

Simulator Input Language:

Progress on Universally Programmable Intelligent Matter

UPIM Report 7

Technical Report UT-CS-04-518

Bruce J. MacLennan*

Department of Computer Science
University of Tennessee, Knoxville
www.cs.utk.edu/~mclennan/UPIM

September 15, 2002

revised February 18, 2003 and February 3, 2004

Abstract

This report defines three file formats to be used by the UPIM simulator. The molecular reaction specification files describe the node species and the reactions that they obey; the molecular program specification files describe the initial reactants for a molecular computation by specifying the initial quantity of the various reactants; the molecular template specification files name templates, which may be matched to molecular structures to allow reporting the concentrations of various reaction products.

* This research is supported by Nanoscale Exploratory Research grant CCR-0210094 from the National Science Foundation. It has been facilitated by a grant from the University of Tennessee, Knoxville, Center for Information Technology Research. This report may be used for any non-profit purpose provided that the source is credited.

1 Introduction

The UPIM simulator reads three different, but closely related file formats. A *molecular reaction specification* file (.mrs extension) describes the node species (**S**, **K**, **A**, etc.) known to the simulator and the chemical reactions that they obey. Since this corresponds to the “laws of physics,” it defines the givens for the other files. The format of a molecular reaction specification file is given by <mrs_file> below. A *molecular program specification* file (.mps extension) describes the initial reactants for a molecular computation by specifying the initial quantity of the various reactants. Several such files might be read, either at the beginning of a simulation or during it; <mps_file> is the syntax of these files. A *molecular template specification* file (.mts extension) names templates, which may be matched to molecular structures to allow reporting the concentrations of various reaction products; <mts_file> is its syntax.

2 Grammar

```
<mrs_file> ::= <species_defs> <reaction_set>+
<mps_file> ::= <reactant_spec>+
<mts_file> ::= <template_spec>+

<species_defs> ::= <sdef> , ... .
<sdef> ::= <name> <int>
<reaction_set> ::= <network> <reaction> ; ... .
<reaction> ::= => ( <name> [ , <params> ] ) <complex>
<reactant_spec> ::= <int> * <network> .
<template_spec> ::= <name> = <network> .

<network> ::= <node_defs> <complex>
<node_defs> ::= <ndef> , ... .
<ndef> ::= <name> [ : <name> ]
<complex> ::= <link> , ...
<link> ::= <bond> <bond>
<bond> ::= <name> [ _ <int> ]
<name> ::= <letter> <alphanum>* <prime>*
<int> ::= <digit>+
<alphanum> ::= <letter> | <digit>
<params> ::= p = <int>
<prime> ::= '

```

- Syntactic notation: A * means 0 or more occurrences of the preceding item; a + means one or more occurrences. The notation “<X> <Y> ...” means one or more <X>s separated by <Y>s. Brackets [] surround optional items. “<X> | <Y>” means a choice of <X> or <Y>.
- Blanks around <name>s and <int>s, as well as around punctuation, should be ignored (but note that blanks must separate two <name>s in a <bond> and the <name> and <int> in an <sdef>). Lines beginning with a % sign should be ignored as comments.
- Uppercase and lowercase letters are considered different.

3 Explanation

3.1 <species_def>

The <species_def> defines all the chemical species used in the reactions and the number of linking sites for each. It is a sequence of one or more <sdef>s separated by commas and terminated by a period. Each <sdef> gives the name of the species followed by the number of links.

3.2 <reaction_set>

A <reaction_set> defines one or more <reaction>s that can take place from a given set of reactants specified as a <network>. Each possible reaction is specified by a <reaction>, the set of them being separated by semicolons and terminated by a period.

3.3 <reactant_spec>

Each <reactant_spec> describes the initial concentration of a reactant. The reactant is described by a <network> and the concentration by an <int>.

3.4 <template_spec>

A <template_spec> gives a name to a molecular structure pattern described by a <network>.

3.5 <network>

A <network> is specified by <node_defs>, which gives names to the nodes, and a <complex>, which described their interconnections (links).

3.6 <node_defs>

The <node_defs> defines all the nodes that appear in the <complex> (interconnection pattern) defining the <reactants>. Each <ndef> (node definition) is a node <name> optionally followed by a colon and a species <name>. If the species name is present, then the node must be of that type; if it is omitted, then it can be of any type. For example, “b: A, x” means that node “b” is of type “A” and node “x” is of any type. Species <name>s may be omitted in the <node_defs> in molecular reaction and template specifications (<mrs_file> and <mts_file>), since in both of these cases a <network>

represents a pattern, and so ``wild cards'' make sense. However, the species <name> must be specified in all the <node_defs> in a molecular program specification (<mps_file>), since each node must be of some type.

3.7 <complex>

A <complex> defines a graph (molecular network) by specifying the links between the nodes defined in the <node_defs>. The list of <link>s is separated by commas. Each <link> is a pair of <bond>s, specifying that the link goes from the first <bond> to the second.

3.8 <bond>

A <bond> specifies a binding site on a node. The binding sites are numbered from 0. For example, if **A**-species nodes are specified to have three sites ("A 3" in the <species_def>), and node b has been defined to be of type A ("b: A" in the <node_defs>), then the legal <bond>s for b are b_0, b_1, and b_2. If the "_<int>" is omitted, then "_0" is assumed; for example, b and b_0 mean the same thing. (By convention, the 0 site is used for the result-link in a node representing an operation in an expression tree.)

3.9 <reaction>

A <reaction> specifies a possible outcome from the <reactants> described at the beginning of the <reaction_set>. The result of the reaction is a graph, which is described by a <complex>. Each <reaction> has a <name>, which is used for reporting reactions and for similar purposes. Reactions may also have other parameters. The only one defined at this time is "p=<int>", which gives the probability (as a percent) that the reaction takes place.

3.10 <prime>

Names can have <prime>s on them; they have no meaning, beyond being a way of generating more names. For example, b, b', and b'' are just three different names.

4 Examples

4.1 Molecular Reaction Specification (*mrs file*)

The following are the definitions of the **K** and **S** reactions. For the sake of the example, I've included an incorrect **K** reaction that takes place 10% of the time.

```
% Species Definitions
A 3, P 1, D 1, K 1, R 3, Q 1, S 1.
% K = K combinator, S = S combinator, A = application,
% D = delete, R = replicate, P = arg cap, Q = result cap.

% K Reactions
a: A, b: A, k: K, d: D, q: Q, u, x, y.
u a, a_1 b, b_1 k, b_2 x, a_2 y, d q
=> (Kreaction, p=90)
    u x, d a, a_1 b, b_1 k, b_2 q, a_2 y;
=> (BadKreaction, p=10)
    u y, d a, a_1 b, b_1 k, b_2 x, a_2 q.

% S Reaction
a: A, a': A, a'': A, s: S, r: R, p: P, p': P, q: Q,
    u, x, y, z.
u a, a_1 a', a'_1 a'', a''_1 s, a''_2 x, a'_2 y, a_2 z,
    p r_1, p' r_2, r q
=> (Sreaction)
    u a, a_1 a', a'_1 x, a'_2 r_1,
        a_2 a'', a''_1 y, a''_2 r_2,
        r z,
    p s, p' q.
```

4.2 *Molecular Program Specification (mps file)*

```
% Initial Reactants for Test of K Substitution
% Tests (rooted on P) KSK => S

% PKSK
1* a: A, b: A, k: K, d: D, q: Q, u: P, x: S, y: K.
   u a, a_1 b, b_1 k, b_2 x, a_2 y, d q.

% Other reaction resources

% DQ
1* d: D, q: Q.  d q.
```

4.3 *Molecular Template Specification (mts file)*

```
% Template to Test for Result of K Substitution
% Tests result of (rooted on P) KSK => S

% PKSK => PS
PS = p: P, s: S.  p s.
```