Complétude de la logique de Hoare

Leçons : 927 ; 930

Références : NIELSON & NIELSON, *Semantics with applications* (p.186) et WINSKEL, *The Formal Semantics of Programming Languages : An Introduction*

Prérequis :

- Connaître différentes sémantiques du langage IMP (voir Appendice)

Définition 1. On notera

$$\vdash_p \{P\}S\{Q\}$$

s'il existe un arbre de dérivation aboutissant à $\{P\}S\{Q\}$ en utilisant les règles de la logique de Hoare (notée $[\dots]_H$). (voir Appendice)

Définition 2. On notera

$$\vDash_p \{P\}S\{Q\}$$

si $\forall s$ tel que $P\ s = tt$ on ait $\langle S, s\rangle \to s'$ implique $Q\ s' = tt$

Dans la deuxième définition, on utilise les règles de la sémantique opérationnelle à grands pas (notée $[\dots]_{NS}$ pour "sémantique naturelle") (voir Appendice).

Introduction :

La logique de Hoare est un outil qui permet de prouver la correction de certains programmes. Pour cela, on a des règles de dérivation que l'on va utiliser pour démontrer (à l'aide d'une preuve) la correction d'un programme et une sémantique (celle opérationnelle à grands pas) qui va être le sens que l'on donne au programme. On veut donc prouver un théorème de correction et complétude face aux règles que l'on s'est donné. La correction consiste à vérifier que les règles que l'on utilise vérifient bien la sémantique que l'on a (on n'abordera pas cet aspect dans ce développement). La complétude consiste à trouver, pour tout programme ayant une certaine sémantique, un arbre de dérivation des règles qui démontre le résultat (c'est ce que l'on va faire dans ce développement).

Théorème 1. Pour tous prédicats P, Q et toute instruction S,

$$\vdash_p \{P\}S\{Q\} \iff \vDash_p \{P\}S\{Q\}$$

Démonstration de la complétude. $\boxed{\Longleftarrow}$

Définition 3. $\mathrm{wlp}(S, Q)$ est la plus faible précondition donnant Q en exécutant S, i.e.

$$\mathrm{wlp}(S, Q)\ s = tt \text{ si et seulement si } \forall s'\ \langle S, s\rangle \to s' \text{ implique } Q\ s' = tt$$

Lemme 1. $\vDash_p \{P\}S\{Q\}$ implique $(P \implies \mathrm{wlp}(S, Q))$.

Démonstration du lemme. On suppose $P\ s = tt$, on veut prouver que $\mathrm{wlp}(S, Q) = tt$. Par définition de $\vDash_p$ et puisque $P\ s = tt$, on a $\langle S, s\rangle \to s'$ implique $Q\ s' = tt$. D'où $\mathrm{wlp}(S, Q)\ s = tt$ par définition de wlp. □

On veut prouver que $\vdash_p \{P\}S\{Q\}$
On va pouvoir utiliser la règle de la conséquence [cons $_H$] avec $P \implies \text{wlp}(S, Q)$.
Donc il faut prouver que $\vdash_p \{\text{wlp}(S, Q)\}S\{Q\}$.

Montrons que $\vdash_p \{\text{wlp}(S, Q)\}S\{Q\}$ par induction sur la structure de S.

→ cas *skip* : On a $\text{wlp}(\mathsf{skip}, Q) = Q$

Preuve. — On suppose que $Q\ s = tt$. Si on a $\langle \mathsf{skip}, s\rangle \to s'$ c'est que $s' = s$ (par définition de la règle [skip $_{NS}$]) d'où $Q\ s' = tt$ et donc $\text{wlp}(\mathsf{skip}, Q)\ s = tt$ par définition de wlp. Ainsi

$$Q \implies \text{wlp}(\mathsf{skip}, Q)$$

— On suppose que $\text{wlp}(\mathsf{skip}, Q)\ s = tt$. Si on a $\langle \mathsf{skip}, s\rangle \to s'$, c'est aussi que $s' = s$ et donc $Q\ s = tt$. Ainsi

$$\text{wlp}(\mathsf{skip}, Q) \implies Q$$ [1]

□

Or on a $\vdash_p \{Q\}\mathsf{skip}\{Q\}$ par la règle [skip $_H$] d'où

$$\vdash_p \{\text{wlp}(\mathsf{skip}, Q)\}\mathsf{skip}\{Q\}$$

→ cas *affectation* : On a $\text{wlp}(x := a, Q) = Q[x \mapsto \mathcal{A}[\![a]\!]]$

Preuve. — On suppose que $Q[x \mapsto \mathcal{A}[\![a]\!]]\ s = tt$. Si on a $\langle x := a, s\rangle \to s'$ c'est que $s' = s[x \mapsto \mathcal{A}[\![a]\!]]$ (par définition de la règle de l'affectation [ass $_{NS}$]) d'où $Q\ s' = Q[x \mapsto \mathcal{A}[\![a]\!]]\ s = tt$ et donc $\text{wlp}(x := a, Q)\ s = tt$ par définition de wlp. Ainsi

$$Q[x \mapsto \mathcal{A}[\![a]\!]] \implies \text{wlp}(x := a, Q)$$

— On suppose que $\text{wlp}(x := a, Q)\ s = tt$. Si on a $\langle x := a, s\rangle \to s'$, c'est aussi que $s' = s[x \mapsto \mathcal{A}[\![a]\!]]$ et donc $Q\ s' = Q[x \mapsto \mathcal{A}[\![a]\!]]\ s = tt$. Ainsi

$$\text{wlp}(x := a, Q) \implies Q[x \mapsto \mathcal{A}[\![a]\!]]$$ [2]

□

Or on a $\vdash_p \{Q[x \mapsto \mathcal{A}[\![a]\!]]\}x := a\{Q\}$ par la règle de l'affectation [ass $_H$] d'où

$$\vdash_p \{\text{wlp}(x := a, Q)\}x := a\{Q\}$$

→ cas *composition* : Par induction, on a

$$\vdash_p \{\text{wlp}(S_2, Q)\}S_2\{Q\} \text{ et } \vdash_p \{\text{wlp}(S_1, \text{wlp}(S_2, Q))\}S_1\{\text{wlp}(S_2, Q)\}$$

Montrons que $\text{wlp}(S_1; S_2, Q) \implies \text{wlp}(S_1, \text{wlp}(S_2, Q))$

On suppose que $\text{wlp}(S_1; S_2, Q)\ s = tt$. Ainsi par définition si on a $\langle S_1; S_2, s\rangle \to s'$ alors $Q\ s' = tt$. Mais si on a $\langle S_1; S_2, s\rangle \to s'$, c'est que, par la règle de la composition [comp $_{NS}$], l'on avait

$$\langle S_1, s\rangle \to s'' \text{ et } \langle S_2, s''\rangle \to s'$$

Donc $\text{wlp}(S_2, Q)\ s'' = tt$ (par la deuxième assertion) et ainsi $\text{wlp}(S_1, \text{wlp}(S_2, Q))\ s = tt$ (par la première assertion). On a utilisé le fait que ce sont des wlp.

1. en réalité, cette implication est suffisante, en utilisant la règle de la conséquence(hoare)
2. de même, cette implication est suffisante, en utilisant la règle de la conséquence [cons $_H$]

On utilise la règle de la conséquence(hoare) :
Comme $\vdash_p \{\text{wlp}(S_1, \text{wlp}(S_2, Q))\}S_1; S_2\{Q\}$ par la règle de la composition [comp $_H$] et $\text{wlp}(S_1; S_2, Q) \implies \text{wlp}(S_1, \text{wlp}(S_2, Q))$, on a donc

$$\vdash_p \{\text{wlp}(S_1; S_2, Q)\}S_1; S_2\{Q\}$$

→ cas *if* : Par induction, on a

$$\vdash_p \{\text{wlp}(S_1, Q)\}S_1\{Q\} \text{ et } \vdash_p \{\text{wlp}(S_2, Q)\}S_2\{Q\}$$

On pose $P = \big(\mathcal{B}[\![b]\!] \wedge \text{wlp}(S_1, Q)\big) \vee \big(\neg B[\![b]\!] \wedge \text{wlp}(S_2, Q)\big)$
Ainsi on a $\vdash_p \{P \wedge \mathcal{B}[\![b]\!]\}S_1\{Q\}$ et $\vdash_p \{P \wedge \neg\mathcal{B}[\![b]\!]\}S_2\{Q\}$. Donc, par la règle [if $_H$], on a

$$\vdash_p \{P\}\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2\{Q\}$$

Montrons que $\text{wlp}(\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2, Q) \implies P$

> On suppose que $\text{wlp}(\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2, Q)\ s = tt$.
> Ainsi par définition si on a $\langle \textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2, s\rangle \to s'$ alors $Q\ s' = tt$.
> — si $\mathcal{B}[\![b]\!]\ s = tt$, alors on a $\langle S_1, s\rangle \to s'$ d'où $\text{wlp}(S_1, Q)\ s = tt$
> — si $\mathcal{B}[\![b]\!]\ s = ff$, alors on a $\langle S_2, s\rangle \to s'$ d'où $\text{wlp}(S_2, Q)\ s = tt$
> Ainsi $P\ s = tt$.

On utilise la règle de la conséquence [cons $_H$] :
Comme $\vdash_p \{P\}\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2\{Q\}$ par la règle [if $_H$] et
$\text{wlp}(\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2, Q) \implies P$, on a donc

$$\vdash_p \{\text{wlp}(\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2, Q)\}\textsf{if } b \textsf{ then } S_1 \textsf{ else } S_2\{Q\}$$

→ cas *while* : On veut montrer que $\vdash_p \{\text{wlp}(\textsf{while } b \textsf{ do } S, Q)\}\textsf{while } b \textsf{ do } S\{Q\}$.
On note $P = \text{wlp}(\textsf{while } b \textsf{ do } S, Q)$.
Par induction , on a $\vdash_p \{\text{wlp}(S, P)\}S\{P\}$.
Montrons que $(\neg\mathcal{B}[\![b]\!] \wedge P) \implies Q$

> On suppose que $(\neg\mathcal{B}[\![b]\!] \wedge P)\ s = tt$, c'est-à-dire $\mathcal{B}[\![b]\!]\ s = ff$ et $P\ s = \text{wlp}(\textsf{while } b \textsf{ do } S, Q)\ s = tt$. Donc si on a $\langle \textsf{while } b \textsf{ do } S, s\rangle \to s'$, c'est que $s = s'$ et donc $Q\ s = tt$. Ainsi
> $$(\neg\mathcal{B}[\![b]\!] \wedge P) \implies Q$$

Montrons que $(\mathcal{B}[\![b]\!] \wedge P) \implies \text{wlp}(S, P)$

> On suppose que $(\mathcal{B}[\![b]\!] \wedge P)\ s = tt$, c'est-à-dire $\mathcal{B}[\![b]\!]\ s = tt$ et $P\ s = \text{wlp}(\textsf{while } b \textsf{ do } S, Q)\ s = tt$. Donc si on a $\langle \textsf{while } b \textsf{ do } S, s\rangle \to s'$, c'est que, par les règles [while $_{NS}$]), on avait
> $$\langle S, s\rangle \to s'' \text{ et } \langle \textsf{while } b \textsf{ do } S, s''\rangle \to s'$$
> On a donc
> $$Q\ s' = tt$$
> car $\text{wlp}(\textsf{while } b \textsf{ do } S, Q)\ s = tt$.
> De ce fait, on a
> $$\text{wlp}(\textsf{while } b \textsf{ do } S, Q)\ s'' = tt$$
> car $Q\ s' = tt$.

Et donc

$$\text{wlp}(S, P) = \text{wlp}(S, \text{wlp}(\textsf{while } b \textsf{ do } S, Q))\ s = tt$$

car $\text{wlp}(\textsf{while } b \textsf{ do } S, Q)\ s'' = tt$.

Ce qui prouve que

$$(\mathcal{B}[\![b]\!] \wedge P) \implies \text{wlp}(S, P)$$

On a dit que, par induction, on avait

$$\vdash_p \{\text{wlp}(S, P)\} S \{P\}$$

On utilise la règle de la conséquence [cons $_H$] pour avoir :

$$\vdash_p \{\mathcal{B}[\![b]\!] \wedge P)\} S \{P\}$$

Ainsi, par la règle du while [while $_H$], on a :

$$\vdash_p \{P\} \textsf{while } b \textsf{ do } S \{\neg\mathcal{B}[\![b]\!] \wedge P\}$$

Puis, de nouveau par la règle de la conséquence [cons $_H$], on a :

$$\vdash_p \{P\} \textsf{while } b \textsf{ do } S \{Q\}$$

□

Remarques :

On ne parle ici que de correction partielle, d'où le p en indice du symbole $\vdash$, c'est-à-dire que l'on n'a pas d'information sur la terminaison. Il existe des règles pour la correction totale avec un théorème de complétude associé.

Pour la correction, la preuve se fait par induction sur la structure de l'arbre. (Nielson & Nielson (p.184))

Astuces de l'agrégatif :

Attention, il faut bien maitriser toutes les règles que l'on utilise (que ce soit celles de la logique de Hoare ou celles de la sémantique opérationnelle à grands pas) pour expliquer de manière claire au jury et ne pas perdre du temps dessus. (C'est déjà un développement relativement long).

Je fais un point sur les bases de quelques sémantiques que l'on peut mettre sur le langage IMP en Appendice.