

Programmation procédurale

Schémas de programmes

D.E ZEGOUR

École Supérieure d'Informatique

ESI

Programmation procédurale / Schémas de programmes

Sommaire

A. Introduction

B. Les différents schémas

B-schémas, D-schémas R-schémas

BJn-schémas REn-schémas

Autres variantes

C. Comparaison des structures

Relations entre schémas

Relations entre structures

Quelques résultats

Programmation procédurale / Schémas de programmes

Introduction

Calculateurs câblés :

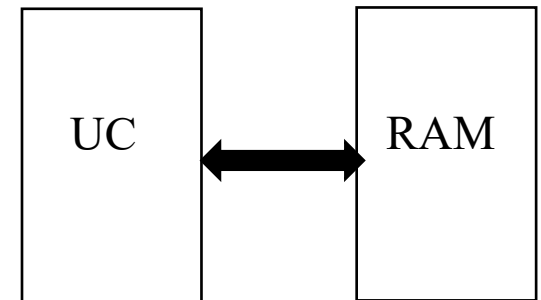
A l'origine de l'informatique, toutes les opérations étaient câblées.

Machine de Von Neumann

- Le programme est stocké en mémoire centrale (RAM).
- Existence d'un canal de communication entre la RAM et l'unité de traitement (UC)

Caractéristiques :

- Séquentiabilité et affectation.
- n'est pas basée sur un système formel.



Donc deux opérations de base :

- traitement (UC)
- mouvement d'informations

Programmation procédurale / Schémas de programmes

Schémas avec Branchement (**B-schémas**)

Syntaxe sous forme étendue de Backus-Naur

$\langle \text{b-algo} \rangle \Rightarrow \langle \text{Inst} \rangle \{ ; \langle \text{Inst} \rangle \}^*$
 $\langle \text{Inst} \rangle \Rightarrow \langle \text{InstSimple} \rangle \mid$
Etiqu: InstSimple
 $\langle \text{InstSimple} \rangle \Rightarrow \langle \text{saut} \rangle \mid$
 $\langle \text{SautCond} \rangle \mid$
stop \mid a \mid b \mid c \mid d
 $\langle \text{Saut} \rangle \Rightarrow$ Allera Etiqu
 $\langle \text{SautCond} \rangle \Rightarrow$ Si $\langle \text{exp} \rangle$ Allera Etiqu
 $\langle \text{Exp} \rangle \Rightarrow T1 \mid T2 \mid T3 \mid \dots$

Les seules structures de contrôle sont les
branchements inconditionnel et conditionnel.

Le langage des B-schémas est
solution de l'équation à point fixe

$$\begin{aligned} B &= A \cup \\ &\text{Et} : B \cup \\ &B ; B \cup \\ &\text{Si } P \text{ allera Et } \cup \\ &\text{Allera Et} \end{aligned}$$

$A = \{a, b, c, \dots\} + \text{Stop}$
 $P = \{T1, T2, \dots\}$

Programmation procédurale / Schémas de programmes

Schémas avec Branchement (B-schémas)

Exemple

Et1 a
 b
 Si T1 Allera Et3

Et2 c
 d
 Si T2 Allera Et2

 a
 Alera Et1

Et3 b
 Stop

Le schéma de programme devient un **programme** lorsqu'on donne un sens aux symboles de fonctions et de prédicats.

C'est ce qu'on appelle une interprétation.

Programmation procédurale / Schémas de programmes

Schémas itératifs (**D-schémas**)

(D pour Dijkstra)

```
<d-algo> =>    <Inst> {; <Inst>}*
<Inst>    =>    <Tantque>| <Sisidon>|a|b|c|d|....
<Tantque>=>    TANTQUE <exp> :
                <Inst>{;<Inst>}*
                FINTANTQUE
<Sisidon>=>    SI <exp>:
                <Inst> {; <Inst>}*
                [SINON
                <Inst>{; <Inst>}* ]
                FSI
<Exp>     =>    T1|T2|T3|.....
```

Le langage des D-schémas est solution de l'équation à point fixe

$$\begin{aligned} D &= A \cup \\ &D ; D \cup \\ \text{Tantque } P &: D \text{ Fintantque } U \\ \text{Si } P &: D \text{ Sinon } D \text{ Finsi} \end{aligned}$$

$$\begin{aligned} A &= \{a, b, c, \dots\} \\ P &= \{T1, T2, \dots\} \end{aligned}$$

Programmation procédurale / Schémas de programmes

BJ_n-schémas

B pour Bohm et J pour Jaccopini

BJ =A U

BJ ; BJ U

Si P alors BJ sinon BJ fsi U

Faire

P --> BJ ;

P --> BJ ;

... ;

P --> BJ

Fait

A={a, b, c, ..}

P ={T1, T2, ...}

L'itération est arrêtée dès qu'une condition P devient fausse.

n : nombre maximal de (P--> BJ) entre Faire et Fait

Programmation procédurale / Schémas de programmes

REn-schémas

Schéma Répétitif avec Exit.

On introduit un nouveau type de d'instruction :

EXIT(i), $1 \leq i \leq n$

sortir directement de i boucles englobantes.

Le langage des RE-schémas est solution de l'équation à point fixe :

$RE = A \cup$
$RE ; RE \cup$
Si P alors RE Sinon RE fsi \cup
Faire RE fait

$A = \{a, b, c, \dots\} + \{\text{Exit}(i)\}$

$P = \{T1, T2, \dots\}$

Programmation procédurale / Schémas de programmes

Autres variantes

RECn-shémas

C'est RE auquel on ajoute l'instruction **Entrée(i)** qui réalise un saut au début de la i-ème itération.

Dans les REn et RECn on ne retrouve pas la structure TQ

On appelle **DREn** (respectivement **DRECn**) schémas l'ensemble des schémas REn et RECn étendus par la construction *Tantque p faire a fait* des D-schémas.

Les instructions Entrée et Exit ne portent pas sur l'instruction Tantque.

Programmation procédurale / Schémas de programmes

Comparaison des structures (Relations entre schémas)

R1 : Pour toute interprétation I
des schémas π_1 et π_2
on a : $I(\pi_1) = I(\pi_2)$.

R2 : Les ensembles d'actions et
de prédicats ayant une
occurrence dans π_1 ou dans π_2
sont les mêmes.

Remarques :

1. R1 et R2 sont des relations d'équivalence.
2. Si deux schémas sont reliés par la relation R2, le passage de l'un à l'autre se fait sans introduction de nouvelles actions ou tests.

Programmation procédurale / Schémas de programmes

Comparaison des structures (Relations entre structures)

Soient $S1$ et $S2$ deux structures et R une relation

$S1$ est dit réductible en $S2$ pour la relation R si quelque soit $\pi1$ de $S1$, il existe $\pi2$ de $S2$ tel que $\pi1 R \pi2$.

notation : $S1 \leq_R S2$

$S1$ est dit équivalent à $S2$ pour R si :

$S1 \leq_R S2$ et $S2 \leq_R S1$

notation : $S1 =_R S2$

$S1$ est dit strictement réductible en $S2$ pour R si :

$S1 \leq_R S2$ et non ($S2 \leq_R S1$)

notation : $S1 <_R S2$

Programmation procédurale /

Schémas de programmes

Comparaison des structures (Relations courantes entre structures)

Conversion fonctionnelle (FN)

Elle correspond à R1 : $S1 \leq_{FN} S2$.

Sens : Pour toute interprétation I , et tout schéma $\pi1$ de $S1$, il existe un schéma $\pi2$ de $S2$ tel que $I(\pi1) = I(\pi2)$.

Conversion sémantique (SE)

Elle correspond à R1 et R2.

$S1 \leq_{SE} S2$.

Sens : Pour toute interprétation I , et tout schéma $\pi1$ de $S1$, il existe un schéma $\pi2$ de $S2$ construit sans introduction de nouvelles actions et tel que $I(\pi1) = I(\pi2)$.

R1 : Pour toute interprétation I des schémas $\pi1$ et $\pi2$
on a : $I(\pi1) = I(\pi2)$.

R2 : Les ensembles d'actions et de prédicats ayant une occurrence dans $\pi1$ ou dans $\pi2$ sont les mêmes.

Quelques résultats

$$D <_{SE} B$$

$$D =_{FN} B$$

(Théorème de Bohm et Jaccopini)

Programmation procédurale / Schémas de programmes

Quelques résultats

Les comparaisons entre structures sont les résultats des travaux de RAO-KOSARAJU (1974). $<, \leq, =$ désigne $<_{SE}, \leq_{SE}, =_{SE}$. (V désigne $<$ et VI désigne \leq)

RE	=	REC	=	DRE	=	DREC	=	B
V								
RE _n	=	REC _n	\leq	DRE _n	=	DREC _n		
V								
...								
RE ₂	=	REC ₂	\leq	DRE ₂	=	DREC ₂		
V								
RE ₁	=	REC ₁	\leq	DRE ₁	=	DREC ₁		
V								
BJ								
...								
V								
BJ ₂								
V								
BJ ₁	=	D						

Programmation procédurale / Schémas de programmes

Conclusion

Soit S une structure quelconque parmi celles que nous venons de définir. D'après le tableau (slide précédent) nous avons

$$D \leq_{SE} S \leq_{SE} B$$

D'où a fortiori

$$D \leq_{FN} S \leq_{FN} B$$

Comme $B =_{FN} D$ (Théorème de Bohm et Jaccopini), on en déduit $S =_{FN} B$

Théorème : Toutes les structures présentées sont fonctionnellement équivalentes.