Correction du QCM THL — Théorie des Langages

EPITA – Promo 2008

Juillet 2006

Il y a toujours exactement une seule réponse valable. Lorsque plusieurs réponses sont possibles, prendre la plus restrictive.

Le langage a^n est

- fini
- rationnel
- non reconnaissable par automate fini
- vide

Le langage $a^n b^n$ pour $n < 42^{51} - 1$ est

- infini
- rationnel
- non reconnaissable par automate fini
- vide

Le langage $(ab)^n$ est

- fini
- rationnel
- non reconnaissable par automate fini
- vide

Le langage $a^n b^m$, où n, m parcourent les entiers naturel, est

- fini
- rationnel
- non reconnaissable par automate fini
- vide

L'expression rationnelle étendue $[a - zA - Z][a - zA - Z0 - 9_]*$ n'engendre pas:

- `__STDC__`
- × `main`
- × `eval_expr`
- × `exit_42`

Un automate fini déterministe. . .

- × n'est pas nondéterministe
- × n'est pas à transitions spontanées
- n'a pas plusieurs états initiaux
- × n'a pas plusieurs états finaux

Le langage $a^n b^n$ est

- × fini
- × rationnel
- non reconnaissable par automate fini
- × vide

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned} P &\rightarrow P \text{ inst } ' ; ' \\ P &\rightarrow \text{inst}' ; ' \end{aligned}$$

- Rationnelle (Type 3)
- × Hors contexte (Type 2)
- × Sensible au contexte (Type 1)
- × Monotone (Type 1)

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned} A &\rightarrow aABC \\ A &\rightarrow abC \\ CB &\rightarrow BC \\ bB &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}$$

- × Rationnelle (Type 3)
- × Hors contexte (Type 2)
- × Sensible au contexte (Type 1)
- Monotone (Type 1)

Quelle propriété de cette grammaire est vraie ?

$$\begin{aligned} S &\rightarrow aSc \\ S &\rightarrow c \end{aligned}$$

- × Linéaire à gauche
- × Linéaire à droite
- Hors contexte
- × Ambiguë

Quelle propriété de cette grammaire est vraie ?

$$\begin{aligned} S &\rightarrow SpS \\ S &\rightarrow n \end{aligned}$$

- × Linéaire à gauche
- × Linéaire à droite
- × Rationnelle
- Ambiguë

Un langage quelconque est

- toujours inclus dans un langage rationnel
- × toujours inclus dans un langage hors-contexte
- × toujours inclus dans un langage sensible au contexte
- × peut ne pas être inclus dans un langage défini par une grammaire

Soit L_r est un langage rationnel. Si $L \subset L_r$, alors

- × L est rationnel
- × L est hors-contexte
- × L est sensible au contexte
- L peut ne pas être définissable par une grammaire

LL(k) signifie

- × lecture en deux passes de gauche à droite, avec k symboles de regard avant
- × lecture en deux passes de gauche à droite, avec une pile limitée à k symboles
- lecture en une passe de gauche à droite, avec k symboles de regard avant
- × lecture en une passe de gauche à droite, avec une pile limitée à k symboles

Si une grammaire est LL(1), alors

- × elle n'est pas rationnelle
- × elle est rationnelle
- elle n'est pas ambiguë
- × elle est ambiguë

Si un parseur LALR(1) a des conflits, alors sa grammaire

- × est ambiguë
- × n'est pas LR(1)
- n'est pas LR(0)
- × n'est pas déterministe

Si une grammaire hors contexte est non ambiguë

- × elle est LL(1)
- × elle est LL(k)
- elle n'est pas nécessairement LL
- × elle produit nécessairement des conflits dans un parseur LL

Quelle forme de l'arithmétique est LL(1)?

×

$$S \rightarrow S + S \mid S * S \mid n$$

×

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow n \end{aligned}$$

→

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid T \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid F \\ F &\rightarrow n \end{aligned}$$

- × LL(1) ne permet pas de traiter l'arithmétique

Lex/Flex sont des

- générateurs de scanners
- × générateurs de parsers
- × parseurs
- × scanners

Yacc repose sur l'algorithme

- × LL(k)
- × YACC(1)
- × LR(k)
- LALR(1)