LIFLC – Logique classique

CM5 – Logique du premier ordre: termes

Licence informatique UCBL – Automne 2018–2019

https://liris.cnrs.fr/ecoquery/dokuwiki/doku.php?id=enseignement:logique:start

Dpt Informatique

Des faits aux objets

Limite de la logique propositionnelle : pas d'objets

- faits uniquement
- objets fixés dans ces faits
- nombre fixé de faits (même si non borné) dans une formule

Un vocabulaire pour le discours

Objets différents

Vocabulaires différents

Symboles différents

Signature (pour les termes)

Lister les symboles utilisés

Definition

Une *signature* $\mathcal{S} = (\mathcal{C}, \mathcal{F}, ar)$:

- $\mathcal{C}$: ensemble (non vide) de **symboles** de constantes
- $\mathcal{F}$: ensemble de **symboles** de fonction
- $ar : \mathcal{F} \to \mathcal{N}^*$: nombre d'arguments de chaque symbole de fonction

Notation

f/n *signifie* $ar(f) = n$

Exemple : expressions arithmétiques sur les entiers naturels

Entiers (de Peano) : $\mathcal{S}_{peano} = (\mathcal{C}_{peano}, \mathcal{F}_{peano}, ar_{peano})$

- $\mathcal{C}_{peano} = \{zero\}$
- $\mathcal{F}_{peano} = \{succ/1, plus/2, mult/2\}$ $\leftarrow$ def implicite de ar_{peano}

Exemple : expressions arithmétiques sur les entiers naturels

Entiers (de Peano) : $\mathcal{S}_{peano} = (\mathcal{C}_{peano}, \mathcal{F}_{peano}, ar_{peano})$

- $\mathcal{C}_{peano} = \{zero\}$
- $\mathcal{F}_{peano} = \{succ/1, plus/2, mult/2\}$ $\leftarrow$ def implicite de ar_{peano}

Exemple : chaînes de caractères sur a, b, c

Chaînes de caractères : $\mathcal{S}_{char} = (\mathcal{C}_{char}, \mathcal{F}_{char}, ar_{char})$

- $\mathcal{C}_{char} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- $\mathcal{F}_{char} = \{concat/2\}$

⚠ concat ↔ 1 symbole (vrai aussi pour $\mathcal{F}_{peano}$)

Exemple : chaînes de caractères sur a, b, c

Chaînes de caractères : $\mathcal{S}_{char} = (\mathcal{C}_{char}, \mathcal{F}_{char}, ar_{char})$

- $\mathcal{C}_{char} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- $\mathcal{F}_{char} = \{concat/2\}$

⚠ concat $\leftrightarrow$ 1 symbole
(vrai aussi pour $\mathcal{F}_{peano}$)

Variables

Parler uniquement d'objets fixés : peu intéressant

Ensemble de variables $\mathcal{V}$ représentant des objets :

- infini
- tel qu'on peut toujours prendre une variable **fraîche**

Variables notées $x, y, x', x_1, \ldots$

⚠ variables au premier ordre $\neq$ variables propositionnelles ⚠

Ensemble inductif des termes

Combinaison bien construite des symboles de la signature et *variables*

Definition

L'ensemble des termes $\mathcal{T}$ sur $\mathcal{S} = (\mathcal{C}, \mathcal{F}, ar)$ est le plus petit ensemble tel que :

- si $x \in \mathcal{V}$ alors $x \in \mathcal{T}$
- si $c \in \mathcal{C}$ alors $c \in \mathcal{T}$
- si $t_1 \in \mathcal{T}, \ldots, t_n \in \mathcal{T}$ et $f/n \in \mathcal{F}$
  alors $f(t_1, \ldots, t_n) \in \mathcal{T}$ $\leftarrow$ *terme composite*

⚠ Pas d'évaluation ici ⚠

Ensemble inductif des termes

Combinaison bien construite des symboles de la signature et *variables*

Definition

L'ensemble des termes $\mathcal{T}$ sur $\mathcal{S} = (\mathcal{C}, \mathcal{F}, ar)$ est le plus petit ensemble tel que :

- si $x \in \mathcal{V}$ alors $x \in \mathcal{T}$
- si $c \in \mathcal{C}$ alors $c \in \mathcal{T}$
- si $t_1 \in \mathcal{T}, \ldots, t_n \in \mathcal{T}$ et $f/n \in \mathcal{F}$
  alors $f(t_1, \ldots, t_n) \in \mathcal{T}$ $\leftarrow$ *terme composite*

⚠ Pas d'évaluation ici ⚠

Exemple : expressions arithmétiques

Avec $\mathcal{S}_{peano} = (\mathcal{C}_{peano}, \mathcal{F}_{peano}, ar_{peano})$

- $\mathcal{C}_{peano} = \{zero\}$
- $\mathcal{F}_{peano} = \{succ/1, plus/2, mult/2\}$

L'ensemble des termes $\mathcal{T}_{peano}$ sur $\mathcal{S}_{peano}$ est le plus petit ensemble tel que :

- si $x \in \mathcal{V}$ alors $x \in \mathcal{T}_{peano}$
- si $c = zero$ alors $c \in \mathcal{T}_{peano}$
- si $t \in \mathcal{T}_{peano}$ alors $succ(t) \in \mathcal{T}_{peano}$
- si $t_1, t_2 \in \mathcal{T}_{peano}$ alors $plus(t_1, t_2) \in \mathcal{T}_{peano}$
- si $t_1, t_2 \in \mathcal{T}_{peano}$ alors $mult(t_1, t_2) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : expressions arithmétiques (suite)

- $zero \in \mathcal{T}_{peano}$
- $0 \notin \mathcal{T}_{peano}$: 0 n'est pas dans la signature
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$: pas le bon nombre d'arguments
- $plus(succ(zero), x) \in \mathcal{T}_{peano}$

Exemple : chaînes de caractères

Avec $\mathcal{S}_{char} = (\mathcal{C}_{char}, \mathcal{F}_{char}, ar_{char})$

- $\mathcal{C}_{char} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
- $\mathcal{F}_{char} = \{concat/2\}$

L'ensemble des termes $\mathcal{T}_{char}$ sur $\mathcal{S}_{char}$ est le plus petit ensemble tel que :

- si $x \in \mathcal{V}$ alors $x \in \mathcal{T}_{char}$
- si $\mathtt{a} \in \mathcal{T}_{char}$, $\mathtt{b} \in \mathcal{T}_{char}$ et $\mathtt{c} \in \mathcal{T}_{char}$
- si $t_1, t_2 \in \mathcal{T}_{char}$ alors $concat(t_1, t_2) \in \mathcal{T}_{char}$

Exemple : chaînes de caractères (suite)

- a $\in \mathcal{T}_{char}$
- $concat(\mathrm{a}, \mathrm{c}) \in \mathcal{T}_{char}$
- ac $\notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

Exemple : chaînes de caractères (suite)

- $\mathtt{a} \in \mathcal{T}_{char}$
- $concat(\mathtt{a}, \mathtt{c}) \in \mathcal{T}_{char}$
- $\mathtt{ac} \notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

Exemple : chaînes de caractères (suite)

- $\mathtt{a} \in \mathcal{T}_{char}$
- $concat(\mathtt{a}, \mathtt{c}) \in \mathcal{T}_{char}$
- $\mathtt{ac} \notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

Exemple : chaînes de caractères (suite)

- $\mathsf{a} \in \mathcal{T}_{char}$
- $concat(\mathsf{a}, \mathsf{c}) \in \mathcal{T}_{char}$
- $\mathsf{ac} \notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

Exemple : chaînes de caractères (suite)

- $\mathtt{a} \in \mathcal{T}_{char}$
- $concat(\mathtt{a}, \mathtt{c}) \in \mathcal{T}_{char}$
- $\mathtt{ac} \notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

Exemple : chaînes de caractères (suite)

- $\mathtt{a} \in \mathcal{T}_{char}$
- $concat(\mathtt{a}, \mathtt{c}) \in \mathcal{T}_{char}$
- $\mathtt{ac} \notin \mathcal{T}_{char}$: pas dans les règles de construction

⚠ Terme $\neq$ évaluation d'une expression ⚠

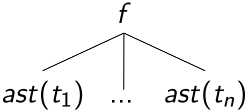
Arbre de syntaxe abstraite d'un terme

Représentation en arbre de la structure d'un terme défini sur $\mathcal{S} = (\mathcal{C}, \mathcal{F}, ar)$

Definition

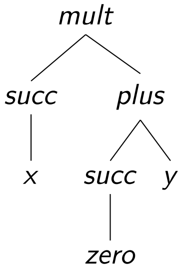
Fonction récursive $ast(t)$:

- si $c \in \mathcal{C}$ alors $ast(c) = c$
- si $x \in \mathcal{V}$ alors $ast(x) = x$
- si $t = f(t_1, \ldots, t_n)$ alors $ast(t) =$



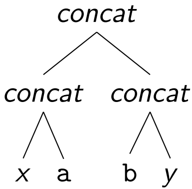
Exemple : expression arithmétique

$ast(mult(succ(x), plus(succ(zero), y))) =$



Exemple : expression sur chaînes de caractères

$ast(concat(concat(x, \texttt{a}), concat(\texttt{b}, y))) =$



Interpréter les termes

Moyen pour donner du sens aux termes

Questions :

- Quelles valeurs possibles ?
- Valeur (?) d'un symbole
- Variables ?

Univers

Definition

Ensemble $\mathcal{U}$ des valeurs possibles pour les termes

Pas vraiment contraint, mais fixé lorsqu'on interprète les termes

Interpréter les symboles

Interprétation I des symboles dans $\mathcal{C}$ et $\mathcal{F}$

Constantes : valeur fixée qui ne dépend que de l'interprétation$I$:

$$\text{Si } c \in \mathcal{C} \text{ alors } I(c) \in \mathcal{U}$$

Symboles de fonction : interprétés par des fonctions
On veut interpréter les termes dans $\mathcal{U}$:

$$\text{Si } f/n \in \mathcal{F} \text{ alors } I(f) : \mathcal{U}^n \to \mathcal{U}$$

Interpréter les variables

Interprétation : fixe une vision du monde

Valeur d'une variable peut changer pour une même interprétation

Definition (Valuation)

Fonction $\zeta : \mathcal{V} \to \mathcal{U}$

Retour sur l'interprétation des termes

Évaluation : fonction récursive

- sur les termes
- qui dépend aussi de l'interprétation I et de la valuation ζ

Definition ($eval(I, \zeta)(t)$)

- Si $c \in \mathcal{C}$ alors $eval(I, \zeta)(c) = I(c)$
- Si $x \in \mathcal{V}$ alors $eval(I, \zeta)(x) = \zeta(x)$
- Si $t = f(t_1, \dots, t_n)$
  si $eval(I, \zeta)(t_1) = u_1, \ldots, eval(I, \zeta)(t_n) = u_n$, et si $I(f) = \phi$ alors

$$eval(I, \zeta)(t) = \phi(u_1, \dots, u_n)$$

Exemple : expression arithmétiques

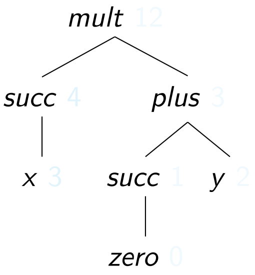
Univers $\mathcal{U}_{\mathcal{N}} = \mathcal{N}$ entiers naturels

$$
\begin{aligned}
I_{\mathcal{N}}(zero) &= 0 \\
I_{\mathcal{N}}(succ) &= n \mapsto n + 1 \\
I_{\mathcal{N}}(plus) &= n, m \mapsto n + m \\
I_{\mathcal{N}}(mult) &= n, m \mapsto n \times m
\end{aligned}
$$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

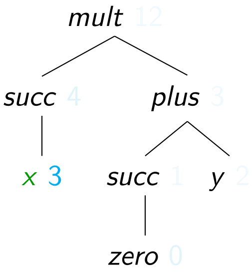
$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$



Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

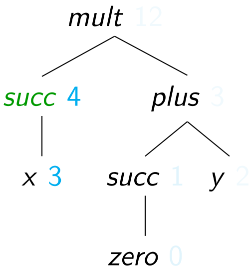


$\zeta(x) = 3$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

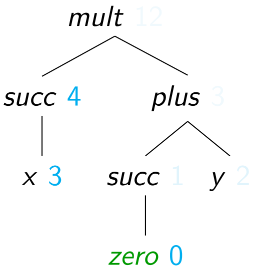


$I_{\mathcal{N}}(succ) = n \mapsto n + 1 \qquad (n \mapsto n + 1)(3) = 4$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

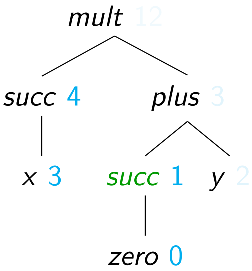


$I_{\mathcal{N}}(zero) = 0$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

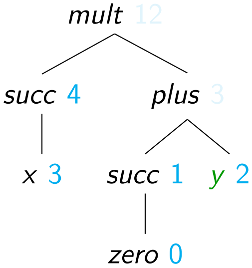


$I_{\mathcal{N}}(succ) = n \mapsto n + 1 \qquad (n \mapsto n + 1)(0) = 1$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

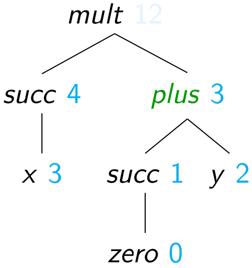


$\zeta(y) = 2$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

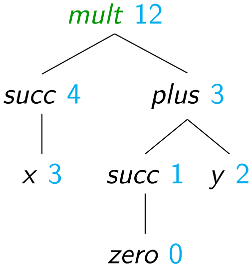


$I_{\mathcal{N}}(plus) = n, m \mapsto n + m \qquad (n, m \mapsto n + m)(1, 2) = 3$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$$

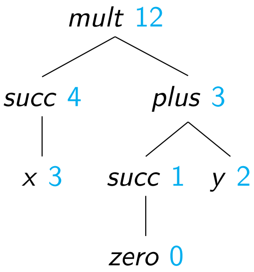


$I_{\mathcal{N}}(mult) = n, m \mapsto n \times m \qquad (n, m \mapsto n \times m)(4, 3) = 12$

Exemple : expressions arithmétiques (évaluation)

Soit $\zeta : x \mapsto 3, y \mapsto 2$

$$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) = 12$$



Exemple : chaînes de caractères

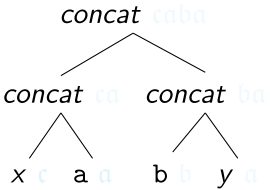
Univers $\mathcal{U}_{str} = \{\mathfrak{a}, \mathfrak{b}, \mathfrak{c}\}^*$: chaînes de caractères sur l'alphabet $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$

$$
\begin{aligned}
I_{str}(a) &= \mathfrak{a} \\
I_{str}(b) &= \mathfrak{b} \\
I_{str}(c) &= \mathfrak{c} \\
I_{str}(concat) &= s_1, s_2 \mapsto s_1 s_2
\end{aligned}
$$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

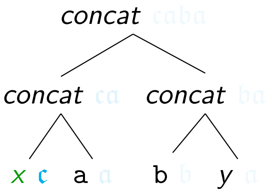
$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$



Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \texttt{a}), concat(\texttt{b}, y))) = \mathfrak{caba}$$

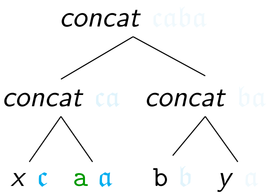


$\zeta(x) = \mathfrak{c}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) =$$

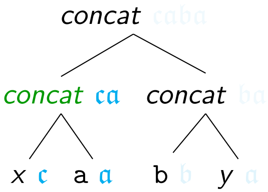


$I_{str}(a) = \mathfrak{a}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$

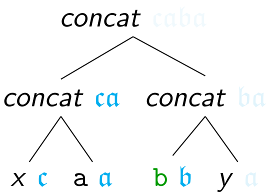


$I_{str}(concat) = s_1, s_2 \mapsto s_1 s_2 \qquad (s_1, s_2 \mapsto s_1 s_2)(\mathfrak{c}, \mathfrak{a}) = \mathfrak{ca}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$

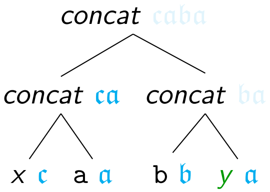


$I_{str}(b) = \mathfrak{b}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$

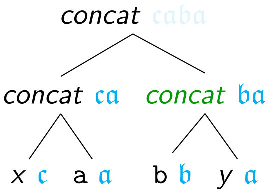


$\zeta(y) = \mathfrak{a}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$

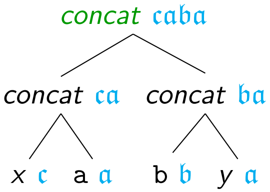


$I_{str}(concat) = s_1, s_2 \mapsto s_1 s_2 \qquad (s_1, s_2 \mapsto s_1 s_2)(\mathfrak{b}, \mathfrak{a}) = \mathfrak{ba}$

Exemple : chaînes de caractères (évaluation)

Soit $\zeta : x \mapsto \mathfrak{c}, y \mapsto \mathfrak{a}$

$$eval(I_{str})(concat(concat(x, \mathtt{a}), concat(\mathtt{b}, y))) = \mathfrak{caba}$$



$I_{str}(concat) = s_1, s_2 \mapsto s_1 s_2$ $\qquad$ $(s_1, s_2 \mapsto s_1 s_2)(\mathfrak{ca}, \mathfrak{ba}) = \mathfrak{caba}$

Interprétations non naturelles

S'appuyer sur
une connaissance *implicite*
de l'univers
est

Illusoire

$\rightarrow$ considérer toutes les interprétations possibles
même les plus farfelues

Exemple : Arithmétique étrange

Univers $\mathcal{U}_{park}$: 10 places de parking en ligne

$I_{park}(zero) =$ la place la plus à droite

$I_{park}(succ) =$ $p \mapsto$ la place à droite de p ou
p s'il n'y a pas de place à droite de p

$I_{park}(plus) =$ $p_1, p_2 \mapsto$ la place plus proche du milieu de p_1 et p_2, en privilégiant la plus à gauche en cas d'égalité.

$I_{park}(mult) =$ $p_1, p_2 \mapsto$ la place plus proche du milieu de p_1 et p_2, en privilégiant la plus à droite en cas d'égalité.

Substitutions

Replacer une partie d'un terme

- une ou plusieurs *variables*
- chacune par un terme

Definition

Une substitution est une fonction $\sigma : \mathcal{V} \to \mathcal{T}$ telle que $dom(\sigma)$ est fini

Notation

Si $dom(\sigma) = \{x_1, \ldots, x_n\}$ *et si pour tout* $1 \leq i \leq n$, $\sigma(x_i) = t_i$ *alors*

$$\sigma \ \textit{notée} \ [x_1 := t_1, x_2 := t_2, \ldots]$$

Appliquer une substitution

Appliquer une σ sur un terme $t \leftrightarrow$ faire le remplacement décrit par σ noté $t\sigma$

Definition (Application de substitution)

Récursivement définie par :

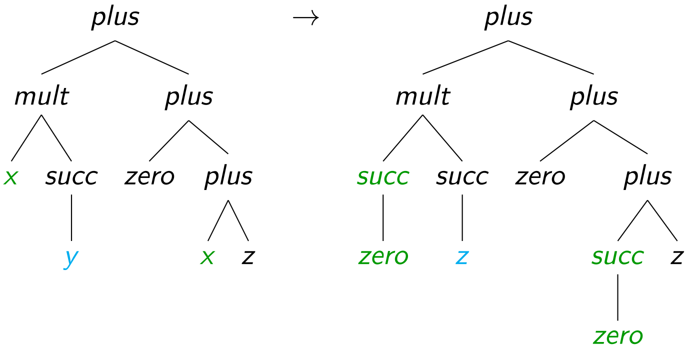
- $c\sigma = c$ si $c \in \mathcal{C}$
- $x\sigma = \sigma(x)$ si $x \in dom(\sigma)$
- $y\sigma = y$ si $y \in \mathcal{V}$ et $y \notin dom(\sigma)$
- $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$

Exemple

$$plus(mult(x, succ(y)), plus(zero, plus(x, z)))[x := succ(zero), y := z]$$

$$=$$

$$plus(mult(succ(zero), succ(z)), plus(zero, plus(succ(zero), z)))$$



Motifs

Besoin correspondance terme $\leftrightarrow$ forme

Décrire la forme

- par un autre terme $\leftarrow$ motif
- dont les variables représentent des “trous” à remplir

Filtrage de motif

Definition (Problème du filtrage de motif)

Étant donné un terme t et un autre terme m (le motif), existe-il une substitution σ telle que $m\sigma = t$.

Exemple

Soit $m = plus(succ(x), mult(y, x))$

- pour $t = plus(succ(zero), mult(succ(z), zero))$:
  $\sigma = [x := zero, y := succ(z)]$
- pour $t = mult(succ(zero), mult(succ(z), zero))$:
  pas de correspondance sur $mult$
- pour $t = plus(succ(zero), mult(succ(z), succ(zero)))$:
  deux termes incompatibles pour x

Combiner des substitutions

Exemple précédent : 2 termes incompatibles pour x
Combinaison de substitution

- pour les substitutions issues du filtrage de branches différentes d'ASA
- pouvant échouer

Definition (merge)

- $\mathit{merge}(\sigma_1, \sigma_2) = \mathtt{fail}$
  si on peut trouver $x \in \mathit{dom}(\sigma_1) \cap \mathit{dom}(\sigma_2)$ telle que $\sigma_1(x) \neq \sigma_2(x)$
- sinon $\mathit{merge}(\sigma_1, \sigma_2) = x \mapsto \begin{cases} \sigma_1(x) \text{ si } x \in \mathit{dom}(\sigma_1) \\ \sigma_2(x) \text{ si } x \in \mathit{dom}(\sigma_2) \setminus \mathit{dom}(\sigma_1) \end{cases}$

Combiner des substitutions

Exemple précédent : 2 termes incompatibles pour x
Combinaison de substitution

- pour les substitutions issues du filtrage de branches différentes d'ASA
- pouvant échouer

Definition (merge)

- $merge(\sigma_1, \ldots, \sigma_n) = \texttt{fail}$
  si on peut trouver i et j tels que $x \in dom(\sigma_i) \cap dom(\sigma_j)$ telle que $\sigma_i(x) \neq \sigma_j(x)$
- sinon $merge(\sigma_1, \ldots, \sigma_n) =$

$$x \mapsto \begin{cases} \sigma_1(x) \text{ si } x \in dom(\sigma_1) \\ \sigma_2(x) \text{ si } x \in dom(\sigma_2) \setminus dom(\sigma_1) \\ \ldots \\ \sigma_n(x) \text{ si } x \in dom(\sigma_n) \setminus (dom(\sigma_{n-1}) \cup \cdots \cup dom(\sigma_1)) \end{cases}$$

Calcul de filtrage de motif

Definition (match)

match(t, m) définie récursivement sur m par :

- si $m = c \in \mathcal{C}$ alors $match(t, c) = \begin{cases} [] \text{ si } t = c \\ \texttt{fail} \text{ si } t \neq c \end{cases}$
- si $m = x$ alors $match(t, x) = [x := t]$
- si $m = f(m_1, \ldots, m_n)$ et $t = f(t_1, \ldots, t_n)$ alors
  soit $\sigma_i = match(t_i, m_i)$ pour $1 \leq i \leq n$
  - si il existe i tel que $\sigma_i = \texttt{fail}$ alors $match(t, m) = \texttt{fail}$
  - sinon $match(t, m) = merge(\sigma_1, \ldots, \sigma_n)$
- sinon $match(t, m) = \texttt{fail}$

Unification

Definition

Soit t_1 et t_2 deux termes.

$$\text{Existe-il une substitution } \sigma \text{ telle que } t_1\sigma = t_2\sigma\,?$$

Filtrage de motif à deux sens

Exemple

Unifier

$$plus(mult(x, y), plus(y, succ(z)))$$

et

$$plus(mult(u, v), plus(w, w))$$

Solution

$$\sigma = [w := succ(z), y := succ(z), v := succ(z), x := u]$$

$$t_1\sigma = plus(mult(u, succ(z)), plus(succ(z), succ(z)))$$

Equations (syntaxiques) entre termes

Résoudre le problème d'unification

Équations entre termes :

$$t_i \stackrel{\triangle}{=} t'_i$$

Transformer ces équations selon un système de règles

Notation

- ***Une*** *équation notée E*
- *Ensemble d'équations noté* Ψ $\leftarrow$ *l'ordre n'est pas important, on omet* $\{\}$
- *Si* $E = \ t \stackrel{\triangle}{=} t'$ *alors* $E\sigma = \ t\sigma \stackrel{\triangle}{=} t'\sigma$
- *Si* $\Psi = E_1, \ldots, E_n$ *alors* $\Psi\sigma = E_1\sigma, \ldots, E_n\sigma$

Résolution des équations : règles

$$
\begin{array}{lrcll}
(Rev) & \Psi, t \stackrel{\triangle}{=} x & \rightsquigarrow & \Psi, x \stackrel{\triangle}{=} t & \text{si } t \notin \mathcal{V} \\
(Dec) & \Psi, f(t_1, \dots, t_n) \stackrel{\triangle}{=} f(t'_1, \dots, t'_n) & \rightsquigarrow & \Psi, t_1 \stackrel{\triangle}{=} t'_1, \dots, t_n \stackrel{\triangle}{=} t'_n & \\
(Sub) & \Psi, x \stackrel{\triangle}{=} t & \rightsquigarrow & \Psi[x := t], x \stackrel{\triangle}{=} t & \\
 & & & \text{si } x \notin V(f(t_1, \dots, t_n)) & \\
 & & & & \\
(FF) & \Psi, f(t_1, \dots, t_n) \stackrel{\triangle}{=} g(t'_1, \dots, t'_m) & \rightsquigarrow & \texttt{fail} & \text{si } f \neq g \\
(CC) & \Psi, c_1 \stackrel{\triangle}{=} c_2 & \rightsquigarrow & \texttt{fail} & \text{si } c_1 \neq c_2 \\
(FC) & \Psi, f(t_1, \dots, t_n) \stackrel{\triangle}{=} c & \rightsquigarrow & \texttt{fail} & \\
(CF) & \Psi, c \stackrel{\triangle}{=} f(t_1, \dots, t_n) & \rightsquigarrow & \texttt{fail} & \\
 & & & & \\
(Cyc) & \Psi, x \stackrel{\triangle}{=} f(t_1, \dots, t_n) & \rightsquigarrow & \texttt{fail} & \\
 & & & \text{si } x \in V(f(t_1, \dots, t_n)) &
\end{array}
$$

Solution au problème d'unification

Appliquer les règles jusqu'à obtention d'un point fixe
(*i.e.* jusqu'à ce qu'aucune règle ne puisse changer l'ensemble Ψ)

Si $\Psi \neq \mathtt{fail}$ alors Ψ est en **forme résolue**

Substitution σ_Ψ associée à Ψ :

$$\sigma_\Psi(x) = t \quad \text{si} \quad x \stackrel{\triangle}{=} t \in \Psi$$

pour avec $dom(\sigma_\Psi) = \{x \mid x \stackrel{\triangle}{=} t \in \Psi\}$

Théorème

Si $t \stackrel{\triangle}{=} t' \rightsquigarrow^* \Psi$ et Ψ en forme résolue alors le problème d'unification de t avec t' admet une solution : σ_Ψ.
Si $t \stackrel{\triangle}{=} t' \rightsquigarrow^* \mathtt{fail}$ alors le problème d'unification de t avec t' n'admet pas de solution.

Exemple

$$succ(plus(x, succ(y))) \triangleq succ(plus(u, u))$$

$$\begin{aligned}
&\rightsquigarrow\ plus(x, succ(y)) \triangleq plus(u, u) && (Dec)\\
&\rightsquigarrow\ x \triangleq u, succ(y) \triangleq u && (Dec)\\
&\rightsquigarrow\ x \triangleq u, succ(y) \triangleq x && (Sub)\\
&\rightsquigarrow\ x \triangleq u, x \triangleq succ(y) && (Rev)\\
&\rightsquigarrow\ succ(y) \triangleq u, x \triangleq succ(y) && (Sub)\\
&\rightsquigarrow\ u \triangleq succ(y), x \triangleq succ(y) && (Rev)
\end{aligned}$$

$$\sigma = [x := succ(y), u := succ(y)]$$

Exemples : échecs

$$succ(\textcolor{red}{mult}(x, succ(y))) \triangleq succ(plus(u, u))$$

$$\begin{aligned}
&\rightsquigarrow mult(x, succ(y)) \triangleq plus(u, u) && (Dec)\\
&\rightsquigarrow \texttt{fail} && (FF)
\end{aligned}$$

$$succ(plus(x, succ(\textcolor{red}{x}))) \triangleq succ(plus(u, u))$$

$$\begin{aligned}
&\rightsquigarrow plus(x, succ(x)) \triangleq plus(u, u) && (Dec)\\
&\rightsquigarrow x \triangleq u, succ(x) \triangleq u && (Dec)\\
&\rightsquigarrow x \triangleq u, succ(x) \triangleq x && (Sub)\\
&\rightsquigarrow x \triangleq u, x \triangleq succ(x) && (Rev)\\
&\rightsquigarrow \texttt{fail} && (Cyc)
\end{aligned}$$