LOG : Preuve de programme

Logique de Hoare

Définition : Triplet de Hoare
On a $\{P\}\ \ C\ \ \{Q\}$ où :

- C est le code qui va tourner
- P est la **précondition** (Assertion sur l'état précédent)
- Q est la **postcondition** (Assertion sur l'état suivant)
- On dit que "C satisfait la spécification (P, Q)" (ou si P est vraie, alors quand je fais tourner C, Q devient vraie.)

Définition : Assertion
Une assertion est une formule de logique du premier ordre qui décrit les relations entre les variables d'un algorithme.

Définition : Inférence et règle d'inférence

- **Inférence :** Déduction de nouveaux faits en combinant (correctement) des faits déjà existants
- **Règle d'inférence :** Mécanisme décrivant comment les faits peuvent être combinés.

Représentation classique d'une règle :

$$\frac{p_1, p_2, ..., p_n}{q}$$

Si $p_1, p_2, ..., p_n$ sont vraies, alors q l'est aussi.

Premiers axiomes et règles :

- Axiome de l'ensemble vide :

$$\overline{\{P\} skip \{P\}}$$

- Axiome d'affectation :

$$\overline{\{P[x/E]\} x := E \{P\}}$$

  $P[x/E]$ désigne l'expression P dans laquelle les occurrences de la variable x ont été remplacées par l'expression E.

- Règle de la conséquence (ou affaiblissement) :

$$\frac{P \to\ P'\ ,\ \{P'\}\ S\ \{Q'\}\ ,\ Q \to\ Q'}{\{P\ \}\ S\ \{Q'\}}$$

  On dit que P est plus faible que P' et que Q' est plus forte que Q'

- Règle de composition :

$$\frac{\{P\}\ C_1\ \{Q\}\ ,\ \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1 ; C_2\ \{R\}}$$

- Première règle conditionnelle :

$$\frac{\{B \wedge P\}\ S\ \{Q\}\ , B \wedge \neg P \Rightarrow Q}{\{B\}\ \textbf{si}\ P\ \textbf{alors}\ S\ \textbf{fin si}\ \{Q\}}$$

- Deuxième règle conditionnelle :

$$\frac{\{B \wedge P\}\ S\ \{Q\}\ ,\ \{B \wedge \neg P\}\ T\ \{Q\}}{\{P\}\ \textbf{si}\ B\ \textbf{alors}\ S\ \textbf{sinon}\ T\ \textbf{fin si}\ \{Q\}}$$

- Règle de l'itération :

$$\frac{\{I \wedge B\}\ S\ \{I\}}{\{I\}\ \textbf{tant que}\ B\ \textbf{faire}\ S\ \textbf{fait}\ \{\neg B \wedge I\}}$$

  On dit que $\{I\}$ est **l'invariant de boucle**

Weakest preconditions : On veut déterminer les WP pour obtenir la postcondition $\{Q\}$ à partir de $C - WP(C,Q)$

- $\mathbf{WP}(nop, Q) \equiv Q$
- $\mathbf{WP}(x := E, Q) \equiv Q[x := E]$
- $\mathbf{WP}(C; D, Q) \equiv \mathbf{WP}(C, \mathbf{WP}(D, Q))$
- $\mathbf{WP}(\texttt{ if } Cond \texttt{ then } C \texttt{ else } D, Q) \equiv (Cond = \texttt{true} \Rightarrow \mathbf{WP}(C,Q)) \wedge (Cond = \texttt{False} \Rightarrow \mathbf{WP}(D,Q)$
- $\mathbf{WP}(\texttt{ while } E \texttt{ do } C \texttt{ done}, Q) \equiv I$ (Avec I l'invariant et V le variant)
  Plus les obligations de preuve suivantes :
  - $(E = \texttt{true} \wedge I \wedge V = z) \Rightarrow \mathbf{WP}(C, I \wedge V < z)$ (Le variant est décrémenté)
  - $I \Rightarrow V \geq 0$ (Le variant reste valide)
  - $(E = \texttt{false} \wedge I) \Rightarrow Q)$ (Une fois terminé, on a Q)