

[Passage au langage pratique]

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.

Elements pour le langage pratique

- « Sucre syntaxique »
 - Définitions de raccourcis syntaxiques
➔ meilleure lisibilité du code
 - Exemple
 - elseif
 - case avec plusieurs valeurs
- Définition d'abstractions linguistiques
 - Définition de la fonction
- Définition d'abstractions de contrôle
 - Définition de la boucle tant que
- Ajout de concept
 - Les exceptions

Définition de fonction

- Langage pratique

```
fun {F X1 ... XN} <statement> <expression> end
```

- Langage noyau

```
proc {F X1 ... XN ?R} <statement> R=<expression> end
```

- Exemple

```
fun {Max X Y}  
  if X>=Y then X else Y end  
end
```



```
proc {Max X Y ?R}  
  R = if X>=Y then X else Y end  
end
```

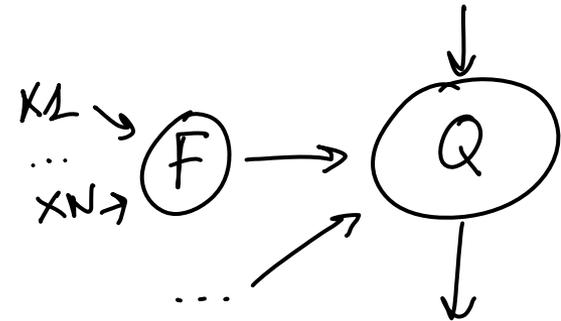


```
proc {Max X Y ?R}  
  if X>=Y then R=X else R=Y end  
end
```

Appel de fonction - exemple

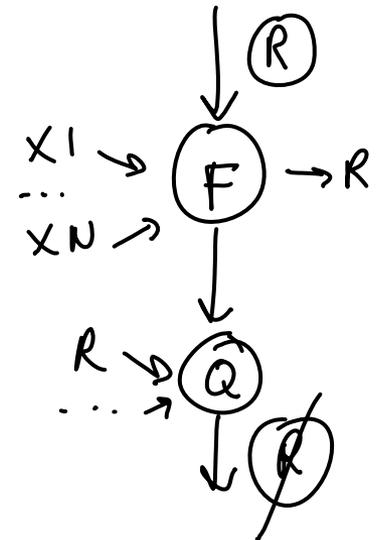
- Langage pratique

```
{Q {F X1 ... XN} ...}
```



- Langage noyau

```
local R in  
  {F X1 ... XN R}  
  {Q R ...}  
end
```



Le calcul itératif

- Schéma général d'une boucle tant que

```
fun {Iterate  $S_i$ }  
  if {IsDone  $S_i$ } then  $S_i$   
  else  $S_{i+1}$  in  
     $S_{i+1} = \{Transform\ S_i\}$   
    {Iterate  $S_{i+1}$ }  
  end  
end
```

- Traduction en code :

paramétriser les fonctions IsDone et Transform

(programmation d'ordre supérieur)

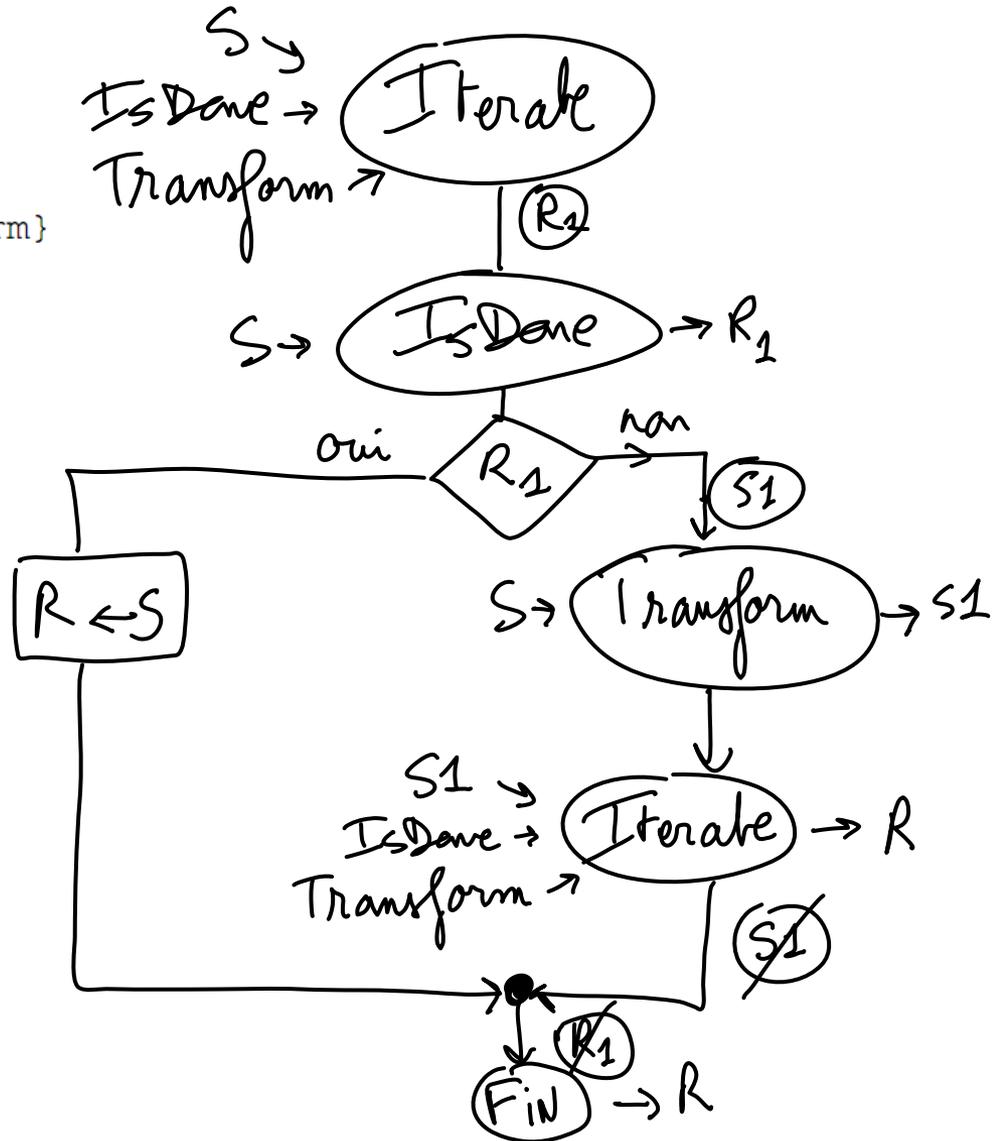
```
fun {Iterate S IsDone Transform}  
  if {IsDone S} then S  
  else S1 in  
    S1 = {Transform S}  
    {Iterate S1 IsDone Transform}  
  end  
end
```

Le calcul itératif

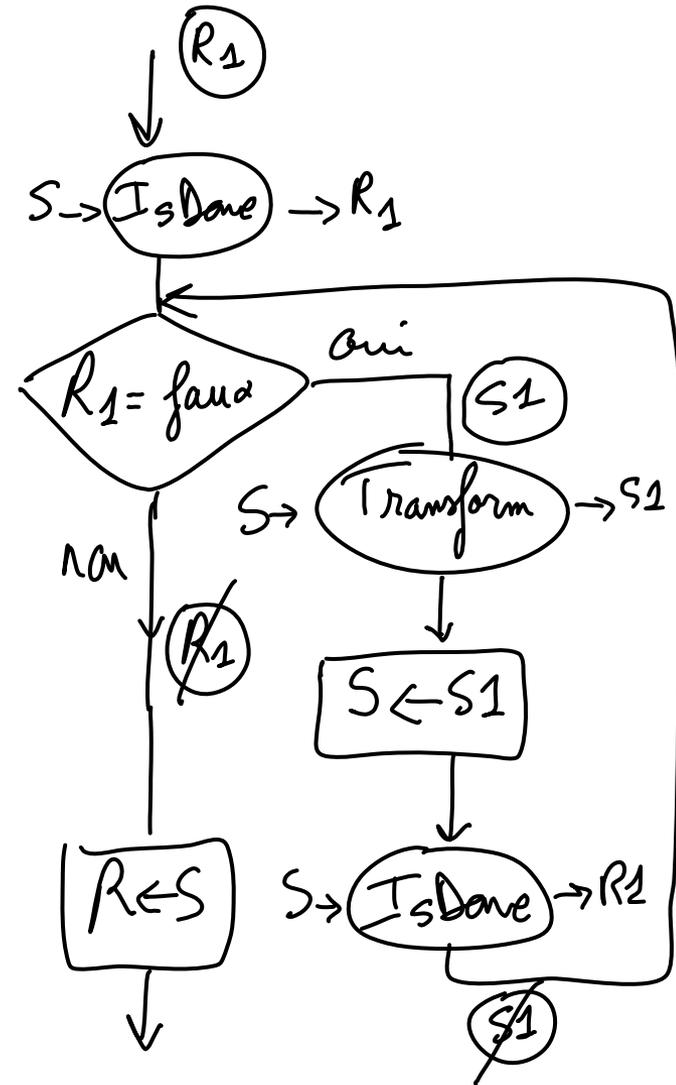
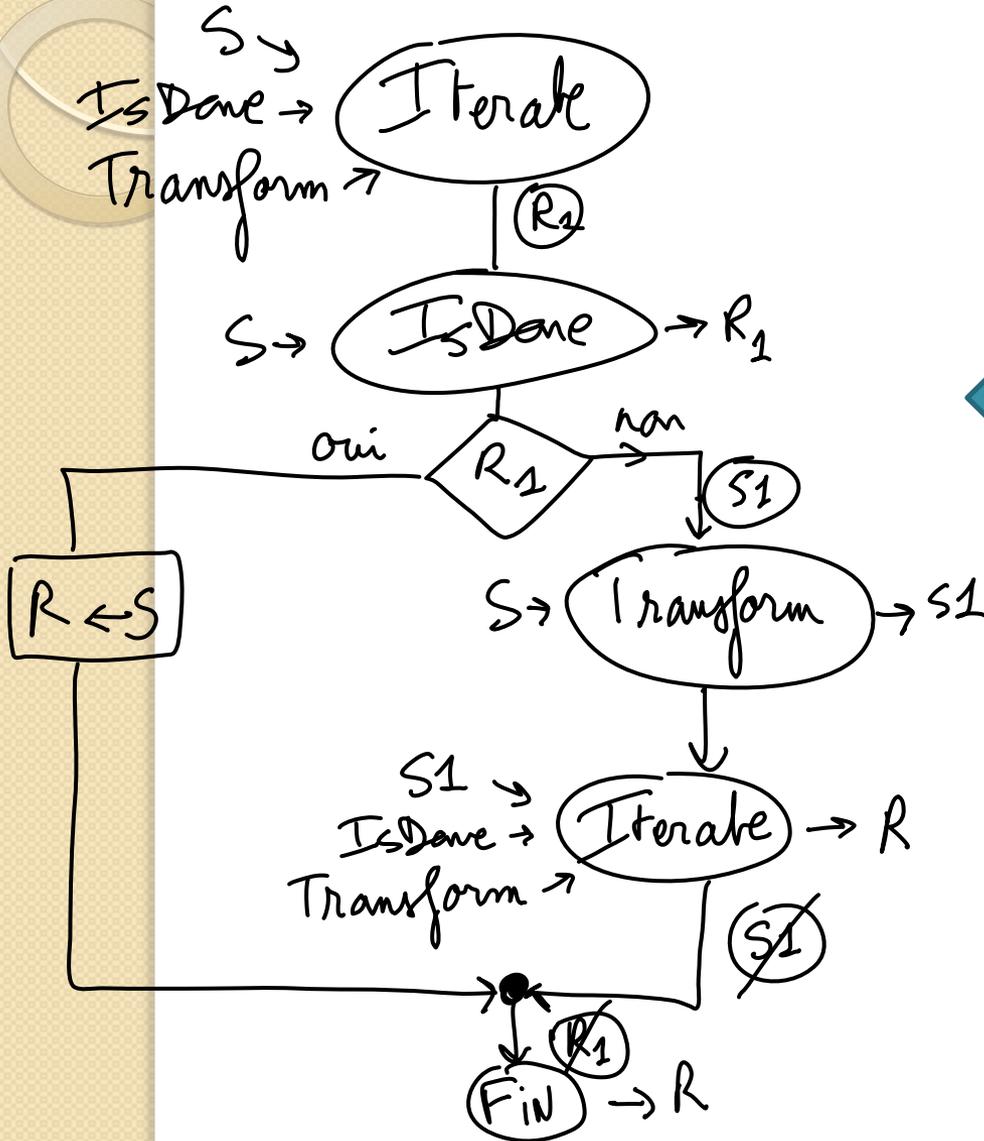
```

fun {Iterate S IsDone Transform}
  if {IsDone S} then S
  else S1 in
    S1={Transform S}
    {Iterate S1 IsDone Transform}
  end
end

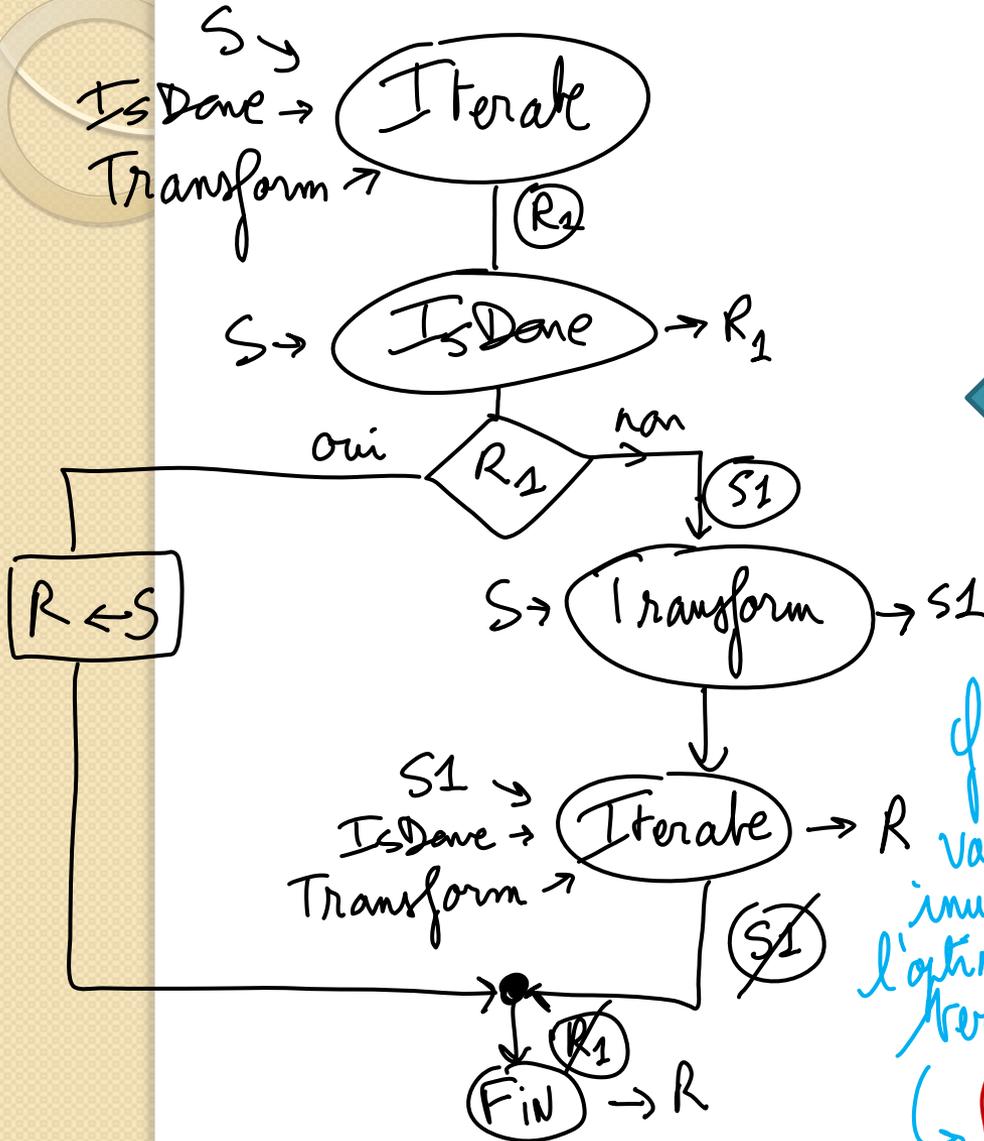
```



Le calcul itératif

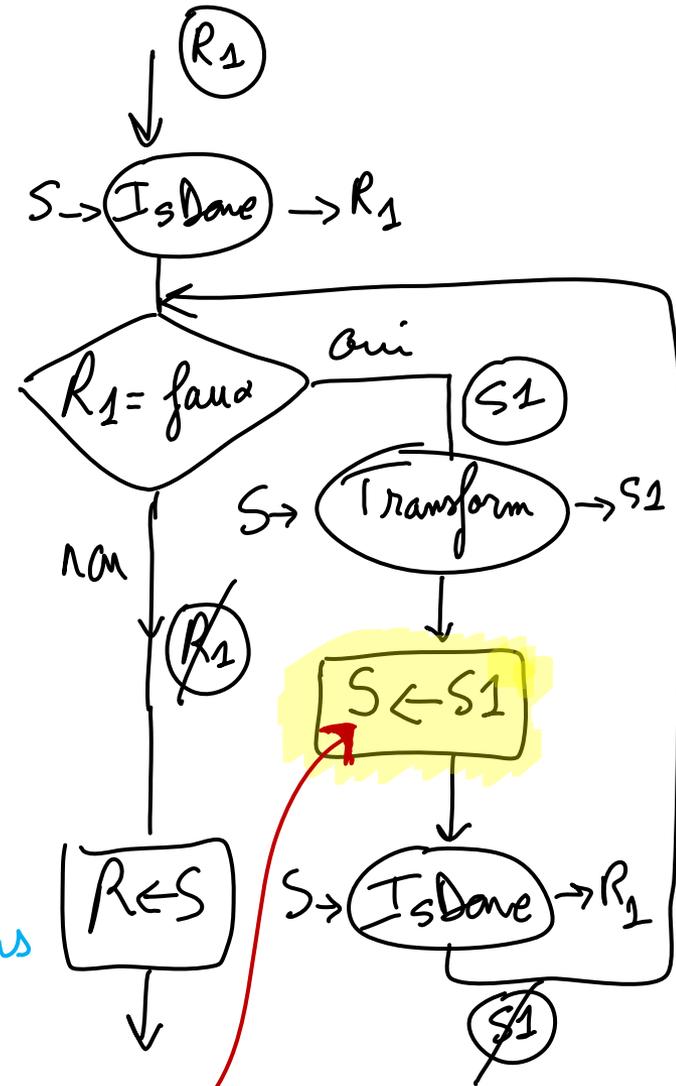


Le calcul itératif



*doublé
des
variables
inutiles dans
l'optimisation
terminale*

S doit être une variable à affectation multiple !!



Le calcul itératif - exemple

- Méthode de Newton pour calculer la racine d'un nombre réel x positif
 - Idée proposer une succession d'estimations de \sqrt{x} , de plus en plus précises jusqu'à une précision donnée
 - Entre deux estimations g et g' :

$$g' = (g + (x/g)) / 2$$
 - Exemple : $x=5$
 - Première estimation de $\sqrt{5}$: $g=1$
 - Deuxième estimation : $g' = (1 + (5/1)) / 2 = 3$
 - Troisième estimation : $g'' = (3 + (5/3)) / 2 = 14/6 = 2,33333$
 - Etc...
- Démonstration de l'amélioration de l'estimation

Soit $\varepsilon = g - \sqrt{x}$ (ε est la différence entre la vraie valeur et l'estimation)

De même $\varepsilon' = g' - \sqrt{x}$

$$\varepsilon' = g' + \varepsilon - g \quad \text{car } \varepsilon = g - \sqrt{x}$$

$$\varepsilon' = \left(\frac{g + (x/g)}{2} \right) + \varepsilon - g \quad \text{car } g' = \frac{g + (x/g)}{2}$$

$$= \frac{g^2 + x + 2g\varepsilon - 2g^2}{2g}$$

$$= \frac{\cancel{g^2} + \varepsilon^2 + \cancel{g^2} - \cancel{2\varepsilon g} + \cancel{2\varepsilon g} - \cancel{2g^2}}{2g} \quad \text{car } x = (\varepsilon - g)^2$$

D'où

$$\varepsilon' = \frac{\varepsilon^2}{2g}$$

A-t-on $\varepsilon' < \varepsilon$?

(sinon divergence)

$$\varepsilon' < \varepsilon \Leftrightarrow \frac{\varepsilon^2}{2g} < \varepsilon$$

$$\Leftrightarrow \varepsilon < 2g$$

(en supposant $\varepsilon > 0$)

$$\Leftrightarrow \sqrt{x} + g > 0$$

Toujours vrai si
 $x > 0$ et $g > 0$

Le calcul itératif - exemple

- Méthode de Newton pour calculer la racine d'un nombre réel x positif
 - Idée proposer une succession d'estimations de \sqrt{x} , de plus en plus précises jusqu'à une précision donnée
 - Entre deux estimations g et g' :
$$g' = (g + (x/g)) / 2$$
 - Exemple : $x=5$
 - Première estimation de $\sqrt{5}$: $g=1$
 - Deuxième estimation : $g' = (1 + (5/1)) / 2 = 3$
 - Troisième estimation : $g'' = (3 + (5/3)) / 2 = 14/6 = 2,33333$
 - Etc...
- Question :
Quel appel à l'iterate pour résoudre ce problème ?

Le calcul itératif - exemple

- Méthode de Newton pour calculer la racine d'un nombre réel x positif
 - Idée proposer une succession d'estimations de \sqrt{x} , de plus en plus précises jusqu'à une précision donnée
 - Entre deux estimations g et g' :
 $g' = (g + (x/g)) / 2$
 - Exemple : $x=5$
 - Première estimation de $\sqrt{5}$: $g=1$
 - Deuxième estimation : $g' = (1 + (5/1)) / 2 = 3$
 - Troisième estimation : $g'' = (3 + (5/3)) / 2 = 14/6 = 2,33333$
 - Etc...
- Question :
Quel appel à Iterate pour résoudre ce problème ?

```
fun {Sqrt X}  
  {Iterate  
    1.0
```

```
    fun {$ G} {Abs X-G*G}/X<0.00001 end  
    fun {$ G} (G+X/G)/2.0 end}  
end
```

IsDone
Transform

[Conclusion La programmation déclarative]

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.

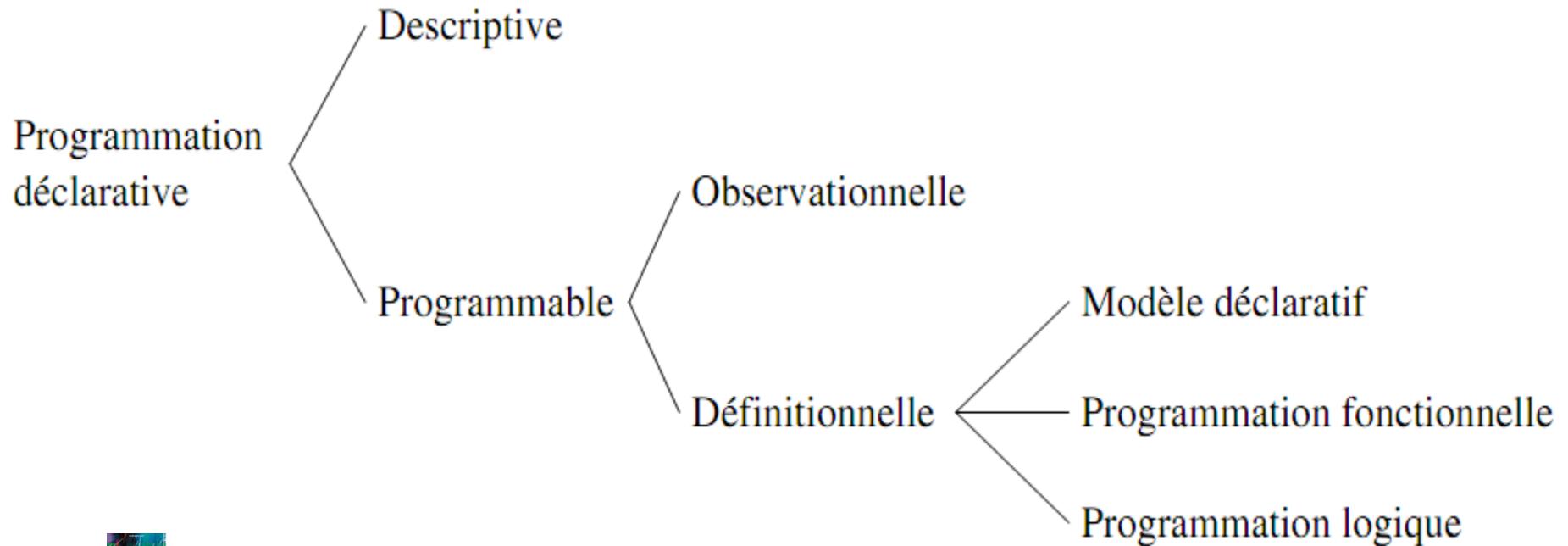
Définition

- Programmer en donnant une définition des résultats, sans donner les algorithmes nécessaires pour les calculer.
→ définition assez vague
- Une opération déclarative a trois propriétés
 - Indépendance
 - Ne dépend pas d'un état d'exécution en dehors d'elle-même
 - E.g. pas de variable « globale »
 - Sans état
 - Ne garde pas d'état d'exécution interne conservé entre 2 appels
 - Déterministe
 - Calcule toujours le même résultat pour les mêmes entrées
- Un programme déclaratif
 - Est compositionnel
 - Faits de composants pouvant être écrits, testés et prouvés indépendamment
 - Facilite le raisonnement

Remarques

- Les programmes écrits en langage noyau sont déclaratifs car les opérations de base le sont.
- Ecriture de programmes déclaratifs
 - Comme des ensembles de fonctions récursives
 - Utiliser l'ordre supérieur pour construire des abstractions facilitant l'écriture

Une classification



La programmation déclarative descriptive

- La moins expressive
- Définit seulement une structure de données
- Langages pouvant définir des enregistrements
- Exemple

$\langle s \rangle$::= skip	Instruction vide
	$\langle s \rangle_1 \langle s \rangle_2$	Séquence d'instructions
	local $\langle x \rangle$ in $\langle s \rangle$ end	Création de variable
	$\langle x \rangle_1 = \langle x \rangle_2$	Lien variable-variable
	$\langle x \rangle = \langle v \rangle$	Création de valeur

Tableau 3.1 Le langage noyau déclaratif descriptif.

- Autres exemples : HTML, XML



La programmation déclarative programmable

- Aussi expressive qu'une machine de Turing
- Vue observationnelle
 - Les composants doivent se comporter de façon déclarative
 - Peuvent être implémentés de façon non déclarative
- Vue définitionnelle
 - La déclarativité est une propriété de l'implémentation des composants.
 - Style fonctionnel
 - Un composant est défini comme une fonction mathématique
 - Haskell, Standard ML
 - Style logique
 - Un composant est défini comme une relation logique
 - Prolog, Mercury
 - Les 2 styles : langage noyau

Au prochain épisode

- Steven et Samantha viennent de faire un joli petit programme déclaratif, mais ils voudraient encore agrandir la famille...
- Jonathan leur conseille d'aller voir Bjarne, spécialiste de la conception des objets.
- Kerry, toujours sceptique sur les vraies intentions de Steven, décide de recueillir un python, symbole de la connaissance.
- Mais ... et Christopher dans tout cela ?

A suivre...