Dokumentace

Konstrukce LL tabulky

1.1 Návrh

Aplikace je vyvíjena v Javě. Nevýhodou tohoto řešení je, že Java je interpretovaný jazyk a dále, že musí být instalováno speciální prostřední pro spuštění dané aplikace. Naopak výhodou je, že aplikace se snadněji může v budoucnu převést na aplikaci, která bude v reálném čase fungovat na www stránkách (formou Java apletů). Projekt byl vyvíjen v NetBeans IDE v5.0 a vyšším. Aktuální verze požadovaného virtuálního stroje Java Runtime Environment 6 i použité vývojářské nástroje jsou zdarma k dispozici na http://java.sun.com a http://www.netbeans.org.

1.2 Jednotlivé třídy v návrhu

1.2.1 Class `ConstructionState` – umožňuje dělat jednotlivé kroky při konstrukci množin *Empty*, *First*, *Follow* a *Predict* a uchovává současný stav konstrukce.

1.2.2 Class `Main` – základní třída pro spuštění aplikace.

1.2.3 Class `MainForm` – definuje uživatelské rozhraní a reakce na různé události. Dále je součástí této třídy simulace čtení jednotlivých vět. Text generovaný metodami z této třídy je zobrazován uživateli.

1.2.4 Class `MyDialogAbout` – tato třída generuje dialog obsahující informace o programu atd.

1.2.5 Class `Rule` – tato třída reprezentuje pravidla bezkontextové gramatiky. Umí z nich vytvořit speciální množiny terminálních a nonterminálních symbolů. Nonterminály jsou reprezentovány slovy, která začínají velkým písmenem. Ostatní slova jsou chápany jako terminály. Mezery jsou oddělovače těchto symbolů (slov). Dále třída definuje metody, které počítají množinu *Predict* z těchto pravidel.

1.2.6 Class `Symbol` – tato abstraktní třída reprezentuje všechny symboly gramatiky. Dále jsou z ní zděděny třídy pro reprezentaci terminálů (**`class Terminal`**), neterminálů (**`class Nonterminal`**) a prázdného řetězce (**`class Epsilon`**). Všechny tyto třídy musí definovat metody pro konstrukci množin *Empty*, *First* a *Follow* pro konkrétní symbol.

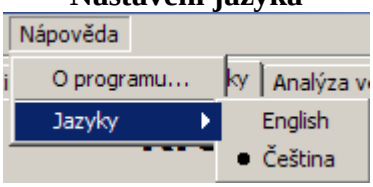
1.2.7 Class `TermNontermString` – objekty této třídy reprezentují řetězce jazyka složené z několika instancí třídy *Symbol* (resp. z ní odvozených tříd). Definování této specifické třídy je klíčové pro výpočty množin *Empty*, *First* a *Follow* pro řetězce symbolů. Například výpočet množiny *Empty* je snadný – pouze stačí zjistit, zda-li množina *Empty* všech obsažených symbolů je množina obsahující prázdný řetězec. Na podobném jednoduchém principu probíhá i výpočet dalších množin. Tato třída je využita pro uchování řetězců na pravých stranách pravidel bezkontextového tvaru.

2 Ovládání programu

2.1 Volba jazyka

Aplikace je realizována v českém a anglickém jazyce. Pro nastavení jazyka zvolte menu *Nápověda* → *Jazyky* a vyberte Vámi požadovaný jazyk. Jazyk je možno tedy měnit přímo za běhu aplikace.

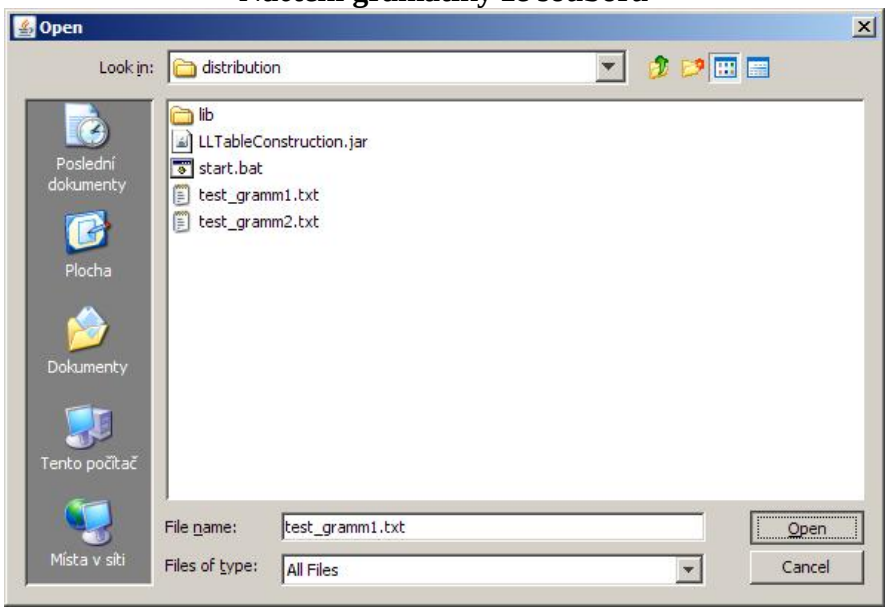
Nastavení jazyka



2.2 Načítání a ukládání gramatiky

Gramatiku lze načíst ze souboru pomocí nabídky: *Gramatika* → *Načíst gramatiku*. Po té může být vybrán XML soubor, který obsahuje popis dané gramatiky.

Načtení gramatiky ze souboru

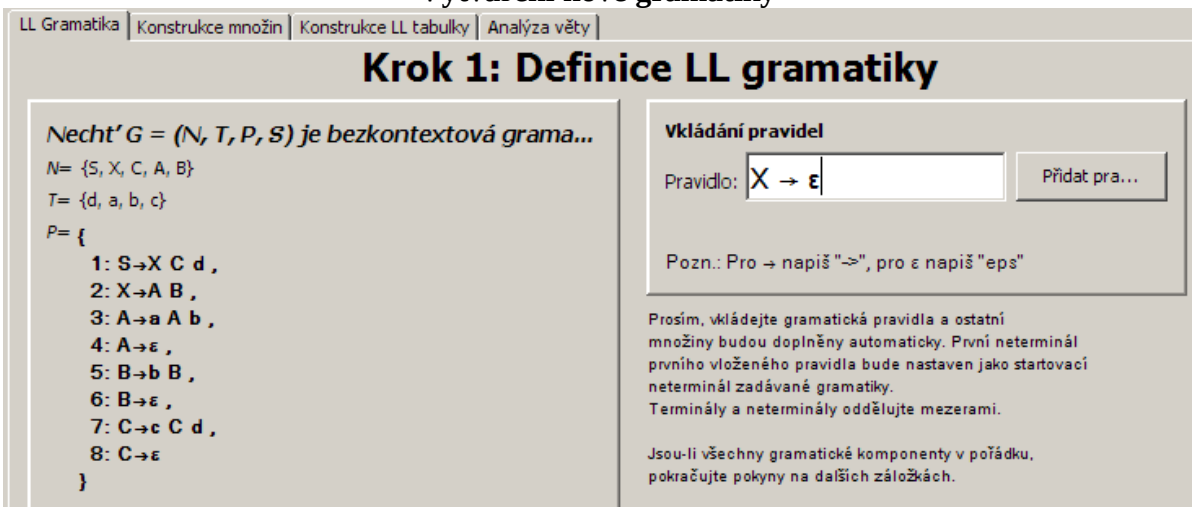


Obdobným způsobem pomocí menu *Gramatika* → *Uložit gramatiku* může být uživatelem vytvořená gramatika uložena do souboru. Pomocí nabídky *Gramatika* → *Vymazat pravidla* jsou vymazána všech vytvořená pravidla ze zatím zadané gramatiky.

3 Vytvoření nové gramatiky

Klikněte na záložku *LL-gramatika* na hlavním panelu. Do příslušného políčka vkládejte jednotlivá pravidla gramatiky, přičemž pravá strana pravidla musí obsahovat jeden neterminální symbol, levá strana řetězec neterminálních a terminálních symbolů oddělené mezerami. Počáteční neterminál je vždy první zadaný neterminál (tj. levá strana prvního zadaného validního pravidla). Slova začínající velkým písmenem a bez mezer jsou automaticky chápána jako neterminální symboly, ostatní slova bez mezer jsou chápany jako terminální symboly. Pro vytvoření prázdného řetězce napište "eps", přičemž se tento výraz automaticky přepíše na znak řecké abecedy ε. Mezi levou a pravou stranou pravidla napište šipku jako sekvenci symbolů "->" (ta je automaticky nahrazena opět jediným symbolem →).

Vytváření nové gramatiky

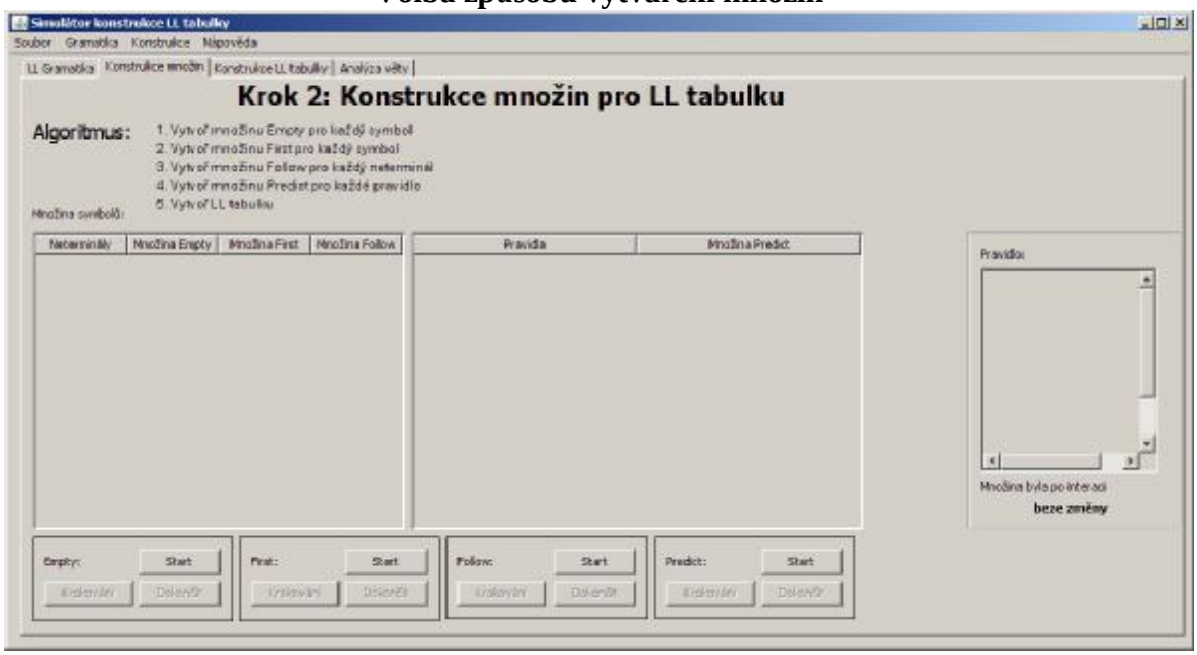


Kliknutím na tlačítko "Testovací gramatika" můžete použít předchystanou LL gramatiku určenou pro testování. Kliknutím na tlačítko "Vymaž gramatiku" je vytvořeno prázdné formulářové okno nachystané pro vytvoření nové gramatiky.

4 Konstrukce množin

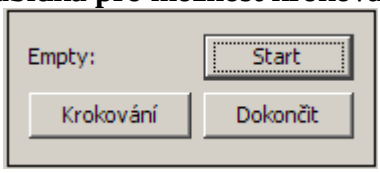
Pro vyplnění LL-tabulky je nejdříve potřeba vypočítat podle zadaných pravidel množiny *Empty*, First, *Follow* a *Predict* v tomto pořadí. Obecně však lze libovolný z těchto kroků vynechat a v případě potřeby bude daná množina automaticky dopočítána/přepočítána, což umožňuje využívat například pouze záložku *Analýza věty* bez předchozího manuálně ovládaného vytvoření LL tabulky.

Volba způsobu vytváření množin



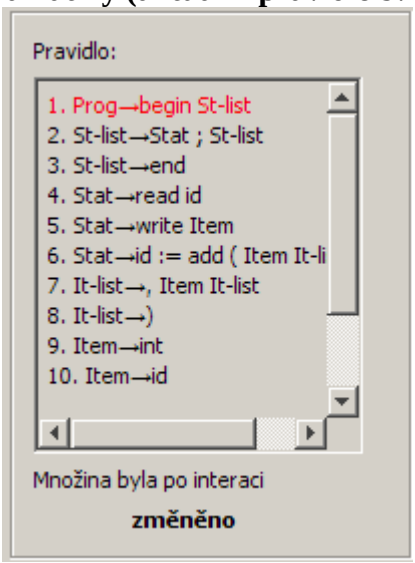
Konstrukci každé z těchto množin můžeme provést buď automaticky naráz a nebo ji můžeme postupně krokovat, aby bylo možné průběžně sledovat, jak se v závislosti na zadaných pravidlech gramatiky dané množiny vytvářejí. Pro vytvoření dané množiny je potřeba nejprve kliknout na tlačítko "*Start*". Následně může uživatel buď klikat na tlačítko "*Krokování*", přičemž v pravé části obrazovky se červeně vysvítí aktuální pravidlo, pro které se výpočet provádí. Nově přidané symboly do dané množiny se vysvítí červeně.

Nabídka pro možnost krokování



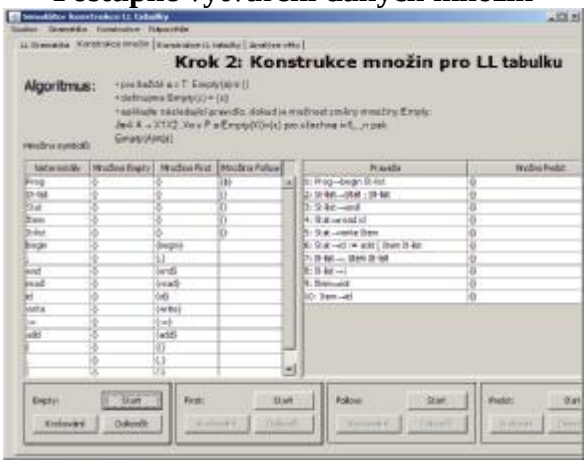
Výpočet množin *Empty*, *First* a *Follow* může být ukončen až v případě, pokud byla zpracována všechna pravidla a žádná množina už nebyla modifikována. Toto je indikováno příznakem *změněno*/*beze změny*, který je zobrazen pod seznamem pravidel.

Pravidla gramatiky (aktuální pravidlo svítí červeně)



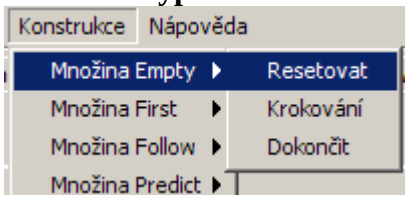
Kliknutím na tlačítko "*Dokončit*" se dopočítají dané množiny automaticky bez didaktického krokování.

Postupné vytváření daných množin



Pokud uživatel chce nechat proběhnout výpočet dané množiny znovu (například pro množinu *Empty*) může v nabídce *Konstrukce* → *Empty* → *Restartovat* odstranit všechny symboly z dané množiny a poté ji opětovně vytvořit. Analogicky je možné toto *Restartování* provést pro ostatní množiny.

Restartování výpočtu dané množiny



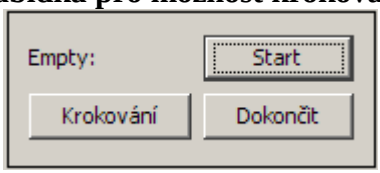
Jakmile máme vytvořenu množinu *Predict*, může být postupně vyplněna LL-tabulka. Výchozí stav tabulky je uveden na následujícím obrázku.

Volba způsobu vytváření LL-tabulky

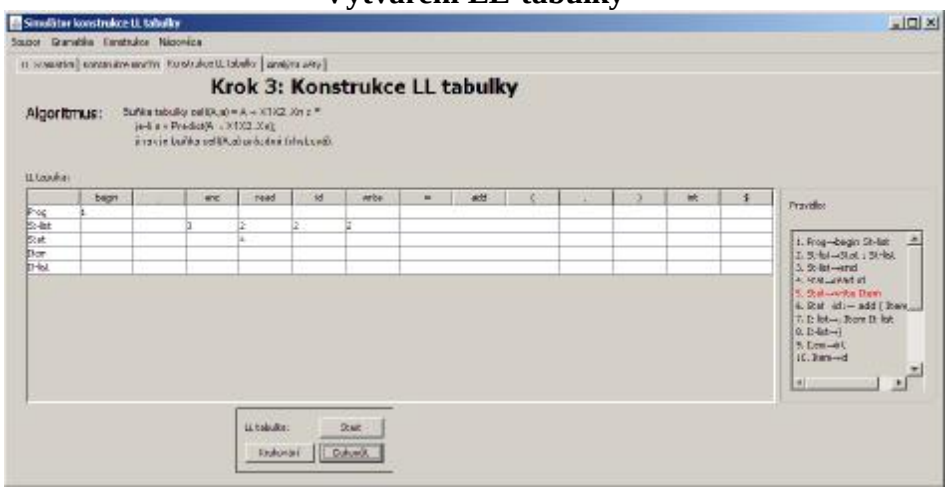
5 Konstrukce LL-tabulky

Konstrukci LL-tabulky můžeme provést buď automaticky a najednou nebo ji opět můžeme postupně krokovat, aby bylo možné průběžně sledovat, jak se tabulka postupně vytváří užitím vypočítaných množin *Predict*. Pro vytvoření LL-tabulky je potřeba nejprve kliknout na tlačítko *Start*. Následně může uživatel klikat na tlačítko *Krokování* a v pravé části obrazovky se červeně vysvítí aktuální množina *Predict*, pomocí níž se tabulka vyplňuje. Kliknutím na tlačítko *Dokončit* se tabulka dopočítá automaticky bez didaktického krokování.

Nabídka pro možnost krokování



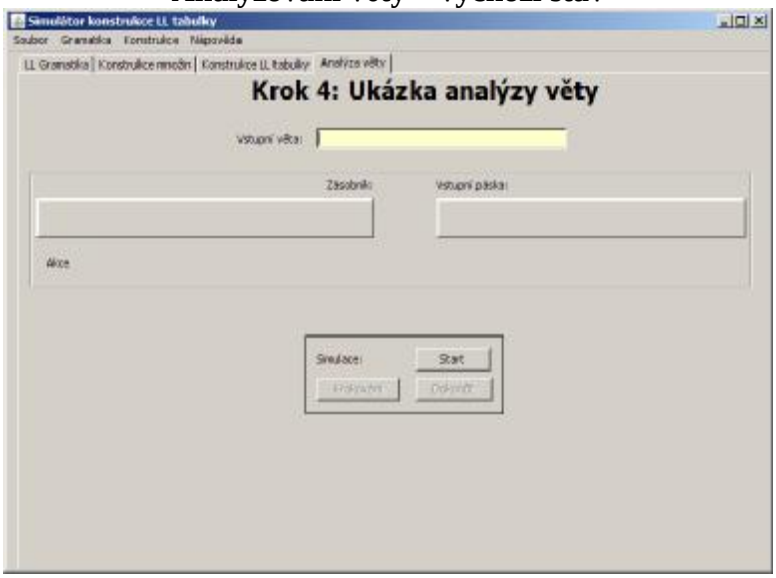
Vytváření LL-tabulky



6 Analýza věty

Jakmile máme vytvořenu LL-tabulku, můžeme pomocí tabulky odsimulovat prediktivní syntaktickou analýzu pro uživatelem zadanou větu. Výchozí stav syntaktického analyzátoru je uveden na následujícím obrázku.

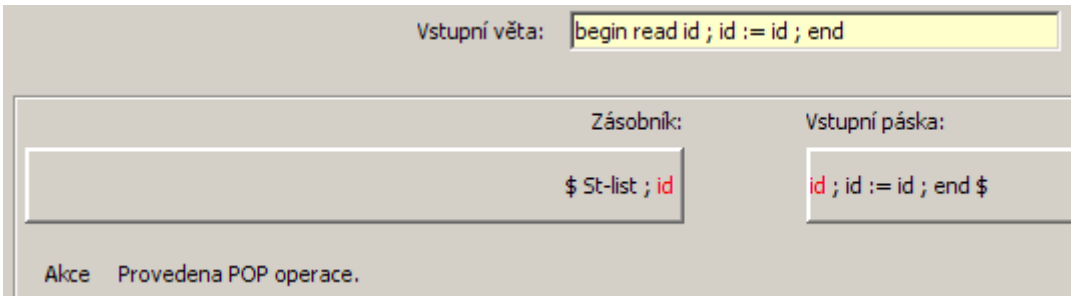
Analyzování věty – výchozí stav



Do políčka *Vstupní věta* uživatel zadá větu, pro kterou chce provést syntaktickou analýzu. **Jednotlivé symboly musí být odděleny mezerou**. Syntaktickou analýzu pro danou větu může provést buď automaticky naráz a nebo ji může postupně krokovat, aby bylo možné průběžně sledovat jednotlivé kroky. V každém kroku tak může sledovat hodnoty prvků uložených na zásobníku a dosud nezpracovanou část vstupního řetězce. Jedním krokem se rozumí provedení porovnání aktuálního vstupního symbolu se symbolem na vrcholu zásobníku a nebo expanzi pomocí jistého pravidla:

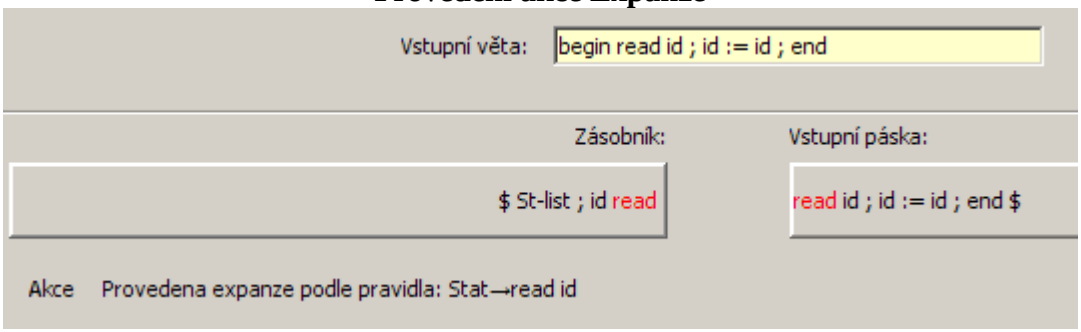
- *Porovnání* (*POP*) – Porovnání se provádí v případě, že na vrcholu zásobníku je obsažen symbol reprezentující terminál. Dojde k porovnání tohoto terminálu na vrcholu zásobníku se symbolem na vstupu. Pokud jsou symboly shodné, dojde k odstranění terminálu na vrcholu zásobníku a k přečtení aktuálního symbolu ze vstupu. Pokud symboly shodné nejsou, nastala chyba syntaktické analýzy.

Provedení akce POP



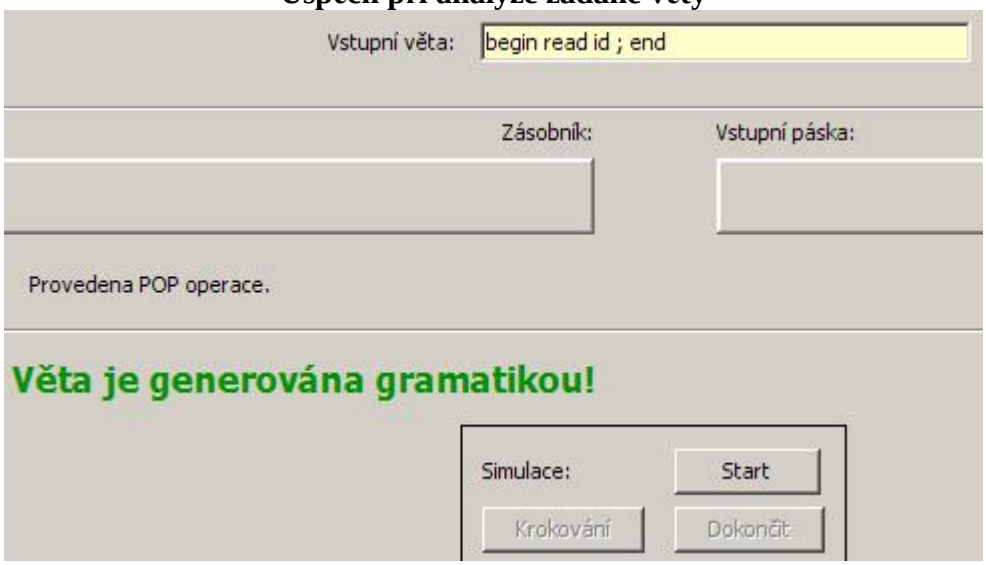
- Expanze – Expanze se provádí v případě, že na vrcholu zásobníku je obsažen symbol reprezentující neterminál. Podle hodnoty tohoto neterminálu a hodnoty vstupního symbolu je vybráno pravidlo z LL-tabulky. Pokud v LL-tabulce na této pozici žádné pravidlo není, nastala chyba syntaktické analýzy. Expanze je potom provedena tak, že symbol reprezentující nonterminál z vrcholu zásobníku je nahrazen reverzovanou pravou stranou vybraného pravidla.

Provedení akce Expanze



Syntaktická analýza může proběhnout úspěšně (daná věta je gramatikou generována) nebo neúspěšně (daná věta není gramatikou generována). Syntaktická analýza končí úspěšně, pokud se na vrcholu zásobníku nachází symbol $ (označuje dno zásobníku) a pokud celá věta zadaná na vstupu byla úspěšně přečtena. Syntaktická analýza končí neúspěšně, pokud během ní nastala chyba při expanzi a nebo při operaci POP (viz výše)

Úspěch při analýze zadané věty



Neúspěch syntaktické analýzy