

SMILES

SMILES je jednoduchý textový formát na ruční zadávání grafů do počítače. Původně je vymyšlený pro molekuly, ale koncept jde jednoduše generalizovat na libovolné grafy. [Přesný popis a nějaké příklady jsou na wikipedii.](#)

Volně přeloženo, SMILES v chemii popisuje anotovaný graf (a.k.a. molekulu), kde všechny vrcholy (a.k.a atomy) mají jméno (a.k.a. prvek) a všechny hrany (a.k.a. chemické vazby) mají přiřazený typ (např. násobnost).

My budeme pracovat se zjednodušeným SMILES systémem, který popisuje jen obyčejný, chemií nezatížený graf, pouze s anotovanými vrcholy. Anotace vrcholu je string začínající jedním velkým písmenem a pokračující libovolným množstvím malých písmen. Budeme uvažovat jen spojité grafy.

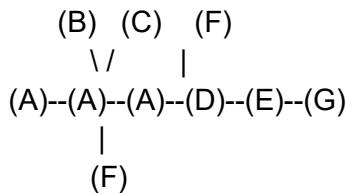
Jednoduché rovné grafy (cesty) se zapisují následovně:

- A je graf obsahující jedený vrchol s anotací (jménem) "A"
- AA je graf obsahující 2 vrcholy se jmény "A" a "A", spojené hranou. Graficky asi takhle: (A)—(A)
- ABA je graf s jednou cestou na které jsou postupně vrcholy "A", "B" a "A": (A)—(B)—(A)
- AhojAhoj obsahuje 2 spojené vrcholy anotované "Ahoj": (Ahoj)—(Ahoj)

Když chcete udělat na grafu odbočku (tj. vyrobit strom), otevřete závorku:

- AB(C)D je graf ve kterém je cesta z A, B a D, a na B je ještě navíc připojené C.
- Uprostred(Kolem)(Kolem)(Kolem)(Kolem) je graf který má na vrchol se jménem "Uprostřed" připojené 4 vrcholy se jménem "Kolem".
- Závorky jde samozřejmě libovolně zahnizdovat do sebe. Třeba [tahle super molekula](#) se zapíše jako HOC(H)(C(H)(H)H)C(H)(H)H (v obrázku na wiki je černě uhlík®, červeně kyslík (O) a bílé vodík (H)).

Graficky: například SMILES AAA(B)(C)(DE(F)G)F popisuje tenhle graf:

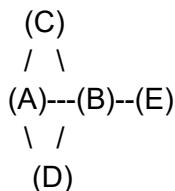


Když chcete v grafu udělat cyklus (tj. vyrobit kolečko), můžete 2 libovolné molekuly ve vytvářeném stromě spojit pomocí číselné zpětné reference — nějaký vrchol označíte

libovolným číslem, a to samé číslo pak můžete použít kdekoliv znova pro výrobení hrany vedoucí k původnímu vrcholu.

- V grafu A1AAA1 je první A označené jedničkou, a poslední A je s ním spojené. Ve výsledku z toho je čtvercový graf s vrcholy označenými A.
- Graf [K4](#) se zapíše jako X1X2X1X(1)2
- Graf [K3,3](#) se dá zapsat jako X10Y20X11Y21(10)X12(20)Y22(10)11.

Grafický příklad — SMILES A123B(C123)(E)D123 vypadá jako graf takhle:



Zadání

Vyrobte predikát smiles_graph(+SmilesString, -Graf), který ze stringu ve formátu SMILES vyrobí nějaký pěkný, strojově zpracovatelný popis grafu. Pěkný popis je například seznam označení vrcholů a seznam hran s indexy sousedících vrcholů.

Příklad

Na dotaz

```
?- smiles_graph("Aaa555BbbCcc555", graph(V,E)).
```

by prolog mohl odpovědět např. takhle:

```
V = ['Aaa', 'Bbb', 'Ccc']  
E = [edge(0,1), edge(1,2), edge(2,0)]
```

Ideálně do řešení zahrňte predikát demo, který bude obsahovat nějaký návod, jak vaši formu predikátu správně spustit. V tomhle případě třeba takhle:

```
demo :-  
    smiles_graph("AaAaAa", graph(V,E)),  
    write("Vrcholy: "), writeln(V),  
    write("Hrany: "), writeln(E).
```

Na neplatný SMILES vzorec by Prolog měl odpovědět false.

Pomůcky

Můžou se hodit následující věci ze standardní knihovny Prologu:

```
?- string_chars("aSd('1",X).
X = [a, 'S', d, '(', '\'', '1'].
?- atom_number('1',N).
Y = 1.
?- string_to_list("aSd('1",ASCII).
ASCII = [97, 83, 100, 40, 39, 49].
?- atom_concat(ka,kao,X).
X = kakao.
?- char_type('a', lower).
true.
?- char_type('a', upper).
false.
?- char_type('a', digit).
false.
```

Parsování SMILES určitě proveděte pomocí prologových gramatik, je to tak nejpohodlnější. Do seznamu cvičení (na hlavní stránce dole) jsem přidal nějakou formálnější definici, příklady přepisování gramatických pravidel na normální, a taky demo s parsováním aritmetiky. (SMILES bude mít podstatně jednodušší parser, ale s vyparsovaným obsahem bude potřeba dělat trochu složitější věci.)

DCG jsou když tak na internetu dokumentované různě [systematicky](#).