

Implementation of *CDIP**

A proper action based reward structure named `cost` is added in order to implement the necessary functionality for the cost calculation of *D*'s strategy synthesis. In this reward structure, each action is determined by its label used in the construction of the model and the guards that enable this action. Listing 1 presents the first commands of the reward structure `cost` that refer to the actions related to the first node of the graph. When *D* chooses to tamper with node 1 (actions `n1_1_to_1` and `n1_1_to_3`) a 1 unit cost is applied while action `n1_donothing` costs 0.0 units.

```
rewards "cost"  
    [n1_1_to_1]      stopped=0 : 1 ;  
    [n1_1_to_3]      stopped=0 : 1 ;  
    [n1_donothing]   stopped=0 : 0.0 ;  
    ...  
endrewards
```

Listing 1: Action based reward structure for *CDIP*

Listing 2 presents the property that computes the Pareto curve for our *CDIP** example. The reserved word `multi` combines the two properties on the parenthesis. The first property queries the minimum consensus that *D** can achieve in a random walk in the graph of 1000000 steps while the second property refers to the minimum cost in the random walk. This combination of properties forms a *Pareto query* for PRISM since both of them are unbounded (i.e., no specific bound is used).

```
multi( R{"consensus"}min=? [ C<=1000000 ],  
       R{"cost"}min=? [ C<=1000000 ] )
```

Listing 2: Mutli-objective property for Pareto frontier computation of *CDIP**