

## Révision IFT2105

Ce qu'il faut savoir, connaître et comprendre, au minimum, pour l'examen final. Bien sur, de longues preuves de théorèmes sont à comprendre mais pas à régurgiter - elle n'apparaissent pas sur des examens.

Ce qui est essentiel, c'est une compréhension au moins intuitive, et la capacité de s'en servir.

- Alphabet  $\Sigma$
- Mots de longueur  $k$  sur  $\Sigma$  :  $\Sigma^k$
- Mots sur  $\Sigma$  :  $\Sigma^* = \bigcup_{k \in \mathbb{N}} \Sigma^k$
- $\Sigma^*$  est dénombrable
- Langage  $L$  sur  $\Sigma$  :  $L \subseteq \Sigma^*$
- L'ensemble  $\{L : L \subseteq \Sigma^*\}$  de langages sur  $\Sigma$  n'est pas dénombrable mais strictement plus grand  $\Sigma^*$  (théorème de Cantor)
- Les mots de  $\Sigma^*$  peuvent être énumérés en ordre canonique et ceci par une machine de Turing
- Les paires de naturels  $(i, j)$  peuvent être énumérés en ordre canonique ( $(i, j) < (i', j')$  quand soit  $i + j < i' + j'$ , soit  $i + j = i' + j'$  et  $i < i'$ ).
- Automate fini déterministe  $M = (Q, \Sigma, \delta, s, F)$  (AFD)
- Automate fini non-déterministe  $M = (Q, \Sigma, \delta, s, F)$  (AFN)
- Equivalence entre AFN et AFD
- Langages réguliers (i.e. ceux acceptés par des AFD) :  $\mathcal{L}_R = \mathcal{L}_{AFD} = \mathcal{L}_{AFN}$
- Expression régulières (définition récursive)
- Equivalence entre AFD et expressions régulières :  $\mathcal{L}_R = \mathcal{L}_{er}$
- Lemme du pompiste (pumping lemma) pour les langages réguliers : ne sert JAMAIS à prouver qu'un langage EST régulier. Utile pour prouver qu'un langage n'est PAS régulier.
- L'ensemble des AFD est dénombrable, donc l'ensemble de langages réguliers est dénombrable.

- Grammaire hors-contexte  $G = (V, \Sigma, S, R)$
- Langages hors-contexte  $\mathcal{L}_{hc}$  : générés par des grammaires hors-contexte
- Notions intuitives des automates à pile et l'équivalence des automates à pile avec les grammaires hors-contexte
- Lemme du pompiste pour des langages hors-contexte (même remarque, mutatis mutandis, que pour le lemme du pompiste pour les langages réguliers)
- L'ensemble des grammaires hors-contexte est dénombrable, donc l'ensemble de langages hors-contexte est dénombrable.
- Machine de Turing déterministe  $M = (Q, \Sigma, \Gamma, \delta, q_A, q_R)$ , MTD
- Machine de Turing non-déterministe  $M = (Q, \Sigma, \Gamma, \delta, q_A, q_R)$ , MTN
- Equivalence de MTD et MTN
- Equivalence de MTD et MTD à plusieurs rubans
- Langages reconnaissables (récursivement énumérables)  $\mathcal{L}_{re}$ , *acceptés* par des MTD
- Langages décidables (récursifs)  $\mathcal{L}_r$ , *décidés* par des MTD
- L'hypothèse de Church-Turing
- L'ensemble de MTD est dénombrable, il existe donc des langages qui ne sont pas reconnaissables et des langages qui ne sont pas décidables.
- Exemples de langages indécidables mais reconnaissables ( $A_{TM} = L_u$ ) et non reconnaissables ( $L_d$ , le problème d'arrêt)
- Théorème de Rice (une compréhension des preuves d'indécidabilité)
- Les classes de complexité  $P$  et  $NP$
- La classe  $NPC$  ; exemples de problèmes dans  $NPC$  (SAT, 3-SAT, VC, CLIQUE, STABLE)