On solving nominal fixpoint equations

Mauricio Ayala-Rincón[1], Washington de Carvalho-Segundo[1], Maribel Fernández[2], and Daniele Nantes-Sobrinho[1]

[1] Depts. de Matemática e Ciência da Computação Universidade de Brasília, Brazil
[2] Department of Informatics, King's College London, UK

Abstract. In nominal syntax, variable binding is specified using atom-abstraction constructors, and alpha-equivalence is formalised using freshness constraints and atom swappings, which implement variable renamings. Composition of swappings gives rise to atom permutations. Algorithms to check equivalence, match and unify nominal terms have been extended to deal with terms where some operators are associative and/or commutative. In the case of nominal C-unification, problems are transformed into finite and complete families of fixpoint equations of the form $\pi.X \approx_? X$, where π is a permutation. To generate nominal C-unifiers, a technique to obtain a sound and complete set of solutions for these equations is needed. In this work we show how complete sets of solutions for nominal fixpoint problems are built and discuss efficient techniques to generate solutions based on algebraic properties of permutations.

1 Introduction

Nominal syntax is an extension of first order syntax, where terms are built using function symbols, abstractions, and two kinds of variables: atoms, which can be abstracted, and unknowns (or simply variables), which behave like first-order variables, except for the fact that they can have "suspended atom permutations", which act when the variable is instantiated by a term. Atom abstractions induce an α-equivalence relation on nominal terms, which is axiomatised using a freshness relation between atoms and terms. Nominal unification [15] is unification of nominal terms, and takes into account the α-equivalence relation. Extensions of nominal unification include equivariant unification [8, 1] and nominal narrowing [5], which are useful tools in equational reasoning and confluence analysis of nominal rewriting systems [2, 9].

In many application domains, function symbols have equational properties, such as associativity and commutativity, which must be taken into account during the unification process. In previous work [3], we studied α-AC-equivalence of nominal terms, and nominal C-unification [4], that is, nominal unification in languages with commutative operators. It is well-known that C-unification is an NP-complete problem (see Chapter 10 in [7]).

To solve nominal C-unification problems, we provided in [4] a set of simplification rules that generates, for each solvable C-unification problem, a finite set of

fixpoint problems that are finite sets of *fixpoint equations* together with a freshness context and a substitution. Fixpoint equations have the form $\pi.X \approx_? X$, where X is a variable and π is a permutation.

Fixpoint problems are also generated in standard nominal unification algorithms, but the presence of commutative operators in the signature complicates their treatment. In nominal unification algorithms, fixpoint equations are solved simply by requiring the support of the permutation to be fresh for the variable, but that is not the only way to solve them if there are commutative operators. In [4], the correctness and completeness of the rule-based algortihm to transform a nominal C-unification problem into a finite set of fixpoint problems was formalised in Coq, and a sound method to generate solutions for fixpoint problems was given, showing that infinite independent solutions are possible for a single fixpoint equation. Thus, nominal C-unification is infinitary.

Sound and complete procedures to solve fixpoint problems modulo commutativity are needed not only within nominal-C-unification algorithms, but also in other application areas where formalisations of syntax with binders involve commutativity axioms, such as the π-calculus. For example, fixpoint problems modulo commutativity are generated when solving unification problems in extensions of the λ-calculus with recursive definitions [10].

Contribution. The main result is a sound and complete procedure to solve fixpoint problems. More specifically:

- We prove the completeness of the procedure to generate solutions for fixpoint problems described in [4]. The analysis is based on the feasibility of combinations of the atoms in the domain of permutations used in the fixpoint equations in a fixpoint problem. Solutions for these problems are built considering the combinatorial properties of atom permutations and by combining atoms using the basic elements of the nominal syntax, that is, pairs, abstractions and variables, as well as the function symbols in the signature. The variables included in these feasible combinations are new, and the atoms in the support of the permutations should be fresh for these variables. The greedy generation of complete sets of solutions for a fixpoint equation is based on the construction of the so called *extended pseudo-cycles* from permutation cycles in the algebraic representation of permutations as products of *permutation cycles*. Only permutation cycles of length (period) a power of two are considered since permutation cycles of other lengths do not generate feasible (commutative) combinations.
- Furthermore, we work out an interesting improvement that avoids the generation of feasible solutions for different fixpoint equations on the same variable. It is based on the fact that the feasible combinations for permutation cycles of the same length (a power of two) with the same domain, that are not algebraic *factors* of each other would not give rise to feasible common solutions.

Organisation. Section 2 introduces the background about nominal syntax and nominal C-unification. Section 3 proves the soundness and completeness of combinatorial solutions for fixpoint equations. Section 4 presents the improvements of the generator of solutions. Section 5 concludes the paper with future work.

2 Nominal syntax and nominal (α-)C-unification

2.1 Nominal Syntax

Consider countable disjoint sets of variables $\mathcal{X} := \{X, Y, Z, \cdots\}$ and atoms $\mathcal{A} := \{a, b, c, \cdots\}$. A *permutation* π is a bijection on $\mathcal{A}$ with a finite *domain*, where the domain (i.e., the *support*) of π is the set $dom(\pi) := \{a \in \mathcal{A} \mid \pi \cdot a \neq a\}$.

We will assume as in [3] countable sets of function symbols with different equational properties such as associativity, commutativity, idempotence, etc. Function symbols have superscripts that indicate their equational properties; thus, f_k^C will denote the k^{th} function symbol that is commutative and $f_j^\emptyset$ the j^{th} function symbol without any equational property.

Definition 1 (Nominal grammar). *Nominal terms are generated by the following grammar.*

$$s, t := \langle\rangle \mid \bar{a} \mid [a]t \mid \langle s, t\rangle \mid f_k^E\, t \mid \pi.X$$

$\langle\rangle$ *denotes the* unit *(that is the empty tuple),* $\bar{a}$ *denotes an* atom term *(that is a* name *that plays the role of an object level variable),* $[a]t$ *denotes an* abstraction *of the atom* a *over the term* t*,* $\langle s, t\rangle$ *denotes a* pair*,* $f_k^E\, t$ *the* application *of* f_k^E *to* t *and,* $\pi.X$ *a* moderated *or* suspended *variable, where* π *is an* atom permutation*.*

Atom permutations are represented by finite lists of *swappings*, which are pairs of different atoms $(a\, b)$; hence, a *permutation* π has the form $(a_1\, b_1) :: \ldots :: (a_n\, b_n) :: nil$, where *nil* denotes the *identity* permutation.

Suspensions of the form $nil.X$ will be represented just by X. The set of variables occurring in a term t will be denoted as $Var(t)$. This notation extends to a set S of terms in the natural way: $Var(S) = \bigcup_{t \in S} Var(t)$.

A *substitution* σ is a mapping from variables to terms such that $X \neq X\sigma$ only for a finite set of variables. This set is called the *domain* of σ and is denoted by $dom(\sigma)$. For $X \in dom(\sigma)$, $X\sigma$ is called the *image* of X by σ. Define the image of σ as $im(\sigma) = \{X\sigma \mid X \in dom(\sigma)\}$. The set of variables occurring in the image of σ is then $Var(im(\sigma))$. A substitution σ with $dom(\sigma) := \{X_0, \cdots, X_n\}$ can be represented as a set of *binds* in the form $\{X_0/t_0, \cdots, X_n/t_n\}$, where for $0 \leq i \leq n$, $X_i\sigma = t_i$. We assume standard definitions for the *action* of permutations and substitutions on nominal terms (see e.g. [11], [15]). Since for our purposes the algebraic combinatorial properties of permutations on atoms are relevant, in this paper atom permutations are seen as *products* of *permutation cycles* (see [12]): for instance, the nominal *swapping* permutation $(a\, b) :: (a\, c) :: (a\, d) :: (e\, f) :: (e\, g) :: nil$ is seen as the product of permutation cycles $(a\, b\, c\, d)\, (e\, f\, g)$.

2.2 The relation $\approx_{\{\alpha,C\}}$ and Nominal $\approx_{\{\alpha,C\}}$-unification

In [3], the relation $\approx_\alpha$ was extended to deal with associative and commutative theories. Here we will consider α-equivalence modulo commutativity, denoted $\approx_{\{\alpha,C\}}$. This means that some function symbols in our syntax are commutative.

The inference rules defining freshness and $\approx_{\{\alpha,C\}}$-equivalence are given in Figures 1 and 2. The *difference set* between two permutations π and π' is the set of atoms where the action of π and π' differs: $ds(\pi,\pi') := \{a \in \mathcal{A} \mid \pi \cdot a \neq \pi' \cdot a\}$.

The symbols ∇ and Δ are used to denote *freshness contexts* that are sets of constraints of the form $a\#X$, meaning that the atom a is fresh in X. The domain of a freshness context $dom(\Delta)$ is the set of atoms appearing in it; $\Delta|_X$ denotes the restriction of Δ to the freshness constraints on X: $\{a\#X \mid a\#X \in \Delta\}$; $dom(\pi)\#X$ and $ds(\pi,\pi')\#X$ denote, respectively, the sets $\{a\#X \mid a \in dom(\pi)\}$ and $\{a\#X \mid a \in ds(\pi,\pi')\}$.

$$\frac{}{\nabla \vdash a \# \langle\rangle}\,(\#\langle\rangle) \qquad \frac{}{\nabla \vdash a \# \overline{b}}\,(\#\,\mathbf{atom}) \qquad \frac{\nabla \vdash a \# t}{\nabla \vdash a \# f_k^E\ t}\,(\#\,\mathbf{app}) \qquad \frac{}{\nabla \vdash a \# [a]t}\,(\#\,\mathbf{a[a]})$$

$$\frac{\nabla \vdash a \# t}{\nabla \vdash a \# [b]t}\,(\#\,\mathbf{a[b]}) \qquad \frac{(\pi^{-1} \cdot a\#X) \in \nabla}{\nabla \vdash a \# \pi.X}\,(\#\,\mathbf{var}) \qquad \frac{\nabla \vdash a \# s \;\; \nabla \vdash a \# t}{\nabla \vdash a \# \langle s,t\rangle}\,(\#\,\mathbf{pair})$$

Fig. 1. Rules for the relation #

$$\frac{}{\nabla \vdash \langle\rangle \approx_\alpha \langle\rangle}\,(\approx_{\{\alpha,\mathbf{C}\}} \langle\rangle) \qquad \frac{}{\nabla \vdash \overline{a} \approx_{\{\alpha,C\}} \overline{a}}\,(\approx_{\{\alpha,\mathbf{C}\}}\,\mathbf{atom})$$

$$\frac{\nabla \vdash s \approx_{\{\alpha,C\}} t}{\nabla \vdash f_k^E\ s \approx_{\{\alpha,C\}} f_k^E\ t}\,,\quad E \neq C \text{ or both } s \text{ and } t \text{ are not pairs}\;\;(\approx_{\{\alpha,\mathbf{C}\}}\,\mathbf{app})$$

$$\frac{\nabla \vdash s_0 \approx_{\{\alpha,C\}} t_i,\;\; \nabla \vdash s_1 \approx_{\{\alpha,C\}} t_{(i+1)\bmod 2}}{\nabla \vdash f_k^C\ \langle s_0,s_1\rangle \approx_{\{\alpha,C\}} f_k^C\ \langle t_0,t_1\rangle}\,,\quad i = 0,1\;\;(\approx_{\{\alpha,\mathbf{C}\}}\,\mathbf{C})$$

$$\frac{\nabla \vdash s \approx_{\{\alpha,C\}} t}{\nabla \vdash [a]s \approx_{\{\alpha,C\}} [a]t}\,(\approx_{\{\alpha,\mathbf{C}\}}\,[\mathbf{aa}]) \qquad \frac{\nabla \vdash s \approx_{\{\alpha,C\}} (a\,b)\cdot t \;\; \nabla \vdash a \# t}{\nabla \vdash [a]s \approx_{\{\alpha,C\}} [b]t}\,(\approx_{\{\alpha,\mathbf{C}\}}\,[\mathbf{ab}])$$

$$\frac{ds(\pi,\pi')\#X \subseteq \nabla}{\nabla \vdash \pi.X \approx_{\{\alpha,C\}} \pi'.X}\,(\approx_{\{\alpha,\mathbf{C}\}}\,\mathbf{var}) \qquad \frac{\nabla \vdash s_0 \approx_{\{\alpha,C\}} t_0 \;\; \nabla \vdash s_1 \approx_{\{\alpha,C\}} t_1}{\nabla \vdash \langle s_0,t_0\rangle \approx_{\{\alpha,C\}} \langle s_1,t_1\rangle}\,(\approx_{\{\alpha,\mathbf{C}\}}\,\mathbf{pair})$$

Fig. 2. Rules for the relation $\approx_{\{\alpha,C\}}$

Key properties of the nominal freshness and α-equivalence relations have been extensively explored in previous works [3, 6, 14, 15]. In [4] we also have formalised analogous properties for $\approx_{\{\alpha,C\}}$. Among them we have *freshness preservation*: If $\nabla \vdash a \# s$ and $\nabla \vdash s \approx_{\{\alpha,C\}} t$, then $\nabla \vdash a \# t$; *equivariance*: for all permutations π, if $\nabla \vdash s \approx_{\{\alpha,C\}} t$ then $\nabla \vdash \pi \cdot s \approx_{\{\alpha,C\}} \pi \cdot t$; and, *equivalence*: $\_ \vdash \_ \approx_{\{\alpha,C\}} \_$ is an equivalence relation, indeed.

Definition 2 (Nominal unification problem). *A* nominal unification problem *is a pair* $\langle \Delta, P\rangle$*, where* Δ *is a* freshness context *and* P *is a finite set of* equations *and* freshness constraints *of the form* $s \approx_? t$ *and* $a\#_? s$*, respectively,*

where $\approx_?$ is symmetric, s and t are terms and a is an atom. Nominal terms in the equations preserve the syntactic restriction that commutative symbols are only applied to tuples.

A formalised sound and complete rule-based algorithm was presented in [4], that transforms a nominal unification problem, say $\langle \Delta, P\rangle$, with commutative function symbols into a finite set of fixpoint problems that consist exclusively of equations of the form $\pi.X \approx_? X$. The transformation starts from the triple $\mathcal{P} = \langle \Delta, id, P\rangle$, where id denotes the substitution identity, and the rules act over triples building a finite set of fixpoint problems of the form $\mathcal{Q}_i = \langle \nabla_i, \sigma_i, Q_i\rangle$, for $0 \leq i \leq n$, where for each i, ∇_i is a freshness context, σ_i a substitution, and Q_i consists only of fixpoint equations.

For ∇ and ∇' freshness contexts and σ and σ' substitutions, $\nabla' \vdash \nabla\sigma$ denotes that $\nabla' \vdash a \# X\sigma$ holds for each $(a\#X) \in \nabla$; $\nabla \vdash \sigma \approx \sigma'$ denotes that $\nabla \vdash X\sigma \approx_{\{\alpha,C\}} X\sigma'$ for all X (in $dom(\sigma) \cup dom(\sigma')$).

Definition 3 (Solution for a triple or problem). *A* solution *for a triple $\mathcal{P} = \langle \Delta, \delta, P\rangle$ is a pair $\langle \nabla, \sigma\rangle$, where the following conditions are satisfied:*

1. $\nabla \vdash \Delta\sigma$;
2. *if* $a\#_? t \in P$ *then* $\nabla \vdash a \# t\sigma$;
3. *if* $s \approx_? t \in P$ *then* $\nabla \vdash s\sigma \approx_{\{\alpha,C\}} t\sigma$;
4. *there exists* λ *such that* $\nabla \vdash \delta\lambda \approx \sigma$.

A *solution for a unification problem* $\langle \Delta, P\rangle$ is a solution for the associated triple $\langle \Delta, id, P\rangle$. The *solution set* for a problem or triple $\mathcal{P}$ is denoted by $\mathcal{U}_C(\mathcal{P})$.

Definition 4 (More general solution and complete set of solutions). *For $\langle \nabla, \sigma\rangle$ and $\langle \nabla', \sigma'\rangle$ in $\mathcal{U}_C(\mathcal{P})$, we say that $\langle \nabla, \sigma\rangle$ is* more general *than $\langle \nabla', \sigma'\rangle$, denoted $\langle \nabla, \sigma\rangle \dot{\preccurlyeq} \langle \nabla', \sigma'\rangle$, if there exists a substitution λ satisfying $\nabla' \vdash \sigma\lambda \approx \sigma'$ and $\nabla' \vdash \nabla\lambda$. A subset $\mathcal{V}$ of $\mathcal{U}_C(\mathcal{P})$ is said to be a* complete set of solutions *of $\mathcal{P}$ if for all $\langle \nabla', \sigma'\rangle \in \mathcal{U}_C(\mathcal{P})$, there exists $\langle \nabla, \sigma\rangle$ in $\mathcal{V}$ that is more general than $\langle \nabla', \sigma'\rangle$.*

Example 1. Given the nominal unification problem $\mathcal{P} = \langle \emptyset, id, \{[a']\langle (a\,c).X \star (a\,b\,c).Y, (a\,b\,c\,d).X\rangle \approx_? [b']\langle X \star Y, X\rangle\}\rangle$, the algorithm in [4] transforms it into the fixpoint problems $\mathcal{Q}_1 = \langle \{a'\#X, a'\#Y\}, id, \{(a'\,b')(a\,c).X \approx_? X,\ (a'\,b')(a\,b\,c\,d).X \approx_? X,\ (a'\,b')(a\,b\,c).Y \approx_? Y\}\rangle$ and $\mathcal{Q}_2 = \langle \{a'\#Y, b'\#Y\}, \{X/(a\,c)(a'\,b').Y\}, \{(a\,b).Y \approx_? Y,\ (d\,c\,b\,a)(a'\,b').Y \approx_? Y\}\rangle$. These fixpoint problems are generated by considering '$\star$' to be a commutative symbol and by inversions on the permutations in the suspended variables.

The results in [4] include formalisations in Coq of theorems related with the following properties: **termination**: there are no possible infinite chains of applications of the unification transformation rules; **soundness**: for each possible transformation from $\mathcal{P}$ to $\mathcal{Q}$, one has that $\mathcal{U}_C(\mathcal{Q}) \subseteq \mathcal{U}_C(\mathcal{P})$; **unsolvability**: if $\mathcal{Q} = \langle \Delta, \sigma, Q\rangle$ cannot be simplified and Q contains non fixpoint equations or freshness constraints then $\mathcal{U}_C(\mathcal{Q}) = \emptyset$; and, **completeness**: if the unification problem $\mathcal{P} = \langle \nabla, id, P\rangle$ is transformed into the finite set of fixpoint problems $\mathcal{Q}_i$, for $1 \leq i \leq n$, then $\mathcal{U}_C(\mathcal{P}) = \bigcup_{i=1}^{n} \mathcal{U}_C(\mathcal{Q}_i)$.

Example 2. (Continuing Ex. 1) The unification algorithm requires a mechanism to enumerate solutions of fixpoint problems. Solutions in $\mathcal{U}(\mathcal{Q}_1)$ are built using the substitution id and combining solutions for the singleton fixpoint problems $\langle\{a'\#X\}, \{(a\,b)(a\,c).X \approx_? X\}\rangle$, $\langle\{a'\#X\}, \{(a\,b)(a\,b\,c\,d).X \approx_? X\}\rangle$ and $\langle\{a'\#Y\}, id, \{(a\,b)(a\,b\,c).Y \approx_? Y\}\rangle$. Solutions in $\mathcal{U}(\mathcal{Q}_2)$ are built with substitution $\{X/(a\,c)(a'\,b').Y\}$ and combining solutions for the fixpoint problems $\langle\{a'\#Y, b'\#Y\}, \{(a\,b).Y \approx_? Y\}\rangle$ and $\langle\{a'\#Y, b'\#Y\}, \{(d\,c\,b\,a)(a'\,b').Y \approx_? Y\}\rangle$.

2.3 Solutions of fixpoint problems through extended pseudo-cycles

The set of solutions of a singleton fixpoint problem $\langle\nabla, \{\pi.X \approx_? X\}\rangle$ is built according to the recursive definition of (*unitary*) *extended pseudo-cycles* below ([4]). The definition of extended pseudo-cycle given below is parametric on a set $\mathcal{X}$ of variables. In this way, we take into account the fact that this fixpoint problem could have been generated by a procedure to solve a problem $\mathcal{P}$ with a given set of variables and freshness constraints.

Definition 5 (Extended Pseudo-cycle). *Let $\pi.X \approx_? X$ and $\mathcal{X}$ a set of variables. The* extended pseudo-cycles *(for short,* `epc`*) κ for π relative to $\mathcal{X}$ are inductively defined from the permutation cycles of π as follows:*

1. *$\kappa = (Y)$, for any variable not occurring in $\mathcal{X}$, is an* `epc` *for π;*
2. *$\kappa = (\overline{a_0} \cdots \overline{a_{k-1}})$ is an* `epc` *for $(a_0 \ \cdots a_{k-1})$ a permutation cycle in π such that $k = 2^l$, for $l > 0$, called* trivial extended pseudo-cycle *of π.*
3. *$\kappa = (A_0 \ ... A_{k-1})$, for a length $k \geq 1$, is an* `epc` *for π, if the following conditions are simultaneously satisfied:*
   - (a) i. *each element of κ is of the form $B_i \star B_j$, where $\star$ is a commutative function symbol in the signature, and B_i, B_j are different elements of κ', an* `epc` *for π; in this case, κ will be called a* first-instance extended pseudo-cycle *of κ' for π; or*
   - ii. *each element of κ is of the form $B_i \star C_j$ for any commutative symbol $\star$, where B_i and C_j are elements of κ' and κ''* `epc`*'s for π, which might both be the same, but κ is not a first-instance* `epc` *for π; or*
   - iii. *each element of κ is of the form $\langle B_i, C_j\rangle$, where B_i and C_j are elements of κ' and κ''* `epc`*'s for π, which might both be the same; or*
   - iv. *either each element of κ is of the form $g\,B_i$ or each element is of the form $[e]\,B_i$, where g is a non commutative function symbol in the signature and $e \notin dom(\pi)$, and each B_i is an element of κ' an* `epc` *for π; or*
   - v. *each element of κ is of the form $[a_j]B_i$, where a_j are atoms in $\kappa' = (\overline{a_0} \cdots \overline{a_{k'-1}})$ a trivial* `epc` *for π, and B_i elements of κ'' an* `epc` *for π; and*
   - (b) *for $\nabla' = \cup_{Y \in Var(\kappa)}\{dom(\pi)\#Y\}$,*
     - i. *it does not hold that $\nabla' \vdash A_i \approx_{\{\alpha, C\}} A_j$ for $i \neq j$, $0 \leq i, j \leq k-1$; and*
     - ii. *for each $0 \leq i \leq k-1$ one has that $\nabla' \vdash \pi(A_i) \approx_{\{\alpha, C\}} A_{(i+1) \bmod k}$.*

Extended pseudo-cycles built using only items 2 and 3.a.i and 3.b are called pseudo-cycles. *Extended pseudo-cycles of length $k = 1$ are called* unitary.

Remark 1. Pseudo-cycles are built just from atom terms in $dom(\pi)$ and commutative function symbols, while epc's consider all nominal syntactic elements including new variables, and also non commutative function symbols.

Example 3. (Continuing Ex. 2) Given the fixpoint equation $(a'\,b')(a\,b\,c\,d).X \approx_? X$. Let $\kappa = (a\,b\,c\,d)$ and $\mathcal{X} = \{X, Y\}$. Assume, $\star$ and $\oplus$ are commutative symbols, f and g non commutative symbols. The following are pseudo-cycles relative to $\mathcal{X}$: $(\overline{a} \star \overline{d} \quad \overline{b} \star \overline{a} \quad \overline{c} \star \overline{b} \quad \overline{d} \star \overline{c})$, $(\overline{a} \star \overline{c} \oplus \overline{b} \star \overline{d})$, etc. The following are epc's relative to $\mathcal{X}$: $(f\langle\overline{a}, \overline{b}\rangle \quad f\langle\overline{b}, \overline{c}\rangle \quad f\langle\overline{c}, \overline{d}\rangle \quad f\langle\overline{d}, \overline{a}\rangle)$, $([e]\overline{a} \star \overline{c} \quad [e]\overline{b} \star \overline{d})$, $(g\langle f\overline{a}, [e]\overline{a}\rangle \quad g\langle f\overline{b}, [e]\overline{b}\rangle \quad g\langle f\overline{c}, [e]\overline{c}\rangle \quad g\langle f\overline{d}, [e]\overline{d}\rangle)$, $(\langle t, f\langle g\langle f\overline{a}, [e]\overline{b}\rangle, Z\rangle \oplus f\langle g\langle f\overline{c}, [e]\overline{d}\rangle, Z\rangle\rangle \star \langle t, f\langle g\langle f\overline{b}, [e]\overline{c}\rangle, Z\rangle \oplus f\langle g\langle f\overline{d}, [e]\overline{a}\rangle, Z\rangle\rangle\rangle)$, etc.

A relevant aspect is that only case 3.a.i of Def. 5 allows generating epc's that might be shorter than the epc's to which this case is applied. When this is the case, the length of the generated epc is half of the original one [4].

Example 4 (Continuing Ex. 3). Applying case 3.a.i to the trivial epc $(\overline{a} \;\; \overline{b} \;\; \overline{c} \;\; \overline{d})$ one obtains the epc $(\overline{a} \star \overline{c} \;\; \overline{b} \star \overline{d})$.

So, unitary epc's can only be obtained from permutation cycles of length a power of two. When a unitary epc is being generated, the last application of 3.a.i transforms a length two epc of the form $(A_0 \;\; A_1)$ into $(A_0 \star A_1)$. By condition 3.b.ii, $\nabla \vdash \pi(A_0) \approx_{\{\alpha,C\}} A_1$ and $\nabla \vdash \pi(A_1) \approx_{\{\alpha,C\}} A_0$. Therefore, $\nabla \vdash \pi(A_0 \star A_1) \approx_{\{\alpha,C\}} A_0 \star A_1$.

Another relevant aspect of this construction is that although, we are using the relation $\approx_{\{\alpha,C\}}$, by the class of nominal terms involved in the generation of epc's, only $\approx_C$ would be necessary, except for considerations related with the freshness constraints (on new variables); hence, the invariant 3.b.ii can be seen as $\pi(A_i) \approx_C A_{i+1}$, where $i+1$ is read modulo the length of the epc.

3 Soundness and completeness

As in the previous section, we consider fixpoint equations of the form $\pi.X \approx_? X$ which occur in a fixpoint problem $\mathcal{Q} = \langle\Delta, \sigma, Q\rangle$ relative to a set of variables $\mathcal{X}$.

Definition 6 (Generated solutions of singleton fixpoint problems). *For $\mathcal{Q}$ and $\pi.X \approx_? X \in Q$ as above, the set of* generated solutions *for $\langle\Delta, \{\pi.X \approx_? X\}\rangle$, denoted as $\langle\Delta, \{\pi.X \approx_? X\}\rangle_{Sol_G}$, consists of pairs of the form $\langle\nabla, \{X/s\}\rangle$ where (s) is a unitary* epc *for π such that $\nabla \vdash dom(\Delta|_X)\#s$, where $\nabla = \Delta \cup_{Y \in Var(s)} (dom(\Delta|_X)\#Y \cup dom(\pi)\#Y)$.*

Theorem 1 (Soundness of solutions of singleton fixpoint problems). *Each $\langle\nabla, \{X/s\}\rangle$ in $\langle\Delta, \{\pi.X \approx_? X\}\rangle_{Sol_G}$ is a solution of $\langle\Delta, \{\pi.X \approx_? X\}\rangle$.*

Proof. The proof follows the lines of reasoning used for non unitary epc's. By construction, the invariant that the elements of an epc of length l, $\kappa' =$

$(e_0 \dots e_{l-1})$, satisfy the property $\nabla' \vdash \pi(e_i) \approx_{\{\alpha,C\}} e_{i+1}$, where $i+1$ abbreviates $i+1$ modulo l, and $\nabla' = \cup_{Y \in Var(\kappa')} dom(\pi) \# Y$, holds. The only case in which the length of an epc decreases is 3.a.i. Thus, when this case applies to a binary epc, say $(s_0\ s_1)$, an unitary epc (s) is built, being this of the form $(s_0 \oplus s_1)$ for a commutative function symbol $\oplus$. Since by the invariant we have that $\nabla' \vdash \pi(s_i) \approx_{\{\alpha,C\}} s_{i+1}$, for $i = 0, 1$, we have that $\nabla' \vdash \pi(s_0 \oplus s_1) \approx_{\{\alpha,C\}} s_0 \oplus s_1$; thus, we have that $\nabla' \vdash \pi(s) \approx_{\{\alpha,C\}} s$. In further steps in the construction of epc's, new unitary epc's (t') might be built from unitary epc's (t) applying cases 3.a.ii, iii, iv and v, that, can easily be checked, preserve the property $\nabla' \vdash \pi(t') \approx_{\{\alpha,C\}} t'$, for $\nabla' = \cup_{Y \in Var(t')} dom(\pi) \# Y$, if $\nabla' \vdash \pi(t) \approx_{\{\alpha,C\}} t$, for $\nabla' = \cup_{Y \in Var(t)} dom(\pi) \# Y$. Therefore all unitary non-trivial epc's give a correct solution of the form $\langle \nabla', \{X/s\} \rangle$ of the problem $\langle \emptyset, \pi.X \approx_? X \rangle$. Hence, if in addition, we have that $\nabla' \cup \Delta \vdash dom(\Delta|_X) \# s$, then for $\nabla := \nabla' \cup \Delta$, the pair $\langle \nabla, \{X/s\} \rangle \in \langle \Delta, \{\pi.X \approx_? X\} \rangle_{Sol_G}$ is a solution of $\langle \Delta, \{\pi.X \approx_? X\} \rangle$. □

Assuming the symbols in the signature are denumerable, it is possible to enumerate the unitary epc's and thus the *generated solutions*. This can be done as usual, enumerating first all possible unitary epc's with an element of length bounded by a small natural, say twice the length of π, and using only the first $|\pi|$ symbols in the signature and atoms in $dom(\pi)$; then, this length is increased generating all extended unitary epc's with elements of length $|\pi| + 1$ and using only the first $|\pi + 1|$ symbols in the signature and atoms in $dom(\pi)$ and so on.

The following result, proved by induction in the construction of the epc's, is used in the proof of completeness of generated solutions for fixpoint problems.

Lemma 1 (Extended pseudo-cycle correspondence for π and π^2). *For $k \geq 1$, $(A_0 \cdots A_{2^k-1})$ is an* epc *for π if, and only if, there exist $(B_0 \cdots B_{2^{k-1}-1})$ and $(C_0 \cdots C_{2^{k-1}-1})$* epc*'s for π^2 with a substitution σ such that atoms in its image belong to $dom(\pi) \backslash dom(\pi^2)$, and for $0 \leq j \leq 2^{k-1} - 1$ one has $B_j\sigma \approx_{\{\alpha,C\}} A_{2j}$ and $C_j\sigma \approx_{\{\alpha,C\}} A_{2j+1}$.*

Example 5. For $(a\,b)$ and $(c\,d\,e\,f)$, permutation cycles of π, one has that (a), (b), $(c\,e)$ and $(d\,f)$ are permutation cycles of π^2, and also, $a, b \in dom(\pi) \backslash dom(\pi^2)$. Therefore, supposing that '$+$', '$*$' and '$\star$' are commutative function symbols, $((\overline{c} * \overline{e}) + \overline{a}) \star ((\overline{d} * \overline{f}) + \overline{b})$ and $((\overline{c} * \overline{e}) + Y) \star ((\overline{d} * \overline{f}) + Y')$ are respectively unitary epc's of π and π^2. Then:

- $\langle \Delta, \{X/((\overline{c} * \overline{e}) + \overline{a}) \star ((\overline{d} * \overline{f}) + \overline{b})\} \rangle \in \langle \Delta, \pi.X \approx_? X \rangle_{Sol_G}$ iff
- $\langle \Delta', \{X/((\overline{c} * \overline{e}) + Y) \star ((\overline{d} * \overline{f}) + Y')\} \rangle \in \langle \Delta, \pi^2.X \approx_? X \rangle_{Sol_G}$,

where $\Delta' = \Delta \cup dom(\pi^2) \# Y, Y' \cup dom(\Delta|_X) \# Y, Y'$. So the σ of Lemma 1 will be $\{Y/\overline{a}, Y'/\overline{b}\}$, so that $((\overline{c} * \overline{e}) + \overline{a}\ \ (\overline{d} * \overline{f}) + \overline{b})$ is an epcof π, $((\overline{c} * \overline{e}) + Y)$ and $((\overline{d} * \overline{f}) + Y')$ are epc's of π^2, with $((\overline{c} * \overline{e}) + Y)\sigma = (\overline{c} * \overline{e}) + \overline{a}$ and $((\overline{d} * \overline{f}) + Y')\sigma = (\overline{d} * \overline{f}) + \overline{b}$.

Example 6. Let $\pi = (a\ b\ c\ d\ e\ f\ g\ h)$ then $\pi^2 = (a\ c\ e\ g)(b\ d\ f\ h)$. There are solutions of $\langle \emptyset, \pi^2 \cdot X \approx_? X \rangle$ that are not solutions of $\langle \emptyset, \pi.X \approx_? X \rangle$:

- $\langle \emptyset, X/(\overline{a} \oplus \overline{e}) \oplus (\overline{c} \oplus \overline{g}) \rangle, \langle \emptyset, X/(\overline{b} \star \overline{f}) \oplus (\overline{d} \star \overline{h}) \rangle \in \langle \emptyset, \pi^2 \cdot X \approx_? X \rangle_{Sol_G}$;

– $\langle \emptyset, X/((\overline{a} \oplus \overline{e}) \oplus (\overline{c} \oplus \overline{g})) \oplus ((\overline{b} \star \overline{f}) \oplus (\overline{d} \star \overline{h}))\rangle \in \langle \emptyset, \pi^2 \cdot X \approx_? X\rangle_{Sol_G}$

but none of them is a solution for $\langle \emptyset, \pi.X \approx_? X\rangle$.

However there exist solutions in the intersection of both problems, for instance, $\langle \emptyset, X/((\overline{a} \oplus \overline{e}) \oplus (\overline{c} \oplus \overline{g})) * (X/(\overline{b} \oplus \overline{f}) \oplus (\overline{d} \oplus \overline{h}))\rangle$.

Theorem 2 (Completeness of solutions for singleton fixpoint problems). *Let $\langle \Delta, \{\pi.X \approx_? X\}\rangle$ be a singleton fixpoint problem with a solution $\langle \nabla, \{X/s\}\rangle$. Then there exists $\langle \nabla', \{X/t\}\rangle \in \langle \Delta, \{\pi.X \approx_? X\}\rangle_{Sol_G}$ such that $\langle \nabla', \{X/t\}\rangle \preceq \langle \nabla, \{X/s\}\rangle$.*

Proof. Since $\langle \nabla, \{X/s\}\rangle$ is a solution of the problem, it follows that $\nabla \vdash \Delta\{X/s\}$ and $\nabla \vdash \pi(s) \approx_{\{\alpha,C\}} s$. The proof is done by induction on the structure of s.
Base Case. This case will be split in two parts.

1. $s = \overline{a}$.
   The pair $\langle \nabla, \{X/\overline{a}\}\rangle$ is a solution only if $a \notin dom(\Delta|_X) \cup dom(\pi)$, then $\emptyset \vdash \pi \cdot \overline{a} = \overline{a}$. Let Y be a new variable and $\nabla' = dom(\Delta|_X)\#Y \cup dom(\pi)\#Y$, then $\langle \nabla', \{X/Y\}\rangle$ is a generated solution. Let $\sigma = \{Y/\overline{a}\}$, notice that $\nabla \vdash \nabla'\sigma$ and $Y\sigma = \overline{a}$. Therefore, $\langle \nabla', \{X/Y\}\rangle \preceq \langle \nabla, \{X/\overline{a}\}\rangle$.
2. $s = \pi'.Y$ and $dom(\pi)\#\pi'.Y$.
   Notice that $\langle \nabla, \{X/\pi'.Y\}\rangle \in \langle \Delta, \pi.X \approx_? X\rangle_{Sol_C}$ only if $\nabla \vdash dom(\Delta|_X)\#\pi'.Y, dom(\pi)\#\pi'.Y$, that is, if $\nabla \vdash (\pi')^{-1} \cdot dom(\Delta|_X)\#Y$ and $\nabla \vdash (\pi')^{-1} \cdot dom(\pi)\#Y$, so that $\Delta \cup ((\pi')^{-1} \cdot dom(\Delta|_X) \cup (\pi')^{-1} \cdot dom(\pi))\#Y \subset \nabla$.
   Let $\langle \nabla', \{X/Z\}\rangle \in \langle \Delta, \pi.X \approx_? X\rangle_{Sol_G}$ with $\nabla' = \Delta \cup dom(\pi) \cup dom(\Delta|_X)\#Z$, Consider the substitution $\sigma = \{Z/\pi'.Y\}$, then $\nabla \vdash Z\sigma \approx_{\{\alpha,C\}} \pi'.Y$ and $\nabla'\sigma = \Delta\sigma \cup (dom(\pi) \cup dom(\Delta|_X))\#Z\sigma = \Delta \cup (\pi')^{-1} \cdot dom(\pi)\#Y \cup (\pi')^{-1} \cdot dom(\Delta|_X)\#Y$, so $\nabla \vdash \nabla'\sigma$. Therefore, $\langle \nabla', \{X/Z\}\rangle \preceq \langle \nabla, \{X/\pi'.Y\}\rangle$.

Induction Step.

1. $s = \langle s_1, s_2\rangle$
   In this case $\nabla \vdash \pi(\langle s_1, s_2\rangle) \approx_{\{\alpha,C\}} \langle s_1, s_2\rangle$, that is, $\nabla \vdash \langle \pi(s_1), \pi(s_2)\rangle \approx_{\{\alpha,C\}} \langle s_1, s_2\rangle$, which implies in $\nabla \vdash \pi(s_i) \approx_{\{\alpha,C\}} s_i$, for $i = 1, 2$.
   By i.h. and Definitions 5 and 6, there exist $\langle \nabla'_1, \{X/t_1\}\rangle, \langle \nabla'_2, \{X/t_2\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ s.t. (t_1), (t_2) and $(\langle t_1, t_2\rangle)$ are unitary `epc`'s w.r.t. π. Furthermore $\langle \nabla'_i, \{X/t_i\}\rangle \preceq \langle \nabla, \{X/s_i\}\rangle$, i.e., there exist substitutions λ_i s.t. $\nabla \vdash \nabla_i\lambda_i$ and $\nabla \vdash t_i\lambda_i \approx s_i$, for $i = 1, 2$. One can choose (t_1) and (t_2) s.t. $Var(t_1) \cap Var(t_2) = \emptyset$ and $dom(\lambda_i) \cap Var(s_j) = \emptyset$, for $i, j = 1, 2$. Then, $\nabla \vdash \langle t_1, t_2\rangle\lambda_1\lambda_2 \approx_{\{\alpha,C\}} \langle s_1, s_2\rangle$, and $\nabla \vdash (\nabla_1 \cup \nabla_2)\lambda_1\lambda_2$, that is, $\langle \nabla_1 \cup \nabla_2, \{X/\langle t_1, t_2\rangle\}\rangle \preceq \langle \nabla, \{X/\langle s_1, s_2\rangle\}\rangle$.
2. $s = fs'$
   Since $\nabla \vdash \pi \cdot fs' \approx_{\{\alpha,C\}} fs'$, it follows that $\nabla \vdash f(\pi(s')) \approx_{\{\alpha,C\}} fs'$ and therefore, $\nabla \vdash \pi(s') \approx_{\{\alpha,C\}} s'$. By i.h. and Defs. 5 and 6, there exist $\langle \nabla', \{X/t'\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ such that (t') and (ft') are unitary `epc`'s w.r.t. π. Furthermore $\langle \nabla', \{X/t'\}\rangle \preceq \langle \nabla, \{X/s'\}\rangle$, that is, there exist a substitution σ such that $\nabla \vdash \nabla'\sigma$ and $\nabla \vdash t'\sigma \approx_{\{\alpha,C\}} s'$, and since

$\nabla \vdash ft'\sigma \approx_{\{\alpha,C\}} f(t'\sigma) \approx_{\{\alpha,C\}} fs'$ and adding f at the top of t' does not change the variables of t', therefore, $\langle \nabla', \{X/ft'\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ and $\langle \nabla', \{X/ft'\}\rangle \preceq \langle \nabla, \{X/fs'\}\rangle$.

3. $s = [e]s'$.
   (a) $e \notin dom(\pi)$
   Since $\nabla \vdash \pi([e]s') \approx_{\{\alpha,C\}} [e]s'$, it follows that $\nabla \vdash \pi(s') \approx_{\{\alpha,C\}} s'$, i.e., $\langle \nabla, X/s'\rangle$ is a solution for $\langle \Delta, \pi \cdot X \approx_? X\rangle$. By i.h. and Defs. 5 and 6, there exist $\langle \nabla', \{X/t'\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ such that (t') and $([e]t')$ are unitary epc's w.r.t. π. Furthermore $\langle \nabla', \{X/t'\}\rangle \preceq \langle \nabla, \{X/s'\}\rangle$, i.e., there exist a substitution σ such that $\nabla \vdash \nabla'\sigma$ and $\nabla \vdash t'\sigma \approx_{\{\alpha,C\}} s'$, therefore, $\langle \nabla', \{X/[e]t'\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ and $\langle \nabla', \{X/[e]t'\}\rangle \preceq \langle \nabla, \{X/[e]s'\}\rangle$.
   (b) $e \in dom(\pi)$.
   By hypothesis, $\nabla \vdash \pi([e]s') \approx_{\{\alpha,C\}} [e]s'$, i.e., $\nabla \vdash [\pi \cdot e](\pi(s')) \approx_{\{\alpha,C\}} [e]s'$, and $\nabla \vdash \pi(s') \approx_{\{\alpha,C\}} (\pi \cdot e \;\; e)(s')$ only if $\nabla \vdash (\pi \cdot e)\#s'$. Notice that e occurs in s' iff $\pi \cdot e$ occurs in s'. Therefore, for $\nabla \vdash e\#s'$, it follows that $\nabla \vdash \pi(s') \approx_{\{\alpha,C\}} s'$ and the result follows by induction hypothesis.

4. $s = s_1 \oplus s_2$
   This case has two parts:
   (a) $\nabla \vdash \pi(s_1) \approx_{\{\alpha,C\}} s_1$ and $\nabla \vdash \pi(s_2) \approx_{\{\alpha,C\}} s_2$.
   By i.h. and Definitions 5 and 6, there exist $\langle \nabla'_1, \{X/t_1\}\rangle, \langle \nabla'_2, \{X/t_2\}\rangle \in \langle \Delta, \pi \cdot X \approx_? X\rangle_{Sol_G}$ s.t. (t_1), (t_2) and $(t_1 \oplus t_2)$ are unitary epc's w.r.t. π. Furthermore $\langle \nabla'_i, \{X/t_i\}\rangle \preceq \langle \nabla, \{X/s_i\}\rangle$, i.e., there exist substitutions λ_i s.t. $\nabla \vdash \nabla_i\lambda_i$ and $\nabla \vdash t_i\lambda_i \approx s_i$, for $i = 1, 2$. One can choose (t_1) and (t_2) s.t. $Var(t_1) \cap Var(t_2) = \emptyset$ and $dom(\lambda_i) \cap Var(s_j) = \emptyset$, for $i, j = 1, 2$. Then, $\nabla \vdash (t_1 \oplus t_2)\lambda_1\lambda_2 \approx_{\{\alpha,C\}} (s_1 \oplus s_2)$, and $\nabla \vdash (\nabla_1 \cup \nabla_2)\lambda_1\lambda_2$, that is, $\langle \nabla_1 \cup \nabla_2, \{X/t_1 \oplus t_2\}\rangle \preceq \langle \nabla, \{X/s_1 \oplus s_2\}\rangle$.
   (b) $\nabla \vdash \pi(s_1) \approx_{\{\alpha,C\}} s_2$ and $\nabla \vdash \pi(s_2) \approx_{\{\alpha,C\}} s_1$.
   Notice that $\nabla \vdash \pi^2(s_1) \approx_{\{\alpha,C\}} \pi(s_2) \approx_{\{\alpha,C\}} s_1$ and $\nabla \vdash \pi^2(s_2) \approx_{\{\alpha,C\}} \pi(s_1) \approx_{\{\alpha,C\}} s_2$. Therefore, $\langle \nabla, \{X/s_1\}\rangle$ and $\langle \nabla, \{X/s_2\}\rangle$ are solutions of $\langle \Delta, \pi^2.X \approx_? X\rangle$. By IH, there exist $\langle \nabla_1, \{X/t_1\}\rangle, \langle \nabla_2, \{X/t_2\}\rangle \in \langle \Delta, \pi^2 \cdot X \approx_? X\rangle_{Sol_G}$ such that $\langle \nabla_i, \{X/t_i\}\rangle \preceq \langle \nabla, \{X/s_i\}\rangle$. Then there exist substitutions λ_i s.t. $\nabla \vdash \nabla_i\lambda_i$ and $\nabla \vdash t_i\lambda_i \approx_{\{\alpha,C\}} s_i$, for $i = 1, 2$. One can choose (t_1) and (t_2) s.t. $Var(t_1) \cap Var(t_2) = \emptyset$ and $dom(\lambda_i) \cap Var(s_j) = \emptyset$, for $i, j = 1, 2$.
   Therefore, $\langle \nabla_1 \cup \nabla_2, X/t_1 \oplus t_2\rangle \in \langle \Delta, \pi^2.X \approx_? X\rangle_{Sol_G}$ and $\langle \nabla_1 \cup \nabla_2, X/t_1 \oplus t_2\rangle \preceq \langle \nabla, X/s_1 \oplus s_2\rangle$, via substitution $\lambda = \lambda_1\lambda_2$.
   Notice that $\nabla \vdash \pi(t_1)\lambda \approx_{\{\alpha,C\}} \pi(s_1) \approx_{\{\alpha,C\}} s_2 \approx_{\{\alpha,C\}} t_2\lambda$ and analogously, $\nabla \vdash \pi(t_2)\lambda \approx_{\{\alpha,C\}} t_1\lambda$. Hence, λ is a solution for the C-unification problem $\{\pi(t_1) =_? t_2, \pi(t_2) =_? t_1\}$. Let $\langle \nabla', \lambda'\rangle$ be a solution more general than $\langle \nabla, \lambda\rangle$ such that the atoms in the image of λ' are in $dom(\pi) \backslash dom(\pi^2)$. Since (t_1) and (t_2) are unitary epc's of π^2, it follows by Lemma 1, that $(t_1\lambda' \; t_2\lambda')$ is an epc for π. By Definition 5, $(t_1\lambda' \oplus t_2\lambda')$ is a unitary epcfor π, such that $\langle \nabla', \{X/t_1\lambda' \oplus t_2\lambda'\}\rangle \in \langle \Delta, \{\pi \cdot X \approx_? X\}\rangle_{Sol_G}$ and $\langle \nabla', \{X/t_1\lambda' \oplus t_2\lambda'\}\rangle \preceq \langle \nabla, \{X/s_1 \oplus s_2\}\rangle$.

□

Remark 2. Notice that to build just a most general set of C-unifiers, without taking into account nominal equivalence, in the proof of Lemma 2 (case 4.b) and Def. 7 one can use the algorithm proposed by Siekmann [13], which provides a finite, minimal and complete set of C-unifiers.

Definition 7 (General C-matchers). *Let s_i, for $i = 1..k$, be nominal terms. A* most general C-matcher *of these terms, if it exists, is a most general C-unifier δ of the C-unification problem $\{s_i =_? Z\}_{i=1..k}$, where Z is a new variable for s_i, with $i = 1..k$.*

Remark 3. Alternatively, Definition 5 could be restricted to ground terms (by removing the first case in the construction of `epc`'s), and then instead of computing C-matchers via C-unification, one could use an α-C-equivalence checker (for example, the one specified in [3]). This would also simplify case iv in Definition 5, since it would be sufficient to consider just one atom e' not in $dom(\pi)$.

Definition 8 (Generated solutions for a variable). *Let the fixpoint problems for X in $\mathcal{P}$ be given by $\langle \nabla, \pi_i.X \approx_? X\rangle$, for $\pi_i \in \Pi_X$, and such that $|\Pi_X| = k$. If there exist*

- *solutions $\langle \nabla_i, \{X/t_i\}\rangle \in \langle \nabla, \pi_i.X \approx_? X\rangle_{Sol_G}$ for each fixpoint problem and*
- *a most general C-matcher δ of the terms $\{t_i\}_{i=1..k}$ with X as new variable*

such that the problem $\langle \emptyset, \cup_{(a\#Y)\in\nabla''}\{a\#Y\delta\}\rangle$, where $\nabla'' := \cup_{i=1}^{k}\nabla_i$, has a solution $\langle \nabla', \emptyset\rangle$, then we say that $\langle \nabla', \{X/X\delta\}\rangle$ is a generated solution *for X. The set of all generated solutions is denoted by $[X]_{\mathcal{P}_G}$.*

Example 7. Let $P_i := \pi_i.X \approx_? X$, for $i = 1..3$, be fixpoint equations for $\pi_1 = (a\ b\ c\ d)$, $\pi_2 = (a\ c)$ and $\pi_3 = (b\ d)$ and suppose that $\mathcal{P} := \langle \nabla, P\rangle$ is a fixpoint problem where P_i for $i = 1..3$ are the fixpoint equations for X in P.

1. $\langle \nabla \cup a, b, c, d\#Y, \delta_1 := \{X/((a * c) * (b * d)) \oplus Y\}\rangle \in \langle \nabla, P_1\rangle_{Sol_G}$;
2. $\langle \nabla \cup a, c\#Y', Y'', \delta_2 := \{X/((a * c) * Y') \oplus Y''\}\rangle \in \langle \nabla, P_2\rangle_{Sol_G}$; and
3. $\langle \nabla \cup b, d\#Y_1', Y_1'', \delta_3 := \{X/((b * d) * Y_1') \oplus Y_1''\}\rangle \in \langle \nabla, P_3\rangle_{Sol_G}$.

Notice that $\delta = \{X/((a*c)*(b*d))\oplus Y'',\ Y'/(b*d),\ Y_1'/(a*c),\ Y/Y'', Y_1''/Y''\}$ is a most general C-unifier of terms $\{t_i := X\delta_i\}$ with variable X.

According to the definiton, the set of initial freshness constraints is given as $\nabla'' = \nabla \cup \{a, b, c, d\#Y,\ a, c\#Y', Y'',\ b, d\#Y_1', Y_1''\}$. Notice that $Y'' \in Var(im(\delta))$, have to satisfy the constraints on Y_1'', Y and X, that is, $a, b, c, d\#Y''$ is a new constraint on Y'', inherited from the constraints of the variables in the domain of δ. $\langle \nabla', \emptyset\rangle$ is the solution of $\langle \emptyset, \cup_{(a\#Y)\in\nabla''}\{a\#Y\delta\}\rangle$, and then it holds that $\nabla' \vdash dom(\nabla''|_Z)\#Z\delta$, for all $Z \in dom(\delta)$. Thus, $\langle \nabla', \{X/X\delta\}\rangle$ belongs to $[X]_{\mathcal{P}_G}$.

Example 8. (Continuing Ex. 3) Consider the singleton fixpoint problems on the variable X in $\mathcal{Q}_1$ of Ex. 1 and 2 relative to the variable set $\mathcal{X} = \{X, Y\}$: $\langle \{a'\#X\}, \{Eq_1 := (a'\ b')(a\ c).X \approx_? X\}\rangle$ and $\langle \{a'\#X\}, \{Eq_2 := (a'\ b')(a\ b\ c\ d).X \approx_? X\}\rangle$. Since $a'\#X$ is in the freshness context, there is no combinatory solution with occurrences of the atoms in the permutation cycle $(a'\ b')$. For the cycles $(a\ c)$ and $(a\ b\ c\ d)$, in equations Eq_1 and Eq_2, possible solutions include, respectively:

- $\langle \nabla, \{X/(\overline{a}+\overline{c}) \star Z\}\rangle$, $\langle \nabla, \{X/(f\overline{a}+f\overline{c}) \star Z\}\rangle$, $\langle \nabla, \{X/([g]\overline{a}+[g]\overline{c}) \star Z\}\rangle$, for $\nabla = a, c, a', b' \# Z, a' \# X$;
- $\langle \nabla', \{X/(\overline{a}+\overline{c}) \star (\overline{b}+\overline{d})\}\rangle$, $\langle \nabla', \{(f\overline{a}+f\overline{c}) \star (f\overline{b}+f\overline{d})\}\rangle$, $\langle \nabla', \{([g]\overline{a}+[g]\overline{c}) \star ([g]\overline{b}+[g]\overline{d})\}\rangle$, for $\nabla' = a' \# X$.

Since the general C-matchers for pairs of these three solutions for Eq_1 and Eq_2 are respectively $\{Z/\overline{b}+\overline{d}\}$, $\{Z/f\overline{b}+f\overline{d}\}$ and $\{Z/[g]\overline{b}+[g]\overline{d}\}$, the combined solutions for both singleton fixpoint problems are those given for Eq_2.

Now, one proves that the set of solutions $[X]_{\mathcal{P}_G}$ is correct and complete.

Corollary 1 (Soundness and completeness of generated solutions for a variable). *Let $\mathcal{P} = \langle \Delta, P\rangle$ be a fixpoint problem. Any solution in $[X]_{\mathcal{P}_G}$ is a solution of each fixpoint equation for X in $\mathcal{P}$. If $\langle \nabla, \{X/s\}\rangle$ is a solution for each fixpoint equation for X in $\mathcal{P}$ then there exists $\langle \nabla', \{X/X\delta\}\rangle \in [X]_{\mathcal{P}_G}$ such that $\langle \nabla', \{X/X\delta\}\rangle \preceq \langle \nabla, \{X/s\}\rangle$*

Proof. By Theorem 1 and Def. 6:
(*Soundness)* Each solution $\langle \nabla_i, \{X/t_i\}\rangle$ in $\langle \Delta, \{\pi_i.X \approx_? X\}\rangle_{Sol_G}$ is a correct solution for $\langle \Delta, \{\pi_i.X \approx_? X\}\rangle$, for $\pi_i \in \Pi_X$. Suppose $\langle \nabla', \{X/X\delta\}\rangle$ belongs to $[X]_{\mathcal{P}_G}$. Since δ is a C-unifier of terms t_i with variable X, we have that $X\delta \approx_C t_i\delta$, and also that $\nabla_i \vdash \pi(t_i) \approx_{\{\alpha,C\}} t_i$. Thus, $\nabla' \vdash \pi(t_i)\delta \approx_{\{\alpha,C\}} t_i\delta$ since by definition we also have that $\nabla' \vdash dom(\nabla|_X) \# X\delta$, because by construction for all $Y \in Var(X\delta)$, ∇' includes the freshness constraints $dom(\nabla''|_X) \# Y$ and ∇'' is an extension of ∇.
(*Completeness)*
For $|\Pi_X| = k$, $i = 1..k$, there are $\langle \nabla_i, \{X/t_i\}\rangle \in \langle \Delta, \{\pi_i.X \approx_? X\}\rangle_{Sol_G}$, solution of $\langle \Delta, \{\pi_i.X \approx_? X\}\rangle$, such that $\langle \nabla_i, \{X/t_i\}\rangle \preceq \langle \nabla, \{X/s\}\rangle$. Then, for each i, there exists a ∇_i s.t. $\nabla \vdash \nabla_i\lambda_i$ and $\nabla \vdash \{X/t_i\}\lambda_i \approx \{X/s\}$. One can choose each t_i in a way to satisfy $\cap_{i=1}^k Var(t_i) = \emptyset$, and then for $\lambda = \lambda_1 \cdots \lambda_k$ and $\nabla'' = \cup_{i=1}^k \nabla_i$, one also has $\nabla \vdash \nabla''\lambda$ and $\nabla \vdash \{X/t_i\}\lambda \approx \{X/s\}$.
Notice that $\langle \nabla, \lambda\rangle$ is a nominal C-unifier for the problem $\langle \nabla_i, \cup_{i=1}^k \{t_i \approx_? X\}\rangle$. Then, given δ, a most general C-unifier for $\{t_i =_? X\}_{i=1..k}$, it holds that there exists λ' such that $\nabla \vdash \delta\lambda' \approx \lambda$.
Let $\langle \nabla', \emptyset\rangle$ be a solution of $\langle \emptyset, \cup_{(a\#Y)\in\nabla''} \{a\#Y\delta\}\rangle$, then, by Def. 6, one has that $\langle \nabla', \{X/X\delta\}\rangle \in [X]_{\mathcal{P}_G}$, and so, since $\nabla \vdash \nabla''\lambda$, also that $\nabla \vdash \nabla''\delta\lambda'$, which is the same that $\nabla \vdash \nabla'\lambda'$. On the other hand, $X\delta \approx_C t_i\delta$ and then $\nabla \vdash s \approx_{\{\alpha,C\}} t_i\lambda \approx_{\{\alpha,C\}} t_i\delta\lambda' \approx_C X\delta\lambda'$, which implies $\nabla \vdash \{X/s\} \approx \{X/X\delta\}\lambda'$. Hence, $\langle \nabla', \{X/X\delta\}\rangle \preceq \langle \nabla, \{X/s\}\rangle$.

□

Definition 9 (Generated Solutions for fixpoint problems). *Let $\mathcal{P}$ be a fixpoint problem. The set of generated solutions for $\mathcal{P}$, denoted as $[\mathcal{P}]_{Sol_G}$, is defined as the set that contains all solutions of the form*

$$\left\langle \bigcup_{X \in Var(P)} \nabla_X, \quad \bigcup_{X \in Var(P)} \{X/s_X\} \right\rangle, \text{ where each } \langle \nabla_X, \{X/s_X\}\rangle \in [X]_{\mathcal{P}_G}.$$

Example 9. (Continuing Ex. 8) Consider third singleton fixpoint problems on the variable Y in $\mathcal{Q}_1$ relative to $\mathcal{X} = \{X, Y\}$: $\langle\{a'\#Y\}, \{(a'\,b')(a\,b\,c).Y \approx_? Y\}\rangle$. There exists no possible combinatorial solution since $a'\#Y$ is in the freshness context and the length of permutation cycle $(a\,b\,c)$ is not a power of two. The only possible solution is given as $\langle a, b, c, a', b'\#Y', \{Y/Y'\}\rangle$. Hence, using the solutions in Ex. 8 for the fixpoint equations on X, one has the following solutions for the fixpoint problem $\mathcal{Q}_1$, where $\Delta = a', b'\#X, a, b, c, a', b'\#Y'$:

- $\langle \Delta, \{X/(\overline{a} + \overline{c}) \star (\overline{b} + \overline{d}), Y/Y'\}\rangle$
- $\langle \Delta, \{X/(f\overline{a} + f\overline{c}) \star (f\overline{b} + f\overline{d}), Y/Y'\}\rangle$
- $\langle \Delta, \{X/([g]\overline{a} + [g]\overline{c}) \star ([g]\overline{b} + [g]\overline{d}), Y/Y'\}\rangle$

A similar analysis can be done for the fixpoint problem $\mathcal{Q}_2$ in Ex. 1. Also, for the fixpoint equations $(a\,b).Y \approx_? Y$ and $(d\,c\,b\,a)(a'\,b').Y \approx_? Y$, the permutation cycle $(a'\,b')$ avoids any possible combinatorial solution with occurrences of the atoms a' or b'. Cycles $(a\,b)$ and $(d\,c\,b\,a)$ will allow combinatorial solutions for each of these equations, but we will see (Ex. 14) that they cannot be combined.

Corollary 2 (Soundness and completeness of generated solutions for fixpoint problems). *Let $\mathcal{P}$ be a fixpoint problem. Any solution in the set of solutions $[\mathcal{P}]_{Sol_G}$ is a correct solution of $\mathcal{P}$. For any $\langle\nabla, \delta\rangle$ solution of $\mathcal{P}$ there exist a pair $\langle\nabla', \sigma\rangle \in [\mathcal{P}]_{Sol_G}$ such that $\langle\nabla', \sigma\rangle \preceq \langle\nabla, \delta\rangle$.*

Proof. By Def. 9 and Corollary 1:
(Soundness) A solution of $\mathcal{P}$ is of the form $\left\langle \bigcup_{X \in Var(P)} \nabla_X, \bigcup_{X \in Var(P)} \{X/s_X\} \right\rangle$, where each $\langle\nabla_X, \{X/s_X\}\rangle \in [X]_{\mathcal{P}_G}$ is a correct solution for all fixpoint equations in $\mathcal{P}$ for the variable X, this completes the soundness proof.
(Completeness) Let $\mathcal{P} = \{\langle \Delta, \cup_{i=1}^{k} \{\pi_{i_1}.X_1 \approx_? X_1\}_{\pi_{i_1} \in \Pi_{X_1}}\}\rangle$ and $\langle\nabla, \delta\rangle$ be a solution of $\mathcal{P}$. There exist more general solutions $\langle\nabla_j, \{X_j/t_j\}\rangle \in [X_j]_{\mathcal{P}_G}$, for $j = 1, \ldots, k$; i.e., $\langle\nabla_j, \{X_j/t_j\}\rangle \preceq \langle\nabla, \delta\rangle$; hence, there is a solution for $\mathcal{P}$ of the form $\langle \bigcup_j \nabla_j, \bigcup_j \{X_j/t_j\}\rangle$ is in $[\mathcal{P}]_{Sol_G}$ and $\langle \bigcup_j \nabla_j, \bigcup_j \{X_j/t_j\}\rangle \preceq \langle\nabla, \delta\rangle$. □

A greedy procedure for the generation of solutions in $[X]_{\mathcal{P}}$ proceeds as follows. Follow the construction of generated solutions in Definition 6 for each fixpoint problem $\langle\nabla, \pi_i.X \approx_? X\rangle$ in P, where $\pi_i \in \Pi_X$, as given in Lemma 1; for each generated solution $\langle\nabla', \{X/s\}\rangle$ build the freshness context $\nabla'' = \nabla' \cup \bigcup_{Y \in Var(s)} dom(\nabla|_X)\#Y \cup dom(\Pi_X)\#Y$ and check whether $\langle\nabla'', \{X/s\}\rangle$ is a solution for all $\langle\nabla, \pi_i.X \approx_? X\rangle$, for $\pi_i \in \Pi_X$. Here, $dom(\Pi_X)\#Y$ abbreviates $\cup_{\pi_i \in \Pi_X} dom(\pi_i)\#Y$.

4 Improvements in the generation of solutions

The greedy procedure can be improved eliminating generation of solution of non interesting permutation cycles in Π_X, according to the observations below.

In first place, notice that according to the theory of pseudo-cycles, we are interested in building solutions with atoms that occur only in permutation cycles of length a power of two in all permutations $\pi \in \Pi_X$.

In second place, notice that if there exist permutation cycles of length a power of two $\kappa_i \in \pi_i$ and $\kappa_j \in \pi_j$, for $\pi_i, \pi_j \in \Pi_X$, such that $dom(\pi_i) \cap dom(\pi_j) \neq \emptyset$, $dom(\pi_i) \setminus dom(\pi_j) \neq \emptyset$ and $dom(\pi_j) \setminus dom(\pi_i) \neq \emptyset$, then there might not be possible solutions with occurrences of atom terms in the domain of π_i and/or π_j for the fixpoint equations related with permutations π_i and π_j. The simplest example is given by permutation cycles $(a\,b)$ and $(a\,c)$. The precise relation between permutation cycles that allows for construction of solutions for all permutations in Π_X is given in the next definition.

Definition 10 (Permutation factor). *A permutation π is said to be an* n-factor *of a permutation π' whenever there exists n such that $\pi^n = \pi'$.*

Example 10. Let $\pi = (a\,b\,c\,d\,e\,f\,g\,h)$. The odd powers of π, π^1, $\pi^3 = (a\,d\,g\,b\,e\,h\,c\,f)$, $\pi^5 = (a\,f\,c\,h\,e\,b\,g\,d)$ and $\pi^7 = (a\,h\,g\,f\,e\,d\,c\,b)$ are the only factors of π.

Remark 4. For a permutation cycle κ of length 2^k, the factors corresponding to permutation cycles of the same length are exactly the permutations cycles κ^p, for p odd such that $0 < p < 2^k$; also, if λ is a p-factor of κ then λ is the q-factor of κ, where q is the minimum odd number such that $0 < q < 2^k$ and $p \cdot q = 1$ modulo 2^k. For instance, if κ is a permutation cycle of length 2^4, κ^3, κ^5, κ^7, etc, are respectively the 11- 13- and 7-factors, etc, of κ.

The key observation about permutation cycles κ and λ, of respective lengths 2^k and 2^l, for $k \geq l \geq 0$, such that, $\kappa^{2^{k-l}}$ contains a permutation cycle, say ν, that is a p-factor of λ, is that this happens if and only if regarding elements in $dom(\lambda)$, possible generated solutions from both permutation cycles coincide. Indeed, first, notice that either $l = 0$ and then $\nu = \lambda$ or $l > 0$ and $\lambda^{2^{l-1}}$ consists of 2^{l-1} permutation cycles of length two; second, observe that if $l > 0$, then $\lambda^{2^{l-1}} = \nu^{p \cdot 2^{l-1}} = \nu^{2^{l-1}}$, since p is an odd number (such that $0 < p < 2^l$). Moreover, notice that $\kappa^{2^{k-l}}|_{dom(\lambda)} = \nu$, that implies that $\kappa^{2^{k-1}}|_{dom(\lambda)} = \nu^{2^{l-1}}$. Thus, the permutation cycles of length two generated from κ and λ, restricted to $dom(\lambda)$ are the same, which implies that commutative combinations built (according to Def. 5) regarding to the elements in $dom(\lambda)$ are the same.

Example 11. Consider $\kappa = (a\,b\,c\,d\,e\,f\,g\,h)$ and $\lambda = (a\,g\,e\,c)$. Notice that $\kappa^2 = (a\,c\,e\,g)(b\,d\,f\,h)$ and λ is a 3-factor of $\nu = (a\,c\,e\,g)$. Then $\lambda^2 = \nu^{3 \cdot 2} = \nu^2 = (a\,e)(c\,g)$. Also, notice that the unitary `epc`'s built from λ and ν are the same.

Definition 11 (Permutation cycles in the top of Π_X). *Let Π_X be the set of permutations for fixpoint equations on the variable X in a fixpoint problem. A permutation cycle $\kappa \in \pi \in \Pi_X$* is in the top *of Π_X, whenever for all atoms $a \in dom(\kappa)$ and all $\pi' \in \Pi_X$, if $a \in dom(\pi')$, and a is an element in a permutation cycle λ in π', then there exists a natural m such that the permutation cycle of the element a in π^{2^m}, say ν, is a factor of the permutation cycle λ.*

Example 12. Consider the permutations $\pi_1 = (a\,b\,c\,d\,e\,f\,g\,h)$, $\pi_2 = (a\,g\,e\,c)(b\,f)$ and $\pi_3 = (a\,e)(c\,g)(d\,h)$. The permutation cycle π_1 is in the top of the set of permutations; indeed, notice that all permutation cycles in all permutations appear

as a factor in powers of two of π_1: $\pi_1^0 = (a\,b\,c\,d\,e\,f\,g\,h)$; $\pi_1^2 = (a\,c\,e\,g)(b\,d\,f\,h)$; $\pi_1^4 = (a\,e)(c\,g)(b\,f)(d\,h)$; $\pi_1^8 = (a)(e)(c)(g)(b)(f)(d)(h)$.

Theorem 3 (Atoms of interest in fixpoint problems on a variable). *Let Π_X be the set of permutations for fixpoint equations on the variable X in a fixpoint problem. Only the set of atoms in the domain of permutation cycles in the top of Π_X might occur in solutions of all fixpoint equations on X.*

Proof. Only atoms that are in permutation cycles of length a power of two in all permutations $\pi \in \Pi_X$ might occur in solutions of all fixpoint equations on X. Suppose a is an atom that only occurs in permutation cycles of length a power of two for all $\pi \in \Pi_X$ and let κ be a permutation cycle in Π_X of maximal length, say 2^k, with $a \in dom(\kappa)$. Suppose λ is a permutation cycle in ϕ, for some $\phi \in \Pi_X$, with $a \in dom(\lambda)$ and let 2^l be the length of λ. Only if λ is a factor of a permutation cycle in $\pi^{2^{k-l}}$, say ν such that $\nu^p = \lambda$, the epc's built from λ (and from κ) will maintain the invariants required, restricted to the atoms in $dom(\lambda)$, that is for an epc built from λ of the form $(A_0 \ldots A_{2^m-1})$, where $m \leq l$, $\phi(A_i) \approx_C A_{i+1}$ and $\phi^{2^{l-m}}(A_i) \approx_C A_i$, where $i+1$ reads modulo 2^m. This also holds for λ. Hence, since ν is a p-factor of λ (and also, $\pi^{2^{k-l}}|_{dom(\lambda)} = \nu$), one has that $\nu^p(A_i) \approx_C A_{i+1}$ and $\nu^{p\cdot 2^{l-m}}(A_i) \approx_C A_i$. If the epc is of length two, that is it is of the form $(A_0\,A_1)$, we have $m = 1$ and $\nu^{p\cdot 2^{l-1}}(A_i) \approx_C A_i$, for $i = 0, 1$, and since p is odd, this implies that $\nu^{2^{l-1}}(A_i) \approx_C A_i$, for $i = 0, 1$. This condition also holds for π, since $(\pi^{2^{k-l}}|_{dom(\nu)})^{2^{l-1}} = (\nu)^{2^{l-1}}$; hence, $\pi^{2^{k-1}}(A_i) = A_{i+1}$, for $i = 0, 1$. If κ is not a permutation cycle in the top of Π_X, then there exists some permutation cycle $\lambda \in \phi \in \Pi_X$, such that $a \in dom(\kappa) \cap dom(\lambda)$, 2^l is the length of λ, but the permutation cycle of length 2^l in $\kappa^{2^{k-l}}$, say ν, such that $a \in dom(\nu)$ is not a factor of λ. Thus, since $\nu^{2^{l-1}} \neq \lambda^{2^{l-1}}$ atoms in the domains of ν and λ cannot be combined uniformly to build common solutions for κ and λ (i.e., for π and ψ).

To finish we show how a common solution can be built when κ is in the top of Π_X. Suppose that (A) is a unitary epc built from λ by successive applications of case 3.a.i. of Definition 5 halving in each step the length of the epc. We have that $\lambda(A) = A$. It is possible to generate an epc for κ of the form $(A\,\kappa(A)\,\kappa^2(A) \ldots \kappa^{2^{k-l}-1}(A))$. From this epc it is possible to build a unitary epc by successive applications of case 3.a.i. of Definition 5, first obtaining $(A \star_1 \kappa^{2^{K-l-1}}(A)\ \kappa(A) \star_1 \kappa^{2^{k-l-1}+1}(A) \ldots\ \kappa^{2^{k-l-1}-1}(A) \star_1 \kappa^{2^{k-l}-1}(A))$, and so on until a unitary epc of the form $((\cdots((A \star_1 B_1) \star_2 B_2)\cdots) \star_{k-l} B_{k-l})$ is obtained where the B_i's, for $1 \leq i \leq k-l$ are adequate combinations of the terms $\kappa(A), \ldots \kappa^{2^{k-l}-1}(A)$ according to the constructions of epc's. From this epc one has the solution for $\pi.X \approx_? X$ of the form $\langle \emptyset, \{X/(\cdots((A \star_1 B_1) \star_2 B_2)\cdots) \star_{k-l} B_{k-l}\}\rangle$, where $\star_j$, for $j = 1, \ldots, l$ are commutative symbols. Using the unitary cycle (A) for λ and cases 1 and 3.a.ii of Definition 5 one can generate the unitary epc $((\cdots((A \star_1 Y_1) \star_2 Y_2)\cdots) \star_{k-l} Y_{k-l})$ which gives the solution $\langle \nabla, \{X/(\cdots((A \star_1 Y_1) \star_2 Y_2)\cdots) \star_{k-l} Y_{k-l}\}\rangle$ for λ, where $\nabla = \{dom(\lambda)\#Y_j | 1 \leq j \leq l\}$. The C-unification problem $\langle \nabla, X \approx_? (\cdots((A \star_1$

$B_1) \star_2 B_2) \cdots) \star_{k-l} B_{k-l}, X \approx_? (\cdots ((A \star_1 Y_1) \star_2 Y_2) \cdots) \star_{k-l} Y_{k-l}\}\rangle$ unifies with solution $\langle \emptyset, \{X/(\cdots ((A \star_1 B_1) \star_2 B_2) \cdots) \star_{k-l} B_{k-l}\}\rangle$ which is a common solution for π and ϕ.

Example 13. (Continuing example 12) First, notice that the permutation cycle $\pi_1 = (a\,b\,c\,d\,e\,f\,g\,h)$ is not in the top of $(a\,d\,e\,b\,g\,h\,c\,f)$; also, π_1 is neither in the top of $(a\,b\,c\,d)$ nor in the top of $(a\,i)$. Since π_1 is not a factor of π_2, solutions generated from the `epc` $(\bar{a}\,\bar{d}\,\bar{e}\,\bar{b}\,\bar{g}\,\bar{h}\,\bar{c}\,\bar{f})$ might not be solutions built for π_1; for instance, consider the unitary `epc` built for π_2, $(((\bar{a} \star \bar{g}) \diamond (\bar{e} \star \bar{c})) \oplus ((\bar{d} \star \bar{h}) \diamond (\bar{b} \star \bar{f}))$, which is not a solution for π_1, since not $\pi_1((\bar{a} \star \bar{g}) \diamond (\bar{e} \star \bar{c})) \approx_C (\bar{d} \star \bar{h}) \diamond (\bar{b} \star \bar{f})$. Also, for the `epc` $(\bar{a}\,\bar{b}\,\bar{c}\,\bar{d})$: the permutation cycles in π_1^2 are $(a\,c\,e\,g)$ and $(b\,d\,f\,h)$, which give different solutions. For $(\bar{a}\,\bar{i})$, the permutation cycle $(a\,e)$ in π_i^4 will produce different solutions.

Now consider solutions of fixpoint equations $\pi_i.X \approx_? X$, for $i = 1, 2, 3$, where Π_X consists of the permutations $\pi_1 = (a\,b\,c\,d\,e\,f\,g\,h)$, $\pi_2 = (a\,g\,e\,c)(b\,f\,)$ and $\pi_3 = (a\,e)(c\,g)(d\,h)$. In this case, we have seen (Example 12) that π_1 is a permutation cycle in the top of Π_X. Among the solutions generated for $\pi_i.X \approx_? X$, for $i = 1, 2, 3$ through `epc`'s we have, respectively:
$\langle \nabla_1, \{X/s_1 = ((\bar{a} + \bar{e}) \star (\bar{c} + \bar{g})) \oplus ((\bar{b} + \bar{f}) \star (\bar{d} + \bar{h})\}\rangle$,
$\langle \nabla_2, \{X/s_2 = ((\bar{a} + \bar{e}) \star (\bar{c} + \bar{g})) \oplus ((\bar{b} + \bar{f}) \star Y\}\rangle$ and
$\langle \nabla_3, \{X/s_3 = ((\bar{a} + \bar{e}) \star (\bar{c} + \bar{g})) \oplus (Z \star (\bar{d} + \bar{h})\}\rangle$,
where $\nabla_1 = \emptyset$, $\nabla_2 = \{a\#Y, b\#Y, c\#Y, e\#Y, f\#Y, g\#Y\}$ and $\nabla_3 = \{a\#Z, c\#Z, d\#Z, e\#Z, g\#Z, h\#Z\}$, and the symbols $\oplus, \star$ and $+$ are commutative. The C-unification problem $\langle \nabla_1 \cup \nabla_2 \cup \nabla_3, \{X \approx_? s_1, X \approx_? s_2, X \approx_? s_3\}\rangle$ has solution $\{X/s_1, Y/\bar{d} + \bar{h}, Z/\bar{b} + \bar{f}\}$ with the respective freshness constraints; thus, restricting this solution to the freshness constraints on X we have the common solution $\langle \emptyset, \{X/s_1\}\rangle$.

Example 14. (Continuing Ex. 9) As we saw in Ex. 9, the fixpoint equations $(a\,b).Y \approx_? Y$ and $(d\,c\,b\,a)(a'\,b').Y \approx_? Y$ in the fixpoint problem $\mathcal{Q}_2$, have no possible combinatorial solution with occurrences of the atoms a' or b'. By Theorem 3, cycles $(a\,b)$ and $(d\,c\,b\,a)$ will not give rise to possible combinatorial solutions for both fixpoint equations. Hence, there is no feasible combinatory solution for this fixpoint problem. Therefore, the unique possible solution for $\mathcal{Q}_2$ is $\langle \{a', b'\#X, a, b, c, d, a', b'\#Y\}, \{X/(a\,c)(a'\,b').Y\}\rangle$.

The greedy generation algorithm can then be improved by generating solutions only for the atoms in permutation cycles in the top of Π_X.

5 Conclusions and future work

We presented a procedure to generate solutions of fixpoint nominal C-unification problems modulo commutativity. The procedure is proved to be sound and complete. This result is relevant to provide a sound and complete procedure to generate solutions of nominal C-unification problems, which consists of an initial phase in which nominal C-unification problems are transformed into an equivalent finite set of fixpoint problems, as described in [4], and a second phase that

generates a potentially infinite set of independent solutions, presented in this paper, based on combinatorial properties of permutations.

Additional improvements of the generation procedure should be investigated exhaustively, as well as possible extensions of nominal unification and matching, and nominal narrowing modulo other equational theories of interest.

References

[1] T. Aoto and K. Kikuchi. *A Rule-Based Procedure for Equivariant Nominal Unification*. In *Pre-proc. of Higher-Order Rewriting (HOR)*, pages 1–5, 2016.

[2] T. Aoto and K. Kikuchi. *Nominal Confluence Tool*. In *Proc. of 8th Int. Joint Conf.: Automated Reasoning (IJCAR)*, volume 9706 of *LNCS*, pages 173–182. Springer, 2016.

[3] M. Ayala-Rincón, W. Carvalho-Segundo, M. Fernández, and D. Nantes-Sobrinho. *A Formalisation of Nominal Equivalence with Associative-Commutative Function Symbols*. *ENTCS*, 332:21–38, 2017. Post-proc. Eveventh Logical and Semantic Frameworks with Applications (LSFA).

[4] M. Ayala-Rincón, W. Carvalho-Segundo, M. Fernández, and D. Nantes-Sobrinho. *Nominal C-Unification*. Av. at `ayala.mat.unb.br/publications.html`, 2017.

[5] M. Ayala-Rincón, M. Fernández, and D. Nantes-Sobrinho. *Nominal Narrowing*. In *Proc. of 1st Int. Conf. on Formal Structures for Computation and Deduction (FSCD)*, volume 52 of *LIPIcs*, pages 1–16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[6] M. Ayala-Rincón, M. Fernández, and A. C. Rocha-oliveira. *Completeness in PVS of a Nominal Unification Algorithm*. *ENTCS*, 323:57–74, 2016.

[7] F. Baader and T. Nipkow. *Term Rewriting and* All That. CUP, 1998.

[8] J. Cheney. Equivariant unification. *J. of Automated Reasoning*, 45:267–300, 2010.

[9] M. Fernández and M. J. Gabbay. *Nominal Rewriting*. *Information and Computation*, 205(6):917–965, 2007.

[10] M. Schmidt-Schauß, T. Kutsia, J. Levy, and M. Villaret. *Nominal Unification of Higher Order Expressions with Recursive Let*. *CoRR*, abs/1608.03771, 2016.

[11] A. M. Pitts. Nominal Logic, a First Order Theory of Names and Binding. *Information and Computation*, 186(2):165–193, 2003.

[12] Sagan, B. E. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*, volume 203 of *Graduate Texts in Math.* Springer, 2nd edition, 2001.

[13] J. H. Siekmann. *Unification of Commutative Terms*. In *Proc. of the International Symposium on Symbolic and Algebraic Manipulation*, volume 72 of *LNCS*, pages 22–29. Springer, 1979.

[14] C. Urban. *Nominal Unification Revisited*. In *Proc. of Int. Work. on Unification (UNIF)*, volume 42 of *EPTCS*, pages 1–11, 2010.

[15] C. Urban, A. M. Pitts, and M. J. Gabbay. *Nominal Unification*. *Theoretical Computer Science*, 323(1-3):473497, 2004.