Cosmos User Manual

Benoît Barbot

October 17, 2011

This is the user manual for the tool Cosmos version 1.

1 File Format

1.1 Generalized Stochastic Petri Net (.gspn)

This file format is used to describe GSPN. First we describe an example: This GSPN is described by the following text:

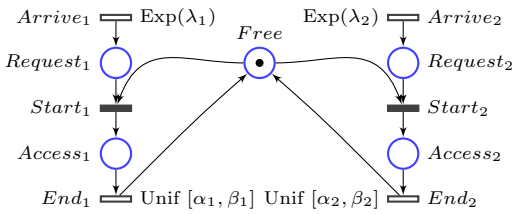

Figure 1: Infinite-state GSPN model of a shared memory system.

```
const double lambda1 = 1;
const double lambda2 = 2;
const double alpha1 = 1;
const double alpha2 = 1;
const double beta1 = 5;
const double beta2 = 5;

NbPlaces = 5;
NbTransitions = 6;

PlacesList = {
   Request_1, Request_2,
   Access_1, Access_2,
   Free
```

```
} ;

TransitionsList = {
   Arrive_1,Arrive_2,
   Start_1 ,Start_2,
   End_1   ,End_2
} ;

Marking={
   (Request_1 , 0); (Request_2 , 0) ;
   (Access_1 , 0) ; (Access_2 , 0) ;
   (Free, 1);
};

Transitions={
   (Arrive_1,EXPONENTIAL(lambda1),1,1, SINGLE);
   (Arrive_2,EXPONENTIAL(lambda2),1,1, SINGLE);
   (Start_1,DETERMINISTIC(0),1,1);
   (Start_2,DETERMINISTIC(0),1,1);
   (End_1,UNIFORM(alpha1,beta1),1,1);
   (End_2,UNIFORM(alpha2,beta2),1,1);
};

InArcs={
   (Request_1,Start_1,1); (Free,Start_1,1);
   (Request_2,Start_2,1); (Free,Start_2,1);
   (Access_1,End_1,1);
   (Access_2,End_2,1);
};

OutArcs={
   (Arrive_1,Request_1,1);
   (Arrive_2,Request_2,1);
   (End_1,Free,1);
   (End_2,Free,1);
};
```

Description: The first bloc is a list of constants definition, constants can be either `double` or `int`.
Then we specify the number of place and transitions with: `NbPlaces = 5; NbTransitions = 6;`
The list of place name and transition name is given in the `PlacesList` and `TransitionsList` statement.
The initial marking of the net is given as a set of pairs in the `Marking` statement.
The transition distribution is given as a set of tuples like this one:
`(Arrive_1,EXPONENTIAL(lambda1),1,1, SINGLE)` each tuple contain first the name of the transition then the probability distribution with some parameters,

then two positive reals define the priority and weight of the event generated. For exponential distribution we can specify the policy of service which can be `SINGLE,INFINITE,MULTIPLE(n)`.
Finally come the description of arcs of the net with the `InArcs`,`OutArcs` and `InhibitorsArcs` statements.

1.1.1 Grammar

the complete grammar is:

```
$accept: GSPN "end of file"

GSPN: declarations definitions
    | declarations definitions redifinitions

declarations: Constants Sizes Lists
            | Sizes Lists

Constants: Constant
         | Constant Constants

Constant: 'const' 'int' str '=' IntStringFormula ';'
        | 'const' 'double' str '=' RealStringFormula ';'

IntStringFormula: ival
                | str
                | '(' IntStringFormula ')'
                | IntStringFormula '+' IntStringFormula
                | IntStringFormula '-' IntStringFormula
                | IntStringFormula '*' IntStringFormula
                | IntStringFormula '^' IntStringFormula
                | FLOOR '(' IntStringFormula ')'
                | FLOOR '(' IntStringFormula '/' IntStringFormula ')'
                | MIN '(' IntStringFormula ',' IntStringFormula ')'
                | MAX '(' IntStringFormula ',' IntStringFormula ')'

RealStringFormula: rval
                 | ival
                 | str
                 | '(' RealStringFormula ')'
                 | RealStringFormula '/' RealStringFormula
                 | RealStringFormula '+' RealStringFormula
                 | RealStringFormula '-' RealStringFormula
                 | RealStringFormula '*' RealStringFormula
                 | RealStringFormula '^' RealStringFormula
                 | FLOOR '(' RealStringFormula ')'
```

```
                | MIN '(' RealStringFormula ',' RealStringFormula ')'
                | MAX '(' RealStringFormula ',' RealStringFormula ')'

Sizes: NbPlaces NbTransitions
     | NbTransitions NbPlaces

NbPlaces: 'NbPlaces' '=' ival ';'
        | 'NbPlaces' '=' str ';'

NbTransitions: 'NbTransitions' '=' ival ';'
             | 'NbTransitions' '=' str ';'

Lists: PlacesList TransitionsList
     | TransitionsList PlacesList

PlacesList: 'PlacesList' '=' '{' PLabels '}' ';'

PLabels: str
       | PLabels ',' str

TransitionsList: 'TransitionList' '=' '{' TLabels '}' ';'

TLabels: str
       | TLabels ',' str

definitions: PlacesDef TransitionsDef InArcs OutArcs
           | PlacesDef TransitionsDef InArcs OutArcs Inhibitors

PlacesDef: 'Marking' '=' '{' PLACES '}' ';'

PLACES: PLACE
      | PLACES PLACE

PLACE: '(' str ',' IntStringFormula ')' ';'

TransitionsDef: 'Transition' '=' '{' TRANSITIONS '}' ';'

TRANSITIONS: TRANSITION
           | TRANSITIONS TRANSITION

TRANSITION: '(' str ',' dist ',' PRIORITY ',' WEIGHT ')' ';'
          | '(' str ',' 'EXPONENTIAL' '(' RealStringFormula ')' ',' PRIORITY ','
            WEIGHT ',' SERVICE ')' ';'
          | '(' str ',' IMDT ',' PRIORITY ',' WEIGHT ')' ';'

dist: str '(' params ')'
```

```
params: RealStringFormula
      | params ',' RealStringFormula

WEIGHT: RealStringFormula

PRIORITY: RealStringFormula

SERVICE: 'SINGLE'
       | 'INFINITE'
       | 'MULTIPLE' '(' ival ')'
       | 'MULTIPLE' '(' str ')'

InArcs: 'InArcs' '=' '{' incells '}' ';'

incells: incell
       | incells incell

incell: '(' str ',' str ',' IntStringFormula ')' ';'
      | '(' str ',' str ')' ';'

OutArcs: 'OutArcs' '=' '{' outcells '}' ';'

outcells: outcell
        | outcells outcell

outcell: '(' str ',' str ',' IntStringFormula ')' ';'
       | '(' str ',' str ')' ';'

Inhibitors: 'inhibitor' '=' '{' inhibcells '}' ';'

inhibcells: inhibcell
          | inhibcells inhibcell

inhibcell: '(' str ',' str ',' IntStringFormula ')' ';'
         | '(' str ',' str ')' ';'
```