

TP N°1 : Premiers pas

L'objectif de ce TP est de vous faire prendre en main l'environnement antlrworks au travers d'un ensemble de petits exercices. Ces derniers ont pour but de vous faire travailler sur les expressions régulières et les grammaires tout en vous permettant de prendre en main l'environnement de travail.

1. ANTLworks

Vous trouverez dans le répertoire/share/esir2/cin deux fichiers. L'un est une archive jar contenant ANTLRworks et le second est un script de lancement. Vous pouvez lancer directement la version d'ANTLRworks depuis le répertoire. Cette version étant en java, elle est compatible Linux, windows et macosx. Lancez le script run-antlrworks pour lancer l'environnement de développement.

Lorsque vous avez écrit une grammaire sous ANTLworks, vous pouvez la tester en déroulant le menu « run » et ne lançant en mode debug. Cette option vous permet de tester vos grammaires, de visualiser les arbres de dérivation syntaxique et de syntaxe abstraite (s'il est généré). D'autre part, si votre compilateur contient du code java décrivant les différentes actions associées à votre grammaire, le mode debug exécutera les actions et vous permettra de visualiser les sorties consoles éventuelles effectuées dans vos traitements.

Attention lors de l'utilisation d'ANTLRworks :

- Ne sauvegardez jamais deux grammaires dans le même répertoire.
- Veuillez à ce que le nom du fichier dans lequel vous sauvez votre grammaire corresponde exactement au nom de la grammaire suivi de l'extension « .g ».

2. Exercices sur les expressions régulières

Dans un premier temps, lancez ANTLRworks et créez une grammaire vide. Pour tester vos expressions régulières, vous allez créer des lexèmes associés à chacune d'entre elle et une règle reconnaissant le lexème. Vous pourrez ensuite tester votre grammaire avec le mode debug de ANTLRworks et vérifier que vos expressions régulières sont correctes.

Question 2.1 : Ecrivez une expression régulière permettant de reconnaître toutes les chaînes de l'alphabet {a..z} qui contiennent les 6 voyelles dans l'ordre alphabétique.

Question 2.2 : Ecrivez une expression régulière qui reconnaît toutes les chaînes composées de 'a' et 'b' contenant au moins trois 'a'.

Question 2.3 : Ecrivez une expression régulière reconnaissant les dates au format jj/mm/aaaa (l'année est forcément composée de 4 chiffres, le mois de 2 chiffres et le jour de 2 chiffres). L'expression proposée devra faire le plus de vérifications possibles i.e. contraindre les jours en fonction des mois. Cependant vous pourrez considérer que tous les mois de février, quel que soit l'année, peuvent faire 29 jours (cette vérification est beaucoup trop complexe à effectuer avec une expression régulière).

3. Une première grammaire : les expressions numériques

Créez une nouvelle grammaire sous ANTLRworks en cochant la génération des lexèmes pour les identifiants, les entiers, les flottants, les commentaires et les caractères d'espace.

Question 3.1 : Ecrivez une grammaire permettant de reconnaître les expressions numériques composées de constantes de type entier et flottant et des opérations d'addition et de soustraction. Testez votre grammaire à l'aide du mode debug d'ANTLRworks. Vérifiez que votre grammaire fonctionne et visualisez l'arbre de dérivation syntaxique associé à cette dernière.

Question 3.2 : Nous souhaitons désormais ajouter les opérateurs de multiplication et de division. Modifiez votre grammaire de manière à ce que l'arbre de dérivation syntaxique gère la priorité des opérations de multiplication et de division par rapport aux opérations d'addition et de soustraction.

Question 3.3 : Modifiez votre grammaire pour ajouter les parenthèses et la gestion du moins unaire, tout en vous assurant que l'arbre de dérivation syntaxique représente bel et bien la priorité des opérations.

Question 3.4 : Décorez votre grammaire avec un code java vous permettant d'évaluer les expressions arithmétiques ainsi saisies et affichant le résultat sur la sortie standard.

Question 3.5 : Ajoutez la possibilité de stocker le résultat d'une expression dans une variable en utilisant la syntaxe « set variable = expression ». Cette variable devra ensuite pouvoir être utilisée dans les expressions saisies et sera remplacée, dans le calcul, par la valeur qui lui est associée.