

# LIFLC – Logique classique

## CM5 – Logique du premier ordre: termes

Licence informatique UCBL – Automne 2017–2018

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



- 1 Premier ordre
- 2 Termes : syntaxe
- 3 Termes : sémantique
- 4 Opérations syntaxiques sur les termes

# 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
- $ar : \mathcal{F} \rightarrow \mathcal{N}^*$  : 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) :  $\mathcal{S}_{peano} = (\mathcal{C}_{peano}, \mathcal{F}_{peano}, ar_{peano})$

- $\mathcal{C}_{peano} = \{zero\}$

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

← def implicite de  $ar_{peano}$

# Exemple : expressions arithmétiques sur les entiers naturels

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

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

## Exemple : chaînes de caractères sur a, b, c

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

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

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



## Exemple : chaînes de caractères sur a, b, c

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

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

⚠ 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  $x, y, x', x_1, \dots$



variables au premier ordre

$\neq$



variables propositionnelles

- 1 Premier ordre
- 2 Termes : syntaxe**
- 3 Termes : sémantique
- 4 Opérations syntaxiques sur les termes

## 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  $t_1 \in \mathcal{T}, \dots, t_n \in \mathcal{T}$  et  $f/n \in \mathcal{F}$   
alors  $f(t_1, \dots, t_n) \in \mathcal{T}$

← *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  $t_1 \in \mathcal{T}, \dots, t_n \in \mathcal{T}$  et  $f/n \in \mathcal{F}$   
alors  $f(t_1, \dots, t_n) \in \mathcal{T}$

← *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 le langage
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero) \notin \mathcal{T}_{peano}$  : 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
- $x \in \mathcal{T}_{peano}$
- $succ(zero) \in \mathcal{T}_{peano}$
- $plus(zero)$  ... pas le bon nombre d'arguments
- $plus(succ(zero), x)$  ...  $\mathcal{T}_{peano}$

## 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
- $plus(succ(zero), x) \notin \mathcal{T}_{peano}$

## 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
- $plus(succ(zero), x) \notin \mathcal{T}_{peano}$

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

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

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

## Exemple : chaînes de caractères (suite)

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

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

## Exemple : chaînes de caractères (suite)

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

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

## Exemple : chaînes de caractères (suite)

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

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

## Exemple : chaînes de caractères (suite)

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

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

## Exemple : chaînes de caractères (suite)

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

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



## Exemple : chaînes de caractères (suite)

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

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

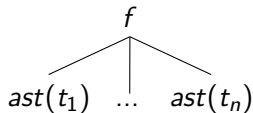
# Arbre de syntaxe abstraite d'un terme

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

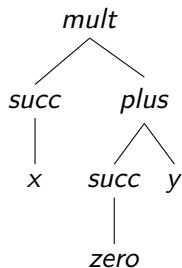
## Definition

Fonction récursive  $ast(t)$  :

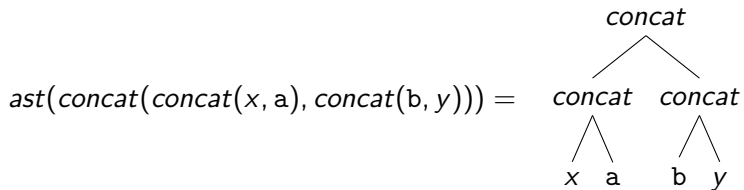
- si  $c \in \mathcal{C}$  alors  $ast(c) = c$
- si  $x \in \mathcal{V}$  alors  $ast(x) = x$
- si  $t = f(t_1, \dots, t_n)$  alors  $ast(t) =$



## Exemple : expression arithmétique

$$\text{ast}(\text{mult}(\text{succ}(x), \text{plus}(\text{succ}(\text{zero}), y))) =$$


## Exemple : expression sur chaînes de caractères



- 1 Premier ordre
- 2 Termes : syntaxe
- 3 Termes : sémantique**
- 4 Opérations syntaxiques sur les termes

# 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  $\mathcal{U}$  :

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



# Interpréter les variables

Interprétation : fixe une vision du monde

Valeur d'une variable peut changer pour une même interprétation

Definition (Valuation)

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

## Retour sur l'interprétation des termes

Évaluation : fonction récursive

- sur les termes
- qui dépend aussi de l'interprétation  $I$  et de la valuation  $\zeta$

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  $eval(I, \zeta)(t_1) = u_1, \dots, eval(I, \zeta)(t_n) = u_n$ , 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

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

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

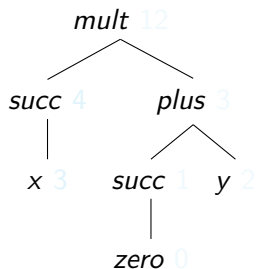
$$I_{\mathcal{N}}(\text{plus}) = n, m \mapsto n + m$$

$$I_{\mathcal{N}}(\text{mult}) = n, m \mapsto n \times m$$

## Exemple : expressions arithmétiques (évaluation)

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

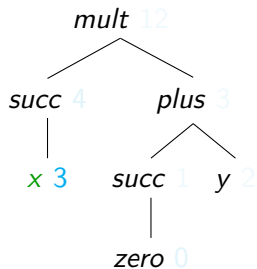
$eval(I_{\mathcal{N}}, \zeta)(mult(succ(x), plus(succ(zero), y))) =$



## Exemple : expressions arithmétiques (évaluation)

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

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

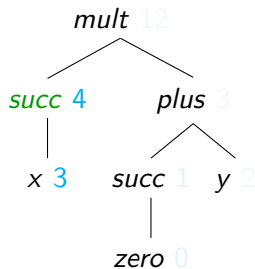


$\zeta(x) = 3$

## Exemple : expressions arithmétiques (évaluation)

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

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

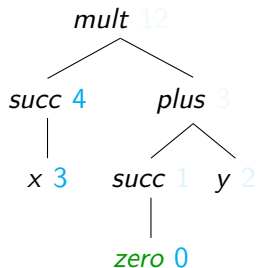


$$I_{\mathcal{N}}(\mathit{succ}) = n \mapsto n + 1 \quad (n \mapsto n + 1)(3) = 4$$

## Exemple : expressions arithmétiques (évaluation)

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

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

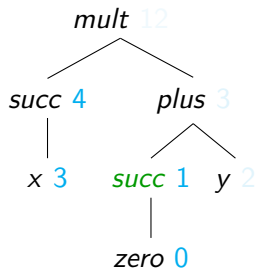


$I_{\mathcal{N}}(\mathit{zero}) = 0$

## Exemple : expressions arithmétiques (évaluation)

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

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



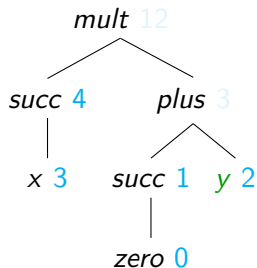
$$I_{\mathcal{N}}(\mathit{succ}) = n \mapsto n + 1 \quad (n \mapsto n + 1)(0) = 1$$



## Exemple : expressions arithmétiques (évaluation)

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

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

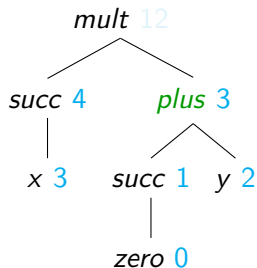


$\zeta(y) = 2$

## Exemple : expressions arithmétiques (évaluation)

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

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

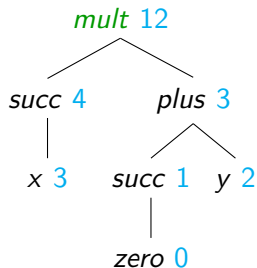


$I_{\mathcal{N}}(\text{plus}) = n, m \mapsto n + m \quad (n, m \mapsto n + m)(1, 2) = 3$

## Exemple : expressions arithmétiques (évaluation)

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

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

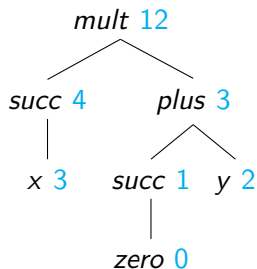


$$I_{\mathcal{N}}(mult) = n, m \mapsto n \times m \quad (n, m \mapsto n \times m)(4, 3) = 12$$

## Exemple : expressions arithmétiques (évaluation)

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

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



## Exemple : chaînes de caractères

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

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

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

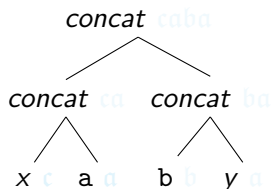
$$I_{str}(c) = c$$

$$I_{str}(concat) = s_1, s_2 \mapsto s_1 s_2$$

## Exemple : chaînes de caractères (évaluation)

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

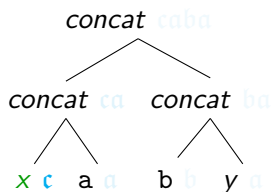
$$\text{eval}(I_{str})(\text{concat}(\text{concat}(x, a), \text{concat}(b, y))) = \mathbf{caba}$$



## Exemple : chaînes de caractères (évaluation)

Soit  $\zeta : x \mapsto c, y \mapsto a$

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

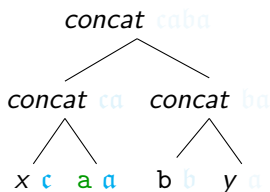


$$\zeta(x) = c$$

## Exemple : chaînes de caractères (évaluation)

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

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



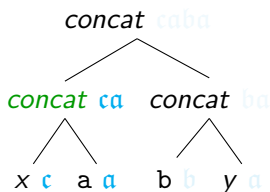
$$I_{\text{str}}(\mathbf{a}) = \mathbf{a}$$



## Exemple : chaînes de caractères (évaluation)

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

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

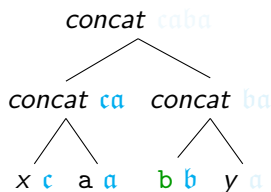


$$I_{str}(\text{concat}) = s_1, s_2 \mapsto s_1 s_2 \quad (s_1, s_2 \mapsto s_1 s_2)(\mathbf{c}, \mathbf{a}) = \mathbf{ca}$$

## Exemple : chaînes de caractères (évaluation)

Soit  $\zeta : x \mapsto c, y \mapsto a$

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

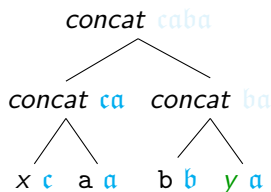


$$I_{\text{str}}(b) = b$$

## Exemple : chaînes de caractères (évaluation)

Soit  $\zeta : x \mapsto c, y \mapsto a$

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

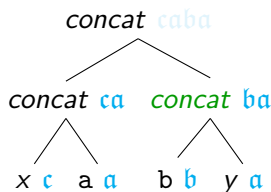


$$\zeta(y) = a$$

## Exemple : chaînes de caractères (évaluation)

Soit  $\zeta : x \mapsto c, y \mapsto a$

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

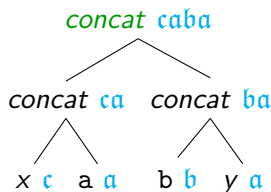


$$I_{str}(\text{concat}) = s_1, s_2 \mapsto s_1 s_2 \quad (s_1, s_2 \mapsto s_1 s_2)(b, a) = ba$$

## Exemple : chaînes de caractères (évaluation)

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

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



$$I_{str}(\text{concat}) = s_1, s_2 \mapsto s_1 s_2 \quad (s_1, s_2 \mapsto s_1 s_2)(\mathbf{ca}, \mathbf{ba}) = \mathbf{caba}$$

# Interprétations non naturelles

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

## Illusoire

→ 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.

- 1 Premier ordre
- 2 Termes : syntaxe
- 3 Termes : sémantique
- 4 Opérations syntaxiques sur les termes**



# Substitutions

Remplacer une partie d'un terme

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

## Definition

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

## Notation

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

$\sigma$  notée  $[t_1/x_1, \dots, t_n/x_n]$

## Appliquer une substitution

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

### Definition (Application de substitution)

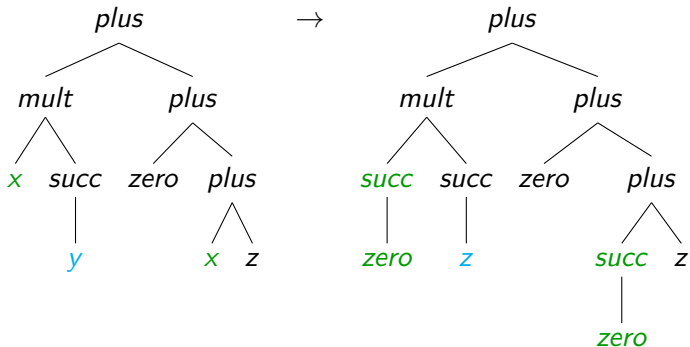
Récursivement définie par :

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

## Exemple

$$\text{plus}(\text{mult}(x, \text{succ}(y)), \text{plus}(\text{zero}, \text{plus}(x, z))) [\text{succ}(\text{zero}) / x, z / y]$$

=

$$\text{plus}(\text{mult}(\text{succ}(\text{zero}), \text{succ}(z)), \text{plus}(\text{zero}, \text{plus}(\text{succ}(\text{zero}), z)))$$


# Motifs

Besoin correspondance terme  $\leftrightarrow$  forme

Décrire la forme

- par un autre terme
- dont les variables représentent des “trous” à remplir

← motif

# 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  $\sigma$  telle que  $m\sigma = t$ .

## Exemple

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

- pour  $t = plus(succ(zero), mult(succ(z), zero))$  :  $\sigma = [zero/x, succ(z)/y]$
- 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)

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

## 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)

- $merge(\sigma_1, \dots, \sigma_n) = \text{fail}$   
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  $merge(\sigma_1, \dots, \sigma_n) =$

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

# Calcul de filtrage de motif

## Definition (match)

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

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



# Unification

## Definition

Soit  $t_1$  et  $t_2$  deux termes.

Existe-il une substitution  $\sigma$  telle que  $t_1\sigma = t_2\sigma$  ?

Filtrage de motif à deux sens

## Exemple

Unifier

$$\textit{plus}(\textit{mult}(x, y), \textit{plus}(y, \textit{succ}(z)))$$

et

$$\textit{plus}(\textit{mult}(u, v), \textit{plus}(w, w))$$

Solution

$$\sigma = [\textit{succ}(z) / w, \textit{succ}(z) / y, \textit{succ}(z) / v, u / x]$$

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

# Equations (syntaxiques) entre termes

Résoudre le problème d'unification

Équations entre termes :

$$t_i \triangleq t'_i$$

Transformer ces équations selon un système de règles

## Notation

- **Une** équation notée  $E$
- Ensemble d'équations noté  $\Psi$  ← l'ordre n'est pas important,  
on omet  $\{$
- Si  $E = t \triangleq t'$  alors  $E\sigma = t\sigma = t'\sigma$
- Si  $\Psi = E_1, \dots, E_n$  alors  $\Psi\sigma = E_1\sigma, \dots, E_n\sigma$

## Résolution des équations : règles

|       |  |                    |   |                                     |
|-------|--|--------------------|---|-------------------------------------|
| (Rev) | $\Psi, t \triangleq x$                                     | $\rightsquigarrow$ | $\Psi, x \triangleq t$                                  | si $t \notin \mathcal{V}$           |
| (Dec) | $\Psi, f(t_1, \dots, t_n) \triangleq f(t'_1, \dots, t'_n)$ | $\rightsquigarrow$ | $\Psi, t_1 \triangleq t'_1, \dots, t_n \triangleq t'_n$ |                                     |
| (Sub) | $\Psi, x \triangleq t$                                     | $\rightsquigarrow$ | $\Psi[t/x], x \triangleq t$                             | si $x \notin V(f(t_1, \dots, t_n))$ |
| (FF)  | $\Psi, f(t_1, \dots, t_n) \triangleq g(t'_1, \dots, t'_m)$ | $\rightsquigarrow$ | fail  | si $f \neq g$                       |
| (CC)  | $\Psi, c_1 \triangleq c_2$                                 | $\rightsquigarrow$ | fail  | si $c_1 \neq c_2$                   |
| (FC)  | $\Psi, f(t_1, \dots, t_n) \triangleq c$                    | $\rightsquigarrow$ | fail  |                                     |
| (CF)  | $\Psi, c \triangleq f(t_1, \dots, t_n)$                    | $\rightsquigarrow$ | fail  |                                     |
| (Cyc) | $\Psi, x \triangleq f(t_1, \dots, t_n)$                    | $\rightsquigarrow$ | fail  | si $x \in V(f(t_1, \dots, t_n))$    |

## 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  $\Psi$ )

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

Substitution  $\sigma_\Psi$  associée à  $\Psi$  :

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

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

### Théorème

*Si  $t \stackrel{\Delta}{=} t' \rightsquigarrow^* \Psi$  et  $\Psi$  en forme résolue alors le problème d'unification de  $t$  avec  $t'$  admet une solution :  $\sigma_\Psi$ .*

*Si  $t \stackrel{\Delta}{=} t' \rightsquigarrow^* \text{fail}$  alors le problème d'unification de  $t$  avec  $t'$  n'admet pas de solution.*

## Exemple

$$\text{succ}(\text{plus}(x, \text{succ}(y))) \triangleq \text{succ}(\text{plus}(u, u))$$

$$\rightsquigarrow \text{plus}(x, \text{succ}(y)) \triangleq \text{plus}(u, u) \quad (\text{Dec})$$

$$\rightsquigarrow x \triangleq u, \text{succ}(y) \triangleq u \quad (\text{Dec})$$

$$\rightsquigarrow x \triangleq u, \text{succ}(y) \triangleq x \quad (\text{Sub})$$

$$\rightsquigarrow x \triangleq u, x \triangleq \text{succ}(y) \quad (\text{Rev})$$

$$\rightsquigarrow \text{succ}(y) \triangleq u, x \triangleq \text{succ}(y) \quad (\text{Sub})$$

$$\rightsquigarrow u \triangleq \text{succ}(y), x \triangleq \text{succ}(y) \quad (\text{Rev})$$

$$\sigma = [\text{succ}(y) / x, \text{succ}(y) / u]$$

## Exemples : échecs

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

$$\rightsquigarrow \text{mult}(x, \text{succ}(y)) \triangleq \text{plus}(u, u) \quad (\text{Dec})$$

$$\rightsquigarrow \text{fail} \quad (\text{FF})$$

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

$$\rightsquigarrow \text{mult}(x, \text{succ}(x)) \triangleq \text{plus}(u, u) \quad (\text{Dec})$$

$$\rightsquigarrow x \triangleq u, \text{succ}(x) \triangleq u \quad (\text{Dec})$$

$$\rightsquigarrow x \triangleq u, \text{succ}(x) \triangleq x \quad (\text{Sub})$$

$$\rightsquigarrow x \triangleq u, x \triangleq \text{succ}(x) \quad (\text{Rev})$$

$$\rightsquigarrow \text{fail} \quad (\text{Cyc})$$