DiceySpicyScript Language Specification

Helen Qian   Maia Wang

May 13, 2024

Introduction

This programming language would help the user write recipes. This program would provide an efficient way to turn jots down on a notepad into an easy-to-read recipe. Writing recipes on a notepad or paper is easy for anyone to do. However, pencil can get smudged and ink can wash away when it gets wet. Papers can get lost, dirty, or torn. Thus, digitalizing recipes is a smarter way to preserve recipes, make them easily replicable, and nice to look at.

Sometimes it can be a struggle to work with Google Docs or excel to make a nice recipe and it may take more time than the average Joe may have. Furthermore, most people don't even know what Latex is or how to use it. Through this language, we hope people can simply write out their quick notes about ingredients and instructions, and we will transform their writing into nicely formatted recipes. No need to waste time trying to order your ingredients separate from your instructions. Simply as you are creating your Michelin star worthy dish just make a note of the ingredient or instruction as you go. Our language will take lay man terms and format them into a file of code that can simply be pasted into a latex compiler like overleaf, where one can obtain a nicely downloaded pdf of their recipe, ingredients and instructions all neatly organized.

Design Principles

Aesthetically we made choices on how the latex recipe template that is outputted. The ingredient list is bulleted and the instruction list is enumerated. We specifically choose the font Latin Modern Dunhill to give a cozy home baked feel to the recipe. The latex recipe also includes design elements like the ingredients and instruction lists being inboxed. We kept the recipe design simple and easy to read but still aesthetically pleasing.

Examples

Example 1:

```
tit[Strawberry Shortcake]
ing[1 cup flour, 1 egg, 2 strawberries]
ins[ add 1 egg to a bowl and mix, 30s
add 1 cup flour to bowl and whisk, 30s
chop strawberries]
```

Example 2:

```
ins[boil water, 20m cook pasta, 10m, add asparagus]
tit[Creamy Asparagus Pasta]
ing[1 lb short pasta, 1 lb asparagus, two tbsp unsalted butter]
```

Example 3:

```
ing[1 bunch lettuce]
ins[rinse and chop lettuce in a bowl]
ing[10 dates]
ins[chop dates in half and add to bowl]
ing[1 slice blue cheese, 1 cucumber, 2 firm tomatoes]
ing[crumble and add blue cheese to bowl, slice up cucumber and add to bowl, slice tomatoes into fourths or even
```

smaller pieces and add to bowl]
ing[Italian dressing]
ins[mix italian dressing evenly into salad, and enjoy!]
tit[Helen's easy salad]

In all of these examples you can see either the title, ingredient(s) or instruction(s) are specified then followed by opening and closing square brackets with content inside. When specifying, the title will be labeled with abbreviation tit, ingredient with ing, instruction with ins. To add a program that contains more than one ingredient or instruction under one label (ing [], ins []), add a comma after each content item.

Place your programs in a text file and run by inputting:

```
    dotnet run <example>.txt
```

Language Concepts

The user needs to understand the basic format of title, ingredients and instructions. They need to know how to format the code so it reads in–when to use commas versus new lines, when to use parentheses, etc. In terms of primitives, we have the title, ingredients, and instructions. Our language combines these primitives into a full recipe. For specifying titles, only the title specified latest will be taken in as the recipe name.

Formal Syntax

```
<Recipe> ::= <expr> | List of <expr>
<expr> ::= <Title> | <Ingredients> | <Instructions> | <expr><expr>
<Title> ::= "name of recipe"
<Ingredient> ::= "ingredient1, ingredient2, ingredient3" | "ingredient"
<Instruction> ::= "instruction"
```

Semantics

Syntax	Type	Meaning
title	string	a primitive that represents the title of a recip
ingredient	string	a primitive that represents an ingredient or set of ingredients
instruction	string	a primitive that represents an instruction or set of instructions in a recipe
recipe	string, string list, string list	a recipe containing a title, list of ingredients, and list of instructions

0.1 Remaining Work

To further enhance this language we could make the latex recipe template more personalizable. Like say the user would like to have a certain font or color. Maybe even going further and allowing them to choose from different templates. We played with the idea of letting them create a cookbook and are continuing to play on ideas of how they would specify a new recipe. In prior drafts we said the user would be able to create a new recipe by adding another label of a title, however we changed this because we wanted the user to be able to change the title if they misspecified it.

We mentioned providing an overall time for the recipe which would be added from the given times for each instruction. We ultimately decided against this though because that would mean changing the language entirely and require users to enter in time frames for the instructions. This kind of goes against our thought of writing an easy fast program as users would have to calculate the amount of time something took them. Not fun for when you are experimenting.

Something else we could also work more on is just returning a pdf. This way the user doesn't even have to put the output into latex, their just are given a pdf.