

Aspects Fondamentaux du Calcul : les Termes

Paysage syntaxique: les termes

- ① omniprésent en informatique
 - ② objet les plus importants du calcul symbolique, de la programmation fonctionnelle et logique
 - les atomes et paires de Lisp
 - les termes de Prolog
 - les types concrets de ML
 - ...
- ⇒ structure abstraite qui favorise un style de programmation algébrique concis et clair
- ③ notion de substitution (remplacements)
 - ④ notion de systèmes de réécriture

Notion de mots

- On se donne un ensemble A , appelé un alphabet
- $a \in A$ est un symbole
- Un mot est une suite finie $u = (u_1, \dots, u_n) \in A^n$ d'éléments de A , noté plus simplement u_1, \dots, u_n
- L'ensemble des mots sur A est noté $A^* = \bigcup_{n \geq 0} A^n$
- Soit $u \in A^n$, $|u| = n$ est la longueur de u
- On introduit une opération $.$ de concaténation définie comme :

$$\begin{aligned} \therefore \quad A^p \times A^q &\rightarrow A^{p+q} \\ (u, v) &\mapsto u.v \end{aligned}$$

où $u = (u_1, \dots, u_p)$, $v = (v_1, \dots, v_q)$ et
 $u.v = (u_1, \dots, u_p, v_1, \dots, v_q)$

Le monoïde A^*

L'opération de concaténation $.$ admet un unique élément neutre, le mot vide, noté $\epsilon \in A^0$

La concaténation est associative, i.e., $(u.v).w = u.(v.w)$

La structure $(A^*, ., \epsilon)$ est donc un monoïde, appelé aussi monoïde libre engendré par A

Étant donné un alphabet A , un langage L sur A est une partie de A^*

Termes

La structure des mots est aplatie ce qui induit un pouvoir d'expression limité

La notion de terme vise à structurer les symboles en fonction de leur usage

- Par exemple $1 + x = 3$ contient des symboles $\{1, +, x, =, 3\}$ très différents:
 - les constantes 1 et 3,
 - la variable x ,
 - la fonction $+$,
 - et la relation d'équivalence $=$.
- les termes Prolog: foncteur, variables, constante, Idots
Ex: $f(X, a)$
 - la constante a ,
 - la variable X ,
 - le symbole de fonction f d'arité 2.

Signature

Definition

Une signature est un ensemble Σ de symboles muni d'une fonction $ar : \Sigma \rightarrow \mathbb{N}$ qui associe à chaque symbole son arité. Une fonction est un symbole $f \in \Sigma$ tel que $ar(f) \geq 1$ et une constante sera un symbole d'arité 0.

Definition

On note T_Σ le plus petit sous ensemble de Σ^* tel que :

- $\forall c \in \Sigma, ar(c) = 0 \Rightarrow c \in T_\Sigma$
- $\forall f \in \Sigma, t_1, \dots, t_n \in T_\Sigma, ar(f) = n \geq 1 \Rightarrow ft_1 \dots t_n \in T_\Sigma$ libres

T_Σ est l'ensemble des Σ -termes finis

Pb : ce sont des mots et nous n'avons pas de structure (on représente classiquement les termes par des arbres)

La définition n'est pas constructive

Définitions inductives

On utilise la structure précédente pour faire des preuves par induction sur les termes

Proposition

Soit P une propriété sur les mots de Σ^* . Si elle est vraie pour les constantes de Σ et si elle l'est pour chaque $f \in \Sigma$, d'arité n , on sait prouver qu'elle est vraie pour le mot $ft_1 \cdots t_n$ si elle vraie pour les mots $t_1 \cdots t_n$, alors P est vraie pour tous les termes de T_Σ .

On parle d'induction structurelle.

Induction monotone

Si nous reprenons l'ensemble des termes T_Σ que nous avons défini de manière imprédicative, nous pouvons le caractériser également de la manière suivante, par le biais d'un opérateur sur $\mathcal{P}(\Sigma^*)$ (l'ensemble des parties de Σ^*)

Soit $A \subseteq \Sigma^*$, nous posons

$$\Phi_\Sigma(A) = \bigcup_{f \in \Sigma} \{ft_1 \cdots t_{ar(f)} \mid t_1 \cdots t_{ar(f)} \in A\}$$

Nous avons alors

$$T_\Sigma = \bigcap_{\Phi_\Sigma(A) \subseteq A} A$$

Représentation des termes

Jusqu'à présent les termes ont été introduits comme des mots sur un alphabet.

Une signature sert de base à la construction de ces termes

La structure des termes peut être rendue par une représentation sous forme d'arbre

Nous présentons d'abord la notion d'arbre puis les termes

Étant donné un alphabet A , l'ordre *prefix* est défini sur les mots de A^* par

α *prefix* β si et seulement $\exists \gamma \in A^* \alpha \gamma = \beta$.

Domaine d'un arbre enraciné étiqueté

Notons $\mathbb{N}_\star = \mathbb{N} - \{0\}$ pour lever l'ambiguïté avec la notation de l'ensemble des mots.

Un domaine d'arbre est un sous-ensemble fini non vide D de \mathbb{N}_\star tel que :

- $p \in D \ \& \ q \text{ prefix } p \Rightarrow q \in D$
- $pi \in D \ \& \ j \in \mathbb{N}_\star \ \& \ j < i \Rightarrow pj \in D$

Remarque Tout domaine contient le mot vide.

Arbre enraciné étiqueté

Un arbre étiqueté par A est une application $T : D \rightarrow A$:

- $T(\epsilon)$ est la racine de l'arbre.
- D est le domaine (ou l'ensemble des positions) de T noté $Pos(T)$.
- $T(p)$ s'appelle l'étiquette à la position p .
- Si $p \in Pos(T)$ et $(\forall i \in \mathbb{N}^*) pi \notin Pos(T)$, alors p est une *feuille* de T .

Arbre enraciné étiqueté: exemple

Soit TT tel que $Pos(TT) = \{\epsilon, 1, 2, 21, 22, 221, 222\}$ et :

$$\epsilon \mapsto +$$

$$1 \mapsto x$$

$$2 \mapsto +$$

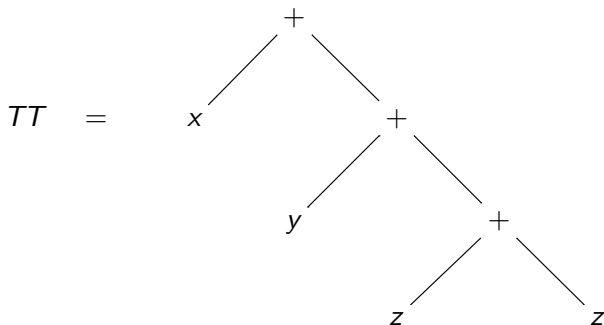
$$21 \mapsto y$$

$$22 \mapsto +$$

$$221 \mapsto z$$

$$222 \mapsto z$$

Arbre enraciné étiqueté



Sous-arbre

$T|_p$ est le sous-arbre à la position p défini par $T|_p(q) = T(pq)$

où $Pos(T|_p) = \{q \mid pq \in Pos(T)\}$

EXERCICE: Décrivez $Pos(TT|_{22})$ et $TT|_{22}$.

Remarque $(T|_p)|_q = T|_{pq}$.

Remplacement

$T[U]_p$ est le remplacement du sous-arbre de T à la position p par l'arbre U , on le définit par :

$$Pos(T[U]_p) = \{q \in Pos(T) \mid \neg(p \text{ prefix } q)\} \cup \{pp' \mid p' \in Pos(U)\}$$

et

$$T[U]_p(q) = \begin{cases} T(q) & \text{si } \neg(p \text{ prefix } q) \\ U(p') & \text{si } q = pp' \text{ avec } p' \in Pos(U) \end{cases}$$

Remplacement

Exemple

Soit $UU = TT[TT]_{21}$.

- Que vaut $UU(212)$?
- Dessiner UU

Remarques

- $T[U[V]_q]_p = T[U]_p[V]_{pq}$.
- $T[U]_p[V]_q = T[V]_q[U]_p$ si p et q sont étrangers¹

Sous-arbre d'un remplacement

Remarque

- Si p et q sont étrangers, alors $(T[U]_p)|_q = T|_q$.
- $(T[U]_p)|_{pp'} = U|_{p'}$.
- $(T[U]_{pp'})|_p = (T|_p)[U]_{p'}$.

Termes: et les variables?

Soit une signature Σ et X un ensemble dit ensemble des variables.

Definition

- Un Σ -terme sur X est un arbre t étiqueté par $\Sigma \cup X$, tel que
 - Si $t(p) \in X$ alors p est une feuille.
 - Si $t(p) \in \Sigma_0$ alors p est une feuille.
 - Si $ar(t(p)) = n$ alors
 - $i \leq n \ \& \ i \in \mathbb{N}_* \Rightarrow pi \in Pos(t)$
 - et $p(n+1) \notin Pos(t)$.
- $T(\Sigma, X)$ est l'ensemble des Σ -termes sur X .
- $Var(t) = \{x \in X \mid (\exists p \in Pos(t)) \ t(p) = x\}$

Definition

- Une Σ -algèbre est un couple $\mathcal{A} = (A, \Sigma^{\mathcal{A}})$ où A est un ensemble dit support de l'algèbre,

$\Sigma^{\mathcal{A}}$ signature

et pour chaque f de Σ d'arité n

$$f^{\mathcal{A}} : A^n \rightarrow A, ar(f) = n$$

Σ -algèbres libres

Definition

- $T(\Sigma, X)$ munie des opérations d'enracinement est une algèbre.
Si $t_1 \dots t_n$ sont des termes et $f \in \Sigma$, $ar(f) = n$ on définit le terme $f(t_1, \dots, t_n)$ obtenu en enracinant par
 - $Pos(f(t_1, \dots, t_n)) = \{\varepsilon\} \cup \bigcup_{i=1}^n iPos(t_i)$
 - $f(t_1, \dots, t_n)(\varepsilon) = f$
 - $f(t_1, \dots, t_n)(ip) = t_i(p)$.

On dit que $T(\Sigma, X)$ est l'algèbre libre sur X .

- $T(\Sigma) = T(\Sigma, \emptyset)$ est l'algèbre initiale.

Morphismes

Definition

- Un morphisme $\phi : T(\Sigma, X) \rightarrow A$ est une application telle que
 - $\phi(x_i) = a_i$
 - $\phi(f(t_1, \dots, t_n)) = f^A(\phi(t_1), \dots, \phi(t_n))$.

Substitutions, domaine, codomaine

Soit X un ensemble dénombrable de variables.

- Une substitution est une application $\sigma : X \rightarrow T(\Sigma, X)$ qui est l'identité presque partout (c-à-d sauf sur un ensemble fini).
- On note : $Dom(\sigma) = \{x \in X \mid \sigma(x) \neq x\}$ le domaine.
- On note : $Range(\sigma) = \{Var(\sigma(x)) \mid x \in Dom(\sigma)\}$ le codomaine.

On étend σ à $T(\Sigma, X)$ en une application $\hat{\sigma}$ (souvent notée σ).

- $\hat{\sigma}(x) = \sigma(x)$,
- $\hat{\sigma}(f(t_1, \dots, t_n)) = f(\hat{\sigma}(t_1), \dots, \hat{\sigma}(t_n))$.

$Sub(T(\Sigma, X))$ est l'ensemble des substitutions.

L'application d'une substitution σ à un terme remplace **simultanément** toutes les occurrences de variables par leur image par σ

Identités et réduction

- Une identité $g \approx d$ est un couple, c'est-à-dire un élément de $T(\Sigma, X) \times T(\Sigma, X)$.
- Soit E un ensemble d'identités, la réduction \xrightarrow{E} est définie par

$$\begin{array}{c} s \xrightarrow{E} t \\ \text{ssi} \\ (\exists g \approx d \in E)(\exists \sigma \in \text{Sub}(T(\Sigma, X)))(\exists p \in \text{Pos}(s)) \\ \text{et} \\ s|_p = \sigma(g) \quad \& \quad t = s[\sigma(d)]_p. \end{array}$$

Notion de système de réécriture

Une règle de réécriture est une identité $g \approx d$ telle que $Var(g) \supseteq Var(d)$.
On écrit $g \rightarrow d$.

- Un système de réécriture est un ensemble de règles de réécriture.
- Un redex de s est une instance (c'est-à-dire un sous-terme de s de la forme $\sigma(g)$ pour $\sigma \in Sub(T(\Sigma, X))$) d'un membre gauche de règle de réécriture $g \rightarrow d$.
- Contracter le redex $s|_p$ à la position p , c'est passer de s à $t = s[\sigma(d)]_p$.