

Exercice

```
local Max C in  
  proc {Max X Y ?Z}  
    if X>=Y then Z=X else Z=Y end  
  end  
  {Max 3 5 C}  
end
```

- Version complète dans le langage noyau
- Algorigramme
- Détail de l'exécution

Etude des programmes

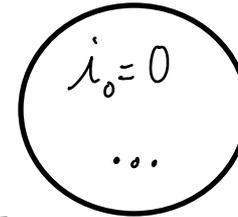
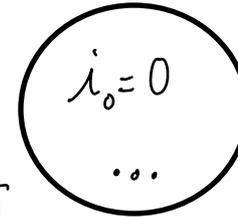
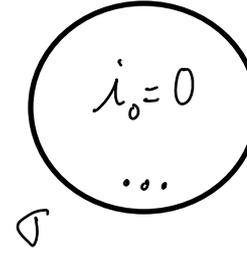
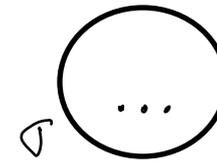
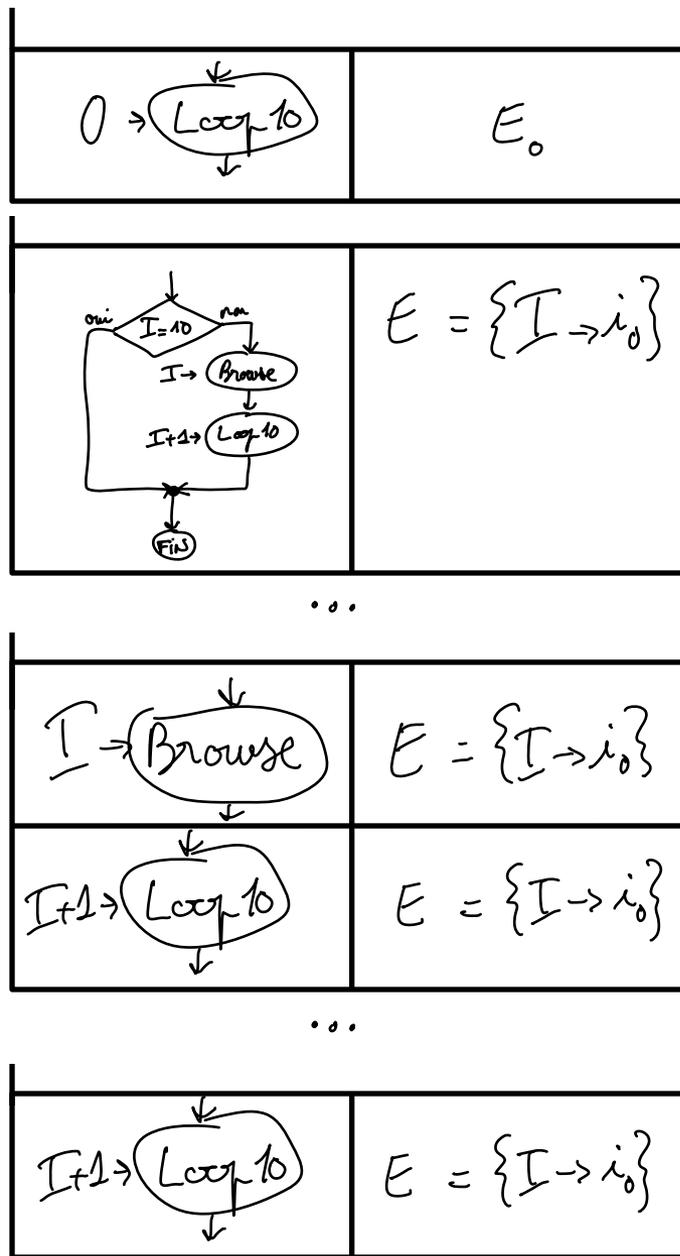
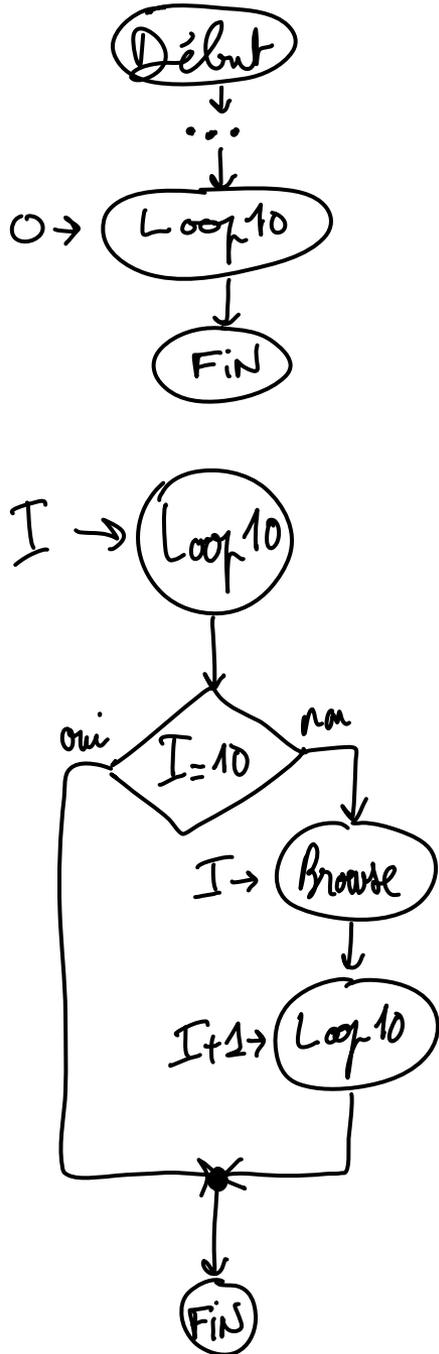
- Sémantique opérationnelle
→ raisonner sur les programmes
- Raisonner pour
 - Vérifier que le programme implémente bien la spécification
 - Déterminer l'efficacité calculatoire, les performances du programme
 - En termes d'espace utilisé
 - En termes de temps de calcul
- Exemple de raisonnement
Montrer qu'une procédure récursive terminale s'exécute avec une pile de taille constante.

Optimisation terminale

- Une procédure réursive terminale appel récursif en dernière instruction
- Exécution avec une pile de taille constante
 - « optimisation du dernier appel »
 - « optimisation terminale »
- Exemple

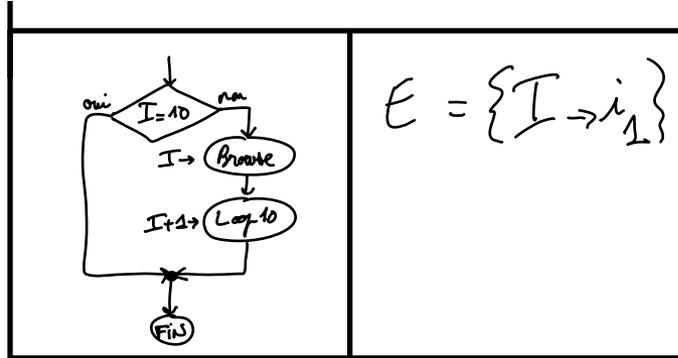
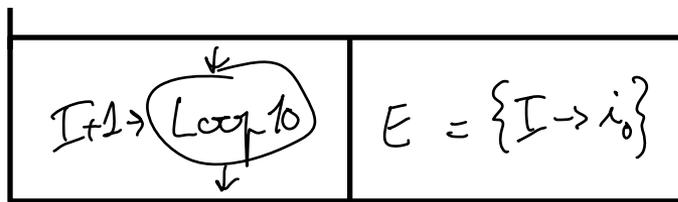
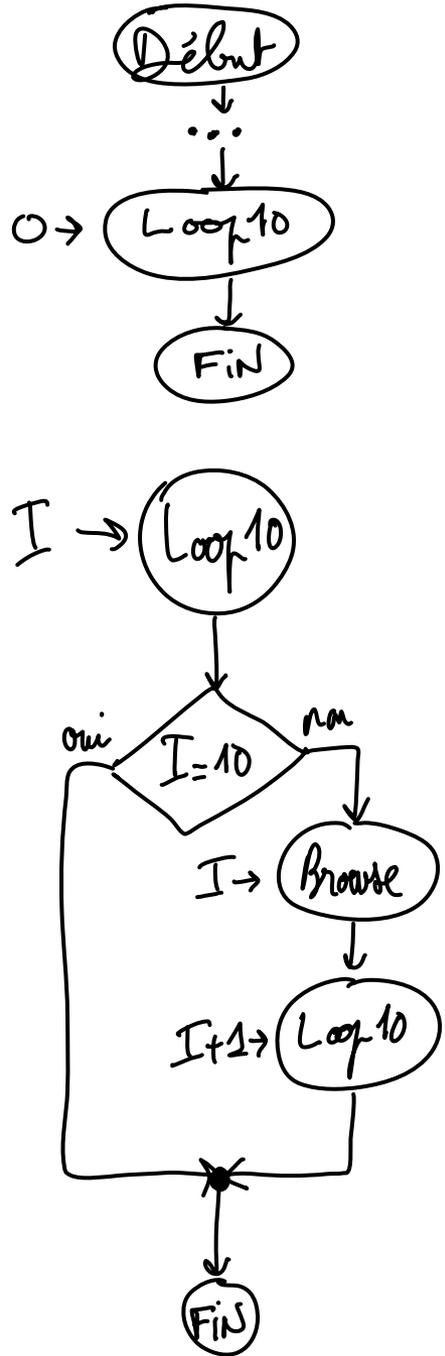
```
proc {Loop10 I}  
  if I==10 then skip  
  else {Browse I} {Loop10 I+1} end  
end
```

Exemple

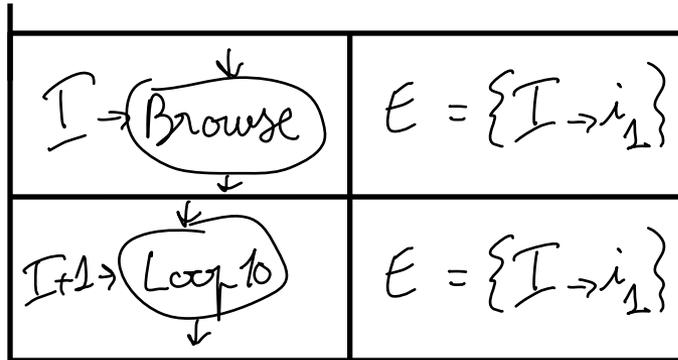


Appel récursif no 1

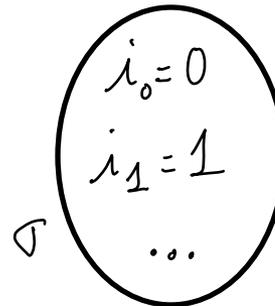
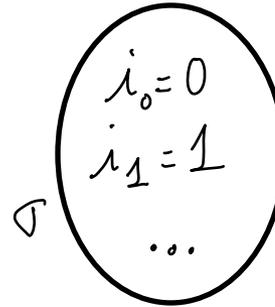
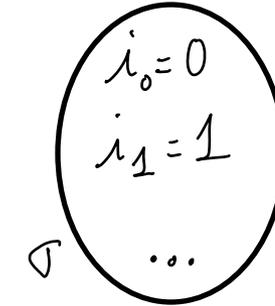
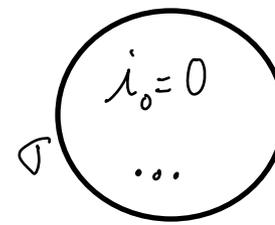
Exemple



...



...

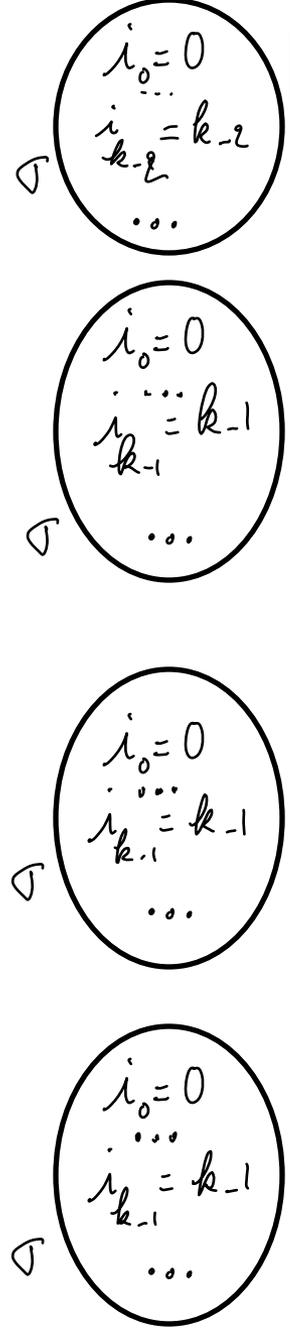
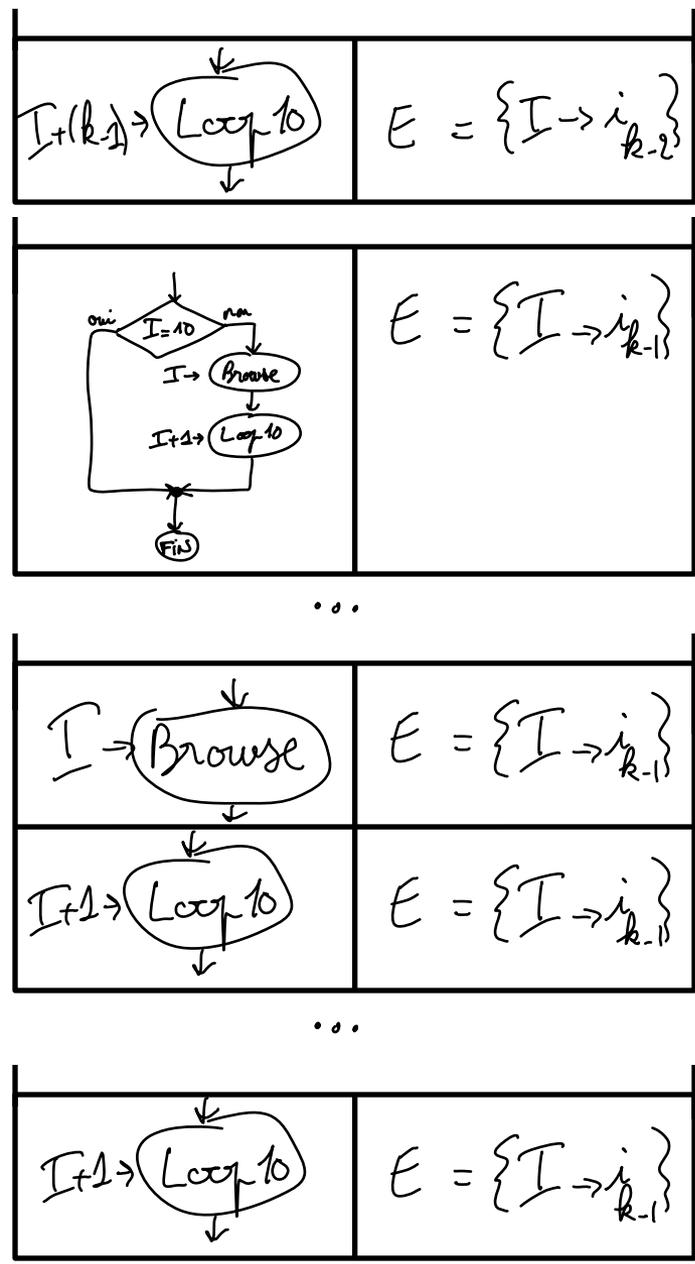
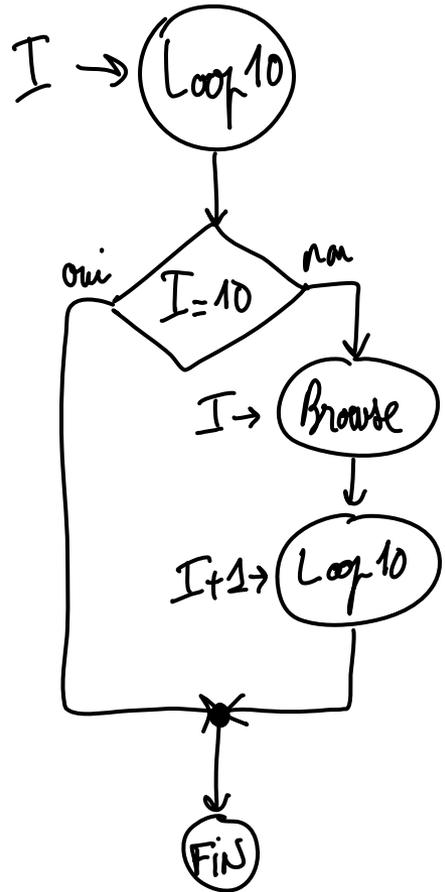
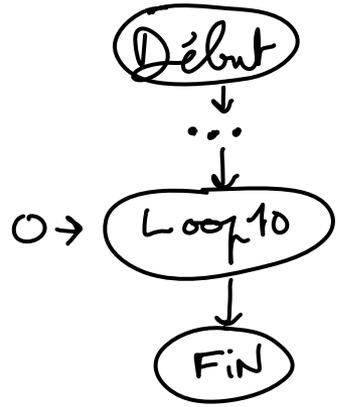


Appel récursif no 2

Example

• • •

Exemple



Appel récursif
no k

Exemple

- Pile sémantique de taille bornée
- Mémoire proportionnelle au nombre d'appel
 - À l'itération k , les variables i_0 à i_{k-2} ne servent pas
 - ➔ on ne garde que i_{k-1}
 - ➔ taille mémoire constante

*Appel récursif
no k*

