

NFP108: exercices de modélisation avec des expressions régulières

F. Barthélemy

22 novembre 2017

Exercice 1 : nombres

1. donnez une expression régulière très simple permettant de décrire les nombres entiers positifs ou nul, en représentation décimale (base 10).
2. donnez une expression régulière représentant ces mêmes nombres en interdisant que le premier chiffre soit 0 sauf dans le cas où il n'y a qu'un seul chiffre, pour représenter le nombre nul.
3. modifiez l'expression précédente pour représenter les nombres entiers positifs, négatifs ou nul.
4. donnez une expression régulière représentant les nombres décimaux (avec une virgule).
5. donnez une expression des nombres représentant les montants en euros et centimes (nombres avec exactement deux chiffres après la virgule).

Exercice 2 : lexèmes

Dans cet exercice, on va chercher à décrire les lexèmes (en anglais, token) d'un langage de programmation au moyen d'expressions régulières. Le langage en question sera un petit sous-ensemble de java, illustré par le programme suivant.

```
public class Test{
    public static void main(String [] args){
        int max, nieme;
        boolean exaequo = false;
        max = Integer.parseInt(args[0]);
        for (int i = 1; i<args.length; i++){
            nieme = Integer.parseInt(args[i]);
            if (nieme > max){
                max = nieme;
                exaequo = false;
            }
        }
    }
}
```

```

        } else if (nieme == max){
            exaequo = true;
        }
    }
    System.out.print("Le nombre le plus grand est ");
    System.out.print(max);
    if (exaequo){
        System.out.print("(plusieurs occurrences)");
    }
    System.out.println();
}
}

```

On va se limiter aux constructions et instructions apparaissant dans cet exemple.

Question 1

- dressez la liste des mots-clés apparaissant dans ce programme (en gras).
- dressez la liste des signes de ponctuations (c'est à dire des signes utilisant un ou plusieurs symboles qui ne sont ni des chiffres ni des lettres) utilisés dans ce programme.
- donnez les expressions régulières correspondant à ces deux listes.

Question 2

- Un identificateur (nom de classe, de variable, de méthode) commence par une lettre et ne comporte que des lettres, des chiffres ou le symbole `_`. Donnez une expression régulière décrivant les identificateurs.
- Donnez une expression régulière pour les entiers positifs, négatifs ou nul.
- Donnez une expression régulière décrivant les chaîne de caractère java (caractères entre doubles quotes).

Question 3

- Donnez une expression régulière donnant l'ensemble des lexèmes (ensemble des constituants de base) du langage de programmation. Vous pourrez utiliser les expressions définies auparavant en les désignant par un nom.
- Donnez une expression régulière décrivant une déclaration de variable (on se limite aux types présents dans notre exemple).

Exercice 3 : dates

On veut représenter au moyen d'un automate fini des dates écrites avec le jour en chiffres, le mois en toutes lettres et l'année en chiffres. Les trois composantes seront séparées par un ou

plusieurs espaces.

Question 1 : petits automates

1. Donnez une expression régulière qui décrit les tantièmes avec des chiffres. Par exemple, dans la date 12 novembre 2017, 12 est le tantième.
2. donnez une expression régulière qui décrit les mois en toutes lettres.
3. donnez une expression régulière qui décrit les années entre 2000 et 2019.
4. donnez une expression régulière qui décrit une suite d'un ou plusieurs espaces.
5. comment peut-on combiner ces quatre automates pour obtenir une date complète ?

Question 2 : raffinement

En décrivant séparément chaque élément de la date, puis en les combinant, on ne décrit pas ce qui dépend de l'interaction de deux ou trois composants élémentaires, à savoir le nombre du jours d'un mois. Ce nombre dépend du mois ou du mois et de l'année pour le cas de février qui a 28 ou 29 jours selon les années. Il a 29 jours les années bissextiles. Entre les années 2000 et 2019, les seules années bissextiles sont 2004, 2008, 2012 et 2016.

On se propose de décrire dans un automate toutes les dates qui ne sont pas correctes parce qu'elles contiennent le nombre 31, 30 ou 29 dans un mois qui ne comporte pas ce numéro. Cet automate pourra être utilisé pour retirer toutes ces exceptions de l'automate écrit à la question précédente, en utilisant l'opération de différence.

Vous pouvez éventuellement décrire séparément des morceaux de l'automate des exceptions, puis donner un calcul permettant de calculer l'automate complet à partir de ces morceaux.