Master Info - 2018-2019
MIF08 Compilation

Contrôle continu 2 (CC-TP) SUJET A - Durée 20 min

Éléments de correction

1 Manipulations préliminaires

— Sauvegardez vos modifications faites dans le git étudiant, par exemple :

```
git commit -a -m "mes modifs"
```

— Récupérer le sujet de contrôle :

```
git pull
```

— Vous travaillez dans le répertoire `CC-TPA`.

— Ouvrir le Makefile et changer `JohnDoe` en votre prénom suivi de votre nom (sans accent, sans espace, sans caractère spécial, tirets autorisés).

Solution:
attention le nom respect des consignes = points en moins. on attend un tgz, avec votre nom, et l'archive doit contenir uniquement ce qu'on a demandé (make clean avant de rendre)
5 points pour la grammaire et respect des consignes, 5 points pour les tests.

2 Exercice - grammaire avec ANTLR

L'objet de cet exercice est d'écrire un analyseur qui reconnaît les palindromes sur l'alphabet $\{a, b\}^*$ (de taille > 0 : ε n'est pas un palindrome), c'est-à-dire les mots qui sont égaux à leur "mot miroir". Votre analyseur complet devra ignorer les espaces, sauts de ligne, mais rejeter pendant la phase d'analyse lexicale les mots comportant d'autres caractères.

1. Écrire un premier fichier de test `tests/ex0.txt`.
2. Éditer le `.g4` pour coder l'analyseur (lexical/syntaxique). Tester avec :

   ```
   make
   python3 sujetA.py tests/ex0.txt
   ```

   On rappelle qu'un fichier accepté par la grammaire ne cause aucun affichage sur la sortie standard, seules d'éventuelles erreurs lexicales ou syntaxiques sont affichées.
3. Dans le répertoire `tests/` ajouter 5 tests pertinents pour cet analyseur (positifs, négatifs).
4. Pour déposer :

   ```
   make clean
   make tar
   ```

   vous fournit un tgz à déposer sur TOMUSS.

Solution:

```
grammar Palin;

//UNCOMMENT prog: EOF {print("Sujet A!");} ;

//START CUT
prog: pal  EOF;

pal:
     'a' pal 'a'
    |'a'
    | 'a' 'a'
    |'b' pal 'b'
    |'b'
    | 'b' 'b'
;

// END CUT

WS : [ \t\r\n]+ -> skip ; // skip spaces, tabs, newlines
```

Master Info - 2018-2019
MIF08 Compilation

Contrôle continu 2 (CC-TP) SUJET B - Durée 20 min

Éléments de correction

1 Manipulations préliminaires

— Sauvegardez vos modifications faites dans le git étudiant, par exemple :

```
git commit -a -m "mes modifs"
```

— Récupérer le sujet de contrôle :

```
git pull
```

— Vous travaillez dans le répertoire `CC-TPB`.

— Ouvrir le Makefile et changer `JohnDoe` en votre prénom suivi de votre nom (sans accent, sans espace, sans caractère spécial, tirets autorisés).

Solution:
attention le nom respect des consignes = points en moins. on attend un tgz, avec votre nom, et l'archive doit contenir uniquement ce qu'on a demandé (make clean avant de rendre)
5 points pour la grammaire et respect des consignes, 5 points pour les tests.

2 Exercice - grammaire avec ANTLR

L'objet de cet exercice est d'écrire un analyseur qui reconnaît les mots sur l'alphabet $\{a, b\}^*$ (de taille > 0 : ε n'appartient pas au langage), tels que chaque a est immédiatement suivi d'au moins un b.

Votre analyseur complet devra ignorer les espaces, sauts de ligne, tabulations mais rejeter pendant la phase d'analyse lexicale les mots comportant d'autres caractères.

1. Écrire un premier fichier de test `tests/ex0.txt`.
2. Éditer le `.g4` pour coder l'analyseur (lexical/syntaxique). Tester avec :

   ```
   make
   python3 sujetB.py tests/ex0.txt
   ```

   On rappelle qu'un fichier accepté par la grammaire ne cause aucun affichage sur la sortie standard, seules d'éventuelles erreurs lexicales ou syntaxiques sont affichées.
3. Dans le répertoire `tests/` ajouter 5 tests pertinents pour cet analyseur (positifs, négatifs).
4. Pour déposer :

   ```
   make clean
   make tar
   ```

vous fournit un tgz à déposer sur TOMUSS.

Solution:

```
grammar Tutu;

//UNCOMMENT prog: EOF {print("Sujet B!");} ;

//START CUT

prog: tutu  EOF;

tutu:
      'a' 'b' tutu
    | 'b' tutu
    | 'a' 'b'
    | 'b'
;

// END CUT

WS : [ \t\r\n]+ -> skip ; // skip spaces, tabs, newlines
```