Licence 2 SdN - Algorithme avancé - Projet nanoml

Contexte

L'objectif du projet est d'écrire un programme qui réalise la vérification syntaxique d'un fichier texte devant contenir un document au « format nanoml ». Ce format est librement inspiré du format html et sa syntaxe est décrite ci-après.

Notation

Le projet se découpe en étapes :

1. Travail minimal sur 14 points : Réaliser l'analyse syntaxique.
2. Version supérieure sur 4 points : Réaliser construire l'arbre n-aire représentant le fichier nanoml.
3. Version complète sur 2 points : Réaliser l'affichage de l'arbre nanoml avec des boites imbriquées comme précisé ci-après.
4. Rapport sur 10 points : Écrire un rapport présentant votre travail.
5. Une fois que vous avez validé le travail minimal, il est possible de proposer des améliorations au projet. La pertinence de chaque amélioration doit être discutée dans le rapport. Ces ajouts pourront être intégrés dans des programmes séparés afin de faciliter la notation du travail minimal. Chaque ajout pourra apporter jusqu'à 2 points bonus (sur le barème sur 30).
6. La note finale sur 20 du projet sera obtenue par une règle de trois à partir de la somme des 4 composantes précédentes.

Grammaire

La grammaire imposée pour le projet est une grammaire de type LR(1) dont l'implantation d'un analyseur syntaxique repose sur des discussions contrôlant qu'un seul élément syntaxique[1] à la fois. Elle doit vous servir de support à votre développement. Nous vous exhortons à la considérer comme un guide[2]. L'analyse syntaxique permettra de construire un arbre n-aire qui sera charger complétement en mémoire[3] (cf. point 2 de la partie notation).

Inspiré de la forme de Backus-Naur

- `::=` signifie « se réécrit comme » ;
- `{truc}` signifie « répétition de truc entre 0 et un nombre arbitraire » ;
- `|` signifie « ou » ;
- Les parenthèses `()` servent à définir des règles de priorité ;
- `epsilon` est un mot vide. C'est un mot qui ne contient rien ;
- `<elt>` signifie que l'élément est définie comme une combinaison d'autre éléments ou termes ;
- `'terme'` signifie que le terme apparait dans le texte tel quel.

1	`<texte_enrichi> ::= <document> <annexes>`
2	`<document> ::= 'debut_doc' <contenu> 'fin_doc'`
3	`<annexes> ::= { 'debut_annexe' <contenu> 'fin_annexe' }`
4	`<contenu> ::= { <section> | <titre> | <mot_enrichi> | <liste> }`
5	`<section> ::= 'debut_section' <contenu> 'fin_section'`
6	`<titre> ::= 'debut_titre' <texte> 'fin_titre'`
7	`<liste> ::= 'debut_liste' { <item> } 'fin_liste'`
8	`<item> ::= 'debut_item' ( <liste_texte> | <texte_liste> ) 'fin_item'`
9	`<liste_texte> ::= <liste> <texte_liste> | epsilon`
10	`<texte_liste> ::= <texte> <liste_texte> | epsilon`
11	`<texte> ::= { <mot_enrichi> }`
12	`<mot_enrichi> ::= <mot_simple> | <mot_important> | 'retour_a_la_ligne'`
13	`<mot_important> ::= 'debut_important' { <mot_simple> } 'fin_important'`

[1] Le fameux Token ou jeton.

[2] Vous pourrez modifier cette grammaire pour simplifier des ajouts au projet cf point 5 de la partie notation.

[3] Une solution consistant à faire le rendu durant l'analyse syntaxique est possible mais n'est pas souhaité.

Interprétation

Le rendu se fera au format texte avec une mise en page spécifique. Cette partie présente ces spécificités afin de définir un standard. Cependant les spécificités seront présentées au travers d'exemples. Si ces exemples ouvrent la porte à des interprétations, vous indiquerez alors dans le rapport les options envisagées et vous justifierez le choix retenu.

Le rendu est un flux ASCII de lignes comportant exactement 50 caractères chacune. Le contenu du document, d'une annexe et d'une section s'écrivent dans une boîte. On constate qu'une boite peut être placer dans une boîte. La place disponible pour écrire le texte diminue à chaque encapsulation de section.

Exemple 1

Source 1

```
<document>
   <section>
      Bonjour le Monde.
   </section>
</document>
<annexe>
   Auteur : Weinberg Benjamin
   Date : 23/01/2024
</annexe>
```

Rendu 1

```
+------------------------------------------------+
|+----------------------------------------------+|
||Bonjour le Monde.                             ||
|+----------------------------------------------+|
+------------------------------------------------+
+------------------------------------------------+
|Auteur : Weinberg Benjamin Date 23/01/2024      |
+------------------------------------------------+
```

On constate sur cet exemple que le document est délimité par la première boîte faisant 50 caractères de large. Dans cette première boîte, il y a une deuxième boîte faisant 48 caractères de large. La troisième boîte délimite la première et unique annexe. On remarque également que les retours à la ligne dans le fichier source n'implique pas de retour à la ligne dans le rendu.

Exemple 2

Source 2

```
<document>
   <section>
      <titre> Sujet de projet Nanoml </titre>
   </section>
</document>
<annexe>
     Auteur : Weinberg Benjamin <br/>
     Date : 23/01/2024 <br/>
</annexe>
```

Rendu 2

```
+------------------------------------------------+
|+----------------------------------------------+|
||SUJET DE PROJET NANOML                        ||
|+----------------------------------------------+|
+------------------------------------------------+
+------------------------------------------------+
|Auteur : Weinberg Benjamin                      |
|Date : 23/01/2024                               |
```

```
+-------------------------------------------+
```

La balise
 permet de retourner à la ligne dans un texte. Le contenu d'un élément de type `<titre>` est écrit en capitale. Bien que ceci n'apparait pas clairement sur l'exemple, le titre impose le retour à la ligne avant d'écrire la suite du document.

Exemple 3

Source 3

```
<document>
   <titre> Sujet de projet Nanoml </titre>
   <section>
           <titre> Grammaire </titre>
           inspiree par la forme BN.
           <liste>
                <item> texte_enrichi ::= document annexes </item>
                <item> document  ::= 'debut_doc' contenu 'fin_doc' </item>
                <item>  annexes  ::=  {  'debut_annexe'  contenu  'fin_annexe'  }
</item>
                <item> ... </item>
           </liste>
   </section>
   <section>
           <titre> Rendu </titre>
   </section>
</document>
<annexe>
     Auteur : Weinberg Benjamin <br/>
     Date : 23/01/2024 <br/>
</annexe>
```

Rendu 3

```
+-----------------------------------------------+
|SUJET DE PROJET NANOML                         |
|+---------------------------------------------+|
||GRAMMAIRE                                     ||
||inspiree par la forme BN.                     ||
||  #  texte_enrichi ::= document annexes       ||
||  #  document ::= 'debut_doc' contenu          ||
||  'fin_doc'                                    ||
||  #  annexes ::= { 'debut_annexe' contenu      ||
||  'fin_annexe' }                               ||
||  #  ...                                       ||
|+---------------------------------------------+|
|+---------------------------------------------+|
||RENDU                                         ||
|+---------------------------------------------+|
+-----------------------------------------------+
+-----------------------------------------------+
|Auteur : Weinberg Benjamin                     |
|Date : 23/01/2024                              |
+-----------------------------------------------+
```

Dans cet exemple, les listes à puces sont utilisées. Les listes à puces indentent son contenu et le rendu des objets énumérés dans la liste[4] débute avec le caractère #. Il est possible d'utiliser des listes à puce à l'intérieur d'un item, ce qui augmentent l'indentations. Dans ce cas, l'item peut contenir un mélange de texte et de liste[5].

On remarque également pour un souci de simplicité le caractère < n'est pas protégé. Ainsi dans la version demandée, il n'est pas possible de voir apparaitre notamment <document>[6] dans le rendu.

Exemple 4

Source 4

```
<document>
   <titre> Sujet de projet Nanoml </titre>
   <section>
           <titre> Grammaire </titre>
           inspiree par la forme BN.
           <liste>
                 <item> texte_enrichi ::= document annexes </item>
                 <item> document  ::= 'debut_doc' contenu 'fin_doc' </item>
                 <item>  annexes  ::=  {  'debut_annexe'  contenu  'fin_annexe'  }
</item>
                 <item> ... </item>
           </liste>
   </section>
   <section>
           <titre> Demarche du projet </titre>
           <liste>
                 <item> Ecrire une permiere version de l'analyseur gerant les
balises
                       <liste>
                             <item> document </item>
                             <item> annexe </item>
                             <item> section </item>
                             <item> et le texte </item>
                       </liste>
                 </item>
                 <item> Decorer l'analyseur pour que l'arbre N-aire soit creer
                       <liste>
                             <item>  Ajouter  les  parametres  necessaire  aux
fonctions </item>
                             <item> Manipuler les noeuds de l'arbre </item>
                             <item>  reprendre  le  programme  pour  qu'il  soit
maintenable/reutilisable </item>
                             <item> reutiliser le travail sur les arbres
                                   <liste>
                                         <item> utils </item>
                                         <item> noeud </item>
                                         <item>  (arbre_binaire  peut  aider)
</item>
                                   </liste>
                             </item>
                       </liste>
                 </item>
                 <item> Afficher l'arbre N-aire
                       <liste>
                             <item> Definir les routines necessaires </item>
                             <item>  Repartir  dans  des  bonnes  bibliotheques
</item>
                             <item> obtenir un rendu </item>
```

[4] Utilisation de la balise <item>

[5] Cf les règles 8 à 10.

[6] Et toutes les autres balises.

```
                    </liste>
                </item>
                <item> Ajouter ce qui manque
                    <liste>
                        <item> les titres </item>
                        <item> les balises retour_a_la_ligne </item>
                        <item> les balises "important" </item>
                        <item> listes et items (Danger) </item>
                    </liste>
                </item>
                <item> Envisager des extensions </item>
            </liste>
    </section>
</document>
<annexe>
    <titre> Contacts </titre>
    Auteur : Weinberg Benjamin <br/>
    Date : 23/01/2024 <br/>
</annexe>
<annexe>
    <section>
        <titre> Nota Bene </titre>
        Dans un premiere version, on ne pourra pas voir apparaitre dans un
rendu les element syntaxique necessaire a la structuration du document.
        En effet pour realiser ceci, il faudrait que l'analyseur puisse
distinguer entre une balise qui structre le document et
        une balise que doit apparaitre comme du texte.
        Ceci pourrait etre realiser en affinant l'analyseur, notamment en
ajoutant des caracteres d'evitement.
    </section>
    <section>
        <titre> Liste non exhaustive d'amelioration </titre>
        <liste>
            <item> affiner l'analyseur pour que les balises se decoupe en
3 parties : les chevrons ouvrant et fermant et le nom de l'element; </item>
            <item> ajouter un moyen de faire un echappement pour qu'un
texte puisse contenir le texte d'une balise; </item>
            <item> gerer une numerotation des titres en fonction de la
hierachie des sections; </item>
            <item> ajouter des balises. </item>
        </liste>
    </section>
</annexe>
```

Rendu 4

```
+-----------------------------------------------+
|SUJET DE PROJET NANOML                         |
|+---------------------------------------------+|
||GRAMMAIRE                                     ||
||inspiree par la forme BN.                     ||
||  #  texte_enrichi ::= document annexes       ||
||  #  document ::= 'debut_doc' contenu          ||
||  'fin_doc'                                    ||
||  #  annexes ::= { 'debut_annexe' contenu      ||
||  'fin_annexe' }                               ||
||  #  ...                                       ||
|+---------------------------------------------+|
|+---------------------------------------------+|
||DEMARCHE DU PROJET                            ||
||  #  Ecrire une permiere version de           ||
```

```
||  l'analyseur gerant les balises              ||
||    #  document                                ||
||    #  annexe                                  ||
||    #  section                                 ||
||    #  et le texte                             ||
||  #  Decorer l'analyseur pour que l'arbre     ||
||  N-aire soit creer                           ||
||    #  Ajouter les parametres necessaire aux  ||
||    fonctions                                 ||
||    #  Manipuler les noeuds de l'arbre        ||
||    #  reprendre le programme pour qu'il soit ||
||    maintenable/reutilisable                  ||
||    #  reutiliser le travail sur les arbres   ||
||      #  utils                                ||
||      #  noeud                                ||
||      #  (arbre_binaire peut aider)           ||
||  #  Afficher l'arbre N-aire                  ||
||    #  Definir les routines necessaires       ||
||    #  Repartir dans des bonnes bibliotheques ||
||    #  obtenir un rendu                       ||
||  #  Ajouter ce qui manque                    ||
||    #  les titres                             ||
||    #  les balises retour_a_la_ligne          ||
||    #  les balises "important"                ||
||    #  listes et items (Danger)               ||
||  #  Envisager des extensions                 ||
|+-----------------------------------------------+|
+-------------------------------------------------+
+-------------------------------------------------+
|CONTACTS                                         |
|Auteur : Weinberg Benjamin                       |
|Date : 23/01/2024                                |
+-------------------------------------------------+
+-------------------------------------------------+
|+-----------------------------------------------+|
||NOTA BENE                                       ||
||Dans un premiere version, on ne pourra pas      ||
||voir apparaitre dans un rendu les element       ||
||syntaxique necessaire a la structuration du     ||
||document. En effet pour realiser ceci, il       ||
||faudrait que l'analyseur puisse distinguer      ||
||entre une balise qui structre le document et    ||
||une balise que doit apparaitre comme du         ||
||texte. Ceci pourrait etre realiser en           ||
||affinant l'analyseur, notamment en ajoutant     ||
||des caracteres d'evitement.                     ||
|+-----------------------------------------------+|
|+-----------------------------------------------+|
||LISTE NON EXHAUSTIVE D'AMELIORATION             ||
||  #  affiner l'analyseur pour que les balises   ||
||  se decoupe en 3 parties : les chevrons        ||
||  ouvrant et fermant et le nom de l'element;    ||
||  #  ajouter un moyen de faire un echappement   ||
||  pour qu'un texte puisse contenir le texte     ||
||  d'une balise;                                 ||
||  #  gerer une numerotation des titres en       ||
||  fonction de la hierachie des sections;        ||
||  #  ajouter des balises.                       ||
|+-----------------------------------------------+|
+-------------------------------------------------+
```