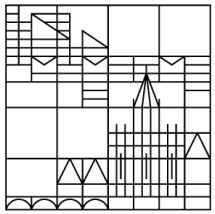

Wikipedia: Improving the rendering of chemical formulae

Student: Manfred Schäfer
Supervisor: Moritz Schubotz
Examiner: Prof. Dr. Bela Gipp
Date: 2017-07-17

Motivation

H_2O

```
<chem>H2O</chem>
```

$(NH_4)_2S$

```
<chem>(NH4)2S</chem>
```

$2\ Na + 2\ H_2O \longrightarrow 2\ Na^+ + OH^- + H{-}H$

```
<chem>2Na + 2H2O -> 2Na+ + 2OH- + H-H</chem>
```

Motivation

```
\ce{2Na + 2H2O -> 2Na+ + 2OH- + H-H}
```

$$2\,\mathrm{Na} + 2\,\mathrm{H_2O} \longrightarrow 2\,\mathrm{Na^+} + 2\,\mathrm{OH^-} + \mathrm{H{-}H}$$

$$\mathbf{2\,Na_2^+\,H_2O \longrightarrow 2\,Na_2^{++}\,OH{-} + H{-}H}$$

Why use **texvcjs** at all?

- Security: **texvcjs** ensures that only “safe” LaTeX expressions are used and attack methods like cross site scripting are prevented
- Replaces MediaWiki specific functions
- Makes LaTeX expressions more consistent by introducing braces or removing whitespaces before rendering

Objective

- *Adjust texvcjs to prevent whitespace modifications in ce-tags*
- Discussed options were:
  (1) Develop a full grammar for **`mhchem`**
  (2) Extend grammar of **`texvcjs`** to treat **`\ce`** like **`\mbox`**
  (3) Bypass **`texvcjs`** altogether for rendering
  (4) Duplicate **`mhchem`** parser into **`texvcjs`**

Research tasks – Milestones

- Familiarize myself with the structure of the rendering pipeline
- Evaluate the four options and chose one: option (1) was chosen
- Implement option (1)
- Develop test cases and test the system

Challenges

- Understanding **`pegjs`**
- Understanding how the abstract syntax tree is built
- Grammar contains left recursion
- Switching back to math mode with **`'$'`**

mhchem Grammar

```
<ce> ::= <sentence>
<sentence> ::= <phrase>
             | <sentence> Space <phrase>
<phrase> ::= <word>
           | <word> <single macro>
           | <single macro>
           | '^'
           | '(^)'
<word> ::= <nonletter>
         | Letter
         | <single macro> <nonletter>
         | <word> <nonletter>
         | <word> Letter
         | <word> <single macro> <nonletter>
<nonletter> ::= '$' <LaTeX expression> '$'
              | ...
```

Testing

- Test cases for the new functionality
  All examples of the **mhchem** manual keep their spacing
- Coverage testing
  100% code coverage is desired, currently 76% of lines are covered
- Regression testing
  The new code should produce the same output like the old code for mathematical formulae

Example

- Input
  `\ce{2Na + 2H2O -> 2Na+ + 2OH- + H-H}`

- Output before the project
  `{\ce {2Na+2H2O->2Na++2OH-+H-H}}`

- Output now
  `{\ce {2Na + 2H2O -> 2Na+ + 2OH- + H-H}}`

Outlook – Next steps

- Finalize code, color modifiers within chemical formulae might still need some work
- Improve code coverage by adding more test cases
- Code review by other developers
- Merge into the upstream code
- Production use of the new code

Conclusion

- Grammar of **`mhchem`** is implemented
- Tests for chemical formulae exists
- It looks like the new code hasn’t broken anything else

References

- [1] Martin Hensel: The mhchem Bundle, Documentation for the LaTeX Packages mhchem v4.04, hpstatement v1.02 and rsphrase v3.11. 2016
- [2] ⚓ T140217 Adjust texvcjs to prevent whitespace modifications in ce-tags, 2016. Retrieved May 14, 2017 from Phabricator, wikimedia: phabricator.wikimedia.org/T140217

Questions?