

1 Zadání

Úkolem bylo vytvořit skript v jazyce Python 3 pro zpracování a případnou minimalizaci konečného automatu. Skript také dále ověřoval, že je automat dobře specifikovaný a provedl jeho normalizovaný zápis.

2 Implementace

Skript je implementován v jazyce Python 3, konkrétně ve verzi 3.6.0.

2.1 Zpracování argumentů

Zpracování argumentů zajišťuje funkce `parseArguments()`, která je volána jako první po spuštění skriptu. Tato funkce využívá knihovnu `argparse`, která je určena právě pro zpracování argumentů. Pro všechny argumenty platí, že mohou být zadány pouze jednou a argumenty `-f` a `-m` se nesmí kombinovat. Pokud funkce narazí během zpracovávání argumentů na nějakou chybu, vypíše patřičnou chybovou hlášku a ukončí skript se správným návratovým kódem.

2.2 Načtení vstupního souboru

K určení vstupního souboru, který obsahuje konečný automat, slouží přepínač `--input`. Pokud není tento přepínač zadán, zpracovává se `STDIN`. Soubor je otevřen pomocí metody `open()` s parametrem `r`, který slouží pouze pro čtení. Pokud zadáný soubor neexistuje, skript vypíše patřičnou chybovou hlášku a ukončí se s návratovým kódem 2.

2.3 Výstupní soubor

K určení souboru, do kterého se má výsledný konečný automat zapsat, slouží přepínač `--output`. Pokud není tento přepínač zadán, zpracovává se `STDOUT`. Soubor je otevřen pomocí metody `open()` s parametrem `w`, který slouží pouze pro zápis do souboru. Skript do výstupního souboru zapisuje přesně podle specifikace ze zadání. Všechny komentáře jsou ze vstupního souboru vynechány. Pokud zadáný výstupní soubor neexistuje, skript vypíše patřičnou chybovou hlášku a ukončí se s návratovým kódem 3.

2.4 Zpracování vstupního konečného automatu

K zpracování vstupního souboru s konečným automatem slouží funkce `parseFile()`. Celá analýza souboru je implementována pomocí konečného automatu, jeho stavy jsou ve třídě typu `Enums` názvem `States`. Pro načtení dalšího znaku tato funkce používá funkci `getNextToken()`, která vrátí dalšího znak ze vstupního souboru. Tato funkce také zajišťuje ignorování bílých znaků tam kde je to potřeba a také o ignoraci komentářů. Pokud během analýzy dojde k nějaké lexikální či syntaktické chybě, skript vypíše patřičnou chybovou hlášku a ukončí se s patřičným návratovým kódem podle typu chyby.

2.5 Ověření dobré specifikovanosti konečného automatu

Po zpracování vstupního souboru se musí ověřit, že zadáný konečný automat je dobře specifikovaný, aby jsme mohli případně provést minimalizaci. Toto ověření má na starost funkce `isWellSpecified()`. Tato funkce ověří všechny podmínky, které určují, že je zadáný konečný automat dobře specifikovaný. Pokud funkce zjistí, že se nejedná o dobře specifikovaný konečný automat, tak ukončí skript s patřičnou chybovou hlášku a návratovým kódem 62.

2.6 Minimalizace konečného automatu)

Pokud byl zadán přepínač `--minimize`, skript provede minimalizaci zadaného dobře specifikovaného konečného automatu. Minimalizace je provedena ve funkci `minimize()`, ve které je voláno několik dalších pomocných funkcí pro dílčí části algoritmu minimalizace. Algoritmus minimalizace je implementován přesně podle algoritmu uváděného v přednáškách předmětu IFJ.

2.7 Nalezení neukončujícího stavu

Skript také umožňuje pomocí přepínače `--find-non-finishing` nalézt neukončující stav dobře specifikovaného konečného automatu. Nalezení tohoto stavu má na starost funkce `findNonFinishing()`. Pokud má zadaný dobře specifikovaný konečný automat neukončující stav, funkce ho vypíše do výstupního souboru a ukončí skript. Pokud automat neukončující stav nemá, zapíše do výstupního souboru znak 0 a ukončí skript.

2.8 Chybové stavy

V případě, že skript narazí na chybový stav, okamžitě zavolá funkci `error()`, která na `STDERR` vypíše příslušnou chybovou hlášku a ukončí program s odpovídajícím návratovým kódem.

3 Závěr

Skript byl otestován pomocí základní sady testů na operačním systému CentOS 6.5 64bit, umístěný na školním serveru `merlin.fit.vutbr.cz`. Výstupní soubory byli porovnány pomocí programu `diff`. Základní testy byli poté rozšířeny a několik mých vlastních testů a opět testovány na stejném školním serveru.