# Outils logiques et algorithmiques – TD 1 – Correction

**Exercice 1** On peut introduire les notation $P$ pour la propriété « il pleut » et $S$ pour la propriété « le fils sort ». L'affirmation de la mère est la propriété $P \implies \neg S$

1. s'il ne pleut pas alors $P$ est faux et $S$ peut être soit vrai soit faux sans contredire $P \implies \neg S$ donc on ne peut rien déduire.
2. si le fils sort alors $S$ est vrai, $\neg S$ est faux et donc la seule manière pour $P \implies \neg S$ d'être vraie est que $P$ soit faux donc on en déduit qu'il ne pleut pas.
3. si le fils ne sort pas alors $S$ est faux, $\neg S$ est vrai et $P$ peut être soit vrai soit faux sans contredire $P \implies \neg S$ donc on ne peut rien déduire.
4. Paul n'est qu'un exemple. Il pourrait être pris pour preuve d'une phrase existentielle, mais il est abusif de le prendre comme preuve d'une formule universelle comme ici.

**Exercice 2**

1. (a) Chacun mange au moins une chose
   (b) Il existe un aliment que tout le monde mange
   (c) Il existe une personne qui mange de tout
   (d) Aucune personne n'aime plus d'une chose
2. (a) $\forall y, \mathsf{mange}(\mathsf{moi}, y) \implies \mathsf{aime}(\mathsf{moi}, y)$
   (b) $\forall y, \mathsf{aime}(\mathsf{moi}, y) \implies \mathsf{mange}(\mathsf{moi}, y)$
   (c) $\neg\mathsf{aime}(\mathsf{moi}, \mathsf{cdb}) \wedge \mathsf{mange}(\mathsf{moi}, \mathsf{cdb})$
   (d) $\mathsf{aime}(\mathsf{moi}, \mathsf{cdb}) \implies \forall y, \mathsf{mange}(\mathsf{moi}, y)$
   (e) $\mathsf{mange}(\mathsf{moi}, \mathsf{cdb}) \wedge \forall y, \mathsf{mange}(\mathsf{moi}, y) \implies y = \mathsf{cdb}$
3. Les formules 2a et 2b ne sont pas équivalentes, par exemple si je ne mange rien mais que j'aime les choux de bruxelles, alors la formule 2a est vraie mais 2b est fausse.
4. Si 2c est vraie alors je n'aime pas les choux de bruxelles et donc la formule 2d est vraie par “défaut” (si les poules avait des dents...)

**Exercice 3**

1. L'algorithme s'applique à des nombres positifs ou nuls. Précondition $n \geq 0$, et spécification du résultat $r = n^2$.
2. Invariant : $r = i^2$.

**Exercice 4**

1. En notant $n$ la longueur de la chaîne $s$ :

$$\forall i \in [0, n[\,, s[i] = s[n-1-i]$$

2. Invariant : la séquence de caractères lue à l'endroit jusqu'à l'indice $i$ correspond bien à la séquence de caractères lue à l'envers, de la fin jusqu'à l'indice $j$ (cela implique aussi une égalité des longueurs).

$$\begin{cases} i = n - 1 - j \\ \forall k \in [0, i[\,, s[k] = s[n-1-k] \end{cases}$$

### Exercice 5

1. Le tableau doit être non vide.
2. Ne pas oublier : l'élément renvoyé doit faire partie du tableau ! En notant $r$ l'élément renvoyé, et $n$ la taille :

$$\begin{cases} \exists k \in [0, n[\,, r = t[k] \\ \forall k \in [0, n[\,, t[k] \geq r \end{cases}$$

3. Invariants : la variable `m` contient un élément minimal du segment $[0, i[$.

$$\begin{cases} \exists k \in [0, i[\,, m = t[i] \\ \forall k \in [0, i[\,, t[i] \geq r \end{cases}$$

4. Variante possible : renvoyer `None` pour un tableau vide, et renvoyer un $r$ selon la spécification précédente pour un tableau non vide.

### Exercice 6

1. Spécification en deux parties : il existe une répétition de la longueur indiquée, et il n'en existe pas de plus longue.

$$\begin{cases} \exists i \in [0, n - r]\,, \forall k \in [0, r[\,, s[i] = s[i + k] \\ \forall r' > r, \forall i \in \left[0, n - r'\right]\,, \exists k \in \left[0, r'\right[\,, s[i] \neq s[i + k] \end{cases}$$

2. Boucle externe : essentiellement la même chose mais sur le segment $[0, i[$ plutôt que sur $[0, n[$. Un élément en plus : le caractère précédant $s[i]$, s'il existe, est différent.

$$i > 0 \implies s[i - 1] \neq s[i]$$

Boucle interne : les caractères entre $i$ et $i + k$ sont égaux.

$$\forall j \in [i, i + k[\,, s[i + j] = s[i]$$