Sémantique dénotationnelle

Frédéric Boulanger

CentraleSupélec

Rappel : sémantique opérationnelle

- La sémantique opérationnelle à grands pas établit une relation entre des paires $(\mathrm{instruction}, \mathrm{état})$ et des états.
- $(\mathrm{instr}, \mathrm{s}) \rightsquigarrow \mathrm{s}'$ siginifie qu'en exécutant instr dans l'état s, on peut arriver dans l'état s'.
- Cette sémantique permet de dérouler la séquence d'états par lesquels passe la machine d'exécution lorsqu'on exécute un programme.

Sémantique à grands pas de Niklaus

CentraleSupélec

Sémantique dénotationnelle

Ce style de sémantique associe à chaque instruction une relation entre états.

On cherche donc à définir une fonction $[\![\_]\!]$ telle que :

$$[\![\text{instr}]\!] = \{(\text{pre}, \text{post}) | (\text{pre}, \text{instr}) \rightsquigarrow \text{post}\}$$

Sémantique dénotationnelle naïve de Niklaus

Sémantique de la récursion

Comment définir une fonction récursive ?

- $\mathrm{f}(n) = \text{if } n = 0 \text{ then } 1 \text{ else } n * \mathrm{f}(n-1)$
- $\mathrm{g}(n) = \text{if } n = 0 \text{ then } 1 \text{ else } \dfrac{\mathrm{g}(n+1)}{n+1}$
- $\mathrm{h}(n) = \text{if } n = 1 \text{ then } 1 \text{ elseif } n \equiv 0[2] \text{ then } \mathrm{h}(\dfrac{n}{2}) \text{ else } \mathrm{h}(3n+1)$

CentraleSupélec

Comment définir une fonction récursive ?

- $\mathrm{f}(n) = \text{if } n = 0 \text{ then } 1 \text{ else } n * \mathrm{f}(n-1)$
- $\mathrm{g}(n) = \text{if } n = 0 \text{ then } 1 \text{ else } \dfrac{\mathrm{g}(n+1)}{n+1}$
- $\mathrm{h}(n) = \text{if } n = 1 \text{ then } 1 \text{ elseif } n \equiv 0[2] \text{ then } \mathrm{h}(\frac{n}{2}) \text{ else } \mathrm{h}(3n+1)$

Fonctionnelles :

- $\mathrm{F}(f) = \lambda n.\, \text{if } n = 0 \text{ then } 1 \text{ else } n * \mathrm{f}(n-1)$
- $\mathrm{G}(g) = \lambda n.\, \text{if } n = 0 \text{ then } 1 \text{ else } \dfrac{\mathrm{g}(n+1)}{n+1}$
- $\mathrm{H}(h) = \lambda n.\, \text{if } n = 1 \text{ then } 1 \text{ elseif } n \equiv 0[2] \text{ then } \mathrm{h}(\frac{n}{2}) \text{ else } \mathrm{h}(3n+1)$

Fonctions récursives

CentraleSupélec

Points fixes et monotonie

Les exemples précédents portent sur des fonctions partiellement définies.
Il est difficile de leur appliquer le théorème de Knaster-Tarski :

Théorème de Knaster-Tarski

Toute fonction f croissante sur un treillis complet admet au moins un point fixe (un élément x du treillis tel que $f(x) = x$), et l'ensemble des points fixes de f est lui-même un treillis complet.
f admet donc notamment un plus petit et un plus grand point fixe.

Un treillis est un ensemble T muni d'un ordre partiel tel que toute paire d'éléments de T admet une borne supérieure et une borne inférieure.
Exemples :

- l'ensemble des parties d'un ensemble, muni de la relation d'inclusion : inf $= \bigcap$ et sup $= \bigcup$
- l'ensemble des entiers naturels, muni de la relation *divise* : inf $=$ PGCD et sup $=$ PPCM

Points fixes d'un foncteur sur des relations

- L'ensemble des relations entre éléments d'un ensemble, muni de la relation d'inclusion, est un treillis complet.
- On peut considérer une fonction partielle comme une relation entre les éléments d'un ensemble.
- On peut alors utiliser le théorème de Knaster-Tarski pour définir une fonction récursive comme le plus petit point fixe d'un foncteur monotone sur les relations.

Définition récursive de la factorielle

Retour à la sémantique du `while`

- La sémantique d'une instruction est une relation sur les états
- La sémantique du `while` est définie récursivement
- On la définit comme plus petit point fixe d'un certain foncteur.

Sémantique dénotationnelle de Niklaus