

L3 informatique, 2021-2022, Projet de LFC

Traduction de Markdown en html

TP2 : Table des symboles

Table des symboles

La table des symboles est une structure (pas forcément un tableau unique) dans laquelle on stocke des informations sur les différents éléments du fichier d'entrée. Cette table doit comporter suffisamment d'informations pour pouvoir faire une analyse sémantique et la traduction du fichier en fin d'analyse.

Dans le cadre du projet de cette année, la table des symboles va devoir permettre de stocker les informations sur les morceaux de texte, ainsi que les étoiles et les underscores.

- Pour chacun de ces éléments, il faut d'abord stocker le fait que ce soit un morceau de texte, une étoile ou un underscore.
 - Pour chaque morceau de texte, il va falloir stocker
 - la chaîne de caractères constituant ce morceau de texte,
 - le fait qu'il s'agisse d'un titre, d'un item de liste à puce ou d'un texte « quelconque »
 - s'il s'agit d'un titre, de quel niveau il est,
 - s'il s'agit d'un item de liste à puce ou d'un texte quelconque, quelle est sa mise en forme (gras, italique, gras+italique, aucune)
 - Pour chaque étoile et chaque underscore, s'il s'agit d'un caractère faisant partie du texte ou d'une balise de mise en forme.

Toutes les informations sur les textes ne pourront pas être stockées en même temps dans la structure. Certaines informations seront connues au moment de l'analyse lexicale, d'autres seulement au moment de l'analyse syntaxique.

Lors de cette séance de TP, vous allez créer et commencer le remplissage de la structure, uniquement la partie que l'on peut remplir lors de l'analyse lexicale.

Voici un exemple de structure que vous pouvez obtenir en fin d'analyse lexicale après lecture du fichier Exemple fourni lors du dernier TP.

	position	longueur	texte/étoile/underscore	catégorie	texte	niveau	titre
0	1	41	texte	normal			Voici un texte qui contient des étoiles
1	0	0	étoile				*
2	42	19	texte	normal			simples ou doubles
3	0	0	étoile				*
4	0	0	étoile				*
5	61	26	texte	normal			un underscore pour tester
6	0	0	underscore				—
7	87	13	texte	normal			et même deux
8	0	0	underscore				—
9	0	0	underscore				—
10	100	37	texte	normal			et des retours à la lignes simples,
11	137	16	texte	normal			des lignes vides
12	153	19	texte	normal			des listes à puces
13	172	6	texte	item			point1
14	178	7	texte	item			point 2
15	185	11	texte	normal			des titres
16	196	6	texte	titre	2		Titre1
17	202	6	texte	titre	3		Titre2
18	208	24	texte	normal			Et des textes incorrects
19	232	31	texte	normal			avec un caractère non autorisé
20	263	30	texte	normal			une balise de titre incorrecte
21	293	12	texte	normal			et une autre
22	0	0	étoile				*
23	305	43	texte	normal			un point de liste incorrect (espace devant)
24	0	0	étoile				*
25	348	32	texte	normal			et un autre (pas d'espace après)

Ce tableau (appelé TAB dans le reste de l'énoncé) est associé à la chaîne de caractère ci-dessous (appelée CH dans la reste de l'énoncé).

Voici un texte qui contient des étoiles simples ou doubles un underscore pour tester et même deux et des retours à la lignes simples, des lignes videsdesdes listes à pucespoint1point 2 des titresTitre1Titre2Et des textes incorrectsavec un caractère non autorisé une balise de titre incorrecteet une autreun point de liste incorrect (espace devant)et un autre (pas d'espace après)

Le tableau est un tableau d'entiers. Il suffit d'associer un code numérique à chaque catégorie et sous-catégorie, code défini en début de programme. Exemples : texte=0, étoile=1, titre=11, etc. Les deux premières colonnes du tableau indiquent à quelle position de la chaîne CH commence chaque morceau de texte et quelle est sa longueur. Les morceaux de texte eux-mêmes sont stockés dans la chaîne CH.

Remarques :

- Les numéros se trouvant à gauche du tableau sont les indices des lignes auxquelles sont stockés les différents éléments dans le tableau. Il ne s'agit pas d'une colonne du tableau.
- Les chaînes se trouvant à droite du tableau sont les chaînes auxquelles correspondent les informations stockées. Les chaînes elles-mêmes ne sont pas dans le tableau.
- Ne cherchez pas à optimiser la place utilisée par la structure ; ce n'est pas l'objet du projet. Déclarez un tableau de 50 ou 100 lignes, cela suffira pour tester votre programme.

Etapes de la réalisation

Si le programme lex que avez rendu à l'étape 2 du projet fonctionne correctement, vous pouvez repartir de celui-ci ; sinon partez du corrigé fourni sur plubel.

Cette étape du projet n'est pas notée mais elle devra être terminée avant le TP4.

Version de base

Commencez par faire en sorte de stocker dans la structure TAB+CH tous les morceaux de texte, étoiles et underscores sans information supplémentaire (trois premières colonnes du tableau TAB et chaîne de caractères CH).

Pour cela, écrivez une fonction qui fait l'insertion des informations dans le tableau TAB et la chaîne CH, en fonction de ce que l'on cherche à insérer, et qui renvoie l'indice de ligne à laquelle les informations ont été stockées dans le tableau TAB. Cette fonction sera appelée dans le programme lex lorsqu'un morceau de texte, une étoile ou un underscore sera reconnu et l'indice renvoyé sera affiché en même temps que l'unité lexicale (vous aurez besoin d'avoir cet indice sous la main dans la suite du projet).

Niveaux des titres

Ecrivez une fonction qui permet de compter, dans une chaîne de caractères, combien il y a de #. Utilisez cette fonction pour déterminer le niveau du titre qui va suivre une balise de titre au moment de la reconnaissance d'une telle balise. Faites en sorte d'afficher ce niveau en même temps que l'unité lexicale « balise de titre ».

Reconnaissance des titres et items de listes

Pour cette partie, vous allez devoir utiliser des états.

Fonctionnement des états dans lex

Dans lex, il existe un état qui s'appelle INITIAL. C'est l'état dans lequel on se trouve par défaut. Il est inutile de le déclarer.

Pour déclarer un nouvel état, par exemple NOUV, il faut écrire `%start NOUV` entre le `%}` et le `%%` du programme lex.

On utilise ensuite les états dans les expressions régulières :

L'expression régulière `<NOUV>[a-z]+` signifie « n'importe quelle chaîne de lettres minuscules non accentuées lue dans l'état NOUV ».

Les changements d'état se font dans les actions : `BEGIN NOUV` est l'instruction qui signifie « passer dans l'état NOUV ». Pour sortir de l'état NOUV, il suffit de repasser dans l'état INITIAL avec l'instruction `BEGIN INITIAL`.

Modifiez votre programme lex de telle façon qu'on change d'état lorsqu'on lit une balise de titre ou un point de choix (deux états différents) et qu'on sorte de cet état lorsqu'on lit le retour à la ligne (ou ligne vide) suivant.

Vous pouvez maintenant ajouter les fonctions/instructions permettant de remplir les colonnes 4 et 5 du tableau TAB avec les informations concernant les items de listes à puces et les titres.