جامعة دمشق

كلية الهندسة المعلوماتية

السنة الثالثة

2018/2019

مشروع عملي مادة اللغات الصورية

إعداد الطلاب:

محمد سامي العتش

محمد علاء خير الله

طارق آمنة

البنية العامة لتمثيل اللغة

لتمثيل أي لغة يجب تمثيل عناصرها الخمسة مهما كانت طريقة التمثيل المستخدمة (مجموعة الحلات – الأبجدية - الحالة البدائية – الحالات النهائية – تابع الانتقال).

في هذا المشروع تم الاعتماد على ملفات JSON كبنية معيارية لتوصيف لغة ما، حيث أن ملف الـ JSON يحتوي على العناصر التالية:

1. statesCount: يمثل عدد الحالات المضمنة ضمن اللغة.
2. alphabet: تحوي أحرف الأبجدية المضمنة جميعها.
3. finalStates: مصفوفة من العناصر التي تمثل مجموعة الحلات النهائية حيث يعبر عن كل حالة نهائية بشكل مصفوفة من عنصرين الأول هو رقم الحالة والثاني هو وصف للحالة

   [state number, “definition of this state ”]
4. transitions: وهي مصفوفة تحوي جميع الانتقالات المعرفة ضمن اللغة وحيث يمثل الإنتقال الواحد بمصفوفة تحوي ثلاث عناصر (الحالة المصدر – محرف الانتقال – الحالة القادمة)

   [source , ‘char’ , destination]
5. startState: رقم الحالة البدائية

```
{
   "startState": ....,
   "transitions": [...],
   "finalStates": [...],
   "alphabet": "....",
   "statesCount": ...
}
```

البنية العامة لتمثيل اللغة برمجيا

تم بناء class عام لتمثيل اللغة الصورية برمجيا "BaseDFA" وتم الاعتماد على بنى المعطيات التالية لتمثيل عناصر اللغة الخمسة وهي:

عدد الحالات	int statesCount
الحالة البدائية	int startState
الأبجدية	String alphabet
الحالات النهائية	Map<Integer, String> finalStates
الانتقالات	List<Map<Character, Integer>> transitions

توضيح بنية finalStates:

عبارة عن HashMap فيها الـ "key" هو رقم الحالة والـ "value" هي وصف الحالة (keyword, number, ...).

توضيح بنية transitions:

عبارة عن List of Map حيث أن كل عنصر في الـ List يقابل حالة وفيها Map تعرف كل الانتقالات عند هذه الحالة.

آلية فحص

يتم فحص الـ string التي ادخلها المستخدم بالمرور على المحارف محرفا محرفا حيث كل محرف يشمل انتقال من حالة لأخرى ويتم التفقد بالطريقة التالية:

1. إذا كان المحرف ليس من عناصر الأبجدية يعيد التابع القيمة (-1)
2. إذا كان المحرف من الأبجدية ولكن لا يوجد انتقال معرف عند الحالة الحالية من خلال هذا المحارف (حالة ميتة) فإن التابع يعيد القيمة (-2).
3. في الحالة النظامية فإن التابع سيعتمد على مصفوفة الانتقالات للانتقال من حالة إلى حالة وفق المحرف الحالي حتى يصل إلى آخر حالة ويعيد التابع رقمها.

تابع getDetails: هو تابع يتفقد القيمة التي يعيدها التابع check فإذا كانت -1 يطبع "Strange character in the statement!"

وإذا كانت -2 يطبع "Wrong statement!"

وإلا فإن التابع سيبحث عن رقم الحالة النهائية في مصفوفة الحالات النهائية فإن وجدها

يطبع تعرف هذه الحالة الذي تم إضافته للمصفوفة وإن لم يجدها فإنه يطبع " Wrong statement**!**"

ملاحظة:

تم استخدام مكتبة رسومية لتمثيل الاوتومات حيث يخرج التمثيل النهائي على شكل صورة png **باسم** automata Visualization.

تمثيل اوتومات للغة جافا: