

# Rattrapage de théorie des langages

26 juin 2015

La clarté et la précision de la rédaction seront prises en compte dans l'évaluation.

Le barème est indicatif.

Aucun document autorisé.

Machines (ordinateurs et calculatrices) interdites.

Durée : 1h 30min

Considérons un langage constitué de 3 symboles permettant d'évaluer une suite de tests unitaires appliqués à un programme informatique vu comme une suite de modules :  $A=\{\mathbf{t}, \mathbf{p}, \mathbf{f}\}$ .  $\mathbf{t}$  correspond à un module valide,  $\mathbf{p}$  un module partiellement valide et  $\mathbf{f}$  un module non valide.

Les mots de ce langage sont uniquement formés de ces symboles et correspondent donc à ce qu'on appelle une suite de tests unitaires, évaluant chacun des modules du programme de manière indépendante.

Le mot vide appartient à ce langage.

Nous allons définir plusieurs grammaires afin de déterminer certaines mesures de tests sur des programmes informatiques.

## Tests réguliers

Soit le langage des suites de tests unitaires dites décroissantes, c'est-à-dire dont les tests valides sont toujours avant les tests partiellement valides, eux même toujours avant les tests non valides. Par exemple  $\mathbf{ttt}$ ,  $\mathbf{t\!t\!p\!f}$  et  $\mathbf{p\!p\!f}$  sont des suites de tests unitaires décroissantes. La suite ne contenant aucun test (mot vide) est considérée comme décroissante.

1. Énumérer tous les mots de taille au plus 3 de ce langage.
2. Construire un automate fini déterministe qui reconnaît les suites de tests unitaires décroissantes.
3. Donner la mise en équation de cet automate et résoudre ce système.
4. Effectuer la méthode des quotients gauches sur le langage obtenu et conclure.

## Tests algébriques

Soit le langage des suites de tests unitaires décroissantes ayant autant de tests valides que de tests non valides.

1. Énumérer tous les mots de taille au plus 3 de ce langage.
2. Écrire une grammaire sous forme normale de Chomsky qui engendre ce langage.
3. Appliquer l'algorithme CKY pour vérifier que la suite `ttpff` appartient à votre langage.
4. Construire un automate à pile déterministe qui reconnaît ce langage.

## Tests de Turing

Donner une machine de Turing qui compte le nombre de doubles tests valides (occurrence du motif `tt`) dans une suite de tests unitaires.

Par exemple la suite `fttfptpftt` contient 2 doubles tests valides et la suite `tttt` en contient 3.

Le résultat sera écrit en binaire sur le ruban.

