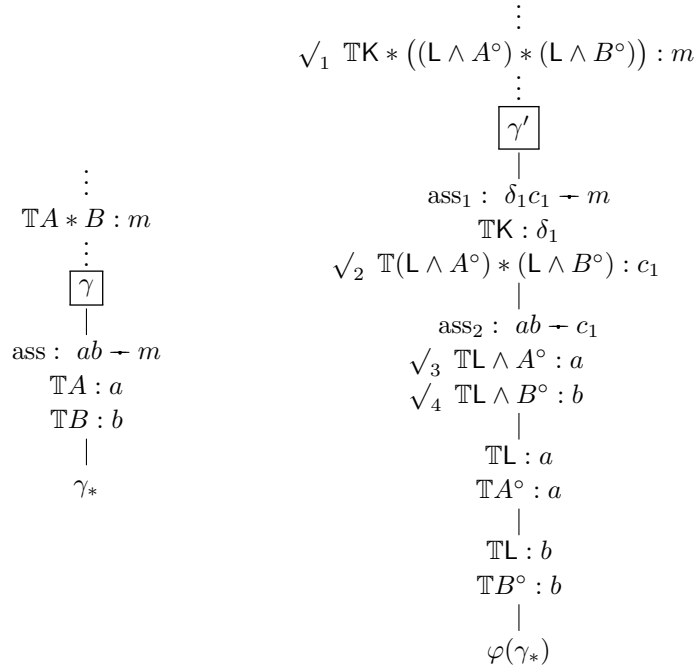
If \mathcal{T} is a TBI-tableau of G obtained by expanding $\mathbb{TL} : m$ on branch γ , then let $\gamma' = \varphi(\gamma)$ and choose $\delta_1 \in K \setminus A_{\gamma'}$ and $c_1 \in M \setminus A_{\gamma'}$ and then apply the following expansion rules to the branch γ' :



Then $(\sqsubseteq_\gamma + \{\epsilon \multimap m\}, \sim_{\gamma'} + \{\delta_1 c_1 \multimap m, \epsilon \multimap c_1\})$ forms an elementary representation according to Lemma 6.13 case 4 because $m \neq \epsilon$ (Proposition 5.7 applied to \mathcal{T}). We also observe that the statement $\mathbb{TK} : \delta_1$ is introduced which fulfills condition 1.

If \mathcal{T} is obtained by expansion of $\mathbb{TA} * B : m$ in branch γ then a, b must have been chosen such that $a \neq b \in L \setminus A_\gamma$. As $(\sqsubseteq_\gamma, \sim_{\gamma'})$ is an elementary representation, we have $A_\gamma = A_{\gamma'} \cap L$ (see Proposition 6.12) and hence $a \neq b \in L \setminus A_{\gamma'}$. Let us choose $\delta_1 \in K \setminus A_{\gamma'}$ and $c_1 \in M \setminus A_{\gamma'}$ and apply the following expansion rules (on the right hand side) to the

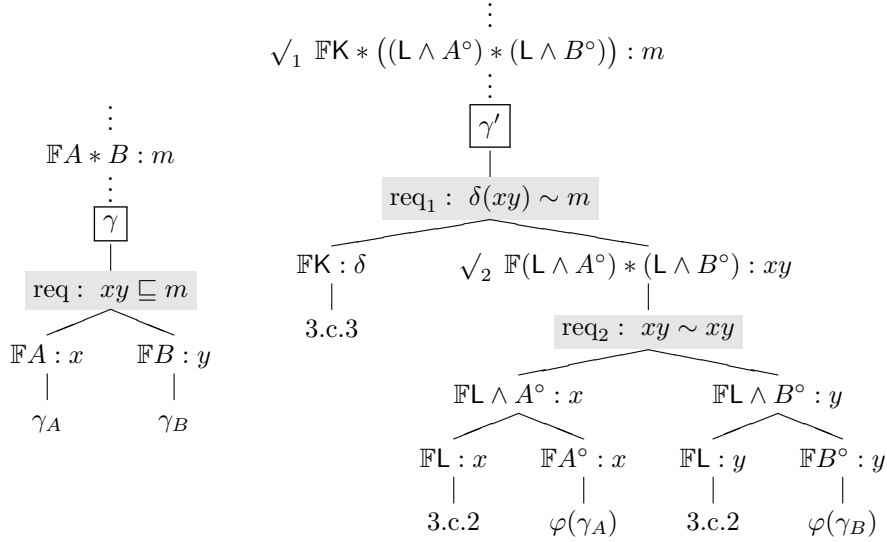
branch γ' :



Then $(\sqsubseteq_\gamma + \{ab \multimap m\}, \sim_{\gamma'} + \{\delta_1 c_1 \multimap m, ab \multimap c_1\})$ is an (L, K, M) elementary representation according to Lemma 6.13 case 1, having checked that $m \neq \epsilon$ (Proposition 5.7 applied to \mathcal{T}). The injective map φ is extended the obvious way since there is only one new branch and this branch fulfills condition 2. We also observe that $\mathbb{T}K : \delta_1$, $\mathbb{T}\mathbb{L} : a$ and $\mathbb{T}\mathbb{L} : b$ are introduced fulfilling condition 1.

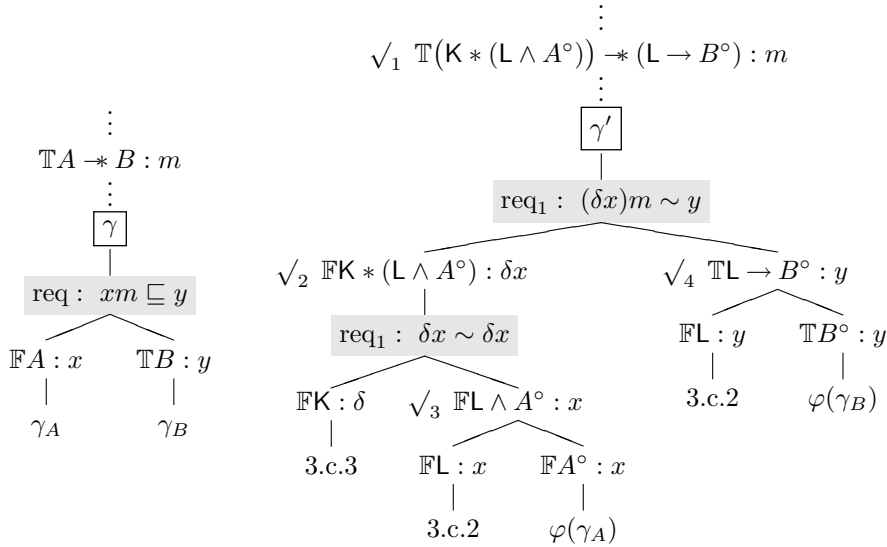
If \mathcal{T} is obtained by expansion of $\mathbb{F}A * B : m$ in branch γ then the requirement $xy \sqsubseteq_\gamma m$ must hold. As $\sqsubseteq_\gamma = \sqsubseteq_{\sim_{\gamma'}}^{L,K}$, there exists $\delta \in K^*$ such that $\delta xy \sim_{\gamma'} m$. We extend γ' the

following way (on the right hand side):



Then, whichever new branch is considered, no new assertion is introduced hence condition 1 is fulfilled and $(\sqsubseteq_\gamma, \sim_{\gamma'})$ is unchanged and thus still an elementary representation. We extend φ so that the branch containing $\mathbb{F}A : x$ (resp. $\mathbb{F}B : y$) in \mathcal{T} corresponds to the branch of $\mathbb{F}A^\circ : x$ (resp. $\mathbb{F}B^\circ : y$) in \mathcal{T}' . Hence condition 2 is fulfilled. We also observe that the three remaining branches contain respectively $\mathbb{F}K : \delta$ with $\delta \in K^*$, $\mathbb{F}L : x$ with $x \in L^*$ and $\mathbb{F}L : y$ with $y \in L^*$ fulfilling condition 3.

If \mathcal{T} is obtained by expansion of $\mathbb{T}A \multimap B : m$ in branch γ then the requirement $xm \sqsubseteq_\gamma y$ must hold. Then there exists $\delta \in K^*$ such that $\delta xm \sim_{\gamma'} y$. We extend γ' the following way:



Then $(\sqsubseteq_\gamma, \sim_{\gamma'})$ is unchanged and still an elementary representation. We extend φ so that the branch containing $\mathbb{F}A : x$ (resp. $\mathbb{T}B : y$) in \mathcal{T} corresponds to the branch of $\mathbb{F}A^\circ : x$ (resp. $\mathbb{T}B^\circ : y$) in \mathcal{T}' . We also see that the three remaining branches contain respectively $\mathbb{F}K : \delta$ with $\delta \in K^*$, $\mathbb{F}L : x$ with $x \in L^*$ and $\mathbb{F}L : y$ with $y \in L^*$ fulfilling condition 3.c.3, 3.c.2 and 3.c.2 respectively.

If \mathcal{T} is obtained by expansion of $\mathbb{F}A \multimap B : m$ in branch γ , then $a \neq b \in L \setminus A_\gamma$. As $(\sqsubseteq_\gamma, \sim_{\gamma'})$ is an elementary representation, we deduce $a \neq b \in L \setminus A_{\gamma'}$. Let us choose $\delta_1 \in K \setminus A_{\gamma'}$ and $c_1 \in M \setminus A_{\gamma'}$ and apply the following expansion rules to γ' :

$$\begin{array}{c}
 \vdots \\
 \sqrt{1} \mathbb{F}(K * (L \wedge A^\circ)) \multimap (L \rightarrow B^\circ) : m \\
 \vdots \\
 \boxed{\gamma'} \\
 \text{ass}_1 : c_1 m \multimap b \\
 \sqrt{2} \mathbb{T}K * (L \wedge A^\circ) : c_1 \\
 \sqrt{4} \mathbb{F}L \rightarrow B^\circ : b \\
 \text{ass}_2 : \delta_2 a \multimap c_1 \\
 \mathbb{T}K : \delta_2 \\
 \sqrt{3} \mathbb{T}L \wedge A^\circ : a \\
 \mathbb{T}L : a \\
 \mathbb{T}A^\circ : a \\
 \mathbb{T}L : b \\
 \mathbb{F}B^\circ : b \\
 \varphi(\gamma_*)
 \end{array}
 \quad
 \begin{array}{c}
 \vdots \\
 \mathbb{F}A \multimap B : m \\
 \vdots \\
 \boxed{\gamma} \\
 \text{ass} : am \multimap b \\
 \mathbb{T}A : a \\
 \mathbb{F}B : b \\
 \gamma_*
 \end{array}$$

Then $(\sqsubseteq_\gamma + \{am \multimap b\}, \sim_{\gamma'} + \{c_1 m \multimap b, \delta_1 a \multimap c_1\})$ is an elementary representation according to Lemma 6.13 case 2, having checked that $m \neq \epsilon$ (Proposition 5.7 applied to \mathcal{T}). φ is extended the obvious way. We also observe that $\mathbb{T}K : \delta_1$, $\mathbb{T}L : a$ and $\mathbb{T}L : b$ are introduced fulfilling condition 1. \square

7.2. From BI-proofs to BBI-proofs

We showed how a TBI-tableau can be transformed into a TBBI-tableau. Now we show that a closed TBI-tableau (i.e. a proof) can be transformed into a closed TBBI-tableau, thus obtaining the soundness part of our embedding. This is done in two steps, first obtain a pseudo-closed TBBI-tableau and then close the pseudo-closed TBBI-tableau.

Definition 7.2. A (L, K) pseudo-closed TBBI-tableau is a TBBI-tableau \mathcal{T} in which every open branch γ verifies the two following conditions:

1. γ contains the following set of statements:

$$\begin{aligned} & \{\mathbb{T}\mathbb{L} : \epsilon, \mathbb{T}\mathbb{T} \multimap (\mathbb{L} * \mathbb{L} \rightarrow \mathbb{L}) : \epsilon\} \cup \{\mathbb{T}\mathbb{K} : \epsilon, \mathbb{T}\mathbb{T} \multimap (\mathbb{K} * \mathbb{K} \rightarrow \mathbb{K}) : \epsilon\} \\ & \cup \{\mathbb{T}\mathbb{L} : a \mid a \in L \cap A_\gamma\} \quad \cup \{\mathbb{T}\mathbb{K} : \delta_0 \mid \delta_0 \in K \cap A_\gamma\} \end{aligned}$$

3. γ verifies at least one of the two following conditions:

3.c.2 $\mathbb{F}\mathbb{L} : l \in \gamma$ for some $l \in L^*$;

3.c.3 $\mathbb{F}\mathbb{K} : \delta \in \gamma$ for some $\delta \in K^*$.

Remark: notice that we have conserved the same terminology as in Lemma 7.1 for the conditions that remain. This explains the hole in the numbering of the conditions.

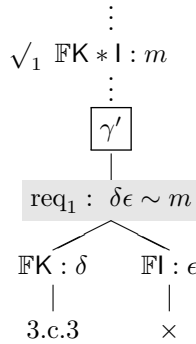
Proposition 7.3. If the formula $G \in \text{BI}$ has a closed TBI-tableau over the alphabet L then the formula $(\mathbb{I} \multimap (\mathbb{I} \wedge \mathbb{H})) \rightarrow (\mathbb{I} * G^\circ) \in \text{BBI}$ has a (L, K) pseudo-closed TBBI-tableau for some alphabet K .

Proof. Let \mathcal{T} be a closed TBI-tableau for G . According to Lemma 7.1, we build a corresponding TBBI-tableau \mathcal{T}' for $G' = (\mathbb{I} \multimap (\mathbb{I} \wedge \mathbb{H})) \rightarrow (\mathbb{I} * G^\circ)$ over $L \cup K \cup M$ and the injective map φ from (maximal) branches of \mathcal{T} to (maximal) branches of \mathcal{T}' . Since \mathcal{T}' verifies condition 1 of Lemma 7.1, then each branch γ' of \mathcal{T}' verifies condition 1 of Definition 7.2 because $\mathbb{F}\mathbb{I} : \epsilon \in \gamma'$ is a closure condition for TBBI-tableaux branches.

There are two kinds of branches in \mathcal{T}' . Those which are images $\gamma' = \varphi(\gamma)$ of branches of \mathcal{T} and those which are not. Let us consider the latter case. According to condition 3 of Lemma 7.1, such a branch is either closed by $\mathbb{F}\mathbb{I} : \epsilon \in \gamma'$ or satisfies conditions 3.c.2 or 3.c.3. In any case, the open branches which are not of the form $\gamma' = \varphi(\gamma)$ verify condition 3.c.2 or 3.c.3 of Definition 7.2.

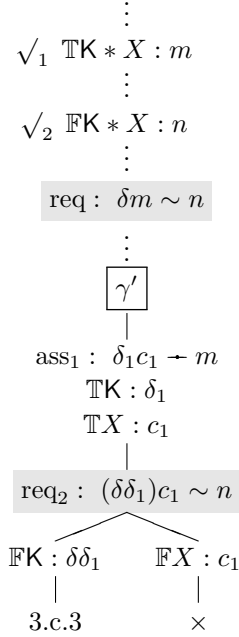
Let us consider the former case which are branches of the form $\gamma' = \varphi(\gamma)$. Since \mathcal{T} is closed, each of its branches are closed, γ in particular. The branch γ is closed by one of the following conditions:

- if $\mathbb{T}\mathbb{I} : m \in \gamma$ then $\mathbb{T}\mathbb{I}^\circ : m \in \gamma'$ and as $\mathbb{I}^\circ = \mathbb{I}$, the branch γ' is closed;
- for the same reason, if $\mathbb{F}\mathbb{T} : m \in \gamma$ then $\mathbb{F}\mathbb{T} : m \in \gamma'$ and γ' is closed;
- if $\mathbb{F}\mathbb{I} : m \in \gamma$ and $\epsilon \sqsubseteq_\gamma m$ then $\mathbb{F}\mathbb{K} * \mathbb{I} : m \in \gamma'$ (because $\mathbb{I}^\circ = \mathbb{K} * \mathbb{I}$) and there exists $\delta \in K^*$ such that $\delta\epsilon \sim_{\gamma'} m$. Then we apply the following branch expansion rules:



and we replace γ' with two branches, one of which is closed by $\mathbb{F}\mathbb{I} : \epsilon$ and the other one satisfying condition 3.c.3 of Definition 7.2. We also observe that condition 1 is still fulfilled by the left branch because, should it remain open, no new assertion is introduced, and thus the alphabet of the branch is left unchanged;

- if $\mathbb{TK} : m \in \gamma$, $\mathbb{FK} : n \in \gamma$ and $m \sqsubseteq_\gamma n$ then $\mathbb{TK} * X : m \in \gamma$, $\mathbb{FK} * X : n \in \gamma$ and there exists $\delta \in K^*$ such that $\delta m \sim_{\gamma'} n$. We choose $\delta_1 \in K \setminus A_{\gamma'}$ and $c_1 \in M \setminus A_{\gamma'}$ and then apply the following branch expansion rules:



We check that $(\delta \delta_1) c_1 \sim_{\gamma''} n$ by rule $\langle e_l \rangle$ with $\gamma'' = [\gamma', \delta_1 c_1 \dashv m, \mathbb{TK} : \delta_1, \mathbb{TX} : c_1]$. γ' is replaced by two branches $\gamma'_l = [\gamma'', \mathbb{FK} : \delta \delta_1]$ and $\gamma'_r = [\gamma'', \mathbb{FK} : c_1]$, the latter γ'_r being closed by $\mathbb{TX} : c_1$ and $\mathbb{FK} : c_1$, the former γ'_l satisfying condition 3.c.3 of Definition 7.2, should it be open. We also remark that the statement $\mathbb{TK} : \delta_1$ is introduced in γ'' . So condition 1 is fulfilled because the alphabet $A_{\gamma'_l}$ of the left branch verifies the equation $A_{\gamma'_l} = A_{\gamma'} \cup \{\delta_1, c_1\}$.

Applying those transformations for every branch $\gamma' = \varphi(\gamma)$ of \mathcal{T}' , we obtain a TBBI-tableau \mathcal{T}'' in which every open branch satisfies condition 1 and either condition 3.c.2 or 3.c.3 of Definition 7.2. Hence \mathcal{T}'' is a (L, K) pseudo-closed TBBI-tableau. \square

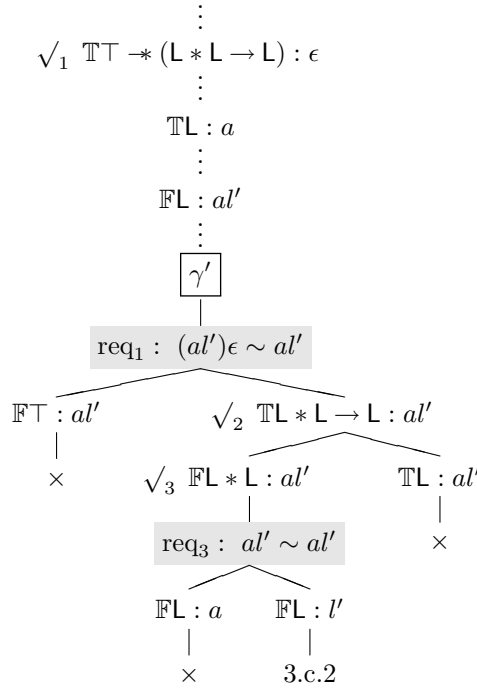
Proposition 7.4. If a formula of BBI has a (L, K) pseudo-closed TBBI-tableau then it has a closed TBBI-tableau.

Proof. Let us define the *weight* of a branch γ' by 0 for a closed branch and the length of the shortest word $x \in A_{\gamma'}^*$ such that either $(x \in L^*$ and $\mathbb{FL} : x \in \gamma')$ or $(x \in K^*$ and $\mathbb{FK} : x \in \gamma')$ otherwise. The weight exists because condition 3.c.2 or 3.c.3 is fulfilled for any open branch of a (L, K) pseudo-closed TBBI-tableau. The *weight of a whole (L, K) pseudo closed TBBI-tableau* is the sum of the weights of all its branches.

Let us prove that any (L, K) pseudo-closed TBBI-tableau can be expanded into a closed TBBI-tableau by induction on the weight of the tableau:

- if the weight of the tableau is 0. Let us consider any branch γ' . γ' must have weight 0

- (because weights are positive numbers) and then, either γ' is closed or $\mathbb{F}\mathbb{L} : \epsilon \in \gamma'$ or $\mathbb{F}\mathbb{K} : \epsilon \in \gamma'$, ϵ being the only word of length 0. Reasoning by contradiction, if γ' is not closed then by condition 1, $\mathbb{T}\mathbb{L} : \epsilon$ and $\mathbb{T}\mathbb{K} : \epsilon$ occur in γ' . Hence γ' is closed because either $\mathbb{F}\mathbb{L} : \epsilon$ or $\mathbb{F}\mathbb{K} : \epsilon$ occur in γ' . So if the weight of the tableau is 0 then the tableau is closed because all its branches are closed. Thus there is no need to expand it;
- if the weight of the tableau is not 0. Let us choose a branch γ' of strictly positive weight $p > 0$. Then γ' is an open branch and let $\mathbb{F}\mathbb{L} : l$ occur in γ' with $l \in L^*$ of length p (the case $\mathbb{F}\mathbb{K} : \delta$ with $\delta \in K^*$ of length p is treated similarly). As $p > 0$, we write $l = al'$ with $a \in L$ and $l' \in L^*$. As $\mathbb{F}\mathbb{L} : l$ occurs in γ' , we must have $l \sim_{\gamma'} l$ by Proposition 5.3, hence $al' \sim_{\gamma'} al'$. Then $a \in L \cap A_{\gamma'}$ and by condition 1, the statements $\mathbb{T}\mathbb{L} : a$ and $\mathbb{T}\mathbb{T} \multimap (L * L \rightarrow L) : \epsilon$ both occur in γ' . We apply to the branch γ' the expansion rules described by:



The order in which the statements $\mathbb{T}\mathbb{T} \multimap (L * L \rightarrow L) : \epsilon$, $\mathbb{T}\mathbb{L} : a$ and $\mathbb{F}\mathbb{L} : al'$ occur in the branch γ' is of no importance. The branch γ' is expanded into four branches, three of which are closed, thus of weight 0, and the last one containing $\mathbb{F}\mathbb{L} : l'$, thus of weight lower than $p-1$ which is the length of l' . The TBBI-tableau obtained after such branch expansion thus has a strictly lower weight. It is also an (L, K) pseudo-closed TBBI-tableau because no assertion is inserted, hence condition 1 is still fulfilled, and the only new branch which is potentially open verifies condition 3.c.2. By induction hypothesis, this TBBI-tableau can itself be expanded into a closed TBBI-tableau.

Thus, we have proved by induction on the weight that any (L, K) pseudo-closed TBBI-tableau can be expanded into a closed TBBI-tableau. \square

Proposition 7.5. For any two BBI-formulae A and B , the formulae $A \rightarrow B$ and $(\mathbf{I} \multimap A) \rightarrow (\mathbf{I} \multimap B)$ are logically equivalent in BBI.

Proof. Let (L, \sim, \Vdash) be a BBI-frame. For any $m \in \mathcal{L}^\sim$, by monotonicity of \Vdash we can show that $m \Vdash A$ holds if and only if $m \Vdash \mathbf{I} \multimap A$ holds, and that $m \Vdash B$ holds if and only if $m \Vdash \mathbf{I} \multimap B$. Thus $m \Vdash A \rightarrow B$ holds if and only if $m \Vdash (\mathbf{I} \multimap A) \rightarrow (\mathbf{I} \multimap B)$ holds. \square

Theorem 7.6 (Soundness of the embedding). Let G be a BI-formula not containing the spare logical variables \mathbf{L} and \mathbf{K} . If G is BI-valid then $(\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ is BBI-valid.

Proof. Let $G' = (\mathbf{I} \multimap (\mathbf{I} \wedge \mathbf{H})) \rightarrow (\mathbf{I} \multimap G^\circ)$. If G is BI-valid then, by completeness of the TBI-tableau system (see Theorem 5.8), G has a closed TBI-tableau. Then, according to Proposition 7.3, G' has a (L, K) pseudo-closed TBBI-tableau. So, by Proposition 7.4, G' has a closed TBBI-tableau, hence is BBI-valid by soundness of the TBBI-tableau system (see Theorem 5.14). Hence $(\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ is BBI-valid because it is BBI-equivalent to G' by Proposition 7.5 (with $A = \mathbf{I} \wedge \mathbf{H}$ and $B = G^\circ$). \square

8. Faithfulness by counter-model transformation

We have proved that if the BI-formula G is BI-valid then the formula $(\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ is BBI-valid. Let us show that if G has a BI-counter-model then $(\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ has a BBI-counter-model.

8.1. From BI-counter-models to BBI-counter-models

Theorem 8.1 (Faithfulness). Let G be a formula of BI not containing the variables \mathbf{L} or \mathbf{K} . If G has a BI-counter-model then the BBI-formula $(\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ has a BBI-counter-model.

Proof. Let us suppose that G has a BI-counter-model. Then it is BI-invalid and by Theorem 5.9, it is possible to obtain a counter-model of G of the form of a BI-simple PMO \sqsubseteq over some language L . So let $(L, \sqsubseteq, \Vdash_\sqsubseteq)$ be a BI-frame where \sqsubseteq is a BI-simple PMO and such that $\epsilon \not\Vdash_\sqsubseteq G$. By Theorem 6.14, there exists a BBI-simple PME \sim over L' such that $L \cup K \subseteq L'$ and $\sqsubseteq = \sqsubseteq_{\sim}^{L, K}$. Thus, by Theorem 4.4, in the BBI-frame (L', \sim, \Vdash_\sim) , we have $\epsilon \not\Vdash_\sim G^\circ$.

In the frame (L', \sim, \Vdash_\sim) , we observe that $\epsilon \Vdash_\sim \mathbf{L}$ and $\epsilon \Vdash_\sim \mathbf{K}$. Let us prove that $\epsilon \Vdash_\sim \mathbf{T} \multimap (\mathbf{L} \ast \mathbf{L} \rightarrow \mathbf{L})$. Let $x\epsilon \sim y$, $x \Vdash_\sim \mathbf{T}$ and $y \Vdash_\sim \mathbf{L} \ast \mathbf{L}$. Let us prove $y \Vdash_\sim \mathbf{L}$. There exists y_1, y_2 such that $y_1 y_2 \sim y$, $y_1 \Vdash_\sim \mathbf{L}$ and $y_2 \Vdash_\sim \mathbf{L}$. Thus, by definition of $(\cdot) \Vdash_\sim \mathbf{L}$, there exists $l_1, l_2 \in L^*$ such that $l_1 \sim y_1$ and $l_2 \sim y_2$. Hence by two applications of rule $\langle e_l \rangle$, we obtain $l_1 l_2 \sim y$. As $l_1 l_2 \in L^*$, we conclude $y \Vdash_\sim \mathbf{L}$. We have proved that $\epsilon \Vdash_\sim \mathbf{T} \multimap (\mathbf{L} \ast \mathbf{L} \rightarrow \mathbf{L})$. By an identical argument, we show that $\epsilon \Vdash_\sim \mathbf{T} \multimap (\mathbf{K} \ast \mathbf{K} \rightarrow \mathbf{K})$.

So, we have $\epsilon \Vdash_\sim \mathbf{H}$. To finish, we get $\epsilon \not\Vdash_\sim (\mathbf{I} \wedge \mathbf{H}) \rightarrow G^\circ$ thus (L', \sim, \Vdash_\sim) is the desired BBI-counter-model. \square

8.2. Example of counter-model transformation

In this section, we explain how a BI-counter-model of $X \vee (X \rightarrow \perp)$ is transformed into a BBI-counter-model of $F = (I \wedge H) \rightarrow (K * X \vee K * ((L \wedge K * X) \rightarrow \perp))$ which is the BBI-formula associated to $X \vee (X \rightarrow \perp)$. We recall that $H = (L \wedge K) \wedge ((\top * (L * L \rightarrow L)) \wedge (\top * (K * K \rightarrow K)))$.

One possible BI-counter-model for $X \vee (X \rightarrow \perp)$ is based on the PMO $\sqsubseteq = \sqsubseteq_C$ over the language $L = \{b\}$ with $\mathcal{C} = \{\epsilon \rightarrow \epsilon, b \rightarrow b, \epsilon \rightarrow b\}$. We can check that:

$$\sqsubseteq = \sqsubseteq_C = \{\epsilon \rightarrow \epsilon, b \rightarrow b, \epsilon \rightarrow b\}$$

We complete the BI-frame $(L = \{b\}, \sqsubseteq, \Vdash_{\sqsubseteq})$ with $\epsilon \not\Vdash_{\sqsubseteq} X$ and $b \Vdash_{\sqsubseteq} X$ and we check the monotonicity of the relation \Vdash_{\sqsubseteq} . Then we verify that we have a BI-counter-model of $X \vee (X \rightarrow \perp)$. Indeed, $b \not\Vdash_{\sqsubseteq} \perp$ and thus, as $\epsilon \sqsubseteq b$, we get $\epsilon \not\Vdash_{\sqsubseteq} X \rightarrow \perp$. Hence, $\epsilon \not\Vdash_{\sqsubseteq} X \vee (X \rightarrow \perp)$. This is the usual Kripke counter-model of the intuitionistic propositional formula $X \vee (X \rightarrow \perp)$.

The PMO \sqsubseteq is clearly BI-elementary (case 3 of Definition 5.4 with $m = \epsilon$). According to Lemma 6.13 case 3, we compute $\mathcal{D} = \{\delta \rightarrow b\}$ with $\delta \in K$: $(\sqsubseteq_{\emptyset}, \sim_{\emptyset})$ is a (L, K, M) elementary representation and thus $(\sqsubseteq_C, \sim_{\mathcal{D}})$ is a (L, K, M) -elementary representation.

Let $\sim = \sim_{\mathcal{D}}$. Then we can check the following identity:

$$\sim = \sim_{\mathcal{D}} = \{\epsilon \rightarrow \epsilon, b \rightarrow b, \delta \rightarrow \delta, \delta \rightarrow b, b \rightarrow \delta\}$$

As $A_{\sim} \cap L = \{b\}$, $A_{\sim} \cap K = \{\delta\}$ and $A_{\sim} \cap M = \emptyset$, we can verify $\sqsubseteq = \sqsubseteq_{\sim}^{L, K}$.

As $(\{b\}, \sqsubseteq, \Vdash_{\sqsubseteq})$ is a BI-counter-model for $X \vee (X \rightarrow \perp)$, let us complete $(\{b, \delta\}, \sim, \Vdash_{\sim})$ into a BBI-counter-model for the translation F of $X \vee (X \rightarrow \perp)$. The forcing relation \Vdash_{\sim} is given by Theorem 4.4: $\epsilon \Vdash_{\sim} L, K$, $\epsilon \not\Vdash_{\sim} X$, $b \Vdash_{\sim} X, L, K$ and $\delta \Vdash_{\sim} X, L, K$.

We check that we indeed have a BBI-counter-model of F . We have $\epsilon, b, \delta \Vdash_{\sim} L * L \rightarrow L$ and $\epsilon, b, \delta \Vdash_{\sim} K * K \rightarrow K$. Thus $\epsilon \Vdash_{\sim} \top * (L * L \rightarrow L)$ and $\epsilon \Vdash_{\sim} \top * (K * K \rightarrow K)$. We obtain $\epsilon \Vdash_{\sim} H$ and thus $\epsilon \Vdash_{\sim} I \wedge H$. We also have $b, \delta \Vdash_{\sim} K * X$ and $\epsilon \not\Vdash_{\sim} K * X$. Thus $b, \delta \Vdash_{\sim} L \wedge K * X$ and $\epsilon \not\Vdash_{\sim} L \wedge K * X$. Thus $b, \delta \Vdash_{\sim} (L \wedge K * X) \rightarrow \perp$ and $\epsilon \not\Vdash_{\sim} (L \wedge K * X) \rightarrow \perp$. Hence, $\epsilon \not\Vdash_{\sim} K * ((L \wedge K * X) \rightarrow \perp)$, because $\delta \Vdash_{\sim} K$ and $\delta \epsilon \not\Vdash_{\sim} (L \wedge K * X) \rightarrow \perp$. Thus $\epsilon \not\Vdash_{\sim} K * X \vee K * ((L \wedge K * X) \rightarrow \perp)$ and then $\epsilon \not\Vdash_{\sim} F$ and we indeed have a BBI-counter-model of the formula F which is the translation of $X \vee (X \rightarrow \perp)$.

8.3. Faithfully embedding BI into BBI

We conclude with the central result of this article. If we add two spare logical variables L and K to the language of BI, we can provide a map from BI-formulae (without L and K) to BBI-formulae which preserves both validity and invalidity.

Theorem 8.2. Let L and K be two different spare logical variables, the map $G \mapsto (I \wedge H) \rightarrow G^{\circ}$, where $H \equiv (L \wedge K) \wedge ((\top * (L * L \rightarrow L)) \wedge (\top * (K * K \rightarrow K)))$, is a sound and faithful embedding of BI into BBI.

Proof. Direct combination of Theorems 7.6 and 8.1. □

9. Application to the expressive power of BI

We briefly present one possible application of some of the semantical results of this article related to the expressivity of BI on BI-frames. These results can trivially be transferred to partially ordered partial monoids.

The property of “being squarable” in a BI-frame can be expressed by the first order logic atomic formula $H'(x) = xx \sqsubseteq xx$. We show that it cannot be represented by any BI-formula: there is no formula F of BI such that for any BI-frame $(L, \sqsubseteq, \Vdash_{\sqsubseteq})$ and for any $m \in \mathcal{L}^{\sqsubseteq}$, $H'(m)$ holds if and only if $m \Vdash_{\sqsubseteq} F$.

Proposition 9.1. BI-simple PMOs do not have square words, except of course ϵ .

Proof. Let \sqsubseteq be a BI-simple PMO over L . We want to show that for any $m \in L^*$, if $mm \sqsubseteq mm$ then $m = \epsilon$. According to Theorem 6.14, let L', K and \sim where \sim is a BBI-simple PME over L' such that $L \cup K \subseteq L'$ and $\sqsubseteq = \sqsubseteq_{\sim}^{L, K}$. Let $m \in L^*$ such that $mm \sqsubseteq mm$. Then we have $\delta mm \sim mm$ for some $\delta \in K^*$. Then, $mm \sim mm$ by rule $\langle r \rangle$. By Corollary 6.10 and property 1 of Proposition 6.7, we obtain $m \sim \epsilon$. Hence $\epsilon m \sim \epsilon$ with $\epsilon \in K^*$ and, as $\sqsubseteq = \sqsubseteq_{\sim}^{L, K}$, we obtain $m \sqsubseteq \epsilon$. Thus by Proposition 5.6, we get $m = \epsilon$. \square

Proposition 9.2. Let $H(\sqsubseteq)$ be a property ranging over PMOs verifying:

1. $H(\sqsubseteq)$ is true for every BI-simple PMO \sqsubseteq ;
2. $H(\sqsubseteq_0)$ is false for some (other) PMO \sqsubseteq_0 .

Then, the property $H(\sqsubseteq)$ cannot be represented by a BI-formula, i.e. there is no formula F of BI such that for any BI-frame (L, \sqsubseteq, \Vdash) , $H(\sqsubseteq)$ holds if and only if $\epsilon \Vdash_{\sqsubseteq} F$.

Proof. Let us suppose that such a formula F exists and deduce a contradiction. Then $H(\sqsubseteq)$ holds if and only if $\epsilon \Vdash_{\sqsubseteq} F$. Thus $\epsilon \Vdash_{\sqsubseteq} F$ holds in every BI-simple PMO. Hence, F cannot have a BI-simple counter-model and is thus BI-valid by Theorem 5.9.

On the other hand, let (L, \sqsubseteq, \Vdash) be a (not simple) PMO such that $H(\sqsubseteq)$ does not hold. Then in this frame we have $\epsilon \not\Vdash_{\sqsubseteq} F$. Hence F has a BI-counter-model which contradicts its validity. \square

Corollary 9.3. The property $mm \sqsubseteq mm$ (“being squarable”) is not BI-definable.

Proof. Let $H'(x) = xx \sqsubseteq xx$ which expresses the property being squarable. Let us suppose that the BI-formula F represents H' , i.e. for any $m \in \mathcal{L}^{\sqsubseteq}$, $H'(m)$ holds if and only if $m \Vdash F$. Let us prove that the existence of F leads to a contradiction. Let us consider the property $H(\sqsubseteq) = \forall m \in \mathcal{L}^{\sqsubseteq} (H'(m) \Rightarrow \epsilon \sqsubseteq m)$ which is represented by the formula $F \multimap \text{I}$, i.e. $H(\sqsubseteq)$ holds if and only if $\epsilon \Vdash_{\sqsubseteq} F \multimap \text{I}$.

By Proposition 9.1, $H(\sqsubseteq)$ is true for every BI-simple PMO \sqsubseteq . On the other hand, let $\sqsubseteq_0 = \{\epsilon \multimap \epsilon, a \multimap a, aa \multimap aa\}$ where a is an arbitrary letter. It is obvious to check that \sqsubseteq_0 is a PMO but $H(\sqsubseteq_0)$ does not hold because $a \in \mathcal{L}^{\sqsubseteq_0}$, $aa \sqsubseteq_0 aa$ and $\epsilon \not\sqsubseteq_0 a$. Thus, by Proposition 9.2, $H(\sqsubseteq)$ cannot be represented by $F \multimap \text{I}$. Contradiction. \square

Corollary 9.4. The property $m \sqsubseteq \epsilon$ is not BI-definable.

Proof. Let us suppose $H'(x) = x \sqsubseteq \epsilon$ is expressed by the BI-formula F , i.e. $m \sqsubseteq \epsilon$ holds if and only if $m \Vdash F$. Let us consider $H(\sqsubseteq) = \forall m \in \mathcal{L}^\sqsubseteq (m \sqsubseteq \epsilon \Rightarrow \epsilon \sqsubseteq m)$. $H(\sqsubseteq)$ holds if and only if $\epsilon \Vdash_\sqsubseteq F \multimap \mathbf{I}$. If \sqsubseteq is a BI-simple PMO then $m \sqsubseteq \epsilon$ holds for no word m other than ϵ , see Proposition 5.6. Thus $H(\sqsubseteq)$ holds whenever \sqsubseteq is a BI-simple PMO. On the other hand, let $\sqsubseteq_0 = \{a^i \multimap a^j \mid i \geq j\}$ be the PMO generated by the singleton constraint $\{a \multimap \epsilon\}$, which of course is not BI-elementary. $H(\sqsubseteq_0)$ does not hold because $a \sqsubseteq_0 \epsilon$ but $\epsilon \not\sqsubseteq_0 a$. Contradiction. \square

10. Conclusion and perspectives

In this article, we have proved that there exists a sound and faithful embedding of intuitionistic BI logic into Boolean BI. The result is based on the study of the relations between constraints based models of BI and BBI, namely PMOs and PMEs, the completeness of the class of simple PMOs w.r.t. intuitionistic BI, and the soundness of the TBBI-tableau method for BBI. We also point out some immediate consequences of our intermediary results on the expressivity of BI.

Another quite direct application of our results would be a new proof and counter-model search method for BI, derived from our embedding and based on partial monoidal constraints (PMEs) instead of the existing resource graphs (Galmiche *et al.* 2005). Resource graphs are mainly a graphical representation for PMOs. The embedding we obtain was quite unexpected and is based on the intuition to represent the order relation $m \sqsubseteq n$ by composition with some unobservable word δ such that $\delta m \sim n$. There may be some practical applications of this idea to distinguish observable and unobservable words. In particular we aim at describing a concrete and complete class of separation logic style models for (intuitionistic) BI.

Of course, the proof of the completeness theorem of the TBBI-tableau method (completeness only, not the already achieved soundness) is one of our immediate goals. The complete study and characterization of constraints based models of BBI is a natural evolution of our work. In particular, the TBBI-tableau method may introduce constraints like $\epsilon \multimap ab$ in which case a and b become invertible words. We aim at a generalization of the notion of resource graph to take invertible elements into account. This potentially constitutes a major evolution because invertible elements do not occur in the TBI-tableau method.

The characterization of TBBI-constraints can lead to an effective decision procedure for those partial monoidal constraints. We wish to compute the explicit form of the extension $\sim + \{\epsilon \multimap m\}$, not only when m is a new letter (as in the present article), but when m is any defined word. Combined with the other explicit forms ($\sim + \{ab \multimap m\}$ and $\sim + \{am \multimap b\}$) described in the present article, this can lead to specific properties of TBBI-generated constraints and then to expressivity results for BBI. For example we think that in a TBBI-generated PME \sim , no word m is squarable ($mm \sim mm$) unless it is invertible ($\exists a \, am \sim \epsilon$). As invertibility is BBI-defined by the formula $\neg(\top \multimap \neg \mathbf{I})$, the consequence would be that, as with BI, “being squarable” is not BBI-definable.

11. Acknowledgments

The authors wish to thank the two anonymous referees for useful comments and suggestions about the overall organization of the paper, their detailed review of the proofs and the typos we had missed.

References

- Biri, N. and Galmiche, D. (2007) Models and Separation Logics for Resource Trees, *Journal of Logic and Computation* **17** (4), 687–726. Oxford University Press.
- Brochenin, R., Demri, S. and Lozes, E. (2008) On the Almighty Wand. In: *Proceedings of the 17th EACSL Annual Conference on Computer Science Logic (CSL'08), Lecture Notes in Computer Science* **5213**, 323–338. Springer-Verlag.
- Caires, L. and Lozes, E. (2006) Elimination of Quantifiers and Undecidability in Spatial Logics for Concurrency. *Theoretical Computer Science* **358** (2-3), 293–314. Elsevier.
- Calcagno, C., Cardelli, L. and Gordon, A.D. (2005) Deciding Validity in a Spatial Logic for Trees *Journal of Functional Programming* **15** (4), 543–572. Cambridge University Press.
- Calcagno, C., Gardner, Ph. and Zarfaty, U. (2007) Context Logic as Modal Logic: Completeness and Parametric Inexpressivity. In: *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 123–134. ACM.
- Cardelli, L. and Gordon, A.D. (2000) Anytime, Anywhere: Modal Logics for Mobile Ambients. In: *Proceedings of the 27th ACM SIGPLAN-SIGACT Principles of Programming Languages conference*, 365–377. ACM.
- Dawar, A., Gardner, Ph. and Ghelli, G. (2004) Adjunct Elimination Through Games in Static Ambient Logic. In: *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science* **3328**, 211–223. Springer-Verlag.
- Fitting, M. (1990) First-Order Logic and Automated Theorem Proving. *Texts and Monographs in Computer Science*. Springer-Verlag.
- Galmiche, D. and Larchey-Wendling, D. (2006) Expressivity properties of Boolean BI through Relational Models. In: *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science* **4337**, 358–369. Springer-Verlag.
- Galmiche, D. and Méry, D. (2003) Semantic Labelled Tableaux for Propositional BI. *Journal of Logic and Computation* **13** (5), 707–753. Oxford University Press.
- Galmiche, D., Méry, D. and Pym, D. (2005) The Semantics of BI and Resource Tableaux. *Mathematical Structures in Computer Science* **15** (6), 1033–1088. Cambridge University Press.
- Ishtiaq, S. and O'Hearn, P. (2001) BI as an Assertion Language for Mutable Data Structures. In: *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 14–26. ACM.
- Lozes, E. (2004) Adjuncts elimination in the static ambient logic. *Electronic Notes in Theoretical Computer Science* **96**, 51–72. Elsevier.
- Méry, D. (2004) Preuves et Sémantiques dans des Logiques de Ressources. PhD thesis, Université Henri Poincaré, Nancy I, France. Available at <http://www.loria.fr/~dmery/these.pdf>.
- O'Hearn, P., Reynolds, J. and Yang, H. (2001) Local Reasoning about Programs that Alter Data Structures. In: *Proceedings of the 10th EACSL Annual Conference on Computer Science Logic (CSL'01), Lecture Notes in Computer Science* **2142**, 1–19. Springer-Verlag.

- Otten, J. and Kreitz, C. (1996) T-String Unification: Unifying Prefixes in Non-classical Proof Methods. In: *Proceedings of the 5th International Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Lecture Notes in Computer Science* **1071**, 244–260. Springer-Verlag.
- Pym, D. (1999) On Bunched Predicate Logic. In: *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*, 183–192. IEEE.
- Pym, D. (2002) The Semantics and Proof Theory of the Logic of Bunched Implications. *Applied Logic Series* **26**. Kluwer Academic Publishers. Errata available at <http://www.cs.ac.uk/~pym/pym-tofts-fac-errata.pdf>.
- Pym, D. and Tofts, C. (2006) A Calculus and logic of resources and processes, *Formal Aspects of Computing* **18** (4), 495–517. Springer-Verlag. Errata available at <http://www.cs.ac.uk/~pym/pym-tofts-fac-errata.pdf>.
- Pym, D. and Tofts, C. (2007) Systems modelling via resources and processes: Philosophy, calculus, semantics, and logic. *Electronic Notes in Theoretical Computer Science* **172**, 545–587. Elsevier. Errata available at <http://www.cs.ac.uk/~pym/pym-tofts-fac-errata.pdf>.
- Statman, R. (1979) Intuitionistic Propositional Logic is Polynomial-Space Complete. *Theoretical Computer Science* **9**, 67–72. Elsevier.

Appendix A. Proofs of free PME extensions identities

Proposition 6.3. Let \sim be a PME over L and b be new to \sim , i.e. $b \in L \setminus A_\sim$. Then $\sim + \{\epsilon \multimap b\} = \sim'$ with $\sim' = \{b^p x \multimap b^q y \mid x \sim y \wedge p, q \geq 0\}$ and $A_{\sim'} = A_\sim \cup \{b\}$.

Proof. First it is obvious that $A_{\sim'} = A_\sim \cup \{b\}$ (see Proposition 3.16). Then, let $\sim'' = \sim + \{\epsilon \multimap b\}$. We prove that $\sim' \subseteq \sim''$, and that \sim' is a PME, which is sufficient to establish $\sim'' = \sim'$, because obviously $\sim \cup \{\epsilon \multimap b\} \subseteq \sim'$ (we have $b^0 \epsilon \sim' b^1 \epsilon$).

As $b \sim'' \epsilon$ (by rule $\langle s \rangle$), for any $p \geq 1$, by p applications of rule $\langle e_l \rangle$, we can show that $b^p \sim'' \epsilon$ (since $b \sim'' \epsilon$ and $\epsilon^p \sim'' \epsilon$). Also $b^0 = \epsilon \sim'' \epsilon$. Hence, $b^p \sim'' \epsilon$ and $b^q \sim'' \epsilon$ for any $p, q \geq 0$. Let x, y be such that $x \sim y$. Then $x \sim'' y$ as $\sim \subseteq \sim''$. Thus $\epsilon x \sim'' \epsilon y$ and by applications of rule $\langle e_l \rangle$ and $\langle e_r \rangle$, we obtain $b^p x \sim'' b^q y$. We have proved that $\sim' \subseteq \sim''$.

The relation \sim' is obviously closed under rules $\langle \epsilon, s \rangle$. For rule $\langle t \rangle$, let us consider x, y, z such that $x \sim' y$ and $y \sim' z$. Then $(x, y) = (b^p x', b^q y')$ for some $p, q \geq 0$ and some x', y' such that $x' \sim y'$. Also $(y, z) = (b^r y'', b^s z')$ with $r, s \geq 0$ and $y'' \sim z'$. As $y', y'' \in A_\sim^*$ and $b \notin A_\sim$, from $y = b^q y' = b^r y''$, we deduce $q = r$ and $y' = y''$. Hence $y' \sim z'$ and then $x' \sim z'$ since \sim is closed under rule $\langle t \rangle$. So $x = b^p x' \sim' b^s z' = z$. The relation \sim' is thus closed under rule $\langle t \rangle$.

For rules $\langle d, c \rangle$, the core argument is the same: $b \notin A_\sim$ so that the decomposition $x = b^p x'$ with $x' \in A_\sim^*$ is unique. Hence, having proved that it is closed under rules $\langle d \rangle$ and $\langle c \rangle$, the relation \sim' is a PME. This completes the proof. \square

Lemma 6.4. Let \sim be a PME over L . Let $m \in L^*$ and $\alpha \in L^*$ such that $m \sim m$, $mm \approx mm$, $\alpha \neq \epsilon$ and $A_\alpha \cap A_\sim = \emptyset$. Then $\sim + \{\alpha \multimap m\} = \sim'$ with

$$\begin{aligned} \sim' = \sim \cup & \{ \delta x \multimap \delta y \mid x \sim y \wedge mx \sim my \wedge \delta \prec \alpha \wedge \delta \notin \{\epsilon, \alpha\} \} \\ & \cup \{ \alpha x \multimap \alpha y \mid mx \sim my \} \\ & \cup \{ \alpha x \multimap y \mid mx \sim y \} \\ & \cup \{ x \multimap \alpha y \mid x \sim my \} \end{aligned}$$

and $A_{\sim'} = A_{\sim} \cup A_{\alpha}$.

Proof. First, it is obvious that $A_{\sim'} \subseteq A_{\sim} \cup A_{\alpha}$. Then $A_{\sim} \subseteq A_{\sim'}$ and $\alpha \sim' \alpha$ (because $m\epsilon \sim m\epsilon$), hence $A_{\alpha} \subseteq A_{\sim'}$. Thus, $A_{\sim'} = A_{\sim} \cup A_{\alpha}$. As $A_{\alpha} \cap A_{\sim} = \emptyset$ we have the following property: whenever $x \sim' y$ holds, x and y can be uniquely decomposed into $x = x_1x_2$ and $y = y_1y_2$ such that $x_1, y_1 \in A_{\alpha}^*$ and $x_2, y_2 \in A_{\sim}^*$. Let $\sim'' = \sim + \{\alpha \vdash m\}$. We prove that $\sim' \subseteq \sim''$, and that \sim' is a PME, which is sufficient to establish $\sim'' = \sim'$, because obviously $\sim \cup \{\alpha \vdash m\} \subseteq \sim'$ (we have $\alpha\epsilon \sim' m$).

For $\sim' \subseteq \sim''$, we already have $\sim \subseteq \sim''$ and we provide the following deduction trees:

$$\begin{array}{c}
\frac{mx \sim my}{\frac{my \sim my}{\alpha \sim'' m} \langle r \rangle} \sim \subseteq \sim'' \\
\frac{\frac{\frac{mx \sim my}{my \sim my} \langle r \rangle}{\alpha \sim'' m} \langle e_l \rangle}{\frac{\frac{\frac{mx \sim my}{my \sim my} \langle r \rangle}{\alpha \sim'' m} \langle e_l \rangle}{\frac{\frac{mx \sim my}{my \sim my} \langle r \rangle}{\alpha \sim'' m} \langle e_l \rangle} \langle p_l \rangle, \delta \prec \alpha} \frac{x \sim y}{x \sim'' y} \sim \subseteq \sim'' \\
\frac{\frac{\frac{mx \sim my}{my \sim my} \langle r \rangle}{\alpha \sim'' m} \langle e_l \rangle}{\frac{\frac{mx \sim my}{my \sim my} \langle r \rangle}{\alpha \sim'' m} \langle e_l \rangle} \langle c \rangle \\
\delta x \sim'' \delta y
\end{array}
\quad
\begin{array}{c}
\frac{mx \sim y}{\alpha \sim'' m} \sim \subseteq \sim'' \\
\frac{\frac{mx \sim y}{\alpha \sim'' m} \langle e_l \rangle}{\alpha x \sim'' y} \langle e_l \rangle
\end{array}$$

$$\begin{array}{c}
\frac{mx \sim my}{\alpha \sim'' m} \sim \subseteq \sim'' \\
\frac{\frac{mx \sim my}{\alpha \sim'' m} \langle e_r \rangle}{\alpha x \sim'' \alpha y} \langle e_l \rangle
\end{array}
\quad
\begin{array}{c}
\frac{x \sim my}{\alpha \sim'' m} \sim \subseteq \sim'' \\
\frac{\frac{x \sim my}{\alpha \sim'' m} \langle e_r \rangle}{x \sim'' \alpha y} \langle e_r \rangle
\end{array}$$

Now let us prove that \sim' is a PME. \sim' is obviously closed under rules $\langle \epsilon \rangle$ and $\langle s \rangle$ (observe the symmetry between x and y in the definition of \sim'). Then, let us consider the case of rule $\langle d \rangle$. Let x, y such that $xy \sim' xy$. If $xy \sim xy$ then $x \sim x$ hence $x \sim' x$. Let us consider the other cases:

- as $\alpha \neq \epsilon$ and $A_{\alpha} \cap A_{\sim} = \emptyset$, it is no possible that $(xy, xy) = (\alpha x', y')$ or $(xy, xy) = (x', \alpha y')$;
- hence $xy = \delta z'$ with $mz' \sim mz'$ and $\epsilon \neq \delta \prec \alpha$ (this covers the cases $\delta \neq \alpha$ and $\delta = \alpha$). As $A_{\alpha} \cap A_{\sim} = \emptyset$, let $x = x_1x_2$ and $y = y_1y_2$ with $x_1, y_1 \in A_{\alpha}^*$ and $x_2, y_2 \in A_{\sim}^*$. From $xy = x_1x_2y_1y_2 = \delta z'$, we obtain $\delta = x_1y_1$ and $z' = x_2y_2$. Then $mx_2y_2 \sim mx_2y_2$. Thus $mx_2 \sim mx_2$ and $x_2 \sim x_2$ by rule $\langle d \rangle$. If $x_1 = \epsilon$ then $x = x_2 \sim x_2 = x$ hence $x \sim' x$. If $x_1 \neq \epsilon$ then $\epsilon \neq x_1 \prec x_1y_1 = \delta \prec \alpha$ hence, as $mx_2 \sim mx_2$ and $x_2 \sim x_2$, we obtain $x = x_1x_2 \sim' x_1x_2 = x$.

Now let us consider rule $\langle t \rangle$. Let x, y, z such that $x \sim' y$ and $y \sim' z$. We want to prove $x \sim' z$. In theory, there are $5 \times 5 = 25$ cases to study for (x, y) and (y, z) . But as we have already proved that \sim' is symmetric, we only need to consider “half” of the matrix of cases, i.e. $5 + \dots + 1 = 15$ cases:

- $x \sim y$ and $y \sim z$. Then $x \sim z$ hence $x \sim' z$;
- $x \sim y$ and $(y, z) = (\delta y', \delta z')$ with $\delta \prec \alpha$ and $\delta \neq \epsilon$ is impossible because $\delta \in A_{\alpha}^*$, $\delta \prec y \in A_{\sim}^*$ and $A_{\alpha} \cap A_{\sim} = \emptyset$;

- $x \sim y$ and $(y, z) = (\alpha y', \alpha z')$ is impossible by the same argument, $\alpha \neq \epsilon$;
- $x \sim y$ and $(y, z) = (\alpha y', z)$ is impossible because $\alpha \neq \epsilon$;
- $x \sim y$ and $(y, z) = (y, \alpha z')$ with $y \sim mz'$. So $x \sim mz'$ and thus $x \sim' \alpha z' = z$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (\delta' y'', \delta' z')$ with $x' \sim y'$, $mx' \sim my'$, $y'' \sim z'$, $my'' \sim mz'$, $\delta \prec \alpha$, $\delta' \prec \alpha$ and $\delta, \delta' \notin \{\epsilon, \alpha\}$. From $y = \delta y' = \delta' y''$, we deduce $\delta = \delta'$ and $y' = y''$. Then $y' \sim z'$ and $my' \sim mz'$, hence by rule $\langle t \rangle$, $x' \sim z'$ and $mx' \sim mz'$. So, $\delta x' \sim' \delta z'$ hence $x \sim' z$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (\alpha y'', \alpha z')$ is impossible because $\delta \neq \alpha$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (\alpha y'', z)$ is impossible because $\delta \neq \alpha$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (y, \alpha z')$ is impossible because $\delta \neq \epsilon$;
- $(x, y) = (\alpha x', \alpha y')$ and $(y, z) = (\alpha y'', \alpha z')$ with $mx' \sim my'$ and $my'' \sim mz'$. Then $y = \alpha y' = \alpha y''$ hence $y' = y''$ and $my' \sim mz'$. Thus $mx' \sim mz'$ and we obtain $\alpha x' \sim' \alpha z'$, hence $x \sim' z$;
- $(x, y) = (\alpha x', \alpha y')$ and $(y, z) = (\alpha y'', z)$ with $mx' \sim my'$ and $my'' \sim z$. Then $y' = y''$ and thus $mx' \sim z$ hence $x = \alpha x' \sim' z$;
- $(x, y) = (\alpha x', \alpha y')$ and $(y, z) = (y, \alpha z')$ is impossible because $\alpha \neq \epsilon$;
- $(x, y) = (\alpha x', y)$ and $(y, z) = (\alpha y', z)$ is impossible because $\alpha \neq \epsilon$;
- $(x, y) = (\alpha x', y)$ and $(y, z) = (y, \alpha z')$ with $mx' \sim y$ and $y \sim mz'$. Then $mx' \sim mz'$ hence $\alpha x' \sim' \alpha z'$ thus $x \sim' z$;
- $(x, y) = (x, \alpha y')$ and $(y, z) = (y, \alpha z')$ is impossible because $\alpha \neq \epsilon$.

Let us consider rule $\langle c \rangle$. Let us consider q, x, y such that $qy \sim' qy$ and $x \sim' y$. We want to prove $qx \sim' qy$. We consider $2 \times 5 = 10$ cases:

- $qy \sim qy$ and $x \sim y$. Then $qx \sim qy$ thus $qx \sim' qy$;
- $qy \sim qy$ and $(x, y) = (\delta x', \delta y')$ is impossible because we would have $\delta \prec qy$ with $\epsilon \neq \delta \in A_\alpha^*$, $qy \in A_\sim^*$ and $A_\alpha \cap A_\sim = \emptyset$;
- $qy \sim qy$ and $(x, y) = (\alpha x', \alpha y')$ is impossible because $\alpha \not\prec qy$;
- $qy \sim qy$ and $(x, y) = (\alpha x', y)$ with $mx' \sim y$. Then $qmx' \sim qy$ by rule $\langle c \rangle$. Hence $qx = \alpha qx' \sim' qy$;
- $qy \sim qy$ and $(x, y) = (x, \alpha y')$ is impossible because $\alpha \not\prec qy$;
- $qy = \delta z'$ and $x \sim y$ with $mz' \sim mz'$, $\epsilon \neq \delta \prec \alpha$. Let $q = q_1 q_2$ with $q_1 \in A_\alpha^*$ and $q_2 \in A_\sim^*$. As $y, z' \in A_\sim^*$, from $qy = q_1 q_2 y = \delta z'$ we get $q_1 = \delta$ and $q_2 y = z'$. Hence $mq_2 y \sim mq_2 y$. Thus $q_2 y \sim q_2 y$ and we obtain $mq_2 x \sim mq_2 y$ and $q_2 x \sim q_2 y$ by rule $\langle c \rangle$. In either case ($\delta = \alpha$ or $\delta \neq \alpha$) we deduce $qx = \delta q_2 x \sim' \delta q_2 y = qy$;
- $qy = \delta z'$ and $(x, y) = (\delta' x', \delta' y')$ with $mz' \sim mz'$, $\epsilon \neq \delta \prec \alpha$, $x' \sim y'$, $mx' \sim my'$, $\delta' \prec \alpha$ and $\delta' \notin \{\epsilon, \alpha\}$. Let $q = q_1 q_2$ with $q_1 \in A_\alpha^*$ and $q_2 \in A_\sim^*$. From $qy = q_1 q_2 \delta' y' = \delta z'$ we obtain $q_1 \delta' = \delta$ and $q_2 y' = z'$. As in the previous case, we derive $mq_2 x' \sim mq_2 y'$ and $q_2 x' \sim q_2 y'$. And thus $qx = q_1 q_2 \delta' x' = \delta q_2 x' \sim' \delta q_2 y' = q_1 q_2 \delta' y' = qy$;
- $qy = \delta z'$ and $(x, y) = (\alpha x', \alpha y')$ with $mz' \sim mz'$, $\epsilon \neq \delta \prec \alpha$ and $mx' \sim my'$. Let $q = q_1 q_2$ with $q_1 \in A_\alpha^*$ and $q_2 \in A_\sim^*$. From $qy = q_1 q_2 \alpha y' = \delta z'$ we obtain $q_1 \alpha = \delta$ and $q_2 y' = z'$. Hence $q_1 = \epsilon$ and $\alpha = \delta$. Thus $q = q_2$ and from $mz' \sim mz'$ we get $mqy' \sim mqy'$. Combining with $mx' \sim my'$ by rule $\langle c \rangle$, we obtain $mqx' \sim mqy'$. Thus $qx = \alpha qx' \sim' \alpha qy' = qy$;

- $qy = \delta z'$ and $(x, y) = (\alpha x', y)$ with $mz' \sim mz'$, $\epsilon \neq \delta \prec \alpha$ and $mx' \sim y$. As $y \in A_\sim^*$, from $qy = \delta z'$ we deduce $\delta \prec q$. Let $\delta q' = q$. Then $q'y = z'$. Hence $mq'y \sim mq'y$. Combining with $mx' \sim y$ by rule $\langle c \rangle$, we derive $mq'mx' \sim mq'y$ hence $mm \sim mm$ by rule $\langle p_l \rangle$. This contradicts the overall hypothesis $mm \approx mm$;
- $qy = \delta z'$ and $(x, y) = (x, \alpha y')$ with $mz' \sim mz'$, $\epsilon \neq \delta \prec \alpha$ and $x \sim my'$. Let $q = q_1 q_2$ with $q_1 \in A_\alpha^*$ and $q_2 \in A_\sim^*$. From $qy = q_1 q_2 \alpha y' = \delta z'$ we obtain $q_1 \alpha = \delta$ and $q_2 y' = z'$. Hence $q_1 = \epsilon$ and $\alpha = \delta$. Thus $q = q_2$ and from $mz' \sim mz'$ we get $mqy' \sim mqy'$. Combining with $x \sim my'$ by rule $\langle c \rangle$, we obtain $qx \sim qmy'$. Hence $qx \sim' \alpha qy' = qy$.

So we have proved that the relation \sim' is closed under all the rules defining PME. \square

Lemma 6.5. Let \sim be a PME over L . Let $m \in L^*$, $\alpha \in L^*$, $b \in L$ such that $m \sim m$, $\alpha \neq \epsilon$, $A_\alpha \cap A_\sim = \emptyset$ and $b \notin A_\sim \cup A_\alpha$. Then $\sim + \{\alpha m \rightarrow b\} = \sim'$ with

$$\begin{aligned} \sim' = \sim \cup \{ & \delta x \rightarrow \delta y \mid x \sim y \wedge \epsilon \neq \delta \prec \alpha \wedge \exists k \, xk \sim m \} \\ & \cup \{ \alpha x \rightarrow jb \mid x \sim jm \wedge \exists k \, jkm \sim m \} \\ & \cup \{ ib \rightarrow \alpha y \mid y \sim im \wedge \exists k \, ikm \sim m \} \\ & \cup \{ ib \rightarrow jb \mid \exists k \, (ikm \sim m \wedge jkm \sim m) \} \end{aligned}$$

and $A_{\sim'} = A_\sim \cup A_\alpha \cup \{b\}$.

Proof. First it is obvious that $A_{\sim'} \subseteq A_\sim \cup A_\alpha \cup \{b\}$. Then $A_\sim \subseteq A_{\sim'}$. As $m \sim m$, we obtain $b \sim' b$ and $\alpha m \sim' b$, hence $A_\alpha \subseteq A_{\sim'}$ and $b \in A_{\sim'}$. Thus we get $A_{\sim'} = A_\sim \cup A_\alpha \cup \{b\}$. Let $\sim'' = \sim + \{\alpha m \rightarrow b\}$. We prove that $\sim' \subseteq \sim''$, and that \sim' is a PME, which is sufficient to establish $\sim'' = \sim'$ because obviously $\sim \cup \{\alpha m \rightarrow b\} \subseteq \sim'$ (we have $\alpha m \sim' \epsilon b$ because $m \sim \epsilon m$ and $\epsilon \epsilon m \sim m$).

For $\sim' \subseteq \sim''$, we already have $\sim \subseteq \sim''$. Let x, y, δ, k be such that $x \sim y$, $\epsilon \neq \delta \prec \alpha$ and $xk \sim m$. We provide the following deduction tree, split in two parts:

$$\begin{array}{c} \frac{xk \sim m}{xk \sim'' m} \sim \subseteq \sim'' \quad \frac{\alpha m \sim'' b}{\alpha m \sim'' \alpha m} \langle l \rangle \quad \frac{x \sim y}{y \sim x} \langle s \rangle \quad \frac{\dots}{\alpha kx \sim'' \alpha m} \langle e_l \rangle \\ \hline \alpha kx \sim'' \alpha m \quad \frac{y \sim'' x}{\delta x \sim'' \delta x} \sim \subseteq \sim'' \quad \frac{\dots}{\delta x \sim'' \delta x} \langle p_l \rangle, \delta \prec \alpha \\ \hline \dots \quad \delta x \sim'' \delta y \quad \langle e_r \rangle \end{array}$$

hence we obtain $\delta x \sim'' \delta y$. Now let x, i, k be such that $x \sim im$ and $ikm \sim m$. We provide the two following deduction trees:

$$\begin{array}{c} \frac{ikm \sim m}{ikm \sim'' m} \sim \subseteq \sim'' \quad \frac{\alpha m \sim'' b}{\alpha m \sim'' \alpha m} \langle e_l \rangle \\ \hline \alpha ikm \sim'' b \quad \frac{\alpha m \sim'' b}{b \sim'' \alpha m} \langle s \rangle \quad \frac{x \sim im}{x \sim'' im} \sim \subseteq \sim'' \quad \frac{\dots}{ib \sim'' i\alpha m} \langle c \rangle \\ \hline i\alpha m \sim'' i\alpha m \quad \frac{ib \sim'' i\alpha m}{ib \sim'' \alpha x} \langle p_l \rangle \quad \frac{\dots}{ib \sim'' \alpha x} \langle e_r \rangle \\ \hline ib \sim'' i\alpha m \quad \dots \end{array}$$

hence we obtain $ib \sim'' \alpha x$. By rule $\langle s \rangle$, we also obtain $\alpha x \sim'' ib$. Let us consider the last

line of the definition of \sim' : let i, j, k be such that $ikm \sim m$ and $jk m \sim m$. Then we can deduce $im \sim jm$:

$$\frac{\frac{ikm \sim m}{\langle p_l \rangle} \quad \frac{jk m \sim m}{\langle p_l \rangle}}{\frac{im \sim im}{\langle e_r \rangle} \quad \frac{jm \sim jm}{\langle e_r \rangle}} \quad \frac{ikm \sim m}{\langle c \rangle}}{\frac{im \sim ijk m}{\langle t \rangle}} \quad \frac{jm \sim jm}{\langle t \rangle}}{\frac{im \sim jm}{\langle t \rangle}}$$

But we have already proved that we can deduce $ib \sim'' i\alpha m$ from $ikm \sim m$ (see first part, on the left, of a previous deduction tree). Replacing i by j we can deduce $jb \sim'' j\alpha m$ from $jk m \sim m$. Thus we provide the deduction tree:

$$\frac{\frac{\dots}{im \sim jm} \quad \frac{im \sim'' jm}{\sim \subseteq \sim''} \quad \frac{\dots}{jb \sim'' \alpha jm}}{\frac{im \sim'' jm}{\langle e_r \rangle} \quad \frac{jb \sim'' \alpha jm}{\langle s \rangle}} \quad \frac{\dots}{ib \sim'' \alpha im} \quad \frac{\alpha im \sim'' jb}{\langle t \rangle}}{\frac{ib \sim'' jb}{\langle t \rangle}}$$

hence $ib \sim'' ib$. So for any x, y such that $x \sim' y$, we have proved that $x \sim'' y$. Thus $\sim' \subseteq \sim''$.

Now let us prove that \sim' is a PME. We first make the following remark: if $x \sim' x$ then either $x \sim x$ or $(x = \delta x'$ with $x' \sim x'$, $\epsilon \neq \delta \prec \alpha$ and $x'k \sim m$ for some k) or $(x = ib$ with $ikm \sim m$ for some k). Now let us consider the five rules defining PMEs one by one. $\epsilon \sim' \epsilon$ since $\epsilon \sim \epsilon$. Hence \sim' is closed under rule $\langle \epsilon \rangle$. Let us consider rule $\langle s \rangle$. Let x, y such that $x \sim' y$. We have five cases:

- $x \sim y$ hence $y \sim x$ and then $y \sim' x$;
- $(x, y) = (\delta x', \delta y')$ with $x' \sim y'$, $\epsilon \neq \delta \prec \alpha$ and $x'k \sim m$. Then $y' \sim x'$ and thus $y'k \sim m$ by rule $\langle e_l \rangle$. Hence $y = \delta y' \sim' \delta x' = x$;
- $(x, y) = (\alpha x', jb)$ with $x' \sim jm$ and $jk m \sim m$. Then $y = jb \sim' \alpha x' = x$;
- $(x, y) = (ib, \alpha y')$ same argument;
- $(x, y) = (ib, jb)$ with $ikm \sim m$ and $jk m \sim m$. Then $y = jb \sim' ib = x$.

In any case we obtain $y \sim' x$. So \sim' is closed under rule $\langle s \rangle$.

Let us consider rule $\langle d \rangle$. Let x, y such that $xy \sim' xy$. We have three cases, because $\alpha z' \neq ib$ ($b \notin A_\alpha \cup A_\sim$):

- $xy \sim xy$ hence $x \sim x$ and then $x \sim' x$;
- $xy = \delta z'$ with $z' \sim z'$, $\epsilon \neq \delta \prec \alpha$ and $z'k \sim m$ for some k . Then $x, y \in (A_\alpha \cup A_\sim)^*$ and let $x = x_1 x_2$, $y = y_1 y_2$ with $x_1, y_1 \in A_\alpha^*$ and $x_2, y_2 \in A_\sim^*$. We thus have $\delta = x_1 y_1$ and $z' = x_2 y_2$. From $x_2 y_2 = z' \sim z' = x_2 y_2$, we deduce $x_2 \sim x_2$ by rule $\langle d \rangle$. On the one hand, if $x_1 = \epsilon$ then $x = x_2 \sim x_2 = x$, hence $x \sim' x$. On the other hand, if $x_1 \neq \epsilon$,

as we have $x_2(y_2k) = z'k \sim m$ and $x_1 \prec x_1y_1 = \delta \prec \alpha$, we obtain $x_1x_2 \sim' x_1x_2$ hence $x \sim' x$;

- $xy = ib$ with $ikm \sim m$. Then either $b \prec x$ or $b \not\prec x$. On the one hand, if $b \not\prec x$ then, as b is a one letter word, $b \prec y$ hence $x(y/b)b = ib$, so $x(y/b) = i$ and thus $x(y/b)km \sim m$. By application of rule $\langle p_l \rangle$, we have $x \sim x$ thus $x \sim' x$. On the other hand, if $b \prec x$ then $(x/b)y = i$ hence $(x/b)(yk)m \sim m$. Thus, $x = (x/b)b \sim' (x/b)b = x$.

In any case we obtain $x \sim' x$. So \sim' is closed under rule $\langle d \rangle$.

Let us consider rule $\langle t \rangle$. Let x, y, z such that $x \sim' y$ and $y \sim' z$. As we have already proved that \sim' is closed under rule $\langle s \rangle$, we only need to consider $5 + \dots + 1 = 15$ cases:

- $x \sim y$ and $y \sim z$. Then $x \sim z$ hence $x \sim' z$;
- $x \sim y$ and $(y, z) = (\delta y', \delta z')$ would imply $\delta \prec \delta y' = y$ which is impossible because $\epsilon \neq \delta \in A_\alpha^*$ and $A_\sim \cap A_\alpha = \emptyset$;
- $x \sim y$ and $(y, z) = (\alpha y', j b)$ is impossible because $\alpha \neq \epsilon$ and $A_\sim \cap A_\alpha = \emptyset$;
- $x \sim y$ and $(y, z) = (ib, \alpha z')$ is impossible because $b \notin A_\sim$;
- $x \sim y$ and $(y, z) = (ib, j b)$ is impossible because $b \notin A_\sim$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (\delta' y'', \delta' z')$ with $x' \sim y', y'' \sim z', \epsilon \neq \delta \prec \alpha, \epsilon \neq \delta' \prec \alpha$, $x'k \sim m$ and $y''k' \sim m$. Then, as $y = \delta y' = \delta' y''$ and $A_\sim \cap A_\alpha = \emptyset$, we have $\delta = \delta'$ and $y' = y''$. As $x' \sim y'$ and $y' = y'' \sim z'$, we obtain $x' \sim z'$. Hence, as $x'k \sim m$, we obtain $x = \delta x' \sim' \delta z' = z$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (\alpha y'', j b)$ with $x' \sim y', \epsilon \neq \delta \prec \alpha, x'k \sim m, y'' \sim j m$ and $j k' m \sim m$. Then $y = \delta y' = \alpha y''$, which implies $\delta = \alpha$ and $y' = y''$ ($A_\sim \cap A_\alpha = \emptyset$). Thus $y' \sim j m$ and hence $x' \sim j m$ by rule $\langle t \rangle$. So $j m \sim x'$ by rule $\langle s \rangle$. As $x'k \sim m$, we obtain $j m k \sim m$ by rule $\langle e_l \rangle$. Hence we have $x' \sim j m$ and $j k m \sim m$. Thus $x = \delta x' = \alpha x' \sim' j b = z$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (ib, \alpha z')$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $(x, y) = (\delta x', \delta y')$ and $(y, z) = (ib, j b)$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $(x, y) = (\alpha x', j b)$ and $(y, z) = (\alpha y', j' b)$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $(x, y) = (\alpha x', j b)$ and $(y, z) = (i' b, \alpha z')$ with $x' \sim j m, j k m \sim m, z' \sim i' m$ and $i' k' m \sim m$. As $y = j b = i' b$ we obtain $j = i'$. As $x' \sim j m$ and $j k m \sim m$, we obtain $x'k \sim m$ by rule $\langle e_l \rangle$. As $z' \sim i' m$, we obtain $j m = i' m \sim z'$ by rule $\langle s \rangle$ and thus $x' \sim z'$ by rule $\langle t \rangle$. Hence, as $\epsilon \neq \alpha \prec \alpha$, we get $x = \alpha x' \sim' \alpha z' = z$;
- $(x, y) = (\alpha x', j b)$ and $(y, z) = (i' b, j' b)$ with $x' \sim j m, j k m \sim m, i' k' m \sim m$ and $j' k' m \sim m$. As remarked previously, we necessarily have $i' m \sim j' m$. Thus $j' m \sim i' m$ by rule $\langle s \rangle$. As $y = j b = i' b$, we get $j = i'$ and thus $i' k m \sim m$. As $j' m \sim i' m$ and $(i' m)k \sim m$, we get $(j' m)k \sim m$ by rule $\langle e_l \rangle$. Hence $j' k m \sim m$. As $x' \sim j m = i' m$ and $i' m \sim j' m$, we get $x' \sim j' m$ by rule $\langle t \rangle$. Thus $x = \alpha x' \sim' j' b = z$;
- $(x, y) = (ib, \alpha y')$ and $(y, z) = (i' b, \alpha z')$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $(x, y) = (ib, \alpha y')$ and $(y, z) = (i' b, j' b)$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $(x, y) = (ib, j b)$ and $(y, z) = (i' b, j' b)$ with $ikm \sim m, j k m \sim m, i' k' m \sim m$ and $j' k' m \sim m$. Then $y = j b = i' b$ hence $j = i'$ and as remarked previously, we necessarily have $i m \sim j m$ and $i' m \sim j' m$. As $j m = i' m$ we get $i m \sim j' m$ by rule $\langle t \rangle$. As

$(j'm)k' \sim m$, we get $(im)k' \sim m$ by rule $\langle e_l \rangle$. Hence $ik'm \sim m$ and $j'k'm \sim m$. Thus $x = ib \sim' j'b = z$;

In any case we obtain $x \sim' z$, so we have proved that \sim' is closed under rule $\langle t \rangle$.

Let us consider rule $\langle c \rangle$. Let q, x, y be such that $qy \sim' qy$ and $x \sim' y$. We consider $3 \times 5 = 15$ cases:

- $qy \sim qy$ and $x \sim y$. Then $qx \sim qy$ and thus $qx \sim' qy$;
- $qy \sim qy$ and $(x, y) = (\delta x', \delta y')$ would imply $\delta \prec qy$ which is impossible because $\epsilon \neq \delta \in A_\alpha^*$, $qy \in A_\sim^*$ and $A_\sim \cap A_\alpha = \emptyset$;
- $qy \sim qy$ and $(x, y) = (\alpha x', j'b)$ is impossible because $b \notin A_\sim$;
- $qy \sim qy$ and $(x, y) = (ib, \alpha y')$ is impossible because $\alpha \neq \epsilon$ and $A_\sim \cap A_\alpha = \emptyset$;
- $qy \sim qy$ and $(x, y) = (ib, j'b)$ is impossible because $b \notin A_\sim$;
- $qy = \delta z'$ and $x \sim y$ with $z' \sim z'$, $\epsilon \neq \delta \prec \alpha$ and $z'k \sim m$. As $\delta \prec \delta z' = qy$, $y \in A_\sim^*$, $\delta \prec \alpha$ and $A_\sim \cap A_\alpha = \emptyset$, we obtain $\delta \prec q$. So let $q = \delta q'$ hence $q'y = z'$. Then $q'y \sim q'y$, hence $q'x \sim q'y$ by rule $\langle c \rangle$. As $q'yk = z'k \sim m$ we obtain $(q'x)k \sim m$ by rule $\langle e_l \rangle$. Hence $qx = \delta q'x \sim' \delta q'y = qy$;
- $qy = \delta z'$ and $(x, y) = (\delta'x', \delta'y')$ with $z' \sim z'$, $\epsilon \neq \delta \prec \alpha$, $z'k \sim m$, $x' \sim y'$, $\epsilon \neq \delta' \prec \alpha$ and $x'k' \sim m$. As $q \prec qy = \delta z'$, we have $q \in (A_\alpha \cup A_\sim)^*$. So let $q = q_1 q_2$ with $q_1 \in A_\alpha^*$ and $q_2 \in A_\sim^*$. From $qy = q_1 q_2 \delta'y' = \delta z'$, we obtain $q_1 \delta' = \delta$ and $q_2 y' = z'$. As $z' \sim z'$, we deduce $q_2 y' \sim q_2 y'$. As $x' \sim y'$, we deduce $q_2 x' \sim q_2 y'$ by rule $\langle c \rangle$. Also $z'k \sim m$ i.e. $q_2 y'k \sim m$. Thus $q_2 x'k \sim m$ by rule $\langle e_l \rangle$. Then, we obtain $qx = q_1 q_2 \delta'x' = \delta q_2 x' \sim' \delta q_2 y' = q_1 \delta' q_2 y' = qy$;
- $qy = \delta z'$ and $(x, y) = (\alpha x', j'b)$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $qy = \delta z'$ and $(x, y) = (ib, \alpha y')$ with $z' \sim z'$, $\epsilon \neq \delta \prec \alpha$, $z'k \sim m$, $y' \sim im$ and $ik'm \sim m$. Then $qy = q\alpha y' = \delta z'$, hence $\alpha \prec \delta z'$. As $A_\sim \cap A_\alpha = \emptyset$ and $z' \in A_\sim^*$, we obtain $\alpha \prec \delta$. Hence $\alpha = \delta$ and thus $qy' = z'$. We deduce $qy'k \sim m$, hence $qy' \sim qy'$ by rule $\langle p_l \rangle$. From $y' \sim im$, we derive $im \sim y'$ by rule $\langle s \rangle$ and then $qy' \sim q(im)$ by rule $\langle e_r \rangle$, hence $qim \sim qy'$ by rule $\langle s \rangle$. From $qy'k \sim m$, we obtain $qimk \sim m$ by rule $\langle e_l \rangle$. Hence we have $(qy') \sim (qi)m$ and $(qi)km \sim m$. Thus $qx = (qi)b \sim' \alpha(qy') = qy$;
- $qy = \delta z'$ and $(x, y) = (i'b, j'b)$ is impossible because $b \notin A_\sim \cup A_\alpha$;
- $qy = ib$ and $x \sim y$ with $ikm \sim m$. We have $b \not\prec y$ because $y \in A_\sim^*$. As b is a single letter word, we deduce $b \prec q$ hence $(q/b)y = i$. Thus $((q/b)y)km \sim m$ and so, as $x \sim y$, we get $((q/b)x)km \sim m$ by rule $\langle e_l \rangle$. Thus we have $qx = (q/b)xb \sim' (q/b)yb = qy$;
- $qy = ib$ and $(x, y) = (\delta x', \delta y')$ would imply $\delta \prec ib$ which is impossible because $\delta \neq \epsilon$ and $A_\alpha \cap (A_\sim \cup \{b\}) = \emptyset$;
- $qy = ib$ and $(x, y) = (\alpha x', j'b)$ with $ikm \sim m$, $x' \sim j'm$ and $j'k'm \sim m$. As $qy = qj'b = ib$, we obtain $qj' = i$. Hence $qj'km \sim m$. So $qj'm \sim qj'm$ by rule $\langle p_l \rangle$. As $x' \sim j'm$, we obtain $qx' \sim (qj')m$ by rule $\langle e_l \rangle$. Since we also have $(qj')km \sim m$, we conclude $qx = \alpha(qx') \sim' (qj')b = qy$;
- $qy = ib$ and $(x, y) = (ib, \alpha y')$ would imply $\alpha \prec ib$ which is impossible because $\alpha \neq \epsilon$ and $A_\alpha \cap (A_\sim \cup \{b\}) = \emptyset$;
- $qy = ib$ and $(x, y) = (i'b, j'b)$ with $ikm \sim m$, $i'k'm \sim m$ and $j'k'm \sim m$. As $qy = qj'b = ib$, we get $qj' = i$. Hence $qj'km \sim m$. From $i'k'm \sim m$ and $j'k'm \sim m$,

we deduce $i'm \sim j'm$ as remarked previously. From $qk(j'm) \sim m$, we thus derive $qk(i'm) \sim m$ by rule $\langle e_l \rangle$. So $(qi')km \sim m$ and $(qj')km \sim m$, and thus $qx = (qi')b \sim' (qj')b = qy$.

In any case we obtain $qx \sim' qy$, so we have proved that \sim' is closed under rule $\langle c \rangle$. \square

Appendix B. Complete proof of the fundamental lemma

Lemma 6.13. Let L , K and M be three mutually disjoint alphabets. Let (\sqsubseteq, \sim) be a (L, K, M) elementary representation. Let m, a, b, δ and c such that $m \sqsubseteq m$, $a \neq b \in L \setminus A_\sqsubseteq$, $\delta \in K \setminus A_\sim$ and $c \in M \setminus A_\sim$. Then in each of the following cases, (\sqsubseteq', \sim') is a (L, K, M) elementary representation:

1. $\sqsubseteq' = \sqsubseteq + \{ab \rightarrow m\}$ and $\sim' = \sim + \{\delta c \rightarrow m, ab \rightarrow c\}$ when $m \neq \epsilon$
2. $\sqsubseteq' = \sqsubseteq + \{am \rightarrow b\}$ and $\sim' = \sim + \{cm \rightarrow b, \delta a \rightarrow c\}$
3. $\sqsubseteq' = \sqsubseteq + \{m \rightarrow b\}$ and $\sim' = \sim + \{\delta m \rightarrow b\}$
- 3'. $\sqsubseteq' = \sqsubseteq + \{m \rightarrow b\}$ and $\sim' = \sim + \{\delta m \rightarrow b, \epsilon \rightarrow c\}$
4. $\sqsubseteq' = \sqsubseteq + \{\epsilon \rightarrow m\}$ and $\sim' = \sim + \{\delta c \rightarrow m, \epsilon \rightarrow c\}$ when $m \neq \epsilon$
- 4'. $\sqsubseteq' = \sqsubseteq + \{\epsilon \rightarrow m\}$ and $\sim' = \sim + \{\epsilon \rightarrow c, m \rightarrow \delta\}$ when $m \neq \epsilon$

Proof. Case 2 was treated as an illustration in the main body of this article. We consider the remaining cases. Let us consider case 1 where $\sqsubseteq' = \sqsubseteq + \{ab \rightarrow m\}$ and $\sim' = \sim + \{\delta c \rightarrow m, ab \rightarrow c\}$ and $m \neq \epsilon$. As $m \neq \epsilon$, \sqsubseteq' is clearly BI-elementary. By Proposition 6.12, we have $m \sqsubseteq m \Rightarrow m \in \mathcal{L}^\sqsubseteq \Rightarrow m \in \mathcal{L}^\sim \Rightarrow m \sim m$. As \sim has no square (see Corollary 6.10), by Proposition 6.7 we have $m \sim \epsilon \Rightarrow m \in I_\sim^* \Rightarrow m \in L^* \cap M^* \Rightarrow m = \epsilon$. Hence $m \not\sim \epsilon$ and thus $\delta c \rightarrow m$ is BBI-elementary w.r.t. \sim ($\delta \neq c$ and $\delta, c \notin A_\sim$). Thus $\sim'' = \sim + \{\delta c \rightarrow m\}$ is BBI-elementary and according to Proposition 6.9, $I_{\sim''} = I_\sim$. Then we have $c \sim'' \epsilon \Rightarrow c \in I_{\sim''} \Rightarrow c \in I_\sim \Rightarrow c \in A_\sim$. Thus from $c \notin A_\sim$ we deduce $c \not\sim'' \epsilon$. As $A_{\sim''} = A_\sim \cup \{\delta, c\}$, we have $A_{\sim''} \cap L = A_\sim \cap L = A_\sqsubseteq$ and thus $L \setminus A_{\sim''} = L \setminus A_\sqsubseteq$. Hence have $a \neq b \in L \setminus A_{\sim''}$. Thus $ab \rightarrow c$ is BBI-elementary w.r.t. \sim'' . Hence, $\sim' = \sim'' + \{ab \rightarrow c\}$ is BBI-elementary, has no square and $I_{\sim'} = I_{\sim''}$. We deduce $I_{\sim'} = I_\sim \subseteq M$.

Obviously $A_{\sim'} = A_\sim \cup \{a, b, \delta, c\}$. Let $d \in M$ such that $d \sim' d$. Then, either $d \in A_\sim$ or $d = c$. On the one hand, if $d \in A_\sim$ then $d \sim d$ and let $x \in L^*$ and $\alpha \in K^*$ such that $x\alpha \sim d$. Thus $x\alpha \sim' d$ because $\sim \subseteq \sim'$. On the other hand, if $d = c$ then $(ab)\epsilon \sim' c$ with $ab \in L^*$ and $\epsilon \in K^*$.

Let us prove $\sqsubseteq' \subseteq \sqsubseteq_{\sim'}^{L,K}$. As $\delta c \sim' m$ and $ab \sim' c$, by rule $\langle e_l \rangle$ we obtain $\delta ab \sim' m$, hence $ab \sqsubseteq_{\sim'}^{L,K} m$. As $\sim \subseteq \sim'$ we obviously have $\sqsubseteq = \sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq_{\sim'}^{L,K}$. Hence, $\sqsubseteq \cup \{ab \rightarrow m\} \subseteq \sqsubseteq_{\sim'}^{L,K}$. We get $\sqsubseteq' = \sqsubseteq + \{ab \rightarrow m\} \subseteq \sqsubseteq_{\sim'}^{L,K}$.

Let us now consider the converse inclusion $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. As $m \not\sim \epsilon$ and $c \not\sim'' \epsilon$ hence

$mm \approx mm$ and $cc \approx'' cc$, we have the following identities according to Lemma 6.4:

$$\begin{aligned}\sim'' &= \sim + \{\delta c \rightarrow m\} = \sim \cup \{\delta x \rightarrow \delta y, cx \rightarrow cy \mid x \sim y \wedge mx \sim my\} \\ &\quad \cup \{\delta cx \rightarrow \delta cy \mid mx \sim my\} \\ &\quad \cup \{\delta cx \rightarrow y, y \rightarrow \delta cx \mid mx \sim y\} \\ \sim' &= \sim'' + \{ab \rightarrow c\} = \sim'' \cup \{ax \rightarrow ay, bx \rightarrow by \mid x \sim'' y \wedge cx \sim'' cy\} \\ &\quad \cup \{abx \rightarrow aby \mid cx \sim'' cy\} \\ &\quad \cup \{abx \rightarrow y, y \rightarrow abx \mid cx \sim'' y\}\end{aligned}$$

Let $\gamma \in K^*$ and $x, y \in L^*$ such that $\gamma x \sim' y$. Let us prove that $x \sqsubseteq' y$. We study each case depending on the form of $(\gamma x, y)$:

- if $\gamma x \sim'' y$ then according to the equations for \sim'' , the only possibility for $\gamma x \sim'' y$ is when $\gamma x \sim y$: indeed, in the other cases, either δ occurs on the right (impossible because we would have $\delta \prec y$ with $\delta \notin L$) or c occurs on the left or on the right (impossible because $c \notin L \cup K$). Since $\gamma x \sim y$, we deduce $x \sqsubseteq y$ hence $x \sqsubseteq' y$;
- $(\gamma x, y) = (ax', ay')$ with $x' \sim'' y'$ and $cx' \sim'' cy'$. As $\gamma x = ax'$ and $a \not\prec \gamma$, then $a \prec x$ and let $x'' = x/a$. Then $x' = \gamma x''$, $\gamma x'' \sim'' y'$ and $c\gamma x'' \sim'' cy'$. According to the equations for \sim'' , as $cx' \sim'' cy'$ and $\delta \not\prec y'$, we must have $x' \sim y'$ and $mx' \sim my'$. Hence, $\gamma x'' \sim y'$ and $m\gamma x'' \sim my'$. Thus, $x'' \sqsubseteq y'$ and $mx'' \sqsubseteq my'$, and as $\sqsubseteq \subseteq \sqsubseteq'$, we obtain $x'' \sqsubseteq' y'$ and $mx'' \sqsubseteq' my'$. We produce the following deduction tree:

$$\frac{\frac{\frac{mx'' \sqsubseteq' my'}{my' \sqsubseteq' my'} \langle r \rangle \quad ab \sqsubseteq' m}{aby' \sqsubseteq' my'} \langle c \rangle}{\frac{\frac{ay' \sqsubseteq' ay'}{ay' \sqsubseteq' ay'} \langle pl \rangle \quad x'' \sqsubseteq' y'}{ax'' \sqsubseteq' ay'} \langle c \rangle}$$

Thus $x = ax'' \sqsubseteq' ay' = y$;

- $(\gamma x, y) = (bx', by')$ with $x' \sim'' y'$ and $cx' \sim'' cy'$. The same argument applies using b instead of a in the last two steps of the left branch of the preceding deduction tree. Hence $x \sqsubseteq' y$;
- $(\gamma x, y) = (abx', aby')$ with $cx' \sim'' cy'$. As $\gamma x = abx'$, we define $x'' = x/ab$ and we have $x = abx''$ and $x' = \gamma x''$. Hence, $\gamma x'' \sim'' cy'$. As $\delta \not\prec y'$, the only possibility is when $\gamma x'' \sim y'$ and $m\gamma x'' \sim my'$. Thus, $x'' \sqsubseteq y'$ and $mx'' \sqsubseteq my'$ and we can repeat the preceding deduction tree using ab instead of a in the last two steps of the left branch. Thus $x = abx'' \sqsubseteq' aby' = y$;
- $(\gamma x, y) = (abx', y)$ with $cx' \sim'' y$. Let $x'' = x/ab$. We have $x = abx''$ and $x' = \gamma x''$. Thus $c\gamma x'' \sim'' y$. As $c \not\prec y$, we must have $\delta \prec \gamma x''$ and $m(\gamma x'')/\delta \sim y$. As $\delta \not\prec x''$ (because $\delta \in K$ and $x'' \in L^*$) then $\delta \prec \gamma$. We thus have $m(\gamma/\delta)x'' \sim y$. Hence $mx'' \sqsubseteq y$. So $mx'' \sqsubseteq' y$. As $ab \sqsubseteq' m$, with rule $\langle e_l \rangle$, we get $x = abx'' \sqsubseteq' y$;
- $(\gamma x, y) = (\gamma x, aby')$ with $\gamma x \sim'' cy'$. As $c \not\prec \gamma x$, we must have $(\gamma x, cy') = (x'', \delta cy'')$ for some x'', y'' such that $my'' \sim x''$. But then $y' = \delta y''$ hence $\delta \prec y' \prec aby' = y$. This is not possible since $y \in L^*$ and $\delta \in K$;

We have proved that for any $\gamma \in K^*$ and $x, y \in L^*$, if $\gamma x \sim' y$ then $x \sqsubseteq' y$. Thus $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. So, in case 1, we have indeed proved that (\sqsubseteq', \sim') is a (L, K, M) elementary representation.

Let us consider case 3 where $\sqsubseteq' = \sqsubseteq + \{\delta m \rightarrow b\}$ and $\sim' = \sim + \{\delta m \rightarrow b\}$. First \sqsubseteq' is clearly BI-elementary. The constraint $\delta m \rightarrow b$ is obviously BBI-elementary w.r.t. \sim because $\delta \neq b$ and $\delta, b \notin A_\sim$ and then \sim' is BBI-elementary. According to Proposition 6.9, $\sim' = \sim + \{\delta m \rightarrow b\}$ has no square and $I_{\sim'} = I_\sim \subseteq M$.

We have $A_{\sim'} = A_\sim \cup \{b, \delta\}$. Let $d \in M$ such that $d \sim' d$. Then we must have $d \in A_\sim$. Let $x \in L^*$ and $\alpha \in K^*$ such that $x\alpha \sim d$. Hence $x\alpha \sim' d$.

As $\delta m \sim' b$, we deduce $m \sqsubseteq_{\sim'}^{L,K} b$ and thus $\sqsubseteq' = \sqsubseteq + \{\delta m \rightarrow b\} \subseteq \sqsubseteq_{\sim'}^{L,K}$. Let us consider the converse inclusion $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. We have the following identity according to Proposition 6.8:

$$\begin{aligned} \sim'' = \sim + \{\delta m \rightarrow b\} &= \sim \cup \{\delta x \rightarrow \delta y \mid x \sim y \wedge \exists k \, xk \sim m\} \\ &\quad \cup \{\delta x \rightarrow jb, jb \rightarrow \delta x \mid x \sim m \wedge j \sim \epsilon\} \\ &\quad \cup \{ib \rightarrow jb \mid i \sim \epsilon \wedge j \sim \epsilon\} \end{aligned}$$

Let $\gamma \in K^*$ and $x, y \in L^*$ such that $\gamma x \sim' y$. Let us prove that $x \sqsubseteq' y$. We study each case depending on the form of $(\gamma x, y)$:

- if $\gamma x \sim y$ then $x \sqsubseteq y$ hence $x \sqsubseteq' y$;
- $(\gamma x, y) = (\delta x', \delta y')$ is impossible because $\delta \not\sim y$;
- $(\gamma x, y) = (\delta x', jb)$ with $x' \sim m$ and $j \sim \epsilon$. Then $j \in I_\sim^*$ hence $j \in M^*$. As $j \prec jb = y$, we have $j \in L^*$. Thus $j \in L^* \cap M^*$. So $j = \epsilon$ and $y = b$. As $\delta \not\sim x$ and $\delta \prec \delta x' = \gamma x$, we have $\delta \prec \gamma$. So let γ' such that $\delta \gamma' = \gamma$. Then $\gamma' x = x'$. Hence, we get $\gamma' x \sim m$. Thus $x \sqsubseteq m$ and so $x \sqsubseteq' m$. As $m \sqsubseteq' b$, we obtain $x \sqsubseteq' b = y$ by rule $\langle t \rangle$;
- $(\gamma x, y) = (ib, \delta y')$ is impossible because $\delta \not\sim y$;
- $(\gamma x, y) = (ib, jb)$ with $i \sim \epsilon$ and $j \sim \epsilon$. Then $i, j \in I_\sim^*$, hence, $i, j \in M^*$. As $i \prec \gamma x$ and $j \prec y$, we must have $i = j = \epsilon$. Hence $x = y = b$ and $\gamma = \epsilon$. As $m \sqsubseteq' b$ we have $b \sqsubseteq' b$ by rule $\langle r \rangle$, thus $x \sqsubseteq' y$;

We have proved that for any $\gamma \in K^*$ and $x, y \in L^*$, if $\gamma x \sim' y$ then $x \sqsubseteq' y$. Thus $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. So, in case 3, we have indeed proved that $\sqsubseteq' = \sqsubseteq_{\sim'}^{L,K}$.

Case 3' is trivial because $\sim + \{\delta m \rightarrow b, \epsilon \rightarrow \epsilon\} = (\sim + \{\delta m \rightarrow b\}) + \{\epsilon \rightarrow \epsilon\} = \sim + \{\delta m \rightarrow b\}$ and we are thus back to case 3.

We temporarily skip case 4 and instead consider case 4' where $\sqsubseteq' = \sqsubseteq + \{\epsilon \rightarrow m\}$ and $\sim' = \sim + \{\epsilon \rightarrow c, m \rightarrow \delta\}$. First, \sqsubseteq' is clearly BI-elementary. As $c \notin A_\sim$, $\epsilon \rightarrow c$ is BBI-elementary w.r.t. \sim hence, according to Proposition 6.9, $\sim'' = \sim + \{\epsilon \rightarrow c\}$ has no square and $I_{\sim''} = I_\sim \cup \{c\} \subseteq M$. We have $m \not\sim'' \epsilon$: otherwise $m \sim'' \epsilon \Rightarrow m \in I_{\sim''}^* \Rightarrow m \in L^* \cap M^* \Rightarrow m = \epsilon$. Thus $m \rightarrow \delta$ is BBI-elementary w.r.t. \sim'' , so $\sim' = \sim'' + \{m \rightarrow \delta\}$ is BBI-elementary and according to Proposition 6.9, we have $I_{\sim'} = I_{\sim''} = I_\sim \cup \{c\} \subseteq M$.

By rule $\langle s \rangle$ we have $\delta \epsilon \sim' m$ so we deduce $\epsilon \sqsubseteq_{\sim'}^{L,K} m$ and thus $\sqsubseteq' = \sqsubseteq + \{\epsilon \rightarrow m\} \subseteq \sqsubseteq_{\sim'}^{L,K}$. Let us consider the converse inclusion $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. As $\sim'' + \{m \rightarrow \delta\} = \sim'' + \{\delta \rightarrow m\}$ by rule $\langle s \rangle$, we have the following identities according to Proposition 6.3 and Lemma 6.4

(as $m \approx'' \epsilon$ we have $mm \approx'' mm$):

$$\begin{aligned}\sim'' &= \sim + \{\epsilon \multimap c\} = \{c^p x \multimap c^q y \mid x \sim y \wedge p, q \geq 0\} \\ \sim' &= \sim'' + \{\delta \multimap m\} = \sim'' \cup \{\delta x \multimap \delta y \mid mx \sim'' my\} \\ &\quad \cup \{\delta x \multimap y, y \multimap \delta x \mid mx \sim'' y\}\end{aligned}$$

Let $\gamma \in K^*$ and $x, y \in L^*$ such that $\gamma x \sim' y$. Let us prove that $x \sqsubseteq' y$. We study each case depending on the form of $(\gamma x, y)$:

- if $\gamma x \sim'' y$ then $(\gamma x, y) = (c^p x', c^q y')$ with $x' \sim y'$ and $p, q \geq 0$. As $c \notin L \cup K$, we must have $p = q = 0$ and then $x' = \gamma x, y' = y$ and $\gamma x \sim y$. Thus $x \sqsubseteq y$ and we obtain $x \sqsubseteq' y$;
- $(\gamma x, y) = (\delta x', \delta y')$ is impossible because $\delta \not\prec y$;
- $(\gamma x, y) = (\delta x', y)$ with $mx' \sim'' y$. As $mx' \in L^*, y \in L^*$ and $c \notin L$, then we must have $mx' \sim y$. Then $\delta \prec \gamma$ and let $\gamma' \delta = \gamma$. Hence $\gamma' x = x'$ and thus $m\gamma' x \sim y$. So we get $mx \sqsubseteq y$ hence $mx \sqsubseteq' y$. But $\epsilon \sqsubseteq' m$ and we derive $x \sqsubseteq' y$ by rule $\langle e_l \rangle$;
- $(\gamma x, y) = (\gamma x, \delta y')$ is impossible because $\delta \not\prec y$;

We have proved that for any $\gamma \in K^*$ and $x, y \in L^*$, if $\gamma x \sim' y$ then $x \sqsubseteq' y$. Thus $\sqsubseteq_{\sim'}^{L,K} \subseteq \sqsubseteq'$. So, in case 4', we have indeed proved that $\sqsubseteq' = \sqsubseteq_{\sim'}^{L,K}$.

Case 4 is obtained from case 4'. Let $\sim'' = \sim + \{\epsilon \multimap c\}$. As $\epsilon \sim'' c$, we can easily prove that $\sim'' + \{\delta c \multimap m\} = \sim'' + \{\delta \multimap m\}$. Thus $\sim + \{\delta c \multimap m, \epsilon \multimap c\} = (\sim + \{\epsilon \multimap c\}) + \{\delta c \multimap m\} = \sim'' + \{\delta c \multimap m\} = \sim'' + \{\delta \multimap m\} = \sim'' + \{m \multimap \delta\} = (\sim + \{\epsilon \multimap c\}) + \{m \multimap \delta\} = \sim + \{\epsilon \multimap c, m \multimap \delta\}$ and we are back to case 4'. \square