

# allCHEMIE – Uživatelská příručka

Ondrej Profant

22. června 2011

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Instalace</b>	<b>3</b>
2.1	Linux . . . . .	3
2.2	Windows . . . . .	3
<b>3</b>	<b>Použití</b>	<b>3</b>
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 . . . . .	5
3.4.1	Malá poznámka k solím . . . . .	5
3.4.2	Organika . . . . .	5
3.5	Převod slovního vyjádření na chem. vzorec . . . . .	6
3.6	Doplnění neúplného vzorce o možné prvky a četnosti . . . . .	6
3.7	Skládání sloučenin . . . . .	7
<b>4</b>	<b>Známé problémy</b>	<b>7</b>
<b>5</b>	<b>Otázky a odpovědi (FAQ)</b>	<b>7</b>
<b>6</b>	<b>Zdroje</b>	<b>7</b>

# 1 Úvod

Program allChemie vznikl jako zápočtový roku 2011 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. Doplnění 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 allCHEMIE:

```
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"<sup>1</sup>. 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" 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:

---

<sup>1</sup>Pokud máte v systému nastavené zobrazování přípon, tak bude zobrazeno "prolog.pro".

- 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> ).
```

Např. takto (2.-4. parametr jsou proměnné):

```
?- koncovka(2,Oxid,Kyselina,Sul).
Oxid = naty,
Kyselina = nata,
Sul = natan.
```

### 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(c,1,o,3).</code>	false. % program nezna odpovídající sloučeninu
<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>	fluorvodík
<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>	dusičan draselný
<code>vstup(nh4,2,c,1,o,3).</code>	uhličitan amonný
<code>vstup(na,1,h,2,p,1,o,4,1).</code>	dihydrogenfosforečan 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>	fluorvodík
<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 ( $\text{NH}_3$ ), res jeho kationt ( $\text{NH}_4$ ) 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  $\text{C}_{19}\text{H}_{40}$  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
?- hydroxid(vapnik,naty).
ca(oh)2
?- hydroxid(sodik,ny).
na(oh)
?- fluorid(draslik,ny).
kf
?- bromid(sodik,ny).
nabr
?- kyselina(brom,na).
hbro
?- kyselina(dusik,icna).
hno3
```

### 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
...
?- vstup(c,X,h,Y).
metan
X = 1,
Y = 4 ;
metyl
X = 1,
Y = 3 ;
etan
X = 2,
Y = 6 ;
ethyl
X = 2,
Y = 5 ;
propan
X = 3,
Y = 8 ;
propyl
```

```
X = 3,  
Y = 7  
...
```

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 vodíku.
```

Je připraven na četnost 2,3,4.

## 4 Známé problémy

AlCHEMIE 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 GIT<sup>2</sup> na github.com, stačí se připojit: <http://kedrigern.github.com/allCHEMIE/>

## 6 Zdroje

Viz. specifikace.

---

<sup>2</sup>Systém pro správu verzí.