

TD/TP 3 de LABD

janvier 2015

Les fichiers mentionnés dans ce sujet sont disponibles dans une archive .tgz, disponible sur le portail

Exercice 1 : Définir les types simples suivants en les dérivant à partir des types simples prédéfinis les plus pertinents¹ :

- une heure comprise entre 2h30 (du matin) et 16h50 ;
- un nombre réel en précision simple, supérieur ou égal à -3476.4 et strictement inférieur à 5 ;
- une chaîne de quatre caractères ;
- une chaîne de caractères qui ne peut être égale qu'à "jpg", "gif" ou "png" ;
- un type de numéro ISBN : c'est un entier à 13 chiffres comme 9782744072369².

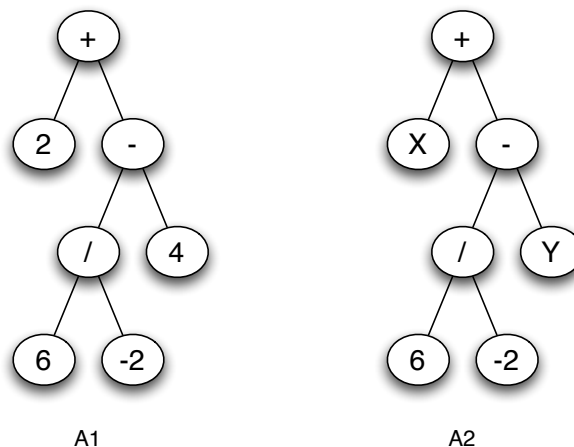
Exercice 2 :

On considère le fichier XML `okaz.xml` disponible dans l'archive.

Question 1 : Définissez un XML-Schema à partir de cette instance XML en privilégiant la concision et l'élégance³ de votre solution.

Exercice 3 :

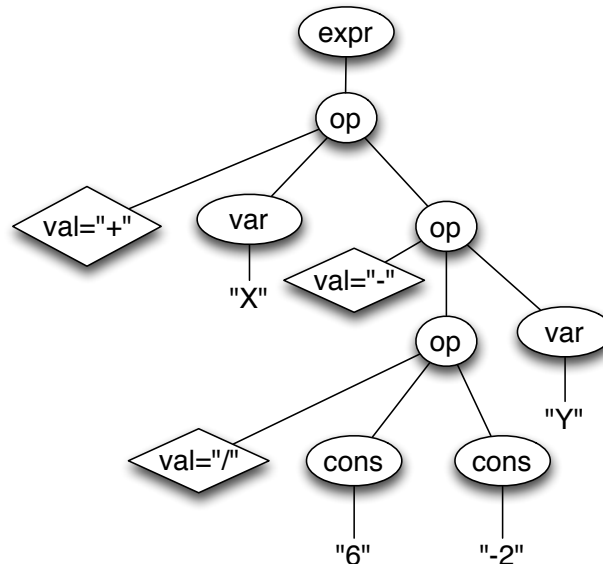
On souhaite stocker dans des fichiers XML des expressions arithmétiques entières (dans lesquelles peuvent apparaître des variables), en utilisant leur représentation arborescente. Par exemple, les expressions arithmétiques $E1 = (2 + ((6 / -2) - 4))$ et $E2 = (X + ((6 / -2) - Y))$ ont comme représentation arborescente respectives les arbres A1 et A2 ci dessous :



On décide de représenter une expression par un élément racine de nom `expr` qui ne possède qu'un seul élément fils qui peut être une constante, une variable ou un opérateur. On représente les constantes entières par des éléments de nom `cons` dont le contenu est la valeur entière correspondante, les variables par des éléments de nom `var` dont le contenu est le nom de la variable et les

1. il peut être utile de consulter la documentation <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>
2. On ne demande pas de vérifier que le numéro est valide ; par exemple le dernier chiffre est un chiffre de contrôle calculé à partir des 12 premiers. Le plus curieux peuvent consulter <http://fr.wikipedia.org/wiki/ISBN>
3. qualité complètement subjective, mais bon...

opérateurs par des éléments de nom `op` possédant un attribut de nom `val` dont la valeur peut être "+", "-", "*" ou "/". Chaque élément de nom `op` a exactement deux éléments fils (correspondant aux sous-expressions gauche et droite), chacun d'entre eux étant donc soit un élément `cons`, soit un élément `var`, soit un élément `op`. Ainsi, l'expression *E2* précédente peut être représentée par l'arbre XML :



Ce qui donne au bout du compte le fichier XML `expression.xml` disponible dans l'archive.

Question 1 : Les documents du type du fichier `expression.xml` doivent être validés par un schéma de nom `expression.xsd` comme on le voit dans l'élément racine du document. Donnez le contenu du fichier `expression.xsd` et testez la validité du fichier `expression.xml`.

Exercice 4 :

En plein championnat de football, on veut mémoriser les scores et les dates du championnat dans un fichier XML.

- Dans le championnat de France, il y a 20 clubs. On mémorisera la liste des clubs, indépendamment du calendrier du championnat. Chaque club possède un nom long (comme "Olympique de Marseille", un nom court (comme "OM").
- Le championnat est divisé en journées, et chaque journée comporte 10 rencontres. On mémorisera la date de chaque journée, mais pas la date de chaque match (parfois un match se joue en avance ou en retard par rapport à la date officielle). Il y a au total 38 journées.
- Les clubs se rencontrent tous 2 fois, une fois à domicile et une fois à l'extérieur. Les 19 premières journées, chaque club rencontre tous les autres (on parle de matches aller). Les 19 journées suivantes, chaque club rencontre aussi tous les autres mais pas sur le même terrain (on parle de matches retour). Une rencontre est donc caractérisée par un couple de (noms courts de) clubs. On mémorisera tous les scores des matches du championnat.

Question 1 : Définissez un schéma XML qui permet de représenter le championnat de France de football. La qualité d'un schéma est aussi liée à sa lisibilité. Évitez les imbrications trop profondes (schéma de type *poupées russes*).