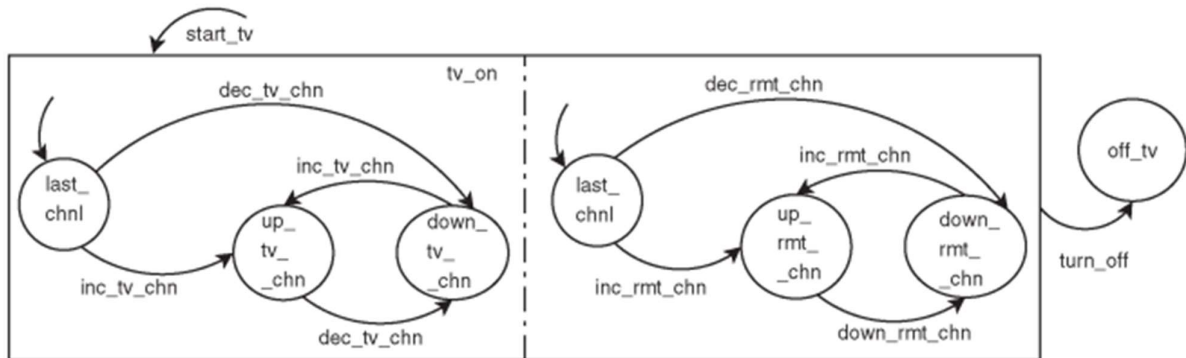


Deriving Hierarchical FSM from C-based Design

By: Neil Alvares [B19EE055] & Kwanit Gupta [B19EE046]

Hierarchical FSM: Hierarchical state machines are finite state machines whose states can be other finite state machines. Many modelling formalisms and tools for software design, requirements, and testing use hierarchy as a useful construct.

Problem:



- We have used an FSM of a Tv control and converted into a hierarchy with a parent FSM called Main and 2 sub/child FSMs called Tv and Remote.
- The whole project is modelled using python classes with a detailed description given in **Readme.md** file.
- When you run the main.py file, the user is prompted to enter a number from [0 - 3, -1] that corresponds to an event in the FSM.
- 0 : Decrease the channel using Tv button
- 1 : Increase the channel using Tv button
- 2 : Decrease the channel using remote button
- 3 : Increase the channel using remote button
- -1 : Turn Off the Tv

Output:

- With a sample user's input, we can see the following:

```
$ python -u "c:\Users\Neil\Desktop\py-test\main.py"
Tv Remote Control FSM
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Starting Tv
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Action= 0
Event: Decrease Tv Channel
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Action= 1
Event: Increase Tv Channel
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Action= 2
Event: Decrease Remote Channel
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Action= 3
Event: Increase Remote Channel
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel

Action= -1
Event: Turn off
FSM state: Main=IDLE -> Tv_buttons=last_channel, Remote_buttons=last_channel
```