Dokumentace úlohy MKA: Minimalizace konečného automatu v Python 3 do IPP 2015/2016

Jméno a příjmení: Peter Miklánek

Login: xmikla10

Zadání

Mým úkolem bylo vytvořit skript v jazyce Python 3 pro zpracování konečného automatu, validaci, jestli je dobře specifikovaný a případnou minimalizaci konečného automatu.

Implementace

Pro implementaci byl zvolen objektový návrh. V programu je implementována funkce `main(),` krerá je volána na posledním řádku programu. Tato metoda řídí celý běh programu a volá si třídy, které budou zmíněny dále.

Zpracováni argumentů

Zpracování argumentů příkazové řádky zajišťuje funkce `processArguments()`, která se stará o jejich načtení, validaci, prácu se soubory a zpřístupnění zbytku programu při jeho běhu.

Tato funkce provádí také kontrolu opakované deklarace, nedovolené kombinace vstupních argumentů, nesprávného kódovaní nebo kontrolu nevalidních hodnot.

Lexikální a sémantická pravidla

Pro správné spracováni vstupního automatu aby vyhovoval popsaným lexikálním i syntaktickým pravidlám byla vytvořena funkce `processInput()` , která v případe automatu nesplňujícího pravidla ukončí program s návratovým kódem 60.

Sémantická kontrola

Pro ovjeření, zda vstupní automat vyhovuje daným sémantickým pravidlám byla vytvořena trída `Semantic()`, která v sobě zarhňuje víc funkcí, které budou zmínené níže.

V této tříde sa oveřuje napr. zda vstupní abeceda není prázdná, nebo jestli počíteční stav nepatří do množíny stavů.

Ak je to pravda, funkce ukončí program s návratovým kódem 61.

V této tříde je dále implementována funkce pro minimalizaci konečného automatu.

Minimalizace konečného automatu

Pro minimalizaci konečného automatu byla vytvořena funkce `minimalizacia()`, ktorej algoritmus byl převzán z IFJ kurzu.

Stavy jsou rozdeleny do dvou skupin : na konečné stavy a ti ostatní. Pak všetchny možné pravidlá pro jeden přechodný symbol jsou použity na stavy v skupine, co vede k nové sade výsledných stavů. Pokud nejsou všechny stavy z této sady patřící do jedné skupiny, stavy jsou rozděleny do nových skupin podle rozdelení provedenej sady výsledných stavů.

Výsledné skupiny repretentují každý stav v minimalizovaném konečném automate.

Bonusová rozšíření

V skriptu sou také implementována 2 rozšíření, a to WHT a MWS. Pro implementaci rozšíření WHT nebyla zvolena vhodná strategie a tak toto rozšírení nemusí pracovan tak, jak má.

Zpracování chyb

Pro zpracování chyb nebyla implementována žádná metoda, ale každá existující funkce při nalezení chyby sama ukončí program s odpovídajícím návratovým kódem.

Závěr

Skript byl řádně odtestován sadou testů přiložených k zadání projektu a s použitím vlastních testů. Testování jsem prováděl na operačním systému GNU/Linux Ubuntu a také na referenčním školním serveru Merlin.