



# Le langage WHILE

compilateur

Olivier Ridoux

# Interface du compilateur (1)

- **whc f.wh [-o g]**

- compile les fonctions WHILE contenues dans le fichier **f.wh**

- produit un exécutable dans le fichier **f** par défaut...

- ...ou dans le fichier **g**, si l'option **-o g** est utilisée

- la dernière fonction du fichier **f.wh** est le point d'entrée du programme

whc

# Interface du compilateur (1)

- **whc -help**

- affiche le manuel de whc dans le style man de UNIX

whc

# Interface de l'exécutable (1)

- $f\ e_1\ e_2\ \dots\ e_N$

- les chaînes  $e_1, e_2, \dots, e_N$  sont analysées selon la syntaxe des expressions WHILE (restreintes aux symboles et à **nil**, **cons** et **list**), ... ou comme des notations d'entiers

...elles sont traduites en des représentations de termes WHILE

...et affectées aux variables **read** de la fonction point d'entrée

whc

# Interface de l'exécutable (2)

- Si le point d'entrée de **f.wh** est **p**...

[unix] % **f nil** revient à l'appel **(p nil)**

[unix] % **f "(cons a b)"** revient à l'appel **(p (cons a b))**

[unix] % **f 3** revient à l'appel

**(p (cons nil (cons nil (cons nil nil))))**

**≡ (p (list nil nil nil))**

# Pas d'erreurs à l'exécution (1)

- Gérer proprement les erreurs à l'exécution est **complexe**...

...ne pas les gérer proprement est **irresponsable**

→ s'arranger pour qu'il n'y ait **pas d'erreur** à l'exécution

# Pas d'erreurs à l'exécution (1)

- Convention **conciliante** pour les paramètres effectif du programme
  - si trop de paramètres effectifs...  
...les passer dans une liste au dernier paramètre formel
  - si pas assez de paramètres effectifs...  
...utiliser la valeur par défaut (**nil**)

# Pas d'erreurs à l'exécution (3)

function p: read  $V_1$ ,  $V_2$  % ...

[unix] % **p a b c d**

$V_1 \leftarrow a$

$V_2 \leftarrow (\text{list } b \ c \ d)$

[unix] % **p a**

$V_1 \leftarrow a$

$V_2 \leftarrow \text{nil}$

whc