SÉANCE DE SOUTIEN

1 Automates à piles

1.1 Si vous ne l'avez pas fait au partiel

1. Soit L un langage algébrique et soit $\mathcal{A}$ un automate à pile reconnaissant L. On suppose qu'il existe un entier k tel que pour toute entrée x, et toute exécution de $\mathcal{A}$ sur x, la pile de l'automate contient au plus k éléments à tout instant de l'exécution. Montrer que L est rationnel.

1.2 Construire des automates à pile

Construisez des automates à pile reconnaissant les langages suivants par état final, et justifier leur correction.

1. L le langage engendré par la grammaire suivante $S \to aSa \mid bSb \mid \varepsilon$.
2. $L = \{u \in \{a,b\}^* : |u|_a = |u|_b\}$.
3. $L = \{u \in \{a,b\}^* : |u|_a = 2|u|_b\}$.
4. $L = \{ba^{i_1}ba^{i_2}b \cdots ba^{i_k}b : k \geq 1, \forall j \in [1,k]\ i_j \geq 1, \exists l \in [1,k]$ t.q. $i_l = l\}$.

2 Lemme d'Ogden

1. Soit $L = \{0^i1^j0^i1^j : i,j \geq 1\}$. Montrer que L n'est pas algébrique.
2. Montrer que le complémentaire de L est algébrique. Est-il rationnel ?
3. Montrer que $L = \{a^ib^ic^j : j \neq i\}$ n'est pas algébrique.

3 Problèmes indécidables

1. Soient $\alpha, \beta \in \{0,1\}^*$. Montrer que le langage $L = \{\sigma\ :\ (M_\sigma(\alpha) = M_\sigma(\beta))$ ou $(M_\sigma(\alpha)$ et $M_\sigma(\beta)$ ne s'arrêtent pas$)\}$ n'est pas récursif. (Remarque : $M_\sigma(\alpha) = M_\sigma(\beta)$ sous-entend que M_σ s'arrête sur α et β).
2. Montrer que le langage $L = \{\sigma\ :\ \exists x$ t.q. $M_\sigma(x)$ s'arrête en au plus $2|x|$ étapes$\}$ est indécidable. Est-il semi-récursif ?

4 Problème de correspondance de Post

Σ est un alphabet fini et P un ensemble fini de paires de mots sur Σ. Le Problème de Correspondance de Post associé à Σ, P est l'existence d'une suite finie non vide $(v_i, w_i)_i$ d'éléments de P telle que la concaténation des v_i soit égale à la concaténation des w_i. Le Problème de Correspondance de Post Modifié est celui de l'existence d'une telle suite lorsque le premier terme est fixé.

1. Résoudre PCP pour les instances suivantes :
   1. $P = (aab, ab), (bab, ba), (aab, abab)$

2. $P = (a, ab), (ba, aba), (b, aba), (bba, b)$
3. $P = (ab, bb), (aa, ba), (ab, abb), (bb, bab)$
4. $P = (a, abb), (aab, b), (b, aa), (bb, bba)$

2. Montrer que si Σ ne contient qu'une lettre le problème est décidable.

3. Montrer l'équivalence entre PCP et PCPM, c'est à dire qu'à partir d'un algorithme résolvant toute instance de PCP, vous pouvez créer un algorithme résolvant toute instance de $PCPM$, et inversement.

   Indication : Dans le cas PCP permet de résoudre $PCPM$, vous pourrez ajouter à l'alphabet deux lettres $*$ et $\$$ et utiliser les deux morphismes p et s suivant :
   $\forall a_1, \ldots, a_k \in \Sigma^*$, $p(a_1 \ldots a_k) = *a_1 * \ldots * a_k$ *et* $s(a_1 \ldots a_k) = s_1 * \ldots s_k *$.

4. Montrer que PCP est indécidable.

   Indication : Pour montrer cela, vous pouvez montrer que PCP permet de résoudre l'arrêt : à une machine M (dont le ruban est semi-infini) et une entrée x on peut créer une instance de PCP qui est acceptée si et seulement si la machine M s'arrête sur l'entrée x.