Language Engineering

Worksheet 1 :: Questions

1 Domain Specific Languages

1.1 Robot

1.1.1 Introduction

We have a very basic turtle robot which can move forward a specific amount (as an *Int*), rotate left and right 90^o as well as stop.

1.1.2 Questions

1. Design a recursive data strucuture (without using lists) which encodes the basic *Robot*. It should have four data constructors relating to the different operations it can do.
2. Write a function which calculates the total distance travelled by the robot, it should have the type given below:

   $$\mathit{distTrav} :: \mathit{Robot} \rightarrow \mathit{Int}$$

   $$\mathit{distTrav} = \bot$$

3. Design a function that calculates the distance travelled in the direction that the robot was first facing.
4. Design a function that calculates the distance the robot ends up away from its starting position, in a straight line, with the output as a *Float*.

1.2 Cooking Master

1.2.1 Introduction

We will attempt to model a basic set of recipes. However, to make life easier we shall initially only include potatoes. This will be done using a shallow embedding, where the semantic output will be the properties of the potatoes:

1. Time taken: The time it takes, as an *Int*, to prepare the potato .
2. Weight: The weight of potatoes prepared as an *Int*.
3. Cooked: Whether the potatoes have been cooked, as a *Bool*.
4. Description: Information about the potatoes as a *String*.

1.2.2 Questions

1. Create a value encoding a single potato, with the below type, which represents the various semantic outputs of a potato dish:

   $$\mathit{potato} :: (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String})$$

   The data type $(\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String})$ in this case relates directly to $(\mathit{time\ taken}, \mathit{weight}, \mathit{cooked}, \mathit{description})$, and *potato* will encode a single potato into this semantics. This means that the we would assume the time taken to be nothing, we will only have one potato with weight 3, it won't be cooked and the only description we can give will be "potato".

2. Create functions for each of the following different culinary expertises exercised on potatoes. They will need to map the current semantics to an updated semantics.

   $$\mathit{ce} :: (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String}) \rightarrow (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String})$$

   (a) peel (takes 2 mins for each potato and adds "peeled" to the description)

   (b) roast (takes 70 mins, makes them cooked and adds "roasted" to description)

   (c) boil em (takes 25 mins, makes then cooked and adds "boiled" to the description)

   (d) mash em (takes 1 min per potato and adds "mashed" to the description)

   (e) stick em in a stew (takes 120 mins, makes them cooked and adds "stewed" to the description)

3. Create a function that lets you mix two sets of potatoes, this should combine the time taken and weights, become uncooked if either is uncooked and combine the two descriptions. The type signature is given below:

   $$\mathit{mix} :: (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String}) \rightarrow (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String}) \rightarrow (\mathit{Int}, \mathit{Int}, \mathit{Bool}, \mathit{String})$$

4. Now allow for your next two favourite root vegtables to be added.

1.3 Languages

1. What is the difference between a GPL and a DSL?

2. What programming methods should be used in shallow embeddings and deep embeddings when dealing with dependent interpretation?