Sémantique axiomatique

Frédéric Boulanger

CentraleSupélec

Rappel : sémantique opérationnelle

- La sémantique opérationnelle à grands pas établit une relation entre des paires (instruction, état) et des états.
- $(\text{instr}, s) \rightsquigarrow s'$ 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.

CentraleSupélec

Sémantique dénotationnelle

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

On définit une fonction $[\![\_]\!]$ telle que :

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

Cette sémantique est compositionnelle : la dénotation d'une suite d'instructions s'obtient par composition de leurs dénotations.

CentraleSupélec

Sémantique axiomatique

On cherche à définir la manière dont l'exécution d'un programme affecte la validité de propositions portant sur l'état de la machine d'exécution.

Hoare : sémantique pré/post

$$\vDash \{\varphi\}\ \text{prog}\ \{\psi\}$$

Si φ est vraie dans un état s,
alors ψ est vraie dans tout état s' tel que $(s,\ \text{prog}) \rightsquigarrow s'$.

CentraleSupélec

Sémantique axiomatique

On cherche à définir la manière dont l'exécution d'un programme affecte la validité de propositions portant sur l'état de la machine d'exécution.

Hoare : sémantique pré/post

$$\vDash \{\varphi\}\ \text{prog}\ \{\psi\}$$

Si φ est vraie dans un état s,
alors ψ est vraie dans tout état s' tel que $(s,\ \text{prog}) \rightsquigarrow s'$.

Substitution

Pour traiter l'affectation, il faut pouvoir substituer une expression à une variable :

$$P\,[\text{with } expr \text{ for } x]$$

est la proposition P dans laquelle on a substitué $expr$ à x.
Exemple : $(\text{even}\, x)[\text{with } 2 \text{ for } x] = \text{even}\, 2$

Règles sémantiques

On indique pour chaque instruction comment elle affecte les propositions :

$$\text{Nop} \vdash \{P\}\ \text{Nop}\ \{P\}$$

$$\text{VDecl} \vdash \{P\,[\text{with}\ (N\ 0)\ \text{for}\ v]\}\ \text{var v}\ \{P\}$$

...

Dans cette dernière règle, on anticipe l'effet de la déclaration en substituant 0 à v dans P.

Validité et dérivabilité

- $\vDash \{\varphi\}P\{\psi\}$: le triplet est valide
- $\vdash \{\varphi\}P\{\psi\}$: le triplet est dérivable selon les règles sémantiques

Sémantique axiomatique de Niklaus

CentraleSupélec

Correction et complétude

Correction (*Soundness*)

La sémantique axiomatique est correcte si toute dérivation est valide :

$$\vdash \{P\}\ \text{instr}\ \{Q\} \implies \vDash \{P\}\ \text{instr}\ \{Q\}$$

Complétude (*Completeness*)

Elle est complète si tout triplet valide peut être dérivé :

$$\vDash \{P\}\ \text{instr}\ \{Q\} \implies \vdash \{P\}\ \text{instr}\ \{Q\}$$

Pour démontrer ceci, il faut introduire la notion de plus faible précondition.
Preuve de la correction et de la complétude

CentraleSupélec

Exemple : calcul du PGCD par soustractions successives

```
while (x <> y) {
  if x < y {
    y := y - x
  } else {
    x := x - y
  }
}
```

Preuve de l'algorithme

CentraleSupélec