Model building for Parsybone
Version 2.0

Adam Streck
Discrete Biomathematics, FU Berlin

November 14, 2013

1 Modeling

Models are described using an internal modeling language, based on the XML syntax [?], called *PMF* (Parsybone model file). The model is provided within a single PMF file that holds specification of the regulatory network.

For description of desired properties of the dynamical system a second file type, also based on the XML, is used. Predictably the name of the format is PPF (Parsybone property file).

Both the files must abide by the general XML rules and be provided as runtime arguments with their suffixes corresponding to their data type, i.e. with the *.pmf* and *.ppf* suffixes.

1.1 Model example

Every model must be enclosed within a pair `NETWORK` tag. A detailed description of the modeling language is provided later in this section, here we present, as an example, a model file for the network depicted in the *introductory pdf*.

This example model has a quite non-uniform syntax, which has been chosen on purpose to present different possibilities of model description.

```
<NETWORK>
    <SPECIE undef="basal" name="cA">
        <REGUL source="B" threshold ="1" label="Observable" />
        <REGUL source="cA" />
    </SPECIE>
    <SPECIE>
        <REGUL source="cA" />
        <REGUL source="B" label="+" />
        <PARAM context="" value="?" />
        <PARAM context="B" value="1" />
        <PARAM context="cA,B" value="0,1" />
    </SPECIE>
</NETWORK>
```

As can be seen, the model is a structure with two species, both being affected by two regulations. For the component B, the possible parametrizations space is reduced by requirements that the regulation from cA must be observable and that the effect of its self-regulation must be positive, if any. Also, the logical parameter of self-regulation of the component B must always be 1. As a result, the parametrization space is reduced to four possibilities.

1.2 Model property

The main purpose of the tool is picking parametrizations that satisfy some property. The description of this property can be given in one of two possible ways - either as Büchi automaton or as a time series. A time series is merely a Büchi automaton specialization, but as will be explained later the Parsybone is optimized for its usage and provides additional features if the time series is employed. To demonstrate the difference between the two, we present a single property described using each formalism.

This property assures that the model depicted in the complementary *introductory pdf* is able to reproduce a time series composed of the following three measurements:

1. $cA = 0 \wedge (B = 0 \vee B = 1)$
2. $cA \Leftrightarrow B$
3. $cA = 1 \wedge B = 0$

Only two out of four parametrizations allow for reproduction of this time series. To obtain them, we can describe the time series either using the Büchi automaton:

```
<AUTOMATON>
    <STATE final="0">
        <EDGE target="0" label="tt" />
        <EDGE target="1" label="cA=0" />
    </STATE>
    <STATE>
        <EDGE target="1" label="tt" />
        <EDGE target="last" label="((cA=0 &amp; B=0) | (cA=1 &amp; B=1))" />
    </STATE>
    <STATE name="last">
        <EDGE target="last" label="tt" />
        <EDGE target="3" label="(cA=1 &amp; B=0)" />
    </STATE>
    <STATE final="1">
        <EDGE target="3" label="tt" />
    </STATE>
</AUTOMATON>
```

Or using the time series directly:

```
<SERIES>
    <EXPR values="cA=0" />
    <EXPR values="((cA=0 &amp; B=0) | (cA=1 &amp; B=1))" />
    <EXPR values="(cA=1 &amp; B=0)" />
</SERIES>
```

As can be seen, the second method makes the model quite shorter and should be used for description of time series.

1.3 Regulatory network description

- `NETWORK`
  - Occurrence: single, mandatory.
  - Type: pair.
  - Parent: none, is a root node.
  - Description: encloses the definition of a regulatory network.
  - Attributes: none.

- `CONSTRAINT`
  - Occurrence: multiple, optional.
  - Type: solo.
  - Parent: NETWORK.
  - Description: specifies additional static constraint on the parametrization space.
  - Attributes: none.
    1. *type*
       * Occurrence: mandatory.
       * Value: force_extremes/bound_loop.
       * Description: a nature of the constraint, for details see Section 1.7.

- `SPECIE`
  - Occurrence: multiple, mandatory.
  - Type: pair.
  - Parent: NETWORK.
  - Description: defines a single specie.
  - Attributes:
    1. *name*

        - Occurrence: optional.
        - Value: a string containing only digits, letter and underscore.
        - Default: Capital letter, starting from `A`.
        - Description: name of the specie under which it will be further addressed.

    2. *undef*
        - Occurrence: optional.
        - Value: basal/param/error.
        - Default: param.
        - Description: tells the system how it should handle values of regulatory contexts that are not specified. Basal means using a basal value, param means using all possible values and error causes error in case there are unspecified parameters.
    3. *max*
        - Occurrence: optional.
        - Value: natural number.
        - Default: 1.
        - Description: maximal activation level this specie can occur in.
    4. *basal*
        - Occurrence: optional.
        - Value: positive integer.
        - Default: 0.
        - Description: basal activation level of this specie - the value towards which the specie tends if not specified otherwise.

- `REGUL`
    - Occurrence: multiple, mandatory.
    - Type: solo.
    - Parent: SPECIE.
    - Description: defines a single incoming regulation of the parent specie.
    - Attributes:
        1. *source*
            - Occurrence: mandatory.
            - Value: name or the ordinal number of a specie.
            - Description: name of the specie that regulates this one.
        2. *threshold*
            - Occurrence: optional.
            - Value: natural number.
            - Default: 1.
            - Description: lowest activation level of the source specie that activates the regulation.
        3. *label*

    - ∗ Occurrence: optional.
    - ∗ Value: a string, see Sec. 1.6.
    - ∗ Default: Free.
    - ∗ Description: describes nature of the regulation.

- `PARAM`
  - Occurrence: multiple.
  - Type: solo.
  - Parent: SPECIE.
  - Description: defines a single kinetic parameter.
  - Attributes:
    1. *context*
       - ∗ Occurrence: mandatory.
       - ∗ Value: comma separated list of active regulations, given by a name or an ordinal number of a regulator.
       - ∗ Description: defines the exact regulatory context in which this kinetic parameter is applied.
    2. *value*
       - ∗ Occurrence: optional.
       - ∗ Value: positive integer or ?.
       - ∗ Default: ?.
       - ∗ Description: specifies target value of the specie in this regulatory context, which must be one of possible activation levels of the specie. Character ? means that an exact number is unknown and a parametrization for each value between 0 and the maximal activation level of the specie is created.

1.4 Büchi automaton description

- `AUTOMATON`
  - Occurrence: single, present if and only if there is no sibling SERIES tag.
  - Type: pair.
  - Parent: none, is a root node.
  - Description: encloses the decription of a Büchi automaton.
  - Attributes: none.

- `STATE`
  - Occurrence: multiple, mandatory.
  - Type: solo.
  - Parent: AUTOMATON.
  - Description: defines a single state of the automaton. The first state in the description is also considered to be the initial state of the automaton.

  - Attributes: none.

  1. *name*
     - Occurrence: optional.
     - Value: string containing letters and numbers.
     - Default: ordinal number of the tag, counting from zero.
     - Description: name of the specie under which it will be further addressed. System also uses its ordinal number (so the first state can be addressed using the name *0*).
  2. *final*
     - Occurrence: optional.
     - Value: Boolean.
     - Default: 0.
     - Description: specifies if the state is final (1) or not (0).

- `EDGE`
  - Occurrence: multiple, mandatory.
  - Type: solo.
  - Parent: STATE.
  - Description: defines an edge leading from the parent state.
  - Attributes:
    1. *target*
       * Occurrence: mandatory.
       * Value: name or the ordinal number of the target state.
    2. *values*
       * Occurrence: mandatory.
       * Value: logical formula (see Section 1.8), variables are atomic propositions (see Section 1.9).
       * Description: conditions that must be met for the edge to be transitive.
    3. *stable*
       * Occurrence: optional.
       * Value: Boolean.
       * Default: 0.
       * Description: If set to (1), the transition satisfying the edge must be a loop.
    4. *transient*
       * Occurrence: optional.
       * Value: Boolean.
       * Default: 0.
       * Description: If set to (1), the transition satisfying the edge must be between different states.

1.5 Time series description

- `SERIES`
  - Occurrence: single, present if and only if there is no sibling AUTOMATON tag.
  - Type: pair.
  - Parent: MODEL.
  - Description: encloses definition of a time series.
  - Attributes: none.

- `EXPR`
  - Occurrence: multiple, mandatory.
  - Type: solo.
  - Parent: SERIES.
  - Description: a single measurement in the time series.
  - Attributes:
    1. *values*
       * Occurrence: mandatory.
       * Value: logical formula (see Section 1.8), variables of the formula must be atomic propositions (see Section 1.9).
       * Description: conditions that must be met for the measurement to be reproduced.
    2. *stable*
       * Occurrence: optional.
       * Value: Boolean.
       * Default: 0.
       * Description: If set to (1), the transition satisfying the edge must be a loop.
    3. *transient*
       * Occurrence: optional.
       * Value: Boolean.
       * Default: 0.
       * Description: If set to (1), the transition satisfying the edge must be between different states.

1.6 Edge label

There are two basic labels:

+ Meaning that the there must be a regulation whose parameter value increases if we add this regulator.

- Has the opposite meaning.

One can compose these labels using a logical formula over $+$ and $-$ or use one of the following predefined descriptions:

- Activating: $+$
- ActivatingOnly: $(+ \wedge \neg -)$
- Inhibiting: $-$
- InhibitingOnly: $(- \wedge \neg +)$
- NotActivating: $\neg +$
- NotInhibiting: $\neg -$
- Observable: $(+ \vee -)$
- NotObservable: $(\neg + \wedge \neg -)$
- Free: *true*

1.7 Static constraints

Constraints that globally restrict the parametrization space based on certain assumptions.

1.7.1 bound_loop

This constraint serves purely to optimization of multi-valued models and should be always used, unless there is a reason not to.

In multi-valued models, a scenario where multiple different parameters spawn the same structure may occur, simply because the values lay outside the activity levels for the respective context. This constraint leaves only a single representative of this behavior.

1.7.2 force_extremes

This constraint simply assigns an exact target value in two cases:

- 0 is assigned for the case when all positive regulators are absent and all negative are present.
- *max* is assigned for the case when all negative regulators are absent and all positive are present.

For this property, an unambiguous edge constraint must be assigned - Observable, NotObservable and Free edge constraints will cause the force_extremes switch to have not effect for the regulated component.

1.8 Formula construction

A formula is constructed using the following set of recursive rules:

1. tt and ff are formulas representing true or false respectively,
2. any atomic proposition is a formula,
3. for every formula A is $\neg A$ a formula,
4. for formulas A, B are $(A|B)$ and $(A\&B)$ formulas representing logical disjunction and conjunction respectively,
5. nothing else is a formula.

Note that in the model $\neg$ is denoted using `!` and for compliance with XML, & is denoted using `&amp;`.

1.9 Atomic propositions

An atomic proposition is a string of the form: $specie * value$, where:

- *specie* denotes the name or ordinal number of a specie,
- $*$ denotes comparison operator from the set $\{<, >, =\}$,
- *value* denotes a positive integer with which the value of the specie is compared.

Note that in the model, for compliance with XML, $<$ is denoted using `&lt;` and $>$ is denoted using `&gt;`.