



Universitat de Lleida

Determinització d'un autòmat finit

Oriol Alàs Cercós

29 d'Abril del 2019

Índex

1	Introducció	2
2	Implementació	2
3	Realització de les proves	2
4	Creació d'una nova prova	3

1 Introducció

Aquest document intenta ser una mena de guia per entendre com utilitzar l'aplicació per determinar un autòmat. Seguidament de la introducció, es parlarà sobre la implementació i, després, la realització de les diferents proves per verificar que el programa funcioni i com un usuari pot crear-ne una nova.

El programa s'ha desenvolupat en `python3`. No obstant això, hi ha un arxiu `Makefile` i un programa en `bash` per poder realitzar diverses proves. A l'hora de voler-lo fer anar, només cal executar el fitxer `./determinitzacio.py`. Per realitzar el projecte, s'ha creat un mòdul anomenat `automat.py` on s'implementa la classe *Autòmat* i la classe anidada *Estat*. Això s'ha servit per crear diferents autòmats i afegir-los-hi estats i transicions entre ells, a més de retornar el seu alfabet, entre altres opcions. A més a més, d'aquesta manera, es pot re-utilitzar la implementació per posteriors projectes.

Un cop executat el programa, es demanarà construir l'autòmat finit, que pot ser determinista o indeterminista. En cas que ja sigui determinista, ho avisarà per pantalla. En cas de no ser-ho, crearà un autòmat equivalent determinat i el mostrarà per pantalla.

2 Implementació

Per realitzar la determinització, s'ha seguit un algorisme descrit de tal forma:

1. Una crida a la funció `afegir_estat` passant per paràmetres els estats inicials. Aquesta funció segueix aquestes pautes:
 - 1.1 Crea un estat, en cas que no estigui creat en l'autòmat nou, on hi hagi els estats passats per paràmetres i aquells que són donats per una lambda-transició dels primers estats.
 - 1.2 Es busquen les transicions de tots els estats i s'ajunten en 1. D'aquesta manera, d'aquest nou estat (conjunt d'estats) amb la lletra 'a', anirà a un sol nou estat que representara un conjunt d'estats de l'autòmat finit indeterminista. Això es guardarà en un diccionari on la clau serà la lletra en qüestió i el valor serà el conjunt d'estats.
 - 1.3 Per cada lletra del diccionari de transicions, es tornarà a cridar `afegir_estat` i, seguidament, s'afegirà la transició desitjada.
2. En cas que hi hagués estats que no tinguessin totes les transicions, s'enviaran aquestes que faltaven a l'estat *conjunt buit*.

3 Realització de les proves

A l'hora de voler realitzar una prova creada pel desenvolupador, l'usuari podrà emprar des de la terminal i situada a la carpeta on hi ha el programa la comanda `make`. Llavors, l'usuari podrà escollir quin autòmat utilitzar per realitzar la prova. A la carpeta, n'hi constarà 6, que tenen trets diferenciadors entre ells:

- **automat1.txt**. Un autòmat on l'estat final no té transicions.
- **automat2.txt**. Hi ha una lambda-transició de l'estat 1 a l'estat 2. A part d'aquesta transició, l'estat 1 no té més transicions.
- **automat3.txt**. L'estat 1 té dos transicions amb la lletra b.
- **automat4.txt**. Semblant amb l'automat3 però amb més estats.
- **automat5.txt**. Incorpora lambda-transicions però no en els estats inicials.
- **automat6.txt**. Incorpora diversos lambda-transicions entre estats.

A més a més, en cas de voler realitzar una prova seguida de tots els fitxers, es pot emprar la comanda `make main`, on s'executarà seqüencialment tots els fitxers `.txt` i s'imprimirà per pantalla els resultats.

4 Creació d'una nova prova

La creació d'una nova prova es pot realitzar de dos vies diferents:

1. Mitjançant la execució del programa. D'aquesta manera, l'autòmat construït per teclat no es guardarà i en cas de voler tornar executar el programa amb el mateix autòmat, no hi haurà més opció que tornar-lo a construir manualment.
2. Creant un arxiu `.txt` que continguin les següents línies:
 - La primera ha de contenir els noms dels estats, que han de ser nombres naturals o el 0. No es poden repetir.
 - La segona línia hi ha d'haver els noms dels estats inicials.
 - La tercera línia hi ha d'haver els noms dels estats finals.
 - En les següents línies hi ha d'haver les transicions per cada estat. L'ordre va en funció de la primera línia. En cas que un estat no tingui cap transició, es deixa aquella línia sense res. En cas d'haver-hi una lambda-transició, es marca amb la lletra `λ`. No importa l'ordre a seguir de les transicions en un estat.
 - La última línia ha d'estar en blanc i significarà el final del text.

Un cop creat l'arxiu, només caldrà emprar la comanda `make` i enviar per teclat: `<nom de l'arxiu>.txt`