List of eligible projects

1. A DSL to design conversational games

The objective of this project is to facilitate the definition of conversational games and automate their execution. In this type of games, the protagonist goes through a maze made of rooms, where s/he finds objects with which s/he can interact in different ways. Zork (http://en.wikipedia.org/wiki/Zork) was one of the first games of this style.

In this project, you will create a textual DSL to configure conversational games with the characteristics that are detailed below. In addition, you will create a program that, given the description of a game created with the DSL, is responsible for executing it.

Requirements

In a conversational game, there is a labyrinth of connected rooms where there can be objects and characters. The player gives instructions to the protagonist with simple phrases to move around the rooms, interact with other characters, and perform actions on the objects.

Below, you can see a moment in the execution of a game. The red text after the symbol > corresponds to phrases that the player writes, while the black text in italics is shown by the game engine:

```
1   You have entered a room with a door to the north and a door to the east. You
2   can see a potion and a magician.
3   The magician greets you: "Abracadabra"
4   > ask magician about potion
5   You ask the magician: "Should I drink this potion?"
6   The magician answers: "Yes, it will increase your health"
7   > take potion
8   Potion taken
9   > water potion
10  That action is not possible.
11  > drink potion
12  Potion drunk. Your level of health increases by 10.
13  > go north
14  You have entered a room. You can see a box.
15  > open box
16  The box does not open.
17  > open box with key
18  The box has been opened. It has a message inside.
19  > take message
    …
```

As the example shows (lines 1-3), when the protagonist enters a room, the game lists the doors the room has, the objects and characters in the room, and if there is some character, her/his greeting phrase.

The interaction with the characters is limited to ask about the objects that are in the room (line 4). The character may respond that s/he does not know anything about the object, that the protagonist should do some action on it (e.g., drink potion, as in the example, because it will increase the level of health), or that the protagonist should not make a certain action (e.g., do not drink the potion, if it reduces the level of health).

Only a limited number of actions can be performed on each object found, depending on the type of the object. For example, it is possible to take a potion (line 7) or drink a potion (line 11), but it is not possible to water a potion (line 9). For some of these actions, it is also necessary to indicate one or more additional objects that the protagonist owns. For example, to open a box, a key is required (lines 15 and 17), which the protagonist must have previously collected in another room. Performing an action on an object yields a result. For example, "take potion" adds the potion to the list of belongings of the protagonist, "drink potion" increases one of the characteristics of the protagonist (the level of health), and "open box with key" makes available a new object (message) that previously did not exist in the room.

To complete the game, the protagonist must achieve a predefined goal, such as reaching a room containing a treasure carrying a key and having a health level > 50. The game may also end unsuccessfully for various predefined reasons, e.g., when the health level of the protagonist reaches 0, or if the protagonist performs a certain action such as "drinking poison".

The DSL to be developed should allow the configuration of (at least) the following characteristics of a game:

- Rooms of the game and how they are connected.
- Properties of the protagonist (e.g., level of health) and initial value (e.g., level of health = 100).
- Existing objects in the game (e.g., key, box, potion, etc.).
- Admissible actions on each object (e.g., drink potion, open box with key, etc.), indicating what happens when doing it: a text is printed on the screen, a certain characteristic of the protagonist is modified, a new object becomes available, etc.
- Characters in the game (e.g., magician, jester, ghost, etc.), indicating how they greet when entering a room they are in (e.g., "Abracadabra"), as well as their response to questions about specific objects.
- Distribution of the elements of the game in the rooms.
- Conditions for ending the game, in case of success or failure.

In addition to the DSL, the project needs to include a program that runs the game based on its definition. The interaction with the program should be similar to the listing shown before (**but the game itself can be in Spanish**). You will probably need to write a code generator to synthesize configuration code to be used by the game engine.

Finally, the project should include a model transformation from the game design into Petri nets, to verify that winning/losing the game is feasible.