Introduction

The family of Rebeca languages includes Rebeca, Timed Rebeca, Probabilisitic Rebeca, and Probabilistic Timed Rebeca. The user can model a system in one of the mentioned languages depending on the system under study and the user's needs. The model-checking tool RMC is used to analyze a Rebeca (or its extensions) model.

RMC is used to convert the input Rebeca model to a set of C++ files. Then, generated C++ files are compiled to an executable file. Running the executable file applies the model checking algorithm and generates the verification result. The generated state space may be saved in an XML file named "statespace.xml".

In the following parts, we explain how RMC can be used for the analysis of a Rebeca model.

How to generate C++ files

All the required libraries for generating C++ files are included in a Java executable Jar file "org.rebecalang.rmc-2.6.0-SNAPSHOT.jar". You can run this file with JRE 1.7 or upper using the following parameters.

Parameter	Example	Description
-s, --source	-s /home/test/model.rebeca -s myModel.rebeca	Location of Rebeca source file
-p, --property	-p /home/test/model.property -p myModel.property	Location of Rebeca model property file Not applicable for Probabilistic Timed Rebeca model
-o, --output	-o /home/test/modelFolder -o myModelFolder	Target of generated C++ file
-v, --version	-v 2.0 -v 2.1	Compiler version which could be Rebeca 2.0 or Rebeca 2.1. Features of Timed Rebeca, Probabilistic Rebeca, and Probabilistic Timed Rebeca are enabled in version 2.1.
-x, --exporttransitionsystem	-x	Exporting the state space in XML format in "statespace.xml" file. The exported state space can be visualized using "state space analysis" library.
-e, --extension	-e CoreRebeca -e TimedRebeca -e ProbabilisticRebeca -e ProbabilisticTimedRebeca	Specifying the type of Rebeca model
-h	-h	Print the parameters description

Finally two examples of typical commands for model checking Rebeca and Timed Rebeca models are depicted in the following. Note that the only mandatory parameter in calling rmc is the location of the Rebeca file and the other parameters are automatically filled with default values.

- `java -jar org.rebecalang.rmc-2.6.0-SNAPSHOT.jar –s model.rebeca -p model.property -o rmc -x`
- `java -jar org.rebecalang.rmc-2.6.0-SNAPSHOT.jar -s timed-model.rebeca -o rmc -v 2.1 -e TimedRebeca -x`

How to execute the generated C++ files

The result C++ files can be compiled using any distribution of C++ compilers. In the following example we use g++ to compile the generated C++ files and set the compilation output file to “executable”.

```
g++ *.cpp -w -o executable
```

Running “executable” file results in model checking of the model. The model checking result is reported on the console. The result includes the number of states and transitions in the generated state space, and the status of default properties. Default properties includes deadlock-freedom, no assertion failing, no queue-overflow, and no deadline-missed. If the given model does not satisfy the default properties, a counter example will be printed on the console.

The model checking result can be saved into a file instead of printing out on console by the following command:

```
executable -o outputName.xml
```

Note that the name of output file, i.e. “outputName”, shouldn’t be “statespace” as it is the default name for the generated state space.

Some notes

- Deadlock-freedom, no assertion failing, and no queue-overflow are three types of default properties which are checked automatically. In case of Timed Rebeca and Probabilistic Timed Rebeca, no deadline-missed is also checked automatically. The model checking result are shown between open and close "result" tag. For example, "<result>deadline missed</result>" shows that “deadline missed” happened, or "<result>satisfied</result>" means that all default properties were satisfied.
- In case of Probabilistic Timed Rebeca, running RMC checks the default properties and generates the state space. But, to model check PCTL properties, the generated state space can be verified by a back-end model checker like PRISM or IMCA. The option “-p myModel.property“ in RMC is not applicable in this case.
- In case of using arrays or actor-types variables, you can enable safety mode to avoid access to null valued variables and the elements outside of the arrays using “-safemode” parameter. Note that this reduces the performance of the tool.

- Using “-x” parameter, the model checker exports the state space of its given model in “statespace.xml” file. This file can be visualized using state-space-transformer library. You can download it from http://rebeca.cs.ru.is/files/statespacetransformer-1.0.0-SNAPSHOT.jar. The parameters of this library should be set as the following. The result file can be transformed to a jpg or png file using graphviz toolset (http://www.graphviz.org).

Parameter	Example	Description
-s, --source	-s /home/test/statespace.xml -s statespace.xml	Location of the xml file of the state space
-o, --output	-o /home/test/modelFolder -o myModelFolder	Target of the transformation
-e, --extension	-e CoreRebeca -e TimedRebeca	Specifying the type of the Rebeca model which the state space is corresponding to
-t	-t GRAPH_VIZ	Transforming the state space into graph-viz like format.
-h	-h	Print the parameters description