

Domácí úkol 12: Zpracování přirozeného jazyka

Hlavní oblasti použití jazyka Prolog jsou umělá inteligence a zpracování přirozeného jazyka. Zjednodušené zpracování přirozeného jazyka si můžete vyzkoušet v této úloze.

Vaším úkolem bude napsat program v jazyce Prolog, který o zadaném řetězci rozhodne, jestli se jedná o správně utvořenou větu. Pro účely tohoto úkolu považujeme za správně utvořenou větu posloupnost slov a čárek oddělených mezerami, která splňuje následující pravidla:

- *Věta* je posloupnost slov, která se skládá z podmětu, slovesa a předmětu.
- *Vedlejší věta* je posloupnost slov, která začíná slovem "**ktera**", po kterém následuje sloveso a předmět.
- *Podmět* je podstatné jméno v prvním pádě, za kterým může následovat vedlejší věta oddělená z obou stran čárkami.
- *Předmět* je podstatné jméno ve čtvrtém pádě, kterému může předcházet libovolný počet přídavných jmen a za kterým může následovat čárka a vedlejší věta.
- *Podstatné jméno v prvním pádě* je jedno ze slov "**zena**", "**ruze**", "**pisen**" a "**kost**".
- *Podstatné jméno ve čtvrtém pádě* je jedno ze slov "**zenu**", "**ruzi**", "**pisen**" a "**kost**".
- *Přídavné jméno* je jedno ze slov "**krasnou**", "**tvrdou**" a "**ostrou**".
- *Sloveso* je jedno ze slov "**zpiva**", "**vidi**" a "**vari**".

Formálněji řečeno, správně utvořené věty je možné generovat bezkontextovou gramatikou s kořenovým neterminálem V a následující množinou pravidel:

$P = \{$	V	\rightarrow	Podmět Sloveso Předmět
	W	\rightarrow	ktera Sloveso Předmět
	Podmět	\rightarrow	Kdo Kdo ₂ W_2
	Předmět	\rightarrow	Koho Koho ₂ W Jakou Předmět
	Kdo	\rightarrow	<u>zena</u> <u>ruze</u> <u>pisen</u> <u>kost</u>
	Koho	\rightarrow	<u>zenu</u> <u>ruzi</u> <u>pisen</u> <u>kost</u>
	Jakou	\rightarrow	<u>krasnou</u> <u>tvrdou</u> <u>ostrou</u>
	Sloveso	\rightarrow	<u>zpiva</u> <u>vidi</u> <u>vari</u> }.

Tedy správně utvořené věty jsou například:

- "zena vidí ruzi"
- "zena , ktera zpiva pisen , vidí ruzi"
- "zena , ktera zpiva krasnou pisen , vidí ostrou ruzi"
- "zena , ktera zpiva krasnou pisen , vidí ostrou krasnou ostrou ruzi"
- "zena , ktera zpiva krasnou pisen , vidí ostrou ruzi , ktera vari tvrdou kost"
- "zena , ktera zpiva krasnou pisen , vidí krasnou zenu , ktera vidí krasnou zenu , ktera vari tvrdou kost "
- "ruze , ktera vari tvrdou zenu , vidí ruzi , ktera zpiva tvrdou ostrou kost"

Naopak následující věty správně utvořené nejsou:

- "zena"
- "zena vidí"
- "krasna zena vidí ruzi"
- "zena vidí ruzi , ktera vidí"
- "zena vidí ruzi , kterou vidí zena"
- "zena vidí ruzi , ktera vidí zena"

Zadání

Napište predikát `isCorrectSentence/1`, který pro zadaný řetězec uspěje právě tehdy, když je tento řetězec správně utvořenou větou podle výše zmíněných pravidel. Tedy například

```
?- isCorrectSentence("zena vidí ruzi").
true.
?- isCorrectSentence("zena , ktera zpiva krasnou pisen , vidí ostrou ruzi").
true.
```

```

?- isCorrectSentence("zena , ktera zpiva krasnou pisen , vidi zenu , ktera vidi zenu").
true.
?- isCorrectSentence("zena").
false.
?- isCorrectSentence("zena vidi").
false.
?- isCorrectSentence("zena vidi ruzi , ktera vidi zena").
false.

```

Váš predikát má fungovat v módu `isCorrectSentence(+String)`; případy, kdy argument není instanciováný nebo není řetězec, nemusíte řešit. Můžete předpokládat, že všechna slova a čárky ve vstupním řetězci jsou od sebe oddělené mezerami. Můžete také předpokládat, že vstupní řetězec neobsahuje více mezer po sobě.

Všimněte si, že v žádném případě interpret jazyka Prolog nenabízí žádné další řešení. Vaše řešení by se mělo chovat stejně, pokud má být ohodnoceno plným počtem bodů.

Nápověda

Pokud si chcete ulehčit život, doporučujeme rozdělit si úkol na dva podúkoly. Nejprve naprogramujte predikát `splitString/2`, který do druhého argumentu unifikuje seznam všech slov a čárek, které se vyskytují v řetězci v prvním argumentu. Například:

```

?- splitString("zena , ktera zpiva krasnou pisen , vidi ostrou ruzi", X).
X = ["zena", ",", "ktera", "zpiva", "krasnou", "pisen", ",", "vidi", "ostrou", "ruzi"].

```

Při programování predikátu `splitString` se vám můžou hodit predikáty na práci s řetězci. Ty můžete vyhledat dotazem `?- apropos(string)..`

Poté naprogramujte predikát `v/1`, který uspěje právě tehdy, když je seznam slov a čárek správně utvořená věta. Například:

```

?- v(["zena", ",", "ktera", "zpiva", "krasnou", "pisen", ",", "zpiva", "ruzi"]).
true.

```

Poté vyjádřit predikát `isCorrectSentence` pomocí těchto predikátů už bude jednoduché.

Odevzdávání a bodování

Váš program odevzdávejte do příslušné odevzdávací skříně¹. Odevzdávejte jeden soubor s příponou `.pl`, který obsahuje okomentovaný zdrojový kód. Do komentářů slovně popište, jak jste při řešení příkladu postupovali. Čas na vypracování úkolu je do středy 21. prosince (včetně).

Za celou úlohu můžete dostat dva body. První bod je za funkčnost kódu, druhý bod je za jeho čitelnost, smysluplnost a slovní popis. Je možné získat i desetinné body, pokud bude řešení jen částečně správné. Oba body uděluje cvičící.

¹https://is.muni.cz/auth/el/1433/podzim2016/IB015/ode/ode_hw12/