Modelling Clock

Dr. Tianxiang Lu

April 27, 2015

1 The first model: Hour Clock

1.1 What is an hour clock

The hour clock shows the hour. The hour is simply a number between all the legal hours, either from 1 to 12 in a 12 hours system, or from 0 to 23 in a 24 hours system.

1.2 What do we need to specify an hour clock

- Variable: hr
- Constant: $Hour$
- Initial state: $hr = 1$?
- Next state: $hr' = hr + 1$
- Constraints: $hr > 12$?
- Specification:

$$Init \wedge \Box[Next]_{hr}$$

1.3 A first model of hour clock

```
---------------------- MODULE HourClock ----------------------
EXTENDS Naturals
VARIABLE hr
HCini  ==  hr \in (1 .. 12)
HCnxt  ==  hr' = IF hr # 12 THEN hr + 1 ELSE 1
HC  ==  HCini /\ [][HCnxt]_hr
--------------------------------------------------------------
THEOREM  HC => []HCini
==============================================================
```

Is there other ways of specifying the same clock?

1.4 A second model of hour clock

Extending the current module, which then include all the definitions and declarations from module HourClock. We can specify the same clock using modulo calculation % from the Naturals module.

```
--------------------------- MODULE HourClock2 ----------------------------

EXTENDS HourClock

HCnxt2  ==  hr' = (hr % 12) + 1
HC2  ==  HCini /\ [][HCnxt2]_hr
```

```
-------------------------------------------------------------------------------
THEOREM  HC <=> HC2
===============================================================================
```

This theorem asserts that formulas HC and $HC2$ are equivalent. The symbol `<=>` which can also be typed as `\equiv` , is typeset as an equivalence symbole (a three-lined equals sign, $\equiv$).

1.5 Model Checking

How many possible states does this model have?

2 The hour-minute clock

What about a clock with hour and minute?

- One more variable: mnt
- One more constance: $Minute$
- Update of action with minute increments.

Finally, we have the following:

```
---------------------------- MODULE HourMinuteCLK ----------------------------
EXTENDS Integers
VARIABLE hr, mn

Hinit==hr \in (1 .. 12)
Minit==mn \in (0 .. 2)
TypeCorrectM == mn \in (0 .. 59)
init==Hinit /\ Minit


Hnext== /\ mn'=IF mn # 59 THEN mn+1 ELSE 0
        /\ hr'=IF mn # 59
              THEN hr
              ELSE IF hr # 12 THEN hr+1 ELSE 1

Time==init /\ [][Hnext]_<<hr, mn>>
-------------------------------------------------------------------------------
THEOREM Time => [](Hinit /\ Minit)
===============================================================================
```

Can you think of another solution for minute clock? This will be the homework.

2.1 Model Checking

How many possible distinct states does this model have?

3 An hour-minute-second clock

What about a complete clock including hour, minute and second?