POROVNÁVÁNÍ VĚTNÝCH ROZBORŮ

DOKUMENTACE K ZÁPOČTOVÉMU PROGRAMU

NAPROGRAMOVALA: KAROLÍNA KUCHYŇOVÁ, I. ROČNÍK, STUDIJNÍ SKUPINA 34

Zadání:

Program vznikl po dohodě s paní doktorkou Barborou Hladkou. Jeho účelem je porovnat větné rozbory vytvořené v editoru Čapek. Vyjádřit míru shody mezi dvěma rozbory s vypsáním případných rozdílů.

Čapek (http://ufal.mff.cuni.cz/capek) je editor tvaroslovných a větných rozborů pro školáky a jejich učitele. Editor umožňuje provádět tvaroslovné a větné rozbory stejným způsobem, na který jsou školáci zvyklí ze škol. Je jazykově nezávislý, tj. je v něm možné provádět rozbory vět kteréhokoli jazyka. Editor rozšiřuje systém elektronické cvičebnice STYX (http://ufal.mff.cuni.cz/styx).

Rozbory zpracované editorem jsou uloženy v souborech ve formátu XML. Jeden soubor obsahuje několik rozebraných vět. Příslušná definice typu dokumentu nebyla k dispozici a při tvorbě programu se vycházelo pouze z poskytnutých datových souborů.

Zadavatelkou byly poskytnuty rozbory 101 vět zpracované dvěma učitelkami a dvěma žákyněmi. Cílem bylo porovnání každé dvojice dokumentů.

Popis datových souborů:

Datový soubor obsahuje rozbory jednotlivých vět v elementech *sentence*. Jeden element *sentence* obsahuje seznam *tokenů* a syntaktický rozbor v elementu *tree*, který je tvořen seznamem uzlů (*node*), obsahujícím jednotlivé větné členy.

Element *token* obsahuje tyto informace:

- *tag* - určením gramatických kategorií,
- *txt* - vlastní slovo
- *ord* – pořadové číslo slova ve větě
- *id* – identifikace tokenu.

Element *node* obsahuje:

- *id* – identifikace uzlu
- *parent.rf* – identifikátor nadřazeného uzlu
- *token.rf* – identifikátor tokenu v uzlu (jeden uzel může obsahovat více tokenů)
- *fnc* – typ větného členu
- *tag* – určení gramatických kategorií hlavního tokenu uzlu.

Kritéria porovnávání:

Porovnávání se provádí na dvou úrovních.

1) Porovnání základních skladebních dvojic. Sledovaným jevem je zde shoda určení podmětů a přísudků v obou větách. Jev je považován za shodný, pokud si vzájemně odpovídají všechna slova příslušných uzlů. V případě neshodného jevu se vypíší veškerá slova, která neobsahuje druhý rozbor v příslušném uzlu. Na konci každého souboru se vypíše počet porovnávaných jevů a počet shodných porovnání.
2) Porovnání jednotlivých uzlů. Pro každý token sledujeme následující jevy:
    - shodu v určení *tagu* (V případě neshody se vypíše, jak byl token označen v každém z rozborů.)
    - shodu v určení *fnc*, tedy o jaký větný člen se jedná (V případě neshody se vypíše, jak byl token označen v každém z rozborů.)
    - shodu v tom, se kterými dalšími tokeny je v uzlu, kdy každý jednotlivý token v uzlu se považuje za jeden jev (Při neshodě se vypisují všechny rozdílné tokeny.)
    - shodnou závislost, tj. jestli se shodují alespoň v jednom tokenu obsaženém v nadřízeném uzlu. (Při neshodě se vypisuje všechny tokeny nadřízených uzlů v každém z rozborů.)

Na konci každého souboru se vypíše počet porovnávaných jevů a počet shodných porovnání.

Uživatelská část:

Program pracuje se vstupním souborem, jehož jméno si přečte jako první parametr příkazového řádku. Na prvním řádku obsahuje soubor název výstupního souboru pro srovnání základních skladebních dvojic. Na druhém řádku obsahuje název výstupního souboru pro srovnání všech uzlů. Další řádky budou obsahovat jména srovnávaných souborů (v pevném pořadí, na každém řádku jeden soubor).

Volání programu z příkazového řádku:

```
RozdeleneAnalyza.exe inputFile
```

kde `inputFile` je jméno souboru s daty.

Počet porovnávaných vět v souborech je omezen pouze velikostí paměti – předpokládá se, že zpracovávaných vět mohou být v souboru stovky až tisíce. Jednotlivé věty (souvětí) se mohou skládat z maximálně deseti vět a jeden uzel může obsahovat maximálně deset tokenů.

Formát výstupních souborů je popsán v části zadání u jednotlivých typů srovnání.

Programátorská část:

Program je rozdělený do tří částí. První část (`RozdeleneNacteni.pas`) je načtení dat z XML dokumentu do datových struktur vhodných pro další zpracování. Druhá část (`RozdelenePorovnavani.pas, RozdelenePorovnavaniII.pas )` provede srovnání základních skladebních dvojic a všech uzlů. Třetí část (`RozdeleneAnalyza.pas`) je hlavní program, který načte zadání ze souboru uživatele a poté načte data a provede srovnání.

1) Načtení dat z XML (`RozdeleneNacteni.pas`)

Pro přístup k souborům XML se používají standardní knihovny Free Pascalu DOM a XMLRead. Hlavní funkcí je `ReadDoc`, která volá proceduru `ReadSentence`, která volá načítání jednotlivých elementů `ReadToken` a `ReadNode`. K doplnění informací z uzlu do tokenu je použita procedura `NodeToTokenI` a `NodeToTokenII`.

Načtená data jsou uložena do následujících struktur:

```
TToken = record
  id : widestring;                                          //identifikace tokenu
  txt : widestring;                                         //slovo
  node : widestring;                                        //identifikace uzlu, ve kterem je token
  inNode : array [0..maxPocetSlovVUzlu] of widestring;      //seznam tokenu v uzlu, do ktereho slovo patri
  inNodeNumber : integer;                                   //pocet tokenu v uzlu, do ktereho slovo patri
  fnc: widestring;                                          //vetny clen, ktery slovo predstavuje
  parentNode : widestring;                                  //identifikace nadrazeneho uzlu
  parents : array [0..maxPocetSlovVUzlu] of widestring;     //seznam tokenu v nadrazenem uzlu
  parentsNumber : integer;                                  //pocet tokenu v nadrazenem uzlu
  tag : widestring;                                         //gramaticka kategorie slova
  next: PToken                                              //pointer na dalsi token
  end;

TNode = rekord
  id : widestring;                                          //identifikace uzlu
  parent: widestring;                                       //identifikace nadrazeneho uzlu
  inNode : array[0..maxPocetSlovVUzlu] of widestring;       //seznam tokenu v uzlu
   fnc : widestring;                                        //vetny clen, ktery uzel predstavuje
  next : PNode;                                             //pointer na dalsi uzel
  inNodeNumber : integer;                                   //pocet tokenu v uzlu
  tag : widestring;                                         //gramaticka kategorie hlavniho tokenu
  end;

TSentence = record
  firstToken : PToken;                                  //pointer na prvni token vety
  pocetPodmetu,pocetPrisudku : integer;                 //pocet podmetu a pocet prisudku ve vete (souveti)
  podmet : array [1..maxPocetVetVSouveti,1..maxPocetSlovVUzlu]
               of widestring;                                   //seznam podmetu
  prisudek :  array [1..maxPocetVetVSouveti,1..maxPocetSlovVUzlu]
               of widestring;                                   //seznam prisudku
  next : PSentence;                                             //pointer na dalsi vetu
  end;
```

2) Porovnávání základních skladebních dvojic (`RozdelenePorovnavani.pas`)

Hlavní funkcí je `CompareDoc`, která pro jednotlivé věty volá proceduru `CompareSentence`, porovnávající jednotlivé přísudky a podměty.

3) Porovnávání všech uzlů (`RozdelenePorovnavaniII.pas`)

Hlavní funkcí je `CompareDocX`, která pro jednotlivé věty volá proceduru `CompareSentence`, porovnávající jednotlivé tokeny v proceduře `CompareToken`.

4) Hlavní program (`RozdeleneAnalyza.pas`)

Načte soubor se seznamem analyzovaných rozborů a zavolá příslušné procedury pro načtení a porovnání. Formát vstupního souboru je popsán v uživatelské části.