Informatique Fondamentale: fonctions primitives récursives

Plan Chapitre - Fonctions récursives

1. Introduction: formalisme du concept de procédures effectives
2. Les fonctions primitives récursives
3. Fonctions calculables mais non primitives récursives

Introduction

Formalisation du concept de procédures effectives
(vu avec les automates de reconnaissance de langages)

But: considérer les fonctions définies sur les nombres naturels et caractériser celles qui sont calculables par une procédure effective

Fonctions sur les naturels

Considérons: des fonctions dont les arguments et la valeur sont des nombres naturels

Exemples:

- $x + y$
- $x \times y$
- x^y
- $x^2 + y^2$

Fonctions calculables:

- soit des fonctions de base ($+$, $\times$) calculables car on connait un algorithme pour les calculer sur une représentation des entiers (ex., binaire, décimal)
- soit des compositions de fonctions de base (ex: $x \times (y + z)$)

Fonctions non décomposables en fonction de base

- Exemple, factorielle:

$$n! = n \times (n-1) \times (n-2) \ldots \times 2 \times 1$$

  - la factorielle est calculable car s'évalue en fonction de $\times$
  - mais ce n'est pas une simple décompostion car le nombre de multiplications dépend de n
- Nécessité: mécanisme permettant un nombre variable de compositions de base
- Une possibilité: la **récursion**

Factorielle en récursif

- factorielle définie par:

$$0! = 1$$

$$(n+1)! = (n+1) \times n!$$

- procédure de calcul pour factorielle (suite de multiplications)
- applique la deuxième partie jusqu'à 0! qui est fixé à 1
- **récursion primitive**:
  - permet la définition de $f(n+1)$ en fonction de $f(n)$
  - avec $f(0)$, permet de calculer la fonction pour toute valeur de n

Définition

Soit

$$F = \{\mathbb{N}^k \rightarrow \mathbb{N} | k \geq 0\}$$

ensemble des fonctions dont les arguments et la valeur sont des nombres naturels.

Fonctions primitives récursives: sous-ensemble de F.

Une fonction primitive récursive est définie à partir:

- de fonctions de base,
- de la composition,
- et de la règle de récursion.

Fonctions primitives récursives de base

Définition: les **fonctions primitives récursives de base** sont:

- la fonction **zéro**

  $$0()$$

  qui n'a pas d'argument et vaut toujours 0

- les **projections**

  $$\pi_i^k(n_1, \ldots, n_k)$$

  avec $k \geq 1$ et $1 \geq i \geq k$.
  π_i^k a comme valeur le i-ème argument parmi k

  $$\pi_i^k(n_1, \ldots, n_k) = n_i$$

- la fonction **sucesseur**

  $$\sigma(n)$$

  définie par: $\sigma(n) = n + 1$

Composition: définition

Définition Soient:

- g une fonction à l arguments
- $h_1, \ldots, h_l$ l fonctions à k arguments
- $\bar{n} = n_1 \ldots, n_k$

Alors, la composition de g et de $h_1, \ldots, h_l$ est la fonction $N^k \rightarrow N$ définie par:

$$f(\bar{n}) = g(h_1(\bar{n}), \ldots, h_l(\bar{n}))$$

Récursion primitive: définition

Définition Soient:

- g une fonction à k arguments
- h une fonction à $k+2$ arguments

Alors, la fonction f à $k+1$ arguments telle que:

$$f(\bar{n}, 0) = g(\bar{n})$$

$$f(\bar{n}, m+1) = h(\bar{n}, m, f(\bar{n}, m))$$

est la fonction définie à partir de g et h par **récursion primitive**.

Propriétés

- Si les fonctions g et h utilisées pour définir une fonction f par récursion primitive sont calculables par une procédure effective, alors f est aussi calculable.
- La récursion est plus générale dans les langages de programmation et ne garantit pas que toute fonction définie par récursion est calculable. L'exemple suivant ne termine pas:

$$f(0) = 0$$

$$f(m) = 1 + f(m)$$

La récursion primitive ne permet pas ce genre de définition car $f(m+1)$ doit être défini en fonction de $f(m)$.

Fonctions primitives récursives

Définition: les fonctions primitives récursives sont:

- les fonctions primitives récursives de base
- toutes fonctions obtenues à partir des fonctions primitives récursives de base par un nombre quelconque d'applications de la composition et de la récursion primitive.

Exemples

Un grand nombre de fonctions courantes sont primitives récursives.

- les fonctions constantes $j() = j$ sont primitives récursives. En effet:

  $$j() = \sigma(\sigma(\ldots\sigma(0))))$$

  avec j fois σ.
- la fonction d'addition $plus(n_1, n_2)$ dont la valeur est la somme arithmétique de n_1 et n_2 est primitive récursive car:

  $$plus(n_1, 0) = \pi_1^1(n_1)$$

  $$plus(n_1, n_2 + 1) = \sigma(plus(n_1, n_2))$$

Exemples

Pour la suite, simplication: omettre les projections. Donc, notez n, au lieu de $\pi_1^1(n)$. Ou notez n_2 au lieu de $\pi_2^3(n_1, n_2, n_3)$.
plus devient:

$$plus(n_1, 0) = n_1$$

$$plus(n_1, n_2 + 1) = \sigma(plus(n_1, n_2))$$

Ex: *plus* appliqué à 7 et 4:

$$\begin{array}{rcl} plus(7,4) & = & plus(7, 3+1) \\ & = & \sigma(plus(7,3)) \\ & = & \sigma(\sigma(plus(7,2))) \\ & = & \sigma(\sigma(\sigma(plus(7,1)))) \\ & = & \sigma(\sigma(\sigma(\sigma(plus(7,0))))) \\ & = & \sigma(\sigma(\sigma(\sigma(7)))) \\ & = & 11 \end{array}$$

Exemples

- la fonction produit ($n \times m$ ou $times(n, m)$) est primitive récursive car elle se définit par récursion primitive sur *plus*:

$$\begin{array}{lcl} n \times 0 & = & 0 \\ n \times (m+1) & = & n + (n \times m) \end{array}$$

- la fonction puissance n^m se définit par récursion primitive sur $\times$:

$$\begin{array}{lcl} n^0 & = & 1 \\ n^{m+1} & = & n \times n^m \end{array}$$

- par analogie, on définit une nouvelle fonction en appliquant la récursion primitive à la fonction puissance:

$$\begin{array}{lcl} n \uparrow\uparrow 0 & = & 1 \\ n \uparrow\uparrow m+1 & = & n^{n \uparrow\uparrow m} \end{array}$$

qui est: $n \uparrow\uparrow m = n^{n^{\cdots^{n}}}$, de hauteur m

Exemples

- la fonction $n \uparrow\uparrow\uparrow m$ est primitive récursive:

$$
\begin{array}{rcl}
n \uparrow\uparrow\uparrow 0 & = & 1 \\
n \uparrow\uparrow\uparrow m+1 & = & n \uparrow\uparrow n \uparrow\uparrow\uparrow m
\end{array}
$$

  et ainsi de suite ...

- factorielle est primitive récursive:

$$
\begin{array}{rcl}
0! & = & 1 \\
(n+1)! & = & (n+1).n!
\end{array}
$$

- la fonction prédecesseur (-1) est primitive récursive en considérant que le prédecesseur de 0 est 0 (sur les entiers naturels), c-à-d, $pred(m) = 0$ si $m = 0$, $m - 1$ sinon:

$$
\begin{array}{rcl}
pred(0) & = & 0 \\
pred(m+1) & = & m
\end{array}
$$

Exemples

- la fonction différence (-) est primitive récursive en considérant que $n - m = 0$ si $n \leq m$, $n - m$ sinon. Elle se définit par récursion primitive comme suit:

$$\begin{array}{lcl} n - 0 & = & n \\ n - (m+1) & = & pred(n - m) \end{array}$$

- la fonction signe (sg) vaut 0 si son argument est zéro, 1 sinon:

$$\begin{array}{lcl} sg(m) & = & 0 \text{ si } m = 0 \\ sg(m) & = & 1 \text{ si } m > 0 \end{array}$$

  Par récursion primitive:

$$\begin{array}{lcl} sg(0) & = & 0 \\ sg(m+1) & = & 1 \end{array}$$

Exemples

- le produit borné par m d'une fonction $g(\bar{n}, i)$:

$$\Pi_{i=0}^{m} g(\bar{n}, i)$$

Si g est primitive récursive,

$$f(\bar{n}, m) = \Pi_{i=0}^{m} g(\bar{n}, i)$$

l'est aussi, car:

$$\begin{array}{lcl} f(\bar{n}, 0) & = & g(\bar{n}, 0) \\ f(\bar{n}, m+1) & = & f(\bar{n}, m) \times g(\bar{n}, m+1) \end{array}$$

De manière stricte, on devrait écrire:

$$f(\bar{n}, m+1) = h(\bar{n}, m, f(\bar{n}, m))$$

avec

$$h(n_1, n_2, n_3) = times(g(\pi_1^3(n_1, n_2, n_3), \sigma(\pi_2^3(n_1, n_2, n_3))), \pi_3^3(n_1, n_2, n_3))$$

Exemples: Ackermann

Dans la fonction $\uparrow\uparrow \ldots \uparrow$ on peut considérer le nombre de flêches k comme un paramètre définissant une famille de fonctions. La récursion est:

$$f(k+1, n, m+1) = f(k, n, f(k+1, n, m))$$

La fonction à partir de cette récursion est la fonction d'Ackermann suivante (sans n qui n'a aucun rôle dans la récursion):

$$\begin{array}{lcl} Ack(0, m) & = & m+1 \\ Ack(k+1, 0) & = & Ack(k, 1) \\ Ack(k+1, m+1) & = & Ack(k, Ack(k+1, m)) \end{array}$$

- pas la forme d'une définition primitive récursive
- on peut montrer qu'il est impossible d'avoir une définition équivalente primitive récursive
- Ackermann n'est donc pas primitive récursive
- mais Ackermann est calculable
- croissance très rapide, plus que les fonctions primitives récursives

Exemples: fonction m

Soit la fonction m:

$$\begin{array}{lcl} m(0,p) & = & 0 \\ m(\sigma(n),0) & = & 0 \\ m(\sigma(n),\sigma(p)) & = & \sigma(m(n,p)) \end{array}$$

- m est-elle récursive primitive?

Fonctions calculables mais non primitives récursives

- Les fonctions primitives récursives sont toutes calculables par une procédure effective
- Il y a des fonctions qui ne sont pas primitives récursives
- nombre dénombrable de fonctions primitives récursives, et non dénombrable de fonctions (l'ensemble des prédicats à un argument sur les naturels; est identique à $\mathcal{P}(\mathbb{N})$ qui est non dénombrable)
- Théorème: il y a des fonctions calculables qui ne sont pas primitives récursives

Démonstration/Exemple

- soit $f_0, f_1, f_2, \ldots$ une suite de fonctions primitives récursives à k arguments
- soit le tableau:

$$
\begin{array}{c|cccccc}
A & 0 & 1 & 2 & \ldots & j & \ldots \\
\hline
f_0 & f_0(0) & f_0(1) & f_0(2) & \ldots & f_0(j) & \ldots \\
f_1 & f_1(0) & f_1(1) & f_1(2) & \ldots & f_1(j) & \ldots \\
f_2 & f_2(0) & f_2(1) & f_2(2) & \ldots & f_2(j) & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\
f_i & f_i(0) & f_i(1) & f_i(2) & \ldots & f_i(j) & \ldots \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \ddots
\end{array}
$$

- $A[i,j]$ est le naturel $f_i(j)$
- soit $g(n) = f_n(n) + 1 = A[n,n] + 1$
- $g(n)$ n'est pas primitive récursive (car change de f_k pour chaque valeur de k dans $g(k)$.
- g est néanmoins calculable