CONCORDIA UNIVERSITY
Department of Computer Science
and Software Engineering

SOEN 331-W: Introduction to Formal Methods for Software Engineering
Winter 2019

Assignment 1 on Extended Finite State Machines

Dr. Constantinos Constantinides, P.Eng.

January 28, 2019

Contents

1 General information

Date posted: Monday 28 January, 2019.

Date due: Monday 11 February, 2019, by 23:59.

Weight: 5% of the overall grade.

2 Introduction

This assignment should be done by teams of 3 or 4. Each team should designate a leader who will submit the assignment electronically.

3 System description

The system in discussion controls and maintains the temperature inside some room. The system uses an electrical furnace in order to build up a desired temperature with respect to the current time, set by a user. Once that temperature is reached, the furnace goes off and a fan is activated which will distribute hot air throughout the room. Furnace and fan are never both activated at the same time.

设置的时候只有设置三个数据，起始，终点时间，需要温度　　可以多次设置，同样的数据也能overide

The system, initially idle, can be configured for setting the desired temperature within time periods in the form of triplets, i.e. `time-from, time-to, desired-temperature`. During configuration a user can set as many triplets as they wish, and one triplet may override another with identical data. Upon entering configuration mode, the system will produce a beep sound, and a led light will switch on and stay on as long as the system is under configuration. Once a user has completed their entries of desired temperature(s), they can indicate the successful completion of configuration, thus letting the system get back to the initial idle mode while producing a double beep sound.

设置状态进入，发出一声BEEP，led开，设置状态完成，用户需要表示这就是final数据了，系统会到起始状态并bep两下

During configuration, the system allows the user to cancel the current mode in which case the system gets back to the initial idle mode while producing a prolonged beep sound. Triplets that are entered prior to cancellation are still registered and considered. Additionally, during configuration a timer is active which, after an inactivity for 1 minute, will cause the system to become idle.

设置的时候，系统允许用户离开当前状态这样系统就回起始状态，发出一声长beep，之前设置的数据仍然有效， 设置的时候会激活一个计时器，如果1分钟没操作，会让system idle

Once idle, the system goes into the following cycle which repeats every 2 minutes: It will read the current time and, based on its configuration, it will subsequently determine what is the desired temperature. The system will read the current room temperature, and if it is not less than the desired temperature, then it will do nothing, but to repeat the cycle. However, if the current temperature is less than one degree lower than the desired temperature, then the system turns the furnace on to build up the desired temperature.

idle状态system每两分钟进行以下操作，读取当前时间，取决于输入，他知道所追求的温度，然后他读取当前房间温度

如果大于等于需求温度，什么都不会干，循环cycle，但是如果小于1度以上，开始运转火炉

加热状态的时候，火炉保持运转，系统每三分钟cycle一次：
阅读火炉内温度，如果比约定温度高了一度以内，什么都不会干如果大了一度以外，system回到idle状态并同时打开风扇来吹热风，发出click的声音，系统允许在warming的时候设置，设置的时候关掉火炉

During warming up, the furnace remains on, and the system goes into the following cycle which repeats every 3 minutes: It reads in the temperature within the furnace and if it less than a degree higher than the desired temperature, then it will do nothing, but to repeat the cycle. When the furnace temperature reaches a degree higher than the desired temperature, then the system goes back to being idle while turning on the fan in order to blow hot air into the room while at the same time producing a "click" sound. The system allows a user to interrupt warming up in order to perform configuration (as discussed above). The furnace cannot be on during system configuration.

When idle, the system can be shut off. When the system is off, both fan and furnace must be switched off.

idle的时候你可以关掉系统，这时fan与furnance必须关掉

4 Simulation

For this part you have to use Prolog to translate the specification into a declarative database, and define the following rules:

1. Rule `ancestor/1` succeeds by returning an ancestor to a given state.
2. Rule `get_all_transitions/1` succeeds by returning a list of all transitions that take place along the entire system, where an event and a guard are both not-null. A transition is a tuple of type `(source, destination, event, guard)`.
3. Rule `get_inherited_transitions/2` succeeds by returning all transitions inherited by a given state, where a transition is expressed as a tuple of type `(source, destination, event, guard)`.

5 What to submit

Please follow the instructions carefully:

1. Use LaTeX(a template is provided) to prepare the formal specification and UML state diagrams of the extended finite state machine that models the behavior of the system. Produce a pdf file called `specification.pdf`.
2. Produce a single Prolog file (a template is provided) called `simulation.pl`.
3. Place the two files into a folder named after the University id of the person who will submit, and zip the folder.

Submit the zip file at the Electronic Assignment Submission portal (`https://fis.encs.concordia.ca/eas`) under **Assignment 1**.

6 Late submissions

No late submissions will be accepted after the due date and time.