

Advanced Topics in Computer Science Course

Logic programming Project

April-June 2024

The aim of the project is to design and program a version of the french_menu predicate that is more complete than what has been done during lectures.

This text gives you indications on what is expected and how to proceed.

1. Method

The project will be done in 2 steps

1. First part for mid-term defense (until “vegetarian or vegan menus” included)
2. Revision of first part and completion of the whole project for final defense

The project will be assessed according to the following criteria

- Personal work
 - Grades of identical works will be divided by the number of copies
- Quality of the submitted code
 - Quality of the updates after the intermediate defense (how well the comments have been taken into account)
- **Quality of the tests**
- **Quality of the explanations during presentation**
- Quality of the slide presentation (to be done on line during the final defense)

In addition, the quality of the tests of the predicates done during the lecture can act as a bonus/malus (upload under “Students programs: *.pl”).

To produce the code, follow the steps of the project process section. **Test thoroughly each step before moving to the next one.** Remember to test true and false cases, as well as verification and generation cases. If you do not manage to make a step work, try to explain why in the test part and move to the next step.

You will upload **only the intermediate and the final version** of the code on the Moodle page of the course: one version only before each defense.

The tests should be included in a separate file.

Keep track of your steps because you will have to present them at the final defense.

2. Slideshow model

The slideshow for the final defense **must** contain 9 slides (not more not less, every section is mandatory)

Slide 1 Presentation of yourself; your cursus, which level; if you plan to go for an international mobility; date of the presentation; logo of TSU and INSA Rennes

Slides 2-6 1 slide per main project step: the essential of the version, how it has been tested, with the important comments on the slide

- New set of facts and the update of french_menu/2
- Vegetarian/vegan menus

- Wines
- Useful interface predicates
- User-friendly display

Slide 7 What was easy and what was difficult (both aspects must be covered)

Slide 8 What you have learnt during the course

Slide 9 How you think you will reuse the acquired knowledge

A pdf version of the slideshow must be inserted at the end of the test file and uploaded together with the program file.

Please, make a personal presentation.

3. Summary of what has been done during lectures

Initial version: Typically, a French menu consists of an appetizer, then a main course (meat, fish or vegetarian) and a dessert.

A valid menu

?- french_menu(salad, trout_with_rice, cake).

Invalid menus

?- french_menu(salad, trout, cake).

?- french_menu(falafel_with_rice, trout_with_rice, cake).

Update 1: Nowadays, however, people tend to eat less. Restaurants often offer the possibility to take either appetizer + main course, or main course + dessert, or appetizer + main course + dessert.

Valid menus

?- french_menu(salad, trout_with_rice).

?- french_menu(trout_with_rice, cake).

?- french_menu(salad, trout_with_rice, cake).

Update 2: Sometimes cheese can replace dessert, sometimes it is offered before dessert

Valid menus

?- french_menu(salad, trout_with_rice).

?- french_menu(trout_with_rice, roquefort).

?- french_menu(salad, trout_with_rice, roquefort).

?- french_menu(salad, trout_with_rice, roquefort, cake).

Update 3: Predicates with same name and different arities are a strong source of bugs, in particular when the program is updated to add new functionalities.

This update makes the french_menu predicate of arity 1 for all possible structures.

Valid menus

?- french_menu([salad, trout_with_rice]).

?- french_menu([trout_with_rice, roquefort]).

?- french_menu([salad, trout_with_rice, roquefort]).

?- french_menu([salad, trout_with_rice, roquefort, cake]).

Invalid menus

```
?- french_menu([trout_with_rice, salad]).
?- french_menu([salad, trout_with_rice, cake, cake]).
?- french_menu([salad, trout_with_rice, roquefort, cake, coffee]).
```

Update 4: In the previous version there is a lot of code duplication, also a strong source of bugs. Update your program for a strictly equivalent version with less duplication.

Update 5: We want to be able to build/verify balanced menus, in terms of Calories. This version computes the calories of a given menu. It did not matter for the exercise if the given calories were not correct.

4. Project process

1. Start a new file called <your_name>_project.pl, insert update 5 (and make sure it works)
2. **New set of facts:** In the appendix of this document, you will find a list of typical French courses, not exhaustive but much more complete than the one given during lectures.
 - i. Interpret this information to make a list of facts
 - ii. We want to distinguish vegetarian (neither meat nor fish) or vegan (no animal products) dish. Note that vegan => vegetarian. Add a new argument to facts to give this information.
 - iii. For main courses, it is asked to **separate the main component** (eg boeuf_bourguignon or falafel) **from the side dish**, “accompagnement” in French, (eg mashed_potatoes, gratin_dauphinois). Therefore boeuf_bourguignon with mashed_potatoes are two dishes instead of one as was the case in the versions of the lectures.
 - iv. There is always a main dish and a side dish in a menu.
 - v. Make different series of facts. Differentiate meat, fish, vegetarian and vegan main components. Consider seafood as fish.
 - vi. As in the previous version add information about calories. It does not have to be fully correct, but **try to make it consistent**. For example, a potatoes dish should have much more calories than a salad dish.
3. *Update of French_menu/2:* to take into account the new format of the facts
 - i. Carefully test it and keep the tests
4. *Vegetarian or vegan menus:* We want to be able to ask for a vegetarian/vegan menu or verify that a given menu is vegetarian or vegan
 - i. Add a new argument in rules wherever it is needed to be able to verify or deduce this information
 - ii. Valid queries


```
?- french_menu(M, Cal, veggy).      % asks for a vegetarian menu
?- french_menu(M, Cal, vegan).      % asks for a vegan menu
?- french_menu(M, Cal, V).          % asks for any kind of menu, if the
produced menu is vegan or veggy, V should say so
```
5. *Wine:* A French menu often comes with wine(s). We would like to be able to check that a wine is proper or to propose a proper one.

- i. Add facts about wine in the form of **wine(Name, Color, Character)**.
Color can be white or red.
Character can be dry, semi-sweet, sweet. Much more could be said about wine characteristics but this will suffice for this project.
- ii. In a first stage, assume that there is only one wine for the whole meal.
The rule is that
 - *if there is fish in any of the dish the wine should be white and dry, otherwise it should be red and dry.*

Valid query

?- french_menu(M, Cal, Veg, wine(Name, Color, Nature)).

- iii. In a second stage, we want to be able to have a matching wine per each course. The rules are that
 - *For appetizers and main course if there is fish in the dish the wine should be white and dry, otherwise it should be red and dry.*
 - *For cheese it should be red and dry.*
 - *For dessert it should be white, semi-sweet or sweet.*
 - *If there is no reason to change the type of wine, the wine should be the same as for the previous dish.*

Valid query

?- french_menu(M, Cal, Veg, ListOfWineTriplets).

Note that M and ListOfWineTriplets should have **the same length by construction**.

6. *Useful interface predicates:* We now want to write some specific predicates, less general than french_menu/4 but that can help users build queries
 - i. low_cal(?Menu, ++Max_cal): generates or verifies a Menu that has less than a given maximum of calories.
 - ii. with_fish(Menu): generates or verifies a Menu that contains some courses with fish
 - iii. no_meat(Menu): generates or verifies a Menu that contains no courses with meat (it can contain fish).
 - iv. veggy(Menu): generates or verifies a Menu that contains no courses with meat or fish.
 - v. vegan(Menu): generates or verifies a Menu that contains only vegan components.

7. *User-friendly display:* To test your program it is sufficient that you use the standard Eclipse interface that gives answers with variable unification. For end-users, however, it is more friendly to display the menu in a structured way, for example

Menu (<vegetarian/vegan/no meat>, Calories: ...)

Appetizer:	wine: ...
Main course: with	wine: ...
Cheese: ...	wine: ...
Dessert: ...	wine: ...

- i. Make sure that appetizer, cheese and dessert areas are only printed if they actually occur in the menu

Appendix – French dishes to be inserted in the program as facts

Hints: do not keep the French accents (Eclipse does not like them). Remember that if an identifier starts with an upper-case letter it is considered as a variable; there should be no space in an atom: Beet root salad -> beet_root_salad.

- Appetizer
 - Beet root salad, vegan
 - Potato Salad, vegetarian
 - Tomato Salad, vegan
 - Fish Mayonnaise
 - Cesar Salad
 - Shellfish cocktail
 - Chilled Melon, vegan
 - Smoked Salmon
 - Pork liver “paté”
 - Root vegetable soup, vegan
 - Fish soup
 - Green Bean soup, vegan
 - Onion soup, vegetarian
 - Poached egg, vegetarian
- Main dish
 - Meat ravioli
 - Spinach ravioli, vegetarian
 - Spaghetti bolognaises
 - Fallafel, vegan
 - Kasha galette, vegan
 - Sole meuniere
 - Grilled salmon
 - Scampi
 - Trout
 - Sauté chicken
 - Chicken suprême
 - Grilled steak
 - Grilled lamb
 - Blanquette de veau
 - Boeuf bourguignon
 - Gigot d’agneau
 - Onion omelette, vegetarian
 - Mushroom omelette, vegetarian
- Side dish
 - Baked potato, vegetarian
 - Fries, vegan

- Mashed potatoes, vegetarian
- Rice, vegan
- Spaghetti, vegan
- Grilled mushrooms, vegan
- Cauliflower with a cheese sauce, vegetarian
- Green beans with butter
- Green beans with oil, vegan
- Green salad, vegan
- Ratatouille, vegan
- Gratin dauphinois, vegetarian
- Cheese
 - Camembert, vegetarian
 - Brie, vegetarian
 - Roquefort, vegetarian
 - Emmental, vegetarian
 - Bleu de Bresse, vegetarian
 - Goat cheese, vegetarian
 - Yoghourt, vegetarian
 - Rice “yoghourt, vegan
- Dessert
 - Fruit sorbet, vegan
 - Crepe suzette, vegetarian
 - Pêche Melba, vegetarian
 - Fresh fruit, vegan
 - Tarte tatin, vegetarian
 - Éclair au chocolat, vegetarian
 - Crème brûlées, vegetarian
- Wine (this is a gross approximation; you can replace it with a more precise set of facts)
 - Bordeaux red dry
 - Bordeaux white dry
 - Bourgogne red dry
 - Alsace white dry
 - Alsace white semi-sweet
 - Anjou white dry
 - Anjou white semi-sweet
 - Muscat white sweet