

Zadanie

Cieľom druhého projektu je vytvoriť skript, ktorý spracuje konečný automat v textovom súbore a podľa zadáných prepínačov vypíše pôvodný automat, ekvivalentný automat bez epsilon prechodov, alebo ekvivalentný deterministický automat bez nedostupných stavov.

Spracovanie parametrov

Parametre príkazového riadku sú spracovávané pomocou funkcie `argparse`. Táto funkcia priradí parametrom hodnoty, prípadne vypíše nápovedu. Keďže funkcia `argparse` automaticky vracia pri chybných zadaniach parametrov hodnotu 2 a pri zadaní parametru `help` hodnotu 1, je potrebné po spracovaní parametrov zachytiť výnimku a vrátiť hodnotu požadovanú zadáním projektu.

Spracovanie vstupného súboru

Vstupný súbor je spracovávaný konečným automatom. Ak tento automat skončí v nekoncovom stave, alebo nie je pre stav nájdený prechod, program končí s návratovou hodnotou 40. Týmto spracovaním je možné určiť aký komponent vstupného automatu je práve spracovávaný a je umožnené ukladať jednotlivé komponenty do odpovedajúcich dátových štruktúr. Pre odstránenie duplicit je použitý dátový typ množina.

Komentáre začínajúce znakom '#' sú preskakované pomocou samostatnej funkcie, ktorá sa posúva v súbore až pokým nenájde znak konca riadku, alebo súboru. Pokiaľ je znak '#' ako symbol vstupnej abecedy, alebo ako vstupný symbol pravidla, žiadne znaky sa nepreskakujú. Podobne sú ignorované aj biele znaky.

Po uložení všetkých komponentov automatu, je vykonaná sémantická kontrola zahrňujúca overenie, že vstupná abeceda nie je prázdna, množina koncových stavov je podmnožinou množiny stavov, pravidlo neobsahuje stav alebo symbol, ktorý sa nenachádza v množine stavov alebo symbolov a začiatkový stav je podmnožinou množiny stavov. Ak niektorá z týchto podmienok nie je splnená, program je ukončený s hodnotou 41.

Odstránenie epsilon prechodov

Pre odstránenie epsilon prechodov je najprv potrebné vytvoriť pre každý stav epsilon uzáver. Vytvorenie epsilon uzáverov je realizované pomocou prechádzania všetkých pravidiel automatu pre každý stav a ukladaním cieľových stavov, ktoré boli dosiahnuté pomocou epsilon prechodu.

Následne, sú prechádzané epsilon uzavery pre každý stav a vytvorené nové pravidlá neobsahujúce epsilon prechody. Ak je prienik (metóda `intersection`) množiny koncových stavov s epsilon uzáverom určitého stavu neprázdny, daný stav je pridaný do novej množiny koncových stavov.

Odstránenie nedeterminizmu

Determinizácia je implementovaná vytváraním spojených cieľových stavov s rovnakým vstupným symbolom a vstupným stavom. Následne je vytvorené nové pravidlo reprezentujúce prechod do nového stavu. Spojenie stavov je realizované pomocou samostatnej funkcie, ktorá lexikograficky zoradí názvy stavov a v prípade, že je stavov viac, oddelí ich znakom '_'. Ak je prienik (metóda `intersection`) množiny stavov vytvárajúcich nový stav a množiny koncových stavov automatu bez epsilon prechodov neprázdny, nový stav je pridaný do množiny koncových stavov.

Rozšírenie STR

Najprv vykonaná determinizácia automatu. Následne je reťazec prechádzaný cyklom a ak je na danej pozícii znak, ktorý nie je v abecede, tak je program ukončený s chybou 1. Ak nie je nájdený prechod pomocou aktuálneho symbolu, reťazec nie je reťazcom jazyka prijímaného zadánym konečným automatom. Reťazcom sa prechádza do posledného znaku, ak je po načítaní posledného znaku a vykonaní prechodu automat v nekoncovom stave, tak daný reťazec nie je reťazcom jazyka prijímaného zadánym konečným automatom.

Rozšírenie WSA

Najprv je vykonaná determinizácia automatu bez generovania nedostupných stavov. Odstránenie neukončujúcich stavov je docieľené prechádzaním automatu od koncových stavov tak, že nasledujúci stav je vstupný stav pravidla, narozdiel od priechodu od začiatkového stavu, kde nasledujúci stav bol cieľový stav pravidla. Týmto spôsobom sa vytvoria len ukončujúce stavy. Následne sú pridané pravidlá pre prechod do jediného neukončujúceho stavu `qFALSE` pre ktoré nebol nájdený vstupný symbol pravidla. Ak začiatkový stav nie je ukončujúci, je nahradený stavom `qFALSE`.