Semestrální práce z KIV/FJP

Tvorba dat pro program IMiGEr

Vít Mazín, Martin Matas
A18N0096P, A18N0095P
mazinv@students.zcu.cz, martinm@students.zcu.cz

20. 11. 2018

Obsah

1 Zadání

Tématem semestrální práce bude převedení SLR gramatik do formátu JSON, který je zobrazen ve vizualizačním nástroji IMiGEr vytvořeným na Katedře informační a výpočetní techniky.

2 Analýza

2.1 Způsob přečtení SLR gramatik

Během analýzy výstupu programu Bison bylo zjištěno, že Bison umožňuje danou gramatiku převést do formátu **DOT** (graph description language). Tento formát je následně možné vizualizovat nástrojem Graphviz. Tímto lze výstupní automat zobrazit jako graf. Tento formát však není podporován nástrojem IMiGEr, proto bylo nutné najít cestu jak formát převést do požadovaného tvaru. Původní myšlenkou byla tvorba vlastního parseru. Při bližší analýze formátu jsme objevili existující knihovnu **digraph-parser** pro jazyk Java, která umožňuje uložený graf přečíst a vytvořit objekty z jeho vrcholů a hran.

2.1.1 Formát DOT

V následující ukázce je vidět struktura formátu DOT.

```
graph graphname {
    // This attribute applies to the graph itself
    size="1,1";
    // The label attribute can be used to change the label of a
       node
    a [label="Foo"];
    // Here, the node shape is changed.
    b [shape=box];
    // These edges both have different line properties
    a -- b -- c [color=blue];
    b -- d [style=dotted];
    // [style=invis] hides a node.
  }
```

2.2 Způsob uložení gramatiky do formátu JSON

Požadovaný výstupní formát byl fomrát JSON. Pro převedení objektů jazyka Java do formátu JSON byla zvolena knihovna **Jackson**. Přesná podoba jak má být graf reprezentován ve formátu JSON je přiložena v repozitáři programu IMiGEr na portále **GitHub**.

3 Programová dokumentace

3.1 Úvod

Pro implementaci nástroje jsme zvolili programovací jazyk Java. Byly použity knihovny **digraph-parser** a **Jackson**. Aplikace sestává z 5 následujících balíků:

cz.zcu.kiv.fjp
: Obsahuje hlavní třídu aplikace, která se stará o její správné spuštění a volá programové API.

cz.zcu.kiv.fjp.core
: API aplikace, které volá metody z balíků parser a serializer.

cz.zcu.kiv.fjp.core.entities
: Jednotlivé entity, které vnitřně reprezentují načtený graf, např. vrchol, hrana, atributy.

cz.zcu.kiv.fjp.core.parser
: Balík obsahuje parser umožňující zpracovat jak SLR gramatiku, tak i obecný a vnitřně je reprzentovat s využitím entit.

cz.zcu.kiv.fjp.core.serializer Slouží k převodu vnitřní reprezentace grafu do formátu JSON, který byl specifikován výše.

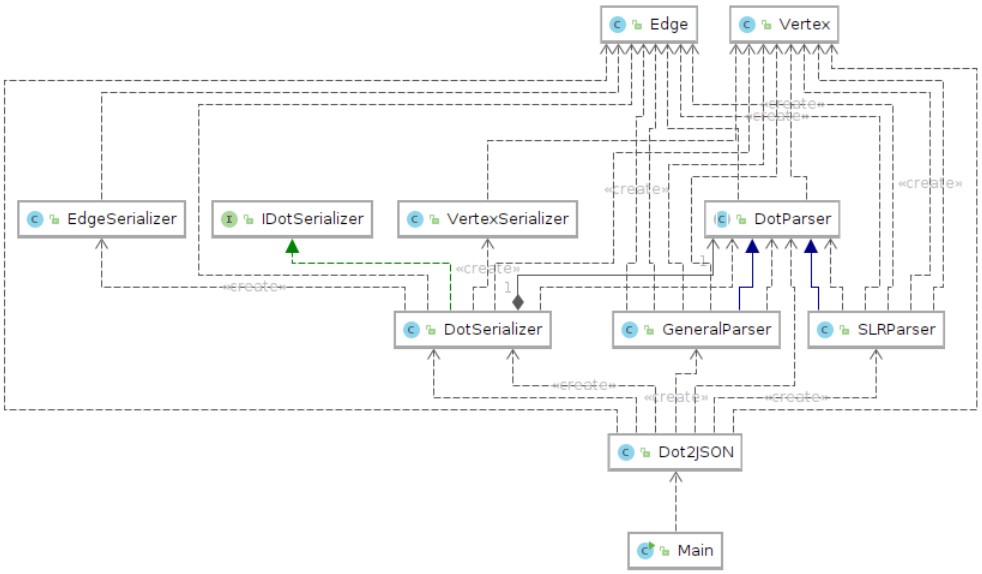
3.2 Běh programu

Program nejprve v hlavní třídě `Main` zpracuje argumenty programu a poté podle nich volá API, jež ja naprogramováno v třídě `core.Dot2JSON`.

Následně se zpracuje vstupní .dot soubor příslušným parserem z balíku `core.parser`, který dědí abstraktní třídu `DotParser`. Pro obecné grafy je to `GeneralParser` a pro grafy vytvořené ze SLR gramatik je to `SLRParser`. Výstupem jsou kolekce vrcholů a hran grafu.

Dále jsou kolekce vrcholů a grafů serializovány příslušnoým serializérem z balíku `core.serializer`, který implementuje rozhraní `IDotSerializer`. Konkrétně se jedná o třídu `DotSerializer`. Tato třída uloží data do výstupního souboru formátu JSON.

Na následujícím UML diagramu 3.1 je znázorněna závislost tříd programu. Diagram je ochuzen o některé třídy z balíku `core.entities`, které specifikují vlastnosti vrcholů a hran. Tyto třídy byly vytvořeny pro potřeby serializace.



Obrázek 3.1: UML diagram programu (zjednodušeno)

4 Uživatelská dokumentace

4.1 Spuštění programu

Program přeložený jako .jar archiv lze spustit ve dvou režimech. V prvním je zadán pouze název a cesta vstupního .dot souboru a název s cestou výstupního .json souboru. V tomto režimu program převádí obecné grafy do požadovaného výstupního formátu. Program v prvním režimu lze spustit například takto:

```
java -jar SLR-visualization-for-IMiGEr.jar input.dot output.json
```

V druhém režimu program převádí do požadovaného výstupu grafy, které vytvoří program Bison. Ten ukládá do formátu .dot automat dané SLR gramatiky. Jako argumenty je nutné zadat přepínač `-slr` a poté zadat vstupní .dot soubor a výstupní .json soubor. Program v druhém režimu lze spustit například takto:

```
java -jar SLR-visualization-for-IMiGEr.jar input.dot output.json
```

Pokud nejsou zadány požadované argumenty, program napoví, že ho lze spustit s přepínačem `-h`, který zobrazí nápovědu jak program korektně spustit.

5 Závěr

Domníváme se, že naše zadání bylo splněno. Program korektně převádí grafy, jež jsou výstupem programu Bison a navíc také dokáže převést libovolný graf, který je ve formátu DOT do požadovaného výstupního JSON formátu. Program lze použít samostatně nebo jako knihovnu, jež ze vstupního souboru poskytne informace o vrcholech a hranách grafu.