

Grammaires et Langages - TD

Préparation au projet

1 Structure XML

Soit le document suivant :

Réaliser un compilateur

Nino Silverio

Introduction : concepts de base

Tout programme rédigé dans un langage de programmation de haut niveau tel que Pascal ou C ne peut être exécuté par un ordinateur que s'il est traduit en instructions exécutables par l'ordinateur, généralement des instructions en langage machine. L'ordinateur qui doit exécuter le programme traduit est appelé machine cible.

Réalisation d'un compilateur

Comme l'écriture d'un compilateur est une tâche fort complexe, le programmeur a tout intérêt à travailler en utilisant un langage de programmation de haut niveau.

1. Proposez une structuration des éléments qui la composent (titre, auteur, *etc.*)
2. Insérez les balises XML dans le texte pour délimiter les éléments précédemment trouvés
3. Proposez un schéma XSD décrivant ces éléments

2 Grammaire XML

Proposez une grammaire au format *bison* permettant d'analyser un document XML bien formé. Vous vous baserez sur la description BNF de la grammaire XML ainsi que la liste des jetons renvoyés par l'analyseur *flex* fournies.

3 Structure de données

1. Proposez une structure de données (diagramme de classes) permettant de stocker une arborescence XML, ses attributs ainsi que ses informations d'en-têtes.
2. Proposez une implémentation en C++ utilisant un maximum les composants standards de la STL.
3. Effectuez une coloration de la grammaire *bison*, c'est-à-dire identifiez quel va être le type associé à chaque symbole non-terminal. Les types seront choisis parmi les classes identifiées dans l'étape précédente.
4. Complétez les méthodes et constructeurs de vos classes afin de pouvoir effectivement construire la structure de données dans les actions *bison* associées à chaque règle.
5. Identifiez comment le document XML sera retourné au contexte appelant par l'intermédiaire de la directive `%parse-param`.