

Correction du QCM THL — Théorie des Langages

EPITA – Promo 2009

Juillet 2007

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

Le langage $\{a^n \mid n \in \mathbb{N}\}$ est

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

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

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

Le langage $\{(ab)^n \mid n \in \mathbb{N}\}$ est

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

Le langage $\{a^n b^m \mid n, m \in \mathbb{N}\}$, est

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

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

- ☒ `_exit`
- ☐ `exit`
- ☐ `exit_`
- ☐ `e`

Un automate fini déterministe. . .

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

Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ est

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

Quelle est la classe la plus stricte de la grammaire suivante ?

S	\rightarrow	$N \mid L$
N	\rightarrow	ceriel dick noam
L	\rightarrow	$E \mid N C L$
$C E$	\rightarrow	and E
and E	\rightarrow	and N
C	\rightarrow	' '

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

Quelle est la classe la plus stricte de la grammaire suivante ?

S	\rightarrow	inst ' ' S
S	\rightarrow	inst ' ' ' '

- 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
- × Ambigüe

Quelle propriété de cette grammaire est vraie ?

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

- × Linéaire à gauche
- × Linéaire à droite
- × Rationnelle
- Ambigüe

Un langage quelconque. . .

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

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

Si une grammaire est LL(1), alors

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

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

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

Si une grammaire hors contexte est non ambiguë

- × elle est LL(1)
- × elle est LL(k)
- elle n'est pas nécessairement LL
- × elle est LR(k)

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

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

$$E \rightarrow E + E \mid E * E \mid n$$

×

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

→

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

Lex/Flex sont des

- générateurs d'analyseurs lexicaux
- × générateurs d'analyseurs syntaxiques
- × analyseurs lexicaux
- × analyseurs syntaxiques

Yacc repose sur l'algorithme

- × GLR
- × ANTLR(k)
- × LR(k)
- LALR(1)