

Logic Summer School, ANU Dec 2003  
Formal Methods 1: Algorithmic Verification  
Ron van der Meyden (UNSW/NICTA)  
Lecture 5: Model Checking Knowledge

## Overview

Logics of knowledge and time  
Model checking knowledge & time  
Examples

## A Model for Runs of a Distributed System

Let  $\mathcal{L}$  be a set of local states of the agents  
and  $S_e$  be a set of states of the environment

Define the set of *global states* as  $\mathcal{G} = \mathcal{L}^n \times S_e$ , i.e., a global state is a  
tuple  $\langle l_1, \dots, l_n, s_e \rangle$ .

- for  $i = 1, \dots, n$ , the component  $l_i$  represents the local state of agent  $i$
- $s_e$  represents the state of the environment

## Runs

A *run* over global states  $\mathcal{G}$  is a mapping  $r : \mathbf{N} \rightarrow \mathcal{G}$ .

If  $r(m) = \langle l_1, \dots, l_n, s_e \rangle$

1. write  $r_i(m)$  for  $l_i$ , and
2. write  $r_e(m)$  for  $s_e$

## Distributed Systems

A *system* over global states  $\mathcal{G}$  is a set of runs over  $\mathcal{G}$ .

Let  $Prop$  be a set of propositional constants.

An *interpretation* for  $\mathcal{G}$  is a function  $L : \mathcal{G} \rightarrow \mathcal{P}(Prop)$

An *interpreted system*  $\mathcal{I} = (\mathcal{R}, L)$  consists of a system  $\mathcal{R}$  together with an interpretation function  $L$ .

## A Language for Knowledge and Time

The following are formulas:

$p$ , where  $p \in Prop$

$\neg\phi, \phi_1 \wedge \phi_2,$

$\mathbf{X}\phi$  (“ $\phi$  at the next moment of time”)

$\phi_1 \mathbf{U} \phi_2$  (“ $\phi_1$  until  $\phi_2$ ”)

$K_i\phi$ , where  $i = 1 \dots n$  (“agent  $i$  knows  $\phi$ ”)

define  $\phi_1 \Rightarrow \phi_2$  as  $\neg\phi_1 \vee \phi_2$ ; etc

$\mathcal{I}, (r, m) \models p$  if  $p \in L(r(m))$ .

$\mathcal{I}, (r, m) \models \neg\phi_1$  if not  $\mathcal{I}, (r, m) \models \phi_1$

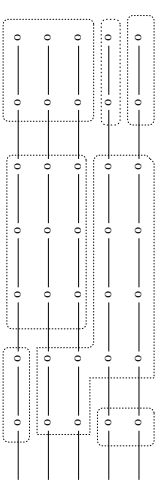
$\mathcal{I}, (r, m) \models \phi_1 \wedge \phi_2$  if  $\mathcal{I}, (r, m) \models \phi_1$  and  $\mathcal{I}, (r, m) \models \phi_2$

$\mathcal{I}, (r, m) \models \mathbf{X}\varphi$  if  $\mathcal{I}, (r, m + 1) \models \varphi$ .

$\mathcal{I}, (r, m) \models \phi_1 \mathbf{U} \phi_2$  if there exists  $m \geq n$  with  $\mathcal{I}, (r, m) \models \phi_2$  and  $\mathcal{I}, (r, k) \models \phi_1$  for all  $k$  with  $n \leq k < m$ .

Two points  $(r, m)$  and  $(r', m')$  are *indistinguishable to agent  $i$* , written  $(r, m) \sim_i (r', m')$  just when  $r_i(m) = r'_i(m')$ .

$\mathcal{I}, (r, m) \models K_i\varphi$  if  $\mathcal{I}, (r', m') \models \varphi$  for all points  $(r', m') \sim_i (r, m)$



Defining branching time in systems:

$\mathcal{I}, (r, n) \models \mathbf{A}\phi$  if  $\mathcal{I}, (r', n) \models \phi$  for all runs  $r'$  of  $\mathcal{I}$  such that  $r(0)r(1)\dots r'(n) = r'(0)r'(1)\dots r'(n)$ .

An *Büchi transition system with observations for  $n$  agents* is a tuple  $E$  of the form  $\langle S, S_0, \rightarrow, L, \alpha, O \rangle$  where the components are as follows:

1.  $S$  is a finite set of *states*.
2.  $S_0$  is a subset of  $S$ , the *initial states*
3.  $\rightarrow$  is a binary relation on  $S$
4.  $L : \mathcal{G} \rightarrow \mathcal{P}(\text{Prop})$  is an *interpretation*,
5.  $\alpha \subseteq S$  is a Büchi acceptance condition
6.  $O = \langle O_1, \dots, O_n \rangle$  is a tuple of *observation functions*  
 $O_i : S \rightarrow \text{Obs}$

A run of a Büchi transition system with observations  $\langle S, S_0, \rightarrow, L, \alpha, O \rangle$  is a run  $\rho = s_0, s_1, s_2, \dots$  of the transition system  $\langle S, S_0, \rightarrow, L \rangle$  such that  $\text{inf}(\rho) \cap \alpha \neq \emptyset$ .

We interpret  $S$  as states of the environment, and lift such runs to runs over global states by assigning local states to the agent at each point, using the observation functions....

### Local state defined wrt a view

Let  $\rho : \mathbf{N} \rightarrow S$  be a run of  $E$ . A *view* associates a local state in some set  $L$  of local states with each agent at each point of time, determining a mapping  $\rho^v : \mathbf{N} \rightarrow L^n \times S$

In all cases  $\rho_e^v(m) = \rho(m)$

Examples:

1. The *observational view*:  $\rho_i^{\text{obs}}(m) = O_i(\rho(m))$
2. The *clock view*:  $\rho_i^{\text{clock}}(m) = (m, O_i(\rho(m)))$
3. The *synchronous perfect recall view*:  
 $\rho_i^{\text{spr}}(m) = O_i(\rho(0)) \dots O_i(\rho(m))$

### System Generated by an Environment wrt a View

Let  $v$  be a view of an Büchi transition system  $E = \langle S, S_e, \rightarrow, L, \alpha, O \rangle$ .

Define  $\mathcal{I}^v(E) = (\mathcal{R}^v(E), L)$  to be the interpreted system with

1.  $\mathcal{R}^v(E)$  the set of  $\rho^v$  such that  $\rho$  is a run of  $E$ .
2.  $L'(\langle l_1, \dots, l_n, s_e \rangle) = L(s_e)$

### Model Checking Knowledge and Time

Inputs:

1. a concurrent program  $P$ , together with a definition of the variables observable to each of the agents.  
Let  $E$  be the Büchi transition system with observations generated by  $P$ .

2. a formula  $\phi$  of  $CTL^* + K$

3. a view  $v$

Problem: determine if  $\mathcal{I}^v(E), (r, 0) \models \phi$  for all runs  $r$  of  $E$

### Cases supported in MCK

|                  | Observational             | Clock                       | Perfect Recall                    |
|------------------|---------------------------|-----------------------------|-----------------------------------|
| leading $X^n$    | spec-obs- $\perp$ $l$ $i$ | spec- $clk$ - $xn$          | spec- $spr$ - $xn$                |
| nested $X/K$     | spec-obs- $\perp$ $l$ $i$ | spec- $clk$ - $ctl$ -nested | spec- $spr$ - $l$ $i$ $l$ -nested |
| nested $AX/EX/K$ | spec-obs- $act$ $i$       |                             |                                   |
| full $CTL/K/CK$  | spec-obs- $act$ $i$       |                             |                                   |
| full $LTL/K/CK$  | spec-obs- $\perp$ $l$ $i$ |                             |                                   |

Brafman, Latombe, Moses, Shoham: Applications of a logic of knowledge to motion planning under uncertainty. JACM 1997



- Sensor  $\in$  [position-1, position+1]
- Robot moves under control of the environment, *at most* one step per unit time.

A knowledge-based program:

```
wait until Know(position in Goal);  
halt.
```

Implementations when  $\text{Goal} = \{2, 3, 4\}$  and agent's view = **Sensor**:

```
I1: wait until Sensor = 3;  
halt.
```

```
I2: wait until Sensor in {3, 4, 5};  
halt.
```

### Dining Cryptographers

David Chaum, J. Cryptology 1988:

Three cryptographers are sitting down to dinner at their favorite three-star restaurant. Their waiter informs them that arrangements have been made with the maitre d'hotel for the bill to be paid anonymously. One of the cryptographers might be paying for the dinner, or it might have been the NSA (US National Security Agency). The three cryptographers respect each other's right to make an anonymous payment, but they wonder if the NSA is paying. They resolve their uncertainty fairly by carrying out the following protocol:

Assumption: at most one cryptographer is paying.

1. Each cryptographer flips an unbiased coin behind his menu, between him and the cryptographer to his right, so that only the two of them can see the outcome
2. Each cryptographer then states aloud whether the two coins that he can see - the one he flipped and the one his left-hand neighbour flipped - fell on the same side or different sides
  - 2e. If one of the cryptographers is the payer, he states the opposite of what he sees.

An odd number of differences uttered at the table indicates that NSA is paying, an even number of differences indicates that a cryptographer is paying.

If a cryptographer is paying neither of the other two learns anything from the utterances about which cryptographer it is.