

Expressions ensemblistes

1 Sujet

Le but du projet est de concevoir à l'aide des outils **Lex** et **Accent**, un analyseur d'expressions ensemblistes préfixées.

2 Description des expressions ensemblistes

2.1 Construction des expressions

Nous considérons les opérations d'affectation dont les membres droits sont des expressions ensemblistes **préfixées**. Les expressions prises en compte sont construites à partir des opérateurs ensemblistes suivants :

- L'union : opérateur n -aire avec $n \geq 2$,
- L'intersection : opérateur n -aire avec $n \geq 2$,
- Le complémentaire : opérateur binaire,
- La différence : opérateur binaire.

Nous nous limitons aux ensembles d'entiers positifs. Nous désignons les ensembles par une lettre majuscule (A, \dots, Z) et le contenu d'un ensemble sera représenté par **une suite d'entiers vide ou non séparés par des virgules, le tout entre accolades**. Enfin, l'opérateur d'affectation sera noté $:=$

Remarque : Un entier sera définie par une suite non vide de chiffres.

2.2 Grammaire

La grammaire ci-dessous permet de décrire une suite d'affectations d'expressions ensemblistes séparées les unes des autres par un seul et unique retour à la ligne et avec un retour à la ligne après la dernière affectation.

D	\rightarrow	$listeAffExp$ rc
$listeAffExp$	\rightarrow	$listeAffExp$ rc $affExp$
		$ $ $affExp$
$affExp$	\rightarrow	lettre $:=$ $ensemble$
$ensemble$	\rightarrow	$\{ listeEntiers \}$
		$ $ $\{ \}$
		$ $ lettre
		$ $ exp
$listeEntiers$	\rightarrow	$listeEntiers$, ent
		$ $ ent
exp	\rightarrow	$(opNaire$ $ensemble$ $ensemble$ $listeEnsemble$ $)$
		$ $ $(opBinaire$ $ensemble$ $ensemble$ $)$
$listeEnsemble$	\rightarrow	$listeEnsemble$ $ensemble$
		$ $ ε
$opNaire$	\rightarrow	union
		$ $ inter
$opBinaire$	\rightarrow	comp
		$ $ diff

où

- le terminal **rc** désigne le retour à la ligne,
- le terminal **lettre** désigne le nom des ensembles (lettre majuscule),
- le terminal **ent** désigne les entiers (suite non vide de chiffres),
- le terminal **union** désigne l'union,
- le terminal **inter** désigne l'intersection,

- le terminal **comp** désigne le complémentaire et
- le terminal **diff** désigne la différence.

3 Découpage du projet

Le projet est décliné en plusieurs versions. Vous devez implémenter chaque version à l'aide des outils **Lex** et **Accent**.

3.1 Première version : Analyses lexicale et syntaxique

A partir de la grammaire de la section 2.2, écrivez dans un répertoire de nom **V1** les fichiers **expens.lex** et **expens.acc** permettant de reconnaître toute suite d'affectations d'expressions ensemblistes reconnue par la grammaire.

Exemple de suite reconnue :

```

A  := {21, 4, 18, 129}
C  := {}
B  := (union A {1} C {10, 3})
F  := (diff B {4, 18, 0, 100})
E  := (comp B {1, 4, 10})
D  := (inter A (diff A {21}) B {18, 4})
G  := {15}

```

Remarques : Votre analyseur lexical (fichier **expens.lex**) doit :

- Ne rien faire (action nulle) lorsqu'il rencontre une suite non vide d'espaces et de tabulations,
- Générer une erreur lexicale lorsqu'il rencontre un caractère non défini dans la grammaire.

3.2 Deuxième version : Vérification

Dans un répertoire de nom **V2**, ajoutez à vos fichiers **expens.lex** et **expens.acc** de la section 3.1, des attributs et des actions sémantiques permettant de vérifier que dans toute suite d'expressions reconnue, il n'existe pas d'instruction d'affectation dans laquelle un ensemble apparaît à la fois en membre gauche et en membre droit de l'opérateur d'affectation.

Exemple : L'expression suivante n'est pas correcte car l'ensemble **C** apparaît en membre droit et en membre gauche de l'affectation :

```
C := (union A B C)
```

Remarques : Pour effectuer le traitement demandé, vous devez :

- utiliser un attribut de type caractère associé au terminal **lettre** pour récupérer le nom de chaque ensemble et donc utiliser un fichier **yystype.h** (voir sujet de TP n°4).
- utiliser un attribut hérité dans votre fichier **expens.acc** (voir section 1.3 du sujet de TP n°3).
Cet attribut doit servir à faire "descendre" le nom de l'ensemble se trouvant en membre gauche d'une affectation dans le sous-arbre d'analyse correspond au membre droit de l'affectation (voir l'attribut hérité utilisé dans l'exercice 2 du TD n°3).
- arrêter l'analyse dès qu'une erreur est détectée en indiquant le problème rencontré.

3.3 Troisième version : Affichage

Dans un répertoire de nom **V3**, ajoutez à vos fichiers **expens.lex** et **expens.acc** de la section 3.2, des attributs et des actions sémantiques permettant d'afficher les expressions reconnues en notation infixée.

Attention, l'opérateur complémentaire restera en notation préfixée.

Expression	Notation infixée
A := {21, 4, 18, 129}	A := {21, 4, 18, 129}
C := {}	C := {}
B := (union A {1} C {10, 3})	B := (A union {1} union C union {10, 3})
F := (diff B {4, 18, 0, 100})	F := (B diff {4, 18, 0, 100})
E := (comp B {1, 4, 10})	E := (comp B {1, 4, 10})
D := (inter A (DIF A {21}) B {18, 4})	D := (A inter (A diff {21}) inter B inter {18, 4})

Remarques :

- Vous devez utiliser un attribut associé au terminal **entier** pour récupérer la valeur de l'entier reconnu par l'analyseur lexical et donc modifier votre fichier **yystype.h**.
- Dans le fichier **expens.acc**, vous devez utiliser plusieurs attributs certains synthétisés, certains hérités pour effectuer l'affichage. Vous pouvez vous inspirer des attributs et traitements de la question 1 de l'exercice 2 du TD n°3.

4 Rendu

Pour le **lundi 1^{er} février au plus tard**, vous devez déposer dans l'activité **Projet - Rendu final** du cours **Langages et Traducteurs - IS2A3** une archive compressée de votre travail de nom **nom_prenom.tgz** où **nom** et **prénom** sont vos nom et prénom :

```
tar -czvf nom_prenom.tgz projet
```

Cette archive doit contenir :

1. Tous les fichiers se trouvant dans vos répertoires **V1**, **V2** et **V3**.
2. Un rapport de 5 pages maximum contenant :
 - (a) L'état d'avancement de chacune des versions du projet et les éventuels problèmes rencontrés pour la réalisation de chacune de ces versions.
 - (b) La déclaration complète des attributs utilisés dans les versions 2 et 3 du projet : attribut synthétisé ou hérité, type de valeur (entier, réel, caractère, ...), symbole(s) de la grammaire associé(s) et rôle.

Remarque : Il n'est demandé ni introduction, ni conclusion pour le rapport.