Stromové štruktúry v logike

Šimon Horvát,
Vedúci práce: Doc. RNDr. Stanislav Krajči, PhD.

Ústav informatiky, Prírodovedecká fakulta, UPJŠ Košice

Abstrakt

Cieľom tejto práce je navrhnúť a implementovať v jazyku Java vhodnú dátovú štruktúru schopnú uchovať a pracovať so stromovými štruktúrami vyskytujúcimi sa v symbolickej logike. Výsledná štruktúra je schopná načítať textový vstup - výraz rôzneho charakteru,
a v závislosti od zvoleného typu stromu transformovať reťazec do stromovej štruktúry. Nad každým načítaným znakom je vykonávaná analýza, na základe ktorej je z databázy každému stromovému uzlu priradená informácia rôzneho charakteru.
Aplikácia následne poskytne kompletnú analýzu a ohodnotenie stromu na základe zvolenej stromovej štruktúry.

Úvod

Stromy sú v informatike jednou z najdôležitejších dátových štruktúr. Táto dátová štruktúra je vhodná na ukladanie a spracovanie dát hierarchickej povahy.
Aj v tejto práci sa zaoberáme troma stromovými štruktúrami, ktorých hierarchickosť nie je na prvý pohľad zjavná.

Výraz

Prvou štruktúrou je trieda výraz, ktorá stromovo interpretuje aritmetický, logický, či iný typ výrazu.
Inicializácia výrazu prebieha načítaním textového vstupu v TeX-ovskej syntaxi.
Výsledkom nie je len ohodnotenie zadaného výrazu, ale aj kompletná analýza vlastností zadaného vstupu.
Vlastnosti skúmaného výrazu sú graficky znázornené do výsledného stromu, ktorý je generovaný len na základe vstupného reťazca .
Táto trieda je zároveň základným stavebným prvkom ďalších, zložitejších stromových štruktúr.

Ukážka

- **vstup**

e: \sum_{i=1}^{4} (5*i)+\sqrt[3]{64} *//aritmetický výraz* $\sum_{i=1}^{4}(5*i)+\sqrt[3]{64}$ *v TeX-u*

- **výstup**

preorder e: +(\sum_(1,4,*(5,i)),\sqrt(4,64))
Vyraz e patri do Exp: true *//Exp je množina správnych výrazov – kontrola korektnosti výrazov*
out e: Real
bve vyrazu: [i]
fve vyrazu: []

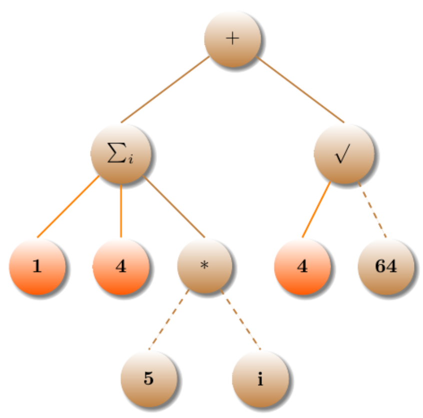
Ohodnotenie vyrazu e: 54

//informácie o jednotlivých symbolov v e
Var: i, Real, E(i)=2.0, cast
Sym: +, ari:2, ins = <Real,Real>, out: Real, Funkciovy symbol
Sym: \sqrt, ari:2, ins = <Integer,Real>, out: Real, Funkciovy symbol
Sym: 64, ari:0, ins = <> , out: Real, Konstantovy symbol, cast
Sym: 4, ari:0, ins = <> , out: Integer, Konstantovy symbol
Sym: *, ari:2, ins = <Real,Real>, out: Real, Funkciovy symbol
...

//vstup do TeX-u

```
[Real,thick] \node [Real, circular drop shadow] {$\textbf{$+$}$}
  child {[Real,thick] node [Real, circular drop shadow] {$\textbf{$\sum_{i}$}$}
   child {[Integer,thick] node [Integer, circular drop shadow] {$\textbf{1}$}}
   child {[Integer,thick] node [Integer, circular drop shadow] {$\textbf{4}$}}
   child {[Real,thick] node [Real, circular drop shadow] {$\textbf{$*$}$}
    child {[Real,thick,dashed] node [Real, circular drop shadow] {$\textbf{5}$}}
...
```

//výstup z TeX-u

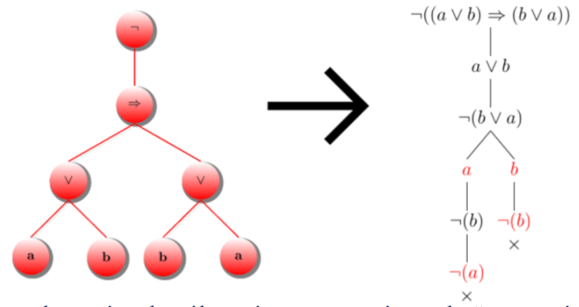


Sémantické tablo

Ďalšou zaujímavou stromovou štruktúrou, ktorou sa zaoberáme je sémantické tablo.
Ide o jednoduchý a univerzálny teoretický prostriedok k overeniu pravdivosti formúl na sémantickej úrovni.
Metóda sémantických tabiel je založená na systematickom postupe transformácie výrokovej formuly do tvaru disjunktívnej normálnej formy (DNF),
ktorý má jednoduché podmienky pre kontradikčnosť alebo splniteľnosť.

Ukážka
Majme formulu $\varphi = (a \vee b) \Rightarrow (b \vee a)$, ktorej tautologickosť chceme overiť pomocou tabla.
Celý proces transformácie vyzera nasledovne:

- zadáme **vstup**: `(a \lor b) \Rightarrow (b \lor a)`
- pomocou triedy Výraz vygenerujeme strom formuly $\neg\varphi$
- **výstupom** je teda zo stromu generované tablo:



//Vetvy, ktoré obsahujú komplementárne literály nazývame uzavreté,- pre lepšiu orientáciu sú komplementárne literály generované červenou farbou.
Z vlastnosti sémantických tabiel, vyplýva že zadaná formula je **tautológia**.

Dôkaz

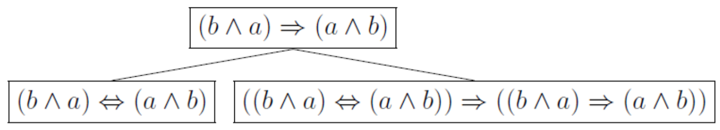
Dôkaz má v symbolickej logike taktiež stromovú štruktúru. Na rozdiel od sémantického tabla, strom dôkazu navyše poskytuje informáciu aj o tom, z ktorých tvrdení skúmaná formula vyplýva.
Algoritmus zatiaľ vie nájsť dôkaz pre jednoduché formuly.

Ukážka

- **vstup** $(b \wedge a) \Rightarrow (a \wedge b)$

`(b \land a) \Rightarrow (a \land b)` // formula

- **výstup**



Algoritmus hľadá najpodobnejšiu axiómu zadanej formuly, pomocou ktorej sa aj vygeneruje nová formula podľa určitých pravidiel. Následne, pomocou odvodzovacích pravidiel logiky, vieme prehlásiť, že zadaná formula je tautológiou.

Poďakovanie

Ďakujem vedúcemu mojej bakalárskej práce, doc. RNDr. Stanislavovi Krajčimu, PhD., za odbornú pomoc a množstvo inšpiratívnych myšlienok.

Literatúra

1. Stanislav Krajči, Symbolická logika
2. Goldstern Martin and Judah Haim , The incompleteness phenomenon, A new course in mathematical logic, A K Peters, Wellesley, Mass., 1995, xiii + 247 pp.
3. Vladimír Kvasnička, Úvod do logiky pre informatikov