

# Hoare Logic and Model Checking

Model Checking Lecture 1 Supplement

### Conrad Watt

Computer Laboratory, University of Cambridge, UK http://www.cl.cam.ac.uk/~caw77

CST Part II - 2018/19

# Lecture 1 Recap

- This course is about checking properties of finite models.
- Our models are expressed as Kripke structures.
- Our properties are expressed using temporal logic.
  - You have not seen a formal definition of temporal logic yet.
  - Temporal logics are less powerful than predicate logic, but are decidable and easy to automate.

Conrad Watt

Hoare Logic and Model Checking

# Lecture 1 Recap - Kripke structures

- A Kripke structure is a transition system plus some labelling information.
- Formally, a Kripke structure/model is written as M = (S, S<sub>0</sub>, R, L)
- S is a finite set of states.
- $S_0 \subseteq S$  is a subset of distinguished starting states.
- $R \subseteq S \times S$  is a transition relation over states.
- $L :: S \to \mathcal{P}(AP)$  is a labelling function.
- The label function L associates each state with a set of propositional atoms (atomic properties).



1/14

2/14

# Lecture 1 Recap - Modelling

- Given a definition of a system, we need to work out how to represent it as a Kripke structure.
  - Then we can begin checking the truth of temporal logic properties.

Thread 1		Thread 2
0:	IF LOCK=0 THEN LOCK:=1;	0: IF LOCK=0 THEN LOCK:=1;
1:	X:=1;	1: X:=2;
2:	IF LOCK=1 THEN LOCK:=0;	2: IF LOCK=1 THEN LOCK:=0;
3:		3:

Assumptions (always good practice to state):

- All variables 0-initialized.
- One entire line executed per step.
- Thread scheduling is non-deterministic.
- "IF" lines are only scheduled if condition is true.

### Conrad Watt

Hoare Logic and Model Checking

5 / 14 Conrad Watt

#### Hoare Logic and Model Checking

Thread 2

X:=2:

0: IF LOCK=0 THEN LOCK:=1;

2: IF LOCK=1 THEN LOCK:=0:

6/14

## Lecture 1 Example

 Thread 1
 Thread 2

 0: IF LOCK=0 THEN LOCK:=1;
 1: F LOCK=0 THEN LOCK:=1;

 1: X:=1;
 1: X:=2;

 2: IF LOCK=1 THEN LOCK:=0;
 2: IF LOCK=1 THEN LOCK:=0;

 3:
 3:

- Need to pick (S, S<sub>0</sub>, R, L) to model the program.
- $S \triangleq \{0, 1, 2, 3\} \times \{0, 1, 2, 3\} \times \{0, 1\} \times \{0, 1, 2\}$
- ►  $S_0 \triangleq \{ (0, 0, 0, 0) \}$
- ▶ To define *R*, look at how states should evolve over time.
- ▶ e.g. in all states of the form (0,0,0,-), either thread 1 or 2 will take the LOCK and advance their pc

# Lecture 1 Example

Lecture 1 Example

X:=1:

integers.

IF LOCK=0 THEN LOCK:=1;

Need to pick (S, S<sub>0</sub>, R, L) to model the program.
 Observe: two program counters, two variables - all

▶ Individual state  $(s \in S) \triangleq (pc_1, pc_2, LOCK, X)$ 

 $\triangleright$  S<sub>0</sub> is just the singleton set { (0, 0, 0, 0) }

• Why not just define  $S \triangleq \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ ?

 $S \triangleq \{0, 1, 2, 3\} \times \{0, 1, 2, 3\} \times \{0, 1\} \times \{0, 1, 2\}$ 

2: IF LOCK=1 THEN LOCK:=0;

Represent state set as

Thread 1

Thread 1		Thread 2
0:	IF LOCK=0 THEN LOCK:=1;	0: IF LOCK=0 THEN LOCK:=1;
1:	X:=1;	1: X:=2;
2:	IF LOCK=1 THEN LOCK:=0;	2: IF LOCK=1 THEN LOCK:=0;
3:		3:

- Need to pick (S, S<sub>0</sub>, R, L) to model the program.
- ►  $S \triangleq \{0, 1, 2, 3\} \times \{0, 1, 2, 3\} \times \{0, 1\} \times \{0, 1, 2\}$
- ►  $S_0 \triangleq \{ (0, 0, 0, 0) \}$
- Define R:

# Lecture 1 Example

Thread 1	Thread 2
0: IF LOCK=0 THEN LOCK:=1;	0: IF LOCK=0 THEN LOCK:=1;
1: X:=1;	1: X:=2;
2: IF LOCK=1 THEN LOCK:=0;	2: IF LOCK=1 THEN LOCK:=0;
3:	3:

- Need to pick (S, S<sub>0</sub>, R, L) to model the program.
- What about L?
- Depends on which properties we want to prove.
- Any potential predicate on program state can be turned into a label.
- e.g. all states satisfying (3, 3, -, -) could be labelled finished
- Then can express and check the temporal property "every execution of the program eventually reaches finished".
  - i.e. all lines of code in both threads are executed
  - You will later see how to state this property formally.

#### Conrad Watt

Hoare Logic and Model Checking

9 / 14 Conrad Watt

Hoare Logic and Model Checking

Thread 2

X:=2:

TE LOCK=0 THEN LOCK .= 1 .

IF LOCK=1 THEN LOCK:=0:

10/14

# **Drawing Kripke Structures**

Lecture 1 Example

X:=1:

at once.

IF LOCK=0 THEN LOCK:=1;

IF LOCK=1 THEN LOCK := 0 :

Convention: annotate states with atomic properties.

Let's imagine we want to prove mutual exclusion.

That is, both threads will never be "inside" the locked code

Can label all states satisfying (1, 1, -, -) with violation.

Then mutual exclusion can be expressed as "no execution

Equivalently, "all reachable states satisfy ¬violation".

of the program will reach a state satisfying violation".

- Strictly speaking, should draw unreachable states too.
  - But sometimes missed out. Use your judgement.
- You will never be asked to draw something this big. (96 states!)



## **Drawing Kripke Structures**



Conrad Watt

# Lecture 1 Recap - Some Definitions

- A transition system or model is called *left-total* if no state is a "dead-end".
- That is, every state can transition to another state.
- $\blacktriangleright R s_1 s_2 \equiv (s_1, s_2) \in R$
- R\* is the transitive closure of R.
- A path, often written π, is a sequence of states.
- Path *R* s  $\pi$  is true iff  $\pi$  is a path starting at s such that successive states in  $\pi$  are related by *R*.
- $\pi(i)$  and  $\pi i$  are both syntax for the *i*th element of path  $\pi$ .
- $\pi \downarrow i$  is the suffix of  $\pi$  starting from its *i*th element.

# Lecture 1 Recap - Other Slide Content

Other content from lecture 1 (less important):

- Reinterpreting slightly different formal models as Kripke models.
- Modelling systems with input (treat it as an extra component of the state).
- Historical syntax for models/interpretations.

Hoare Logic and Model Checking

13 / 14 Conrad Watt

Hoare Logic and Model Checking

14/14