

Dokumentace úlohy DKA: Determinizace konečného automatu v PHP 5 do IPP 2015/2016

Jméno a příjmení: Petr Flajšingr

Login: xflajs00

Zadání úlohy

Úkolem projektu bylo vytvoření skriptu v jazyce PHP. Tento skript načte vstupní data ze souboru nebo `stdin`, zpracuje je, zkontroluje jejich validitu jakožto obecný konečný automat a následně provede buď přepis do normalizované formy, odstranění epsilon přechodů nebo jeho determinizaci.

Postup řešení

Kontrola parametrů

Kontrola parametrů skriptu je provede pomocí funkce `getopt`. V případě použití nesprávných parametrů, nepovolené kombinace parametrů nebo vícenásobného zadání stejného parametru je skript ukončen chybou. Pokud jsou všechny parametry validní, skript nastaví proměnné, které slouží jako přepínače pro dané funkce. Při použití `getopt` jsem narazil na několik problému, které jsem musel řešit dalšími podmínkami.

Zpracování vstupních dat

Vstupní data jsou zpracována pomocí konečného automatu. Kontrola a zpracování dat probíhá po jednom znaku. Pro čtení dat v kódování UTF-8 jsem si vytvořil pomocnou funkci `readUTFChar`. V případě zadání parametru `--case-insensitive` funkce převádí velké znaky na malé. Další pomocnou funkcí je `skipWhite`. Tato funkce vynechává všechny bílé znaky, nebo komentáře, které se mohou na vstupu objevit.

Při zpracování pravidel, počátečního stavu a finálních stavů je prováděna i jejich validita, tzn. existence stavů a znaků v již nadefinovaných množinách (sémantická analýza). Dále je zároveň prováděna syntaktická analýza, tzn. validita konečného automatu.

Uložení dat

Pro ukládání dat konečného automatu je využito několik proměnných. Abeceda, množina stavů a finální stavy jsou uloženy v polích, počáteční stav jako jednoduchá proměnná a pro množinu pravidel jsem využil vlastní třídu `rule`. Prvky této třídy jsou `startState` – vstupní stav pravidla, `symbol` – vstupní znak a `endState` – výstupní stav. Využil jsem pole pro uložení těchto pravidel.

Determinizace

Při provádění determinizace je nejdříve volána funkce `removeEps`, která ostraňuje všechny epsilon pravidla. Samotná implementace determinizace je prováděna podle algoritmu uvedeném v přednáškách IFJ. Funkce prochází všechny dostupné stavy. Dále zjišťuje pro každý znak vstupní abecedy možné cílové stavy. Z těchto stavů se poté generuje nový stav.

Doplňující informace

Pro přehlednost je skript rozdělen do několika souborů. Hlavním souborem je `dka.php`. Další jsou `param.php` obsahující zpracování parametrů, `fsm.php`, který zpracovává vstupní a výstupní data, `removeeps.php` pro odstranění epsilon přechodů a `deter.php` pro funkci determinizace.

Závěr

S jazykem PHP jsem pracoval poprvé, ale práce s ním mi nedělala problém. Pro psaní skriptu jsem využíval editor Sublime Text 3 a testoval jsem ho na OS Linux Mint a serveru merlin.

Samotný algoritmus pro determinizaci není příliš složitý, ovšem setkal jsem se při jeho implementaci s nějakými obtížemi.