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

Part 2: Model Checking Knowledge

Slide 3

The Verification Problem

Given a particular scenario, and a claim about the evolution of states of knowledge in the example, prove formally that the claim is correct.

Slide 4

Proof Theoretic Approach

Encode the (runs, system) of the scenario as a formula φ_S.

Encode the claim as a formula φ_C.

Prove validity of $\varphi_S \rightarrow \varphi_C$.

Difficulty/Inconvenience: the environment is common knowledge, $\mathcal{L}_{\{K_1,\ldots,K_n,C,\bigcirc,\mathcal{U}\}}$ not axiomatizable in the context of $\mathtt{pr}$

Slide 5

Model Checking

represent the scenario as a model M

represent the claim as a formula φ_C

show by algorithmic means that $M \models \varphi_C$

Slide 6

Environments (transition form)

An *environment* in transition form is a tuple of the form $E = \langle S_e, I_e, T, O, \pi_e \rangle$ where

1. S_e is a set of *states of the environment*.
2. $I_e \subseteq S_e$, is the set of *initial states* of the environment.
3. $T \subseteq S_e \times S_e$ is a transition relation.
4. O is a tuple $\langle O_1, \ldots, O_n \rangle$ such that for each $i = 1..n$, $O_i : S_e \to O$ is an observation function O.
5. $\pi_e : S_e \times Prop \to \{0,1\}$ is a valuation.

Slide 7

Assume T is *serial*: $\forall s \in S_e \exists t \in S_e(sTt)$

A *run* of an environment E is an *infinite* sequence $\varepsilon = s_0 s_1 \ldots$ of states of E such that

1. $s_0 \in I_e$,
2. $s_k T s_{k+1}$ for all $k \geq 0$,

A *trace* of E is a *finite* sequence $\rho = s_0 \ldots s_m$ of states satisfying conditions 1 and 2.

Slide 8

Local state defined wrt a view

Let ε be a run of E. A *view* associates a local state with each agent at each point of time, determining a mapping $\varepsilon^v : \mathbf{N} \to L^n \times S_e$

In all cases $\varepsilon_e^v(m) = \varepsilon(m)$

Examples:

1. The *observational* view: $\varepsilon_i^{\mathtt{obs}}(m) = O_i(\varepsilon(m)))$
2. The *synchronous perfect recall* view: $\varepsilon_i^{\mathtt{spr}}(m) = O_i(\varepsilon(0)) \ldots O_i(\varepsilon(m))$
3. The *asynchronous perfect recall* view: $\varepsilon_i^{\mathtt{pr}}(m)$ is $\varepsilon_i^{\mathtt{spr}}(m)$ with consecutive repetitions removed.

Slide 9

System Generated by an Environment wrt a View

Let *v* be a view of an environment E. Define $I^v(E) = (\mathcal{R}^v(E), \pi)$ to be the interpreted system with

1. $\mathcal{R}^v(E)$ the set of ε^v such that ε is a run of E.
2. $\pi(r(m), p) = \pi_e(r_e(m), p)$ for all $r \in \mathcal{R}^v(E)$, $p \in$ *Prop*

Slide 10

Recall, for each agent i we define the relation $\sim_i$ on points by $(r,m) \sim_i (r',m')$ if $r_i(m) = r'_i(m)$.

Given a point (r,m) of $I^v(E)$, define

$$trace(r,m) = r_e(0) \ldots r_e(m).$$

For two traces τ, τ', define $\tau \sim_i \tau'$ if there exist points (r,m), (r',m') such that $trace(r,m) = \tau$ and $trace(r',m') = \tau'$ and $(r,m) \sim_i (r',m')$.

Slide 11

Let $v \in \{\mathtt{obs}, \mathtt{pr}, \mathtt{spr}\}$

Proposition: Suppose $(r,m), (r',m)$ are points of $I^v(E)$ and let $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$. If $trace(r,m) = trace(r',m')$ then $I^v(E), (r,m) \models \varphi$ iff $I^v(E), (r',m') \models \varphi$.

If τ is a trace of E, write $I^v(E), \tau \models \varphi$ when $I^v(E), (r,m) \models \varphi$ for some point (r,m) with $trace(r,m) = \tau$.

Slide 12

Model Checking $\mathcal{L}_{\{K_1,\ldots,K_n,C\}}$ at a Trace

Let *v* be a view.

Problem: Given a finite environment E, a trace τ of E and a formula $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$, determine if $I^v(E), \tau \models \varphi$.

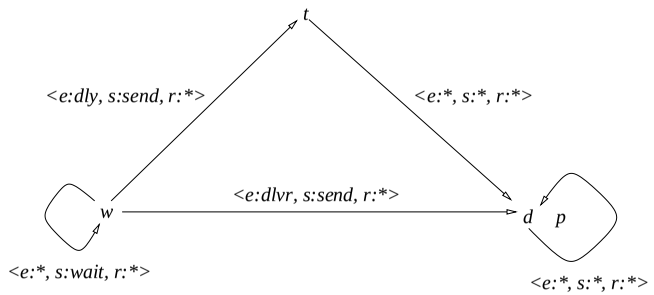
Comment: $I^v(E)$ is not a finite structure, so this is an *infinite state* model checking problem

Slide 13

Consider an environment E in which

- agent s (sender) can send the single message "hello" to agent r (receiver), but can only do this once
- agent s observes a variable that records whether or not the message has been sent
- agent r observes a variable that records whether the message has arrived
- the channel either delivers the message either immediately, or with a delay of one second
- the proposition p means "the message has arrived"

Slide 14



$I_e = \{w\}$ $\qquad \pi_e(x,p) = \textbf{true}$ iff $x = d$.

$O_s(w) = \bot, \quad O_s(t) = O_s(d) = sent$

$O_r(w) = O_r(t) = \bot, \quad O_r(d) = rcvd$

Slide 15

$\texttt{traces}(E) = \{w^k d^m \mid k > 0, m \geq 0\} \cup \{w^k t d^m \mid k > 0, m \geq 0\}$

Slide 16

Message transmission example (observational view)

Suppose agent s sends the message at time 1, and the environment delivers the message immediately, then the agents wait for $n-1$ ticks of the clock, i.e. consider the trace wd^{n-1}

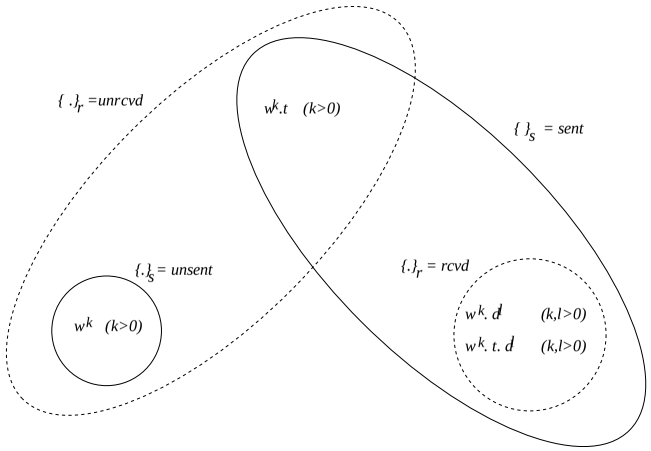
Under the observational view,

- $wd^{n-1} \sim_r \tau$ implies $\texttt{fin}(\tau) = d$
- $wd^{n-1} \sim_s w^{n-1}t$

Thus $I^{\texttt{obs}}(E), wd^{n-1} \models K_r p$ but

$I^{\texttt{obs}}(E), wd^{n-1} \models \neg K_s p$.

Slide 17



Slide 18

Message transmission example (synchronous perfect recall view)

Under the perfect recall view,

$$\{wd^{n-1}\}_s^{\mathtt{spr}} = \bot \cdot (sent)^{n-1} = \{wtd^{n-2}\}_s^{\mathtt{spr}}$$

so $wd^{n-1} \sim_s wtd^{n-2}$.

More generally, for each length n:

$$wd^{n-1} \sim_s wtd^{n-2} \sim_r w^2d^{n-2} \sim_s w^2td^{n-3} \ldots$$

$$\ldots \sim_r w^{n-1}d \sim_s w^{n-1}t \sim_r w^n$$

Slide 19

$I^{\mathtt{spr}}(E),\ wd^{n-1} \models (K_rK_s)^j p$ for all $j < n-1$,

$I^{\mathtt{spr}}(E),\ wd^{n-1} \models \neg(K_rK_s)^{n-1}p$

$I^{\mathtt{spr}}(E),\ \tau \models \neg Cp$ for all $\tau \in traces(E)$

Slide 20

S5$_n$ Kripke Structures

An S5$_n$ Kripke structure is a tuple $M = \langle W, \mathcal{K}_1, \ldots, \mathcal{K}_n, \pi \rangle$ where

1. W is a set of worlds
2. $\mathcal{K}_i$ is an equivalence relation on W for each $i = 1 \ldots n$
3. $\pi : W \times Prop \to \{0,1\}$ is an assignment

Define $\mathcal{K}_C = (\bigcup_i \mathcal{K}_i)^*$

1. $M, w \models K_i\varphi$ if $M, w' \models \varphi$ for all $w'\,\mathcal{K}_i w$
2. $M, w \models C\varphi$ if $M, w' \models \varphi$ for all $w'\,\mathcal{K}_C w$

Slide 21

Given an environment E and view *v*, define $M_E^v = \langle \mathtt{traces}(E), \sim_1, \ldots, \sim_n, \pi \rangle$ where the $\sim_i$ are the equivalence relations on traces defined wrt the view and $\pi(\tau, p) = \pi_e(\mathtt{fin}(\tau), p)$.

Proposition: For $\tau \in traces(E)$ and $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$,

$$M_E^v, \tau \models \varphi \quad \text{iff} I^v(E), \tau \models \varphi$$

Slide 22

Model Checking in Finite Kripke Structures

For a finite Kripke structure M define $|M|$ to be the number of symbols needed to write down M, all edges of the equivalence relations included.

Proposition: Given a finite S5$_n$ structure M and a formula $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$ determining $M, w \models \varphi$ can be done in time $O(|M| \cdot |\varphi|)$.

Slide 23

Algorithm:

For each subformula ψ of φ, in order of increasing size,
For each world *w* of M label *w* by φ or $\neg\varphi$:
CASE

1. if $\psi = p$ then label *w* by p iff $\pi(w, p) = 1$
2. if $\psi = \neg\psi'$ then label *w* by ψ iff *w* not labelled ψ'
3. if $\psi = \psi_1 \wedge \psi_2$ then label *w* by ψ iff *w* labelled by ψ_1 and by ψ_2
4. if $\psi = K_1\psi'$label *w* by ψ iff w' labelled ψ' for all $w' \mathcal{K}_i w$

END CASE
END (for *w*)

Slide 24

if $\psi = C\psi'$ then

1. do a depth first search from all worlds labelled $\neg\psi'$ through edges $\mathcal{K}_i$, labelling all worlds reached $\neg C\psi'$
2. label all worlds not reached in the above by $C\psi'$

END (for ψ)

Output YES if *w* is labelled φ

Slide 25

Model Checking at a Trace (Observational View)

Let $E = \langle S_e, I_e, T, O, \pi_e \rangle$ be a finite state environment.

A state $t \in S_e$ is *reachable* if sT^*t for some $s \in I_e$.

Define $M = \langle W, \mathcal{K}_1, \ldots, \mathcal{K}_n, \pi \rangle$ by

1. W is the set of reachable states of E.
2. $s\mathcal{K}_i t$ iff $O_i(s) = O_i(t)$
3. $\pi = \pi_e$

Slide 26

Proposition: For $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$, we have $I^{\mathtt{obs}}(E), \tau \models \varphi$ iff $M, \mathtt{fin}(\tau) \models \varphi$.

Corollary: For $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$, determining whether $I^{\mathtt{obs}}(E), \tau \models \varphi$ can be done in time $O(|E| \cdot |\varphi|)$.

Slide 27

Model Checking at a Trace (Synchronous Perfect Recall View)

Theorem: [van der Meyden, TARK 94]
Determining $I^{\mathtt{spr}}(E), \tau \models \varphi$ for $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n,C\}}$ is in PSPACE.

Theorem: [van der Meyden, TARK94] There exists an environment E for two agents and a propositional constant p such that the set

$$\{\tau \in \mathtt{traces}(E) \mid I^{\mathtt{spr}}(E), \tau \models C_{\{1,2\}} p\}$$

is PSPACE complete.

Slide 28

Model Checking at a Trace - Synchronous Perfect Recall View

Theorem: [van der Meyden, TARK94] Let E be a finite environment. For formulae $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n\}}$ of (alternation) depth bounded by k, and $\tau \in \mathtt{traces}(E)$, $I^{\mathtt{spr}}(E), \tau \models \varphi$ can be decided in time $O(C_k(E) \cdot (|\varphi| + |\tau|))$ where $C_k(E) = exp(n \times |S_e|, k)/n$.

Here $exp(a, b)$ is the function defined by $exp(a, 0) = a$ and $exp(a, b+1) = a2^{exp(a,b)}$.

Slide 29

Model reduction for a single agent

Let $E = \langle S_e, I_e, T, O, \pi_e \rangle$ be an environment for one agent.

Let S' be the set of reachable states of E.

Define $M = \langle W, \mathcal{K}_1, \pi \rangle$ where

1. $W = \{ (s,U) \mid s \in S',\ U \in \mathcal{P}(S') \text{ and } s \in U \}$,
2. $(s,U)\,\mathcal{K}_1(t,V)$ iff $U = V$,
3. $\pi((s,U),p) = \pi_e(s,p)$

Slide 30

Define $F : \mathtt{traces}(E) \to W_M$ by

$$F(\tau) = (\mathtt{fin}(\tau), U(\tau))$$

where

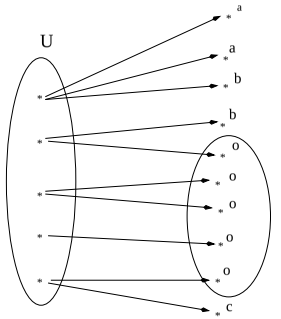
$$U(\tau) = \{\mathtt{fin}(\tau') \mid \tau' \sim_1 \tau\}$$

Proposition:
For all $\tau \in \mathtt{traces}(E)$, and $\varphi \in \mathcal{L}_{\{K_1\}}$, we have $I^{\mathtt{spr}}(E), \tau \models \varphi$ iff $M, F(\tau) \models \varphi$

Slide 31

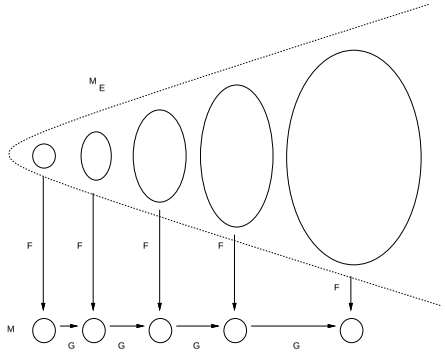
Define $G : \mathcal{P}(S_e) \times O \to \mathcal{P}(S_e)$ by

$$G(U,o) = \{t \mid sTt \text{ and } O_1(t) = o \text{ for some } s \in U\}$$



Slide 32

Proposition: For all $\tau \in \mathtt{traces}(E)$, $U(\tau s) = G(U(\tau), O_1(s))$
Thus, $F(\tau)$ can be computed in time $O(|r|)$.



Slide 33

k-trees

Fix an environment E.

A 0-tree is a tuple $\langle s, \emptyset, ...\emptyset \rangle$ where s is a state of E.

A $k+1$-tree is a tuple $\langle s, U_1, ...U_n \rangle$, where

1. s is a state of E and
2. U_i is a set of k-trees, for $i = 1 \ldots n$.

Write $\mathcal{T}_k$ for the set of k-trees.

Proposition: Let $k \geq 0$ be a natural number and E be a finite environment for n agents with states S_e. Then $|\mathcal{T}_k| \leq C_k(E)$

Slide 34

From traces to trees

Suppose we are given a system $I^v(E)$ with accessibility relations $\sim_i$.

Define $F_k : \mathtt{traces}(E) \to \mathcal{T}_k$ by

1. $F_0(\tau) = \langle \mathtt{fin}(\tau), \emptyset, \ldots, \emptyset \rangle$.
2. $F_{k+1}(\tau) = \langle \mathtt{fin}(\tau), U_1, \ldots, U_m \rangle$, where for each agent i,

$$U_i = \{\, F_{k-1}(\tau') \mid \tau' \sim_i \tau \,\}$$

Slide 35

Interpreting depth k formulae at k-trees

Let $\langle s, U_1, \ldots, U_n \rangle$ be a k-tree and φ a formula of $\mathcal{L}_{\{K_1,\ldots,K_n\}}$ of knowledge depth at most k. Define

$\langle s, U_1, \ldots, U_n \rangle \models_k p$ if $\pi_e(s, p) = 1$

$\langle s, U_1, \ldots, U_n \rangle \models_k K_i\varphi$ if $w \models_{k-1} \varphi$ for all $w \in U_i$.

(booleans as usual)

Slide 36

Theorem: For all traces τ of E and all $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n\}}$ of knowledge depth at most k, we have

$$I^v(E), \tau \models \varphi \quad \text{iff} \quad F_k(\tau) \models_k \varphi$$

Slide 37

Computing F_k with respect to `spr`

Theorem: There exists a function $G_k : \mathcal{T}_k \times S_e \to \mathcal{T}_k$ such that for all traces $\tau \cdot s$ of E,

$$F_k(\tau \cdot s) = G_k(F_k(\tau), s)$$

Slide 38

Incremental Computation of F_k

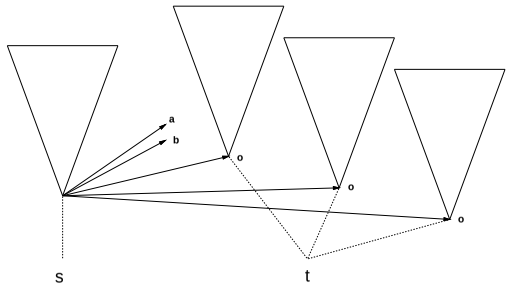
Define $G_k : \mathcal{T}_k \times S \to \mathcal{T}_k$ by

$$G_{k+1}(\langle s, U_1, \ldots, U_n \rangle, t) = \langle t, V_1, \ldots, V_n \rangle$$

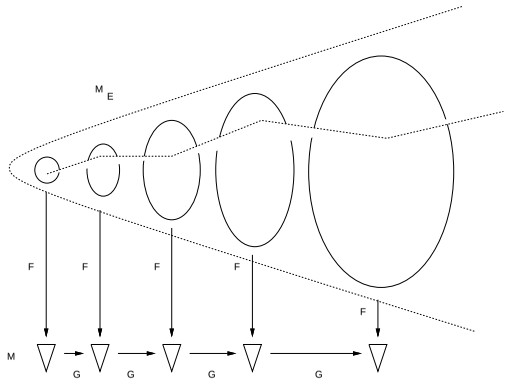
where

$$V_i = \{G_k(v, t') \mid v \in U_i,\ \mathtt{root}(v) T t',\ O_i(t') = O_i(t)\}$$

Slide 39



Slide 40



Slide 41

Model Checking at a Trace - (Asynchronous) Perfect Recall View

Theorem: [van der Meyden, TARK94] There exists an environment E for two agents and a propositional constant p such that the set

$$\{\tau \in \mathtt{traces}(E) \mid I^{\mathrm{pr}}(E), \tau \models C_{\{1,2\}} p\}$$

is undecidable.

Slide 42

Theorem: [van der Meyden, TARK94] Let E be a finite environment. For formulae $\varphi \in \mathcal{L}_{\{K_1,\ldots,K_n\}}$ of alternation depth bounded by k, and $\tau \in \mathtt{traces}(E)$, $I^{\mathrm{pr}}(E), \tau \models \varphi$ can be decided in time $O(C_k(E) \cdot (|\varphi| + |\tau|))$.

Slide 43