Q2 1.2 Equations and conditional axioms defnition
How to use Graceful presentations

on a une quantification universelle des variables

Axiome(equation) = couple de termes t=t' de même type $(t,t' \in (T_{\Sigma,X})_s)$
où Σ=<S,F> est une signature et X un S-sorted set de variables.
Exemple d'axiome pour le String :
Pour isEmpty, nous avons besoin de ces axiomes :
isEmpty(new) = true; // isEmpty(add c to x) = false;
En fait nous avons besoin d'appeler isEmpty sur les générateurs du type String.
append : récursif:
append(new,x)=x; // append(add c to x,y) = add c to (append(x,y));
= :
(new = new) = true; // (add c to x = new) = false // (new = add c to x) = false
(add c to x = add d to y) = (c=d) and (x=y)

La spécification ci-dessus a besoin d'opérations aditionnels des sorts auxiliaires (boolean, naturals, characters)
Exemple pour les axiomes booléens (opérations : true, false, not, and, or, xor, =) :
not(true)=false, not(false)=true, (true and b) = b, ..., (false xor b) = b, (true xor b) = not(b)

Conditional axiom :
Un axiome possédant une condition telle que si la condition est vérifiée, l'égalité doit être vérifiée
Formellement :
Σ=<S,F> une signature et X un S-sorted set de variables. Les axiomes conditionnels sur X sont définis ainsi:
n conditions : t1=t1',, ...tn=tn'->t=t' $(t1,t1' \in (T_{\Sigma,X})_{s1}$, ..., $tn,tn' \in (T_{\Sigma,X})_{sn})$
et t=t' est l'axiome $(t,t' \in (T_{\Sigma,X})_s)$
Exemple :
isEmpty(x)=false -> first(add c to x) = first x
isEmpty(x)=true -> first(add c to x) = c

on peut appliquer l opperation à droite que si les conditions sont

Pas de confusion car ils sont complémentaires

ceci ne serait pas vraiment possible sans la condition

Graceful presentations :
Écrire tous les axiomes en évitant :
- écrire des axiomes contradictoires
- oublier des cas
Nous souhaitons trouver les axiomes pour chaque opération de la signature. Pour chaque opération :
1. Écrire à gauche un terme avec le nom de cette opération
2. Itérer sur les paramètres de cette opération :
    - variable pour ce paramètre
    - Si on ne peut pas faire un axiome->décomposer la variable avec les générateurs
    - Si le générateur suffit pas, utiliser des conditions
Propriété : complet et cohérent
Exemple : +
1. x+y=?
2. x+y=? -> décomposition du second paramètre (y) : x+0=x et x+succ(y)=succ(x+y)
Exemple : >
x > y = ? -> décompose y : x>0 =?, x>succ(0)=?
->décompose x : 0>0=false, succ(x)>0=true, 0>succ(y)=false, succ(x)>succ(y)=x>y

décomposer les paramètres pour couvrir tous les cas et ne pas avoir d'intersections entre les axiomes (partition)