Rattrapage 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

- **a.** fini
- **b.** rationnel
- **c.** non reconnaissable par automate fini
- **d.** vide

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

- **a.** infini
- **b.** rationnel
- **c.** non reconnaissable par automate fini
- **d.** vide

Le langage $(ab)^n$ est

- **a.** fini
- **b.** rationnel
- **c.** non reconnaissable par automate fini
- **d.** vide

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

- **a.** fini
- **b.** rationnel
- **c.** non reconnaissable par automate fini
- **d.** vide

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

- **a.** `__STDC__`
- **b.** *main*
- **c.** *eval_expr*
- **d.** *exit*_42

Un automate fini déterministe...

- **a.** n'est pas nondéterministe
- **b.** n'est pas à transitions spontanées
- **c.** n'a pas plusieurs états initiaux
- **d.** n'a pas plusieurs états finaux

Le langage $a^n b^n$ est

- **a.** fini
- **b.** rationnel
- **c.** non reconnaissable par automate fini
- **d.** vide

Quelle est la classe de la grammaire suivante ?

$$\begin{array}{lcl} P & \rightarrow & P\ inst\ ';' \\ P & \rightarrow & inst';' \end{array}$$

- **a.** Rationnelle (Type 3)
- **b.** Hors contexte (Type 2)
- **c.** Sensible au contexte (Type 1)
- **d.** Monotone (Type 1)

Quelle est la classe de la grammaire suivante ?

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

- **a.** Rationnelle (Type 3)
- **b.** Hors contexte (Type 2)
- **c.** Sensible au contexte (Type 1)
- **d.** Monotone (Type 1)

Quelle propriété de cette grammaire est vraie ?

$$\begin{array}{rcl} S & \rightarrow & aSc \\ S & \rightarrow & c \end{array}$$

- **a.** Linéaire à gauche
- **b.** Linéaire à droite
- **c.** Hors contexte
- **d.** Ambiguë

Quelle propriété de cette grammaire est vraie ?

$$\begin{array}{lcl} S & \rightarrow & SpS \\ S & \rightarrow & n \end{array}$$

a. Linéaire à gauche

b. Linéaire à droite

c. Rationnelle

d. Ambiguë

Un langage quelconque est

a. toujours inclus dans un langage rationnel

b. toujours inclus dans un langage hors-contexte

c. toujours inclus dans un langage sensible au contexte

d. 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

a. L est rationnel

b. L est hors-contexte

c. L est sensible au contexte

d. L peut ne pas être définissable par une grammaire

LL(k) signifie

a. lecture en deux passes de gauche à droite, avec k symboles de regard avant

b. lecture en deux passes de gauche à droite, avec une pile limitée à k symboles

c. lecture en une passe de gauche à droite, avec k symboles de regard avant

d. lecture en une passe de gauche à droite, avec une pile limitée à k symboles

Si une grammaire est LL(1), alors

a. elle n'est pas rationnelle

b. elle est rationnelle

c. elle n'est pas ambiguë

d. elle est ambiguë

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

a. est ambiguë

b. n'est pas LR(1)

c. n'est pas LR(0)

d. n'est pas déterministe

Si une grammaire hors contexte est non ambiguë

a. elle est LL(1)

b. elle est LL(k)

c. elle n'est pas nécessairement LL

d. elle produit nécessairement des conflits dans un parseur LL

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

a.

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

b.

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

c.

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

d. LL(1) ne permet pas de traiter l'arithmétique

Lex/Flex sont des

a. générateurs de scanners

b. générateurs de parsers

c. parseurs

d. scanners

Yacc repose sur l'algorithme

a. LL(k)

b. YACC(1)

c. LR(k)

d. LALR(1)