Complétude partielle de la logique de Hoare

Julie Parreaux

2018-2019

Référence du développement : Nielson & Nielson [1, p.223]

Leçons où on présente le développement : 927 (Analyse : terminaison et correction) ; 930 (Sémantique).

1 Introduction

La logique de Hoare est une sémantique axiomatique. Elle nous permet d'automatiser les preuves de programmes et notamment leur correction. En effet, deux versions de cette logique existe : la logique partielle et la logique complète. La logique partielle que nous utilisons ici nous permet de garantir les propriétés sur un programme sous l'hypothèse que celui-ci termine. Autrement dit, elle ne nous permet pas de prouver la terminaison des algorithmes. La logique complète, elle prouve égalament la correction de notre programme (nous ne l'évoquons pas ici).

La logique partielle de Hoare est complète et correcte. Ici, on montre que la complétude de la logique de Hoare. Nous rappelons le système d'inférence de la logique partielle de Hoare. Ensuite, on montrera la complétude partielle, puis la correction.

Remarques sur le développement

Ce développement est long, il faut alors faire des choix dans les propriétés démontrées. De plus, il demande une bonne compréhension de la logique de Hoare ainsi que de la sémantique à grands pas.

1. Présentation des règles de la logique et de la sémantique (à écrire dans un coin du tableau).
2. Preuve du théorème par induction (on ne rédige pas les cas de l'affectation et de la conditionnelle).

2 Logique de Hoare partielle

2.1 Présentation de la logique

La logique de Hoare partielle [1, p.221] permet de garantir des propriétés sur le programme uniquement si celui-ci termine. On définit les triplets de Hoare qui sont alors une formule dans cette logique puis les formules étendues qui nous permettent de définir les règles d'inférences.

Définition. Un triplet de Hoare est la donnée d'une précondition P, d'une instruction S et d'une post-condition Q. On le note $\{P\}\ S\ \{Q\}$.

Définition. Comme P et Q sont des formules, on définit le langage des formules étendues suivant.

— $P = P_1 \wedge P_2$ si $\forall s \in State$, Ps si et seulement si $P_1 s$ et $P_2 s$.

— $P = P_1 \vee P_2$ si $\forall s \in State$, Ps si et seulement si $P_1 s$ ou $P_2 s$.

— $P = \neg P_1$ si $\forall s \in State$, Ps si et seulement si $\neg(P_1 s)$.

— $P = P_1[x \mapsto \mathcal{A}[\![a]\!]]$ si $\forall s \in State$, Ps si et seulement si $P_1\ (s[x \mapsto \mathcal{A}[\![a]\!]_s])$.

— $P = P_1 \Rightarrow P_2$ si $\forall s \in State$, Ps implique $P_2 s$.

Définition. On définit la logique de Hoare pour les instructions du langage IMP à l'aide des règles d'interférence. Soit S une instruction et pour toute précondition P, P' et postcondition Q, Q', on a :

$[skip_p]$ $\quad \{P\}\ \mathsf{skip}\ \{P\}$ $\qquad$ $[ass_p]$ $\quad \{P[x \mapsto \mathcal{A}[\![a]\!]]\}\ x := a\ \{P\}$

$[comp_p]$ $\quad \dfrac{\{P\}\ S_1\ \{Q\} \qquad \{Q\}\ S_2\ \{R\}}{\{P\}\ S_1; S_2\ \{R\}}$ $\qquad$ $[if_p]$ $\quad \dfrac{\{\mathcal{B}[\![b]\!] \wedge P\}\ S_1\ \{Q\} \qquad \{\neg\mathcal{B}[\![b]\!] \wedge P\}\ S_2\ \{Q\}}{\{P\}\ \mathsf{if}\ b\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\ \{Q\}}$

$[while_p]$ $\quad \dfrac{\{\mathcal{B}[\![b]\!] \wedge P\}\ S\ \{P\}}{\{P\}\ \ \mathsf{while}\ b\ \mathsf{do}\ S\ \{\neg\mathcal{B}[\![b]\!] \wedge P\}}$ $\qquad$ $[cons_p]$ $\quad \dfrac{\{P'\}\ S\ \{Q'\}}{\{P\}\ S\ \{Q\}}$ si $P \Rightarrow P'$ et $Q' \Rightarrow Q$

Définition. On notera $\vdash_p \{P\}\ S\ \{Q\}$ s'il existe une preuve donnée par un arbre d'interférence tel que $\{P\}\ S\ \{Q\}$ soit à la racine et que toutes ses feuilles sont des axiomes.

Proposition. *Pour toute instruction S et propriété P, on a* $\vdash_p \{P\}\ S\ \ \{True\}$.

Démonstration. On applique la règle $[\text{CONS}_p]$ avec $True \Rightarrow Q$ qui est toujours vraie. □

Définition. Deux instructions S_1 et S_2 sont sémantiquement équivalente si et seulement si $\forall P, Q, \vdash_p \{P\}\ S_1\ \{Q\}$ est équivalent à $\vdash_p \{P\}\ S_2\ \{Q\}$.

Définition. (Validité pour la sémantique à grands pas) $\models_p \{P\}\ S\ \{Q\}$ si et seulement si $\forall s \in State$ tel que $Ps = tt$, si $\langle S, s\rangle \to s'$ alors $Qs' = tt$.

2.2 Correction de la logique de Hoare partielle

La logique de Hoare partielle est également correcte. Nous allons donc montrer cette correction.

Théorème. *La logique de Hoare partielle est correcte. Pour toute formule partielle* $\{P\}\ S\ \{Q\}$, *on a :* $\vdash_p \{P\}\ S\ \{Q\}$ *implique* $\models_p \{P\}\ S\ \{Q\}$.

Démonstration. On raisonne par induction sur l'arbre de preuve de $\vdash_p \{P\}\ S\ \{Q\}$.

Cas $[ass_p]$ Soient $s, s' \in State$ tels que $\langle x := a, s\rangle \to s'$ et soit P un prédicat tel que $(P[x \mapsto \mathcal{A}[\![a]\!]])s = tt$. Montrons que $Ps' = tt$. Par $[ass_{NS}]$, on a $s' = s[x \mapsto \mathcal{A}[\![a]\!]_s]$. Comme $(P[x \mapsto \mathcal{A}[\![a]\!]])s = tt$, on a $P(s[x \mapsto \mathcal{A}[\![a]\!]_s]) = tt$. Donc $Ps' = tt$. D'où la propriété pour $[ass_p]$.

Cas $[skip_p]$ Immédiat par la règle $[skip_{NS}]$

Cas $[comp_p]$ Soient P, Q, R trois prédicats et S_1, S_2 deux instructions tels qu'on ait $\models_p \{P\}\ S_1\ \{Q\}$ et $\models_p \{Q\}\ S_2\ \{R\}$. Montrons que $\models_p \{P\}\ S_1; S_2\ \{R\}$. Soient $s, s'' \in State$ tels que $Ps = tt$ et $\langle S_1; S_2, s\rangle \to s''$. Par la règle $[comp_{NS}]$, on a l'existence de $s' \in State$ tel que $\langle S_1, s\rangle \to s'$ et $\langle S_2, s'\rangle \to s''$. Comme $\langle S_1, s\rangle \to s'$, $Ps = tt$ et $\models_p \{P\}\ S_1\ \{Q\}$, on a $Qs' = tt$. De même, on montre que $Rs'' = tt$. D'où la propriété pour $[comp_p]$.

Cas $[if_p]$ Soient P, Q deux prédicats et S_1, S_2 deux instructions tels qu'on ait $\models_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S_1\ \{Q\}$ et $\models_p \{\neg\mathcal{B}[\![b]\!] \wedge P\}\ S_2\ \{Q\}$. Montrons que $\models_p \{P\}\ \mathsf{if}\ b\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\ \{Q\}$. Soient $s, s' \in State$ tels que $Ps = tt$ et $\langle \mathsf{if}\ b\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2, s\rangle \to s'$. On distingue deux cas :

- Si $\mathcal{B}[\![b]\!] = tt$, alors $(\mathcal{B}[\![b]\!] \wedge P)s = tt$. Par la règle $[if_{NS}]$, on a $\langle S_1, s\rangle \to s'$. Comme, $\models_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S_1\ \{Q\}$, on a $Qs' = tt$.
- Si $\mathcal{B}[\![b]\!] = ff$, on raisonne de manière analogue.

Cas $[while_p]$ Soient P un prédicat et S une instruction tels qu'on ait $\models_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S\ \{P\}$. Montrons que $\models_p \{P\}\ \mathsf{while}\ b\ \mathsf{do}\ S\ \{\neg\mathcal{B}[\![b]\!] \wedge P\}$. Soient $s, s'' \in State$ tels que $Ps = tt$ et $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \to s''$. Montrons que $(\neg\mathcal{B}[\![b]\!] \wedge P)s'' = tt$. On raisonne par induction sur la dérivation de l'arbre.

- Si $\mathcal{B}[\![b]\!]_s = ff$, alors, par la règle $[while_{NS}]$, $s'' = s$ donc $(\neg\mathcal{B}[\![b]\!] \wedge P)s'' = tt$.
- Si $\mathcal{B}[\![b]\!]_s = tt$, alors il existe $s' \in State$ tel que $\langle S, s\rangle \to s'$ et $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s'\rangle \to s''$. Comme $(\mathcal{B}[\![b]\!] \wedge P)s = tt$, par application de l'hypothèse $\models_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S\ \{P\}$, $Ps' = tt$. On applique alors l'hypothèse d'induction sur la dérivation $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s'\rangle \to s''$ qui donne $(\neg\mathcal{B}[\![b]\!] \wedge P)s'' = tt$.

Cas $[cons_p]$ Soient P, Q, P', Q' quatre prédicats et S une instruction tels qu'on ait $\models_p \{P'\}\ S\ \{Q'\}$ et $P \Rightarrow P'$ ainsi que $Q' \Rightarrow Q$. Montrons que $\models_p \{P\}\ S\ \{Q\}$. Soient $s, s' \in State$ tels que $Ps = tt$ et $\langle S, s\rangle \to s'$. Comme $ps = tt$ et $P \Rightarrow P'$, on a $P's = tt$. De plus, par hypothèse $Q's' = tt$ et $Q' \Rightarrow Q$, donc $Qs = tt$. Donc $\models_p \{P\}\ S\ \{Q\}$.

D'où la correction. □

3 Sémantique opérationnelle à grands pas

La sémantique opérationnelle à grands pas [1, p.20] est une sémantique permettant de décrire comment on calcul à l'aide de grandes étapes (dans une boucle par exemple, on ne détaille pas toutes les exécutions mais seulement le résultat si elle termine).

Définition. On définit la sémantique à grands pas des instructions du langage IMP à l'aide de règle du type $\langle S, s\rangle \to s'$ avec $s, s' \in State$.

$[skip_{NS}]$ $\langle \mathsf{skip}, s\rangle \to s$ $\quad$ $[ass_{NS}]$ $\langle x := a, s\rangle \to s\,[x \mapsto \mathcal{A}[\![a]\!]_s]$

$[comp_{NS}]$ $\dfrac{\langle S_1,s\rangle\to s' \quad \langle S_2,s'\rangle\to s''}{\langle S_1;S_2,s\rangle\to s''}$

$[if^{tt}_{NS}]$ $\dfrac{\langle S_1,s\rangle\to s'}{\langle \text{ if } b \text{ then } S_1 \text{ else } S_2,s\rangle\to s'}$ si $\mathcal{B}[\![b]\!]_s = tt$ $\quad$ $[if^{ff}_{NS}]$ $\dfrac{\langle S_2,s\rangle\to s'}{\langle \text{ if } b \text{ then } S_1 \text{ else } S_2,s\rangle\to s'}$ si $\mathcal{B}[\![b]\!]_s = ff$

$[while^{tt}_{NS}]$ $\dfrac{\langle S,s\rangle\to s' \ \langle \text{while } b \text{ do } S,s'\rangle\to s''}{\langle \text{while } b \text{ do } S,s\rangle\to s''}$ si $\mathcal{B}[\![b]\!]_s = tt$ $\quad$ $[while^{ff}_{NS}]$ $\langle \text{while } b \text{ do } S, s\rangle \to s$ si $\mathcal{B}[\![b]\!]_s = ff$

Définition. On appelle arbre de dérivation la succession d'applications des règles et des axiomes.

Définition. Le langage IMP est déterministe sous la sémantique à grands pas, on définit donc une fonction sémantique : $\mathcal{S}_{NS} : Stm \to (State \hookrightarrow State)$ par

$$\mathcal{S}_{NS}[\![S]\!]s = \begin{cases} s' & \text{si } \langle S, s\rangle \to s' \\ \mathsf{undef} & \text{sinon} \end{cases}$$

Limite : La sémantique à grands pas ne permet pas toujours d'avoir la terminaison de notre instruction.

4 La complétude de la logique de Hoare partielle

Avant de définir la complétude partielle (on ne garantie les propriétés que si le programme termine (c'est une des hypothèses)), on n'a besoin d'un prédicat particulier : le weakest liberal precondition wlp.

Définition. Soit S une instruction du langage IMP et Q un prédicat :

$$\mathsf{wlp}(S, Q) = tt \text{ si et seulement si } \forall s' \in State, \text{ si } \underbrace{\langle S, s\rangle \to s'}_{\text{sémantique à grands pas}} \text{ alors } Qs' = tt$$

Lemme 1. *Pour toute instruction du langage IMP et prédicat Q, on a*

1. $\models_p \{wlp(S, Q)\}\ S\ \{Q\}$.
2. *Si* $\models_p \{P\}\ S\ \{Q\}$ *alors* $P \Rightarrow wlp(S, Q)$.

$wlp(S, Q)$ est alors la plus faible précondition pour S et Q.

Démonstration.
1. Soient $s, s' \in State$ tels que $\langle S, s\rangle \to s'$ et $\mathsf{wlp}(S, Q)s = tt$. Par définition de wlp, $Qs = tt$. Donc, $\models_p \{\mathsf{wlp}(S, Q)\}\ S\ \{Q\}$.
2. Supposons que $\{P\}\ S\ \{Q\}$ et $s \in State$ tel que $Ps = tt$. Si $\langle S, s\rangle \to s'$ alors $Qs' = tt$ (par $\models_p \{P\}\ S\ \{Q\}$). Donc, $\mathsf{wlp}(S, Q)s = tt$ (par définition).

□

Remarque. Il existe la strongest possible postcondition : $\mathsf{sp}(P, S)s' = tt$ si et seulement si $\exists s$ tel que $\langle S, s\rangle \to s'$ et $Ps = tt$. On peut également prouver la complétude partielle avec cette postcondition.

Théorème. *La logique de Hoare partielle telle que l'on a définie est complète. Pour tout formule partiellement correcte $\{P\}\ S\ \{Q\}$, on a : $\models_p \{P\}\ S\ \{Q\}$ implique $\vdash_p \{P\}\ S\ \{Q\}$.*

Démonstration. Il suffit de montrer que nous pouvons inférer pour toute instruction S et toute postcondition Q, $\vdash_p \{\mathsf{wlp}(S,Q)\}\ S\ \{Q\}$. En effet, supposons que $\vdash_p \{\mathsf{wlp}(S,Q)\}\ S\ \{Q\}$ et $\models_p \{P\}\ S\ \{Q\}$. Par le lemme 1, on a $P \Rightarrow \mathsf{wlp}(S,Q)$. En appliquant la règle $[\text{CONS}_p]$ avec $P \Rightarrow \mathsf{wlp}(S,Q)$, on a $\vdash_p \{P\}\ S\ \{Q\}$.

Montrons alors que pour toute instruction S et tout prédicat Q, $\vdash_p \{\mathsf{wlp}(S,Q)\}\ S\ \{Q\}$. On raisonne par induction structurelle sur les instructions S.

Cas du *skip* Montrons que $\mathsf{wlp}(S,Q) \Rightarrow Q$ (on peut également montrer que $\mathsf{wlp}(S,Q) = Q$ mais on a besoin que de l'implication dans ce développement). Soit $s \in State$ tel que $\mathsf{wlp}(S,Q)_s = tt$. Par définition, pour tout s' tel que $\langle S,s\rangle \to s'$ alors $Qs' = tt$. Par application de la règle $[skip_{NS}]$ de la sémantique à grands pas, on a $s = s'$. Donc $Qs = tt$. Comme $\mathsf{wlp}(S,Q) = Q$, en appliquant la règle $[skip_p]$, on a $\vdash_p \{\mathsf{wlp}(S,Q)\}\ skip\ \{Q\}$.

Cas $x := a$ Analogue car $wlp(S,Q) \Rightarrow Q[x := a]$.

Cas de la composition Soit Q un prédicat.

En appliquant les hypothèses d'induction à S_1 et S_2, on obtient : $\vdash_p \{\mathsf{wlp}(S_2,Q)\}\ S_2\ \{Q\}$ et $\vdash_p \{\mathsf{wlp}(S_1,\mathsf{wlp}(S_2,Q))\}\ S_1\ \{\mathsf{wlp}(S_2,Q)\}$ ($\mathsf{wlp}(S_2,Q)$ est légal car Q est un prédicat quelconque dans l'hypothèse d'induction). En appliquant la règle de la composition $[comp_p]$, on obtient $\vdash_p \{\mathsf{wlp}(S_1,\mathsf{wlp}(S_2,Q))\}\ S_1;S_2\ \{Q\}$.

Montrons que $\mathsf{wlp}(S_1;S_2,Q) \Rightarrow \mathsf{wlp}(S_1,\mathsf{wlp}(S_2,Q))$ (on veut appliquer $[comp_p]$). On suppose qu'il existe $s \in State$ tel que $\mathsf{wlp}(S_1;S_2,Q)s = tt$. Montrons que $\mathsf{wlp}(S_1,\mathsf{wlp}(S_2,Q))s = tt$. Soit $s' \in State$ tel que $\langle S_1,s\rangle \to s'$ et montrons que $\mathsf{wlp}(S_2,Q)s' = tt$. Soit $s'' \in State$ tel que $\langle S_2,s'\rangle \to s''$ et montrons que $Qs'' = tt$. Par la règle $[comp_{NS}]$ de la sémantique à grands pas, on a $\langle S_1;S_2,s\rangle \to s''$. Comme $\mathsf{wlp}(S_1;S_2,Q)s = tt$ (par hypothèse), $Qs'' = tt$ (on peut peut-être condenser ce passage).

On applique alors $[cons_p]$ avec $\mathsf{wlp}(S_1;S_2,Q) \Rightarrow \mathsf{wlp}(S_1,\mathsf{wlp}(S_2,Q))$, d'où $\vdash_p \{\mathsf{wlp}(S_1;S_2,Q)\}\ S_1;S_2\ \{Q\}$.

Cas de la conditionnelle Soit Q un prédicat.

On pose $P = (\mathcal{B}[\![b]\!] \wedge \mathsf{wlp}(S_1,Q)) \vee (\neg\mathcal{B}[\![b]\!] \wedge \mathsf{wlp}(S_2,Q))$ (si j'ai b alors je dois pouvoir appliquer S_1, sinon je dois pouvoir appliquer S_2). On a alors $\mathcal{B}[\![b]\!] \wedge P = (\mathcal{B}[\![b]\!] \wedge \mathsf{wlp}(S_1,Q)) \vee (\mathcal{B}[\![b]\!] \wedge \neg\mathcal{B}[\![b]\!] \wedge \mathsf{wlp}(S_2,Q)) = \mathcal{B}[\![b]\!] \wedge \mathsf{wlp}(S_1,Q) \Rightarrow \mathsf{wlp}(S_1,Q)$. De même, on montre que $\neg\mathcal{B}[\![b]\!] \wedge P \Rightarrow \mathsf{wlp}(S_2,Q)$.

Montrons que $\vdash_p \{P\}$ if b then S_1 else S_2 $\{Q\}$. En appliquant les hypothèses d'induction à S_1 et S_2, on obtient : $\vdash_p \{\mathsf{wlp}(S_1,Q)\}\ S_1\ \{Q\}$ et $\vdash_p \{\mathsf{wlp}(S_2,Q)\}\ S_2\ \{Q\}$. En appliquant la règle $[cons_p]$ aux deux preuves avec $\mathcal{B}[\![b]\!] \wedge P \Rightarrow \mathsf{wlp}(S_1,Q)$ et $\neg\mathcal{B}[\![b]\!] \wedge P \Rightarrow \mathsf{wlp}(S_2,Q)$, on obtient $\vdash_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S_1\ \{Q\}$ et $\vdash_p \{\neg\mathcal{B}[\![b]\!] \wedge P\}\ S_2\ \{Q\}$. On applique la règle $[if_p]$, donc $\vdash_p \{P\}$ if b then S_1 else S_2 $\{Q\}$.

Montrons maintenant que $\mathsf{wlp}(\text{if } b \text{ then } S_1 \text{ else } S_2, Q) \Rightarrow P$. Supposons qu'il existe $s \in State$ tel que $\mathsf{wlp}(\text{if } b \text{ then } S_1 \text{ else } S_2, Q)s = tt$ et montrons que $Ps = tt$. On distingue deux cas :

— Cas $\mathcal{B}[\![b]\!]_s = tt$. Soit $s' \in State$ tel que $\langle \text{if } b \text{ then } S_1 \text{ else } S_2\rangle \to s'$. Montrons que $Qs' = tt$. Comme $\mathcal{B}[\![b]\!]_s = tt$, en appliquant la règle $[if^{tt}_{NS}]$, $\langle \text{if } b \text{ then } S_1 \text{ else } S_2\rangle \to s'$ si $\langle S_1,s\rangle \to s'$. Or, par hypothèse d'induction ($\vdash_p \{\mathsf{wlp}(S_1,Q)\}\ S_1\ \{Q\}$) $Ps' = tt \Rightarrow \mathsf{wlp}(S_1,Q)s = tt \Rightarrow Qs' = tt$.

— Cas $\mathcal{B}[\![b]\!]_s = ff$: analogue

En appliquant la règle $[cons_p]$ avec $\mathsf{wlp}(\text{if } b \text{ then } S_1 \text{ else } S_2, Q) \Rightarrow P$ permet de conclure.

Cas de la boucle while On pose $P = \mathsf{wlp}(\text{while } b \text{ do } S, Q)$.

Montrons que $(\neg\mathcal{B}[\![b]\!] \wedge P) \Rightarrow Q$. Soit $s \in State$ tel que $(\neg\mathcal{B}[\![b]\!] \wedge P)s = tt$. Dans ce cas, (application de la conjonction) $(\mathcal{B}[\![b]\!])s = ff$ et $Ps = tt$. Comme $(\mathcal{B}[\![b]\!])s = ff$, en appliquant la règle $[while^{ff}_{NS}]$, on a $\langle \text{while } b \text{ do } S, s\rangle \to s$. Donc, $Qs = tt$ (par $Ps = tt$, sa définition et la définition de wlp).

Montrons que $(\mathcal{B}[\![b]\!] \wedge P) \Rightarrow \mathsf{wlp}(S,P)$. Soit $s \in State$ tel que $(\mathcal{B}[\![b]\!] \wedge P)s = tt$, donc (application de la conjonction) $(\mathcal{B}[\![b]\!])s = tt$ et $Ps = tt$. Montrons que $\mathsf{wlp}(S,P)s = tt$. Soit $s' \in State$ tel que $\langle S,s\rangle \to s'$ et montrons que $Ps' = tt$. On distingue deux cas.

— Supposons qu'il existe $s'' \in State$ tel que $\langle \text{while } b \text{ do } S, s'\rangle \to s''$. En appliquant la règle $[while^{tt}_{NS}]$, on a $\langle \text{while } b \text{ do } S, s\rangle \to s''$. Comme $Ps = tt$ (hypothèse), on a alors $Qs'' = tt$ par le lemme 1.

— Supposons maintenant que pour tout $s'' \in State$ on n'ait pas $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s' \rangle \rightarrow s''$. On en déduit que pour tout $s'' \in State$, $Ps' = ff$.

L'hypothèse d'induction sur S donne $\vdash_p \{\mathsf{wlp}(S,P)\}\ S\ \{P\}$. En appliquant la règle $[cons_p]$ avec $(\mathcal{B}[\![b]\!] \wedge P) \Rightarrow \mathsf{wlp}(S,P)$, on obtient $\vdash_p \{\mathcal{B}[\![b]\!] \wedge P\}\ S\ \{P\}$. En appliquant la règle $[while_p]$, on a $\vdash_p \{P\}\ \mathsf{while}\ b\ \mathsf{do}\ S\ \{\neg\mathcal{B}[\![b]\!] \wedge P\}$. En appliquant une seconde fois la règle $[cons_p]$, on a $\vdash_p \{P\}\ \mathsf{while}\ b\ \mathsf{do}\ S\ \{Q\}$.

D'où la complétude. □

Références

[1] H. R. Nielson ; F. Nielson. *Semantics with application*.