

# Elisa

**Zápočtový program do Neprocedurálního programování**

Martin Mareš

Červen 2016

# 1 Popis programu

Elisa je program, který vede s člověkem rozhovor v přirozeném jazyce. Tento rozhovor se týká jeho pocitů a problémů. Elisa se tedy chová jako jakýsi psycholog. Z důvodu složitosti Českého jazyka je rozhovor veden v Angličtině.

Celý program byl napsán v *Prologu*. Pro jeho spuštění je doporučován *SWI-Prolog 7.2.2 (swipl)*. Následující posloupností příkazů v kořenu programu lze program spustit:

```
$ swipl
?- consult("elisa.pl").
?- elisa.
```

Po spuštění Elisa pozdraví uživatele, a nabídne mu prostor pro odpověď.

```
Hello , I'm Elisa , how can I help you?
> [VSTUP]
```

Program lze ukončit rozloučením s Elisou a následným ukončením interpretu Prologu:

```
> Bye, have a nice day.
Goodbye. My secretary will send you a bill.

?- halt.
```

## 2 Popis fungování

Program čte vstup po řádcích. Celý řádek převede na malá písmena, a odstraní z něj interpunkční znaménka. Poté řádku rozseká na slova. Slova zjednoduší a upraví pomocí zjednodušovacích pravidel. Zejména použije vhodná synonyma, odstraní množná čísla a převrátí osoby. Například větu: „Moje mamka má počítače.” převede na „Tvoje matka má počítač.”.

Ke vzniklému textu se poté snaží najít předpřipravenou odpověď. Ke každé takové odpovědi existuje vzor vstupní věty a její skóre, takže program může vždy nalézt tu nejvhodnější.

## 3 Rozšíření Elisy o další schopnosti

V kořenové složce programu se nacházejí dva soubory *reply.rules* a *simplification.rules*. První soubor obsahuje odpovědi na vzory otázek a druhý soubor obsahuje zjednodušovací pravidla. Základní vnitřní texty byli založeny na textech od *Virena Patela* z jeho programu *ELIZA in Prolog*.

Zjednodušovací pravidla slouží pro hledání synonym a převrácení osob. Jsou ve formátu

```
sr ([ hi ,Name|X] , [ hello ,Name|Y] ,X,Y) .
```

Kde  $X$  a  $Y$  označují zbytek řádky a  $Name$  může označovat společnou část. Použitím tohoto pravidla by se text „hi martin” převedl na „hello martin”. Pravidla se čtou v pořadí od začátku souboru a pro každý úsek textu se použije maximálně jedno pravidlo.

Druhý soubor obsahuje odpovědi na vstupní větu. Pro každý vzor vstupní věty může existovat více odpovědí.

```
rules (ID , importance of rule , [the pattern] , [
    [response 1] ,
    [response 2] ,
    ...
    [response n] ] ) .
```

V odpovědích je možné využívat standardní matchování z Prologu. Ve vzoru je vhodné dovolit libovolný konec řádky a případně ho využít v odpovědi. Na všechny sufixy řádky se postupně zkouší matchovat všechny vzory. Z možných odpovědí se vybere ta s největším skóre. Odpovědi se po použití cyklicky střídají.

## 4 Vývojářská dokumentace

Program je dělen do tří souborů. Soubory *\*.rules* obsahují samotná pravidla pro odpovědi a zjednodušování. Hlavním kód je obsažen v *elisa.pl*.

Soubor je členěn do čtyř částí, první část obsahuje predikát *test*, který slouží pro spouštění testů vnitřních funkcí. Druhá část obsahuje predikáty *in* a *out* pro čtení vstupu a výstupu. Ty poskytují API pro možné portování programu do jiného prostředí (webová stránka apod...). Třetí částí jsou vnitřní funkce, které nijak nesouvisí se samotným účelem programu, ale jsou užitečné pro práci s listy a Stringy. Poslední část tvoří predikáty důležité pro fungování Elisy - zjednodušování textu, hledání odpovědi, změna odpovědí.

Program je spouštěn pomocí predikátu *elisa*, tento predikát natáhne soubory pravidel (*\*.rules*) pomocí predikátu *consult*. Tento způsob umožňuje relativně jednoduše vytvářet složitější pravidla, která využívají unifikací a funkcí z *Prologu*. Po načtení pravidel se vypíše úvodní zpráva a čeká se na vstup od uživatele.

Vstup je čten znak po znaku a ukládán do listu znaků až do znaku nového řádku. Poté je znak po znaku převeden na malá písmena a je z něj odstraněna interpunkce. Vzniklý text je pomocí akumulátoru převeden na list slov. Na tomto listu jsou pro jednotlivé sufixy zkoušeny pravidla pro hledání synonym. Pokud se nalezne synonymum, tak se pokračuje v úpravě textu od prvního nezměněného sufixu.

Po nalezení synonym se pro jednotlivé sufixy program pokouší nalézt odpověď. Mezi pravidly existuje univerzální odpověď, takže vždy nějakou odpověď lze nalézt. V průběhu čtení sufixů se uchovává pouze odpověď s nejvyšším skóre. Po nelezení té nejlepší se využije ID použitého pravidla pro úpravu predikátu, který určuje, která odpověď se má vybrat pro které pravidlo.

```
resID(+ID,+ResID) .
```

Predikát se poté změní tak, že ID odpovědi (*resID*) se cyklicky posuneme na další odpověď. Pokud tento predikát zatím neexistuje, tak je vytvořen nově. Pokud existuje, tak se nejprve smaže.

Odpověď je poté vypsána na výstup. První písmeno ve větě je převedeno do velkých písmen a slova jsou oddělena mezerou (před interpunkcí se mezera nevytváří). Takto program čte vstup dokud odpověď neobsahuje *quit* nebo jeho synonymum.