Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER Electronic Notes in Theoretical Computer Science 125 (2005) 91–113

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

An Isomorph-Free SEM-Like Enumeration of Models [1]

Thierry Boy de la Tour[2] Prakash Countcham[3]

LEIBNIZ laboratory, IMAG - CNRS
INPG, 46 avenue Félix Viallet F-38031 Grenoble Cedex, France

Abstract

We investigate the integration of the enumeration of finite models of a formula, including unit propagation and pruning mechanisms, as provided by the system SEM, into McKay's general method of isomorph-free exhaustive enumeration. The two techniques turn out to be nicely compatible, though this requires some adaptations, and to prove some non-trivial properties.

Keywords: finite model building, isomorph-free search, computational group theory

1 Introduction

In the recent past much work has been devoted to the automated construction of finite models of first order formulas. Such models may bring much help either in automated reasoning (in semantic strategies) or for interactive use, for instance for debugging purposes, or simply for refuting a conjecture. Many systems have been developed, and many efforts devoted to reducing the huge search spaces involved. The most successful methods are those able to derive as much information as possible from the given formula.

Other powerful methods are used that are quite independent of the given formula: this is the case of *symmetries*. Systems like FMC (see [7]) and SEM (see [9]) both profit from techniques that help to eliminate counter-models

[1] This work has been supported by the CNRS and the MENRT
[2] Email:Thierry.Boy-de-la-Tour@imag.fr
[3] Email:Prakash.Countcham@imag.fr

doi:10.1016/j.entcs.2005.01.003

which present some restricted form of isomorphisms to known counter-models. SEM's method is the Least Number Heuristic, LNH, and has been extended to XLNH in [1], in order to account for more general isomorphisms (see our analysis in [4]). But no method has yet been found to account for *general* isomorphisms between interpretations.

There is however a general method for building isomorph-free enumerations of combinatorial structures, due to Brendan McKay, see [6]. We show how this method can be used in conjunction with the most powerful mechanisms of SEM, which are *unit propagation* and *partial evaluation* (i.e., evaluation in partial interpretations). We provide in Section 2 an abstract though accurate account of these features, and prove some original properties. In Section 3 we show that they are compatible with the necessary group theoretic framework. We then show how McKay's method can be used in this context, and in Section 4 we provide an algorithm SEMK that also includes the pruning mechanism of SEM. But SEMK is exhaustive only if called with many different inputs. We show in Section 5 that a slight modification ensures completeness with only one call to SEMK.

2 Unit Propagation in Partial Interpretations

We are given a sorted signature Σ, whose elements are *function symbols*, and a finite domain $\mathcal{D}$. There is a sort of boolean values, and $\mathcal{D}$ contains the boolean values $\top$ (true) and $\bot$ (false). The domain $\mathcal{D}$ is the disjoint union of the nonempty domains attributed to each sort, and we may say that each element of $\mathcal{D}$ has a unique sort. A function symbol whose range sort is the boolean sort is a *predicate* symbol.

For any symbol $f \in \Sigma$ of arity n, an *f-cell* is a tuple of the form $\langle f, v_1, \ldots, v_n \rangle$ where the v_i's are elements of $\mathcal{D}$. However, we will only consider the f-cells that are compatible with the sort profile of f as given in Σ, i.e., v_i is taken only in the domain associated to the sort of the i^{th} argument of f. The set of these compatible f-cells, for all symbols $f \in \Sigma$, will be noted $\mathfrak{C}$. The cell $\langle f, v_1, \ldots, v_n \rangle$ will be noted $f[v_1, \ldots, v_n]$.

A Σ-interpretation in $\mathcal{D}$ is a function from $\mathfrak{C}$ to $\mathcal{D}$, though obviously not all functions from $\mathfrak{C}$ to $\mathcal{D}$ can be considered as well-sorted interpretations, since once again we may only assign values of the correct sort to any given cell (this kind of restriction is a particular case of the notion of constraint used in SEM).

We are next given a first-order Σ-formula with equality φ; the problem addressed by SEM is to find a model (or all models) of φ in $\mathcal{D}$. The search for such models is considered in SEM as a Constraint Satisfaction Problem

(CSP).

We now give a short description of SEM. By suitable transformations described below we may restrict our considerations to the case where φ is a set of ground clauses. To each cell is initially associated a *set* of possible values. A search tree is then developed according to the following basic principle: at each node we choose a cell c whose value is not yet defined, i.e., it still has a set V of possible values. Then for each possible value $v \in V$ we recursively consider the case where c has the unique value v. At the leaves of the search tree all cells have a single value; these leaves correspond to all possible interpretations of Σ, and among them can be found all possible models of φ.

This basic search tree is pruned in two ways. The first is based on the fact that we may not need definite values for *all* cells in order to compute a truth value for φ. If for instance we have enough information to evaluate a clause of φ to false, then no further refinement may lead to a model, and it is therefore safe to backtrack. The second optimization is to use short-cuts down the tree, by directly inferring values for cells, from the knowledge that each clause must eventually become true. This is performed by unit propagation, thoroughly defined and analyzed below. But we first turn to evaluation with partial information.

2.1 Partial Interpretations and Evaluation

It is therefore convenient to define a *partial interpretation* as a binary relation $\mathcal{I}$ on $\mathfrak{C} \times \mathcal{D}$ such that to each cell corresponds at least one element in $\mathcal{D}$. For a cell c in $\mathfrak{C}$ and a relation $\mathcal{R} \subseteq \mathfrak{C} \times \mathcal{D}$, we note $\mathcal{R}[c]$ the set $\{v \in \mathcal{D} \,|\, c\,\mathcal{R}\,v\}$; the *domain* of $\mathcal{R}$ is the set $\mathrm{dom}(\mathcal{R}) = \{c \in \mathfrak{C} \,|\, \mathcal{R}[c] \neq \emptyset\}$. Therefore, the relation $\mathcal{I}$ is a partial interpretation when $\mathrm{dom}(\mathcal{I}) = \mathfrak{C}$. Given two partial interpretations $\mathcal{I}$ and $\mathcal{J}$, we say that $\mathcal{J}$ is a *refinement* of $\mathcal{I}$ if $\mathcal{J} \subseteq \mathcal{I}$.

The required restriction for the possible values of cells can then easily be expressed by considering only the refinements of a given *initial* partial interpretation $\mathcal{I}_0$. For instance, if $P \in \Sigma$ is a predicate symbol, and c a P-cell, then $\mathcal{I}_0[c]$ is the set of boolean values $\{\top, \bot\}$. For any $f \in \Sigma$ (except for some constants, see below), and any f-cell $c \in \mathfrak{C}$, the set $\mathcal{I}_0[c]$ is the domain associated to the range sort of f. We note $\mathfrak{I}$ the set of all partial interpretations $\mathcal{I}$ such that $\mathcal{I} \subseteq \mathcal{I}_0$.

For any relation $\mathcal{R} \subseteq \mathfrak{C} \times \mathcal{D}$ and any subset A of $\mathfrak{C}$, the *restriction* $\mathcal{R}|_A$ of $\mathcal{R}$ to A is the relation on $\mathfrak{C} \times \mathcal{D}$ defined by: $c\,\mathcal{R}|_A\,v$ iff $c\,\mathcal{R}\,v$ and $c \in A$. The *functional part* $\mathrm{fp}(\mathcal{R})$ of $\mathcal{R}$ is the biggest restriction $\mathcal{R}|_A$ which is a function (from A to $\mathcal{D}$). Finally, for any $\mathcal{I} \in \mathfrak{I}$, the *masking* of $\mathcal{I}$ by $\mathcal{R}$ is the relation

$\mathcal{I} \ll \mathcal{R}$ defined by:

$$c\ (\mathcal{I} \ll \mathcal{R})\ v \ \Leftrightarrow\ c\,\mathcal{R}\,v \text{ or } (\mathcal{R}[c] = \emptyset \text{ and } c\,\mathcal{I}\,v).$$

Example 2.1 Consider two constants a and b, and two values 1 and 2 in the domain, and let $I = \{\langle a,1\rangle, \langle a,2\rangle, \langle b,1\rangle, \langle b,2\rangle\}$ and $R = \{\langle a,1\rangle\}$. Obviously, R is a partial function, while $\mathrm{fp}(I) = \emptyset$. We have $I \ll R = \{\langle a,1\rangle, \langle b,1\rangle, \langle b,2\rangle\}$, so that $\mathrm{fp}(I \ll R) = R$. Masking may also increase an interpretation (or change it in other ways), for instance we have $(I \ll R) \ll \{\langle a,1\rangle, \langle a,2\rangle\} = I$.

For any $\mathcal{I} \in \mathfrak{I}$, it is easy to see that if $\mathcal{R} \subseteq \mathcal{I}$, then $\mathcal{I} \ll \mathcal{R}$ is a refinement of $\mathcal{I}$, and is in $\mathfrak{I}$. For any cell c and any subset $E \subseteq \mathcal{D}$, we note $[c, E]$ the relation $\{\langle c, v\rangle \mid v \in E\}$. For $v \in \mathcal{D}$, we write $[c, v]$ for $[c, \{v\}]$. An *interpretation* is a partial interpretation $\mathcal{I} \in \mathfrak{I}$ which is a function, hence such that $\mathcal{I} = \mathrm{fp}(\mathcal{I})$.

The notion of Σ-term of a given sort is as usual. An *atomic formula* is either a Σ-term of boolean sort, or an equation $t = t'$ where t and t' are two Σ-terms of the same sort. A *literal* is a possibly negated atomic formula, a *clause* is a disjunction of literals, and a *formula* φ is a conjunction of clauses.

We do not consider variables and quantifiers for the following reason: in SEM each input clause with n universally quantified variables $C(x_1, \ldots, x_n)$ is replaced by the conjunction of the ground clauses $C(v_1, \ldots, v_n)$ for all possible[4] values $v_i \in \mathcal{D}$. Since the elements of $\mathcal{D}$ are not constant symbols, we replace each value $v \in \mathcal{D}$ by a special, new constant symbol a_v. These constants are special because they are given special values in $\mathcal{I}_0$; we take $\mathcal{I}_0[a_v] = \{v\}$, and of course we implicitly add them to Σ. We will not prove that any first order formula with equality ψ can thus be transformed into a set of ground clauses φ, such that searching models of ψ in $\mathcal{D}$ is equivalent (through a 1-1 correspondence) to searching models of φ among the refinements of $\mathcal{I}_0$.

Example 2.2 For instance, if the original (skolemized) formula contains the clause $P(x, y)$, and the universal variables x, y have the same sort, which is interpreted in the domain $\{1, 2\}$, then φ contains the clauses $P(a_1, a_1)$, $P(a_1, a_2)$, $P(a_2, a_1)$ and $P(a_2, a_2)$. Moreover, we have $\mathcal{I}_0[a_1] = \{1\}$ and $\mathcal{I}_0[a_2] = \{2\}$, while for all P-cells c we have $\mathcal{I}_0[c] = \{\top, \bot\}$.

Another special property of these new symbols is that for any $u, v \in \mathcal{D}$ of the same sort, for any clause C in φ where a_u occurs, there is a clause C' in φ obtained from C by replacing a_u by a_v.

We now need to extend the well-known notion of the *value* of a term, literal, clause or formula *in* an interpretation, to the notion of value in a *partial* interpretation. But what could the value $[\![\varphi]\!]_{\mathcal{I}}$ of a formula φ in a

[4] This is restricted by the sorts of the variables x_i. This transformation is of course exponential in n.

partial interpretation $\mathcal{I}$ be? An obvious answer is to take the *set* of values of φ for all interpretations included in $\mathcal{I}$. But this would of course solve the satisfiability problem, hence would require exponential time to compute. SEM consequently adopts a more efficient computation rule, yielding a *superset* of the set of all possible values.

Definition 2.3 The *value* $[\![t]\!]_\mathcal{I}$ of a Σ-term t in a partial interpretation $\mathcal{I}$ is defined inductively as:

$$[\![f(t_1,\ldots,t_n)]\!]_\mathcal{I} = \begin{cases} \mathcal{I}[\,f[v_1,\ldots,v_n]\,] & \text{if } |[\![t_i]\!]_\mathcal{I}| = 1 \text{ for all } 1 \le i \le n, \\ \mathcal{I}_0[\,f[v_1,\ldots,v_n]\,] & \text{otherwise,} \end{cases}$$

where $v_i \in [\![t_i]\!]_\mathcal{I}$ for all $1 \le i \le n$. Note that $\mathcal{I}_0[f[v_1,\ldots,v_n]]$ does not depend on the particular v_i's chosen in $[\![t_i]\!]_\mathcal{I}$. Note that $[\![t]\!]_\mathcal{I}$ cannot be empty.

This definition is valid for atomic formulas other than equations. The *value* of an equation $t = t'$ is similarly defined by:

$$[\![t = t']\!]_\mathcal{I} = \begin{cases} \{\top\} & \text{if } |[\![t]\!]_\mathcal{I}| = |[\![t']\!]_\mathcal{I}| = 1 \text{ and } [\![t]\!]_\mathcal{I} = [\![t']\!]_\mathcal{I}, \\ \{\bot\} & \text{if } [\![t]\!]_\mathcal{I} \cap [\![t']\!]_\mathcal{I} = \emptyset. \\ \{\top,\bot\} & \text{otherwise,} \end{cases}$$

The *value* of any literal is thus obviously defined. We then define the *value* of a clause C (resp. a formula φ) in $\mathcal{I}$ as the set of all truth values obtained by disjunction (resp. conjunction) of all the possible truth values of its literals (resp. clauses) in $\mathcal{I}$. In other words, we have

$$\begin{array}{lll} \bot \in [\![C]\!]_\mathcal{I} & \text{iff} & \forall l \in C, \bot \in [\![l]\!]_\mathcal{I} \\ \top \in [\![C]\!]_\mathcal{I} & \text{iff} & \exists l \in C, \top \in [\![l]\!]_\mathcal{I} \\ \top \in [\![\varphi]\!]_\mathcal{I} & \text{iff} & \forall C \in \varphi, \top \in [\![C]\!]_\mathcal{I} \\ \bot \in [\![\varphi]\!]_\mathcal{I} & \text{iff} & \exists C \in \varphi, \bot \in [\![C]\!]_\mathcal{I}. \end{array}$$

Since the formula φ is an input of the search process, for convenience we may keep it implicit, and note $\widehat{\mathcal{I}}$ for $[\![\varphi]\!]_\mathcal{I}$. An interpretation $\mathcal{I}$ is a *model* (resp. a *counter-model*) if $\widehat{\mathcal{I}} = \{\top\}$ (resp. $\{\bot\}$).

Example 2.4 As an example, we consider a constant a and a unary function symbol f, and two values 1 and 2. We therefore have

$$I_0 = \{\langle f[1],1\rangle,\ \langle f[1],2\rangle,\ \langle f[2],1\rangle,\ \langle f[2],2\rangle,\ \langle a,1\rangle,\ \langle a,2\rangle,\ \langle a_1,1\rangle,\ \langle a_2,2\rangle\},$$

and we consider the clause $f(a) = a$. We have $[\![a]\!]_{I_0} = \{1,2\}$, hence $[\![f(a)]\!]_{I_0} = \{1,2\}$, and therefore $[\![f(a) = a]\!]_{I_0} = \{\top,\bot\}$.

This evaluation mechanism is *monotonic* in the sense that, if $\mathcal{J} \subseteq \mathcal{I}$ are two partial interpretations, then for any term t we have $[\![t]\!]_\mathcal{J} \subseteq [\![t]\!]_\mathcal{I}$, which can easily be proved by induction on t. This property then obviously extends to literals and clauses, hence:

$$\mathcal{J} \subseteq \mathcal{I} \Rightarrow \widehat{\mathcal{J}} \subseteq \widehat{\mathcal{I}}.$$

The search for a model in SEM is guided by the evaluation of $\widehat{\mathcal{I}}$ for successive refinements $\mathcal{I}$ of $\mathcal{I}_0$, based on the following property, which follows directly from the previous one:

For all $\mathcal{I} \in \mathfrak{I}$, if there is a $\mathcal{J} \subseteq \mathcal{I}$ which is a model of φ, then $\top \in \widehat{\mathcal{I}}$.

Hence the search may be pruned if $\widehat{\mathcal{I}} = \{\bot\}$. We now show how the search is conducted, by propagating refinements obtained from unit clauses.

2.2 Unit propagation

Unit propagation is an important feature of SEM. We have to define precisely in our setting, in order to prove a confluence property that is not essential to SEM, but that will be needed later.

Definition 2.5 We say that a term t is *definable* in $\mathcal{I}$ if $[\![s]\!]_{\mathcal{I}}$ is a singleton for all strict subterms s of t. If $t = f(t_1, \ldots, t_n)$ we have unique values $v_i \in [\![t_i]\!]_{\mathcal{I}}$, and we therefore associate to t the cell $\mathrm{cell}_{\mathcal{I}}(t) = f[v_1, \ldots, v_n]$. A definable term t in $\mathcal{I}$ is *defined* in $\mathcal{I}$ if $[\![t]\!]_{\mathcal{I}}$ is a singleton. In both cases, by Definition 2.3 we have $[\![t]\!]_{\mathcal{I}} = \mathcal{I}[\mathrm{cell}_{\mathcal{I}}(t)]$.

This definition holds for atomic formulas of the form $P(t_1, \ldots, t_n)$, by considering them as terms (of boolean sort). Similarly, an equation $t = t'$ is *definable* in $\mathcal{I}$ if t and t' are definable in $\mathcal{I}$; and it is moreover *defined* in $\mathcal{I}$ if $[\![t = t']\!]_{\mathcal{I}}$ is a singleton. Note however that an equation $t = t'$ may be defined though t or t' aren't (if their undefined values are disjoint).

A conjunction, disjunction or negation φ is *definable* in $\mathcal{I}$ whenever all its conjuncts, disjuncts or negated formula are definable in $\mathcal{I}$; and it is moreover *defined* in $\mathcal{I}$ if $[\![\varphi]\!]_{\mathcal{I}}$ is a singleton.

A clause C is *unit* in $\mathcal{I}$ if it is definable yet undefined in $\mathcal{I}$, and contains only one literal l which is not defined in $\mathcal{I}$. We refer to l as the *unit literal* in C (relative to $\mathcal{I}$).

Example 2.6 Following Example 2.4, a is not defined in I_0 (though it is definable, as any constant), hence $f(a)$ is not definable in I_0. However, if we take $I_1 = I_0 \ll [a, 1]$, then a is clearly defined in I_1, hence $f(a)$ is definable, and $\mathrm{cell}_{I_1}(f(a)) = f[1]$. The literal $f(a) = a$ is also definable in I_1. Hence the clause $f(a) = a$ is unit in I_1.

It is of course possible to draw useful information from the knowledge that a unit clause of φ, hence its unit literal, must be true. We thus define an abstract reduction relation on $\mathfrak{I}$:

Definition 2.7 For any clause C of φ we define a *refiner* relation $\mathcal{R}_{C,\mathcal{I}}$. If C is unit in $\mathcal{I}$, with unit literal l, we may apply one of the following rules:

(i) if l is $P(t_1, \ldots, t_n)$, then $\mathcal{R}_{C,\mathcal{I}} = [\mathrm{cell}_{\mathcal{I}}(P(t_1, \ldots, t_n)), \top]$,

(ii) if l is $\neg P(t_1, \ldots, t_n)$, then $\mathcal{R}_{C,\mathcal{I}} = [\mathrm{cell}_{\mathcal{I}}(P(t_1, \ldots, t_n)), \bot]$,

(iii) if l is $t_1 = t_2$, then $\mathcal{R}_{C,\mathcal{I}} = [c_1, E] \cup [c_2, E]$, where $E = \mathcal{I}[c_1] \cap \mathcal{I}[c_2]$,

(iv) if l is $t_1 \neq t_2$ and $|\mathcal{I}[c_2]| = 1$, then $\mathcal{R}_{C,\mathcal{I}} = [c_1, \mathcal{I}[c_1] \setminus \mathcal{I}[c_2]]$,

(v) if l is $t_1 \neq t_2$ and $|\mathcal{I}[c_1]| = 1$, then $\mathcal{R}_{C,\mathcal{I}} = [c_2, \mathcal{I}[c_2] \setminus \mathcal{I}[c_1]]$,

where $c_i = \mathrm{cell}_{\mathcal{I}}(t_i)$. In all other cases, we take $\mathcal{R}_{C,\mathcal{I}} = \emptyset$.

For any two partial interpretations $\mathcal{I}, \mathcal{J} \in \mathfrak{I}$, we say that $\mathcal{I}$ *unit-refines* to $\mathcal{J}$, and write $\mathcal{I} \rightarrowtail_{\varphi} \mathcal{J}$, if there exists a clause C in φ, such that $\mathcal{J} = \mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}$ and $\mathcal{J} \neq \mathcal{I}$. As above, we generally omit φ and just write $\mathcal{I} \rightarrowtail \mathcal{J}$. If there is a $\mathcal{K}$ such that $\mathcal{I} \rightarrowtail^{\star} \mathcal{K}$ and $\mathcal{J} \rightarrowtail^{\star} \mathcal{K}$, we say that $\mathcal{I}$ and $\mathcal{J}$ are *joinable*, and write $\mathcal{I} \downarrow \mathcal{J}$. If there is no $\mathcal{J}$ such that $\mathcal{I} \rightarrowtail \mathcal{J}$, then $\mathcal{I}$ is a *normal form.*

It is easy to see that $\mathcal{R}_{C,\mathcal{I}}$ is included in $\mathcal{I}$, hence so is $\mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}$. Therefore, if $\mathcal{I} \rightarrowtail \mathcal{J}$ then $\mathcal{J} \subsetneq \mathcal{I}$, which proves that the unit-refinement reduction $\rightarrowtail$ terminates. In Example 2.6, the clause $f(a) = a$ is unit in I_1, and we get the refiner $\mathcal{R}_{f(a)=a,I_1} = [f[1], 1]$, so that $I_1 \rightarrowtail I_1 \ll [f[1], 1]$; this assigns the value 1 to the cell $f[1]$.

Each time a value is assigned to a cell, new clauses may become unit, and thus induce further refinements of $\mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}$. This is why $\rightarrowtail$ is not obviously confluent, and in fact, the following example shows that it is *not* confluent.

Example 2.8 Consider two propositional variables P and Q, and let $I = \{\langle P, \top\rangle, \langle P, \bot\rangle, \langle Q, \bot\rangle\}$. We consider the formula φ with two clauses $C = P \vee Q$ and $C' = \neg P \vee Q$, which are both unit in I. Let $J = I \ll \mathcal{R}_{C,I} = \{\langle P, \top\rangle, \langle Q, \bot\rangle\}$ and $J' = I \ll \mathcal{R}_{C',I} = \{\langle P, \bot\rangle, \langle Q, \bot\rangle\}$, we have $I \rightarrowtail J$ and $I \rightarrowtail J'$, and of course $J \neq J'$. No clause is unit in J or J', hence they are normal forms. They are also both counter-models of φ.

This last fact however suggests that a restricted form of local confluence may hold. This is indeed the case, as we now prove. Since we have to consider all five rules of Definition 2.7 the proof is somewhat tedious, though not very difficult, so we give it in the Appendix.

Lemma 2.9 *If $\mathcal{I} \rightarrowtail \mathcal{J}$, $\mathcal{I} \rightarrowtail \mathcal{J}'$, and $\top \in \widehat{\mathcal{J}}$ or $\top \in \widehat{\mathcal{J}'}$ then $\mathcal{J} \downarrow \mathcal{J}'$.*

Note that even if both $\mathcal{J}$ and $\mathcal{J}'$ are non-counter-models, it may be the case that they only join on a counter-model. Hence we have not proven local confluence of the *restriction* of $\rightarrowtail$ to non-counter-models, and cannot use

Newman's Lemma (see e.g. [2]) to get a confluence result for this restriction. We can still adapt the proof of Newman's Lemma to obtain a computationally meaningful notion of a normal form.

Theorem 2.10 *If $\mathcal{I} \rightarrowtail^\star \mathcal{J}$ and $\mathcal{K}$ is a normal form of $\mathcal{I}$ (i.e., such that $\mathcal{I} \rightarrowtail^\star \mathcal{K}$) and $\top \in \widehat{\mathcal{K}}$ then $\mathcal{J} \rightarrowtail^\star \mathcal{K}$.*

Proof. We prove by well-founded induction based on $\rightarrowtail$ that it is true for all $\mathcal{I}$. So we suppose that it is true for all $\mathcal{I}'$ such that $\mathcal{I} \rightarrowtail \mathcal{I}'$, and show that it must then be true for $\mathcal{I}$. If $\mathcal{I} = \mathcal{J}$ or $\mathcal{I} = \mathcal{K}$ then the conclusion $\mathcal{J} \rightarrowtail^\star \mathcal{K}$ is obvious.

We now suppose that $\mathcal{I} \neq \mathcal{J}$ and $\mathcal{I} \neq \mathcal{K}$, so we have $\mathcal{I} \rightarrowtail \mathcal{J}' \rightarrowtail^\star \mathcal{J}$ and $\mathcal{I} \rightarrowtail \mathcal{I}' \rightarrowtail^\star \mathcal{K}$. Since $\top \in \widehat{\mathcal{K}}$ and $\mathcal{K} \subseteq \mathcal{I}'$ we have $\top \in \widehat{\mathcal{I}'}$, and we may apply Lemma 2.9. We therefore have a partial interpretation $\mathcal{L}$ such that $\mathcal{I}' \rightarrowtail^\star \mathcal{L}$ and $\mathcal{J}' \rightarrowtail^\star \mathcal{L}$. By induction hypothesis, applied to $\mathcal{I}'$, we have $\mathcal{L} \rightarrowtail^\star \mathcal{K}$, hence $\mathcal{J}' \rightarrowtail^\star \mathcal{K}$. Again by induction hypothesis, applied to $\mathcal{J}'$, we obtain $\mathcal{J} \rightarrowtail^\star \mathcal{K}$. □

It is then clear that $\mathcal{I}$ can have at most one normal form which is not a counter-model. Moreover, its existence is easily decided:

Corollary 2.11 *If $\mathcal{I} \rightarrowtail^\star \mathcal{J}$ and $\widehat{\mathcal{J}} = \{\bot\}$, then $\mathcal{I}$ has no normal form $\mathcal{K}$ such that $\top \in \widehat{\mathcal{K}}$.*

Proof. Suppose $\mathcal{K}$ is a normal form of $\mathcal{I}$ and $\top \in \widehat{\mathcal{K}}$, then by Theorem 2.10 we have $\mathcal{J} \rightarrowtail^\star \mathcal{K}$. We then have $\mathcal{K} \subseteq \mathcal{J}$, hence $\top \in \widehat{\mathcal{J}}$, a contradiction. □

This means that, if we find a contradiction (i.e., a counter-model) in the course of computing a normal form of $\mathcal{I}$, then we know that no other sequence of unit-reductions would lead to a normal form which is not a counter-model. In other words, we only lose confluence on counter-models. Of course, we may stop reducing when a contradiction is found.

Definition 2.12 If $\mathcal{I}$ has a normal form $\mathcal{J}$ such that $\top \in \widehat{\mathcal{J}}$, then we note it $\mathcal{I}{\downarrow}$. Otherwise, $\mathcal{I}{\downarrow}$ denotes any counter-model such that $\mathcal{I} \rightarrowtail^\star \mathcal{I}{\downarrow}$. Despite this last non-determinism, we call $\mathcal{I}{\downarrow}$ *the normal form* of $\mathcal{I}$.

In Figure 1 we give the version of SEM that does not stop on the first model of φ that it finds. It uses the Assign-and-Propagate function $\mathrm{AP}_\varphi(\mathcal{I}, c, v) = (\mathcal{I} \ll [c, v]){\downarrow}$, that assigns the value v to the cell c, and propagates this value through unit-refinements, by computing the normal form of $\mathcal{I} \ll [c, v]$.

Example 2.13 Following Example 2.4, where φ only contains the clause $f(a) = a$, we have $\mathrm{AP}_\varphi(I_0, a, 1) = I_1{\downarrow}$ where $I_1 = I_0 \ll [a, 1]$. As noted above, we have $\mathcal{R}_{f(a)=a, I_1} = [f[1], 1]$, so that $I_1 \rightarrowtail I_2 = I_1 \ll [f[1], 1]$. Note

```
$\mathrm{SEM}_\varphi(\mathcal{I}) =$ if $\top \notin \widehat{\mathcal{I}}$ then $\emptyset$
                  else if $\bot \notin \widehat{\mathcal{I}}$ then $\{\mathcal{I}\}$
                  else
                     choose $c \in \mathfrak{C}$ such that $\mathcal{I}[c]$ is not a singleton;
                     $\bigcup_{v \in \mathcal{I}[c]} \mathrm{SEM}_\varphi(\ \mathrm{AP}_\varphi(\mathcal{I}, c, v)\ )$
```

Fig. 1. An abstract of SEM

that $[\![f(a) = a]\!]_{I_2} = \{\top\}$, so that no clause is unit in I_2, and I_2 is not a counter-model, hence $I_1{\downarrow} = I_2$.

Of course $\mathrm{SEM}_\varphi(\mathcal{I}_0{\downarrow})$ is complete in the sense that for any interpretation $\mathcal{M}$ which is a model of φ, there is a model $\mathcal{M}' \in \mathrm{SEM}_\varphi(\mathcal{I}_0{\downarrow})$ such that $\mathcal{M} \subseteq \mathcal{M}'$. It is easy to see that for any two different models $\mathcal{M}, \mathcal{M}' \in \mathrm{SEM}_\varphi(\mathcal{I}_0{\downarrow})$, we have $\mathcal{M} \not\subseteq \mathcal{M}'$. However, for any $\mathcal{M} \in \mathrm{SEM}_\varphi(\mathcal{I}_0{\downarrow})$, and any $\mathcal{M}'$ which is isomorphic to $\mathcal{M}$, we have $\mathcal{M}' \in \mathrm{SEM}_\varphi(\mathcal{I}_0{\downarrow})$. In order to prune the search space so as to obtain an isomorph-free search, we now give a group theoretic account of isomorphisms.

It should first be mentioned that the very simple description of SEM given above misses an important source of efficiency, obtained by a complex backtrack mechanism. But since we are going to replace the enumeration provided by the algorithm SEM above by a rather different algorithm, it is not possible to recover this backtracking as such. We rather focus on propagation, which is probably the most important feature of SEM.

3 The Invariance of Propagation

Our aim is to reduce the search space of SEM by making it *isomorph-free.* This means that once a partial interpretation has been considered, than no partial interpretation isomorphic to this one will be considered. The notion of isomorphism between Σ-interpretations is well-known: they are special bijections between their carrier sets. But since we have only one carrier set, namely $\mathcal{D}$, our isomorphisms are permutations of $\mathcal{D}$. It is therefore more convenient to define isomorphisms starting from permutation groups.

So we first consider the group $\mathrm{Sym}(\mathcal{D})$ of permutations σ of $\mathcal{D}$. For each $v \in \mathcal{D}$, the image of v by σ is noted v^σ, as is standard in group theory. The product in $\mathrm{Sym}(\mathcal{D})$ is defined by $v^{\sigma\sigma'} = (v^\sigma)^{\sigma'}$. An *action* of a group G on a set S is a morphism h from G to the group $\mathrm{Sym}(S)$. If only one action is defined on S, it is unambiguous to write s^σ for $s^{h(\sigma)}$, for any $s \in S$ and $\sigma \in G$. For example, if σ is a permutation of integers, say $\sigma = (2\ 3)$, we may apply it directly to a set of integers, say $\{1,2\}^{(2\ 3)} = \{1,3\}$, rather than translating explicitly σ as a permutation of sets of integers, which would yield $h(\sigma) = (\{1,2\}\ \{1,3\})(\{2\}\ \{3\})(\{2,4\}\ \{3,4\})$ etc.

The *G-orbit* of an element $s \in S$ is the set $s^G = \{s^\sigma \mid \sigma \in G\}$, and the *automorphism group* of s is the set $\mathrm{Aut}(s) = \{\pi \in G \mid s^\pi = s\}$, which is a subgroup of G. For H a subgroup of G, a *coset* of H in G is a set $H\sigma = \{\mu\sigma \mid \mu \in H\}$, or a set $\sigma H = \{\sigma\mu \mid \mu \in H\}$, for a $\sigma \in G$. It is easy to see that for all $\sigma \in G$, we have $\mathrm{Aut}(s^\sigma) = \{\sigma^{-1}\pi\sigma \mid s^\pi = s\} = \sigma^{-1}\mathrm{Aut}(s)\sigma$.

We define an action of $\mathrm{Sym}(\mathcal{D})$, first on Σ, by taking $f^\sigma = f$ for all $f \in \Sigma$ except the special constants a_v for $v \in \mathcal{D}$, for which we take $(a_v)^\sigma = a_{v^\sigma}$. We then define an action on the set $\mathfrak{C}$ of cells by: $(f[v_1,\ldots,v_n])^\sigma = f^\sigma[v_1^\sigma,\ldots,v_n^\sigma]$. We further define an action on the set of binary relations on $\mathfrak{C} \times \mathcal{D}$ by taking $\mathcal{R}^\sigma = \{\langle c^\sigma, v^\sigma\rangle \mid c\,\mathcal{R}\,v\}$. In other words, we have $c\,\mathcal{R}\,v$ iff $c^\sigma\,\mathcal{R}^\sigma\,v^\sigma$, which can also be written $\mathcal{R}^\sigma[c^\sigma] = (\mathcal{R}[c])^\sigma = \{v^\sigma \mid v \in \mathcal{R}[c]\}$. It is easy to see that for any partial interpretation $\mathcal{I}$, the relation $\mathcal{I}^\sigma$ is also a partial interpretation. Moreover, since $\mathcal{I}[a_v] = \{v\}$, then $\mathcal{I}^\sigma[a_v] = (\mathcal{I}[a_v^{\sigma^{-1}}])^\sigma = \{v^{\sigma^{-1}}\}^\sigma = \{v\}$.

Example 3.1 Following Example 2.4, we let σ be the permutation swapping values 1 and 2, which is noted in cycle notation by $\sigma = (1\ 2)$. We may apply it to the cell $f[1]$, which yields $(f[1])^\sigma = f^\sigma[1^\sigma] = f[2]$. Similarly, we have $[f[1],2]^\sigma = [f[2],1]$, and the reader can check that $(I_0)^\sigma = I_0$.

In the sequel, we consider the following group of *isomorphisms*[5]

$$\mathfrak{G} \;=\; \{\sigma \in \mathrm{Sym}(\mathcal{D}) \mid\ \mathcal{I}_0^\sigma = \mathcal{I}_0\ \wedge \top^\sigma = \top\ \},$$

which means that we enforce the preservation of sorts *and* of truth values. We now define the action of $\mathfrak{G}$ on syntactic objects. For any Σ-term t and any permutation $\sigma \in \mathfrak{G}$, the term t^σ is simply obtained from t by replacing all symbols f by f^σ. The definition is similar for literals, clauses and formulas. Note that for any clause C of φ, the clause C^σ is also a clause of φ (due to the special property of the constants a_v mentioned above), so that $\varphi^\sigma = \varphi$ (up to the AC property of conjunction).

It is now very easy to prove, by structural induction, that for all terms t we have $[\![t^\sigma]\!]_{\mathcal{I}^\sigma} = ([\![t]\!]_{\mathcal{I}})^\sigma$. Hence for all literals l we have $[\![l^\sigma]\!]_{\mathcal{I}^\sigma} = [\![l]\!]_{\mathcal{I}}$, and for all clauses C we have $[\![C^\sigma]\!]_{\mathcal{I}^\sigma} = [\![C]\!]_{\mathcal{I}}$. Finally, we have

$$\forall\sigma \in \mathfrak{G},\ \widehat{\mathcal{I}^\sigma} \;=\; \widehat{\mathcal{I}}.$$

As usual, we say that the partial interpretations $\mathcal{I}$ and $\mathcal{I}^\sigma$ are *isomorphic*; this defines an equivalence relation. The isomorphism class of $\mathcal{I}$ is the $\mathfrak{G}$-orbit of $\mathcal{I}$.

[5] Any element in the group is an isomorphism, in the usual sense, between two suitable interpretations.

We now show that SEM's propagation is compatible with the action of $\mathfrak{G}$. First, it is obvious that for any term t and literal l, t and l are defined (resp. definable) in $\mathcal{I}$ if and only if t^σ and l^σ are defined (resp. definable) in $\mathcal{I}^\sigma$. Therefore, a clause C is unit in $\mathcal{I}$ iff C^σ is unit in $\mathcal{I}^\sigma$. In this case we have the refiner relation $\mathcal{R}_{C^\sigma,\mathcal{I}^\sigma}$, and the reader can easily check that it is exactly $(\mathcal{R}_{C,\mathcal{I}})^\sigma$. Finally, it is trivial to prove that for any relation $\mathcal{R}$ we have $\mathcal{I}^\sigma \ll \mathcal{R}^\sigma = (\mathcal{I} \ll \mathcal{R})^\sigma$, which clearly yields

$$\mathcal{I} \rightarrowtail \mathcal{J} \;\Rightarrow\; \mathcal{I}^\sigma \rightarrowtail \mathcal{J}^\sigma.$$

Therefore, if $\mathcal{I}\downarrow$ is not a counter-model, we get $\mathcal{I}^\sigma\downarrow = (\mathcal{I}\downarrow)^\sigma$, and this is not a counter-model either. If $\mathcal{I}\downarrow$ is a counter-model, we still have $\mathcal{I}^\sigma \rightarrowtail^\star (\mathcal{I}\downarrow)^\sigma$, which is also a counter-model. The freedom given by the non-determinism in the definition of the normal-forms of $\mathcal{I}$ and $\mathcal{I}^\sigma$ therefore leaves us the possibility to choose $(\mathcal{I}\downarrow)^\sigma$ as the value of $\mathcal{I}^\sigma\downarrow$. This yields the invariance of the Assign-and-Propagate function:

$$\mathrm{AP}_\varphi(\mathcal{I}^\sigma, c^\sigma, v^\sigma) \;=\; [\mathrm{AP}_\varphi(\mathcal{I}, c, v)]^\sigma.$$

If in practice the property is not true on counter-models, this is not a problem since counter-models are immediately pruned. The argument above proves that this pruning is invariant under the action of $\mathfrak{G}$.

4 McKay's Exhaustive Enumeration

McKay's algorithm is general in the sense that it leaves some freedom in the definition of a number of mathematical objects, which should of course meet a number of properties. We will see how this freedom allows us to use the propagation defined above, and what problems this use entails. We also try to provide an explanation of McKay's method, which is by far nontrivial, while building our objects.

4.1 Canonical Elements

In the beginning are "labeled" and "unlabeled" objects: the former are the objects we know how to enumerate (i.e., partial interpretations), and the latter are the isomorphism classes of labeled objects, that we wish to enumerate. The terminology comes from graph theory, since a graph's vertices are actual elements of an actual set (the labels), while an isomorphism class of graphs can be seen as one abstract graph, or a Platonic notion of a graph, hence unlabeled.

A more trivial example is provided by representing sets as lists. Lists are the labeled objects, and many different lists represent the same set. By giving a suitable definition of isomorphism between lists (by permuting the elements' positions), we can obtain the property that two lists represent the same set exactly when they are isomorphic. Sets are then the unlabeled objects, and correspond exactly to the isomorphism classes of lists.

An essential requirement in the method is to be able to compute a single canonical element for each isomorphism class. In the case of lists, this can be performed efficiently by sorting. But in the case of partial interpretations, this problem is equivalent (by polynomial time Turing reductions) to the problem of deciding whether two graphs are isomorphic (see e.g. [3]). This is a problem in **NP** which is not (known to be) in **P**, but is however known not to be **NP**-complete (conjecturing that the polynomial hierarchy does not collapse, see [8]). This nice theoretical property is met in practice by the very efficient and complex algorithm *nauty*, also by McKay (see [5]). We just mention that it can be used to compute a canonical form of any partial interpretation $\mathcal{I}$, and incidentally a generating set of the group $\mathrm{Aut}(\mathcal{I})$.

More precisely, the canonical element in $\mathcal{I}^{\mathfrak{G}}$ is reached from $\mathcal{I}$ by applying a canonical permutation $\gamma(\mathcal{I})$, i.e., the canonical element is $\mathcal{I}^{\gamma(\mathcal{I})}$. It is canonical since it can be reached from any element in $\mathcal{I}^{\mathfrak{G}}$, hence for all $\sigma \in \mathfrak{G}$ we have $\mathcal{I}^{\gamma(\mathcal{I})} = \mathcal{I}^{\sigma\gamma(\mathcal{I}^{\sigma})}$. This is strictly equivalent to saying that $\mu = \gamma(\mathcal{I})(\sigma\gamma(\mathcal{I}^{\sigma}))^{-1}$ is an automorphism of $\mathcal{I}$. But saying that $\mu \in \mathrm{Aut}(\mathcal{I})$ is equivalent to saying that the coset $\mu\mathrm{Aut}(\mathcal{I})$ is $\mathrm{Aut}(\mathcal{I})$. Hence this is equivalent to:

$$\begin{aligned} &\gamma(\mathcal{I})\gamma(\mathcal{I}^{\sigma})^{-1}\sigma^{-1}\mathrm{Aut}(\mathcal{I}) = \mathrm{Aut}(\mathcal{I}) \\ \Leftrightarrow\quad &\gamma(\mathcal{I}^{\sigma})^{-1}\sigma^{-1}\mathrm{Aut}(\mathcal{I}) = \gamma(\mathcal{I})^{-1}\mathrm{Aut}(\mathcal{I}). \end{aligned}$$

4.2 Inductive Construction

How can we enumerate the canonical elements by *efficiently* eliminating non-canonical elements? The idea of McKay's method is to model the construction of canonical elements as an inductive process, compatible with $\mathfrak{G}$. Objects can generally be built from smaller objects, by an easy and efficient enumeration (think of lists). This results in a construction tree, on which the notion of isomorphism between objects can be applied, yielding a notion of isomorphic subtrees[6]. The idea behind McKay's algorithm is to prune all non-canonical branches, as soon as possible. This idea requires that we fill in a number of details.

We first ensure well-foundedness by attributing an integer to each object: its *order*, so that the order increases during the inductive construction. A natural choice for partial interpretations $\mathcal{I}$ is the cardinality of the functional

[6] since an object can usually be built in many different ways, we should speak of a directed acyclic graph rather than a tree. However, the structure does not exist explicitly in memory, hence it is as expensive to travel through as if it were a tree.

part $\mathrm{fp}(\mathcal{I})$. The order is minimal for $\mathcal{I}_0$, and maximal for interpretations.

We next have to define *lower objects* and *upper objects*. Intuitively, an upper object should contain all the necessary information to build one successor of $\mathcal{I}$ in the construction tree. In our case, each successor of $\mathcal{I}$ will be obtained by assigning a value v to a cell c, provided that $v \in \mathcal{I}[c]$, and also that $\mathcal{I}[c]$ is not reduced to the singleton $\{v\}$. The successor will then be $\mathrm{AP}_\varphi(\mathcal{I}, c, v)$, whose order is then guaranteed to be greater than $\mathcal{I}$'s order. We thus define the set of upper objects associated to $\mathcal{I}$ as:

$$\mathrm{U}(\mathcal{I}) \;=\; \{\; \langle \mathcal{I}, c, v\rangle \;\mid\; \{v\} \subsetneq \mathcal{I}[c] \;\}.$$

Similarly, lower objects should contain all the necessary information to compute all possible predecessors. A given partial interpretation $\mathcal{J}$ may be obtained as $\mathcal{J} = \mathrm{AP}_\varphi(\mathcal{I}, c, v)$ where $\mathcal{I}$ contains $\mathcal{J}$, and as above we must have $\{v\} \subsetneq \mathcal{I}[c]$. In order to avoid redundant information, we need only provide the relation $\mathcal{S} = \mathcal{I} \setminus \mathcal{J}$. Also note that $\mathcal{J}[c] = \{v\}$, so we define the set of lower objects associated to $\mathcal{J}$ as:

$$\begin{aligned}\mathrm{L}(\mathcal{J}) \;=\; \{\; \langle \mathcal{J}, \mathcal{S}, c\rangle \;\mid\; \exists v,\; & \mathcal{J} \cap \mathcal{S} = \emptyset \;\wedge\; \{v\} = \mathcal{J}[c] \;\wedge\; \mathcal{S}[c] \neq \emptyset \\ & \wedge\; \mathcal{J} = \mathrm{AP}_\varphi(\mathcal{J} \cup \mathcal{S}, c, v) \;\}.\end{aligned}$$

Note that the order of $\mathcal{J} \cup \mathcal{S}$ is strictly less than the order of $\mathcal{J}$. We now have to define actions of $\mathfrak{G}$ on lower and upper objects, and check that for all $\sigma \in \mathfrak{G}$ we have $\mathrm{L}(\mathcal{J}^\sigma) = \mathrm{L}(\mathcal{J})^\sigma$ and $\mathrm{U}(\mathcal{I}^\sigma) = \mathrm{U}(\mathcal{I})^\sigma$. Obviously, we take $\langle \mathcal{I}, c, v\rangle^\sigma = \langle \mathcal{I}^\sigma, c^\sigma, v^\sigma\rangle$ and $\langle \mathcal{J}, \mathcal{S}, c\rangle^\sigma = \langle \mathcal{J}^\sigma, \mathcal{S}^\sigma, c^\sigma\rangle$, and the two invariance properties are easy to check, by using the invariance of propagation.

Note that we have not used propagation in the definition of upper objects, hence these are not yet related to lower objects. McKay's method requires that we define such a relation. We note it $\mathfrak{R}$, and define it by:

$$\begin{aligned}&\langle \mathcal{J}, \mathcal{S}, c\rangle \;\mathfrak{R}\; \langle \mathcal{I}, c', v\rangle \text{ iff}\\ &\text{there is a } \sigma \text{ in } \mathfrak{G} \text{ such that } \mathcal{I}^\sigma = \mathcal{J} \cup \mathcal{S},\; c'^\sigma = c \;\text{ and }\; \mathcal{J}[c] = \{v^\sigma\}.\end{aligned}$$

As required, the order of $\mathcal{I}$ is smaller than the order of $\mathcal{J}$, and every lower object is related to an upper object. There are two other properties that $\mathfrak{R}$ should meet. The first is that if an upper object $\langle \mathcal{I}, c, v\rangle$ is related to some lower object, then it should also be the case for $\langle \mathcal{I}, c, v\rangle^\sigma$, for any $\sigma \in \mathfrak{G}$. Considering the definition of $\mathfrak{R}$, this is obvious. Finally, we let the reader check that, if we have both $\langle \mathcal{J}_1, \mathcal{S}_1, c_1\rangle \;\mathfrak{R}\; \langle \mathcal{I}_1, c'_1, v_1\rangle$ and $\langle \mathcal{J}_2, \mathcal{S}_2, c_2\rangle \;\mathfrak{R}\; \langle \mathcal{I}_2, c'_2, v_2\rangle$, then we have:

$$\langle \mathcal{J}_1, \mathcal{S}_1, c_1\rangle^{\mathfrak{G}} = \langle \mathcal{J}_2, \mathcal{S}_2, c_2\rangle^{\mathfrak{G}} \;\Leftrightarrow\; \langle \mathcal{I}_1, c'_1, v_1\rangle^{\mathfrak{G}} = \langle \mathcal{I}_2, c'_2, v_2\rangle^{\mathfrak{G}}.$$

```
$\mathrm{SEMK}_\varphi(\mathcal{I}) =$ let $A = \emptyset$ in
        for all $\mathrm{Aut}(\mathcal{I})$-orbit $O$ in $\mathrm{U}(\mathcal{I})$ do
          choose $\langle \mathcal{I}, c, v\rangle \in O$;
          let $\mathcal{J} = \mathrm{AP}_\varphi(\mathcal{I}, c, v)$ in
          if $\top \in \widehat{\mathcal{J}}$ and $\langle \mathcal{J}, \mathcal{I} \setminus \mathcal{J}, c\rangle \in m(\mathcal{J})$ then
            if $\bot \notin \widehat{\mathcal{J}}$ then $A := A \cup \{\mathcal{J}\}$
            else $A := A \cup \mathrm{SEMK}_\varphi(\mathcal{J})$
        done;
        $A$
```

Fig. 2. The algorithm SEMK

4.3 Selecting Lower Objects

The search for models will be pruned by constructing only the partial interpretations corresponding to selected lower objects. This selection is based on canonicity, through the following function:

$$m(\mathcal{J}) = \begin{cases} \emptyset & \text{if } \mathrm{L}(\mathcal{J}) = \emptyset, \\ \langle \mathcal{J}, \mathcal{S}, c\rangle^{\mathrm{Aut}(\mathcal{J})} & \text{otherwise,} \end{cases}$$

where $\langle \mathcal{J}, \mathcal{S}, c\rangle = l(\mathcal{J}^{\gamma(\mathcal{J})})^{\gamma(\mathcal{J})^{-1}}$. The function l, applied to $\mathcal{J}$, yields an element of $\mathrm{L}(\mathcal{J})$ if there is one (it is a choice function). As mentioned above, the computation of $\gamma(\mathcal{J})$ also yields generators for the group $\mathrm{Aut}(\mathcal{J})$. The conditions M1 and M2 in [6] are obvious by definition, and there only remains to show the invariance of m under $\mathfrak{G}$. For any $\sigma \in \mathfrak{G}$, we have $\mathrm{L}(\mathcal{J}^\sigma) = \emptyset$ iff $\mathrm{L}(\mathcal{J}) = \emptyset$, so in that case we obviously have $m(\mathcal{J}^\sigma) = m(\mathcal{J})^\sigma = \emptyset$. Otherwise, we have:

$$\begin{aligned} m(\mathcal{J}^\sigma) &= l(\mathcal{J}^{\sigma\gamma(\mathcal{J}^\sigma)})^{\gamma(\mathcal{J}^\sigma)^{-1}\mathrm{Aut}(\mathcal{J}^\sigma)} \\ &= l(\mathcal{J}^{\gamma(\mathcal{J})})^{\gamma(\mathcal{J}^\sigma)^{-1}\sigma^{-1}\mathrm{Aut}(\mathcal{J})\sigma} \\ &= l(\mathcal{J}^{\gamma(\mathcal{J})})^{\gamma(\mathcal{J})^{-1}\mathrm{Aut}(\mathcal{J})\sigma} \\ &= m(\mathcal{J})^\sigma. \end{aligned}$$

We now adapt the generating procedure from [6], in order to recover the pruning mechanism of SEM. This yields the algorithm SEMK given in Figure 2. Its result does not depend on the choice of $\langle \mathcal{I}, c, v\rangle$, which therefore could be performed non-deterministically. It is easy to see that the pruning performed according to the value of $\widehat{\mathcal{J}}$ (which does not exist in McKay's procedure) is correct in the sense that no branch leading to a canonical model can be pruned.

However, we still have a problem of completeness, since McKay's generating procedure is exhaustive only if applied to all *irreducible* objects, i.e., the partial interpretations $\mathcal{I}$ such that $\mathrm{L}(\mathcal{I}) = \emptyset$. But $\mathrm{L}(\mathcal{I})$ is nonempty only if $\mathcal{I}$ can be obtained as the result of assigning and propagating, and therefore if it is a normal form w.r.t. $\rightarrowtail$. Therefore all reducible interpretations in the sense of $\rightarrowtail$ are irreducible in the sense of McKay. Even if we were able to compute these interpretations, it is not realistic to run SEMK on all of them.

5 A Restricted Enumeration

It is however not necessary to generate *all* isomorphism classes of partial interpretations, and we may restrict the search to those interpretations which are logically meaningful, i.e., the interpretations that can be obtained by sequences of assignments and propagations from $\mathcal{I}_0$.

More precisely, we define inductively the set $\mathfrak{P}$ of partial interpretations *reachable* from $\mathcal{I}_0$ as the smallest set containing $\mathcal{I}_0\!\downarrow$ and such that:

$$\mathcal{I} \in \mathfrak{P} \;\Rightarrow\; \forall c \in \mathfrak{C},\ \forall v \in \mathcal{I}[c],\ \mathrm{AP}_\varphi(\mathcal{I}, c, v) \in \mathfrak{P}.$$

It is easy to prove, by induction, that $\mathfrak{P}$ is stable under the action of $\mathfrak{G}$, hence that for all $\mathcal{I} \in \mathfrak{P}$ and $\sigma \in \mathfrak{G}$, we have $\mathcal{I}^\sigma \in \mathfrak{P}$. Hence it should be possible, by restricting SEMK, to obtain an isomorph-free enumeration of $\mathfrak{P}$.

We first restrict upper objects to those in the sets $\mathrm{U}(\mathcal{I})$ for all $\mathcal{I} \in \mathfrak{P}$, and call them *reachable* upper objects. For lower objects we take, for all $\mathcal{J} \in \mathfrak{P}$,

$$\mathrm{L}'(\mathcal{J}) \;=\; \{\ \langle \mathcal{J}, \mathcal{S}, c\rangle \in \mathrm{L}(\mathcal{J}) \ \mid\ \mathcal{J} \cup \mathcal{S} \in \mathfrak{P}\ \}.$$

The *reachable* lower objects are those in the sets $\mathrm{L}'(\mathcal{J})$ for all $\mathcal{J} \in \mathfrak{P}$. It is easy to show that for all $\sigma \in \mathfrak{G}$, we still have $\mathrm{L}'(\mathcal{J}^\sigma) = \mathrm{L}'(\mathcal{J})^\sigma$.

Finally, we note $\mathfrak{R}'$ the restriction of $\mathfrak{R}$ to reachable lower and upper objects. The only property of $\mathfrak{R}$ that is not trivially preserved by this restriction is the required property that every reachable lower object $\langle \mathcal{J}, \mathcal{S}, c\rangle$ should be related to a reachable upper object $\langle \mathcal{I}, c', v\rangle$. But we know that $\mathcal{J} \cup \mathcal{S}$ is in $\mathfrak{P}$, so that $\langle \mathcal{J} \cup \mathcal{S}, c, v\rangle$, where $\{v\} = \mathcal{J}[c]$, is a reachable lower object, related to $\langle \mathcal{J}, \mathcal{S}, c\rangle$. This of course explains our definition of $\mathrm{L}'(\mathcal{J})$.

The only irreducible object, i.e., element $\mathcal{J}$ of $\mathfrak{P}$ such that $\mathrm{L}'(\mathcal{J}) = \emptyset$, is clearly $\mathcal{I}_0\!\downarrow$. The enumeration $\mathrm{SEMK}_\varphi(\mathcal{I}_0\!\downarrow)$ is therefore complete as $\mathrm{SEM}_\varphi(\mathcal{I}_0\!\downarrow)$ is. This completeness however does not come for free: we have to replace the choice function l relative to L by a choice function l' relative to L′.

In the first version of SEMK a lower object $l(\mathcal{J}) = \langle \mathcal{J}, \mathcal{S}, c\rangle$ is easy to compute from a reducible $\mathcal{J}$, by choosing a cell c such that $\mathcal{J}[c]$ is a singleton, and taking $\mathcal{S} = \mathcal{I}_0[c] \setminus \mathcal{J}[c]$. The problem for computing $l'(\mathcal{J}) = \langle \mathcal{J}, \mathcal{S}, c\rangle$, for $\mathcal{J} \in \mathfrak{P} \setminus \{\mathcal{I}_0\}$, is that we must now ensure that $\mathcal{J} \cup \mathcal{S} \in \mathfrak{P}$.

Membership in $\mathfrak{P}$ can be characterized by:

$$\mathcal{I} \in \mathfrak{P} \;\Leftrightarrow\; \mathcal{I} = (\mathcal{I}_0 \ll \mathrm{fp}(\mathcal{I}))\!\downarrow .$$

```
$l'(\mathcal{J})$ = let $j = \mathrm{fp}(\mathcal{J})$ and $I := \mathcal{I}_0\!\downarrow$ in
        let $C = \mathrm{dom}(j) \setminus \mathrm{dom}(\mathrm{fp}(I))$ in
        while $C \neq \emptyset$ do let $c \in C$ in
          $I' := I$;
          $I := \mathrm{AP}_\varphi(I, c, j(c))$;
          $C := C \setminus \mathrm{dom}(\mathrm{fp}(I))$
        done;
        $\langle \mathcal{J}, I' \setminus \mathcal{J}, c \rangle$
```

Fig. 3. A choice function

Unfortunately, there is no guarantee that this property holds for $\mathcal{I} = \mathcal{J} \cup \mathcal{I}_0[c]$, and we may have $\mathcal{J} \subseteq (\mathcal{I}_0 \ll \mathrm{fp}(\mathcal{I}))\!\downarrow\subsetneq \mathcal{I}$. Also note that this membership test is not computationally harmless. The algorithm for computing l' that we propose in Figure 3 requires approximately the same amount of time.

This algorithm clearly terminates since the number of cells in C decreases after each iteration (since $c \notin \mathrm{dom}(\mathrm{fp}(\mathcal{I}))$). Each value of $\mathcal{I}$ is obviously in $\mathfrak{P}$, and contains $\mathcal{J}$. This last fact is rather difficult to prove, and is true only because $\mathcal{J}$ is not a counter-model. The key point is the following result:

Lemma 5.1 *If $\mathcal{I} \rightarrowtail \mathcal{J}$ and $v \in \mathcal{J}[c]$ then $\mathcal{I} \ll [c, v] \downarrow \mathcal{J} \ll [c, v]$.*

Proof. Let C be the clause such that $\mathcal{J} = \mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}$, and l be the corresponding unit literal, and let $\mathcal{I}' = \mathcal{I} \ll [c, v]$ and $\mathcal{J}' = \mathcal{J} \ll [c, v]$. We have $v \in \mathcal{J}[c] \subseteq \mathcal{I}[c]$, hence if $\mathcal{I}[c]$ is a singleton we have $\mathcal{I} = \mathcal{I}'$ and $\mathcal{J} = \mathcal{J}'$, hence obviously $\mathcal{I}' \rightarrowtail \mathcal{J}'$. We now suppose that $\mathcal{I}[c]$ is not a singleton, and consider the rule that yields $\mathcal{R}_{C,\mathcal{I}}$ in Definition 2.7.

- If it is rule (i) or (ii), then $\mathcal{R}_{C,\mathcal{I}} = [c', w]$ where w is a truth value, and we must have $\mathcal{I}[c'] = \{\top, \bot\}$. If $\mathcal{I}'[c']$ is also equal to $\{\top, \bot\}$ then C is still unit in $\mathcal{I}'$, and of course $c \neq c'$. We then have $\mathcal{R}_{C,\mathcal{I}'} = \mathcal{R}_{C,\mathcal{I}}$, hence
$$\mathcal{J}' \;=\; (\mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}) \ll [c, v] \;=\; \mathcal{I} \ll (\mathcal{R}_{C,\mathcal{I}} \cup [c, v]) \;=\; \mathcal{I}' \ll \mathcal{R}_{C,\mathcal{I}'},$$
so that $\mathcal{I}' \rightarrowtail \mathcal{J}'$. If $\mathcal{I}'[c'] \neq \mathcal{I}[c']$, we must have $c = c'$ and $v = w$, hence $\mathcal{I}' = \mathcal{J} = \mathcal{J}'$.
- If it is rule (iii), then l is an equation $t_1 = t_2$, and $\mathcal{R}_{C,\mathcal{I}} = [c_1, E] \cup [c_2, E]$ with $E = \mathcal{I}[c_1] \cap \mathcal{I}[c_2]$. Since $\mathcal{I} \neq \mathcal{J}$ we have $E \neq \emptyset$ and $c_1 \neq c_2$.

  If C is not unit in $\mathcal{I}'$, then we have, say, $c = c_1$, thus $\mathcal{J}[c] = E$, hence $v \in E$. But $c \neq c_2$, hence $\mathcal{I}'[c_2] = \mathcal{I}[c_2]$ contains v, hence $\mathcal{I}'[c] \cap \mathcal{I}'[c_2] = \{v\}$. The clause C must have a definite truth value in $\mathcal{I}'$; it must therefore be true, and $\mathcal{I}'[c_2] = \{v\}$. We therefore have $\mathcal{I}[c_2] = \{v\}$, and then $\mathcal{J}[c_2] = \mathcal{J}[c] = \{v\}$, and obviously $\mathcal{I}' = \mathcal{J} = \mathcal{J}'$.

  If C is unit in $\mathcal{I}'$, then $\mathcal{R}_{C,\mathcal{I}'} = [c_1, E'] \cup [c_2, E']$ with $E' = \mathcal{I}'[c_1] \cap \mathcal{I}'[c_2]$.

If $c \neq c_1$ and $c \neq c_2$ then $E = E'$ and $\mathcal{R}_{C,\mathcal{I}'} = \mathcal{R}_{C,\mathcal{I}}$, hence as above $\mathcal{J}' = \mathcal{I}' \ll \mathcal{R}_{C,\mathcal{I}'}$, so that $\mathcal{I}' \rightarrowtail \mathcal{J}'$.

But if, say, $c = c_1$, then of course $c \neq c_2$, so that $\mathcal{I}'[c_2] = \mathcal{I}[c_2]$. Since $v \in E$ we have $v \in \mathcal{I}[c_2]$, and since $\mathcal{I}'[c] = \{v\}$ we get $E' = \{v\}$. Let $\mathcal{K} = \mathcal{I}' \ll \mathcal{R}_{C,\mathcal{I}'}$, we have of course $\mathcal{I}' \rightarrowtail^\star \mathcal{K}$, and $\mathcal{K} = \mathcal{I}' \ll [c_2, v]$. We have $\mathcal{J}' = \mathcal{I}' \ll [c_2, E]$, hence if E is not reduced to the singleton $\{v\}$, i.e., if $\mathcal{J}' \neq \mathcal{K}$, then C is unit in $\mathcal{J}'$, with $\mathcal{R}_{C,\mathcal{J}'} = [c, v] \cup [c_2, v]$. Then obviously $\mathcal{K} = \mathcal{J}' \ll \mathcal{R}_{C,\mathcal{J}'}$, and we get $\mathcal{J}' \rightarrowtail \mathcal{K}$.

- If it is, say, rule (iv), then l is a disequation $t_1 \neq t_2$, $\mathcal{I}[c_2]$ is a singleton $\{v'\}$ (hence $c \neq c_2$), and we have $\mathcal{R}_{C,\mathcal{I}} = [c_1, \mathcal{I}[c_1] \setminus \{v'\}]$. We also have $\mathcal{I}'[c_2] = \{v'\}$, hence if $\mathcal{I}'[c_1] = \mathcal{I}[c_1]$ (hence $c \neq c_1$) then C is still unit in $\mathcal{I}'$, with $\mathcal{R}_{C,\mathcal{I}'} = \mathcal{R}_{C,\mathcal{I}}$, and we get $\mathcal{J}' = \mathcal{I}' \ll \mathcal{R}_{C,\mathcal{I}'}$ as above, so that $\mathcal{I}' \rightarrowtail \mathcal{J}'$.

  If $\mathcal{I}'[c_1] \neq \mathcal{I}[c_1]$ then $c = c_1$, and since $\mathcal{I}$ and $\mathcal{J}$ differ only on c_1, we obviously have $\mathcal{J}' = \mathcal{I}'$.

□

It is then easy to prove:

Lemma 5.2 *If $v \in \mathcal{I}\!\downarrow [c]$ then $(\mathcal{I}\!\downarrow \ll [c, v])\!\downarrow = (\mathcal{I} \ll [c, v])\!\downarrow$.*

Proof. It is easy to prove using Lemma 5.1 that if $\mathcal{I} \rightarrowtail^\star \mathcal{J}$ and $v \in \mathcal{J}[c]$ then $(\mathcal{I} \ll [c, v])\!\downarrow = (\mathcal{J} \ll [c, v])\!\downarrow$, by induction on the length of the derivation $\mathcal{I} \rightarrowtail^\star \mathcal{J}$. We then take $\mathcal{J} = \mathcal{I}\!\downarrow$. □

We may then prove that the order in which assignments are made is not relevant.

Theorem 5.3 *Let $J = \mathrm{AP}_\varphi(\mathrm{AP}_\varphi(\mathcal{I}, c, v), c', v')$, if $\top \in \widehat{J}$ then $J = \mathrm{AP}_\varphi(\mathrm{AP}_\varphi(\mathcal{I}, c', v'), c, v)$.*

Proof. Since $\top \in \widehat{J}$ we must have $v' \in (\mathcal{I} \ll [c, v])\!\downarrow [c']$, hence by Lemma 5.2 we have

$$J = ((\mathcal{I} \ll [c, v])\!\downarrow \ll [c', v'])\!\downarrow = (\mathcal{I} \ll ([c, v] \cup [c', v']))\!\downarrow,$$

hence we must also have $v \in (\mathcal{I} \ll [c', v'])\!\downarrow [c]$, and again by Lemma 5.2 we get

$$J = ((\mathcal{I} \ll [c', v'])\!\downarrow \ll [c, v])\!\downarrow = \mathrm{AP}_\varphi(\mathrm{AP}_\varphi(\mathcal{I}, c', v'), c, v).$$

□

Now, since $\mathcal{J} \in \mathfrak{P}$, it must be obtained from $\mathcal{I}_0$ by some of the assignments of $\mathrm{fp}(\mathcal{J})$, in some order. Propagating the same assignments in any order eventually leads to the same result $\mathcal{J}$. The value $I = \mathcal{J}$ is obtained exactly when $C = \emptyset$, so that the preceding value I' of I has an order strictly smaller than the order of $\mathcal{J}$. Hence the final value $\langle \mathcal{J}, I' \setminus \mathcal{J}, c \rangle$ is a reachable lower object.

Note that I' depends on the unspecified order in which the cells $c \in C$ are taken in this algorithm. Any order yields a correct result, but since l' must be a function, the choice must be deterministic. The reason is that the algorithm for l' will be called several times on the same canonical element $\mathcal{J}$, and must always return the same result.

Example 5.4 We may illustrate this point on Example 2.4. Starting from $\mathrm{SEMK}_\varphi(I_0)$, we have three orbits in $\mathrm{U}(I_0)$, whose representatives may be chosen to be $\langle I_0, a, 1\rangle$, $\langle I_0, f[1], 1\rangle$ and $\langle I_0, f[1], 2\rangle$, in this order. We obtain a first $J_1 = \mathrm{AP}_\varphi(I_0, a, 1) = [a, 1] \cup [f[1], 1] \cup [f[2], \{1, 2\}]$. We keep J_1 (i.e., there is a recursive call $\mathrm{SEMK}_\varphi(J_1)$) only if we have $\langle J_1, [a, 2] \cup [f[1], 2], a\rangle \in m(J_1)$. Supposing J_1 is canonical, hence that $\gamma(J_1)$ is the identity, we then compute $l'(J_1)$. The initial value of C is $\{a, f[1]\}$, and we clearly keep J_1 if the cell a is chosen first in the **while** loop. Suppose this is the case.

The next iteration after J_1 yields $J_2 = \mathrm{AP}_\varphi(I_0, f[1], 1) = J_1 \cup [a, 2]$. Suppose we keep J_2, then one iteration in $\mathrm{SEMK}_\varphi(J_2)$ yields $\mathrm{AP}_\varphi(J_2, a, 1) = J_1$. We then keep this J_1 only if $\langle J_1, [a, 2], a\rangle \in m(J_1)$, which is false. The search $\mathrm{SEMK}_\varphi(J_1)$ is performed only once.

The implementation of these algorithms is not finished yet. It is of course not certain that the time saved by isomorph-pruning is not entirely lost in the technicalities of SEMK, and only experimentations will assess the interest of this technique. A few discrepancies between SEM, or more generally the problem of searching for finite models of a first order formula, and McKay's algorithm, should be emphasized.

McKay's method is very successful at enumerating structures with rather simple inductive definitions, hence simple choice functions. In SEMK the choice function l' is not simple, and requires increasing computational power deeper in the search tree. This departs from what we may call the intension behind McKay's method. In some sense it may be due to the fact that we cannot consider the enumerated structures (finite models of a formula) as fundamentally inductive in nature.

Another difference is that SEMK may not be efficient for finding a first model, since potentially successful branches may be pruned simply because they are not canonical. Thus SEMK may not have the same range of applications as SEM; it should rather be seen as a way of exhaustively enumerating abstract structures defined in an abstract way, by a formula.

References

[1] Gilles Audemard and Laurent Henocque. The extended least number heuristic. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *First IJCAR*, LNAI 2083, pages 427–442.

Springer-Verlag, 2001.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[3] Thierry Boy de la Tour. On the complexity of finite sorted algebras. In Ricardo Caferra and Gernot Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, LNAI 1761, pages 95–108. Springer Verlag, 2000.

[4] Thierry Boy de la Tour. A note on symmetry heuristics in SEM. In Andrei Voronkov, editor, *Proceedings of CADE-18*, LNAI 2392, pages 181–194. Springer Verlag, 2002.

[5] Brendan D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

[6] Brendan D. McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 26(2):306–324, February 1998.

[7] Nicolas Peltier. A new method for automated finite model building exploiting failures and symmetries. *Journal of Logic and Computation*, 8(4):511–543, 1998.

[8] Uwe Schöning and Randall Pruim. *Gems of Theoretical Computer Science.* Springer-Verlag, 1998.

[9] J. Zhang and H. Zhang. SEM: a system for enumerating models. In Chris S. Mellish, editor, *Proceedings of the 14th IJCAI*, pages 298–303. Morgan Kaufmann, 1995.

Appendix: proof of Lemma 2.9

If $\mathcal{I} \rightarrowtail \mathcal{J}$, $\mathcal{I} \rightarrowtail \mathcal{J}'$, *and* $\top \in \widehat{\mathcal{J}}$ *or* $\top \in \widehat{\mathcal{J}'}$ *then* $\mathcal{J} \downarrow \mathcal{J}'$.

Proof. Let C and C' be the clauses of φ, unit in $\mathcal{I}$ (with respective unit literal l and l'), such that $\mathcal{J} = \mathcal{I} \ll \mathcal{R}_{C,\mathcal{I}}$ and $\mathcal{J}' = \mathcal{I} \ll \mathcal{R}_{C',\mathcal{I}}$, and let $\mathcal{K} = \mathcal{J} \ll \mathcal{R}_{C',\mathcal{J}}$ and $\mathcal{K}' = \mathcal{J}' \ll \mathcal{R}_{C,\mathcal{J}'}$. There may be one or two cells involved in each of $\mathcal{R}_{C,\mathcal{I}}$, $\mathcal{R}_{C',\mathcal{I}}$. If these cells are all different, then it is easy to see that $\mathcal{R}_{C,\mathcal{I}} = \mathcal{R}_{C,\mathcal{J}'}$ and $\mathcal{R}_{C',\mathcal{I}} = \mathcal{R}_{C',\mathcal{J}}$, and therefore that $\mathcal{K} = \mathcal{K}'$. This proves that $\mathcal{J} \rightarrowtail^\star \mathcal{K}$ and $\mathcal{J}' \rightarrowtail^\star \mathcal{K}$, as required. Suppose now that these cells are not all different.

If among these there is a P-cell for some predicate symbol P, say that $l = P(t_1, \ldots, t_n)$, then l' involves the same cell $c = \mathrm{cell}_{\mathcal{I}}(P(t_1, \ldots, t_n))$. If l' is positive, then $\mathcal{R}_{C,\mathcal{I}} = \mathcal{R}_{C',\mathcal{I}}$, hence $\mathcal{J} = \mathcal{J}'$. If l' is negative, then $[\![C']\!]_{\mathcal{J}} = [\![C]\!]_{\mathcal{J}'} = \{\bot\}$, which is impossible since $\top \in \widehat{\mathcal{J}}$ or $\top \in \widehat{\mathcal{J}'}$.

If none of these cells correspond to a predicate symbol, then l and l' are equations or disequations, and there remains to consider the rules (iii) to (v) of Definition 2.7. Let c, d (resp. c, d') be the cells involved in l (resp. l'). By hypothesis we have $c \neq d$ (since $\mathcal{I} \neq \mathcal{J}$) and $c \neq d'$, but we may have $d = d'$. Let $\mathcal{L} = \mathcal{K} \ll \mathcal{R}_{C,\mathcal{K}}$ and $\mathcal{L}' = \mathcal{K}' \ll \mathcal{R}_{C',\mathcal{K}'}$, we clearly have $\mathcal{J} \rightarrowtail^\star \mathcal{K} \rightarrowtail^\star \mathcal{L}$ and $\mathcal{J}' \rightarrowtail^\star \mathcal{K}' \rightarrowtail^\star \mathcal{L}'$. We will join $\mathcal{J}$ and $\mathcal{J}'$ at least on $\mathcal{L} = \mathcal{L}'$, or eventually on $\mathcal{K} = \mathcal{K}'$, as illustrated below.

$$\begin{tikzcd}
 & \mathcal{J} \arrow[r, "C'"'] & \mathcal{K} \arrow[r, "C"] & \mathcal{L} \\
\mathcal{I} \arrow[ur, "C"] \arrow[dr, "C'"'] & & & \\
 & \mathcal{J}' \arrow[r, "C"] & \mathcal{K}' \arrow[r, "C'"'] & \mathcal{L}'
\end{tikzcd}$$

We now investigate all possible cases for l and l'.

- If l and l' are both equations, then rule (iii) applies in the reductions $\mathcal{I} \rightarrowtail \mathcal{J}$ and $\mathcal{I} \rightarrowtail \mathcal{J}'$, and also in the (eventual) reductions $\mathcal{J} \rightarrowtail^\star \mathcal{K} \rightarrowtail^\star \mathcal{L}$ and $\mathcal{J}' \rightarrowtail^\star \mathcal{K}' \rightarrowtail^\star \mathcal{L}'$. By assuming that C is unit in $\mathcal{J}'$ and $\mathcal{K}$, and that C' is unit in $\mathcal{J}$ and $\mathcal{K}'$, we easily compute the values of $\mathcal{J}, \mathcal{J}', \mathcal{K}, \mathcal{K}', \mathcal{L}$ and $\mathcal{L}'$ on the cells c, d and d' by applying rule (iii). We let $A = \mathcal{I}[d] \cap \mathcal{I}[c] \cap \mathcal{I}[d']$, and we have

	d	c	d'
$\mathcal{J}$	$\mathcal{I}[d] \cap \mathcal{I}[c]$	$\mathcal{I}[d] \cap \mathcal{I}[c]$	$\mathcal{I}[d']$
$\mathcal{J}'$	$\mathcal{I}[d]$	$\mathcal{I}[c] \cap \mathcal{I}[d']$	$\mathcal{I}[c] \cap \mathcal{I}[d']$
$\mathcal{K}$	$\mathcal{J}[d] = \mathcal{I}[c] \cap \mathcal{I}[d]$	$\mathcal{J}[c] \cap \mathcal{J}[d'] = A$	A
$\mathcal{K}'$	$\mathcal{J}'[d] \cap \mathcal{J}'[c] = A$	A	$\mathcal{J}'[d'] = \mathcal{I}[c] \cap \mathcal{I}[d']$
$\mathcal{L}$	$\mathcal{K}[d] \cap \mathcal{K}[c] = A$	A	$\mathcal{K}[d'] = A$
$\mathcal{L}'$	$\mathcal{K}'[d] = A$	$\mathcal{K}'[c] \cap \mathcal{K}'[d'] = A$	A

which proves that $\mathcal{L} = \mathcal{L}'$. We now have to consider the case where one of C, C' is not unit in a corresponding interpretation. Since C and C' are unit in $\mathcal{I}$, and we consider refinements of $\mathcal{I}$, these clauses may no longer be unit only by becoming true or false.

If they become true, say $[\![C]\!]_{\mathcal{J}'} = \{\top\}$, obviously because $[\![l]\!]_{\mathcal{J}'} = \{\top\}$, then $\mathcal{J}'[d]$ and $\mathcal{J}'[c]$ must be the same singleton $\{v\}$, and then we have $\mathcal{K}' = \mathcal{J}'$ (because $\mathcal{R}_{C,\mathcal{J}'} = \emptyset$), so that $\mathcal{K}'[d] = \mathcal{K}'[c] = \{v\} = \mathcal{J}'[d] \cap \mathcal{J}'[c]$. This means that if clause C or C' becomes true, the formulas above are still valid, and still prove $\mathcal{L} = \mathcal{L}'$.

We now prove that C and C' cannot become false. First, since $\mathcal{K}[c] \subseteq \mathcal{K}[d]$, we have $\mathcal{K}[c] \cap \mathcal{K}[d] \neq \emptyset$, so that $[\![l]\!]_{\mathcal{K}} \neq \{\perp\}$. Hence C is not false in $\mathcal{K}$, and symmetrically C' is not false in $\mathcal{K}'$. Next, we have $\mathcal{J}'[d] \cap \mathcal{J}'[c] = \mathcal{J}[c] \cap \mathcal{J}[d'] = A$, hence if either $[\![C]\!]_{\mathcal{J}'}$ or $[\![C']\!]_{\mathcal{J}}$ equals $\{\perp\}$, then they are both equal to $\{\perp\}$, so that $\widehat{\mathcal{J}} = \widehat{\mathcal{J}'} = \{\perp\}$, which is impossible by hypothesis.

- If l is an equation, and l' a disequation, then as above rule (iii) applies to $\mathcal{I} \rightarrowtail \mathcal{J}$, and either rule (iv) or (v) applies to $\mathcal{I} \rightarrowtail \mathcal{J}'$, so we have either $|\mathcal{I}[c]| = 1$ or $|\mathcal{I}[d']| = 1$. We first suppose that $\mathcal{I}[c] = \{v\}$, then by rule (iv) or (v) we have

	c	d	d'
$\mathcal{J}$	$\mathcal{I}[c] \cap \mathcal{I}[d] = \{v\}$	$\mathcal{I}[c] \cap \mathcal{I}[d] = \{v\}$	$\mathcal{I}[d']$
$\mathcal{J}'$	$\mathcal{I}[c] = \{v\}$	$\mathcal{I}[d]$	$\mathcal{I}[d'] \setminus \{v\}$
$\mathcal{K}$	$\mathcal{J}[c] = \{v\}$	$\mathcal{J}[d] = \{v\}$	$\mathcal{J}[d'] \setminus \{v\} = \mathcal{I}[d'] \setminus \{v\}$
$\mathcal{K}'$	$\mathcal{J}'[c] \cap \mathcal{J}'[d] = \{v\}$	$\{v\}$	$\mathcal{J}'[d'] = \mathcal{I}[d'] \setminus \{v\}$

so that $\mathcal{K} = \mathcal{K}'$. Note that $v \in \mathcal{I}[d]$, since otherwise $\mathcal{R}_{C,\mathcal{I}}$ would be empty, hence $\mathcal{J} = \mathcal{I}$, which is impossible since $\mathcal{I} \rightarrowtail \mathcal{J}$. We also know that C is unit in $\mathcal{J}'$ and C' is unit in $\mathcal{J}$, as they are unit in $\mathcal{I}$, since $\mathcal{J}[c] = \mathcal{J}'[c] = \mathcal{I}[c]$, $\mathcal{J}[d'] = \mathcal{I}[d']$ and $\mathcal{J}'[d] = \mathcal{I}[d]$.

We next suppose that $\mathcal{I}[d'] = \{v\}$. Since C' is unit in $\mathcal{I}$, we must have $v \in \mathcal{I}[c]$, which cannot be a singleton. Let $A = \mathcal{I}[c] \cap \mathcal{I}[d]$, since C is unit in $\mathcal{I}$ we also know that $A \neq \emptyset$. We have by the same rules as above

$$\mathcal{J}[c] = \mathcal{J}[d] = A, \qquad \mathcal{J}[d'] = \{v\},$$
$$\mathcal{J}'[c] = \mathcal{I}[c] \setminus \{v\},\ \mathcal{J}'[d] = \mathcal{I}[d],\ \mathcal{J}'[d'] = \{v\}.$$

The clauses C and C' may not be unit in $\mathcal{J}'$ and $\mathcal{J}$ (respectively). In particular we have

$$[\![l]\!]_{\mathcal{J}'} = \{\bot\} \;\Leftrightarrow\; \mathcal{J}'[c] \cap \mathcal{J}'[d] = \emptyset \;\Leftrightarrow\; A \setminus \{v\} = \emptyset \;\Leftrightarrow\; A = \{v\},$$
$$[\![l']\!]_{\mathcal{J}} = \{\bot\} \;\Leftrightarrow\; \mathcal{J}[c] = \{v\} \;\Leftrightarrow\; A = \{v\},$$

hence l or l' are false in $\mathcal{J}$ or $\mathcal{J}'$ only if $\widehat{\mathcal{J}} = \widehat{\mathcal{J}'} = \{\bot\}$, which is impossible (thus $A \setminus \{v\} \neq \emptyset$). We consider now the case where l is true in $\mathcal{J}'$:

$$\begin{aligned}
[\![l]\!]_{\mathcal{J}'} = \{\top\} &\Rightarrow \mathcal{J}'[c] = \mathcal{J}'[d] = \{v'\} \ \text{ where } v' \neq v\\
&\Rightarrow \mathcal{I}[c] = \{v, v'\} \ \text{ and } \ \mathcal{I}[d] = \{v'\}\\
&\Rightarrow \mathcal{J}[c] = \mathcal{J}[d] = \{v'\}\\
&\Rightarrow \mathcal{J} = \mathcal{J}',
\end{aligned}$$

and we are done. We may now assume that C is unit in $\mathcal{J}'$, and thus apply rule (iii), yielding

$$\mathcal{K}'[c] = \mathcal{K}'[d] = \mathcal{J}'[c] \cap \mathcal{J}'[d] = A \setminus \{v\},\ \mathcal{K}'[d'] = \{v\}.$$

We now consider the case where l' is true in $\mathcal{J}$:

$$[\![l']\!]_{\mathcal{J}} = \{\top\} \;\Rightarrow\; v \notin \mathcal{J}[c] = A \;\Rightarrow\; A \setminus \{v\} = A \;\Rightarrow\; \mathcal{K}' = \mathcal{J},$$

and we are done. We assume now that $v \in A$, hence that C' is unit in $\mathcal{J}$, and rule (iv) or (v) applies:

$$\mathcal{K}[c] = \mathcal{J}[c] \setminus \{v\} = A \setminus \{v\},\ \mathcal{K}[d] = \mathcal{J}[d] = A,\ \mathcal{K}[d'] = \{v\}.$$

We then see that C is still unit in $\mathcal{K}$, since $\mathcal{K}[c] \cap \mathcal{K}[d] = A \setminus \{v\} \neq \emptyset$ and $\mathcal{K}[d] = A$ is not a singleton. By rule (iii) we have

$$\mathcal{L}[c] = \mathcal{L}[d] = \mathcal{K}[c] \cap \mathcal{K}[d] = A \setminus \{v\},\ \mathcal{L}[d'] = \{v\},$$

and we get $\mathcal{L} = \mathcal{K}'$.

- The last case to consider is when l and l' are both disequations. Then rules (iv) or (v) apply to both $\mathcal{I} \rightarrowtail \mathcal{J}$ and $\mathcal{I} \rightarrowtail \mathcal{J}'$, and we have either $|\mathcal{I}[c]| = 1$ or $|\mathcal{I}[d]| = |\mathcal{I}[d']| = 1$. We first suppose that $\mathcal{I}[c] = \{v\}$, thus

$$\mathcal{J}[c] = \{v\},\ \mathcal{J}[d] = \mathcal{I}[d] \setminus \{v\},\ \mathcal{J}[d'] = \mathcal{I}[d'].$$

Note that C' is unit in $\mathcal{J}$ as in $\mathcal{I}$ since $\mathcal{J}[c] = \mathcal{I}[c]$ and $\mathcal{J}[d'] = \mathcal{I}[d']$, so that

$$\mathcal{K}[c] = \{v\},\ \mathcal{K}[d] = \mathcal{J}[d] = \mathcal{I}[d] \setminus \{v\},\ \mathcal{K}[d'] = \mathcal{J}[d'] \setminus \{v\} = \mathcal{I}[d'] \setminus \{v\}$$

Symmetrically we get $\mathcal{J}'[d'] = \mathcal{I}[d'] \setminus \{v\}$ and $\mathcal{J}'[d] = \mathcal{I}[d]$, and $\mathcal{K}' = \mathcal{K}$, so we are done.

We next suppose that $\mathcal{I}[d] = \{v\}$ and $\mathcal{I}[d'] = \{v'\}$, so that $\mathcal{J}$, $\mathcal{J}'$, $\mathcal{K}$ and $\mathcal{K}'$ may differ only on c. We have $\mathcal{J}[c] = \mathcal{I}[c] \setminus \{v\}$ and $\mathcal{J}'[c] = \mathcal{I}[c] \setminus \{v'\}$, and since C and C' are unit in $\mathcal{I}$ we must have $v, v' \in \mathcal{I}[c]$.

Note that we may not have $\mathcal{I}[c] = \{v, v'\}$, because we would then have $\mathcal{J}[c] = \{v'\}$ and $\mathcal{J}'[c] = \{v\}$, thus $[\![C']\!]_{\mathcal{J}} = [\![C]\!]_{\mathcal{J}'} = \{\perp\}$, which is impossible. If $v = v'$ then $\mathcal{J} = \mathcal{J}'$ and we are done, so we assume that $v \neq v'$, so that neither $\mathcal{J}[c]$ nor $\mathcal{J}'[c]$ can be a singleton. This means that C' must be unit in $\mathcal{J}$, and C unit in $\mathcal{J}'$. We therefore have $\mathcal{K}[c] = \mathcal{I}[c] \setminus \{v, v'\} = \mathcal{K}'[c]$, thus $\mathcal{K} = \mathcal{K}'$.

$\square$