The Symbolic Term Abstract Domain

Patrick Cousot
Coutant Institute of Mathematical Studies, Computer Science Department
New York University
New York, NY, USA
pcousot@cims.nyu.edu

***Abstract*—We construct the abstract domain of symbolic terms ordered by subsumption by abstraction of the powerset of ground terms ordered by inclusion.**

***Index Terms*—Abstract interpretation, Abstract domain, Symbolic term, Subsumption.**

I. Introduction

In mathematical logic, Jacques Herbrand introduced *ground terms* [1, Ch. 1][1] to denote a basic mathematical object (for example, 0) or operation on objects (such as $+(1, 2)$) as well as *symbolic terms* that is *terms with variables* (where the variables x are unknowns standing for any ground term [1, Ch. 2]) (for example, $+(1, x)$).

Gordon Plotkin [3], [4] and John Reynolds [5] proved that the set of symbolic terms form a complete lattice with the *less general/subsumption* partial order $\preceq^\nu$ on terms. For example, $+(1, 2) \preceq^\nu +(1, y) \preceq^\nu +(x, y) \preceq^\nu z$.

Symbolic terms are of interest in various areas of Computer Science such as refutation theorem-proving based on the resolution rule of inference [6], [7], satisfiability modulo theories [8], symbolic execution [9], type inference [10], [11], logic and constraint programming [12]–[16], [17]–[19], pointer analysis in imperative [20] or logic languages [21], and so on.

In [22], we showed that Hindley's monotypes with variables [23] as well as Milner's polymorphic types [10] are abstractions of sets of Church's monotypes [24]. Thanks to restrictions on types (for example, no union type) and on the language (for example, the two branches of a conditional must have the same type), the set of monotypes of a lambda-expression is exactly represented by a monotype with variables. In that case the abstraction is exact. This is no longer the case for polymorphic types, for which a widening is needed (all recursive calls must have the same type).

Generalizing this initial point view, our objective is to study the complete lattice of symbolic terms by abstraction of the powerset of ground terms.

This material is based upon work supported by DARPA under Agreement No. HR00112020022. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the United States Government or DARPA.

[1] English translation in [2].

II. The complete lattice of ground terms

The *signature* F defines a set of function symbols $f\backslash n$ (f for brevity), each one with an *arity* n, that is, a fixed number of parameters (0 for constants). The round parentheses "(", ")" and comma "," do not belong to F.

$$\begin{array}{lclll} f\backslash n,\ g\backslash n,\ h\backslash n & \in & \mathsf{F}\backslash n & & \text{signature } n \geqslant 0 \\ f, g, h & \in & \mathsf{F} & = \displaystyle\bigcup_{n\in\mathbb{N}} \mathsf{F}\backslash n & \end{array}$$

We assume that the signature F has at least two different function symbols. Ground terms denote uninterpreted functional expressions.

$$\begin{array}{llll} \mathbf{t} \in \mathsf{T} & ::= & & \text{ground terms} \\ & & f\backslash 0 & \text{constants of arity } 0 \\ & \mid & f\backslash n(\mathbf{t}_1, \ldots, \mathbf{t}_n) & \text{term of arity } n \in \mathbb{N}^+ \end{array}$$

The set T of all ground terms is called the *Herbrand universe* with signature F.

Sets of ground terms form a complete lattice partially ordered by inclusion

$$\langle \wp(\mathsf{T}),\ \subseteq,\ \emptyset,\ \mathsf{T},\ \cup,\ \cap\rangle \qquad \text{sets of ground terms} \tag{1}$$

III. Terms with variables

A term with variables (also called *symbolic term*) abstracts a set of terms. For example the set of ground terms $\{+(0, 1), +(0, +(1, 1)), +(0, +(1, +(1, 1))), +(0, +(1, +(1, +(1, 1))))), \ldots\}$ can be abstracted by the term $+(0, \alpha)$ with variable α. The abstraction can be very imprecise. For example $\{0, +(0, 1)\}$ would be abstracted by variable α which concretization is the set of all ground terms. So the abstraction is precise enough only for set of terms with adequate regularity properties.

$$\begin{array}{llll} \alpha, \beta, \gamma & \in & \mathbb{V}_{\mathfrak{t}} & \text{term variables} \\ \boldsymbol{\tau} \in \mathsf{T}^\nu & ::= & & \text{terms with variables (2)} \\ & & f\backslash 0 & \text{constants} \\ & \mid & f\backslash n(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n) & \text{term of arity } n \in \mathbb{N} \\ & \mid & \alpha & \text{term variable} \end{array}$$

The round parentheses "(", ")", comma " , ", and variables $\alpha \in \mathbb{V}_{\mathfrak{t}}$ do not belong to F.

We write $vars[\![\boldsymbol{\tau}]\!]$ for the free variables of a term $\boldsymbol{\tau}$.

$$\begin{array}{rcll} vars[\![\alpha]\!] & \triangleq & \{\alpha\} & \alpha \in \mathbb{V}_{\mathfrak{t}} \quad (3) \\ vars[\![f(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n)]\!] & \triangleq & \displaystyle\bigcup_{i=1}^{n} vars[\![\boldsymbol{\tau}_i]\!] & \end{array}$$

By default "term" means "term with or without variables" and we say "ground term" for "term without variable" and "symbolic term" for "term with variables." The tag $^\nu$ means "with variables".

Example 1. *Type expressions in OCaml [25] are terms with variables (using a quote* 'ident *to stand for the type variable named* ident*, the infix notation* -> *for the function type, and the postfix notation* list *for lists)*

```
# List.map;;
- : ('a -> 'b) -> 'a list -> 'b list = <fun>
```

In prefix notation that would be $\texttt{->}(\texttt{->}(\alpha,\beta),\texttt{->}(\texttt{list}(\alpha),\texttt{list}(\beta)))$.

The *syntactic replacement* $\boldsymbol{\tau}[\alpha \leftarrow \boldsymbol{\tau}']$ on terms with variables $\boldsymbol{\tau}$ replaces all instances of a variable α in the term $\boldsymbol{\tau}$ by another term with variables $\boldsymbol{\tau}'$.

$$\begin{aligned}
\alpha[\alpha \leftarrow \boldsymbol{\tau}'] &\triangleq \boldsymbol{\tau}' && (4)\\
\beta[\boldsymbol{\tau}' \leftarrow \alpha] &\triangleq \beta \qquad \text{when } \beta \neq \alpha\\
f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)[\alpha \leftarrow \boldsymbol{\tau}'] &\triangleq f(\boldsymbol{\tau}_1[\alpha \leftarrow \boldsymbol{\tau}'],\ldots,\boldsymbol{\tau}_n[\alpha \leftarrow \boldsymbol{\tau}'])
\end{aligned}$$

IV. Term assignments

An assignment maps variables to ground terms.

$$\boldsymbol{\varrho} \in \mathsf{P}^\nu \quad \triangleq \quad \mathbb{V}_{\mathfrak{t}} \to \mathsf{T} \qquad \text{assignment} \tag{5}$$

An assignment can be homomorphically extended to a term with variables, as follows.

$$\boldsymbol{\varrho}(f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)) \quad \triangleq \quad f(\boldsymbol{\varrho}(\boldsymbol{\tau}_1),\ldots,\boldsymbol{\varrho}(\boldsymbol{\tau}_n)) \tag{6}$$

The intuition is that $\boldsymbol{\varrho} \in \mathsf{T}^\nu \to \mathsf{T}$ is the evaluation $\boldsymbol{\varrho}(\boldsymbol{\tau})$ of term $\boldsymbol{\tau}$ by replacing variables α of $\boldsymbol{\tau}$ by their value $\boldsymbol{\varrho}(\alpha)$ which is a ground term. Variable assignment $\boldsymbol{\varrho}[\alpha \leftarrow \mathbf{t}]$ can be used to change the value of a variable α to $\mathbf{t}$

$$\begin{aligned}
\boldsymbol{\varrho}[\mathrm{x} \leftarrow v](\mathrm{x}) &\triangleq v && (7)\\
\boldsymbol{\varrho}[\mathrm{x} \leftarrow v](\mathrm{y}) &\triangleq \boldsymbol{\varrho}(\mathrm{y}) \quad \text{when} \quad \mathrm{x} \neq \mathrm{y}
\end{aligned}$$

We use the same notation for syntactic replacement (4) and variable assignment (7) because of the following lemma 1 showing that instantiation of syntactic replacement and environment assignment commute.

Lemma 1. $\boldsymbol{\varrho}(\boldsymbol{\tau}[\alpha \leftarrow \boldsymbol{\tau}']) = \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau})$.

Proof of lemma 1. By structural induction on $\boldsymbol{\tau}$.

- If $\boldsymbol{\tau} = \alpha$ then, by (4), $\boldsymbol{\varrho}(\boldsymbol{\tau}[\alpha \leftarrow \boldsymbol{\tau}']) = \boldsymbol{\varrho}(\boldsymbol{\tau}'[\alpha \leftarrow \alpha]) = \boldsymbol{\varrho}(\boldsymbol{\tau}')$ while, by (7), $\boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau}) = \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\alpha)$ $= \boldsymbol{\varrho}(\boldsymbol{\tau}')$, as required;
- If $\boldsymbol{\tau} = \beta \neq \alpha$ then, by (4), $\boldsymbol{\varrho}(\boldsymbol{\tau}[\alpha \leftarrow \boldsymbol{\tau}']) = \boldsymbol{\varrho}(\boldsymbol{\tau}'[\alpha \leftarrow \beta]) = \boldsymbol{\varrho}(\beta)$. This is equal to $\boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau}) = \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\beta) = \boldsymbol{\varrho}(\beta)$, by (7);
- Otherwise, $\boldsymbol{\tau} = f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)$ so that, by (4), (6), ind. hyp., and (6) again, $\boldsymbol{\varrho}(\boldsymbol{\tau}'[\alpha \leftarrow \boldsymbol{\tau}]) = \boldsymbol{\varrho}(f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)[\alpha \leftarrow \boldsymbol{\tau}']) = f(\boldsymbol{\varrho}(\boldsymbol{\tau}_1[\alpha \leftarrow \boldsymbol{\tau}']),\ldots,\boldsymbol{\varrho}(\boldsymbol{\tau}_n[\boldsymbol{\tau}' \leftarrow \alpha])) = f(\boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau}_1), \ldots, \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau}_n)) = \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)) = \boldsymbol{\varrho}[\alpha \leftarrow \boldsymbol{\varrho}(\boldsymbol{\tau}')](\boldsymbol{\tau})$. □

Let us call $\boldsymbol{\varrho}(\boldsymbol{\tau})$ the *ground instance* of $\boldsymbol{\tau}$ for the assignment $\boldsymbol{\varrho}$. Unless it is reduced to a variable, a term with variables cannot have the same instance as any one of its variables (this is known as *occur-check*).

Lemma 2. *For all variables* $\alpha \in vars[\![\boldsymbol{\tau}]\!]$ *of a term with variables* $\boldsymbol{\tau} \in \mathsf{P}^\nu \setminus \mathbb{V}_{\mathfrak{t}}$*, there is no assignment* $\boldsymbol{\varrho} \in \mathsf{P}^\nu \triangleq \mathbb{V}_{\mathfrak{t}} \to \mathsf{T}$ *such that* $\boldsymbol{\varrho}(\alpha) = \boldsymbol{\varrho}(\boldsymbol{\tau})$.

Proof of lemma 2. Since $\boldsymbol{\tau} \in \mathsf{P}^\nu \setminus \mathbb{V}_{\mathfrak{t}}$, we have $\boldsymbol{\tau} = f^1(\boldsymbol{\tau}^1_1,\ldots,f^n(\boldsymbol{\tau}^n_1,\ldots,\alpha,\ldots,\boldsymbol{\tau}^n_{m^n}),\ldots,\boldsymbol{\tau}^1_{m^1})$ where $f^i \in \mathsf{F}_{m^i}$ with $m^i > 0$, $i \in [1,n]$. By (6), we have $\boldsymbol{\varrho}(\boldsymbol{\tau}) = f^1(\boldsymbol{\tau}^1_1,\ldots,f^n(\boldsymbol{\tau}^n_1,\ldots,\boldsymbol{\varrho}(\alpha),\ldots,\boldsymbol{\tau}^n_{m^n})$ $= f^1(\boldsymbol{\tau}^1_1,\ldots,f^n(\boldsymbol{\tau}^n_1,\ldots,f^1(\boldsymbol{\tau}^1_1,\ldots,f^n(\boldsymbol{\tau}^n_1,$ $\ldots,\boldsymbol{\varrho}(\alpha),\ldots,\boldsymbol{\tau}^n_{m^n}),\ldots,\boldsymbol{\tau}^n_{m^n}) = \ldots$. So $\boldsymbol{\varrho}(\boldsymbol{\tau})$ is an infinite object not in P^ν. □

V. The symbolic abstraction

The symbolic abstraction abstracts a set of ground terms into a term with variables. The symbolic abstraction is easily defined by its concretization, that is, it's set of ground instances.

$$\begin{aligned}
ground(\boldsymbol{\tau}) &\triangleq \{\boldsymbol{\varrho}(\boldsymbol{\tau}) \mid \boldsymbol{\varrho} \in \mathsf{P}^\nu\} && (8)\\
ground(\overline{\varnothing}^\nu) &\triangleq \emptyset
\end{aligned}$$

Since all terms with variables $\boldsymbol{\tau} \in \mathsf{P}^\nu$ have a nonempty concretization $ground(\boldsymbol{\tau})$, we add the empty term $\overline{\varnothing}^\nu \notin \mathsf{P}^\nu$ to denote the empty set $\emptyset$ with $\boldsymbol{\varrho}(\overline{\varnothing}^\nu) = \overline{\varnothing}^\nu$.

Remark 1 (the symbolic abstraction is relational)**.** *An important remark on the definition of ground in (8) is that all instances* $\boldsymbol{\varrho}(\alpha)$ *of a variable* α *in a term with variables* $\boldsymbol{\tau}$ *are the same in a given instance* $\boldsymbol{\varrho}(\boldsymbol{\tau})$ *of the term. For example,* $f(a,b) \notin ground(f(\alpha,\alpha))$ *when* $a \neq b$*. However, two terms with variables equal up to variable renaming have the same concretization. Therefore, the names attributed to the same instances of variables in terms with variables do not matter. For example,* $ground(f(\alpha,\alpha)) = ground(f(\beta,\beta)) = \{f(\mathbf{t},\mathbf{t}) \mid \mathbf{t} \in \mathsf{T}\}$.

VI. The Herbrand symbolic abstract domain

The symbolic abstract domain is an abstraction of the complete lattice of ground term properties (1).

A. The subsumption partial order

We define the preorder $\preceq^\nu$ on terms with variables, called *subsumption*, as the inclusion of sets of their ground instances. (This will be shown to be equivalent to the classical definition in theorem 2.)

$$(\boldsymbol{\tau} \preceq^\nu \boldsymbol{\tau}') \quad \triangleq \quad (ground(\boldsymbol{\tau}) \subseteq ground(\boldsymbol{\tau}')) \tag{9}$$

This is a preorder $\langle \mathsf{T}^\nu \cup \{\overline{\varnothing}^\nu\}, \preceq^\nu\rangle$ with infimum $\overline{\varnothing}^\nu$. For example $f(a,b) \preceq^\nu f(\alpha,b) \preceq^\nu f(\alpha,\beta) \preceq^\nu \gamma$.

Lemma 3. *Observe that for all terms with variables* $\boldsymbol{\tau},\boldsymbol{\tau}' \in \mathsf{T}^\nu$*, we have* $\boldsymbol{\tau} \preceq^\nu \boldsymbol{\tau}'$ *if and only if* $\forall \boldsymbol{\varrho} \in \mathsf{P}^\nu \,.\, \exists \boldsymbol{\varrho}' \in \mathsf{P}^\nu \,.\, \boldsymbol{\varrho}(\boldsymbol{\tau}) = \boldsymbol{\varrho}'(\boldsymbol{\tau}')$.

Proof of lemma 3. The case of $\emptyset$ is trivial. Otherwise,

$$
\begin{array}{ll}
 & \boldsymbol{\tau} \preceq^{\nu} \boldsymbol{\tau}' \\
\Leftrightarrow & \mathit{ground}(\boldsymbol{\tau}) \subseteq \mathit{ground}(\boldsymbol{\tau}') \qquad \wr\text{def. (9) of } \preceq^{\nu}\wr \\
\Leftrightarrow & \{\boldsymbol{\varrho}(\boldsymbol{\tau}) \mid \boldsymbol{\varrho} \in \mathsf{P}^{\nu}\} \subseteq \{\boldsymbol{\varrho}'(\boldsymbol{\tau}') \mid \boldsymbol{\varrho}' \in \mathsf{P}^{\nu}\} \\
 & \qquad \wr\text{def. (8) of } \mathit{ground}\wr \\
\Leftrightarrow & \forall \boldsymbol{\varrho} \in \mathsf{P}^{\nu} \,.\, \exists \boldsymbol{\varrho}' \in \mathsf{P}^{\nu} \,.\, \boldsymbol{\varrho}(\boldsymbol{\tau}) = \boldsymbol{\varrho}'(\boldsymbol{\tau}') \qquad \wr\text{def. } \subseteq\wr \quad \square
\end{array}
$$

The corresponding equivalence relation is $\simeq^{\nu}$. The quotient is a partial order $\langle \mathcal{P}^{\mathrm{H}}, \preceq_{\simeq^{\nu}}\rangle$ where

$$
\begin{array}{rcl}
\boldsymbol{\tau} \simeq^{\nu} \boldsymbol{\tau}' & \triangleq & \boldsymbol{\tau} \preceq^{\nu} \boldsymbol{\tau}' \wedge \boldsymbol{\tau}' \preceq^{\nu} \boldsymbol{\tau} \\
\mathcal{P}^{\mathrm{H}} & \triangleq & (\mathsf{T}^{\nu} \cup \{\overline{\varnothing}^{\nu}\})/_{\simeq^{\nu}} \\
 & \triangleq & \{[\boldsymbol{\tau}]_{\simeq^{\nu}} \mid \boldsymbol{\tau} \in \mathsf{T}^{\nu} \cup \{\overline{\varnothing}^{\nu}\}\} \\
[\boldsymbol{\tau}]_{\simeq^{\nu}} \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}']_{\simeq^{\nu}} & \triangleq & \exists \overline{\boldsymbol{\tau}} \in [\boldsymbol{\tau}]_{\simeq^{\nu}}, \overline{\boldsymbol{\tau}}' \in [\boldsymbol{\tau}']_{\simeq^{\nu}} \,.\, \overline{\boldsymbol{\tau}} \preceq^{\nu} \overline{\boldsymbol{\tau}}'
\end{array} \tag{10}
$$

For example $f(\alpha,\alpha) \simeq^{\nu} f(\beta,\beta)$ and $[f(\alpha,\alpha)]_{\simeq^{\nu}} = \{f(\gamma,\gamma) \mid \gamma \in \mathbb{V}_{\mathfrak{t}}\}$. More generally, equivalent terms are equal up to variable renaming.

Lemma 4. *A renaming is an assignment $\rho \in \mathbb{V}_{\mathfrak{t}} \twoheadrightarrow \mathbb{V}_{\mathfrak{t}}$ between variables extended to terms with variables by (6), that is $\rho(f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)) = f(\rho(\boldsymbol{\tau}_1),\ldots,\rho(\boldsymbol{\tau}_n))$. Equivalent terms have a bijective renaming of their variables and reciprocally, that is, $\forall \boldsymbol{\tau}, \boldsymbol{\tau}' \in \mathsf{T}^{\nu} \,.\, (\boldsymbol{\tau} \simeq^{\nu} \boldsymbol{\tau}') \Leftrightarrow (\exists \rho \in \mathit{vars}[\![\boldsymbol{\tau}]\!] \rightarrowtail\!\!\!\!\rightarrow \mathit{vars}[\![\boldsymbol{\tau}']\!] \,.\, \rho(\boldsymbol{\tau}) = \boldsymbol{\tau}')$.*

Proof of lemma 4. ($\Rightarrow$) Assume $\boldsymbol{\tau} \simeq^{\nu} \boldsymbol{\tau}'$ are equivalent so that, by def. $\simeq^{\nu}$ and lemma 3, $\forall \boldsymbol{\varrho} \,.\, \exists \boldsymbol{\varrho}' \,.\, \boldsymbol{\varrho}(\boldsymbol{\tau}) = \boldsymbol{\varrho}'(\boldsymbol{\tau}')$ and $\forall \boldsymbol{\varrho}' \,.\, \exists \boldsymbol{\varrho} \,.\, \boldsymbol{\varrho}(\boldsymbol{\tau}) = \boldsymbol{\varrho}'(\boldsymbol{\tau}')$. Let us define a relation $\rho \in \wp(\mathit{vars}[\![\boldsymbol{\tau}]\!] \times \mathit{vars}[\![\boldsymbol{\tau}']\!])$, starting from $\emptyset$ as follows.

- If $\boldsymbol{\tau}$ is a variable α and $\boldsymbol{\tau}'$ is not then $\boldsymbol{\tau} \not\simeq^{\nu} \boldsymbol{\tau}'$ so $\boldsymbol{\tau}'$ must be a variable β and we let $\langle\alpha,\ \beta\rangle \in \rho$.
- It $\boldsymbol{\tau} = f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_)$ then $\boldsymbol{\tau} \simeq^{\nu} \boldsymbol{\tau}'$ implies that $\forall n \in [1,n] \,.\, \boldsymbol{\tau}_k \simeq^{\nu} \boldsymbol{\tau}'_k$, so by structural induction, there is a relation $\rho_k \in \wp(\mathit{vars}[\![\boldsymbol{\tau}_k]\!] \times \mathit{vars}[\![\boldsymbol{\tau}'_k]\!])$ so we take $\rho = \bigcup_{k=1}^{n} \rho_k$.

We have to prove that ρ is a function. By contradiction, if ρ is not a function then there is a variable α of $\boldsymbol{\tau}$ and two variables β and γ of $\boldsymbol{\tau}'$ with at least one which is not α, say γ. Then instances of γ in $\boldsymbol{\tau}'$ cannot be replicated with α in $\boldsymbol{\tau}$ so the two terms cannot be equivalent. Now if ρ is not injective there there are two variables β and γ of $\boldsymbol{\tau}$ with only one correspondent α of $\boldsymbol{\tau}$, so again the instances of β and γ cannot be matched with α. Finally, if ρ is not surjective, then there is a variable γ with arbitrary instantiations in $\boldsymbol{\tau}'$ with no correspondent in $\boldsymbol{\tau}$, which again prevents equivalence. In conclusion, ρ is a bijection.

($\Leftarrow$) Conversely let $\rho \in \mathit{vars}[\![\boldsymbol{\tau}]\!] \rightarrowtail\!\!\!\!\rightarrow \mathit{vars}[\![\boldsymbol{\tau}']\!]$ be such $\rho(\boldsymbol{\tau}) = \boldsymbol{\tau}'$. Given any $\boldsymbol{\varrho} \in \mathsf{P}^{\nu}$, define $\boldsymbol{\varrho}'(\alpha) \triangleq \boldsymbol{\varrho}(\rho^{-1}(\alpha))$, $\alpha \in \mathbb{V}_{\mathfrak{t}}$. Let us show, by structural induction on α, that $\boldsymbol{\varrho}(\boldsymbol{\tau}) = \boldsymbol{\varrho}'(\boldsymbol{\tau}')$.

- If $\boldsymbol{\tau} = \alpha$ then $\rho \in \mathit{vars}[\![\boldsymbol{\tau}]\!] \rightarrowtail\!\!\!\!\rightarrow \mathit{vars}[\![\boldsymbol{\tau}']\!]$ and $\rho(\boldsymbol{\tau}) = \boldsymbol{\tau}'$ imply that $\boldsymbol{\tau}' = \beta$ is the variable $\beta = \rho(\boldsymbol{\tau})$. It follows that $\boldsymbol{\varrho}'(\boldsymbol{\tau}') = \boldsymbol{\varrho}'(\beta) \triangleq \boldsymbol{\varrho}(\rho^{-1}(\beta)) = \boldsymbol{\varrho}(\alpha) = \boldsymbol{\varrho}(\boldsymbol{\tau})$;
- Otherwise, $\boldsymbol{\tau} = f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)$ so that $\rho(\boldsymbol{\tau}) = \rho(f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)) = f(\rho(\boldsymbol{\tau}_1),\ldots,\rho(\boldsymbol{\tau}_n)) = \boldsymbol{\tau}'$ implies that $\boldsymbol{\tau}' = f(\boldsymbol{\tau}'_1,\ldots,\boldsymbol{\tau}'_n)$ with $\boldsymbol{\tau}'_1 = \rho(\boldsymbol{\tau}_i)$, $i \in [1,n]$. It follows, by (6) and ind. hyp., that $\boldsymbol{\varrho}'(\boldsymbol{\tau}') = \boldsymbol{\varrho}'(f(\boldsymbol{\tau}'_1,\ldots,\boldsymbol{\tau}'_n)) = f(\boldsymbol{\varrho}'(\boldsymbol{\tau}'_1),\ldots,\boldsymbol{\varrho}'(\boldsymbol{\tau}'_n)) = f(\boldsymbol{\varrho}(\boldsymbol{\tau}_1),\ldots,\boldsymbol{\varrho}(\boldsymbol{\tau}_n)) = \boldsymbol{\varrho}(f(\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_n)) = \boldsymbol{\varrho}(\boldsymbol{\tau})$. $\square$

The comparison of equivalence classes is equivalent to the comparison of the representatives of these classes.

Lemma 5. $[\boldsymbol{\tau}_1]_{\simeq^{\nu}} \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}_2]_{\simeq^{\nu}} \Leftrightarrow \boldsymbol{\tau}_1 \preceq^{\nu} \boldsymbol{\tau}_2$.

Proof of lemma 5.

$$
\begin{array}{ll}
 & [\boldsymbol{\tau}_1]_{\simeq^{\nu}} \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}_2]_{\simeq^{\nu}} \\
\Leftrightarrow & \exists \boldsymbol{\tau}'_1 \in [\boldsymbol{\tau}_1]_{\simeq^{\nu}}, \boldsymbol{\tau}'_2 \in [\boldsymbol{\tau}_2]_{\simeq^{\nu}} \,.\, \boldsymbol{\tau}'_1 \preceq^{\nu} \boldsymbol{\tau}'_2 \quad \wr\text{def. (10) of } \preceq_{\simeq^{\nu}}\wr \\
\Leftrightarrow & \exists \boldsymbol{\tau}'_1, \boldsymbol{\tau}'_2 \,.\, \boldsymbol{\tau}'_1 \simeq^{\nu} \boldsymbol{\tau}_1 \wedge \boldsymbol{\tau}'_2 \simeq^{\nu} \boldsymbol{\tau}_2 \wedge \boldsymbol{\tau}'_1 \preceq^{\nu} \boldsymbol{\tau}'_2 \quad \wr\text{def. } [\boldsymbol{\tau}]_{\simeq^{\nu}}\wr \\
\Leftrightarrow & \exists \boldsymbol{\tau}'_1, \boldsymbol{\tau}'_2 \,.\, \boldsymbol{\tau}_1 \preceq^{\nu} \boldsymbol{\tau}'_1 \wedge \boldsymbol{\tau}'_1 \preceq^{\nu} \boldsymbol{\tau}'_2 \wedge \boldsymbol{\tau}'_2 \preceq^{\nu} \boldsymbol{\tau}_2 \wedge \boldsymbol{\tau}'_1 \preceq^{\nu} \boldsymbol{\tau}_1 \wedge \boldsymbol{\tau}_2 \preceq^{\nu} \boldsymbol{\tau}'_2 \quad \wr\text{def. } \simeq^{\nu}\wr \\
\Leftrightarrow & \boldsymbol{\tau}_1 \preceq^{\nu} \boldsymbol{\tau}_2 \\
 & \quad \wr(\Rightarrow) \ \text{transitivity} \\
 & \quad \ (\Leftarrow) \ \text{choosing } \boldsymbol{\tau}'_1 = \boldsymbol{\tau}_1, \boldsymbol{\tau}'_2 = \boldsymbol{\tau}_2, \text{and reflexivity}\wr \quad \square
\end{array}
$$

B. The symbolic abstraction function

The abstraction of $\{f(a,a), f(b,b), f(c,c)\}$ is $f(\alpha,\alpha)$ since the parameters of f are equal while $\{f(a,b), f(b,a), f(a,a)\}$ is $f(\beta,\gamma)$ since the parameters of f are not always related. The abstraction function must select variables so as to identify equal parameters on all instances of f. For this purpose, we encode sets as families, for example, sequences $\langle f(a,a),\ f(b,b),\ f(c,c)\rangle$ and $\langle f(a,b),\ f(b,a),\ f(a,a)\rangle$. In the first case, the subterms all yield $\langle a,\ b,\ c\rangle$ which is abstracted by a variable say α. In the second case we get $\langle a,\ b,\ a\rangle$ encoded by β and $\langle b,\ a,\ a\rangle$ which is different so is encoded by a different variable γ. Notice that the variable name does not matter and that the order in the sequences does not matter either (so sets of ground terms encoded differently as index families will have the same abstraction, up to variable renaming via a bijection between variables; see lemma 6).

We arbitrarily define a scheme to name sets of ground terms by a unique variable thanks to an injective function

$$
\boldsymbol{\nu} \in (\Delta \to \mathsf{T}) \rightarrowtail \mathbb{V}_{\mathfrak{t}} \qquad \text{(naming scheme)} \tag{11}
$$

assigning a variable $\boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta\})$ to any arbitrary family of ground terms $\{\mathbf{t}_i \mid i \in \Delta\}$. Injectivity ensures uniqueness, that is, different families of terms are abstracted by different variables.

The abstraction is called the *least common generalization* (*lcg*).

$$
\begin{array}{l}
\mathit{lcg}[\boldsymbol{\nu}](\emptyset) \ \triangleq \ \overline{\varnothing}^{\nu} \\
\mathit{lcg}[\boldsymbol{\nu}](\{f_i(\mathbf{t}_i^1,\ldots,\mathbf{t}_i^{n_i}) \mid i \in \Delta\}) \ \triangleq \\
\quad \text{if } \forall i,j \in \Delta \,.\, f_i = f_j = f \wedge n_i = n_j = n \ \mathsf{then} \\
\quad\quad \mathsf{let}\ T^k = \mathit{lcg}[\boldsymbol{\nu}](\{\mathbf{t}_i^k \mid i \in \Delta\}), k = 1,\ldots,n \ \mathsf{in} \\
\quad\quad\quad f(T^1,\ldots,T^n) \\
\quad \mathsf{else}\ \boldsymbol{\nu}(\{f_i(\mathbf{t}^1,\ldots,\mathbf{t}^{n_i}) \mid i \in \Delta\})
\end{array} \tag{12}
$$

If all the terms in the family have the same structure then the abstraction proceeds recursively else the family is abstracted by a variable. Equalities between all subterms of the family are preserved by the abstraction since the families of these subterms are abstracted by the same variable when they have different structures.

Example 2. *Assume that* $\boldsymbol{\nu}(\langle a, b\rangle) = \alpha$ *and* $\boldsymbol{\nu}(\langle b, a\rangle) = \beta$, *then*

$$\begin{aligned}
& lcg[\boldsymbol{\nu}](\langle f(g(a,a), h(b,b), a, b),\ f(g(b,b), h(a,a), b, a)\rangle) \\
= \ & f(lcg[\boldsymbol{\nu}](\langle g(a,a), g(b,b)\rangle), lcg[\boldsymbol{\nu}](\langle h(b,b), h(a,a)\rangle), lcg[\boldsymbol{\nu}](\langle a,\ b\rangle), lcg[\boldsymbol{\nu}](\langle b,\ a\rangle)) \\
= \ & f(g(lcg[\boldsymbol{\nu}](\langle a, b\rangle), lcg[\boldsymbol{\nu}](\langle a, b\rangle)), h(lcg[\boldsymbol{\nu}](\langle b, a\rangle), lcg[\boldsymbol{\nu}](\langle b,\ a\rangle)), lcg[\boldsymbol{\nu}](\langle a,\ b\rangle), lcg[\boldsymbol{\nu}](\langle b,\ a\rangle)) \\
= \ & f(g(\boldsymbol{\nu}(\langle a,\ b\rangle), \boldsymbol{\nu}(\langle a,\ b\rangle)), h(\boldsymbol{\nu}(\langle b,\ a\rangle), \boldsymbol{\nu}(\langle b,\ a\rangle)), \boldsymbol{\nu}(\langle a, b\rangle), \boldsymbol{\nu}(\langle b,\ a\rangle)) \\
= \ & f(g(\alpha,\alpha)), h(\beta,\beta), \alpha, \beta) \qquad \square
\end{aligned}$$

Lemma 6. *The definition* (12) *of the symbolic abstraction* $lcg[\boldsymbol{\nu}]$ *is independent of the naming scheme* $\boldsymbol{\nu}$*. If* $\boldsymbol{\nu}, \boldsymbol{\nu}' \in (\Delta \to \mathbf{T}) \rightarrowtail \mathbb{V}_{\mathbf{t}}$ *then* $\forall T \in \wp(\mathbf{T}) \ .\ lcg[\boldsymbol{\nu}](T) \simeq^{\nu} lcg[\boldsymbol{\nu}'](T)$.

Proof of lemma 6. Since $\boldsymbol{\nu} \in (\Delta \to \mathbf{T}) \rightarrowtail \mathbb{V}_{\mathbf{t}}$ is injective, it has a left inverse (improperly) denoted $\boldsymbol{\nu}^{-1}$ such that $\boldsymbol{\nu}^{-1} \circ \boldsymbol{\nu} = \mathbb{1}_{\wp(\mathbf{T})}$ is the identity on $\wp(\mathbf{T})$ (encoded as families $\Delta \to \mathbf{T}$). Define $\rho \triangleq \boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1}$. Given $T \in \wp(\mathbf{T})$, let us show that $\rho(lcg[\boldsymbol{\nu}](T)) = lcg[\boldsymbol{\nu}'](T)$, by structural induction and case analysis on the def. (12) of $lcg[\boldsymbol{\nu}]$.

1) If $T = \emptyset$ then $\rho(lcg[\boldsymbol{\nu}](T)) = \rho(lcg[\boldsymbol{\nu}](\emptyset)) = \overline{\varnothing}^{\nu} = lcg[\boldsymbol{\nu}'](\emptyset) = lcg[\boldsymbol{\nu}'](T)$;
2) Else, if $lcg[\boldsymbol{\nu}](T) = \boldsymbol{\nu}(T)$ then $lcg[\boldsymbol{\nu}'](T) = \boldsymbol{\nu}'(T)$ so that $\rho(lcg[\boldsymbol{\nu}](T)) = \rho(\boldsymbol{\nu}(T)) = \boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1} \circ \boldsymbol{\nu}(T) = \boldsymbol{\nu}' \circ \mathbb{1}_{\wp(\mathbf{T})}(T) = \boldsymbol{\nu}'(T) = lcg[\boldsymbol{\nu}'](T)$;
3) Otherwise, $T = \{f(\mathbf{t}_i^1, \ldots, \mathbf{t}_i^n) \mid i \in \Delta\}$, so that by ind. hyp., $\rho(T^k_{\boldsymbol{\nu}}) = \rho(lcg[\boldsymbol{\nu}](\{\mathbf{t}^k \mid i \in \Delta\})) = lcg[\boldsymbol{\nu}'](\{\mathbf{t}^k \mid i \in \Delta\}) = T^k_{\boldsymbol{\nu}'}$, $k = 1, \ldots, n$. Therefore, by (6), $\rho(lcg[\boldsymbol{\nu}](T)) = \rho(f(T^1_{\boldsymbol{\nu}}, \ldots, T^n_{\boldsymbol{\nu}})) = f(\rho(T^1_{\boldsymbol{\nu}}), \ldots, \rho(T^n_{\boldsymbol{\nu}})) = f(T^1_{\boldsymbol{\nu}'}, \ldots, T^n_{\boldsymbol{\nu}'}) = lcg[\boldsymbol{\nu}'](T)$.

If $\alpha \in vars[\![lcg[\boldsymbol{\nu}'](T)]\!]$ then case 1. of the above proof shows that $\alpha = lcg[\boldsymbol{\nu}'](T') = \boldsymbol{\nu}'(T')$ for some $T' \in \wp(\mathbf{T})$ and therefore $\rho(lcg[\boldsymbol{\nu}](T')) = \rho(\boldsymbol{\nu}(T')) = \boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1} \circ \boldsymbol{\nu}(T') = \boldsymbol{\nu}' \circ \mathbb{1}_{\wp(\mathbf{T})}(T') = \boldsymbol{\nu}'(T') = \alpha$, proving that $\rho \in vars[\![lcg[\boldsymbol{\nu}](T)]\!] \twoheadrightarrow vars[\![lcg[\boldsymbol{\nu}'](T)]\!]$ is surjective.

If $\alpha_1, \alpha_2 \in vars[\![lcg[\boldsymbol{\nu}](T)]\!]$ then case 1. of the above proof shows that $\alpha_1 = lcg[\boldsymbol{\nu}](\alpha_1) = \boldsymbol{\nu}(\alpha_1)$ and $\alpha_2 = lcg[\boldsymbol{\nu}](\alpha_2) = \boldsymbol{\nu}(\alpha_2)$ for some $\alpha_1, \alpha_2 \in \wp(\mathbf{T})$. Assume that $\rho(\alpha_1) = \rho(\alpha_2)$. Then we have $\boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1}(\alpha_1) = \boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1}(\alpha_2)$ that is $\boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1}(\boldsymbol{\nu}(\alpha_1)) = \boldsymbol{\nu}' \circ \boldsymbol{\nu}^{-1}(\boldsymbol{\nu}(\alpha_2))$, which implies $\boldsymbol{\nu}'(\alpha_1) = \boldsymbol{\nu}'(\alpha_2)$ since $\boldsymbol{\nu}^{-1} \circ \boldsymbol{\nu} = \mathbb{1}_{\wp(\mathbf{T})}$. It follows that $\alpha_1 = \alpha_2$ since $\boldsymbol{\nu}'$ is injective. Therefore $\alpha_1 = \boldsymbol{\nu}(\alpha_1) = \boldsymbol{\nu}(\alpha_2) = \alpha_2$, proving that ρ is injective.

It follows that $\rho \in vars[\![lcg[\boldsymbol{\nu}](T)]\!] \rightarrowtail vars[\![lcg[\boldsymbol{\nu}'](T)]\!]$ is bijective so that $lcg[\boldsymbol{\nu}](T) \simeq^{\nu} lcg[\boldsymbol{\nu}'](T)$ by lemma 4. $\square$

We now want to identify a Galois connection with abstraction $lcg[\boldsymbol{\nu}]$ and concretization *ground*. Several preliminary results are needed. First, the symbolic abstraction $lcg[\boldsymbol{\nu}]$ is $\preceq^{\nu}$-increasing.

Lemma 7. *Let* $\Delta \subseteq \Delta'$ *be index sets and* $\mathbf{t} \in \Delta' \to \mathbf{T}$ *(and therefore* $\{\mathbf{t}_i \mid i \in \Delta\} \subseteq \{\mathbf{t}_i \mid i \in \Delta'\}$*). Then* $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \preceq^{\nu} lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta'\})$.

Proof of lemma 7. By lemma 3, we must prove that $\forall \boldsymbol{\varrho} \in \mathbf{P}^{\nu} \ .\ \exists \boldsymbol{\varrho}' \in \mathbf{P}^{\nu} \ .\ \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta'\}))$. Given any $\boldsymbol{\varrho}$, let us define $\boldsymbol{\varrho}'$ such that, for any $\{\mathbf{t}_i \mid i \in \Delta'\} \in \Delta' \to \mathbf{T}$, we have (including the case of constants when $n = 0$ or $m = 0$)

- If $\{\mathbf{t}_i \mid i \in \Delta\} = \{f(\mathbf{t}_i^1, \ldots, \mathbf{t}_i^n) \mid i \in \Delta\}$ and $\exists j \in \Delta' \setminus \Delta \ .\ \mathbf{t}_j = g(\mathbf{t}'^1_j, \ldots, \mathbf{t}'^m_j)$ with $g \neq f$ then
$$\boldsymbol{\varrho}'(\boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta'\})) \quad \triangleq \quad \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$$
- Otherwise, $\{\mathbf{t}_i \mid i \in \Delta'\} = \{f(\mathbf{t}_i^1, \ldots, \mathbf{t}_i^n) \mid i \in \Delta'\}$, in which case
$$\boldsymbol{\varrho}'(\boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta'\})) \quad \triangleq \quad \boldsymbol{\varrho}(\boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta\}))$$

Let us show that $\boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta'\}))$, by structural induction on $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})$. There are two cases.

- If $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})$ is a variable $\alpha = \boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta\})$ then $\exists j, k \in \Delta \ .\ \mathbf{t}_j = f(\mathbf{t}_j^1, \ldots, \mathbf{t}_j^{n_j}) \wedge \mathbf{t}_k = g(\mathbf{t}_k^1, \ldots, \mathbf{t}_k^{n_k}) \wedge f \neq g$. By (12), since $\mathbf{t}_j, \mathbf{t}_k \in \{\mathbf{t}_i \mid i \in \Delta'\}$, $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta'\})$ is a variable $\beta = \boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta'\})$. By our definition of $\boldsymbol{\varrho}'$, we have $\boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) = \boldsymbol{\varrho}(\alpha) = \boldsymbol{\varrho}'(\beta) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta'\})))$;
- Otherwise $\{\mathbf{t}_i \mid i \in \Delta'\} = \{f(\mathbf{t}_i^1, \ldots, \mathbf{t}_i^n) \mid i \in \Delta'\}$ and then

$$\begin{aligned}
& \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) \\
= \ & \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{f(\mathbf{t}_i^1, \ldots, \mathbf{t}_i^n) \mid i \in \Delta\})) && \wr \Delta \subseteq \Delta' \int \\
= \ & \boldsymbol{\varrho}(f(lcg[\boldsymbol{\nu}](\{\mathbf{t}_1^i \mid i \in \Delta\}), \ldots, lcg[\boldsymbol{\nu}](\{\mathbf{t}_n^i \mid i \in \Delta\}))) && \wr \text{def. (12) of } lcg[\boldsymbol{\nu}] \int \\
= \ & f(\boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_1^i \mid i \in \Delta\})), \ldots, \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_n^i \mid i \in \Delta\})) && \wr (6) \int \\
= \ & f(\boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_1^i \mid i \in \Delta'\})), \ldots, \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_n^i \mid i \in \Delta'\})) && \wr \text{ind. hyp.} \int \\
= \ & \boldsymbol{\varrho}'(f(lcg[\boldsymbol{\nu}](\langle \mathbf{t}_1^i,\ i \in \Delta'\rangle)), \ldots, lcg[\boldsymbol{\nu}](\{\mathbf{t}_n^i \mid i \in \Delta'\})) && \wr (6) \int \\
= \ & \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\langle \mathbf{t}_i,\ i \in \Delta'\rangle)) && \wr \text{def. (12) of } lcg[\boldsymbol{\nu}] \int \quad \square
\end{aligned}$$

Let us prove that the abstraction of a set of terms over approximates any term of the set.

Lemma 8. *Let* Δ *be a nonempty set and* $\mathbf{t} \in \Delta \to \mathbf{T}$ *be a family of terms. Then* $\forall j \in \Delta \ .\ \mathbf{t}_j \preceq^{\nu} lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})$ *that is* $\forall j \in \Delta \ .\ \exists \boldsymbol{\varrho}' \in \mathbf{P}^{\nu} \ .\ \mathbf{t}_j = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$.

Proof of lemma 8. By lemma 3, we must prove that $\forall j \in \Delta \ .\ \forall \boldsymbol{\varrho} \in \mathbf{P}^{\nu} \ .\ \exists \boldsymbol{\varrho}' \in \mathbf{P}^{\nu} \ .\ \boldsymbol{\varrho}(\mathbf{t}_j) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$, By $\mathbf{t}_j \in \mathbf{T}$ has no variable so $\forall \boldsymbol{\varrho} \in \mathbf{P}^{\nu} \ .\ \boldsymbol{\varrho}(\mathbf{t}_j) = \mathbf{t}_j$. It follows that we have to prove that $\forall j \in \Delta \ .\ \exists \boldsymbol{\varrho}' \in \mathbf{P}^{\nu} \ .\ \mathbf{t}_j = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$.

For any element of j be of Δ, let us define $\boldsymbol{\varrho}'(\beta) \triangleq \mathbf{t}'_j$ for all $\beta = \boldsymbol{\nu}(\{\mathbf{t}'_i \mid i \in \Delta\})$ and $\{\mathbf{t}'_i \mid i \in \Delta\} \in \Delta \to \mathbf{T}$.

The proof that $\mathbf{t}_j = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$ is by structural induction on $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \in \mathbf{T}^{\nu}$.

- If $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})$ is a ground term $a \in \mathbf{F}_0$ then, by (12), $\forall i \in \Delta \;.\; \mathbf{t}_i = a$ so $\mathbf{t}_j = a = \boldsymbol{\varrho}'(a) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$.
- If $lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})$ is a variable $\alpha = \boldsymbol{\nu}(\{\mathbf{t}_i \mid i \in \Delta\})$ then, by (12), there exist $i, k \in \Delta$ such that $\mathbf{t}_i = f(\mathbf{t}_{i,1}, \ldots, \mathbf{t}_{i,n})$ and $\mathbf{t}_k = g(\mathbf{t}_{k,1}, \ldots, \mathbf{t}_{k,m})$ with $f \neq g$. Then, by def. $\boldsymbol{\varrho}'$, $\boldsymbol{\varrho}'(\alpha) = \mathbf{t}_j$.
- Otherwise, by (12), we have $\forall i \in \Delta \;.\; \mathbf{t}_i = f(\mathbf{t}_{i,1}, \ldots, \mathbf{t}_{i,n})$, so by structural induction hypothesis, $\forall \ell \in [1, n] \;.\; \mathbf{t}_{j,\ell} = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,\ell} \mid i \in \Delta\}))$. Therefore $\mathbf{t}_j = f(\mathbf{t}_{j,1}, \ldots, \mathbf{t}_{j,n}) = f(\boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,1} \mid i \in \Delta\})), \ldots, \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,n} \mid i \in \Delta\}))) = \boldsymbol{\varrho}'(f(lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,1} \mid i \in \Delta\})), \ldots, lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,n} \mid i \in \Delta\})) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{f(\mathbf{t}_{i,1}, \ldots, \mathbf{t}_{i,n}) \mid i \in \Delta\})) = \boldsymbol{\varrho}'(lcg[\boldsymbol{\nu}](\{\mathbf{t}_{i,1} \mid i \in \Delta\}))$. □

The following corollary shows that the symbolic abstraction is an over approximation of properties of ground terms, that is, $ground \circ lcg[\boldsymbol{\nu}]()$ is extensive.

Corollary 1. *If Δ is a nonempty set and $\{\mathbf{t}_i \mid i \in \Delta\} \in \Delta \to \mathbf{T}$, then $\{\mathbf{t}_i \mid i \in \Delta\} \subseteq ground(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}))$.*

Proof of corollary 1.

$$\begin{aligned}
& \{\mathbf{t}_i \mid i \in \Delta\} \subseteq ground(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) \\
\Leftrightarrow\ & \{\mathbf{t}_i \mid i \in \Delta\} \subseteq \{\boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) \mid \boldsymbol{\varrho} \in \mathbf{P}^{\nu}\} \\
& \hfill \wr\text{def. (9) of } ground\wr \\
\Leftrightarrow\ & \forall j \in \Delta \;.\; \exists \boldsymbol{\varrho} \in \mathbf{P}^{\nu} \;.\; \mathbf{t}_j = \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})) \ \wr\text{def. } \subseteq\wr
\end{aligned}$$

which is true by lemma 8. □

The following corollary shows that the abstraction of the concretization of a term with variables looses no information.

Corollary 2. *For all $\boldsymbol{\tau} \in \mathbf{T}^{\nu} \;.\; ground \circ lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau}) = ground(\boldsymbol{\tau})$.*

Proof of corollary 2. By corollary 1 (where $ground(\boldsymbol{\tau}) = \{\mathbf{t}_i \mid i \in \Delta\}$), $ground(\boldsymbol{\tau}) \subseteq ground \circ lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau})$. It remains to prove that

$$\begin{aligned}
& ground \circ lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau}) \subseteq ground(\boldsymbol{\tau}) \\
\Leftrightarrow\ & \{\boldsymbol{\varrho}(lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau})) \mid \boldsymbol{\varrho} \in \mathbf{P}^{\nu}\} \subseteq \{\boldsymbol{\varrho}'(\boldsymbol{\tau}) \mid \boldsymbol{\varrho}' \in \mathbf{P}^{\nu}\} \\
& \hfill \wr\text{def. (8) of } ground\wr \\
\Leftrightarrow\ & \forall \boldsymbol{\varrho} \in \mathbf{P}^{\nu} \;.\; \exists \boldsymbol{\varrho}' \in \mathbf{P}^{\nu} \;.\; \boldsymbol{\varrho}(lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau})) = \boldsymbol{\varrho}'(\boldsymbol{\tau}) \\
& \hfill \wr\text{def. } \subseteq\wr
\end{aligned}$$

which holds by lemma 8. □

C. The symbolic term Galois connection

In order to take into account the equivalence of terms with variables up to variable renaming (see lemma 4), we reason on the quotient partial order of terms $\langle \mathcal{P}^{\mathrm{H}}, \preceq_{\simeq^{\nu}} \rangle$. We extend the concretization (8) and the abstraction (12) to equivalence classes as follows.

$$\begin{aligned}
lcg_{\simeq^{\nu}}[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) &\triangleq [lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})]_{\simeq^{\nu}} \quad (13)\\
ground_{\simeq^{\nu}}([\boldsymbol{\tau}]_{\simeq^{\nu}}) &\triangleq ground(\boldsymbol{\tau}).
\end{aligned}$$

Theorem 1. *For any naming scheme $\boldsymbol{\nu} \in (\Delta \to \mathbf{T}) \rightarrowtail \mathbb{V}_{\mathbf{t}}$,*

$$\langle \wp(\mathbf{T}), \subseteq \rangle \underset{lcg_{\simeq^{\nu}}[\boldsymbol{\nu}]}{\overset{ground_{\simeq^{\nu}}}{\leftarrows\!\!\!\twoheadrightarrow}} \langle \mathcal{P}^{\mathrm{H}}, \preceq_{\simeq^{\nu}} \rangle \qquad (14)$$

This definition of the Galois retraction is independent of the choice of the naming scheme $\boldsymbol{\nu}$. □

Proof of theorem 1. By def. of a Galois connection, we must prove that for all families of terms $\{\mathbf{t}_i \mid i \in \Delta\} \in \wp(\mathbf{T})$ and term with variables $\boldsymbol{\tau}' \in \mathbf{T}^{\nu} \cup \{\overline{\varnothing}^{\nu}\}$,

$$\begin{aligned}
& lcg_{\simeq^{\nu}}[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}']_{\simeq^{\nu}} \quad \Leftrightarrow \\
& \qquad \{\mathbf{t}_i \mid i \in \Delta\} \subseteq ground_{\simeq^{\nu}}([\boldsymbol{\tau}']_{\simeq^{\nu}}).
\end{aligned}$$

This is obvious for $\Delta = \emptyset$ since $lcg_{\simeq^{\nu}}[\boldsymbol{\nu}](\emptyset) = \overline{\varnothing}^{\nu}$ which is the infimum. Otherwise, we have

$$\begin{aligned}
& lcg_{\simeq^{\nu}}[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}']_{\simeq^{\nu}} \\
\Leftrightarrow\ & [lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\})]_{\simeq^{\nu}} \preceq_{\simeq^{\nu}} [\boldsymbol{\tau}']_{\simeq^{\nu}} \\
& \hfill \wr\text{def. (13) of } lcg_{\simeq^{\nu}}[\boldsymbol{\nu}]\wr \\
\Leftrightarrow\ & lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \preceq^{\nu} \boldsymbol{\tau}' \quad \wr\text{def. (10) of } \preceq_{\simeq^{\nu}} \text{ and } \simeq^{\nu}\wr \\
\Leftrightarrow\ & ground \circ lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \subseteq ground(\boldsymbol{\tau}') \\
& \hfill \wr\text{def. (9) of } \preceq^{\nu}\wr \\
\Leftrightarrow\ & \{\mathbf{t}_i \mid i \in \Delta\} \subseteq ground(\boldsymbol{\tau}')
\end{aligned}$$

$\wr$($\Rightarrow$) by corollary 1 and transitivity;
($\Leftarrow$) By lemma 7, $lcg[\boldsymbol{\nu}]$ is increasing. By def. (9) of $\preceq^{\nu}$, $ground$ is increasing. Their composition is increasing so $ground \circ lcg[\boldsymbol{\nu}](\{\mathbf{t}_i \mid i \in \Delta\}) \subseteq ground \circ lcg[\boldsymbol{\nu}] \circ ground(\boldsymbol{\tau}') = ground(\boldsymbol{\tau}')$ by corollary 2. $\wr$

$$\begin{aligned}
\Leftrightarrow\ & \{\mathbf{t}_i \mid i \in \Delta\} \subseteq ground_{\simeq^{\nu}}([\boldsymbol{\tau}']_{\simeq^{\nu}}) \\
& \hfill \wr\text{def. (13)) of } ground_{\simeq^{\nu}}\wr
\end{aligned}$$

Moreover $ground_{\simeq^{\nu}}$ is injective so (14) is a Galois retraction (also called Galois insertion).

By lemma 6, if $\boldsymbol{\nu}, \boldsymbol{\nu}' \in (\Delta \to \mathbf{T}) \rightarrowtail \mathbb{V}_{\mathbf{t}}$ then $\forall T \in \wp(\mathbf{T}) \;.\; lcg[\boldsymbol{\nu}](T) \simeq^{\nu} lcg[\boldsymbol{\nu}'](T)$ so that this definition of the Galois retraction (14) is independent of the choice of the naming scheme $\boldsymbol{\nu}$. □

In a Galois connection $\langle C, \leqslant \rangle \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\!\twoheadrightarrow}} \langle A, \sqsubseteq \rangle$, $\alpha = \alpha \circ \gamma \circ \alpha$. It follows immediately that $\langle ground_{\simeq^{\nu}}(\mathcal{P}^{\mathrm{H}}), \subseteq \rangle \underset{lcg_{\simeq^{\nu}}[\boldsymbol{\nu}]}{\overset{ground_{\simeq^{\nu}}}{\twoheadleftarrow\!\!\!\twoheadrightarrow}} \langle \mathcal{P}^{\mathrm{H}}, \preceq_{\simeq^{\nu}} \rangle$ is a Galois isomorphism, an essential remark for completeness in typing [22].

D. The symbolic abstract domain is a complete lattice

By [26, Theorem 4.1], the image of a complete lattice by an upper closure operator is a complete lattice. This extends to a Galois retraction $\langle C, \leqslant \rangle \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\!\twoheadrightarrow}} \langle A, \sqsubseteq \rangle$ since $\gamma \circ \alpha$ is an upper closure operator, α is surjective so that $\gamma(A)$ and $\alpha(C)$ are isomorphic, $\alpha = \alpha \circ \gamma \circ \alpha$, and α preserves existing joins. Therefore, the terms with variables form a complete lattice since they are the image of the complete lattice of properties of ground terms by the Galois retraction (14).

Corollary 3 (symbolic abstract domain)**.** *For any naming scheme* $\boldsymbol{\nu} \in (\Delta \to \mathsf{T}) \rightarrowtail \mathbb{V}_{\mathfrak{t}}$*,* $\langle \mathcal{P}^{\mathrm{H}},\ \preceq_{\simeq^\nu},\ [\overline{\varnothing}^\nu]_{\simeq^\nu},\ [\alpha]_{\simeq^\nu},\ \mathit{LCG}_{\simeq^\nu},\ \mathit{GCI}_{\simeq^\nu}\rangle$ *is a complete lattice where* $\alpha \in \mathbb{V}_{\mathfrak{t}}$*, the least upper bound is* $\mathit{LCG}_{\simeq^\nu}(S) \triangleq \mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\bigcup \mathit{ground}_{\simeq^\nu}(S))$ *(binary* lcg *for symbolic terms and* $\mathit{lcg}_{\simeq^\nu}$ *for term classes), and the greatest lower bound is* $\mathit{GCI}_{\simeq^\nu}(S) \triangleq \mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\bigcap \mathit{ground}_{\simeq^\nu}(S))$ *(binary* gci *and* $\mathit{gci}_{\simeq^\nu}$*). This characterization of the lattice operations is independent of the naming scheme* $\boldsymbol{\nu}$ *which is used.*

Proof of corollary 3. Since $\langle \wp(\mathsf{T}),\ \subseteq,\ \emptyset,\ \mathsf{T},\ \bigcup,\ \bigcap\rangle$ is a complete lattice and (14) is a Galois retraction, it follows that, for any naming scheme $\boldsymbol{\nu} \in (\Delta \to \mathsf{T}) \rightarrowtail \mathbb{V}_{\mathfrak{t}}$, its image $\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\wp(\mathsf{T})) = \mathcal{P}^{\mathrm{H}}$ by $\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}]$ is also a complete lattice $\langle \mathcal{P}^{\mathrm{H}},\ \preceq_{\simeq^\nu},\ [\overline{\varnothing}^\nu]_{\simeq^\nu},\ [\alpha]_{\simeq^\nu},\ \mathit{LCG}_{\simeq^\nu},\ \mathit{GCI}_{\simeq^\nu}\rangle$ where the infimum is $\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\emptyset) = [\overline{\varnothing}^\nu]_{\simeq^\nu}$, the supremum is $\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\mathsf{T}) = [\boldsymbol{\nu}(\langle \mathsf{t},\ \mathsf{t} \in \mathsf{T}\rangle)]_{\simeq^\nu} \simeq^\nu [\alpha]_{\simeq^\nu}$, $\alpha \in \mathbb{V}_{\mathfrak{t}}$, the least upper bound is $\mathit{LCG}_{\simeq^\nu}(S) \triangleq \mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\bigcup \mathit{ground}_{\simeq^\nu}(S))$ and the greatest lower bound is $\mathit{GCI}_{\simeq^\nu}(S) \triangleq \mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\bigcap \mathit{ground}_{\simeq^\nu}(S))$.

By lemma 6, if $\boldsymbol{\nu}, \boldsymbol{\nu}' \in (\Delta \to \mathsf{T}) \rightarrowtail \mathbb{V}_{\mathfrak{t}}$ then $\forall T \in \wp(\mathsf{T})\ .\ \mathit{lcg}[\boldsymbol{\nu}](T) \simeq^\nu \mathit{lcg}[\boldsymbol{\nu}'](T)$ so that this characterization of the lattice operations is independent of the choice of the naming scheme $\boldsymbol{\nu}$. □

We use $\mathit{lcg}_{\simeq^\nu}$ (respectively $\mathit{gci}_{\simeq^\nu}$) for the binary version of $\mathit{LCG}_{\simeq^\nu}$ (respect. $\mathit{GCI}_{\simeq^\nu}$).

Observe that ground terms $[\mathsf{t}]_{\simeq^\nu} \in \mathcal{P}^{\mathrm{H}}$ belongs to the abstract domain and abstract the concrete property $\{\mathsf{t}\}$ of being that ground term. Then $\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\{\mathsf{t}\}) = \mathit{LCG}_{\simeq^\nu}(\{[\mathsf{t}]_{\simeq^\nu}\})$, because, by (14), we have

$$\begin{aligned}
&\mathit{LCG}_{\simeq^\nu}(\{[\mathsf{t}]_{\simeq^\nu}\})\\
\triangleq\ &\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\textstyle\bigcup \mathit{ground}_{\simeq^\nu}(\{[\mathsf{t}]_{\simeq^\nu}\}))\\
=\ &\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\textstyle\bigcup \mathit{ground}_{\simeq^\nu}(\{[\mathsf{t}]_{\simeq^\nu}\}))\\
=\ &\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\textstyle\bigcup \mathit{ground}_{\simeq^\nu}(\{\mathsf{t}\}))\\
=\ &\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\textstyle\bigcup \{\mathsf{t}\})\\
=\ &\mathit{lcg}_{\simeq^\nu}[\boldsymbol{\nu}](\mathsf{t}).
\end{aligned}$$

This explains why the abstraction and the lub in the complete lattice have been given the same name.

VII. The classical definition of the subsumption partial order using substitutions

The subsumption preorder $\preceq^\nu$ is classically defined syntactically, using substitutions [7, pp. 180–188] (instead of (9)) [3]–[5]. We show that this classical syntactic definition is equivalent to the semantic definition (9) based on the interpretation of terms with variables as properties of ground terms.

A. Substitutions

The same way that assignments (5) record ground values of variables, we use substitutions to record symbolic values of some variables, so substitutions are partial functions

$$\vartheta \ \in\ \Sigma \ \triangleq\ \mathbb{V}_{\mathfrak{t}} \nrightarrow \mathsf{T}^\nu \tag{15}$$

mapping variables α in its domain $\mathsf{dom}(\vartheta)$ to terms with variables $\vartheta(\alpha)$.

A substitution is extended to a total function $\vartheta \in \mathbb{V}_{\mathfrak{t}} \to \mathsf{T}^\nu$ and homomorphically to terms with variables, as follows

$$\begin{aligned}
\vartheta(\alpha) &\ \triangleq\ \alpha \qquad \text{when } \alpha \notin \mathsf{dom}(\vartheta) \qquad (16)\\
\vartheta(f(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n)) &\ \triangleq\ f(\vartheta(\boldsymbol{\tau}_1), \ldots, \vartheta(\boldsymbol{\tau}_n))
\end{aligned}$$

Observe that the substitution is carried out simultaneously on all variable occurrences.

The empty substitution $\boldsymbol{\varepsilon}$ is totally undefined, that is $\mathsf{dom}(\boldsymbol{\varepsilon}) = \emptyset$. It's total extension is the identity $\forall \alpha \in \mathbb{V}_{\mathfrak{t}}\ .\ \boldsymbol{\varepsilon}(\alpha) = \alpha$. By structural induction on terms with variables, we have $\forall \boldsymbol{\tau} \in \mathsf{T}^\nu\ .\ \boldsymbol{\varepsilon}(\boldsymbol{\tau}) = \boldsymbol{\tau}$.

B. The classical characterization of the subsumption preorder using substitutions

The following theorem 2 shows that the syntactic and semantic definitions of subsumption are equivalent. It follows that the subsumption lattice of [3]–[5], [27] is the complete lattice considered in corollary 3 since the partial order is the same (although defined differently).

Theorem 2. $\forall \boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \mathsf{T}^\nu\ .\ [\boldsymbol{\tau}_1]_{\simeq^\nu} \preceq_{\simeq^\nu} [\boldsymbol{\tau}_2]_{\simeq^\nu} \Leftrightarrow \exists \vartheta \in \Sigma\ .\ \vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1$*.*

Proof of theorem 2. Let us first show that for all $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \mathsf{T}^\nu$,

$$(\forall \boldsymbol{\varrho} \in \mathsf{P}^\nu\ .\ \exists \boldsymbol{\varrho}' \in \mathsf{P}^\nu\ .\ \boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)) \Leftrightarrow \tag{17}$$
$$(\exists \vartheta \in \Sigma\ .\ \vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1) \tag{18}$$

($\Leftarrow$) Choose $\boldsymbol{\varrho}' = \boldsymbol{\lambda}\, \alpha \centerdot \boldsymbol{\varrho}(\vartheta(\alpha))$ so that, by structural induction on terms with variables, $\forall \boldsymbol{\tau} \in \mathsf{T}^\nu\ .\ \boldsymbol{\varrho}'(\boldsymbol{\tau}) = \boldsymbol{\varrho}(\vartheta(\boldsymbol{\tau}))$. Then $\vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1$ implies $\boldsymbol{\varrho}(\vartheta(\boldsymbol{\tau}_2)) = \boldsymbol{\varrho}(\boldsymbol{\tau}_1)$ and so $\boldsymbol{\varrho}'(\boldsymbol{\tau}_2) = \boldsymbol{\varrho}(\boldsymbol{\tau}_1)$.

($\Rightarrow$) We assume that $\forall \boldsymbol{\varrho} \in \mathsf{P}^\nu\ .\ \exists \boldsymbol{\varrho}' \in \mathsf{P}^\nu\ .\ \boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$. The structural proof is by cases on the pair $\langle \boldsymbol{\tau}_1, \boldsymbol{\tau}_2\rangle$ ordered lexicographically. It consists in constructing ϑ given $\boldsymbol{\varrho}$ and $\boldsymbol{\varrho}'$.

- $\boldsymbol{\tau}_1 = a \in \mathsf{F}_0$
  - $\boldsymbol{\tau}_2 = b \in \mathsf{F}_0$. If $b \neq a$, the hypothesis is false and the implication is true. Otherwise $b = a$ and any substitution has $\vartheta(\boldsymbol{\tau}_1) = \vartheta(a) = a = \vartheta(\boldsymbol{\tau}_2)$.
  - $\boldsymbol{\tau}_2 = \beta \in \mathbb{V}_{\mathfrak{t}}$. Any substitution s.t. $\vartheta(\beta) = a$ has $\vartheta(\boldsymbol{\tau}_2) = \vartheta(\beta) = a = \boldsymbol{\tau}_1$.
  - $\boldsymbol{\tau}_2 = g(\boldsymbol{\tau}''_1, \ldots, \boldsymbol{\tau}''_m)$, $g \in \mathsf{F}_m$. $\forall \boldsymbol{\varrho}, \boldsymbol{\varrho}'\ .\ \boldsymbol{\varrho}(\boldsymbol{\tau}_1) \neq \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$ so that the hypothesis is false and the implication is true.
- $\boldsymbol{\tau}_1 = \alpha \in \mathbb{V}_{\mathfrak{t}}$
  - $\boldsymbol{\tau}_2 = b \in \mathsf{F}_0$. If $\boldsymbol{\varrho}(\alpha) \neq b$, the hypothesis is false and the implication is true. Otherwise $\boldsymbol{\varrho}(\alpha) = b$.
  - $\boldsymbol{\tau}_2 = \beta \in \mathbb{V}_{\mathfrak{t}}$. Any substitution such that $\vartheta(\beta) = \alpha$ will do.
  - $\boldsymbol{\tau}_2 = g(\boldsymbol{\tau}''_1, \ldots, \boldsymbol{\tau}''_m)$, $g \in \mathsf{F}_m$. Since we assume that the signature F has at least two different function symbols, there is a term $\boldsymbol{\tau}$ with this different symbol at

the root. For $\boldsymbol{\varrho}$ such that $\boldsymbol{\varrho}(\alpha) = \boldsymbol{\tau}$ there is no $\boldsymbol{\varrho}'$ such that $\boldsymbol{\varrho}(\alpha) = \boldsymbol{\tau} = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2) = g(\boldsymbol{\varrho}'(\boldsymbol{\tau}''_1), \ldots, \boldsymbol{\varrho}'(\boldsymbol{\tau}''_m))$. In that case, the hypothesis is false and the implication is true.

- $\boldsymbol{\tau}_1 = f(\boldsymbol{\tau}'_1, \ldots, \boldsymbol{\tau}'_n)$, $f \in \mathsf{F}_n$
  - $\boldsymbol{\tau}_2 = b \in \mathsf{F}_0$. $\forall \boldsymbol{\varrho}, \boldsymbol{\varrho}' \mathrel{.} \boldsymbol{\varrho}(\boldsymbol{\tau}_1) \neq \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$ so the hypothesis is false and the implication is true.
  - $\boldsymbol{\tau}_2 = \beta \in \mathbb{V}_{\mathfrak{t}}$. Simply choose $\vartheta(\beta) = \boldsymbol{\tau}_1$.
  - $\boldsymbol{\tau}_2 = g(\boldsymbol{\tau}''_1, \ldots, \boldsymbol{\tau}''_m)$, $g \in \mathsf{F}_m$. The hypothesis that $\boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$, that is $f(\boldsymbol{\varrho}(\boldsymbol{\tau}'_1), \ldots, \boldsymbol{\varrho}(\boldsymbol{\tau}'_n)) = g(\boldsymbol{\varrho}'(\boldsymbol{\tau}''_1), \ldots, \boldsymbol{\varrho}'(\boldsymbol{\tau}''_m))$, implies that $f = g$, $m = n$, and $\forall i \in [1, n] \mathrel{.} \boldsymbol{\varrho}(\boldsymbol{\tau}'_j) = \boldsymbol{\varrho}'(\boldsymbol{\tau}''_j)$. So, by structural ind. hyp., $\exists \vartheta_j \mathrel{.} \vartheta_j(\boldsymbol{\tau}''_j) = \boldsymbol{\tau}'_j$ that is $\boldsymbol{\tau}_1 = f(\boldsymbol{\tau}'_1, \ldots, \boldsymbol{\tau}'_n) = f(\vartheta_1(\boldsymbol{\tau}''_1), \ldots, \vartheta_n(\boldsymbol{\tau}''_n))$. We now have to define ϑ such that $\vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1$.
    - For the variables α that do not occur in $\boldsymbol{\tau}_2$, we choose $\vartheta(\alpha) = \alpha$;
    - For variables α that occur once or more in $\boldsymbol{\tau}_2$, say in $\boldsymbol{\tau}''_j$ and $\boldsymbol{\tau}''_k$, $j, k \in [1, n]$, we have
    $$\begin{aligned} \boldsymbol{\tau}''_j &= f^1_j(\ldots f^{m_j}_j(\ldots, \alpha, \ldots) \ldots) \\ \boldsymbol{\tau}''_k &= f^1_k(\ldots f^{m_k}_k(\ldots, \alpha, \ldots) \ldots) \end{aligned}$$
    There are two subcases:
      - If for all occurrences, the substitutions are identical, we choose $\vartheta(\alpha) = \vartheta_j(\alpha) = \vartheta_k(\alpha)$;
      - Otherwise, $\exists \alpha, j \neq k \mathrel{.} \vartheta_j(\alpha) \neq \vartheta_k(\alpha)$[2] and so $\vartheta_j(\boldsymbol{\tau}''_j) \neq \vartheta_k(\boldsymbol{\tau}''_k)$. Therefore, $\exists \boldsymbol{\varrho} \mathrel{.} \boldsymbol{\varrho}(\vartheta_j(\boldsymbol{\tau}''_j)) \neq \boldsymbol{\varrho}(\vartheta_k(\boldsymbol{\tau}''_k))$. By hypothesis, $\exists \boldsymbol{\varrho}' \mathrel{.} \boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$ so $\boldsymbol{\varrho}(\boldsymbol{\tau}_1) = f(\boldsymbol{\varrho}(\boldsymbol{\tau}'_1), \ldots, \boldsymbol{\varrho}(\boldsymbol{\tau}'_n)) = f(\boldsymbol{\varrho}(\vartheta_1(\boldsymbol{\tau}''_1)), \ldots, \boldsymbol{\varrho}(\vartheta_n(\boldsymbol{\tau}''_n))) = f(\boldsymbol{\varrho}'(\boldsymbol{\tau}''_1), \ldots, \boldsymbol{\varrho}'(\boldsymbol{\tau}''_n)) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$. If follows that $\boldsymbol{\varrho}(\boldsymbol{\tau}'_j) = \boldsymbol{\varrho}(\vartheta_j(\boldsymbol{\tau}''_j)) = \boldsymbol{\varrho}'(\boldsymbol{\tau}''_j)$ and $\boldsymbol{\varrho}(\boldsymbol{\tau}'_k) = \boldsymbol{\varrho}(\vartheta_k(\boldsymbol{\tau}''_k)) = \boldsymbol{\varrho}'(\boldsymbol{\tau}''_k)$. For the term $\boldsymbol{\tau}''_j$, we have $\boldsymbol{\varrho}'(\boldsymbol{\tau}''_j) = f^1_j(\ldots f^{m_j}_j(\ldots, \boldsymbol{\varrho}'(\alpha), \ldots) \ldots) = \boldsymbol{\varrho}(\vartheta_j(\boldsymbol{\tau}''_j)) = f^1_j(\ldots f^{m_j}_j(\ldots, \boldsymbol{\varrho}(\vartheta_j(\alpha)), \ldots) \ldots)$ and so $\boldsymbol{\varrho}'(\alpha) = \boldsymbol{\varrho}(\vartheta_j(\alpha))$. Similarly, for the term $\boldsymbol{\tau}''_k$, we get $\boldsymbol{\varrho}'(\alpha) = \boldsymbol{\varrho}(\vartheta_k(\alpha))$. It follows that $\boldsymbol{\varrho}(\vartheta_j(\alpha)) = \boldsymbol{\varrho}(\vartheta_k(\alpha))$, a contradiction. This case is therefore impossible.

In conclusion, in all cases which are possible, the hypothesis $\forall \boldsymbol{\varrho} \mathrel{.} \exists \boldsymbol{\varrho}' \mathrel{.} \boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2)$ implies that $\exists \vartheta \in \Sigma \mathrel{.} \vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1$. It follows that

$$\begin{aligned} & [\boldsymbol{\tau}_1]_{\simeq^\nu} \preceq_{\simeq^\nu} [\boldsymbol{\tau}_2]_{\simeq^\nu} \\ \Leftrightarrow\ & \boldsymbol{\tau}_1 \preceq^\nu \boldsymbol{\tau}_2 && \wr\text{lemma 5}\wr \\ \Leftrightarrow\ & \mathit{ground}(\boldsymbol{\tau}_1) \subseteq \mathit{ground}(\boldsymbol{\tau}_2) && \wr\text{def. (9) of } \preceq^\nu\wr \\ \Leftrightarrow\ & \{\boldsymbol{\varrho}(\boldsymbol{\tau}_1) \mid \boldsymbol{\varrho} \in \mathsf{P}^\nu\} \subseteq \{\boldsymbol{\varrho}'(\boldsymbol{\tau}_2) \mid \boldsymbol{\varrho}' \in \mathsf{P}^\nu\} && \wr(8)\wr \\ \Leftrightarrow\ & \forall \boldsymbol{\varrho} \in \mathsf{P}^\nu \mathrel{.} \exists \boldsymbol{\varrho}' \in \mathsf{P}^\nu \mathrel{.} \boldsymbol{\varrho}(\boldsymbol{\tau}_1) = \boldsymbol{\varrho}'(\boldsymbol{\tau}_2) && \wr\text{def. } \subseteq\wr \\ \Leftrightarrow\ & \exists \vartheta \in \Sigma \mathrel{.} \vartheta(\boldsymbol{\tau}_2) = \boldsymbol{\tau}_1 && \wr(17)\wr \quad \square \end{aligned}$$

VIII. Conclusion

We have shown that a concrete program property represented by a set of ground terms can be over-approximated by a term with variables. Of course a concrete property P represented by a set of terms with variables can be overapproximated by the term with variables $LCG_{\simeq^\nu}(P)$ (this is Galois connection of the homomorphic abstraction $\alpha(X) \triangleq \bigsqcup\{h(x) \mid x \in X\}$ of a set X, where h is the identity). The complete lattice structure follows from the fact that the image of a complete lattice by a Galois retraction is a complete lattice [26, Theorem 4.1]. This approach yields algorithms together with their soundness proof by abstraction preservation [28, section 48.8].

We have shown that this semantic construction yields the same subsumption partial order defined syntactically by Gordon Plotkin [3], [4] and John Reynolds [5].

One can avoid variables by representing a term with variables as a rooted directed acyclic graph (DAG) that is the term syntax tree where the leaf nodes that have the same variable are joined [29] (mathematically represented by a tree and an equivalence relation between leaves that have the same variable). The lattice structure of symbolic terms generalizes to rooted order-sorted feature (OSF) graphs [30], [31].

[2] An example is $f(\alpha, \beta) \not\preceq^\nu f(\alpha, \alpha)$ with $\vartheta_1(\alpha) = \alpha$ and $\vartheta_2(\alpha) = \beta$.

References

[1] J. Herbrand, “Recherches sur la théorie de la démonstration,” Thèse, Université de Paris, 1930, ch. V of “Écrits logiques”, Jean Van Heijenoort (Ed.), Presses Universitaires de France, 1968, pp. 35–143.

[2] ——, “Investigations in proof theory,” Thesis, Université de Paris, 1930, ch. V of “Logical Writings”, Warren D. Goldfarb (Ed.), Springer Netherlands, 1971, pp. 44–202, English translation of [1].

[3] G. D. Plotkin, “A note on inductive generalization,” in *Machine Intelligence*, D. Meltzer, B.; Michie, Ed. Edinburgh University Press, 1970, vol. 5, pp. 153—163. [Online]. Available: http://homepages.inf.ed.ac.uk/gdp/publications/MI5_note_ind_gen.pdf

[4] ——, “A further note on inductive generalization,” in *Machine Intelligence*, D. Meltzer, B.; Michie, Ed. Edinburgh University Press, 1971, vol. 6, pp. 101—124. [Online]. Available: http://homepages.inf.ed.ac.uk/gdp/publications/MI6_further_note.pdf

[5] J. C. Reynolds, “Transformational systems and the algebraic structure of atomic formulas,” in *Machine Intelligence*, D. Meltzer, B.; Michie, Ed. Edinburgh University Press, 1970, vol. 5, pp. 135—151. [Online]. Available: http://www.cs.cmu.edu/afs/cs/user/jcr/ftp/transysalg.pdf

[6] J. A. Robinson, “A machine-oriented logic based on the resolution principle,” *J. ACM*, vol. 12, no. 1, pp. 23–41, 1965.

[7] ——, *Logic: Form and Function – The Mechanization of Deductive Reasoning*, ser. Artificial Intelligence. Elsevier North-Holland, 1979.

[8] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, “Satisfiability modulo theories,” in *Handbook of Satisfiability*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2009, vol. 185, pp. 825–885.

[9] J. C. King, “Symbolic execution and program testing,” *Commun. ACM*, vol. 19, no. 7, pp. 385–394, 1976.

[10] R. Milner, “A theory of type polymorphism in programming,” *J. Comput. Syst. Sci.*, vol. 17, no. 3, pp. 348–375, 1978.

[11] L. Damas and R. Milner, “Principal type-schemes for functional programs,” in *POPL*. ACM Press, 1982, pp. 207–212.

[12] A. Colmerauer, “Prolog in 10 figures,” *Commun. ACM*, vol. 28, no. 12, pp. 1296–1310, 1985.

[13] R. A. Kowalski, “The early years of logic programming,” *Commun. ACM*, vol. 31, no. 1, pp. 38–43, 1988.

[14] A. Colmerauer and P. Roussel, “The birth of prolog,” in *HOPL Preprints*. ACM, 1993, pp. 37–52.

[15] L. Sterling and E. Shapiro, *The Art of Prolog - Advanced Programming Techniques, 2nd Ed.* MIT Press, 1994.

[16] K. L. Clark and S. Åke Tärnlund, *Logic Programming.* Academic Press, New York, NY, US, 1982.

[17] R. Barbuti, R. Giacobazzi, and G. Levi, "A general framework for semantics-based bottom-up abstract interpretation of logic programs," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 1, pp. 133–181, 1993.

[18] M. V. Hermenegildo, G. Puebla, F. Bueno, and P. López-García, "Program development using abstract interpretation (and the ciao system preprocessor)," in *SAS*, ser. Lecture Notes in Computer Science, vol. 2694. Springer, 2003, pp. 127–152.

[19] P. Cousot, R. Cousot, and R. Giacobazzi, "Abstract interpretation of resolution-based semantics," *Theor. Comput. Sci.*, vol. 410, no. 46, pp. 4724–4746, 2009.

[20] B. Steensgaard, "Points-to analysis in almost linear time," in *POPL.* ACM Press, 1996, pp. 32–41.

[21] K. Muthukumar and M. V. Hermenegildo, "Determination of variable dependence information through abstract interpretation," in *NACLP.* MIT Press, 1989, pp. 166–185.

[22] P. Cousot, "Types as abstract interpretations," in *POPL.* ACM Press, 1997, pp. 316–331.

[23] J. R. Hindley, *Basic Simple Type Theory*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2008.

[24] A. Church, "A formulation of the simple theory of types," *J. Symb. Log.*, vol. 5, no. 2, pp. 56–68, 1940.

[25] X. Leroy, D. Doligez, A. Frisch, J. Garrigue, D. Rémy, and J. Vouillon, "The OCaml system, release 4.10, Documentation and user's manual," 2020, institut National de Recherche en Informatique et en Automatique. [Online]. Available: http://caml.inria.fr/pub/docs/manual-ocaml/

[26] M. Ward, "The closure operators of a lattice," *Annals of Mathematics*, vol. 43, no. 2, pp. 191–196, 1942.

[27] G. P. Huet, "Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems," *J. ACM*, vol. 27, no. 4, pp. 797–821, 1980.

[28] P. Cousot, *Principle of Abstract Interpretation.* MIT Press, 2021.

[29] M. Paterson and M. N. Wegman, "Linear unification," *J. Comput. Syst. Sci.*, vol. 16, no. 2, pp. 158–167, 1978.

[30] H. Aït-Kaci, "A lattice theoretic approach to computation based on a calculus of partially ordered type structures (property inheritance, nets, graph unification)," PhD thesis, Computer and Information Science Dept., University of Pennsylvania, 1984.

[31] H. Aït-Kaci, A. Podelski, and S. C. Goldstein, "Order sorted feature theory unification," *J. Log. Program.*, vol. 30, no. 2, pp. 99–124, 1997.