[Calculs avec des environnements]

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.

Adjonction et restriction

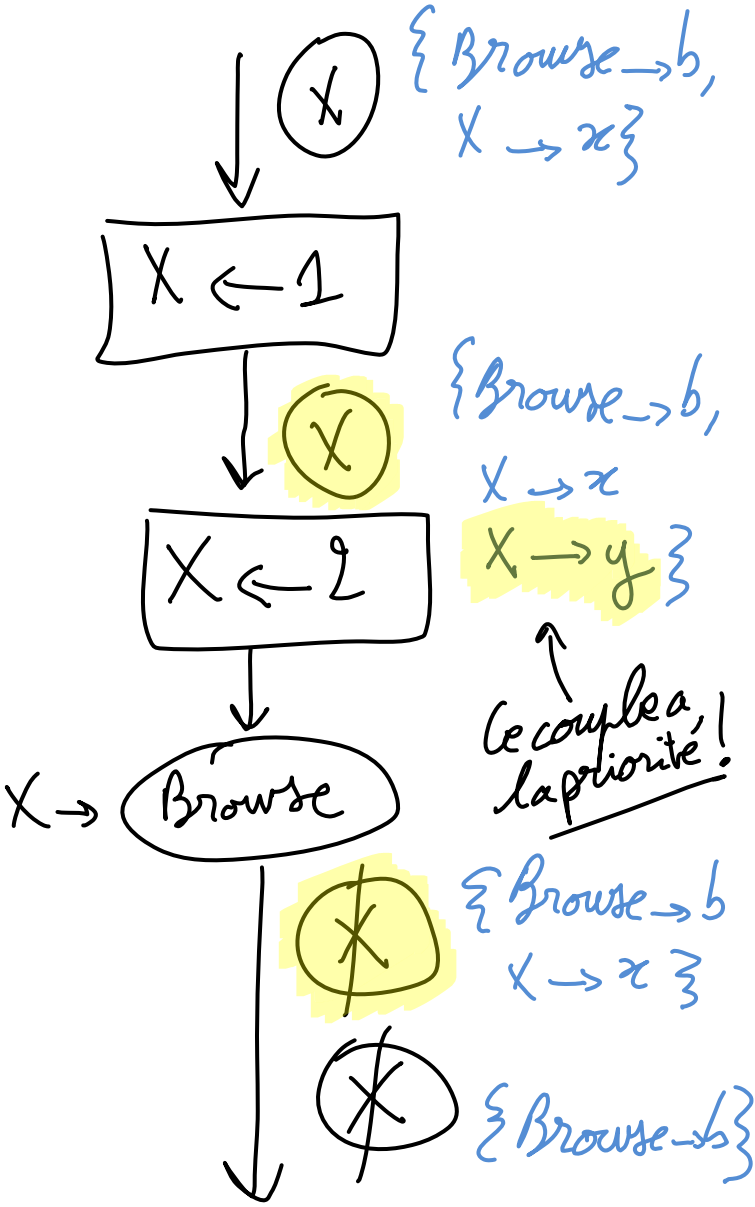
- Pendant l'exécution, on fait deux sortes de calculs avec les environnements
- Adjonction: $E_2 = E_1 + \{X \rightarrow y\}$
  - Ajouter un lien dans un environnement
  - Pour **local** <x> **in** <s> **end**
- Restriction: $CE = E_{|\{X,Y,Z\}}$
  - Enlever des liens d'un environnement
  - Pour le calcul d'un environnement contextuel

L'adjonction

- Pour une instruction **local**

```
local X in (E1)
      X=1
      local X in (E2)
            X=2
            {Browse X}
      end
end
```

- E_1={Browse $\rightarrow b$, $X \rightarrow x$}
- E_2=E_1+{$X \rightarrow y$}
- E_2={Browse $\rightarrow b$, $X \rightarrow y$}



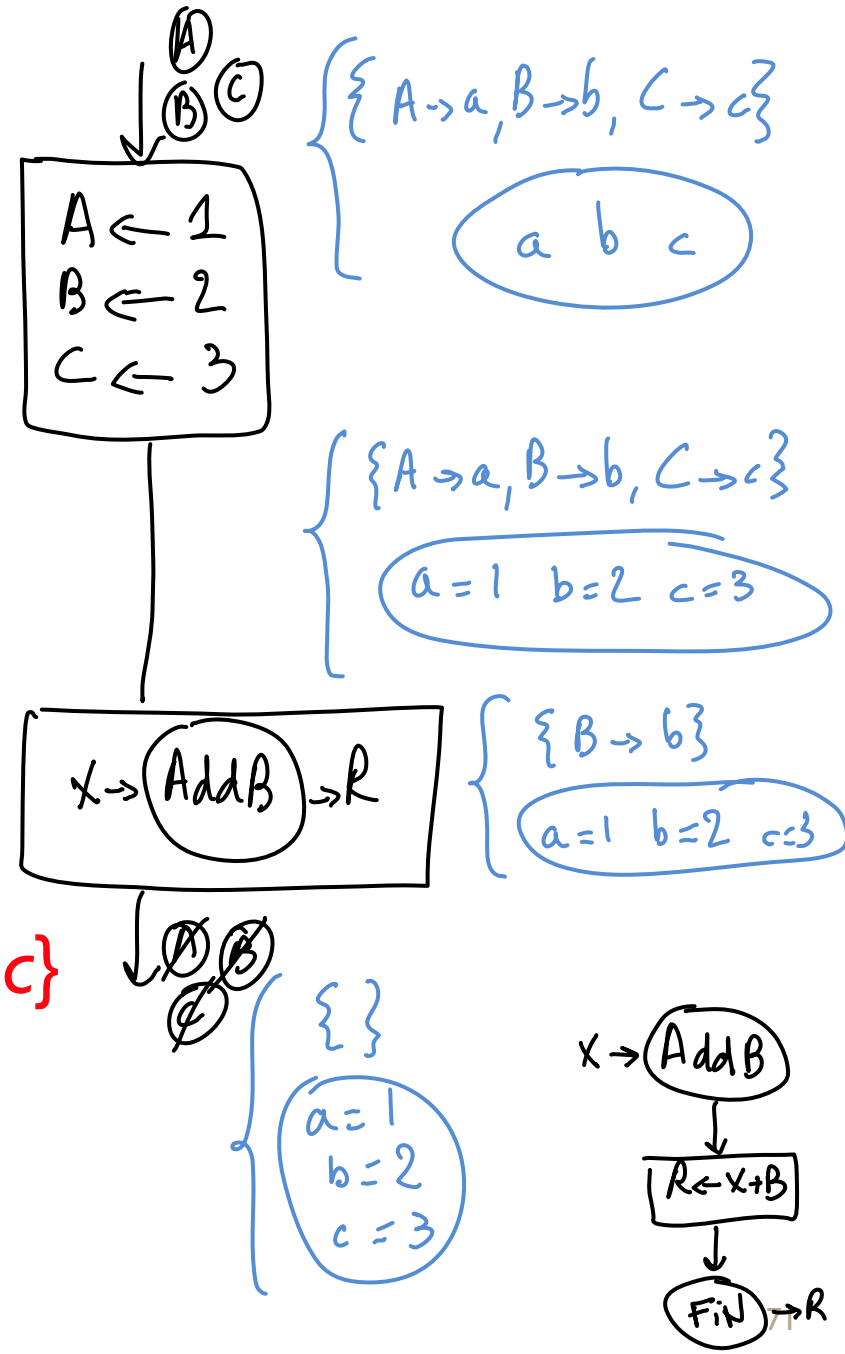
➔ La dernière variable associée à un identificateur a la priorité sur les autres.

La restriction

- Pour une définition de procédure

```
local A B C in
      A=1  B=2 C=3 (E)
      fun {AddB X}
            X+B
      end
end
```

- E={A → *a*, B → *b*, C → *c*}
  σ={*a*=1,*b*=2,*c*=3}
- CE=$E_{|\{B\}}$={B → *b*}



[Sémantique de chaque instruction]

pptPlex Section Divider

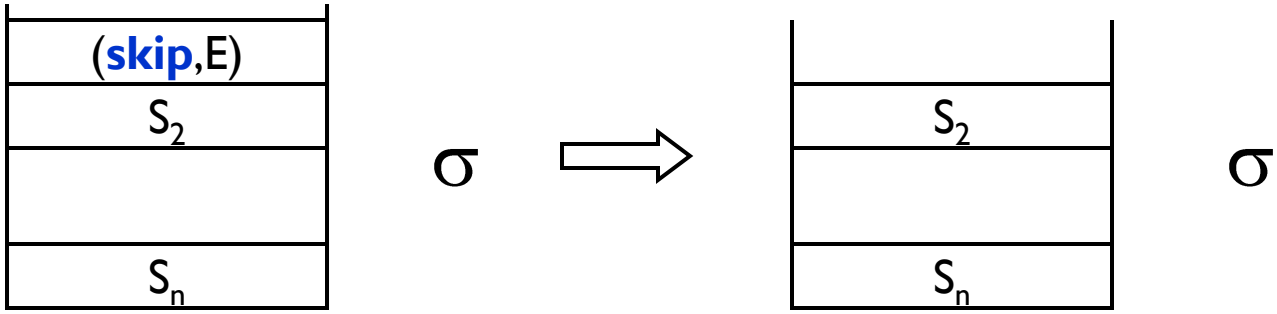
The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Sémantique des instructions

- Pour chaque instruction du langage noyau, il faut préciser ce qu'elle fait avec l'état d'exécution
- Chaque instruction prend un état d'exécution et donne un autre état d'exécution
  - Etat d'exécution = pile sémantique + mémoire
- Rappel des instructions du langage noyau
  - **skip**
  - <s>$_1$ <s>$_2$ (composition séquentielle)
  - **local** <x> **in** <s> **end** (déclaration de variable)
  - <x>=<y> (liaison de variables)
  - **if** <x> **then** <s>$_1$ **else** <s>$_2$ **end** (conditionnel)
  - <x>=<v> (affectation ou "création de valeur")
    - Création de nombre et d'enregistrement
    - Définition de procédure
  - **{**$ <x>$_1$ … <x>$_n$**}** (appel de procédure)
  - **case** <x> **of** <p> **then** <s>$_1$ **else** <s>$_2$ **end** (correspondance de formes)

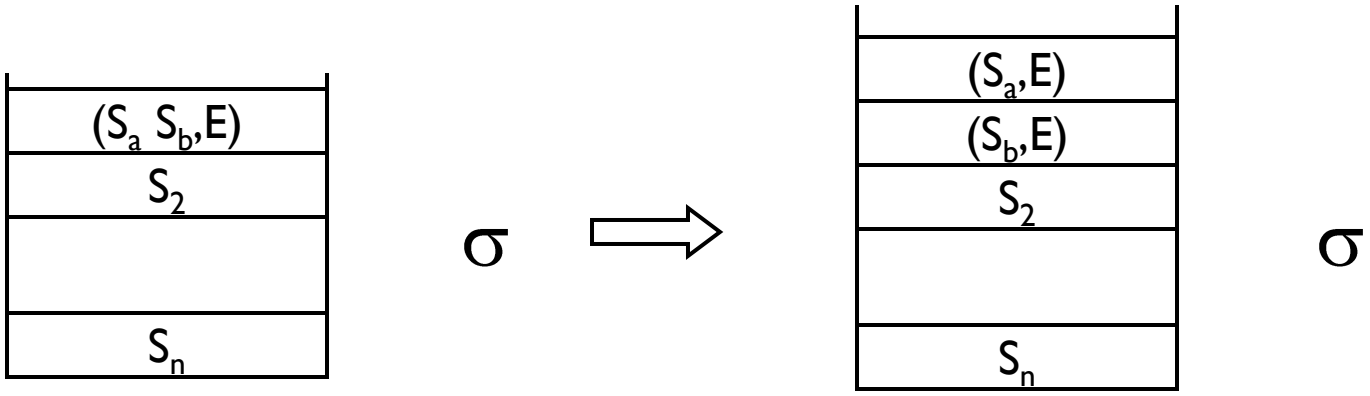
Skip (instruction vide)

- L'instruction la plus simple
- Elle ne fait "rien"!
- Etat de l'entrée: ([(**skip**,E), S_2, …, S_n], σ)
- Etat de la sortie: ([S_2, …, S_n], σ)
- C'est tout!



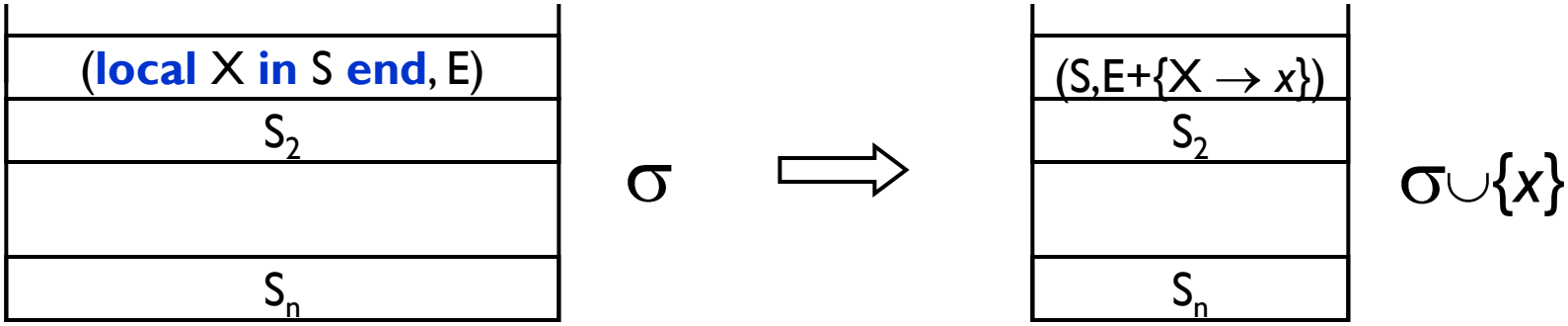
$\langle s \rangle_1 \langle s \rangle_2$ (composition séquentielle)

- Presque aussi simple
- L'instruction
  - enlève le sommet de la pile
  - et en ajoute deux nouveaux éléments avec recopie de l'envronnement



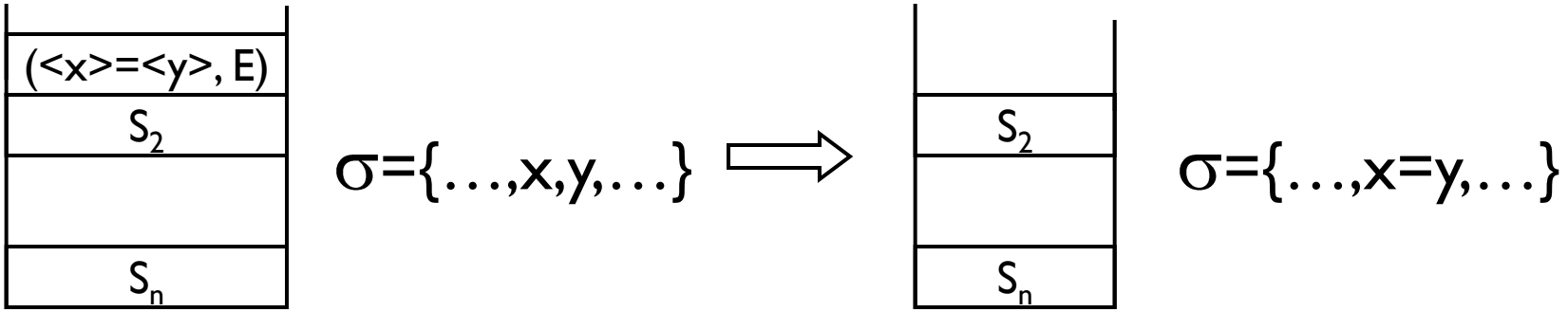
local <x> in <s> end (déclaration de variable)

- Créez la nouvelle variable *x*
- Ajoutez la correspondance {X → *x*} à l'environnement E (l'adjonction)
- La dernière variable associée à un identificateur a la priorité sur les autres.



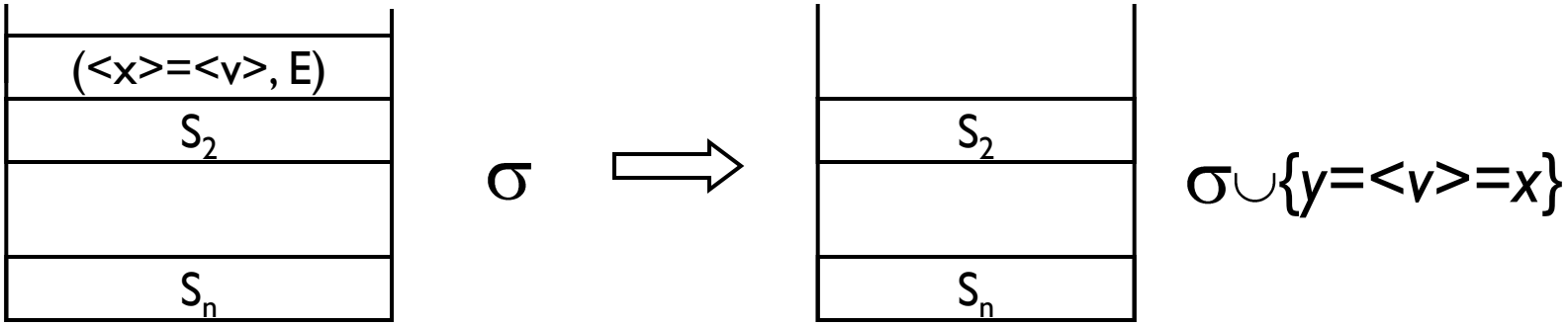
<x>=<y> (le lien variable-variable)

- Soit E l'environnement associé
- Action effectuée
  - Lier E(<x>) et E(<y>) en mémoire.

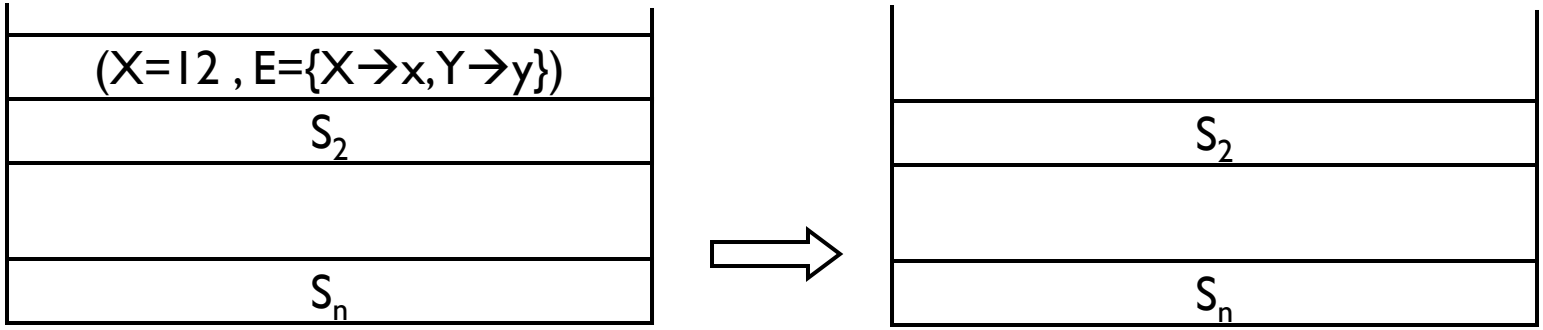


<x>=<v> (l'affectation)

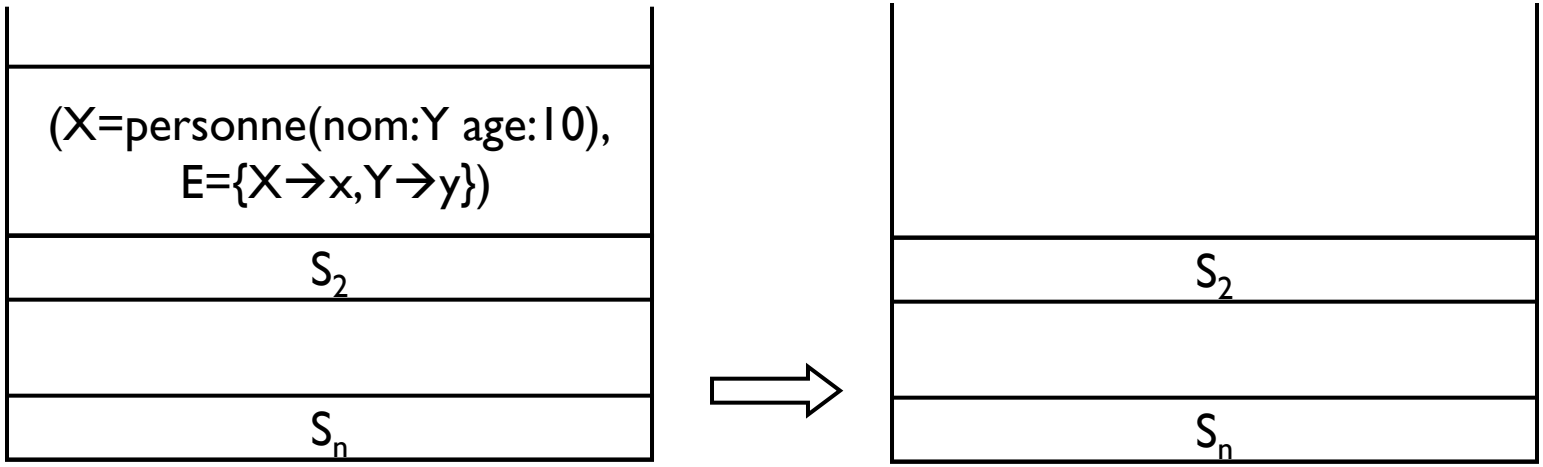
- Soit E l'environnement associé
- Affectation, ou création de valeur
- Actions effectuées
  - Création d'une nouvelle variable y en mémoire.
  - Construction de la valeur représentée par <v> dans la mémoire
    Tous les identificateurs dans <v> sont remplacés par leur contenu en mémoire donné par E.
  - Liaison de y à cette valeur
  - Lier E(<x>) et y en mémoire.
- 2 cas de construction de valeur
  - <v> est un enregistrement ou un nombre
  - <v> est une valeur procédurale (cf transparents sur les procédures)



Exemples d'affectation



$\sigma=\{y, x\}$ $\sigma=\{y, x=y_2=12\}$

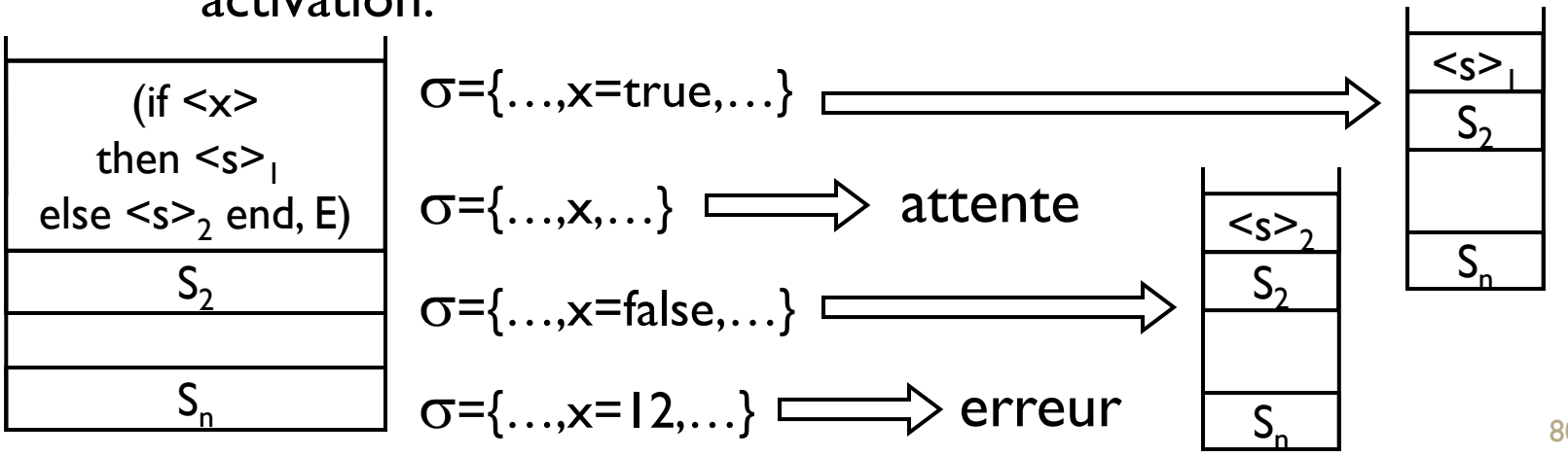


$\sigma=\{y=Jean, x\}$

$\sigma=\{y,$
$x=y_2=personne(nom:Jean\ age:10)\}$

if <x> then <s>$_1$ else <s>$_2$ end (instruction conditionnelle)

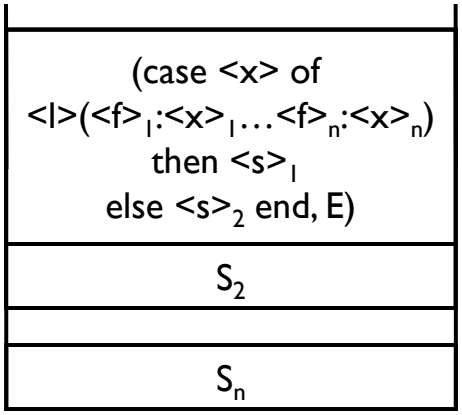
- La condition d'activation
  "<x> est lié à une valeur" dans la mémoire
- Etapes d'exécution
  - Si activation vraie
    - Si E(<x>) n'est pas un booléen, lever une condition d'erreur
    - Si E(<x>) est true, empiler (<s>$_1$, E)
    - Si E(<x>) est false, empiler (<s>$_2$, E)
  - Si l'activation est fausse, suspendre l'exécution jusqu'à activation.

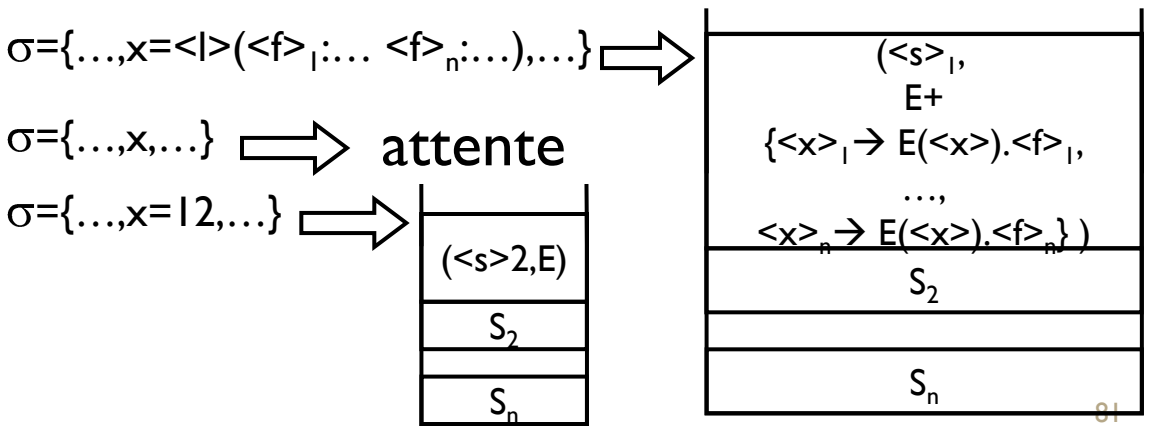


case <x> of <p> then $\langle s \rangle_1$ else $\langle s \rangle_2$ end (correspondance de formes)

(**case** $\langle x \rangle$ **of** $\langle lit \rangle(\langle feat \rangle_1 : \langle x \rangle_1 \cdots \langle feat \rangle_n : \langle x \rangle_n)$ **then** $\langle s \rangle_1$ **else** $\langle s \rangle_2$ **end**, E)

- Si la condition d'activation est vraie ($E(\langle x \rangle)$ est déterminée), faire les actions suivantes :
  - Si l'étiquette de $E(\langle x \rangle)$ est $\langle lit \rangle$ et son arité est $\{\langle feat \rangle_1, \ldots, \langle feat \rangle_n\}$, empiler $(\langle s \rangle_1, E + \{\langle x \rangle_1 \rightarrow E(\langle x \rangle).\langle feat \rangle_1, \ldots, \langle x \rangle_n \rightarrow E(\langle x \rangle).\langle feat \rangle_n\})$.
  - Sinon empiler $(\langle s \rangle_2, E)$.
- Si la condition d'activation est fausse, suspendre l'exécution.





[Les procédures en mémoire]

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.

La sémantique des procédures

- Définition de procédure
  - Il faut créer l'environnement contextuel
- Appel de procédure
  - Il faut faire le lien entre les arguments formels et les arguments actuels
  - Il faut utiliser l'environnement contextuel

Une procédure en mémoire est une valeur!

- Une procédure est stockée en mémoire, tout comme n'importe quelle valeur
  - Rappel des 3 types de valeurs : nombres, enregistrements, procédures

```
proc {Inc X Y} Y=X+1 end
Inc=proc {$ X Y} Y=X+1 end
```

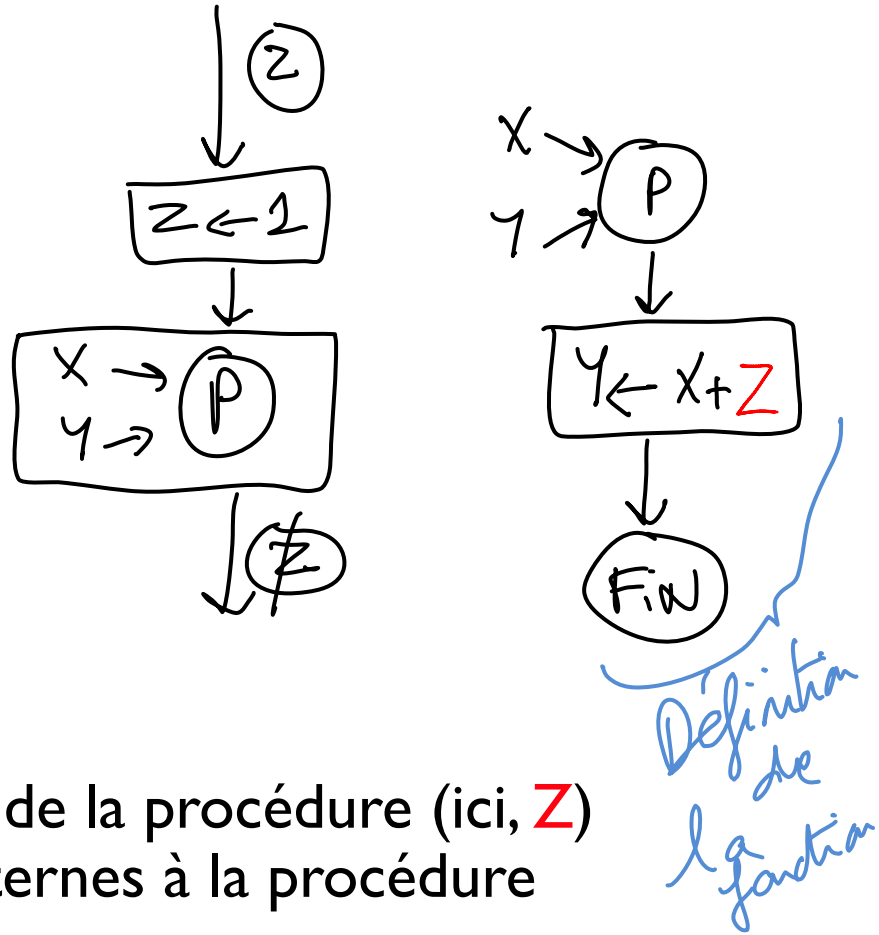
- La valeur est "**proc** {$ X Y} Y=X+1 **end**"
  - L'identificateur Inc, le "nom" de la procédure, ne fait pas partie de la valeur
- Analogie avec les entiers:
  declare N=23
  - La valeur est 23, l'identificateur N ne fait pas partie de la valeur

Importance de la valeur procédurale

- Une valeur procédurale contient deux choses
  - Le code de la procédure et l'environnement contextuel
  - L'appel de la procédure construit l'environnement de l'appel:
    - on commence avec l'environnement contextuel
    - on y ajoute les arguments formels
- La valeur procédurale est un des concepts les plus puissants dans les langages de programmation
  - Première utilisation en Algol 68 (appellé "closure" – "fermeture")
  - Existe dans quasi tous les langages, parfois accessible au programmeur (notamment dans les langages fonctionnels)
- Plus difficile à utiliser en Java, C, C++ !
  - Procédure/fonction/méthode: il n'y a que du code
  - L'environnement contextuel manque (sauf pour un "inner class")
  - En Java et C++ on peut simuler une valeur procédurale avec un objet

L'environnement contextuel

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



- Les identificateurs libres de la procédure (ici, Z) prennent des valeurs externes à la procédure
- Ils sont dans l'environnement contextuel, qui doit faire partie de la procédure!

Définition d'une procédure

- L'instruction sémantique de la définition est
  (<x>=**proc** {$ <x>$_1$ … <x>$_n$} <s> **end**, E)

  ou en raccourci
  (**proc** {<x> <x>$_1$ … <x>$_n$} <s> **end**, E)

- Arguments (paramètres) formels:
  <x>$_1$, …, <x>$_n$
- Identificateurs libres dans <s>:
  <z>$_1$, …, <z>$_k$
- Environnement contextuel:
  CE=E$_{|<z>1, …, <z>k}$ (restriction de E)
- Valeur en mémoire:
  x=(**proc** {$ <x>$_1$ … <x>$_n$} <s> **end**, CE)

Ni les paramètres
Ni les var locales

Environnement restreint aux identificateurs libres

Appel de procédure (1)

- L'instruction sémantique est

$$(\{\langle x\rangle\ \langle y\rangle_1\ \ldots\ \langle y\rangle_n\}, E)$$

- Si la condition d'activation est fausse ($E(\langle x\rangle)$ pas lié)
  - Suspension (attente d'exécution)
- Si $E(\langle x\rangle)$ n'est pas une procédure
  - Erreur
- Si $E(\langle x\rangle)$ est une procédure mais le nombre d'arguments n'est pas bon ($\neq n$)
  - Erreur

Appel de procédure (2)

- L'instruction sémantique est

  $$(\{\langle x\rangle\ \langle y\rangle_1\ \ldots\ \langle y\rangle_n\}, E)$$

  avec

  $$E(\langle x\rangle) = (\textbf{proc}\ \{\$\ \langle z\rangle_1 \ldots \langle z\rangle_n\}\ \langle s\rangle\ \textbf{end}, CE)$$

- Alors empilez

  $$(\langle s\rangle, CE + \{\langle z\rangle_1 \rightarrow E(\langle y\rangle_1), \ldots, \langle z\rangle_n \rightarrow E(\langle y\rangle_n)\})$$