Cours 1

Boulanger Jean-Louis

RATP

DESS QUASI : 2001-2002

Méthodes semi-formelles

- SADT, SA-RT, SSADM
- Merise, Axial
- OOA, OMT
- UML

Trois paradigmes

– Aspect Donnée
– Aspect Opération
– Aspect Processus

Méthodes formelles

- Méthode orientée modèle (on s'intéresse aux données)
  - Séquentielle : B , Z et VDM,
  - Processus : CSP
- Méthode axiomatique (on s'intéresse aux propriétés)
  - Séquentielle : ACT ONE, OBJ,
  - Alèbre des Processus : LOTOS
- Méthode hybrides ex : RAISE (VDM, CSP)

Principe de base des méthodes formelles

Une spécification est dite formelle si :

- elle est écrite en suivant une syntaxe (précise) bien définie,
- la syntaxe est accompagnée d'une sémantique rigoureuse qui définit des modèles mathématiques représentant les réalisations acceptables,
- elle est accompagné d'un système de raisonnement formel.

Développements classiques

1. Méthodes classiques insuffisantes :

   p est de l'ordre de 10^{-3}

2. Tests intensifs :
   - Ne prouvent rien
   - Couverture insuffisante
     - Conditions aux limites
     - Défaillances ultra-improbables

$$\Rightarrow$$

Une méthode « mathématique »

Pourquoi utiliser une méthode formelle ?

- Enjeu
  - La sûreté de fonctionnement des logiciels critiques
- Apport des méthodes formelles
  - Notations mathématiques pour spécifier
  - Preuve formelle pour vérifier la conception

Une méthode formelle

- **Objectif**

  $\boxed{\textbf{démontrer}}$ qu'un $\boxed{\textbf{logiciel}}$ réalise sa $\boxed{\textbf{spécification}}$.

- **Moyen**

  On ne sait pas faire :

  $$\text{Logiciel} \Rightarrow \text{Spécification}$$

  Par contre, on sait à peu près faire :

  $$\text{Prédicat} \Rightarrow \text{Prédicat}$$

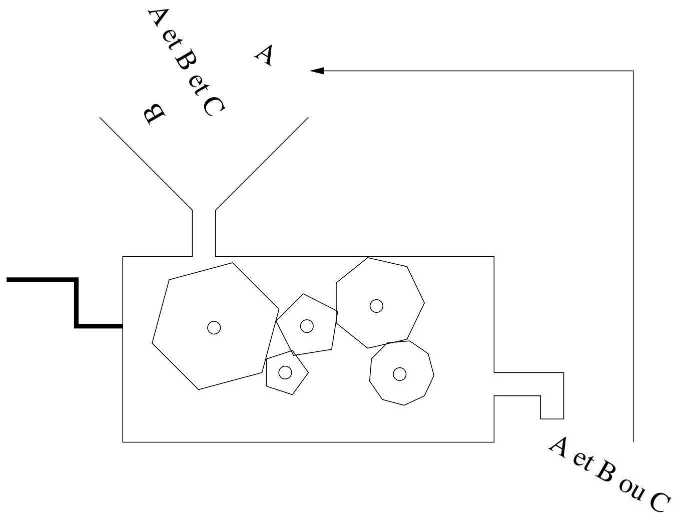
- **Problème**

  $$\text{Logiciel} \leadsto \text{Prédicat}$$

  ou plutôt

  $$\text{Prédicat} \leadsto \text{Logiciel}$$

Un système formel



Notion de preuve

$$H_1 \wedge \ldots \wedge H_n \Rightarrow B$$

Un système formel

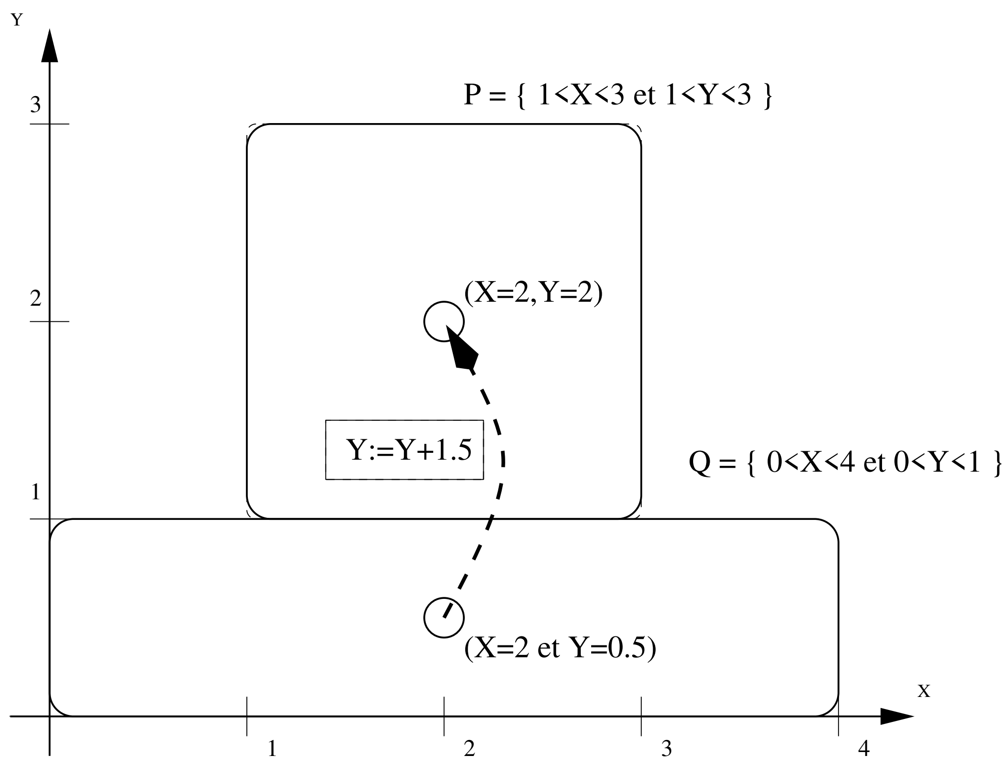
Une preuve s’effectue toujours dans un **système formel**

1. Des « formules » initiales (les **axiomes**)
2. Des transformations (les **règles d’inférences**)

La notion de sémantique

- La notion de **preuve** n'est pas *intrinsèquement* liée à la notion de **vérité**.
- C'est la **sémantique** qui permet de donner un sens « intuitif » aux formules manipulées.
- Ainsi, la sémantique des formules logiques classiques est obtenue par affectation des valeurs `vrai` et `faux`.

Modèle d'exécution



Notation de Hoare

$$\{P\}\ C\ \{Q\}$$

- C dénote le programme
- P et Q sont des conditions exprimées sur les variables du programme C

$\{P\}C\{Q\} \equiv$

1. Pour toute exécution de C,
2. à partir d'un état vérifiant P,
3. alors Q est vérifiée par l'état dans lequel **se termine** C.

Terminologie

$$\{P\}\ C\ \{Q\}$$

- P s'appelle la **pré-condition**
- Q s'appelle la **post-condition**

- Établir $\{P\}C\{Q\}$, c'est établir la **correction partielle** de C.
- Correction totale $=$ Terminaison + Correction partielle
- La **correction totale** se note $[P]C[Q]$

Exemple

$\{T\}$

```
BEGIN
    R:=X;Q:=0;
    WHILE Y <= R DO
        BEGIN
            R:=R-Y;Q:=Q+1;
        END
    END
END
```

$\{R \leq Y \wedge X = R + (Y \times Q)\}$

L'affectation

$$\{P[E/V]\}\boxed{\texttt{V :=E}}\{P\}$$

– $P[E/V]$ dénote la **substitution** de toutes les occurences de V dans P par le terme E

Vrai ou Faux ?

$\{x = 1\}$

```
x := x + 1
```

$\{x = 2\}$

Application

$$\{P[E/V]\}\boxed{\texttt{V :=E}}\{P\}$$

Soit le programme suivant réalisant la permutation de X et Y

$$\{X = x \wedge Y = y\} \quad \boxed{\texttt{R :=X; X :=Y; Y :=R}} \quad \{Y = x \wedge X = y\}$$

On démontre

- En progressant « à reculons » (*backward*)
- Par applications successives de la règle d'affectation
  1. $\{X = x \wedge Y = y\}$ $\boxed{\texttt{R :=X}}\,;\,\boxed{\texttt{X :=Y}}$ $\{R = x \wedge X = y\}$
  2. $\{X = x \wedge Y = y\}$ $\boxed{\texttt{R :=X}}$ $\{R = x \wedge Y = y\}$
  3. $\{X = x \wedge Y = y\}$ CQFD $\{X = x \wedge Y = y\}$

Obligations de preuve (*Verification conditions*)

Pour vérifier la spécification suivante :

$$\{P\}\ \boxed{\texttt{V :=E}}\ \{Q\}$$

il faut *décharger* l'*obligation de preuve* suivante :

$$P \Rightarrow Q[E/V]$$

- Spécification :

$$\{X = 0\}\ \boxed{\texttt{X :=X+1}}\ \{X = 1\}$$

- Preuve :

$$X = 0 \Rightarrow (X + 1) = 1$$

(Laissé en exercice)

Obligations de preuve

Pour vérifier la spécification suivante :

$$\{P\}\ \boxed{\texttt{WHILE}\ S\ \texttt{DO}\ \{I\}\ \texttt{C}}\ \{Q\}$$

il faut décharger les obligations de preuve suivantes :

1. $P \Rightarrow I$
2. $P \wedge \neg S \Rightarrow Q$
3. Les OPs générées par

$$\{I \wedge S\}C\{Q\}$$

Vrai ou Faux ?

$\{x = 1\}$

```
WHILE TRUE DO
     SKIP
END
```

$\{y = 2\}$

Spécifier

Spécifier c'est décrire **ce qui doit être calculé** et non pas **comment le calculer**.