SReach: Usage

Qinsi Wang

Computer Science Department, Carnegie Mellon University, USA

1 The SReach tool

1.1 Input format

The inputs to our **SReach** tool are descriptions of hybrid automata with random variables (representing the probabilistic system parameters), and the reachability property to be checked. Following roughly the same format as the above definition of hybrid automata, and adding the declarations of random variables, the description of a automaton is of the following structure.

Preprocessor. We can use the C language syntax to define constants and macros. When defining, random variables, which will be declared later, can also be used.

Variable declaration. For a random variable, the declaration specifies its distribution and name. For the variables which are not random variables, they are required to be declared within bounds.

Hybrid automaton. A hybrid automaton is represented by a set of modes. Within each mode declaration, users can specify statements for mode invariant(s), flow function(s), and jump condition(s). For a mode invariant, we can give any logic formula for the variables. For a flow function, it is expressed by an ODE. As for a jump condition, it is written as

```
<logic_formula1>  ==> @<tagert_mode>  <logic_formula2>,
```

where the first logic formula is given as the guard of the jump, and the later one specifies the reset condition after the jump.

Initial conditions and Goals. Following the declaration of modes, we can declare one initial mode with corresponding conditions, and the reachability properties in the end.

Example 3.1. The following is an example input file. Currently, users can specify random variables with Bernoulli distribution, Uniform distribution, Gaussian distribution, and Exponential distribution. (Note: it is easy to include additional distributions if needed.)

```
#define pi 3.1416
N(1,0.1) mu1;
U(10,15) thro;
E(0.49) theta1;
B(0.75) xinit;
[0,5] x;
```

```
[0,3] time;
{ mode 1;
  invt:
         (x<=1.5);
         (x>=0);
  flow:
         d/dt[x]=thro*(1/(theta1*sqrt(2*pi)))
                    *exp(0-((x-mu1)^2)/(2*theta1^2));
  jump:
        (x>=(thre1+5))==>@2(x’=x);
}
init:
@1      (x=xinit);
goal:
@4      (x>=50);
```

1.2 Command line

After building, **SReach** can be simply used through:

`SReach <statistical_testing_option> <filename> <dReach> <k> <delta>`

where:

- `statistical_testing_option` is a text file containing a sequence of test specifications. We will introduce the usages of statistical testing options in the following part;
- `filename` is a .pdrh file describing the model of a hybrid system with probabilistic system parameters. It is of the input format described in last subsection;
- `dReach` is a bounded reachability analyzing tool for hybrid systems based on dReal;
- `k` is the number of steps of the model that the tool will explore; and
- `delta` is the precision for the δ-decision problem.

1.3 Statistical testing options

SReach can be used with different statistical testing methods through the following specifications.

Lai's test: `Lai <theta> <cost_per_sample>`, where `theta` indicates the probability threshold.

Bayes factor test: `BFT <theta> <T> <alpha> <beta>`, where `theta` is a probability threshold satisfying `0 < theta < 1`, `T` is a ratio threshold satisfying `T > 1`, and `alpha`, and `beta` are beta prior parameters.

BFT with indifference region: `BFTI <theta> <T> <alpha> <beta> <delta>`, where, besides the parameters used in the above Bayes factor test, `delta` is given to create the indifference region - $[p_0, p_1]$, where $p_0 =$ `theta` + `delta` and $p_1 =$ `theta - delta`. Now, it tests $H_0 : \ p \geq p_0$ against $H_1 : \ p \leq p_1$.

Sequential probability ratio test (SPRT): `SPRT <theta> <T> <delta>`.

Chernoff-Hoeffding bound: `CHB <delta1> <coverage_probability>`, where `delta1` is the given precision, and `coverage_probability` indicates the confidence.

Bayesian Interval Estimation with Beta prior:
`BEST <delta1> <coverage_probability> <alpha> <beta>`.