

allCHEMIE – Uživatelská příručka

Ondrej Profant

27. května 2011

Obsah

1	Úvod	3
2	Instalace	3
2.1	Linux	3
2.2	Windows	3
3	Použití	3
3.1	Predikáty v Prologu	3
3.2	Informace o jednotlivých prvcích	4
3.3	Koncovky českého názvosloví	4
3.4	Převod chem. vzorce na slovní vyjádření	4
3.4.1	Malá poznámka k solím	5
3.4.2	Organika	5
3.5	Převod slovního vyjádření na chem. vzorec	5
3.6	Doplnění neúplného vzorce o možné prvky a četnosti	6
3.7	Skládání sloučenin	6
4	Známé problémy	6
5	Otázky a odpovědi (FAQ)	7
6	Zdroje	7

1 Úvod

Program allChemie vznikl jako zápočtový roku 2010 a je psán v jazyce Prolog. Slouží k základní práci se vzorci převážně anorganické chemie. Přesněji řečeno má tyto základní možnosti užití:

1. Informace o jednotlivých prvcích
2. Koncovky českého názvosloví, řecké přípony etc.
3. Převod chem. vzorce na slovní vyjádření ($\text{CO}_2 \rightarrow \text{Oxid uhličitý}$)
4. Ze slovního vyjádření určí vzorec ($\text{Oxid uhličitý} \rightarrow \text{CO}_2$)
5. Doplnování možností ($\text{CX}_N \rightarrow \text{CO}_2$)

2 Instalace

2.1 Linux

Je třeba nainstalovat <http://www.swi-prolog.org/>, např. v Debianu/Ubuntu:

```
sudo apt-get install swi-prolog
```

Ten pak spustíme příkazem:

```
prolog
```

to nás, ale úplně nezajímá. My chceme spustit konkrétně allCHEMII:

```
prolog -q -f <cesta_k_souboru/main.pro>
```

(očekává se, že všechny soubory programu budou v dané složce)

2.2 Windows

Stačí stáhnout a nainstalovat <http://www.swi-prolog.org/>, poté stáhnout zdrojové kódy programu, rozbalit a kliknout na soubor "main". Program allChemie pak bude asociovat s Prologem, čili můžete začít pracovat s programem podle dalšího návodu. Správnou instalaci programu allChemie poznáte tak, že se po kliknutí na soubor "main" se zobrazí hlavička s nápisem CHEMIE a s možnostmi využití programu.

3 Použití

3.1 Predikáty v Prologu

Varování: Tato kapitola bude asi velmi těžko pochopitelná technicky méně zdatným uživatelům, ale při čtení dalšího textu bude velmi užitečná.

Program není „klasickým program“, ale spíše „skriptem“ pro jazyk Prolog. Je dobré uvědomit si několik základních faktů:

- Spuštěním prologu se dostaneme do interaktivního prostředí, kde bude: „?- ” a kurzor. V tomto prostředí zadáváme predikáty.

- Predikát je „vlastně funkce”: má název a za ním v závorce parametry
 - Pokud ho zadáváme, tak za ním musíme napsat tečku.
- Enterem predikát potvrdíme a pak se Prolog pokusí vyhovět takovému predikátu. Pokud nám výsledek nevyhovuje můžeme zmáčknout středník a dostaneme další možnosti.
- Parametry jsou:
 - Velká písmena, což jsou proměnné, které chceme doplnit
 - Malá písmena, což jsou pevně daná fakta (data)
 - Podtržítka, což je taková proměnná, co nás nezajímá (anonymní)

3.2 Informace o jednotlivých prvcích

Stačí zadat: `vstup(Prvek)`, kde za Prvek dosadíte značku prvku malými písmeny, např:

`vstup(h).`

Pro vyhledávání dle více parametrů slouží predikát `prvek`, který má 6 parametrů. Ovládá se jako každý predikát v Prologu. Malé písmeno znamená pevně daný fakt (například parametr vyhledávání), velké písmeno znamená proměnnou (tedy to, co chceme najít) a podtržítka znamená údaj, který nás nezajímá. Syntaxe predikátu `prvek` je následující:

```
prvek( <prot. cislo> , <znacka> , <nazev> , {kov,pol,nek} ,
      <skupina> , <sezn. oxid. cisel> ).
```

, kde {} značí množinu - jsou zde právě tři možnosti. Čili použití následující:

```
prvek( 1 , h , vodik , _ , _ , X ).
```

V odpovědi se nám dostane X, které bude seznamem oxidačních čísel, kterých nabývá vodík, tedy:

```
X = [1].
```

3.3 Koncovky českého názvosloví

Koncovky se používají obdobně jako predikát `prvek` z předchozí kapitoly. Používá se predikát `koncovka`, jehož syntaxe je následující:

```
koncovka( <cislo> , <pripon.> , <pripon. kys> , <pripon. sul> ).
```

3.4 Převod chem. vzorce na slovní vyjádření

Zde je důležitý predikát `vstup()`, který je silně přetížený - tj. má proměnný počet parametrů (2-10). Stačí mu zadat kompletní vzorec (pro správnou funkci je třeba vyplnit všechna čísla vč. 1). Např:

<code>vstup(h,2,o,1).</code>	voda
<code>vstup(c,1,o,2).</code>	oxid uhličitý
<code>vstup(cl,2,o,7).</code>	oxid chloristý
<code>vstup(ba,1,(oh),2).</code>	hydroxid barnatý
<code>vstup(h,1,f,1).</code>	flurovodík
<code>vstup(h,2,s,1,o,4).</code>	kyselina sírová
<code>vstup(h,3,p,1,o,4).</code>	kyselina trihydrogen fosforečná
<code>vstup(na,1,cl,1).</code>	chlorid sodný
<code>vstup(k,1,n,1,o,3).</code>	dušičnan draselný
<code>vstup(nh4,2,c,1,o,3).</code>	uhličitan amonný
<code>vstup(na,1,h,2,p,1,o,4,1).</code>	dihydrogenfosforečnan sodný
<code>vstup(ca,1,s,1,o,4,1).</code>	síran vápenatý
<code>vstup(ca,1,s,1,o,4,1,*,5,h2o).</code>	pentahydrát síranu vápenatého

Poznámka: Program neimplementuje české znaky a morfologii (ohýbání a skládání slov), čili vypisuje zjednodušeně uhlík-icity, namísto uhličitý. V manuálu uvádím plné názvy pro přehlednost.

Některé sloučeniny mají nadefinované zjednodušené vzory, ale samozřejmě to také zvyšuje pravděpodobnost výskytu chyby (res. špatné interpretace), např:

<code>vstup(h,2,o).</code>	voda
<code>vstup(c,o,2).</code>	oxid uhličitý
<code>vstup(h,f).</code>	flurovodík
<code>vstup(ba,oh,2).</code>	hydroxid barnatý
<code>vstup(h,2,s,o,4).</code>	kyselina sírová

3.4.1 Malá poznámka k solím

Amoniak (NH_3), res jeho kationt (NH_4) je považován za atom a vepisuje se vcelku.

Poslední číslo u solí je obvykle dobrovolné a udává počet celé aniontové skupiny, např v: $\text{Ca}_1(\text{SO}_4)_1$ je to jedna.

3.4.2 Organika

Implementována je jen základní řada do $\text{C}_1\text{H}_4\text{O}$ a její uhlíkové zbytky. Obsluhuje se stejně jako v předchozích příkladech:

```
?- vstup(c,1,h,4).
metan
?- vstup(c,1,h,3).
metyl
?- vstup(c,5,h,12).
penta-an
?- vstup(c,5,h,11).
penta-yl
```

3.5 Převod slovního vyjádření na chem. vzorec

Pro sloučeniny jsou připraveny spec. predikáty. Nejjednoduší bude ukázat to na příkladech:

```

?- oxid(uhlik,icity).
co2
?- oxid(lithium,ny).
li2o
?- fluorid(draslik,ny).
kf
?- bromid(sodik,ny).
nabr
?- kyselina(brom,na).
hbro

```

3.6 Doplnění neúplného vzorce o možné prvky a četnosti

K této funkci je využíván samotný Prolog, stačí nahradit prvek proměnnou a Prolog již najde možné prvky.

```

vstup(X,1,o,2).
X = c ;          oxid uhličitý
X = n ;          oxid dusičitý
X = si ;         oxid křemičitý
X = s ;          oxid siřičitý
...

```

Vynechat lze i více prvků, či četnost nějakého atomu.

3.7 Skládání sloučenin

Predikát `sloz()` jednoduše najde možné sloučeniny. Četnosti si doplňuje sám.

```

sloz(cl,na).
chlorid sodný

sloz(na,cl).
chlorid sodný

sloz(h,o).
voda ;
peroxid vodiku.

```

Je připraven na četnost 2,3,4.

4 Známé problémy

AllCHEMIE zcela jistě obsahuje velké množství chyb. Je to zapříčiněno těmito fakty:

1. Autor není chemik a jen velmi omezeně konzultoval s chemiky.
2. Ne vše je v chemii tak pravidelné, jak by se mohlo zdát.
3. Jedná se o zápočtový projekt s omezenými časovými možnostmi (účelem je napsat program v Prologu, nikoliv ošetřit každou nepravidelnost).

5 Otázky a odpovědi (FAQ)

Q: Mohou se v programu vyskytovat chyby:

A: Samozřejmě.

Q: Je možné studovat zdrojový kód?

A: Ano, program má plně otevřený kód, koneckonců je distribuován pouze formou kódu.

Q: Je možné přispívat svými vylepšeními?

A: Ano, celý projekt je spravován skrze GIT¹ na github.com, stačí se připojit: <http://kedrigern.github.com/allCHEMIE/>

6 Zdroje

Viz. specifikace.

¹System pro správu verzí.