

1 Zadanie

Úlohou projektu bolo vytvoriť skript v programovacom jazyku Python verzie 3 pre spracovanie a determinizáciu konečného automatu. Skript pracuje so vstupným súborom, ktorý obsahuje textový zápis konečného automatu. Tento skript je schopný transformovať vstupný konečný automat na deterministický, alebo na konečný automat bez *epsilon* pravidiel, prípadne len vypísať konečný automat v predpísanom výstupe.

2 Návrh

Návrh skriptu je možné rozdeliť na dve hlavné podúlohy. Prvou úlohou je navrhnuť správne spracovávanie vstupného konečného automatu. Druhou je správne navrhnutie operácií nad spracovaným konečným automatom (odstránenie epsilon pravidiel, determinizácia). Obe časti sú popísané nižšie.

2.1 Spracovanie vstupného konečného automatu

Vstupný konečný automat je podrobený lexikálno-syntaxickej analýze. Konečný automat je prehľadávaný znak po znaku, ktoré sú kontrolované či je zápis vstupného konečného automatu správny. V prípade syntaxickej nesprávnosti vstupného konečného automatu dôjde k lexikálno-syntaxickej chybe. Pri spracovávaní konečného automatu sú ignorované všetky komentáre a prázdne miesta, ktoré sa nachádzajú vo vstupnom súbore.

Počas prehľadávania konečného automatu je vykonávané aj spracovanie a ukladanie stavov a pravidiel konečného automatu. Stavý a pravidlá sú podrobené lexikálnej analýze a následne sú ukladané pre ďalšie použitie. Po spracovaní celého konečného automatu sú dáta pripravené pre ďalšie operácie nad týmto konečným automatom.

2.2 Operácie nad konečným automatom

Najdôležitejšou časťou návrhu je správne navrhnutie operácií vykonávaných nad konečným automatom.

Prvou operáciou je vytvorenie epsilon uzáveru pre každý stav z množiny všetkých stavov daného konečného automatu. Výsledkom je množina všetkých stavov, do ktorých sa je možné dostať z daného stavu pomocou epsilon prechodu.

Druhou operáciou je odstránenie epsilon pravidiel, pričom sa využíva epsilon uzáver pre daný stav. Takto sú vytvorené nové pravidlá bez epsilon prechodov a rovnako nové koncové stavy.

Poslednou operáciou je samotná determinizácia. Úlohou determinizácie je nájdenie všetkých stavov, do ktorých sa je možné dostať pomocou rovnakého vstupného symbolu z konkrétneho stavu. Z množiny nájdených stavov je vytvorený zložený stav. Ak je spracovávaný zložený stav je spracovávaný aj každý z jeho pôvodných stavov. Po prejdení všetkých stavov sú vytvorené nové pravidlá a stavý konečného automatu.

3 Implementácia

Pri implementácii boli použité 4 triedy.

- Param - Parsovanie argumentov, čítanie zo vstupného súboru
- Parser - Spracovanie vstupného konečného automatu
- Rule - Datová reprezentácia pravidla konečného automatu
- Fsm - Odstránenie epsilon prechodov, determinizácia, zápis do výstupného súboru

Po spustení skriptu je vytvorený objekt *Param*, ktorý následne parsuje argumenty skriptu. V prípade, že je parsovanie úspešné, je možné pomocou tohto objektu získať hodnoty argumentov.

Nasleduje vytvorenie objektu *Parser.*, ktorý spracováva vstupný konečný automat. V tomto kroku dochádza ku kontrolovaniu chýb vstupného konečného automatu a ukladanie stavov a pravidiel do vnútornej dátovej reprezentácie. Po úspešnom spracovaní je vytvorený objekt *Fsm*, ktorý vykonáva operácie (napr. determinizácia) nad konečným automatom. Po vykonaní sú nové pravidlá a stavy konečného automatu uložené, prípadne nahradené a dôjde k normovanému výpisu konečného automatu do výstupného súboru. Skript je ukončený odpovedajúcim chybovým kódom.