

Vaucanson XML format description

Author: Valentin David
Contact: valentin@lrde.epita.fr
Date: December 2003

This document describes the [Vaucanson](#) XML format.

Contents

[Namespace](#)

[Global format](#)

[Type description](#)

[Monoid](#)

[Semiring](#)

[Content](#)

[States](#)

[Transitions](#)

[Initial states and final states](#)

[Geometry](#)

[Session](#)

Vaucanson XML obeys to W3C XML 1.0 recommandations.

Namespace

The namespace of the Vaucanson XML format is <http://www.lrde.epita.fr/vaucanson>.

Global format

An automaton is described by its type and its content. The global format looks like this:

```
<automaton>
  <type>
    <!-- type description -->
  </type>
  <content>
    <!-- content description -->
  </content>
</automaton>
```

Type description

Labels on transitions of automata are elements of rational series. This serie is built on a monoid an a semiring. They have both to be defined. We can have something like that:

```
<type>
  <monoid type="free" generators="letters">
    <generator value="A"/>
    <generator value="B"/>
  </monoid>
  <semiring set="B" operations="boolean"/>
</type>
```

Monoid

Monoids are defined with a **type** attribute and a **generators** one. Generators have to be passed as children. **type** can be set to “free” or “unit”. **generators** can be “letters”, “pair”, “weighted” or “integers”.

Here are some example of monoids:

```
<monoid type="free" generators="pair">
  <generators value="(a,a)"/>
  <generators value="(a,b)"/>
  <generators value="(b,a)"/>
  <generators value="(b,b)"/>
</monoid>

<monoid type="free" generators="weighted">
  <generators value="1x"/>
  <generators value="2y"/>
  <generators value="3z"/>
</monoid>

<monoid type="free" generators="integers">
  <generators value="1"/>
  <generators value="2"/>
  <generators value="4"/>
  <generators value="8"/>
</monoid>
```

Semiring

Semiring is defined with two attributes : **set** and **operations**. **set** describes the set where the semiring is defined, and **operations** define the operators used. **set** can be “B”, “Z”, “R”, or “ratseries”. When using simple sets, this **operation** attribute can be “boolean”, “numerical”, “tropicalMax” or “tropicalMin”. Here is an example:

```
<semiring set="Z" operations="tropicalMin"/>
```

When the semiring is a “ratseries” one, a **semiring** and a **monoid** have to be give as children. Then the **operations** attribute can be set to either “function” or “hadamard” or “shuffle”. Here is an example:

```
<semiring set="ratseries" operations="function">
  <monoid type="free" generators="letters">
    <generator value="A"/>
```

```

    <generator value="B"/>
    <generator value="C"/>
</monoid>
<semiring set="Z" operations="numerical"/>
</semiring>

```

Content

The content is divided in four parts:

- the states
- the transitions
- the initial states
- the final states

Each of these parts are lists of elements.

States

States are mainly described by a name. This name is require and is unique. According to the XML 1.0 recommandation, this name must begin with a alphabetic letter. A optionnal label attribute can be set.

Transitions

Transitions must refer to states as source and destination. The label is give with a regular expression, spontaneous by default.

Initial states and final states

Like transition but there is only one reference to a state.

Here is an example of content:

```

<content>
  <states>
    <state name="a"/>
    <state name="b"/>
  </states>
  <transitions>
    <transition src="a" dst="b" label="(2 A)*"/>
  </transitions>
  <initials>
    <initial state="a"/>
  </initials>
  <finals>
    <final state="b"/>
  </finals>
</content>

```

Geometry

Geometry can be passed on all stages by a `geometry` element. The geometry is conserved to all descendant node. Geometry attribute are mainly taken from Vancanson-G project. See the DTD for more informations.

Here is an example:

```
<automaton>
  <geometry
    ZZSize="1cm"
  />
  <type>
  <content>
    <states>
      <state name="a">
        <geometry
          x="0"
          y="0"
        />
      </state>
      <state name="b">
        <geometry
          x="2"
          y="0"
        />
      </state>
    </states>
    <transitions>
      <geometry
        curvature="edge"
      />
      <transition src="a" dst="b" label="(2 A)*"/>
    </transitions>
    <initials>
      <initial state="a">
        <geometry
          direction="W"
        />
      </initial>
    </initials>
    <finals>
      <final state="b">
        <geometry
          direction="E"
        />
      </final>
    </finals>
  </content>
</automaton>
```

Session

Severals automata can be saved into the same XML document with sessions. It is just a list of automata:

```
<session>
  <automaton name="automanton_1">
    <!-- definition of automanton_1 -->
  </automaton>
  <automaton name="automanton_2">
    <!-- definition of automanton_2 -->
  </automaton>
</session>
```