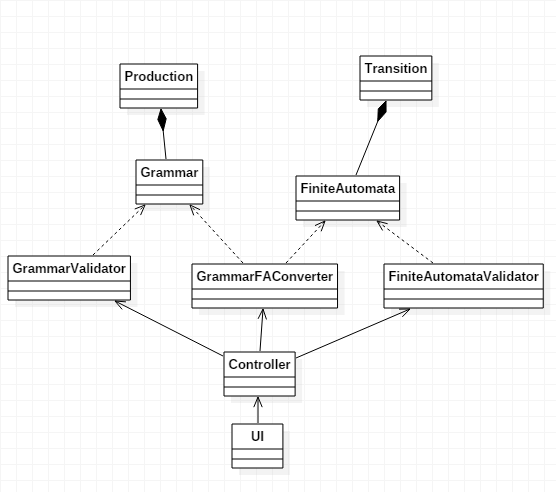
Laborator 2 – LFTC

Stefan Sebastian 235

Analiza cerintelor

Sa se scrie un program care citeste un automat finit sau o gramatica (din fisier sau de la tastatura), afiseaza elementele lor, verifica daca gramatica e regulara, construieste un automat finit echivalent cu o gramatica regulara, construieste o gramatica regulara echivalenta cu un automat finit.

Proiectare



Grammar, cu o lista de Production, si FiniteAutomata, cu o lista de Transition, sunt reprezentarile gramaticilor si a automatelor finite folosite de program. Acestea au fiecare cate un validator care verifica elementele folosite in definirea lor. GrammarFAConverter contine metode de conversie din automat finit in gramatica regulara si din gramatica regulara in automat finit. Controller mentine o lista de gramatici si automate finite iar UI afiseaza un meniu tip consola pentru interactiunea cu utilizatorul.

Detalii de implementare

Fiecare gramatica are un id, un simbol de start, o lista de terminali, o lista de nonterminali si o lista de productii. O productie contine un string reprezentand un nonterminal, leftSide, si o lista de liste de stringuri, rightSide. Fiecare lista reprezinta partea dreapta a unei productii care porneste din leftSide. Formatul in care se citeste o gramatica este urmatorul: pe prima linie un identificator, pe a doua linie lista de nonterminali separati prin virgula, pe a treia linie lista de terminali separati prin virgula, pe a patra linie simbolul de start si pe urmatoarele linii pana la intalnirea cuvantului ‘end’ o insiruire de productii. Fiecare productie are formatul A ->a,B|c|e,F.

Fiecare automat finit are un id, o lista de stari, un alfabet (lista de simboluri), o stare initiala, o lista de stari finale si o lista de tranzitii. O tranzitie are o stare initiala, o stare finala si o valoare. Formatul in care se citeste un automat finit este urmatorul: pe prima linie un identificator, pe urmatoare linie alfabetul (simboluri separe prin virgula), pe a treia linie o lista de stari separate prin virgula, pe a patra linie starea initiala, pe linia cinci o lista de stari finale iar pe urmatoarele linii pana la intalnirea cuvantului ‘end’ o lista de tranzitii. O tranzitie are formatul state1,state2,value.

Verificarea daca o gramatica este regulara incepe prin a identifica daca simbolul de start duce in epsilon. Apoi se parcurge fiecare productie. Pentru productiile in care partea dreapta are dimensiunea de 1 se verifica ca simbolul sa fie terminal sau (in cazul in care partea stanga e simbolul de start) epsilon. Pentru productiile de dimensiune 2 se verifica ca primul simbol sa fie terminal si al doilea nonterminal. Daca suntem in cazul in care simbolul de start duce in epsilon atunci verificam si fiecare simbol din partea dreapta sa fie diferit de simbolul de start.

Pentru conversia din gramatica regulara in automat finit am declarat un numar de stari egal cu numarul de nonterminali + 1. Terminalii corespund alfabetului, simbolul de start corespunde starii initiale. Pentru fiecare productie de forma A->aB am adaugat o tranzitie intre starea care ii corespunde lui A si starea care ii corespunde lui B cu valoarea a. Pentru fiecare productie de format A -> a am adaugat o tranzitie de la starea care ii corespunde lui A la starea finala cu a. Daca apare tranzitia S->epsilon am marcat starea initiala ca fiind si stare finala.

Pentru conversia din automat finit in gramatica regulara am creat o lista de nonterminali care corespunde listei de stari. Am identificat cazul in care starea initiala este finala si exista o tranzitie cu destinatie in acea stare. In acest caz am adaugat un nonterminal nou. Pentru fiecare tranzitie de forma q0->q1 cu valoarea a, am adaugat cate o productie de format A->aB. Pentru tranzitiile care au ca destinatie o stare finala am adaugat o productie de forma A->a. Am tratat separat cazul in care starea initiala este si stare finala, caz in care am adaugat productia S->epsilon, S reprezentand simbolul de start.