Exercise Sheet 09 – Errors and Finite State Machines

Sebastian Höffner Aline Vilks

Deadline: Mon, 05 June 2017 08:00 +0200

Submission

By the end of this sheet you will have a number of different files to submit. In Stud.IP you will have a directory for your own group, please upload them there. It is easier for you if you just archive (preferably zip) all files and upload your archive, but it is okay if you upload them one by one.

Exercise 1: Making coffee

Problem description

We can describe the process of making coffee as a finite state machine, in this case an acceptor.

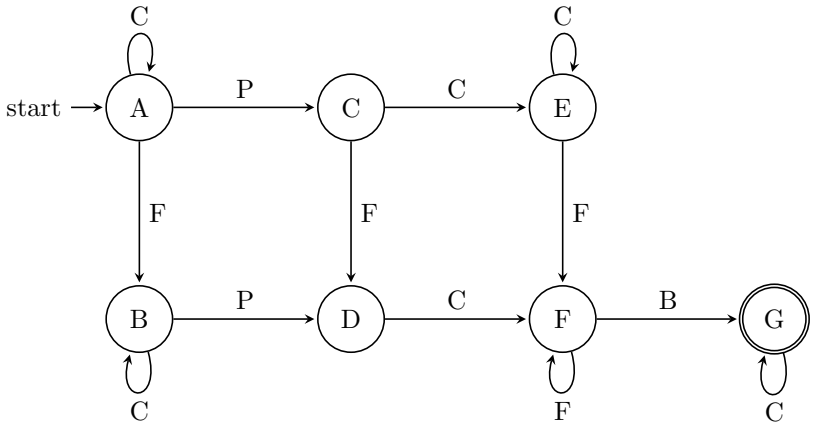
The coffee machine has four parts: a pot, a water container, a filter, and a red button.

The coffee pot can be filled with water if it is empty.

When the coffee pot is full with water, you can pour the water into the machine's water container. If you try to pour it while the pot is empty, nothing happens.

The filter can be filled with coffee grind at any time during the process, but only once.

If you push the red button while the pot is empty, the water container is filled, and there is coffee grind inside the filter, then the coffee will be perfectly brewed. If you push the red button at any other time, the process will fail.



The states in the diagram above can be described like this, with A being the start, G being the accepting state (Note that H is not in the diagram as it is the implicit error state):

State	Pot	Container	Filter	Button
A	empty	empty	empty	released
B	empty	empty	filled	released
C	filled	empty	empty	released
D	filled	empty	filled	released
E	empty	filled	empty	released
F	empty	filled	filled	released
G	empty	filled	filled	pushed
H	error	error	error	error

The possible state transitions (actions) are as follows:

ID	Action
P	Fill pot.
C	Empty pot / fill container.
F	Fill filter.
B	Push button.

Taking everything into account we can come up with a transition function. The first column in the following table describes the current state, the first row the action to be performed in that state. The letter where state and action match is the follow-up state. This results in the following transition function δ:

δ	P	C	F	B
A	C	A	B	H
B	D	B	H	H
C	H	E	D	H
D	H	F	H	H
E	H	E	F	H
F	H	F	H	G
G	H	G	H	H
H	H	H	H	H

All in all our acceptor is formally defined with:

- Input alphabet $\Sigma = \{P, C, F, B\}$
- State set $S = \{A, B, C, D, E, F, G, H\}$
- Start state $S_0 = \{A\}$
- Transition function δ as defined above
- Accepting states $F = \{G\}$

Your task

Inside the file `coffeerecipes.txt` there are 10 different attempts of brewing coffee. They are described by the state transitions ("actions"), for example `PCFB`. This line would be accepted, while e.g. `PFCFB` would not.

Write a file `coffee.py` which reads all recipes in `coffeerecipes.txt` and decides for each whether it was successful or not. Output your results.

```
Recipe:  PFCFB
Result: Fail.

Recipe:  PFCB
Result: Okay.
```

Bonus task

Bonus: Sort the recipes by their result and output the states as well:

```
Recipe:  PFCFB
States: ACDFHH
Result: Fail.

Recipe:  PFCB
States: ACDFG
Result: Okay.
```