

+323/1/60+



EPITA_ING1_2019_S1 THL — Sans document ni machine


Nom et prénom, lisibles :

.....
.....
.....
.....

Identifiant (de haut en bas) :

.....
.....
.....
.....

Q.1 Ne rien écrire sur les bords de la feuille, ni dans les éventuels cadres grisés «  ». Noircir les cases plutôt que cocher. Renseigner les champs d'identité. Les questions marquées par «  » peuvent avoir plusieurs réponses justes. Toutes les autres n'en ont qu'une; si plusieurs réponses sont valides, sélectionner la plus restrictive (par exemple s'il est demandé si 0 est nul, non nul, positif, ou négatif, cocher nul). Il n'est pas possible de corriger une erreur, mais vous pouvez utiliser un crayon. Les réponses justes créditent; les incorrectes pénalisent; les blanches et réponses multiples valent 0.

 J'ai lu les instructions et mon sujet est complet: les 8 entêtes sont +323/1/xx+...+323/8/xx+.

1 Incontournables

Chacune des questions suivantes est notée 0 pour la réponse correcte, -5 dans tous les autres cas.

Q.2 Si $\{a^n b^n \mid n \in \mathbb{N}\} \subseteq L$, alors L n'est pas rationnel.

☐ vrai

Q.3 Combien existe-t-il de sous-ensembles de $\{1, 2, \dots, n\}$?

☐ $n!$

☐ $\frac{n(n+1)}{2}$

☒ 2^n

☐ $\frac{n(n-1)}{2}$

☐ n^2

Q.4 Si une grammaire n'est pas LR(1), alors elle est ambiguë.

☐ vrai

☒ faux

2 Théorie des langages rationnels

Q.5 Que vaut $\text{Fact}(L)$ (l'ensemble des facteurs) :

☐ $\text{Suff}(\text{Suff}(L))$

☐ $\text{Suff}(\overline{\text{Pref}(L)})$

☒ $\text{Suff}(\text{Pref}(L))$

☐ $\text{Pref}(\overline{\text{Pref}(L)})$

☐ $\text{Pref}(\text{Pref}(L))$

Q.6 Que vaut $\text{Suff}(\{a\}\{b\}^*)$

☐ $\{a\}\{b\}^*\{a\}$

☐ $\{a, b\}^*\{b\}\{a, b\}^*$

☐ $\{\varepsilon\} \cup \{a\}\{a\}\{a\}^*$

☐ $\{b\}\{a\}^* \cup \{b\}^*$

☒ $\{a\}\{b\}^* \cup \{b\}^*$

Q.7 Soit Σ un alphabet. Pour tout $a \in \Sigma$, $L_1, L_2 \subseteq \Sigma^*$, on a $L_1^* = L_2^* \implies L_1 = L_2$.

☒ faux

☐ vrai

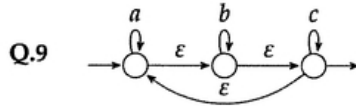


Q.8 Ces deux expressions rationnelles :

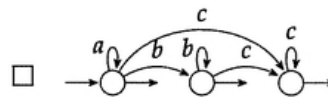
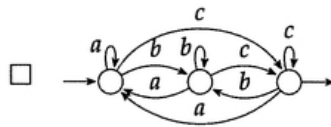
$$(a^* + b)^* + c((ab)^*(bc))^*(ab)^* \quad c(ab + bc)^* + (a + b)^*$$

2/2

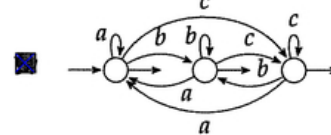
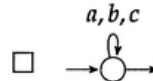
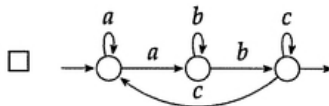
- ☐ dénotent des langages différents ☐ ne sont pas équivalentes
☒ sont équivalentes ☐ sont identiques



Quel est le résultat d'une élimination arrière des transitions spontanées ?

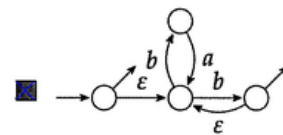
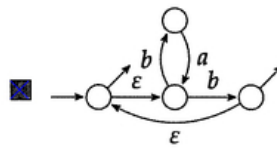
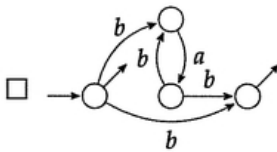


2/2



Q.10 🎵 Parmi les 3 automates suivants, lesquels sont équivalents ?

2/2



☐ Aucune de ces réponses n'est correcte.

Q.11 Quelle séquence d'algorithmes teste l'appartenance d'un mot au langage d'une expression rationnelle ?

2/2

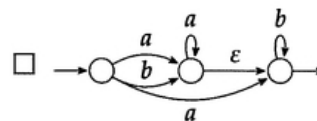
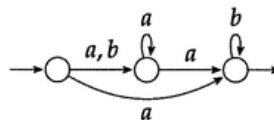
- ☐ Thompson, déterminisation, évaluation.
☒ Thompson, élimination des transitions spontanées, déterminisation, minimisation, évaluation.
☐ Thompson, déterminisation, élimination des transitions spontanées, évaluation.
☐ Thompson, déterminisation, Brzowski-McCluskey.

Q.12 Combien d'états au moins a un automate déterministe émondé qui accepte les mots sur $\Sigma = \{a, b, c, d\}$ dont la n -ième lettre avant la fin est un a (i.e., $(a + b + c + d)^* a (a + b + c + d)^{n-1}$) :

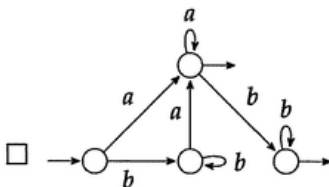
2/2

- ☐ Il n'existe pas. ☐ $\frac{n(n+1)(n+2)(n+3)}{4}$ ☐ 4^n ☒ 2^n

Q.13 Déterminiser cet automate.

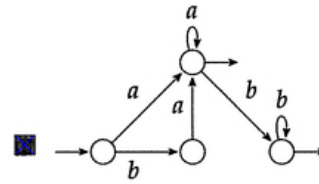
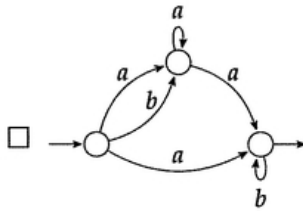


2/2

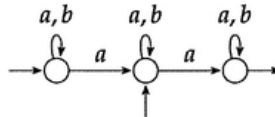




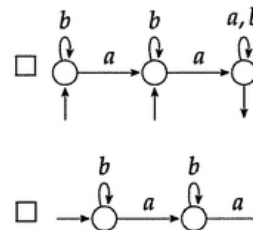
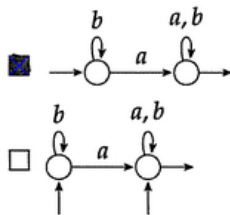
2/2



Q.14 Déterminiser cet automate :



2/2



Q.15 Qu'un langage vérifie le lemme de pompage

0/2

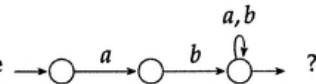
- ☐ est suffisant pour qu'il soit rationnel ☒ est nécessaire s'il est rationnel
☐ Aucune de ces réponses n'est correcte.

Q.16 Considérons \mathcal{P} l'ensemble des *palindromes* (mot u égal à son transposé/image miroir u^R) de longueur paire sur Σ , i.e., $\mathcal{P} = \{v \cdot v^R \mid v \in \Sigma^*\}$.

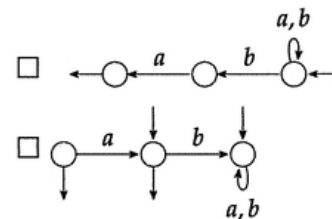
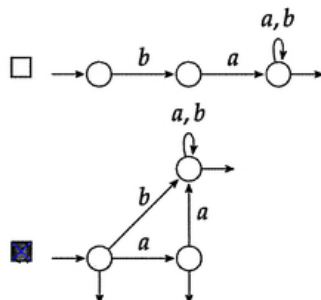
2/2

- ☒ \mathcal{P} ne vérifie pas le lemme de pompage ☐ Il existe un NFA qui reconnaisse \mathcal{P}
☐ Il existe un DFA qui reconnaisse \mathcal{P} ☐ Il existe un ε -NFA qui reconnaisse \mathcal{P}

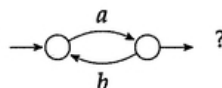
Q.17 Sur $\{a, b\}$, quel automate reconnaît le complémentaire du langage de



2/2

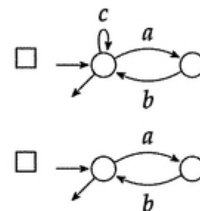
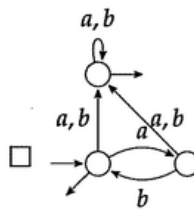
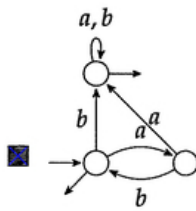


Q.18 Sur $\{a, b\}$, quel est le complémentaire de





2/2



3 Grammaires et Machines abstraites

Q.19 Un transducteur est

2/2

- ☐ un automate infini
☒ une machine ayant une entrée et une sortie
☐ un élément de transitor
☐ un automate fini avec des transductions spontanées

Q.20 Quelle est la classe du langage $\{a^n b^n c^n \mid n \in \mathbb{N}\}$?

2/2

- ☐ Fini (Type 4)
☐ Rationnel (Type 3)
☒ Sensible au contexte (Type 1)
☐ Général (Type 0)
☐ Hors contexte (Type 2)

Q.21 Quelle est la classe du langage $\{a^n b^n \mid n \in \mathbb{N}\}$?

2/2

- ☐ Rationnel (Type 3)
☐ Sensible au contexte (Type 1)
☐ Général (Type 0)
☐ Fini (Type 4)
☒ Hors contexte (Type 2)

Q.22 Quelle propriété cette grammaire vérifie? $S \rightarrow SpS \mid n$

2/2

- ☒ Ambigüe
☐ Linéaire à gauche
☐ Rationnelle
☐ Linéaire à droite

Q.23 Il existe un formalisme qui permette une description finie de tout langage.

2/2

- ☒ Non.
☐ Oui.
☐ Ça dépend de l'alphabet.
☐ Ça dépend du formalisme.

4 Analyseurs

Q.24 Les "start conditions" de Lex/Flex (%s et %x) permettent

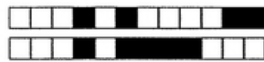
2/2

- ☐ la conversion des chaînes de chiffres en la valeur qu'elles représentent
☐ le choix du parseur à utiliser
☒ de supporter différents contextes lexicaux
☐ de déterminer quand l'analyse lexicale doit commencer

Q.25 Un parser sert à

2/2

- ☐ construire un analyseur syntaxique
☐ s'assurer de la correction du typage
☒ faire de l'analyse syntaxique
☐ éliminer les récursions terminales



+323/5/56+

2/2

☐ segmenter un flux de caractères en un flux de tokens

Q.26 Comment désambigüiser pour Yacc/Bison le morceau d'arithmétique suivant :

exp: exp '+' exp | exp '-' exp | NUM;

2/2

☐ %left '-' %left '+' ☐ %left '+' %left '-' %nonassoc NUM
☐ %left '+' %left '-' ☒ %left '+' '-'

Q.27 Avec la grammaire suivante, quel état atteint l'automate LR(1) après une transition sur E puis sur '?'?

$S \rightarrow E \$$
 $E \rightarrow E ? E : E \mid E + E \mid 0$

2/2

<input type="checkbox"/>	$\begin{array}{l} S \rightarrow E \bullet \$ \quad [\$] \\ E \rightarrow E \bullet ? E : E \quad [\$?+ :] \\ E \rightarrow E \bullet + E \quad [\$?+ :] \end{array}$	<input type="checkbox"/>	$\begin{array}{l} S \rightarrow E \bullet \$ \quad [\$] \\ E \rightarrow E \bullet ? E : E \quad [\$?+ :] \\ E \rightarrow E \bullet + E \quad [\$?+ :] \end{array}$	<input checked="" type="checkbox"/>	$\begin{array}{l} E \rightarrow E ? \bullet E : E \quad [\$?+ :] \\ E \rightarrow \bullet E ? E : E \quad [?+ :] \\ E \rightarrow \bullet E + E \quad [?+ :] \\ E \rightarrow \bullet 0 \quad [?+ :] \end{array}$
<input type="checkbox"/>	$\begin{array}{l} E \rightarrow E ? \bullet E : E \quad [\$?+ :] \\ E \rightarrow \bullet E ? E : E \quad [\$?+ :] \\ E \rightarrow \bullet E + E \quad [\$?+ :] \\ E \rightarrow \bullet 0 \quad [\$?+ :] \end{array}$	<input type="checkbox"/>	$\begin{array}{l} E \rightarrow E ? \bullet E : E \quad [\$?+ :] \\ S \rightarrow \bullet E \$ \quad [\$] \\ E \rightarrow \bullet E ? E : E \quad [\$?+ :] \\ E \rightarrow \bullet E + E \quad [\$?+ :] \\ E \rightarrow \bullet 0 \quad [\$?+ :] \end{array}$		

5 Logique Propositionnelle

Soit le langage de la logique propositionnelle, composé de deux symboles \top (vrai) et \perp (faux), de l'opération unaire \neg (non), des opérations binaires \vee (ou) et \wedge (et), et des parenthèses notées $[,]$. Ce langage inclut des mots tels que $\perp \wedge \perp$, $\top \vee \perp$ et $\neg[\top \wedge \top] \vee [\perp \wedge \perp]$.

Q.28 Que dire de la grammaire suivante?

$S \rightarrow S \wedge S \mid S \vee S \mid \neg S \mid [S] \mid \top \mid \perp \quad (G_1)$

2/2

☐ rationnelle ☒ ambiguë ☐ infiniment ambiguë ☐ non ambiguë

Q.29 Dans la grammaire suivante, quelles sont les priorités/associativités des opérateurs?

$S \rightarrow S \vee T \mid T \quad T \rightarrow T \wedge F \mid F \quad F \rightarrow \neg F \mid [S] \mid \top \mid \perp \quad (G_2)$

2/2

☐ \wedge et \vee associatives à gauche, priorités croissantes : $\neg < \wedge < \vee$
☒ \wedge et \vee associatives à gauche, priorités croissantes : $\vee < \wedge < \neg$
☐ \wedge et \vee associatives à droite, priorités croissantes : $\vee < \wedge < \neg$
☐ \wedge et \vee associatives à droite, priorités croissantes : $\neg < \wedge < \vee$

Q.30 Que dire de la grammaire (G_2) ?

2/2

☐ non ambiguë et LL(1) ☐ ambiguë et non LL(1)
☒ non ambiguë et non LL(1) ☐ ambiguë et LL(1)

Q.31 Que dire de la grammaire suivante par rapport à (G_2) ?

$S \rightarrow TS' \quad T \rightarrow FT' \quad F \rightarrow \neg F \mid [S] \mid \top \mid \perp \quad (G_3)$
 $S' \rightarrow \vee TS' \mid \varepsilon \quad T' \rightarrow \wedge FT' \mid \varepsilon$



+323/6/55+

-1/2

- ☒ même langage, priorités et/ou associativités différentes, mais LL(1)
- ☐ même langage, mêmes priorités et associativités, pas LL(1)
- ☐ même langage, priorités et/ou associativités différentes, pas LL(1)
- ☒ même langage, mêmes priorités et associativités, mais LL(1)
- ☐ langage différent

Q.32 Quels sont les symboles annulables dans la grammaire (G_3)?

2/2

- ☐ S, T, F ☐ F ☒ S', T' ☐ S', T', F ☐ S, S', T, T', F

Q.33 Quels sont les FIRST dans la grammaire (G_3)?

0/2

- | | | | | | | | |
|--------------------------|----------------------|--------------------------|--------------------|--------------------------|--------------------|-------------------------------------|--------------------|
| <input type="checkbox"/> | FIRST | <input type="checkbox"/> | FIRST | <input type="checkbox"/> | FIRST | <input checked="" type="checkbox"/> | FIRST |
| S | $\neg[\top \perp]$ | S | T | S | $\neg[\top \perp]$ | S | $\neg[\top \perp]$ |
| S' | $\varepsilon \vee$ | S' | \vee | S' | $\vee \wedge$ | S' | \vee |
| T | $\neg[\top \perp]$ | T | F | T | $\neg[\top \perp]$ | T | $\neg[\top \perp]$ |
| T' | $\varepsilon \wedge$ | T' | \wedge | T' | $\vee \wedge$ | T' | \wedge |
| F | $\neg[\top \perp]$ | F | $\neg[\top \perp]$ | F | $\neg[\top \perp]$ | F | $\neg[\top \perp]$ |

Q.34 Quels sont les FOLLOW dans la grammaire (G_3)?

0/2

- | | | | | | | | | | |
|--------------------------|---------|--------------------------|----------------|--------------------------|----------------|--------------------------|------|-------------------------------------|----------------|
| <input type="checkbox"/> | FOLL | <input type="checkbox"/> | FOLL | <input type="checkbox"/> | FOLL | <input type="checkbox"/> | FOLL | <input checked="" type="checkbox"/> | FOLL |
| S | $]$ | S | $]$ | S | $\varepsilon]$ | S | $]$ | S | $]$ |
| S' | $]$ | S' | $\vee \wedge]$ | S' | $\vee \wedge]$ | S' | $]$ | S' | $]$ |
| T | $\vee]$ | T | $\vee]$ | T | $\vee]$ | T | $]$ | T | $\vee]$ |
| T' | $\vee]$ | T' | $\vee]$ | T' | $\vee]$ | T' | $]$ | T' | $\vee]$ |
| F | $\vee]$ | F | $\wedge \vee]$ | F | $\wedge \vee]$ | F | $]$ | F | $\wedge \vee]$ |

Q.35 Que dire de la grammaire étendue suivante par rapport à (G_2)?

$$S \rightarrow T(\vee T)^* \quad T \rightarrow F(\wedge F)^* \quad F \rightarrow \neg F \mid [S] \mid \top \mid \perp \quad (G_4)$$

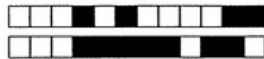
2/2

- ☐ même langage, priorités et/ou associativités différentes, mais LL(1)
- ☐ langage différent
- ☐ même langage, priorités et/ou associativités différentes, pas LL(1)
- ☐ même langage, mêmes priorités et associativités, pas LL(1)
- ☒ même langage, mêmes priorités et associativités, mais LL(1)

Q.36 Quelle routine parse et calcule correctement S pour la grammaire (G_4) de la logique booléenne? La variable la désigne le lookahead courant, et la routine eat (*expect*) vérifie que le lookahead actuel est *expect* puis stocke le suivant dans la .

2/2

- | | | |
|---|---|---|
| <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <pre>bool S() { bool res = T(); while (la == 'v') { eat('v'); res = F(); while (la == '^') { eat('^'); res &= F(); } } return res; }</pre> | <pre>bool S() { bool res = T(); while (la == 'v') { res = T(); eat('v'); } return res; }</pre> | <pre>bool S() { bool res = T(); while (la == 'v') { eat('v'); res = T(); } return res; }</pre> |



+323/7/54+

2/2

☐ `bool S()
{
 bool res = false;
 do
 {
 eat('v');
 res |= T();
 }
 while (la == 'v');
 return res;
}`

☐ `bool S()
{
 bool res = true;
 do
 {
 eat('v');
 res |= T();
 }
 while (la == 'v');
 return res;
}`

Q.37 Quelle est la séquence de décalages/réductions pour un parser Yacc/Bison implémentant la grammaire (G_1) avec des directives précisant correctement priorités et associativités?

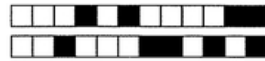
2/2

☐ `┌ T ^ T V T +
s ┌ "T" ^ T V T +
r ┌ S ^ T V T +
s ┌ S "Λ" T V T +
s ┌ S "Λ" "T" V T +
r ┌ S "Λ" S V T +
s ┌ S "Λ" S "V" T +
s ┌ S "Λ" S "V" "T" +
r ┌ S "Λ" S "V" S +
r ┌ S "Λ" S +
r ┌ S +
s ┌ S +
accept`

☒ `┌ T ^ T V T +
s ┌ "T" ^ T V T +
r ┌ S ^ T V T +
s ┌ S "Λ" T V T +
s ┌ S "Λ" "T" V T +
r ┌ S "Λ" S V T +
r ┌ S V T +
s ┌ S "V" T +
s ┌ S "V" "T" +
r ┌ S "V" S +
r ┌ S +
s ┌ S +
accept`

☐ `┌ T ^ T V T +
s ┌ "T" ^ T V T +
r ┌ S ^ T V T +
s ┌ S "Λ" T V T +
s ┌ S "Λ" "T" V T +
r ┌ S "Λ" S V T +
s ┌ S "Λ" S "V" T +
s ┌ S "Λ" S "V" "T" +
r ┌ S "Λ" S "V" S +
r ┌ S "V" S +
r ┌ S +
s ┌ S +
accept`

☐ `┌ T ^ T V T +
s ┌ "T" ^ T V T +
r ┌ S ^ T V T +
s ┌ S "Λ" T V T +
s ┌ S "Λ" "T" V T +
r ┌ S V T +
s ┌ S "V" T +
s ┌ S "V" "T" +
r ┌ S +
s ┌ S +
accept`



+323/8/53+

2/2

\vdash		$T \wedge T \vee T \dashv$
$S \vdash$	"T"	$\wedge T \vee T \dashv$
$r \vdash$	S	$\wedge T \vee T \dashv$
$S \vdash$	S "	$T \vee T \dashv$
$S \vdash$	S "	$\vee T \dashv$
$S \vdash$	S "	$T \dashv$
$S \vdash$	S "	$T \dashv$
$r \vdash$	S "	$S \dashv$
$r \vdash$	S "	\dashv
$r \vdash$	S	\dashv
$S \vdash$	S	\dashv
accept		

Fin de l'épreuve.