



Tutorials for Verification Exercise sheet 1

Exercise 1: Weakest Precondition

In lecture 2 the operator Pre was presented. A similar operator is $wlp(C)$ (weakest liberal precondition) defined as follows:

$$wlp(C) = \{s \in S \mid \forall s' \in S, \alpha \in Act. s \xrightarrow{\alpha} s' \Rightarrow s' \in C\}$$

- (a) Give an example for $wlp(C) \neq Pre(C)$.
- (b) Prove or refute:

$$Post(C_1) \subseteq C_2 \Leftrightarrow C_1 \subseteq wlp(C_2) \tag{1}$$

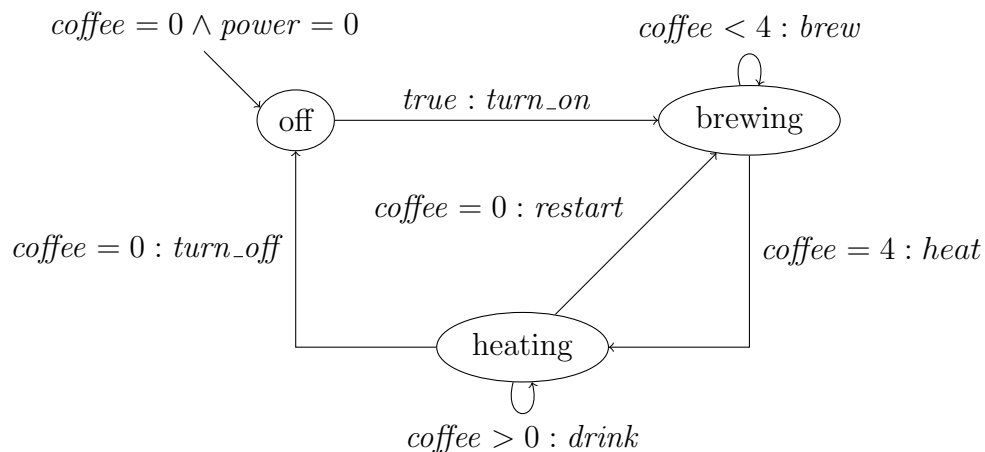
$$C_1 \subseteq Post(C_2) \Leftrightarrow wlp(C_1) \subseteq C_2 \tag{2}$$

- (c) How can $wlp(C)$ be defined in terms of
 - (i) $Pre(\cdot)$ or
 - (ii) $Post(\cdot)$?

You are allowed to use subset, union, arbitrary union ($\bigcup\{X \mid \dots\}$), complement.

Exercise 2: Coffee Machine

The following program graph describes a simple coffee machine:



The effect of the operations is given by:

$$\begin{aligned} \text{Effect}(\text{turn_on}, \eta) &= \eta[\text{power} := 1] \\ \text{Effect}(\text{turn_off}, \eta) &= \eta[\text{power} := 0] \\ \text{Effect}(\text{brew}, \eta) &= \eta[\text{coffee} := \text{coffee} + 1] \\ \text{Effect}(\text{drink}, \eta) &= \eta[\text{coffee} := \text{coffee} - 1] \\ \text{Effect}(\text{restart}, \eta) &= \eta \\ \text{Effect}(\text{heat}, \eta) &= \eta \end{aligned}$$

- (a) Draw the transition system corresponding to the program graph.
- (b) Check the following temporal properties. Label the transition system with the corresponding atomic propositions.
- (i) If the machine is turned off ($\text{power} = 0$) it contains no coffee ($\text{coffee} = 0$).
 - (ii) If there are two cups of coffee ($\text{coffee} = 2$) there are either three or four cups of coffee in the next step.
 - (iii) There are always at most four cups of coffee ($\text{coffee} \leq 4$).
 - (iv) The coffee machine will be eventually turned off.
 - (v) If there is no coffee ($\text{coffee} = 0$), there will be coffee after at most three steps.

Exercise 3: Collatz

Convert the following C program into a program graph representation and into a transition system.

```
int i = 5;
while (i != 1) {
    if ((i % 2) == 0)
        i = i / 2;
    else
        i = 3*i + 1;
}
```

If you find out whether the program terminates for any value of i , you will become very famous.

Exercise 4: A variant of *König's Lemma* *

If a transition system has finitely many initial states but infinitely many reachable states, then either there is a reachable state with infinitely many direct successors or there is an infinite execution which is not self-intersecting (i.e. no state occurs more than once in the execution). Don't use König's Lemma itself.