Vérifiez votre énoncé: les 5 entêtes doivent être `+1/1/xx+...` `+1/5/xx+`.

EPITA_ING1_2016_S1 THL — Sans document ni machine

Noircir les cases plutôt que cocher. Renseigner les champs d'identité. Les questions marquées du symbole ♣ peuvent avoir plusieurs réponses justes. Toutes les autres questions n'ont qu'une seule réponse juste; si plusieurs réponses sont valides, sélectionner la plus restrictive (par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négatif*, sélectionner *nul*). Il n'est pas possible de corriger une erreur. Les réponses justes créditent; les incorrectes pénalisent; et les blanches et réponses multiples valent 0.

Nom et prénom :

....

....

....

....

Cochez votre identifiant (de haut en bas):

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

1 Incontournables

Chaque erreur ou non réponse aux trois questions suivantes retire 1/6 de la note finale. Avoir tout faux divise donc la note par 2.

Q.1 Si $\{a^n b^n \mid n \in \mathbb{N}\} \subseteq L$, alors L n'est pas rationnel.

☐ faux ☐ vrai

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

☐ $\frac{n(n+1)}{2}$ ☐ n^2 ☐ $n!$ ☐ 2^n ☐ $\frac{n(n-1)}{2}$

Q.3 Si une grammaire n'est pas LR(1), alors elle est ambiguë.

☐ vrai ☐ faux

2 Grammaires et Machines abstraites

Q.4 Un transducteur est

☐ une machine ayant une entrée et une sortie

☐ un automate fini avec des transductions spontanées

☐ un automate infini

☐ un élément de transitor

Q.5 Quelle est la classe de la grammaire suivante ? $P \rightarrow P$ `"stm" ";"` | `"stm" ";"`

☐ Hors contexte (Type 2)

☐ Monotone (Type 1)

☐ Rationnelle (Type 3)

☐ Sensible au contexte (Type 1)

Q.6 Quelle est la classe de la grammaire suivante ?

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

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

Q.7 Il existe un formalisme qui permette une description finie de tout langage.

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

Q.8 Quelle propriété cette grammaire vérifie ? $S \to Sac \mid c$

☐ Linéaire à gauche ☐ Linéaire à droite ☐ Hors contexte ☐ Ambiguë

3 Analyseurs

Q.9 Si une grammaire hors contexte est LL(1), alors elle est...

☐ non ambiguë ☐ rationnelle ☐ non rationnelle ☐ ambiguë

Q.10 Si une grammaire hors contexte est non ambiguë, alors...

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

Q.11 LL(k) signifie
☐ lecture en une passe de gauche à droite, avec une pile limitée à k symboles
☐ 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

Q.12 Si un parseur LALR(1) a des conflits, alors sa grammaire

☐ n'est pas déterministe
☐ n'est pas LR(1)
☐ est ambiguë
☐ n'est pas LR(0)

4 Logique Propositionnelle

Soit le langage de la logique propositionnelle, composé de deux symboles $\top$ (vrai) et $\bot$ (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 $\bot \wedge \bot$, $\top \vee \bot$ et $\neg\neg[\top \wedge \top] \vee [\bot \wedge \bot]$.

Q.13 Que dire de la grammaire suivante ?

$$S \to S \wedge S \mid S \vee S \mid \neg S \mid [S] \mid \top \mid \bot \quad (G_1)$$

☐ infiniment ambiguë ☐ non ambiguë ☐ rationnelle ☐ ambiguë

Q.14 Dans la grammaire suivante, quelles sont les priorités/associativités des opérateurs ?

$$S \to S \vee T \mid T \qquad T \to T \wedge F \mid F \qquad F \to \neg F \mid [S] \mid \top \mid \bot \quad (G_2)$$

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

Q.15 Que dire de la grammaire (G_1) ?

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

Q.16 Que dire de la grammaire suivante par rapport à (G_2) ?

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

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

Q.17 Quels sont les symboles annulables dans la grammaire (G_3) ?

☐ S, S', T, T', F ☐ S, T, F ☐ S', T' ☐ F ☐ S', T', F

Q.18 Quels sont les FIRST dans la grammaire (G_3) ?

☐

	FIRST
S	$\neg[\top\bot$
S'	$\vee\wedge$
T	$\neg[\top\bot$
T'	$\vee\wedge$
F	$\neg[\top\bot$

☐

	FIRST
S	$\neg[\top\bot$
S'	$\varepsilon\vee$
T	$\neg[\top\bot$
T'	$\varepsilon\wedge$
F	$\neg[\top\bot$

☐

	FIRST
S	T
S'	$\vee$
T	F
T'	$\wedge$
F	$\neg[\top\bot$

☐

	FIRST
S	$\neg[\top\bot$
S'	$\vee$
T	$\neg[\top\bot$
T'	$\wedge$
F	$\neg[\top\bot$

Q.19 Quels sont les FOLLOW dans la grammaire (G_3) ?

☐

	FOLL
S	$\varepsilon]$
S'	$\vee\wedge]$
T	$\vee]$
T'	$\vee]$
F	$\wedge\vee]$

☐

	FOLL
S	$]$
S'	$]$
T	$\vee]$
T'	$\vee]$
F	$\wedge\vee]$

☐

	FOLL
S	$]$
S'	$\vee\wedge]$
T	$\vee]$
T'	$\vee]$
F	$\wedge\vee]$

☐

	FOLL
S	$]$
S'	$]$
T	$]$
T'	$]$
F	$]$

☐

	FOLL
S	$]$
S'	$]$
T	$\vee]$
T'	$\vee]$
F	$\vee]$

Q.20 Que dire de la grammaire étendue suivante par rapport à (G_2) ?

$$S \to T(\vee T)* \qquad T \to F(\wedge F)* \qquad F \to \neg F \mid [S] \mid \top \mid \bot \quad (G_4)$$

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

Q.21 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`.

☐

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

☐
```
bool S()
{
  bool res = T();
  while (la == '∨')
  {
    eat('∨');
    res |= F();
    while (la == '∧')
    {
      eat('∧');
      res &= F();
    }
  }
  return res;
}
```

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

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

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

Q.22 Terminer la séquence de décalages/réductions suivante pour un parser Yacc/Bison implémentant la grammaire (G_1) avec des directives précisant correctement priorités et associativités.

☐
```
  ⊢       ⊤ ∧ ⊤ ∨ ⊤ ⊣
s ⊢ "⊤"     ∧ ⊤ ∨ ⊤ ⊣
r ⊢ S       ∧ ⊤ ∨ ⊤ ⊣
s ⊢ S "∧"     ⊤ ∨ ⊤ ⊣
s ⊢ S "∧" "⊤"   ∨ ⊤ ⊣
r ⊢ S "∧" S     ∨ ⊤ ⊣
s ⊢ S "∧" S "∨"   ⊤ ⊣
s ⊢ S "∧" S "∨" "⊤" ⊣
r ⊢ S "∧" S "∨" S   ⊣
r ⊢ S "∧" S         ⊣
r ⊢ S               ⊣
s ⊢ S ⊣
accept
```

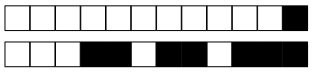
☐
```
  ⊢       ⊤ ∧ ⊤ ∨ ⊤ ⊣
s ⊢ "⊤"     ∧ ⊤ ∨ ⊤ ⊣
r ⊢ S       ∧ ⊤ ∨ ⊤ ⊣
s ⊢ S "∧"     ⊤ ∨ ⊤ ⊣
s ⊢ S "∧" "⊤"   ∨ ⊤ ⊣
r ⊢ S "∧" S     ∨ ⊤ ⊣
r ⊢ S           ∨ ⊤ ⊣
s ⊢ S "∨"         ⊤ ⊣
s ⊢ S "∨" "⊤"       ⊣
r ⊢ S "∨" S         ⊣
r ⊢ S               ⊣
s ⊢ S ⊣
accept
```

```
  ⊢        ⊤ ∧ ⊤ ∨ ⊤ ⊣
s ⊢ "⊤"      ∧ ⊤ ∨ ⊤ ⊣
r ⊢ S        ∧ ⊤ ∨ ⊤ ⊣
s ⊢ S "∧"      ⊤ ∨ ⊤ ⊣
s ⊢ S "∧" "⊤"    ∨ ⊤ ⊣
s ⊢ S "∧" "⊤" "∨"   ⊤ ⊣
s ⊢ S "∧" "⊤" "∨" "⊤" ⊣
r ⊢ S "∧" "⊤" "∨" S   ⊣
r ⊢ S "∧" S           ⊣
r ⊢ S                 ⊣
s ⊢ S ⊣
accept
```

```
  ⊢       ⊤ ∧ ⊤ ∨ ⊤ ⊣
s ⊢ "⊤"     ∧ ⊤ ∨ ⊤ ⊣
r ⊢ S       ∧ ⊤ ∨ ⊤ ⊣
s ⊢ S "∧"     ⊤ ∨ ⊤ ⊣
s ⊢ S "∧" "⊤"   ∨ ⊤ ⊣
r ⊢ S           ∨ ⊤ ⊣
s ⊢ S "∨"         ⊤ ⊣
s ⊢ S "∨" "⊤"       ⊣
r ⊢ S               ⊣
s ⊢ S ⊣
accept
```

```
  ⊢       ⊤ ∧ ⊤ ∨ ⊤ ⊣
s ⊢ "⊤"     ∧ ⊤ ∨ ⊤ ⊣
r ⊢ S       ∧ ⊤ ∨ ⊤ ⊣
s ⊢ S "∧"     ⊤ ∨ ⊤ ⊣
s ⊢ S "∧" "⊤"   ∨ ⊤ ⊣
r ⊢ S "∧" S     ∨ ⊤ ⊣
s ⊢ S "∧" S "∨"   ⊤ ⊣
s ⊢ S "∧" S "∨" "⊤" ⊣
r ⊢ S "∧" S "∨" S   ⊣
r ⊢ S "∨" S         ⊣
r ⊢ S               ⊣
s ⊢ S ⊣
accept
```

Fin de l'épreuve.



+1/6/55+

Pour votre examen, imprimez de préférence les documents compilés à l'aide de auto-multiple-choice.