Causally Deterministic Markov Decision Processes

Tobias Meggendorfer

Building

Extract the code by unpacking `sources.zip`. Run `./gradlew compileJava` to compile all code and `./gradlew distZip` to compile and assemble a runnable package in `build/distributions`.

Running

Unpack the provided `cmdp.zip` (or build it yourself with the above commands). Run the tool with `<unpacked folder>/bin/cmdp <arguments>`. The following arguments are available:

- `--model <path>` path to PRISM model
- `--constants <comma separated list>` PRISM constants
- `--property <property>` the local reachability query to check
- `--assume-cd` do not check for causal determinacy via syntactic checks, simply assume the property holds
- `--render` render the strategy to `out.png` (you need to have Graphviz / `dot` installed)

The tool then performs syntactic checking whether the input model is CD and then computes the maximal reachability probability. The tool has some limitations (for example, global variables are not yet supported).

Evaluation

To run the instances from the paper, unpack the `models.zip` into the folder `models` and use the following commands:

- sync: `bin/cmdp --model models/sync.prism --property "(s01=2)"`
- dining philosophers: `bin/cmdp --model models/phil_think.prism --property "(s1=happy)" --assume-cd`
- production: `bin/cmdp --model models/production.prism --property "(made>0)"`
- scheduling: `bin/cmdp --model models/scheduling.prism --property "(state=10)"`
- scheduling_large: `bin/cmdp --model models/scheduling_large.prism --property "(state=20)"`

Structure

All relevant code is located in `src/main/java/de/cmdp`. `Main.java` contains the overall algorithm and `ModuleGraph` contains the representation of the MDP's factors / modules. All other classes are standard probabilistic model checking concepts and taken from the Partial Exploration Tool. The `lib/` folder contains the relevant files of PRISM used to parse its models.