3. Soit R une relation sur un ensemble E. On dit qu'un élément u est un réduit de t, $t\ R^+\ u$, s'il existe une suite de réductions finie et non réduite à un élément qui va de t à u. Soit A un sous-ensemble de E tel que

   pour tout élément x de E
   si tous les réduits de x sont dans A, alors x est dans A

   Montrer que si x n'est pas dans A, il ne termine pas fortement. Montrer que si x termine fortement, il appartient à A. Montrer que si R est bien fondée alors tous les éléments de E appartiennent à A.

Exercice 4.9 (Le théorème de Newman)

Cet exercice demande d'avoir fait l'exercice 4.8.

Une relation R définie sur un ensemble E est dite *localement confluente* si à chaque fois que $t\ R\ u$ et $t\ R\ v$, il existe un élément w tel que $u\ R^*\ w$ et $v\ R^*\ w$.

1. On considère un ensemble formé de quatre éléments a, b, c et d et la relation définie par $a\ R\ b$, $b\ R\ a$, $a\ R\ c$ et $b\ R\ d$. Cette relation est-elle localement confluente ? Est-elle confluente ? Une relation localement confluente est-elle confluente ?
2. Montrer qu'une relation bien fondée et localement confluente est confluente.

4.2 Le lambda-calcul

L'idée du lambda-calcul est de rapprocher le langage des programmes de celui utilisé habituellement, en mathématiques, pour exprimer les fonctions.

Si e est une fonction, qui, à l'entier p, associe l'entier 2^p, la fonction qui à l'entier p associe l'entier 2^{2^p} s'exprime dans le langage de la section 3.4.2 par le terme $\circ_1^1(e, e)$ ou encore par le terme $\circ_1^1(e, \circ_1^1(e, \pi_1^1))$. On peut, cependant, l'écrire plus simplement $x \mapsto App(e, App(e, x))$, ou $\lambda x\ App(e, App(e, x))$, ou encore $fun\ x \rightarrow App(e, App(e, x))$, en utilisant un symbole binaire App, qui ne lie pas de variables dans ses arguments et un symbole unaire $\mapsto$, aussi noté λ ou *fun*, qui lie une variable dans son argument. En notant $(t\ u)$ le terme $App(t, u)$ l'expression ci-avant s'écrit plus simplement $fun\ x \rightarrow (e\ (e\ x))$.

Il n'est pas nécessaire d'étendre la notation *fun* pour construire des fonctions de plusieurs arguments, car on peut toujours construire de telles fonctions comme des fonctions d'un argument unique en utilisant l'isomorphisme $(A \times B) \rightarrow C = A \rightarrow (B \rightarrow C)$. Par exemple, la fonction qui à x et y associe le nombre $x \times x + y \times y$ est définie comme la fonction qui à x associe la fonction qui à y associe le nombre $x \times x + y \times y$: $fun\ x \rightarrow fun\ y \rightarrow (x \times x + y \times y)$.