

Slide 1

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

Algorithmic Verification for Epistemic Logic

Ron van der Meyden

University of New South Wales/National ICT Australia

Slide 3

References

Constructing finite state implementations of knowledge based programs with perfect recall, R. van der Meyden, PRICAL workshop on theoretical and practical foundations of intelligent agents, 1996

Slide 2

Part 3:

Updating Kripke Structures

Model Checking Knowledge and Branching Time

Slide 4

Simple Environments for Synchronous Perfect Recall

Define $\Phi : \text{traces}(E) \rightarrow \mathcal{P}(\mathcal{L}_{\{K_1, \dots, K_n, C\}})$ by

$$\Phi(\tau) = \{\varphi \in \mathcal{L}_{\{K_1, \dots, K_n, C\}} \mid I^{\text{SPR}}(E), \tau \models \varphi\}$$

An environment E is *simple* if $\{\Phi(\tau) \mid \tau \in \text{traces}(E)\}$ is finite, i.e. agents have only a finite number of possible states of knowledge in that environment (with the synchronous perfect recall semantics)

Example: Environment E with initial states I is a *broadcast* environment if all initial states t are sources (sTt for no $s \in S$) and for all non-initial states $s \in S \setminus I$,

$$O_1(s) = O_2(s) = \dots = O_n(s)$$

Proposition: Every broadcast environment is simple

(The definition can be strengthened with the same result!)

Slide 5

Progression Structures

A *progression structure* for environment E is a pair $\langle M, \sigma \rangle$ consisting of an $S5_n$ Kripke structure $M = \langle W, \mathcal{R}_1, \dots, \mathcal{R}_n, \pi \rangle$ and a *state mapping* $\sigma : W \rightarrow S_e$ such that

$$\pi(w, p) = \pi_e(\sigma(w), p)$$

for all $w \in W$ and $p \in Prop$

Example: $P_{E,n} = \langle M_n, \text{fin} \rangle$, where M_n is the substructure of M_E^{SDr} consisting of the traces of length n

Slide 6

The environment E operates on its progression structures by

$$\langle M, \sigma \rangle * E = \langle M', \sigma' \rangle$$

where $M' = \langle W', \mathcal{R}'_1, \dots, \mathcal{R}'_n, \pi' \rangle$ is the Kripke structure with

1. $W' = \{(w, s) \mid w \in W, s \in S_e, \sigma(w)Ts\}$
2. $(w, s) \mathcal{R}'_i(v, t)$ iff $w \mathcal{R}_i v$ and $O_i(s) = O_i(t)$
3. $\pi'((w, s), p) = \pi_e(s)$
4. $\sigma'((w, s)) = s$

Proposition: $P_{E,n+1}$ is isomorphic to $P_{E,n} * E$

Slide 7

Zig-Zag Morphisms

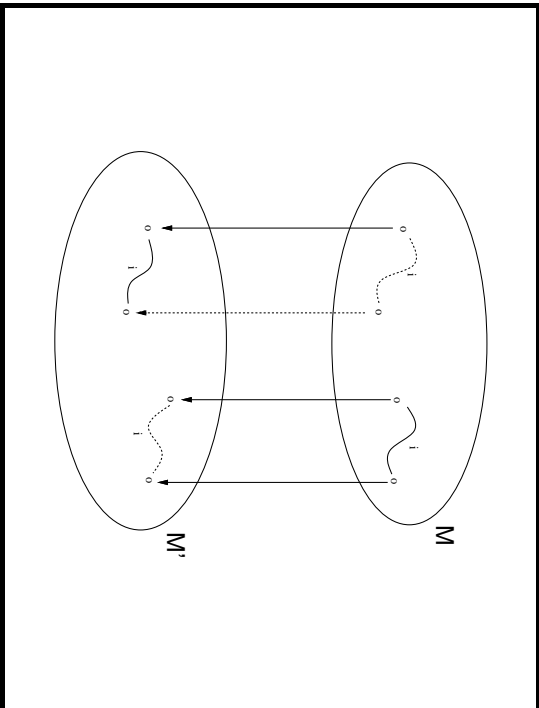
Let $M = \langle W, \mathcal{R}_1, \dots, \mathcal{R}_n, \pi \rangle$ and $M' = \langle W', \mathcal{R}'_1, \dots, \mathcal{R}'_n, \pi' \rangle$ be Kripke structures

A *zig-zag morphism* (bisimulation) from $\langle M, \sigma \rangle$ to $\langle M', \sigma' \rangle$ is a mapping α from W to W' such that

1. $\pi'(\alpha(u), p) = \pi(u, p)$ for all $u \in W$ and $p \in Prop$,
2. $\sigma(w) = \sigma'(\alpha(w))$ for all $w \in W$,
3. $u \mathcal{R}_i v$ implies $\alpha(u) \mathcal{R}'_i \alpha(v)$, for all $i = 1 \dots n$ and $u, v \in W$,
4. if $\alpha(u) \mathcal{R}'_i v'$ then there exists $v \in W$ with $\alpha(v) = v'$ and $u \mathcal{R}_i v$.

Slide 8

Slide 9



Slide 11

Say propositions determine state in E when for all states $s, t \in S_e$, if $\pi_e(s, p) = \pi_e(t, p)$ for all $p \in Prop$ then $s = t$

Theorem: If propositions determine state in the environment E then the following are equivalent:

- (i) E is simple
- (ii) The progression structures $\text{red}(P_{E, n})$ take on a finite number of values (up to isomorphism)
- (iii) There exists a zig-zag morphism from M_E^{SPR} to a finite Kripke structure

Structure Minimization

Theorem: If α is a zig-zag morphism from $\langle M, \sigma \rangle$ to $\langle M', \sigma' \rangle$ then for all formulae $\varphi \in \mathcal{L}_{\{K_1, \dots, K_n, C\}}$ and all worlds w of M , we have $M, w \models \varphi$ if and only if $M', \alpha(w) \models \varphi$.

Theorem: For all finite progression structures P there exists a minimal size progression structure $\text{red}(P)$ with a zig-zag morphism $\alpha : P \rightarrow \text{red}(P)$. The structure $\text{red}(P)$ is unique up to isomorphism, and may be computed in time $O(|P| \cdot \log |P|)$.

Slide 12

Incremental Computation

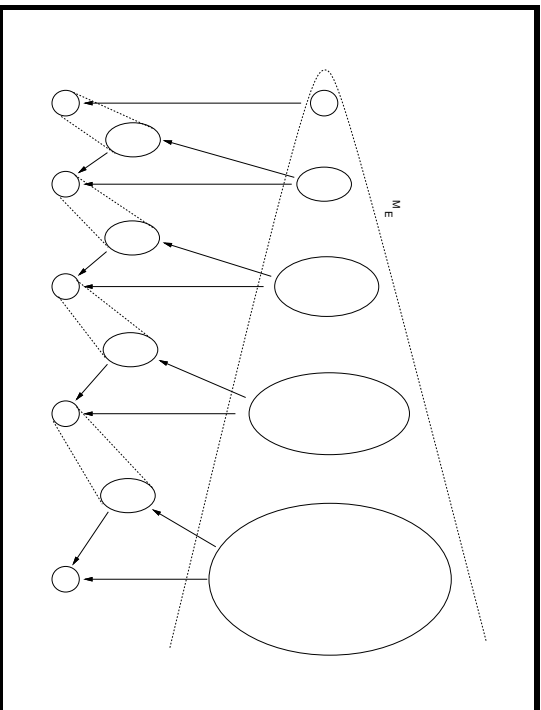
Theorem: For all environments E , and all n ,

$$\text{red}(P_{E, n+1}) \simeq \text{red}(\text{red}(P_{E, n}) * E)$$

Theorem: There exists an algorithm $A(E, \tau, \varphi)$ such that for all environments E in which propositions determine state

1. For all traces τ and sentences $\varphi \in \mathcal{L}_{\{K_1, \dots, K_n, C\}}$, $A(E, \tau, \varphi) = \text{'yes'}$ iff $I^{\text{SPR}}(E), \tau \models \varphi$
2. If E is a fixed simple environment then $A(E, \tau, \varphi)$ runs in time $O(|\tau| + |\varphi|)$

Slide 13



Slide 15

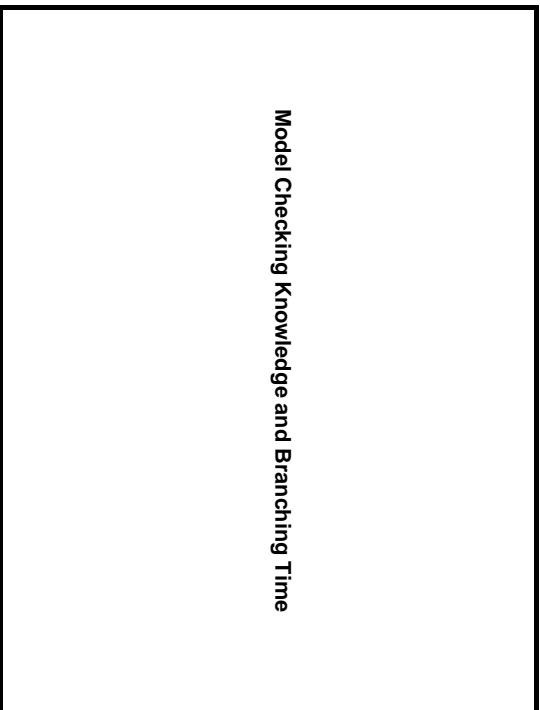
Environments (transition form) with fairness

An *environment* in transition form with fairness condition is a tuple of the form $E = \langle S_e, I_e, T, O, \pi_e, \alpha \rangle$ 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 $\langle O_1, \dots, O_n \rangle$ such that for each $i = 1..n$, $O_i : S_e \rightarrow Obs$ is an observation function O .
5. $\pi_e : S_e \times Prop \rightarrow \{0, 1\}$ is a valuation.
6. $\alpha \subseteq S_e$ is a Büchi acceptance condition

Slide 14

Model Checking Knowledge and Branching Time



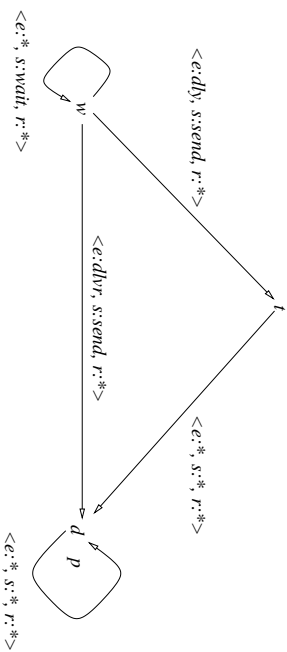
Slide 16

A *run* of an environment E with fairness condition is an *infinite* sequence $\varepsilon = s_0 s^1 \dots$ of states of E such that

1. $s_0 \in I_e$,
2. $s_k T s_{k+1}$ for all $k \geq 0$,
3. some $s \in \alpha$ occurs infinitely often.

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

Assumption: Every trace of E can be extended to a run of E .



$I_e = \{w\}$ $\pi_e(x, p) = \text{true}$ if $x = d$.
 $O_s(w) = \perp$, $O_s(t) = O_s(d) = \text{sent}$
 $O_r(w) = O_r(t) = \perp$, $O_r(d) = \text{rcvd}$

Slide 17

$\text{traces}(E) = \{w^k d^m \mid k > 0, m \geq 0\} \cup \{w^k t d^m \mid k > 0, m \geq 0\}$
 If $\alpha = \{w, t, d\}$ then
 $\mathcal{R}(E) = \{w^\infty\} \cup \{w^k d^\infty \mid k > 0\} \cup \{w^k t d^\infty \mid k > 0\}$
 If $\alpha = \{d\}$ then
 $\mathcal{R}(E) = \{w^k d^\infty \mid k > 0\} \cup \{w^k t d^\infty \mid k > 0\}$

Slide 18

Realization

Let v be view.

Say that a formula ϕ is realized in E (wrt v) if

$$I^v(E), (r, 0) \models \phi$$

for all runs $r \in \mathcal{R}^v(E)$.

Problem: Given a finite environment E and a formula ϕ of the language of knowledge and (linear/branching) time, decide if ϕ is realized in E wrt v .

Slide 19

Observation: Realization generalizes model checking $\mathcal{L}\{K_1, \dots, K_n, C\}$ at a trace.

Suppose p_s is a proposition true only at state s of environment E . Let τ be the trace $s_0 s_1 \dots s_m$. Then the following are equivalent

1. $I^v(E), \tau \models \phi$
2. $p_{s_0} \rightarrow \bigcirc(p_{s_1} \rightarrow \dots \bigcirc(p_{s_m} \rightarrow \phi))$ is realized in E wrt v .
3. $p_{s_0} \rightarrow \forall \bigcirc(p_{s_1} \rightarrow \dots \forall \bigcirc(p_{s_m} \rightarrow \phi))$ is realized in E wrt v .

Slide 20

Slide 21

Branching Time, Observational View

Theorem: Realization of $\varphi \in \mathcal{L}_{\{K_1, \dots, K_n, C, \forall O, \forall tL, \exists tU\}}$ in E with respect to obs is in PTIME.

Slide 22

Let S' be the set of reachable states of E

label states of $M = \langle S', \mathcal{K}_1, \dots, \mathcal{K}_n, \pi_e \rangle$ by subformulas of φ

1. label s by $K_i \psi$ ($C \psi$) if $s \mathcal{K}_i t$ ($s \mathcal{C} t$) implies t labelled ψ
2. label s by $\forall O \psi$ if $s T t$ implies t labelled by ψ
3. label s by $\exists t (\psi_1 \ \&L \ \psi_2)$ if there exists a sequence $s = s_0, s_1, \dots, s_k$ such that s_k labelled ψ_2 and for $l < k$ $s_l T s_{l+1}$ and s_l is labelled ψ_1

Slide 23

4. label s by $\neg \forall (\psi_1 \ \&L \ \psi_2)$ if there exists a sequence

$s = s_0, s_1, \dots, s_k, \dots, s_m$ of states such that

- (a) $s_l T s_{l+1}$ for all $l < m$
- (b) $s_k = s_m$
- (c) $\{s_k, \dots, s_m\} \cap \alpha \neq \emptyset$
- (d) either s_l is not labelled ψ_2 for all $l \leq m$, or, for the least l such that s_l is labelled ψ_2 there exists $l' < l$ such that $s_{l'}$ is not labelled ψ_1