

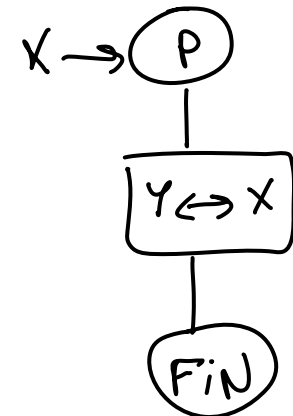
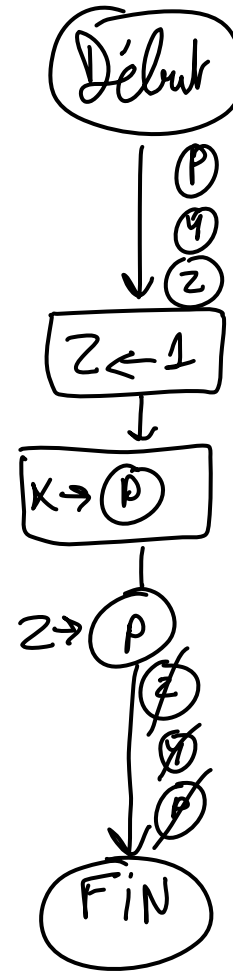
[Exemple avec un appel de procédure]

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

L'instruction en langage noyau

```
local P in
  local Y in
    local Z in
      Z=1
      proc {P X}
        Y=X
      end
    end
  end
end
```



Avec du “sucre syntaxique”

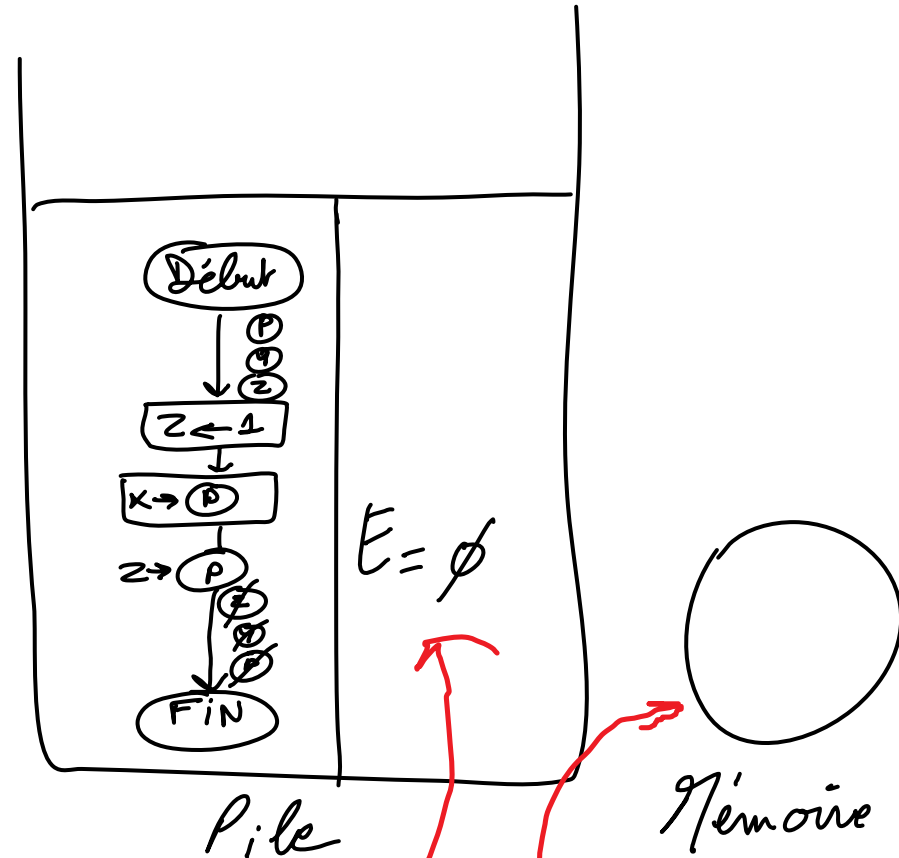
```
local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end
```

Nous allons tricher un peu: nous allons faire toutes les déclarations en même temps

Ce genre de petite “tricherie” est très utile et on peut la justifier en ajoutant des règles de sémantique

Début de l'exécution

```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```

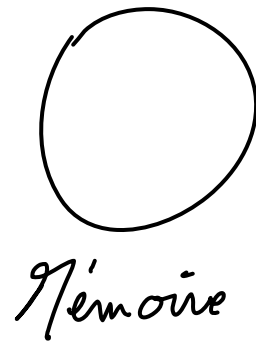
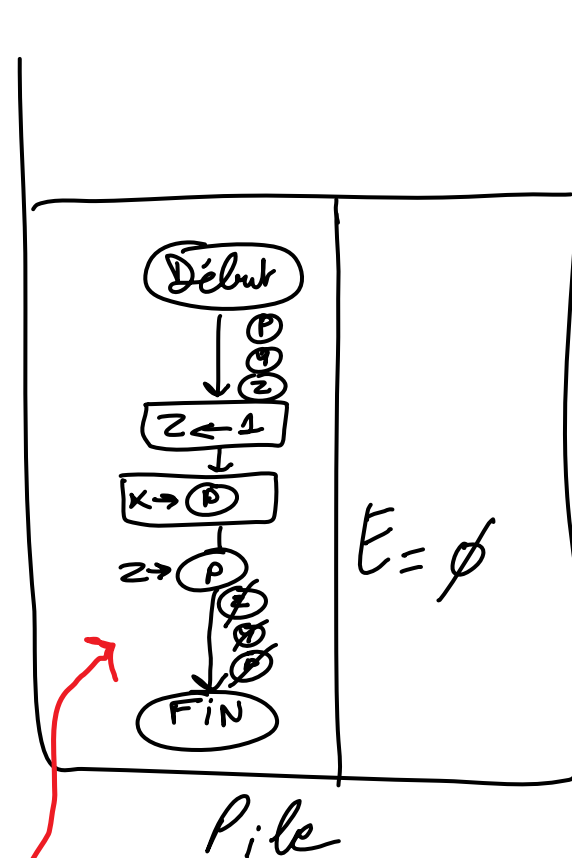


Etat d'exécution initial

Environnement vide et mémoire vide

Début de l'exécution

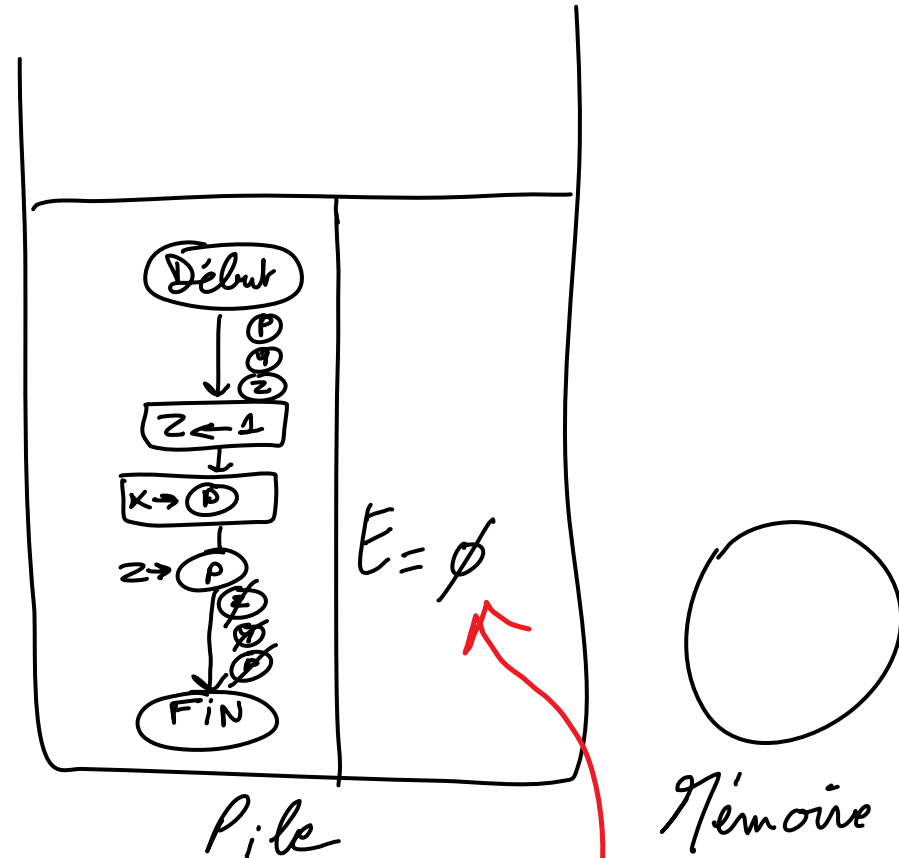
```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```



Première instruction

Début de l'exécution

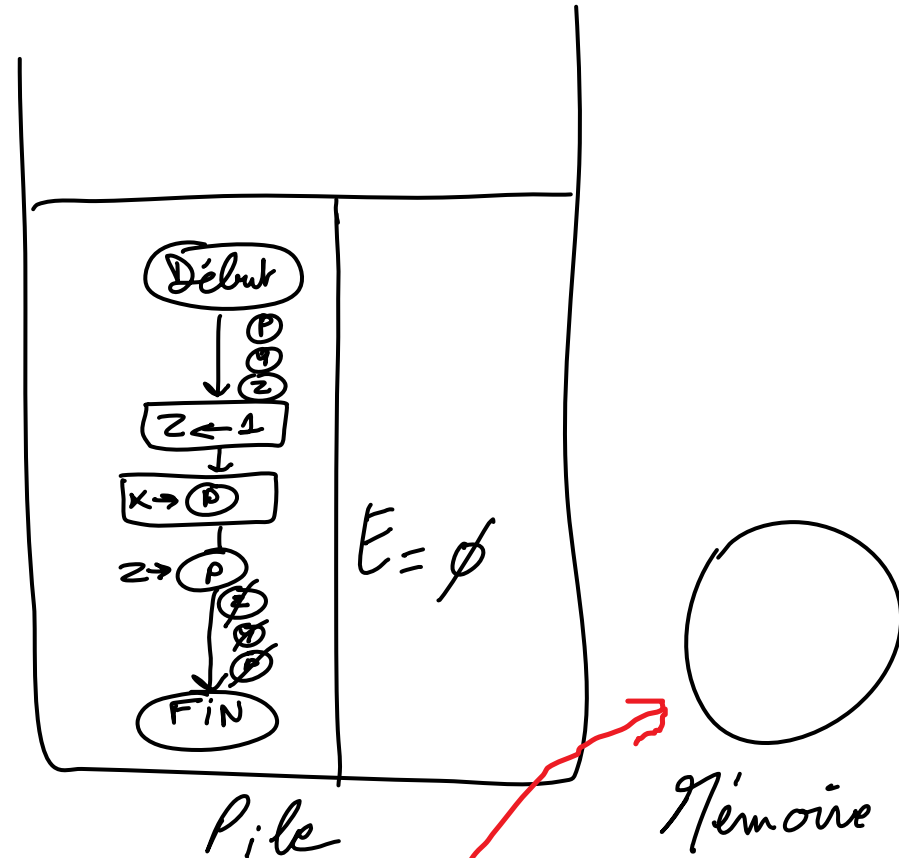
```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```



Environnement initial vide

Début de l'exécution

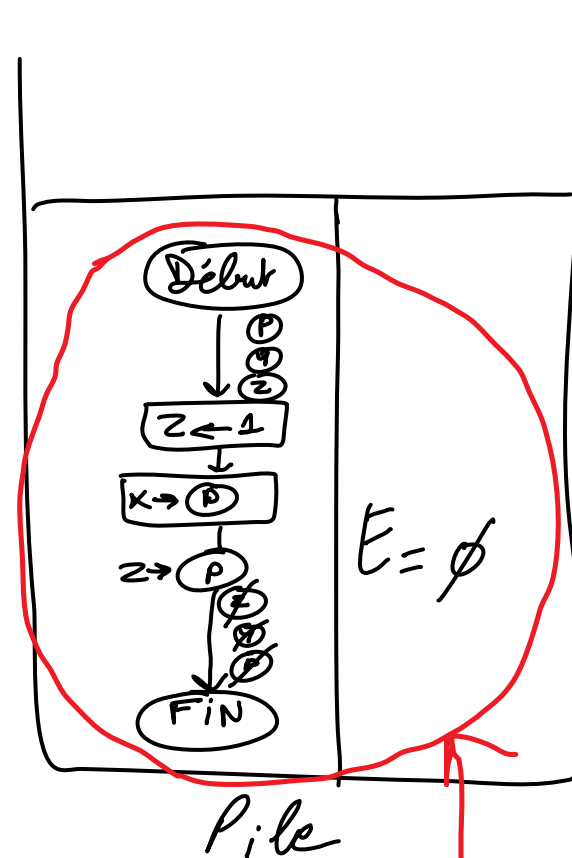
```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```



Mémoire vide

Début de l'exécution

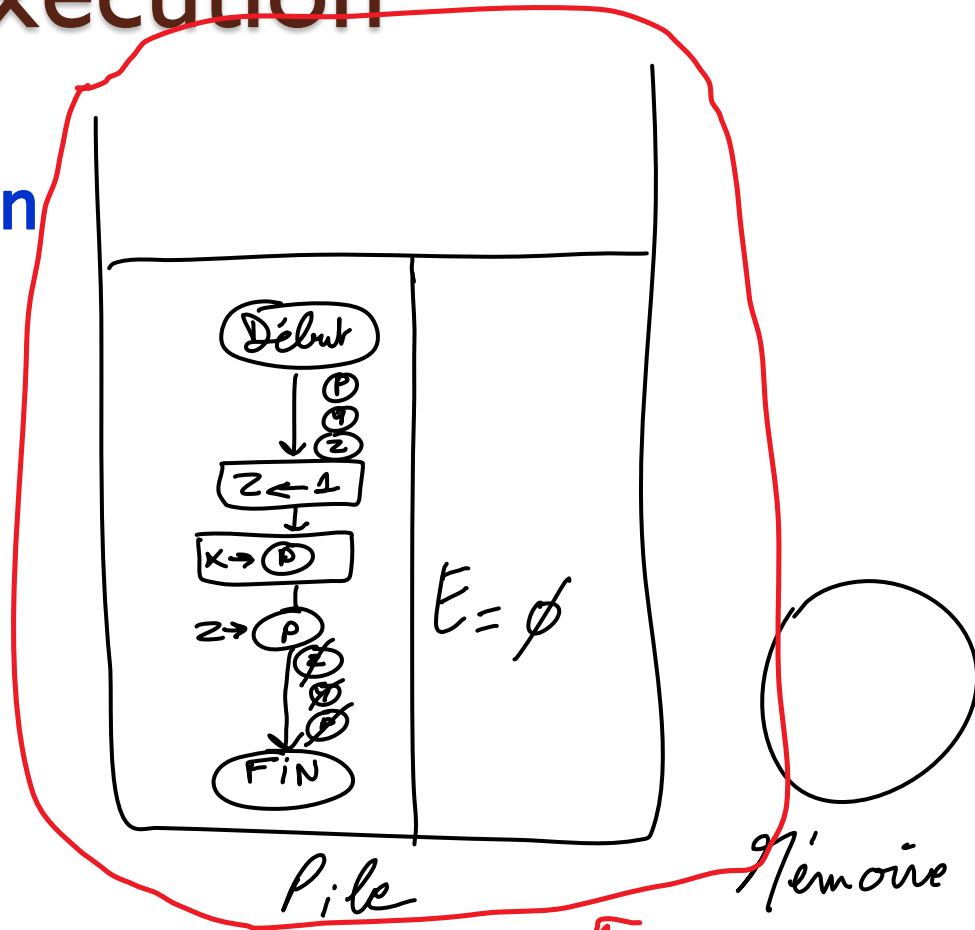
```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```



Première instruction sémantique

Début de l'exécution

```
([(local P Y Z in
  Z=1
  proc {P X}
    Y=X
  end
  {P Z}
end, ∅)],
∅)
```



Pile sémantique initiale (qui contient une instruction sémantique)