Slide 1

North American Summer School in Logic Language and Information, June 2003

Algorithmic Verification for Epistemic Logic

Ron van der Meyden

University of New South Wales/National ICT Australia

Slide 2

Overview of the Course

Motivation

Logics of knowledge and time

Model checking knowledge & time

A model checking system & examples

Synthesis & knowledge based program implementation

Slide 3

The main problem *unique* to distributed systems is a lack of (global) knowledge. It is difficult (probably impossible) for one node to know everything about the rest of the network. Yet global knowledge seems to be required to answer questions such as "Where is the file A", "Is there a deadlock", [or] "What is the best way to answer the question.... " (Gray, 1979)

"Once the sender receives the acknowledgement, it *knows* that the current packet has been delivered; it can then safely discard the current packet and send the next."

Slide 4

Sequence Transmission Problem

Sender S to communicate bits x_1,x_2, .. to receiver R across a possibly unreliable medium (Halpern and Zuck, JACM 1990)

Sender S:

```
For each i = 0.. do
 while not knows(S,knows(R,x_i))
  do
    send(R,⟨ i, x_i ⟩)
    wait(T)
end
```

Receiver R:

```
For each i = 0 ... do
while not knows(R,x_i) do
    send(S,⟨ i, ? ⟩)
    wait(T')
end
```

Slide 5

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

Environmental Constraints:

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

Slide 6

A knowledge-based program:

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

Implementations:

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

(when agent's view = Sensor)

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

(when agent's view = Sensor, and when agents view = Sensor + clock value)

Slide 7

Benefits Claimed for Knowledge-Based Programs

- **Abstractness:** correctness proofs at the knowledge level are simpler and more intuitive
- **Generality:** The same knowledge-based program describes distinct protocols running under different environmental assumptions
- **Optimality:** Implementations of knowledge-based programs make optimal use of information

Slide 8

Questions concerning the implementation of knowledge-based programs:

1. Do implementations exist? Are they unique?
2. How can one verify that a given protocol is an implementation?
3. How can an agent compute what it knows in a given system?
4. If an implementation exists, how complex is it to construct one?
5. What is the inherent complexity of the implementations themselves? Can they be finite state protocols?

Slide 9

Part 1: Logics of Knowledge and Time

Slide 10

References

The Complexity of Reasoning about Knowledge and Time I (Lower Bounds), Joe Halpern and Moshe Vardi, JCSS 1989

Complete Axiomatizations for Knowledge and Time, Joe Halpern, Ron van der Meyden, Moshe Vardi, to appear SIAM J. Computing

Complete Axiomatizations for Knowledge and Branching Time, Ron van der Meyden and Ka-Shu Wong, to appear Studia logica

Slide 11

Background

- M. Sato 1977 — $\mathcal{C}^{\mathtt{pr,sync}}$
- D. Lehman 1984 — $\mathcal{C}^{\mathtt{pr,sync}}$
- R. Fagin, J. Halpern and M. Vardi 1984 — $\mathcal{C}^{\mathtt{pr}}$
- R. Parikh and R. Ramanujam 1985 — $\mathcal{C}^{\mathtt{pr}}$
- R. Ladner and J. Reif 1985 — $\mathcal{C}^{\mathtt{pr,nl,uis}}$

Slide 12

Overview

- Semantics of knowledge in distributed systems
- Properties of agents: abstract and concrete characterizations
- Axiomatizations of knowledge and linear time
- Axiomatizations of knowledge and branching time

Slide 13

A Model for Runs of a Distributed System

Let 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 $\mathcal{G} = L^n \times S_e$, i.e., a global state is a is a tuple $\langle l_1, \ldots, l_n, s_e \rangle$.

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

Slide 14

Runs

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

Write $r_i(m)$ for the i-th component of $r(m)$, and $r_e(m)$ for the $n+1$-st component (the state of the environment)

A pair (r, m) consisting of a run r and a natural number m is called a *point.*

Slide 15

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 $\pi : \mathcal{G} \times \mathit{Prop} \rightarrow \{0,1\}$.

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

Slide 16

A Language for Knowledge and Time

The following are formulas:

p, where $p \in \mathit{Prop}$

$\neg\varphi$, $\varphi_1 \wedge \varphi_2$,

$\bigcirc\varphi$ (“φ at the next moment of time”)

$\varphi_1 \, \mathcal{U} \, \Phi_2$ (“φ_1 until φ_2”)

$K_i\varphi$, where $i = 1 \ldots n$ (“agent i knows φ”)

define $\varphi_1 \rightarrow \varphi_2$ as $\neg\varphi_1 \vee \varphi_2$, etc

Slide 17

Sublanguages

Write $\mathcal{L}_{\{Op_1,\ldots,Op_n\}}$ for the sublanguage based just on the operators $Op_1,\ldots,Op_n$

E.g.

$\mathcal{L}_{\{\bigcirc,\mathcal{U}\}}$ (temporal logic)

$\mathcal{L}_{\{K_1,\ldots,K_n\}}$ (logic of knowledge)

$\mathcal{L}_{\{K_1,\ldots,K_n,\bigcirc,\mathcal{U}\}}$ (logic of knowledge & time)

Slide 18

$I,(r,m) \models p$ if $\pi(r,m)(p) = 1$.

$I,(r,m) \models \neg\varphi_1$ if not $I,(r,m) \models \varphi_1$

$I,(r,m) \models \varphi_1 \wedge \varphi_2$ if $I,(r,m) \models \varphi_1$ and $I,(r,m) \models \varphi_2$

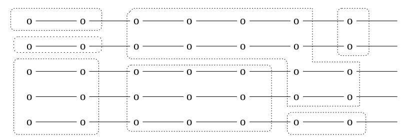
$I,(r,m) \models \bigcirc\varphi$ if $I,(r,m+1) \models \varphi$.

$I,(r,m) \models \varphi_1 U \varphi_2$ if there exists $m \geq n$ with $I,(r,m) \models \varphi_2$ and $I,(r,k) \models \varphi_1$ for all k with $n \leq k < m$.

Slide 19

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

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



Slide 20

Common Knowledge

$$E\varphi \equiv \bigwedge_{i=1}^{n} K_i\varphi$$

$$C\varphi \equiv E\varphi \wedge EE\varphi \wedge EEE\varphi \wedge \ldots$$

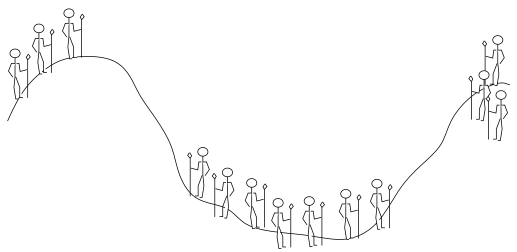
Slide 21

Semantics of Common Knowledge

Define the (equivalence) relation $\sim$ on $Points(I)$ to be the transitive closure of $\bigcup_{i=1\ldots n} \sim_i$.

$$I,(r,m) \models C\varphi \quad \text{if } I,(r',m') \models \varphi \text{ for all points } (r',m') \sim (r,m)$$

Slide 22



attack ≡ C attack

Slide 23

Axioms for Knowledge: S5$_m$

A1. All tautologies of propositional logic.

K1. $K_i\varphi \wedge K_i(\varphi \rightarrow \psi) \rightarrow K_i\psi$

K2. $K_i\varphi \rightarrow \varphi$

K3. $K_i\varphi \rightarrow K_iK_i\varphi$

K4. $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$

RK. If φ then $K_i\varphi$

RA. If φ and $\varphi \rightarrow \psi$ then ψ.

Slide 24

Political Knowledge
(Donald Rumsfeld, 2003)

As we know
There are known knowns
There are things we know we know
We also know
There are known unknowns
That is to say
We know there are some things
We do not know
But there are also unknown unknowns
The ones we don't know we don't know

Slide 25

+ Axioms for Common Knowledge: S5C$_m$

C1. $E\varphi \equiv \bigwedge_{i=1}^{m} K_i\varphi$

C2. $C\varphi \rightarrow E(\varphi \wedge C\varphi)$

RC. If $\varphi \rightarrow E(\psi \wedge \varphi)$ then $\varphi \rightarrow C\psi$

Slide 26

Axioms for Linear Time: LT

T1. $\bigcirc(\varphi) \wedge \bigcirc(\varphi \rightarrow \psi) \rightarrow \bigcirc\psi$

T2. $\bigcirc(\neg\varphi) \Leftrightarrow \neg\bigcirc\varphi$

T3. $\varphi U \psi \Leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi U \psi))$

RT1. If φ then $\bigcirc\varphi$

RT2. If $\varphi' \rightarrow \neg\psi \wedge \bigcirc\varphi'$ then $\varphi' \rightarrow \neg(\varphi U \psi)$

Slide 27

Properties of systems

sync: A system $\mathcal{R}$ is *synchronous* if for all agents i, if $(r,m) \sim_i (r',m')$ then $m = m'$.

uis: A system has *unique initial states* if for all runs r, r' and all agents i, $(r,0) \sim_i (r',0)$.

Slide 28

Concordant intervals

Two intervals (possibly infinite) of two runs are *concordant* wrt agent i if agent i goes through the same sequence of local states over those intervals, not counting consecutive repeats.

E.g. if

$$r_i[19,\infty] = aaabbaacc\ldots$$

and

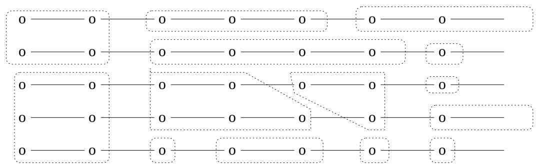
$$r_i'[2,\infty] = abaaaaaaaaaaaaaacc\ldots$$

then $r[19,\infty]$ and $r'[2,\infty]$ are concordant for agent i.

Slide 29

Properties of Systems (continued)

pr: A system $\mathcal{R}$ has *perfect recall* (or no forgetting) if for all points (r,m) and all agents i, if $(r,m) \sim_i (r',m')$ then the intervals $r[0,m]$ and $r'[0,m']$ are concordant wrt agent i.



Slide 30

Concrete constructions for synchrony and perfect recall

Let S_e be the set of states of the environment.

A *run of the environment* is a function $\varepsilon : \mathbf{N} \to S_e$.

Let Obs be a set of *observations*.

An *observation function* is a mapping $O : S_e \to Obs$.

Slide 31

Given a run ε of the environment, and an observation function O_i for each agent $i = 1 \ldots n$, define the runs $r^{\varepsilon,\texttt{obs}}$, $r^{\varepsilon,\texttt{pr}}$, $r^{\varepsilon,\texttt{clock}}$, $r^{\varepsilon,\texttt{spr}}$, by

$$r_e^{\varepsilon,x}(m) = \varepsilon(m) \quad \text{for each } x \in \{\texttt{obs},\texttt{clock},\texttt{spr},\texttt{pr}\}$$

and

$$r_i^{\varepsilon,\texttt{obs}}(m) = O_i(\varepsilon(m))$$

$$r_i^{\varepsilon,\texttt{clock}}(m) = (m, O_i(\varepsilon(m)))$$

Slide 32

$$r_i^{\varepsilon,\texttt{spr}}(m) = \langle O_i(\varepsilon(0)),\ldots,O_i(\varepsilon(m))\rangle$$

$$r_i^{\varepsilon,\texttt{pr}}(m) = O_i(\varepsilon(0))\#\ldots\#O_i(\varepsilon())$$

where # is absorbtive concatenation:

$$(\sigma x)\#y = \begin{cases} \sigma x y & \text{if } x \neq y \\ \sigma x & \text{if } x = y \end{cases}$$

Slide 33

Given a set $\mathcal{R}_e$ of runs of the environment, define the system

$$\mathcal{R}_e^x = \{r^{\varepsilon,x} \mid \varepsilon \in \mathcal{R}_e\}$$

for $x \in \{\texttt{obs}, \texttt{clock}, \texttt{spr}, \texttt{pr}\}$

Slide 34

Say two systems $\mathcal{R}, \mathcal{R}'$ are *isomorphic* if there exists a bijection $f : \mathcal{R} \to \mathcal{R}'$ such that for all i and points $(r,m), (r',m')$ of $\mathcal{R}$ we have $(r,m) \sim_i (r',m')$ iff $(f(r),m) \sim_i (f(r'),m')$.

Proposition: Let $f : \mathcal{R} \to \mathcal{R}'$ be an isomorphism and let π and π' be interpretations such that $\pi'(f(r)(m), p) = \pi(r(m), p)$ for all $r \in \mathcal{R}$, $m \in \mathbf{N}$ and $p \in \mathit{Prop}$. Then for all points (r,m) of $\mathcal{R}$ and $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,\bigcirc,\mathcal{U},C\}}$, we have $(\mathcal{R},\pi)(r,m) \models \varphi$ iff $(\mathcal{R}',\pi')(f(r),m) \models \varphi$.

Slide 35

Proposition:

1. A system $\mathcal{R}$ is synchronous iff it is isomorphic to a system $\mathcal{R}_e^{\texttt{clock}}$ for some set $\mathcal{R}_e$ of some environment and some set of observation functions
2. A system $\mathcal{R}$ is a system with perfect recall iff it is isomorphic to a system $\mathcal{R}_e^{\texttt{pr}}$ for some set $\mathcal{R}_e$ of some environment and some set of observation functions
3. A system $\mathcal{R}$ is a system with synchrony and perfect recall iff it is isomorphic to a system $\mathcal{R}_e^{\texttt{spr}}$ for some set $\mathcal{R}_e$ of some environment and some set of observation functions

Slide 36

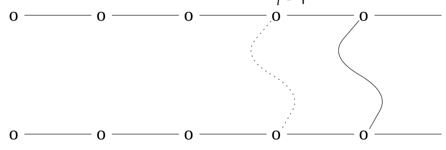
Complexity results (Halpern and Vardi 86,88)

Class of systems	$\mathcal{L}_{\{K_1,\bigcirc,\mathcal{U}\}}$	$\mathcal{L}_{\{K_1,\ldots,K_n,\bigcirc,\mathcal{U}\}}$	$\mathcal{L}_{\{K_1,\ldots,K_n,C,\bigcirc,\mathcal{U}\}}$
$\mathcal{C}$, $\mathcal{C}^{\texttt{uis}}$, $\mathcal{C}^{\texttt{sync}}$, $\mathcal{C}^{\texttt{uis,sync}}$	PSPACE	PSPACE	EXPTIME
$\mathcal{C}^{\texttt{pr}}$, $\mathcal{C}^{\texttt{pr,uis}}$, $\mathcal{C}^{\texttt{pr,sync}}$, $\mathcal{C}^{\texttt{pr,sync,uis}}$	TIME($2^{2^{p(n)}}$)	nonelementary	Π_1^1

Slide 37

An Axiom for Synchronous Systems with Perfect Recall

$\mathrm{KT}^{\mathtt{pr,sync}}$: $K_i \bigcirc \varphi \rightarrow \bigcirc K_i \varphi$



Slide 38

An Axiom for Asynchronous Systems with Perfect Recall

$\mathrm{KT}^{\mathtt{pr}}$:

$$K_i\varphi_1 \wedge \bigcirc(K_i\varphi_2 \wedge \neg K_i\varphi_3) \rightarrow$$
$$\neg K_i \neg \{(K_i\varphi_1) U [(K_i\varphi_2) U \neg\varphi_3]\}$$

Slide 39

A Characterization of Perfect Recall

Let I be an interpreted system. Then the following are equivalent:

(a) I is a system with perfect recall.

(b) For all agents i, for all runs r, s and for all numbers n, m, if $(r, m+1) \sim_i (s, m)$ then either $(r, m) \sim_i (s, m)$ or there exists a number $l < m$ such that $(r, m) \sim_i (s, l)$ and for all k with $l < k \leq m$ we have $(r, m+1) \sim_i (s, k)$.

Slide 40

Class of Systems	Complete Axiomatization
$\mathcal{C}$, $\mathcal{C}^{\mathtt{uis}}$, $\mathcal{C}^{\mathtt{sync}}$, $\mathcal{C}^{\mathtt{uis,sync}}$	S5(C)$_m$+LT
$\mathcal{C}^{\mathtt{pr}}$, $\mathcal{C}^{\mathtt{pr,uis}}$	S5$_m$+ LT +KT$^{\mathtt{pr}}$
$\mathcal{C}^{\mathtt{pr,sync}}$, $\mathcal{C}^{\mathtt{pr,sync,uis}}$	S5$_m$+ LT + KT$^{\mathtt{pr,sync}}$

Slide 41

Branching Time

Extend the temporal language to a variant of CTL* (Emerson & Halpern)

if φ is a formula, then so is

1. $\mathtt{A}\varphi$ (read "on all paths φ")
2. $\mathtt{E}\varphi$ (read "on some path φ").

Slide 42

Two runs r, r' are said to be equivalent to time n, if $r[0\ldots n] = r'[0\ldots n]$.

$(I, r, n) \models \mathtt{A}\varphi$ if for all runs r' of I that are equivalent to r to time n, we have $(I, r', n) \models \varphi$.

(This is the *bundle* semantics (Burgess, Stirling).)

Slide 43

Axioms for Branching Time: AXB

B1. $p \rightarrow \mathtt{A}p$, where p is atomic

B2. $\exists p \rightarrow p$, where p is atomic

B3. $\mathtt{A}\varphi \rightarrow \varphi$

B4. $\mathtt{A}(\varphi \rightarrow \psi) \rightarrow (\mathtt{A}\varphi \rightarrow \mathtt{A}\psi)$

B5. $\mathtt{A}\varphi \rightarrow \mathtt{A}\mathtt{A}\varphi$

B6. $\exists\varphi \rightarrow \mathtt{A}\exists\varphi$

RB. From φ infer $\mathtt{A}\varphi$.

Slide 44

Interaction Axioms

FC. $\mathtt{A}\bigcirc\varphi \rightarrow \bigcirc\mathtt{A}\varphi$

Theorem: AXB + LT + FC is sound and complete for $\mathcal{L}_{\{\mathtt{A},\bigcirc,\mathcal{U}\}}$ in the class of all interpreted systems.

Slide 45

An Interaction between Knowledge and Branching

KB. $K_i\varphi \rightarrow \mathtt{A}K_i\varphi$

Slide 46

Class of Systems	Complete Axiomatization
$\mathcal{C}$, $\mathcal{C}^{\mathtt{uis}}$, $\mathcal{C}^{\mathtt{sync}}$, $\mathcal{C}^{\mathtt{uis,sync}}$	S5(C)$_m$+ AXB + LT + FC + KB
$\mathcal{C}^{\mathtt{pr}}$, $\mathcal{C}^{\mathtt{pr,uis}}$	S5$_m$+ AXB + LT + FC + KB +KT$^{\mathtt{pr}}$
$\mathcal{C}^{\mathtt{pr,sync}}$, $\mathcal{C}^{\mathtt{pr,sync,uis}}$	S5$_m$+ AXB + LT + FC + KB + KT$^{\mathtt{pr,sync}}$

Slide 47

Related Work

Ladner and Reif (TARK 86)

$\mathcal{C}^{\mathtt{pr}}$ and $\mathcal{C}^{\mathtt{pr,nl},uis}$ with respect to $\mathcal{L}_{\{K_1,\ldots,K_n,\mathtt{A}\circ\varphi,\mathtt{A}\Box\varphi\}}$

discuss the axioms

KT1$'$. $K_i\mathtt{A}\circ\varphi \rightarrow \mathtt{A}\circ K_i\varphi$.

KT2$'$. $K_i\mathtt{A}\Box\varphi \rightarrow \mathtt{A}\Box K_i\varphi$.

Completeness for KT1$'$ by Halpern and Vardi (IBM TR)

Slide 48

Branching Time as a Special Case of The Logic of Knowledge

Compare:

KT$^{\mathtt{pr,sync}}$. $K_i \circ \varphi \rightarrow \circ K_i\varphi$, $i = 1,\ldots,m$.

FC. $\mathtt{A}\circ\varphi \rightarrow \circ\mathtt{A}\varphi$

Say i *has complete information* in $\mathcal{I}$ if for all points $(r,m),(r',m')$ of $\mathcal{I}$, if $r_i(n) = r'_i(n')$ then $r(n) = r'(n)$.

If i has synchronous perfect recall and complete information in $\mathcal{I}$, then $\mathcal{I},(r,m) \models K_i\varphi$ iff $\mathcal{I},(r,m) \models \mathtt{A}\varphi$.

(In Z+AFA, can define i has complete information as $r_i(n) = r(n)$)