Implementation of DG^*

The PRISM model that implements DG^* uses three players: $p1, p2$ and $p3$. Players $p1$ and $p2$ are used to construct strategies σ_1^* and σ_2^* respectively while player $p3$ is used for synchronisation purposes. In our *stochastic multiplayer game* model (SMG) the weights of the graph's arcs are represented in form of variables in order to model effectively the enhancements of each player's actions and each enhancement is modelled with the use of separate variables. Distinct modules are constructed for $p1, p2$ and $p3$. Players' $p1$ and $p2$ modules ($player1moves$ and $player2moves$ respectively) implement their available action sets while $p3$'s module synchronizes the players' actions. A segment of $p1$'s module is stated in Listing 1.

```
module player1moves
        [p1_1_to_1] selectedn=1 & t=0 ->
                (e11'=e11+p1b1)&(t'=1);
        [p1_1_to_3] selectedn=1 & t=0 ->
                (e13'=e13+p1b1)&(t'=1);
        ...
endmodule
```

Listing 1: Segment of p1's module

In the presented segment of $p1$, the model provides two actions to $p1$ when the random walk visits node 1: actions `p1_1_to_1` and `p1_1_to_3`. Action `p1_1_to_1` favours the transition from node 1 to node 1 by enhancing factor $p1b1$ to the corresponding arc $e11$ while action `p1_1_to_3` favours the transition to node 3 in a similar manner (i.e., enhancement of arc $e13$). Variable t is used for synchronization purposes by the module of $p3$.

```
module player2moves
  [p2_1_to_1] selectedn=1 & t=1 ->
    ((e11+p2b1)/(n1+p2b1)):
      (selectedn'=1)&(e11'=100)&(e13'=100)&(t'=2) +
    ((e13)/(n1+p2b1)):
      (selectedn'=3)&(e11'=100)&(e13'=100)&(t'=2);
  [p2_1_to_3] selectedn=1 & t=1 ->
    ((e11)/(n1+p2b1)):
      (selectedn'=1)&(e11'=100)&(e13'=100)&(t'=2) +
    ((e13+p2b1)/(n1+p2b1)):
      (selectedn'=3)&(e11'=100)&(e13'=100)&(t'=2);
```

```
        ...
endmodule
```

Listing 2: Segment of p2's module

Listing 2 presents a segment of $p2$'s module. The player is provided with two actions when node 1 is visited: actions `p2_1_to_1` and `p2_1_to_2`. In `p2_1_to_1` player $p2$ chooses to enhance the probability of reaching node 1 by a factor $p2b1$ $(((e11 + p2b1)/(n1 + p2b1)))$ while in `p2_1_to_2` $p2$ enhances the arc of node 1 to 3 $e13$ by the same factor $p2b1$ $(((e13 + p2b1)/(n1 + p2b1)))$. Number $n1$ is the sum of all weights of node's 1 outgoing arcs before $p2$'s action and is computed with the use of a formula. $n1$ is used for the normalisation process of the probabilities. In each action, $p2$ reinitializes the weights of the affected arcs for future moves $((e11' = 100)\&(e13' = 100)\&(t' = 2))$.

The proper incentives for the construction of the players' strategies are expressed with the use of rPATL. The rPATL properties uses a reward structured named *op* similar to the models of DP^* and DIP^*. The property concerning D_2^*'s incentive is stated in Listing 3 and it can be restated as the maximum reward (i.e., opinion) that $p2$ (i.e., D_2^*) can achieve before the end of the random walk (*walkstop*).

```
<<p2>> R{"op"}max=? [ Fc "walkstop" ]
```

Listing 3: rPATL property for p2