

Codeworker et Cnorm - un front-end C extensible

Auroux Lionel

9 juillet 2011

1 Codeworker

2 Le Cnorm

3 Questions

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

- Présentation
- Fonctionnalité de parsing
- Gestion d'entrée sortie
- Grammaire extensible

Le Cnorm

Questions

1 Codeworker

- **Présentation**
- Fonctionnalité de parsing
- Gestion d'entrée sortie
- Grammaire extensible

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

Outil ou langage ?

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

- ❶ Langage de script (function, if, while, ...);
- ❷ Langage de parsing ($R ::= A \mid [B \mid C]^*$);
- ❸ Langage de template (%>ceci est un @variable@.<%);
- ❹ Interface CLI riche (-parseBNF,-generate,-translate,-expand,-script);
- ❺ Mode interprète interactif.

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

Outil ou langage ?

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

- 1 Langage de script (function, if, while, ...);
- 2 Langage de parsing ($R ::= A \mid [B \mid C]^*$);
- 3 Langage de template (%>ceci est un @variable@.<%);
- 4 Interface CLI riche (-parseBNF,-generate,-translate,-expand,-script);
- 5 Mode interprète interactif.

- 1 Langage de script (function, if, while, ...);
- 2 Langage de parsing ($R ::= A \mid [B \mid C]^*$);
- 3 Langage de template (`%>ceci est un @variable@.<%>`);
- 4 Interface CLI riche
(`-parseBNF,-generate,-translate,-expand,-script`);
- 5 Mode interprète interactif.

Outil ou langage ?

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

- ① Langage de script (function, if, while, ...);
- ② Langage de parsing ($R ::= A \mid [B \mid C]^*$);
- ③ Langage de template (%>ceci est un @variable@.<%);
- ④ Interface CLI riche (-parseBNF,-generate,-translate,-expand,-script);
- ⑤ Mode interprète interactif.

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

- ① Langage de script (function, if, while, ...);
- ② Langage de parsing ($R ::= A \mid [B \mid C]^*$);
- ③ Langage de template (`%>ceci est un @variable@.<%>`);
- ④ Interface CLI riche
(`-parseBNF,-generate,-translate,-expand,-script`);
- ⑤ Mode interprète interactif.

1 Codeworker

- Présentation
- **Fonctionnalité de parsing**
- Gestion d'entrée sortie
- Grammaire extensible

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

Pred LL(k) vs LALR(1) (#check)

```
R ::=
    A:var
    #check(myFunction(var))
    B
    | C
;
```

#ignore vs channel (antlr)

```
#ignore(blanks)
translation_unit ::= ...
;

myHexaLiteral ::= #!ignore ['a'..'f'|'
    A'..'F'|'0'..'9']+
;

#ignore(C++)
myCxx ::= ...
;

#ignore["something else"] ::= ...
;
```

Règles templates

```
R ::= identifieur:id Stmt<id>
;
Stmt<T> ::= ...
;
Stmt<"if"> ::= ...
;
Stmt<"while"> ::= ...
;
```

1 Codeworker

- Présentation
- Fonctionnalité de parsing
- **Gestion d'entrée sortie**
- Grammaire extensible

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Présentation

Fonctionnalité de parsing

Gestion d'entrée sortie

Grammaire extensible

Le Cnorm

Questions

① mode translate

② #implicitCopy/#explicitCopy

③ injection dynamique dans le flux d'entrée via
#parsedFile ou #parsedString

- 1 mode translate
- 2 #implicitCopy/#explicitCopy
- 3 injection dynamique dans le flux d'entrée via
#parsedFile ou #parsedString

- 1 mode translate
- 2 #implicitCopy/#explicitCopy
- 3 injection dynamique dans le flux d'entrée via #parsedFile ou #parsedString

Exemple du code d'un préprocesseur like

```
myMacro ::= ...  
;  
ident ::=  
  [#explicitCopy #readIdentif: id]  
  [#check(findElement(id, this.macro)  
    && !findElement(id, this.rekurs))  
    => insert this.rekurs[id] = true;  
  #parsedString(this.macro[id]) myMacro  
    => removeElement(this.rekurs, id);  
  | => {%>@id@<%}  
  ]  
;
```

1 Codeworker

- Présentation
- Fonctionnalité de parsing
- Gestion d'entrée sortie
- **Grammaire extensible**

Codeworker

- Présentation
- Fonctionnalité de parsing
- Gestion d'entrée sortie
- Grammaire extensible**

Le Cnorm

Questions

Dans un fichier myRootGrammar.cwp

```
R ::=
    A B C
;
A ::= ...
;
B ::= ...
;
C ::= ...
;
```

```
#include "myRootGamma.cwp"

#overload
R ::=
    #super::R
    | D E
;
D ::= '{' A '}'
;
E ::= B C
;
```

1 Codeworker

2 Le Cnorm

3 Questions

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Présentation

Les fonctionnalités

Un exemple

Questions

2 Le Cnorm

- **Présentation**
- Les fonctionnalités
- Un exemple

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Présentation

Les fonctionnalités

Un exemple

Questions

- Front-end C développé en codeworker dans le cadre du module ÉPITECH : KOOC (Kind of objective-C) ;
- Disponible : <http://code.google.com/p/cnorm> .

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Présentation

Les fonctionnalités

Un exemple

Questions

- 2 Le Cnorm
 - Présentation
 - **Les fonctionnalités**
 - Un exemple

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Présentation

Les fonctionnalités

Un exemple

Questions

- 1 Grammaire extensible et dynamique (qualifier, spécifier) ;
- 2 Lib normalisé d'accès à l'Ast ;
- 3 PatchLib : fonctions de création dynamique de sous-arbre ;
- 4 PrettyPrinter.

- 1 Grammaire extensible et dynamique (qualifier, spécifier) ;
- 2 Lib normalisé d'accès à l'Ast ;
- 3 PatchLib : fonctions de création dynamique de sous-arbre ;
- 4 PrettyPrinter.

- 1 Grammaire extensible et dynamique (qualifier, spécifier) ;
- 2 Lib normalisé d'accès à l'Ast ;
- 3 PatchLib : fonctions de création dynamique de sous-arbre ;
- 4 PrettyPrinter.

- ① Grammaire extensible et dynamique (qualifier, spécifier) ;
- ② Lib normalisé d'accès à l'Ast ;
- ③ PatchLib : fonctions de création dynamique de sous-arbre ;
- ④ PrettyPrinter.

- 2 Le Cnorm
 - Présentation
 - Les fonctionnalités
 - Un exemple

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Présentation

Les fonctionnalités

Un exemple

Questions

Parse du C et le re-affiche

```
#include "cnorm.inc.cws"  
#include "cnorm2c/cnorm2c.inc.cws"  
local source;  
local sOutput = _ARGS[0] + ".2.c";  
cnormFile(source, _ARGS[0]);  
saveToFile(sOutput, cnorm2c(source));
```

1 Codeworker

2 Le Cnorm

3 Questions

Codeworker et
Cnorm - un
front-end C
extensible

Auroux Lionel

Codeworker

Le Cnorm

Questions