Java Solution Checker for XCSP3
Version 1.1

Gilles Audemard and Christophe Lecoutre

CRIL CNRS, UMR 8188
University of Artois, France

{audemard,lecoutre}@cril.fr

April 10, 2019

The version 1.1 of the Java Solution Checker for XCSP3 recognizes all constraints of XCSP3-core. It can check if a given instantiation is a solution, and if a given cost associated with the solution is correct.

For example, for the following XCSP3 instance called AllInterval-005.xml:

```
<instance format="XCSP3" type="CSP">
  <variables>
    <array id="x" note="x[i] the ith value of the series" size="[5]"> 0..4 </array>
    <array id="y" note="y[i] the dist. from x[i] to x[i+1]" size="[4]"> 1..4 </array>
  </variables>
  <constraints>
    <allDifferent> x[] </allDifferent>
    <allDifferent> y[] </allDifferent>
    <group class="channeling">
      <intension> eq(%0,dist(%1,%2)) </intension>
      <args> y[0] x[0] x[1] </args>
      <args> y[1] x[1] x[2] </args>
      <args> y[2] x[2] x[3] </args>
      <args> y[3] x[3] x[4] </args>
    </group>
  </constraints>
</instance>
```

one can check that if the following instantiation is a solution (this is the case).

```
<instantiation>
  <list> x[] y[] </list>
  <values> 4 0 3 1 2 4 3 2 1 </values>
</instantiation>
```

Remark 1 *The attribute* `type`*, which can be set the value* `"solution"` *or* `"optimum"`*, is not required. Anyway, the checker will never test if a specified instantiation represents an "optimal" solution (this is not a polynomial task).*

There are three ways of checking a solution (and possibly its cost). The first one is:

```
java org.xcsp.parser.callbacks.SolutionChecker <instanceFileName> <solutionFileName>
```

For example, if the file "sol.txt" contains the element `<instantiation>` above, then we can just write:

```
java org.xcsp.parser.callbacks.SolutionChecker AllInterval-005.xml sol.txt
```

and we obtain:

```
LOG: Check variables
LOG: Check constraints
LOG: Check objectives
VALID Solution!
```

It is also possible to put the element `<instantiation>` as second argument of the command line under the form of a String (framed with double quotes), which gives:

```
java org.xcsp.parser.callbacks.SolutionChecker  AllInterval-005.xml
   "<instantiation>
     <list> x[] y[] </list>
     <values> 4 0 3 1 2 4 3 2 1 </values>
    </instantiation>"
```

The third usage is to run its own solver that outputs elements `<instantiation>` that can be checked by redirecting (under Linux) the output with a |. For example:

```
java abscon.Resolution AllInterval-005.xml | java org.xcsp.parser.callbacks.SolutionChecker
  AllInterval-005.xml
```

Remark 2 *Note that we have to indicate twice the file name of the instance.*