CONCORDIA UNIVERSITY
Department of Computer Science and Software Engineering

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

Tutorial exercise on Extended Finite State Machines - Home temperature control system

Dr. Constantinos Constantinides, P.Eng.

February 6, 2020

1 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.

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.

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.

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 or equal to one degree lower than the desired temperature, then the system turns the furnace on to build up the desired temperature.

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.

2 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/2` 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)`.