Nominal C Unification

Seminário de Computação - UnB

Gabriel Silva, Advisor: Mauricio Ayala-Rincón

November 14, 2018

Department of Mathematics - University of Brasília

Table of contents

Nominal C Unification

Nominal C Unification

Definition of the Problem

Unification Problem

Definition (Unification Problem)

A unification problem is a pair $\langle \Delta, P \rangle$, where Δ is a freshness context and P is a finite set of equations $(s \overset{?}{\approx}_{\alpha} t)$ and freshness constraints $(a \overset{?}{\#} s)$.

Solution to a Unification Problem

Definition (Solution to a Unification Problem)

TO DO.

More General Solutions

Definition (More General Solution)

TO DO

Nominal C Unification

Differences from Nominal Unification

Difference from Nominal Unification - Fixpoint Equations

TO DO

Difference from Nominal Unification - Set of Solutions

TO DO

Nominal C Unification

A Functional Nominal C-Unification Algorithm

General Comments about the Functional Nominal C-Unification Algorithm

A Functional Nominal C-Unification Algorithm I

1: **procedure** UNIFY$(\Delta, \sigma, \mathit{UnPrb}, \mathit{FxPntEq})$
2: **if** null(UnPrb) **then**
3: **return** list$((\Delta, \sigma, \mathit{FxPntEq}))$
4: **else**
5: $t = \text{head}(\mathit{UnPrb})[1]$
6: $s = \text{head}(\mathit{UnPrb})[2]$
7: $\mathit{UnPrb}' = \text{tail}(\mathit{UnPrb})$
8: **if** $(s == \pi \cdot X)$ and $(X$ not in $t)$ **then**
9: $\sigma' = \{X \to t\}$
10: $\sigma'' = \sigma' \cup \sigma$
11: $(\Delta'$, bool1$) = \text{appSub2Ctxt}(\sigma_1, \Delta)$
12: $\mathit{UnPrb}' = (\mathit{UnPrb})\sigma' + (\mathit{FxPntEq})\sigma'$

A Functional Nominal C-Unification Algorithm II

13: **if** bool1 **then return** $\text{UNIFY}(\Delta', \sigma'', \mathit{UnPrb}', \mathit{null})$

14: **else return** null

15: **else**

16: **if** $t == a$ **then**

17: **if** s == a **then**

18: **return** $\text{UNIFY}(\Delta, \sigma, \mathit{UnPrb}', \mathit{FxPntEq})$

19: **else return** null

20: **else if** $t == \pi \cdot X$ **then**

21: **if** (X not in s) **then**

22: ▷ Similar to case above where s is a suspension

A Functional Nominal C-Unification Algorithm III

23: **else if** $(s == \pi' \cdot X)$ **then**

24: $FxPntEq' = FxPntEq \cup \{((\pi')^{-1} \oplus \pi) \cdot X\}$

25: **return** $\textsc{unify}(\Delta, \sigma, UnPrb, FxPntEq')$

26: **else return** null

27: **else if** $t == \langle\rangle$ **then**

28: **if** $s == \langle\rangle$ **then**

29: **return** $\textsc{unify}(\Delta, \sigma, UnPrb, FxPntEq)$

30: **else return** null

31: **else if** $t == \langle t_1, t_2 \rangle$ **then**

32: **if** $s == \langle s_1, s_2 \rangle$ **then**

33: $UnPrb'' = UnPrb' + [(s_1, t_1)] + [(s_2, t_2)]$

34: **return** $\textsc{unify}(\Delta, \sigma, UnPrb'', FxPntEq)$

35: **else return** null

A Functional Nominal C-Unification Algorithm IV

36: **else if** $t == [a]t_1$ **then**
37: **if** $s == [a]s_1$ **then**
38: $UnPrb'' = UnPrb' + [(t_1, s_1)] + [(t_2, s_2)]$
39: **return** $\textsc{unify}(\Delta, \sigma, UnPrb'', FxPntEq)$
40: **else if** $s == [b]s_1$ **then**
41: $(\Delta', bool1) = fresh(a, s_1)$
42: $\Delta'' = \Delta \cup \Delta'$
43: $UnPrb'' = UnPrb + [(t_1, \ (a\ b)\ s_1)]$
44: **if** bool1 **then**
45: **return** $\textsc{unify}(\Delta'', \sigma, UnPrb'', FxPntEq)$
46: **else return** null
47: **else**
48: **return** null

A Functional Nominal C-Unification Algorithm V

49: **else if** $t == ft_1$ **then** ▷ f is not commutative

50: **if** s != fs_1 **then return** null

51: **else**

52: $UnPrb'' = UnPrb + [(t_1, s_1)]$

53: **return** $\text{UNIFY}(\Delta, \sigma, UnPrb'', FxPntEq)$

A Functional Nominal C-Unification Algorithm VI

54: **else** ▷ t is of the form $f(t_1, t_2)$

55: **if** s != $f(s_1, s_2)$ **then return** null

56: **else**

57: $UnPrb_1 = UnPrb' + [(s_1, t_1)] + [(s_2, t_2)]$

58: $s_1 =$ **return** $\text{UNIFY}(\Delta, \sigma, UnPrb_1, FxPntEq)$

59: $UnPrb_2 = UnPrb' + [(s_1, t_2)] + [(s_2, t_1)]$

60: $s_2 =$ **return** $\text{UNIFY}(\Delta, \sigma, UnPrb_2, FxPntEq)$

61: **return** $\text{APPEND}(s_1, s_2)$

62: