

L3 informatique, 2022-2023, Projet de LFC

Traduction iCal en html

1 Corrigé de l'étape 1

Liste des unités lexicales à reconnaître par l'analyseur lexical

n°	Unité lexicale	Description
1	début calendrier	chaîne « BEGIN:VCALENDAR »
2	fin calendrier	chaîne « END:VCALENDAR »
3	début événement	chaîne « BEGIN:VEVENT »
4	fin événement	chaîne « END:VEVENT »
5	intro heure début evt unique	chaîne « DTSTART: »
6	intro heure fin evt unique	chaîne « DTEND: »
7	date et heure evt unique	chaîne de caractères constituée de 8 chiffres entre 0 et 9 puis le caractère « T » puis 6 chiffres entre 0 et 9 puis le caractère « Z »
8	intro titre	chaîne « SUMMARY: »
9	intro lieu	chaîne « LOCATION: »
10	intro description	chaîne « DESCRIPTION: »
11	titre/lieu/description	chaîne de caractères quelconques sans \n ni « : »
12	début alarme	chaîne « BEGIN:VALARM »
13	fin alarme	chaîne « END:VALARM »
14	intro position alarme	chaîne « TRIGGER: »
15	position alarme	chaîne de caractères constituée de la chaîne « -P » suivie d'un entier puis de « DT » puis d'un entier puis de « H » puis d'un entier puis de « M » puis d'un entier puis de « S ».
16	intro heure début evt répétitif	chaîne de caractères commençant par « DTSTART;TZID= » suivie d'une chaîne de caractères (que l'on va limiter aux lettres non accentuées et aux /) suivie de « : »
17	intro heure fin evt répétitif	chaîne de caractères commençant par « DTEND;TZID= » suivie d'une chaîne de caractères (que l'on va limiter aux lettres non accentuées et aux /) suivie de « : »
18	date et heure evt répétitif	chaîne de caractères constituée de 8 chiffres entre 0 et 9 puis le caractère « T » puis 6 chiffres entre 0 et 9
19	intro règle répétition	chaîne « RRULE: »
20	intro fréquence	chaîne « FREQ= »
21	intro compteur	chaîne « COUNT= »
22	intro liste jours	chaîne « BYDAY= »
23	intro limite	chaîne « UNTIL= »
24	frequence	chaîne de caractères parmi {DAILY, WEEKLY, MONTHLY, YEARLY}
25	nombre entier	chaîne de caractères de longueur supérieure ou égale à 1 constituée de chiffres entre 0 et 9
26	liste jours	suite de chaînes de caractères prises parmi {SU, MO, TU, WE, TH, FR, SA} et séparées par des virgules
27	changement de semaine	chaîne « WKST=SU »
28	séparateur d'options	caractère « ; »
29	intro heure début evt journée	chaîne « DTSTART;VALUE=DATE: »
30	intro heure fin evt journée	chaîne « DTEND;VALUE=DATE: »
31	date evt journée	chaîne de caractères constituée de 8 chiffres entre 0 et 9

Explications :

Lorsqu'on établit la liste des unités lexicales à reconnaître, on ne se préoccupe pas de l'endroit où on peut trouver ces chaînes. C'est en écrivant le programme lex que l'on fera en sorte que l'analyseur lexical ne retienne que celles qui nous intéressent. Par exemple, on peut trouver des dates et heures au format `date_débutTheure_débutZ` après « DTSTART: » et « DTEND: » mais aussi après « DTSTAMP: », « CREATED: » et « LAST-MODIFIED: ». Ici, on indique juste de quoi sont constituées les dates et heures. On ajoutera ce qu'il faut dans le programme lex pour qu'il ne garde que celles qui sont après « DTSTART: » et « DTEND: ».

Pour les dates et heures : On se contente au niveau de l'analyse lexicale de reconnaître des suites de chiffres à la bonne longueur. C'est au niveau de l'analyse sémantique que l'on vérifiera que ces suites de chiffres correspondent à des heures et dates valides.

Pour tous les mots clés suivis d'un « : » (DTSTART, DTEND, RRULE, ...) on inclut le « : » dans la description de l'unité lexicale car il n'y a aucun intérêt à gérer ce symbole individuellement. Cela nous simplifiera la suite de l'analyse.

De même pour le « = » suivant certains mots clés comme `FREQ`, `COUNT`, ...

Pour l'instant, on gère la liste de jours (après « BYDAY= ») comme une seule unité lexicale. On pourra ensuite, au moment de l'analyse sémantique, extraire les jours de la suite.

2 Etape 2

Ecrire le programme lex qui va générer l'analyseur lexical qui reconnaît toutes les unités lexicales présentes dans un fichier ical.

Pour cela, procédez par étapes.

1. Commencez par écrire un analyseur lexical qui va reconnaître uniquement les mots clés et le séparateur d'options (n° 1 à 6, 8 à 10, 12 à 14, 19 à 24, 27 à 30) et qui ignore tout le reste. Si vous testez cette première version sur le fichier `exemple_complet`, vous devez obtenir le résultat 1. Il y a des affichages qui ne nous intéressent pas mais vous les ferez disparaître plus tard.
2. Ajoutez ensuite les reconnaissances des unités lexicales 7, 15, 16, 17, 18, 25, 26 et 31 avec affichage de la chaîne lue pour toutes, sauf 16 et 17. Sur le fichier `exemple2`, le résultat à obtenir est le résultat 2. Le même programme sur `exemple_complet` doit donner le résultat 3.
3. Ajoutez ce qu'il faut dans le programme lex (utilisez des états) afin qu'il ignore toutes les lignes se trouvant entre « BEGIN:VCALENDAR » et le premier « BEGIN:VEVENT ». Utilisez la même démarche pour faire en sorte qu'il récupère :
 - les dates et heures avec un T et un Z seulement si elles sont après « DTSTART: » ou « DTEND: » pour les événements uniques, ainsi qu'après « UNTIL= » dans la règle de répétition des événements répétitifs,
 - les dates et heures avec un T sans Z seulement si elles sont après « DTSTART;TZID=...: » ou « DTEND;TZID= » pour les événements avec répétitions,
 - les dates sans T sans Z et sans heure seulement si elles sont après « DTSTART;VALUE=DATE: » ou « DTEND;VALUE=DATE: » pour les événements sur une journée,
 - etc.pour au final pouvoir obtenir le résultat 4 sur le fichier `exemple_complet`.
4. Il restera ensuite à ajouter la reconnaissance des lieux, titres et descriptions des événements pour que l'analyseur appliqué au fichier `exemple_complet` donne le résultat final.

Remarque : il est important de récupérer les titres, lieux et descriptions mêmes s'ils sont vides pour la traduction en html plus tard.

Cette étape devra être rendue, au plus tard :

- lundi 20 février pour le groupe 3
- mercredi 22 février pour le groupe 6
- jeudi 23 février pour les groupes 2, 4 et 5
- vendredi 24 février pour le groupe 1

Seul le programme lex est à rendre ainsi que les instructions nécessaires à la création de l'exécutable.

Rappels : Inutile de fournir un exécutable car il ne fonctionnera pas. Si vous fournissez un makefile, celui-ci ne doit pas inclure l'exécution de votre programme sur un de vos fichiers tests ni la suppression des fichiers intermédiaires créés lors de la compilation (ici lex.yy.c). Afin que nous puissions tester vos programmes, merci de faire en sorte que ceux-ci puissent être compilés et exécutés sur les machines des salles d'enseignement.