TD 1
Preuves, calculabilité et distances

Version du 26 septembre 2016

Exercice 1 – Preuves par récurrence

1. (**Récurrence simple pour s'échauffer**) Montrez par récurrence que séparer les n carrés d'une plaque de chocolat demande de casser la plaque $n-1$ fois (peu importe l'ordre dans lequel on choisit les rainures).
2. (**Attention au(x) cas de base**) Montrez que tout entier naturel supérieur ou égal à 8 peut s'écrire comme $3a + 5b$, avec $a, b \in \mathbb{N}$. (e.g., $19 = 3 + 3 + 3 + 5 + 5$.)
3. (**Récurrence structurelle**) Soit un arbre défini de la façon suivante :
   — Un *nœud* isolé est un arbre et c'est aussi sa racine. Le degré de ce nœud est 0.
   — À partir d'une liste d'arbres $A_1, A_2, \dots A_n$ un nouvel arbre peut être construit de la façon suivante : on crée un nouveau nœud N qui sera la racine de cet arbre, et on le relie par des arcs aux racines de chacun des A_i. n est le *degré* du nœud N.

   a) Montrez que tout arbre a un nœud de plus qu'il n'a d'arcs.

   b) On considère un arbre dont tous les nœuds sont de degré 0 ou 2. Montrez qu'un tel arbre possède k nœuds de degré 2 si et seulement si il possède $k+1$ nœuds de degré 0.
4. Quel est le lien entre la question 1 et la question 3 ?

Exercice 2 – Calculabilité

1. Les ensembles suivants sont-ils (a) récursivement énumérables, (b) récursifs ?
   — L'ensemble des nombres premiers.
   — L'ensemble des polynômes dont les coefficients sont des entiers naturels.
   — L'ensemble vide.
   — L'ensemble des programmes qui ne bouclent pas infiniment.
   — L'ensemble des programmes qui terminent en moins de 10s.
2. (**Un langage non récursivement énumérable**) Soit Σ un alphabet quelconque, notons $m_0, m_1, m_2 \dots$ la suite des mots de $\Sigma^\star$ ordonnés par ordre militaire. Par exemple si $\Sigma = \{a, b\}$, on considère l'ordre $\epsilon, a, b, aa, ab, ba, bb, \dots$

   Notons de même $A_0, A_1, A_2 \dots$ l'ensemble des algorithmes reconnaissant les langages récursivement énumérables de $\Sigma^\star$, ordonnés de la même façon (ordre militaire de leur codage, i.e. leur représentation dans votre langage de programmation préféré).

   Montrez (par l'absurde) que le langage L défini ci-après n'est pas récursivement énumérable.

$$L = \{m_i \mid A_i \text{ ne reconnaît pas } m_i\}$$

Exercice 3 – Distance préfixe

Soient u, v, et w trois mots quelconques construits sur un alphabet Σ. Rappelons que $\mathrm{plpc}(u, v)$ désigne le plus long préfixe commun à u et v.

1. Justifiez que $|\mathrm{plpc}(\mathrm{plpc}(u,v),\mathrm{plpc}(v,w))| \leq |\mathrm{plpc}(u,w)|$.
2. Justifiez que $|\mathrm{plpc}(u,v)| + |\mathrm{plpc}(v,w)| \leq \min(|\mathrm{plpc}(u,v)|,|\mathrm{plpc}(v,w)|) + |v|$.
3. Justifiez que $|\mathrm{plpc}(\mathrm{plpc}(u,v),\mathrm{plpc}(v,w))| = \min(|\mathrm{plpc}(u,v)|,|\mathrm{plpc}(v,w)|)$.
4. Déduisez-en que $|\mathrm{plpc}(u,v)| + |\mathrm{plpc}(v,w)| \leq |\mathrm{plpc}(u,w)| + |v|$
5. Montrez que $d_p(u,v) = |uv| - 2|\mathrm{plpc}(u,v)|$ est une distance.
   (Les questions précédentes vous serviront à prouver l'inégalité triangulaire.)

Exercice 4 – Distance d'édition

Rappel : la distance d'édition (ou de Levenshtein) entre u et v est le plus petit nombre d'opérations d'édition élémentaires (insertion ou suppression de symbole) à effectuer pour passer de u à v.

1. Proposez une définition récursive de cette distance.

$$d_L(u,v) = ?$$

2. Montrez qu'il s'agit bien d'une distance.