

**L3 Info, Année 2023-2024. Examen de LF.**  
**Durée 2h30. documents autorisés : poly corrigé TD et cours, notes manuscrites. Aller au WC avec son portable fera l'objet d'un procès verbal de fraude.**

**Conseils.** Chaque question teste un point précis, en général mentionné dans le titre précédent la question. Les réponses sont courtes, genre 3 ou 4 lignes, voire une seule ligne. On a essayé de présenter les exercices par ordre croissant de difficulté. Suivre cet ordre facilite la correction, mais vous n'y êtes pas obligés. C'est long, afin de pouvoir vous évaluer plus précisément.. Pour avoir les points, répondre aux questions. Si vous savez que vous ne savez pas répondre, c'est mieux de ne rien dire plutôt que de recopier des morceaux du cours que vous avez sous la main, car cela énerve juste le correcteur.

**Questions pendant l'exam.** Les seules questions aux surveillants sont celles portant sur une ambiguïté d'énoncé et il n'y en a sans doute pas, donc ne soyez pas trop déçus si on ne répond en général pas à vos questions. On est là pour vous surveiller, pas pour vous aider. On ne souhaite pas aider une personne en particulier plutôt qu'une autre avec qui on serait moins familier.

**Protocole Sortie.** Quand vous commencez une copie remplissez tout de suite l'encart nom et prénom et le reste. Merci de bien vouloir anonymiser : c'est plus confortable et plus impartial, de corriger sans voir qui on corrige. Pour anonymiser, humidifier un peu la partie gommée, replier et appuyer pour que cela colle. Quand le temps est écoulé, numérotez vos copies, posez vos stylos, mettez vos copies les unes dans les autres pour faire un seul tas, rangez vos affaires et prenez-les avec vous, dirigez-vous vers la porte de sortie, au fond (et pas sur la grande table devant), posez votre tas sur le tas de copies, sur une table avec le papier marqué "rendu". Prenez un corrigé dispo juste devant la porte, et sortez. Les sorties avec plus de 5 min de retard sont pénalisées de 0.5 points, il n'y a pas de raison que certain aient plus de temps. Bon courage.

## 1 Language de Grammaire 1pt

Quel est le langage engendré par la grammaire suivante? Justifiez.  $S \rightarrow aSb|TE|\epsilon$ ;  $T \rightarrow aTc|\epsilon$ . Les non-terminaux sont  $S, T, E$  et les terminaux 'a', 'b', 'c'. Non il n'y a pas d'erreurs d'énoncé.

## 2 Automates à pile 3pt

Construire un automate à pile, déterministe, reconnaissant le Langage sur  $\Sigma = \{a, b\}$  défini par

$\{a^n b^m, n \neq m\}$ . Pour chacun des quelques états que vous utiliserez, vous pourrez décrire brièvement le langage des mots qui permet d'y arriver, et ce, afin de vous aider (cette info ne rapporte pas de points, mais si elle n'y est pas et que l'automate n'est pas lisible je m'autorise à ne pas corriger). Vous préciserez également le mode de reconnaissance.

## 3 Ambiguïté 1,25pt.

Soit la grammaire suivante :  $S \rightarrow A|B$ ;  $A \rightarrow Aa|\epsilon$ ;  $B \rightarrow bBc|\epsilon$ .

1. Montrer que cette grammaire est ambiguë
2. Modifier la légèrement, pour obtenir une grammaire non ambiguë qui génère le même langage.

## 4 Mini vrai langage 3pt

On considère un langage utilisé pour décrire des structures de données. Voici un exemple de programme :

```
{ "équipe" : "les super-héros",
  "ville" : "Gif sur Yvette",
  "formé" : 2024,
  "Base secrète" : "Université Paris-Saclay",
  "actif" : true,
  "membres" : [
    { "nom" : "Homme Molécule",
      "âge" : 25,
      "Identité" : "Thomas Soullard",
      "pouvoirs" : [
        "Résistance aux radiations",
        "Devenir minuscule",
        "Souffle brulant",
        "Barbe rendant invisible",
      ] },
    { "nom" : "Madame Uppercut",
      "âge" : 39,
      "Identité" : "Sandrine Delahaye",
      "pouvoirs" : [
        "boxe anglaise",
        "Résistance aux dommages",
        "Réflexes surhumains"
      ] },
    { "nom" : "Flamme éternelle",
      "âge" : 1000000,
      "Identité" : "Fred Gruau",
      "pouvoirs" : [
        "Immortalité",
        "Immunité à la chaleur",
        "Téléportation",
        "Voyage interdimensionnel"
      ] } ] }
```

1. Récrivez sur une seule ligne, en laissant de la place dessous, le début de ce programme jusqu'à 2024 inclus avec la virgule qui suit 2024. Séparer les tokens en intercalant des petits traits verticaux. Quel est le nombre de tokens ?

2. Pour ce début, pour les tokens qui ont une valeur, indiquez en dessous la classe de ces tokens, puis, encore en dessous, leur valeur.
3. A présent on considère tout le programme. Donner la liste de toutes les classes de token (terminaux de la grammaire), combien en trouvez vous ?
4. Ecrivez une grammaire qui permet de générer la liste des classes de token correspondant au langage utilisé pour écrire ce programme. Vous utiliserez les 7 non-terminaux suivant, Prog, Attribut, Valeur, Liste, ListeSet, ListeString, Set.

Attention : le nom des non-terminaux doit correspondre à leur fonction : Set correspond à un contenu entre accolades. Les non-terminaux Liste, ListeSet, ListeString correspondent à un contenu entre crochets, ou on précise aussi ce qu'il y a entre les crochets. Attribut associe un nom à une valeur. Prog correspond à tout le programme.

## 5 Calcul des premier 1,5 pt

Soit la grammaire  $S \rightarrow A|B|C$ ;  $A \rightarrow Bb|a$ ;  $B \rightarrow Cc|b$ ;  $C \rightarrow Aa|c$ . Calculer les premiers de chaque non-terminal. on écrira d'abord les équations et on résoudra ces équations de manière itérative.

## 6 Analyse ascendante 4,5 pt

Soit la grammaire dont les symboles terminaux sont les deux parenthèses :  $\{ (, ) \}$ , et les règles :  $S \rightarrow (S)S|\epsilon$

1. Décrire de façon informelle quel est le langage généré par cette grammaire.
2. Cette grammaire est-elle LR(0), SLR(1) ? Justifiez vos réponses soigneusement : S'il y a des conflits indiquer tous les états où il y en a, indiquer la nature de chaque conflit, puis essayez de les résoudre. Si vous avez pu résoudre ces conflits expliquer bien pourquoi, en dessinant une ligne de la table SLR(1) correspondant à l'un des états avec conflits, afin de qui montrer que le conflit est effectivement résolu. Si vous calculez des premiers ou des suivants, il faut montrer comment vous calculez en écrivant les équations.
3. Si vous avez réussi à construire un automate à pile déterministe, ALORS utiliser le pour analyser la chaîne de caractères  $((()()))$ . N'allez pas jusqu'au bout, c'est trop long. Allez simplement jusqu'à l'étape où on fait pour la première fois, une réduction utilisant la règle récursive :  $S \rightarrow (S)S$ . Il faut inclure cette étape.

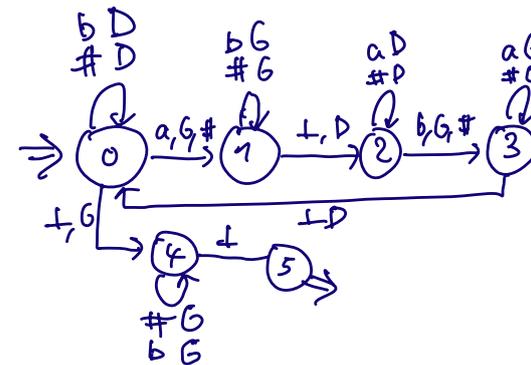
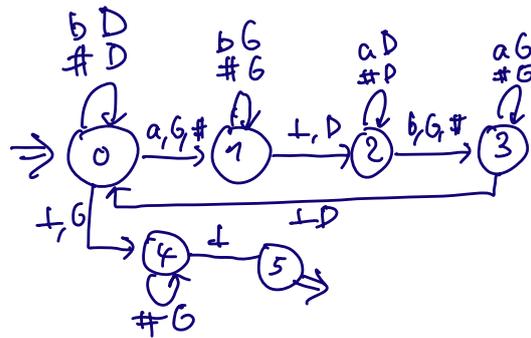
## 7 Décidabilité 1,25pt

Un professeur souhaite automatiser l'attribution de notes à un TP noté. Son sujet se limite à écrire des programmes qui implémentent toujours des fonctions entières, comme la fonction factorielle. Il fournit un environnement qui peut ensuite automatiquement comparer la sortie calculée par rapport à la sortie attendue, et ce pour une dizaine de valeurs fixées. Si cela correspond, les points sont donnés, et sinon, non. On

peut alors définir l'ensemble des programmes que cette procédure accepte. Cet ensemble constitue un langage tel que défini en cours de LF. Ce langage est-il semi-décidable oui/non ? Est-il Décidable oui/non ? Justifiez en une ou deux phrases.

## 8 Machine de Turing 3,5pt.

1. Construire un machine de Turing qui réponds oui, sur le mot vide, et qui réponds rien (i.e. qui boucle) sur tout autre mots. L'alphabet contient une seule lettre : la lettre 'a'.
2. Considérons les deux machines de Turing suivantes, sur l'alphabet  $\{0, 1, \#\}$ . Notez que elles sont presque identiques. Ecrire quel est le langage reconnu pour chacune d'entre elles. On donnera une description en compréhension, pour chacun des deux langages. On pourra commencer par expliquer le rôle de chacun des six états, afin d'arriver à comprendre. Le corrigé utilise seulement une dizaine de mots pour ces explications. Pour avoir les points il est nécessaire de prendre du recul. Il faut dire ce qui se passe globalement, et non pas une description mot à mot de micro-mouvements de tête de lecture.



## 9 Pompe Algébrique 1,5 pt

Un langage algébrique  $L$  doit vérifier le lemme de la pompe algébrique : il existe un entier  $N$ , tel que dès que un mot  $m$  de  $L$  a une longueur plus grande que  $N$ , il devient "pompable". Un mot  $m$  est pompable, si il existe une décomposition  $u, x, v, y, w$  avec  $xy$  non vide, telle que  $m = uxv y w$  et pour tout  $k$ , le pompé par  $k$  qui est  $u x^k v y^k w \in L$ . De plus, il y a une condition sur la longueur de  $xvy$  dont on n'a pas besoin ici. Notez que si un mot est pompable, son pompé par  $k$  devient

arbitrairement grand en choisissant  $k$  de plus en plus grand.

1. Un langage fini  $L$  est algébrique, démontrez le (ici, le lemme de la pompe est inutile).
2. Un langage fini doit être pompable, mais un mot pompé devient arbitrairement grand... Réolvez cette contradiction apparente en démontrant que tout langage fini vérifie le lemme de la pompe algébrique. La preuve est toute simple et tient en une seule phrase. Il faut juste bien choisir l'entier  $N$ .