

Le langage WHILE

pretty-printer

Olivier Ridoux

The WHILE pretty-printer (1)

- Une ligne
 - une commande élémentaire entière
 - **nop** ou **vars := exprs**
 - une tête de commande composée
 - **if e then, while c do, for c do**
 - une fin de commande composée
 - **fi, od, else**
 - un **%**
 - Un **function** ..., un **read** ... ou un **write** ...

The WHILE pretty-printer (2)

- L'indentation
 - valeur par défaut
 - configurable, ex. 2
 - valeurs paramétrables pour chaque structure de contrôle
 - **while, for, if**
 - indentation nulle (= 0) pour
 - **read, write** et **%**
 - indentation identique pour tête et fin d'une même commande

The WHILE pretty-printer (3)

- Le ';' toujours sur la même ligne que la commande qui précède

The WHILE pretty-printer (4)

- Un espace
 - de part et d'autre de **:=** et **=?**
 - après les **function**, **read**, **write**, **if**, **while**, **for**, **cons**, **hd**, **tl**, etc
 - avant les **;**, **do**, **then**, etc
 - après les **:**
 - entre les paramètres

The WHILE pretty-printer (5)

- Pas d'espace
 - après '('
 - avant ')'

The WHILE pretty-printer (6)

- *WH tools*

- **whpp**

- *WHILE pretty-printer*

- **whpp sth.wh –all 1 –if 3 –o pretty-sth.wh**

- **whpp –help**

- ...

- **whc**

- *WHILE compiler*

- **whc sth.wh –o sth**

- ...

The WHILE pretty-printer (7)

- whpp sth.wh -all 1 -if 3 -o pretty-sth.wh
 - all **X** : valeur par défaut de l'indentation, par défaut **X=2**
 - if Y : valeur spécifique pour les blocs if, par défaut **X**
 - ... : pareil pour toutes les autres structures de contrôle
 - o : nom du fichier de sortie, par défaut : sth.whpp

The WHILE pretty-printer (8)

- whpp –help
 - aide en ligne, sur le modèle du man UNIX

NAME

SYNOPSIS

DESCRIPTION

OPTIONS

BUGS

SEE ALSO

AUTHORS

The WHILE pretty-printer (9)

- Messages d'erreur de syntaxe
**pas plus d'un message d'erreur par
commande élémentaire**
- Si possible,
**plusieurs erreurs dans commandes
élémentaires différentes
⇒ plusieurs messages**

The WHILE pretty-printer (exemple)

```
function p : read X % Y
              := nil ;
while X do Y   := (cons
                  (hd X
                  )Y) ; X:=(tl
                  X) od % write Y
```



```
function p:
read X
%
Y := nil ;
while X do
    Y := (cons (hd X) Y) ;
    X := (tl X)
od
%
write Y
```

The WHILE pretty-printer (bonus)

- Fixer une largeur de page
 - valeur par défaut (ex. 80)
 - valeur paramétrable : **-page 90**
- Casser les lignes qui débordent
 - caractère de suite (ex. ...)

blabla...\$

^...blabla

modifier l'analyseur syntaxique



- alignement vertical des sous-expressions, **factorisation**



(cons a

(cons b
c))

introduction projet

(cons a

(cons b
c))

(cons a

**b
c)**

Propriétés fonctionelles

- **$\text{whpp}(p) = \text{whpp}^2(p)$**
 - facile
- $\text{whpp}(pp) = pp$
 - **marche aussi si $\text{whpp} = \text{id}$!**
- $\text{whpp}^{-1}(p) = \text{whpp}^{-1}(\text{whpp}(p))$
et $\text{whpp}(p) = \text{whpp}(\text{whpp}^{-1}(p))$
 - programmer le whpp^{-1} ...
 - ...éliminer toute la mise en page
- **Couverture des consignes**

Propriétés non-fonctionnelles

- $\text{Coût}_{\text{whpp}}(p) = O(|p|)$