Slide 1

COMP3152/9152
Lecture 5
Dynamics of Common Knowledge
Ron van der Meyden

Slide 2

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 $i = 1..n$, $O_i : S_e \to \mathcal{O}$ is an observation function $\mathcal{O}$.
5. $\pi_e : S_e \times \mathit{Prop} \to \{0, 1\}$ is a valuation.

Slide 3

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 $\rho = s_0 s_1 \ldots$ 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 $\tau = s_0 \ldots s_m$ of states satisfying conditions 1 and 2.

Slide 4

Local state defined wrt a view

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

In all cases $\rho^v_e(m) = \rho(m)$

Examples:

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

Slide 5

System Generated by an Environment wrt a View

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

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

Slide 6

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 $\mathcal{I}^v(E)$, define

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

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

Slide 7

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

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

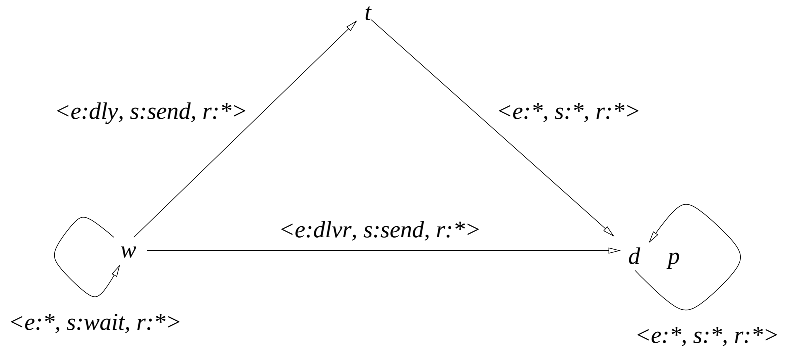
If τ is a trace of E, write $\mathcal{I}^v(E), \tau \models \varphi$ when $\mathcal{I}^v(E), (r, m) \models \varphi$ for some point (r, m) with $\mathit{trace}(r, m) = \tau$.

Slide 8

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 9



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

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

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

lide 10

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

Slide 11

Notation: If $\tau = x_0, \ldots x_k$ is a sequence, then $\texttt{fin}(\tau) = x_n$ is the last element.

Slide 12

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}

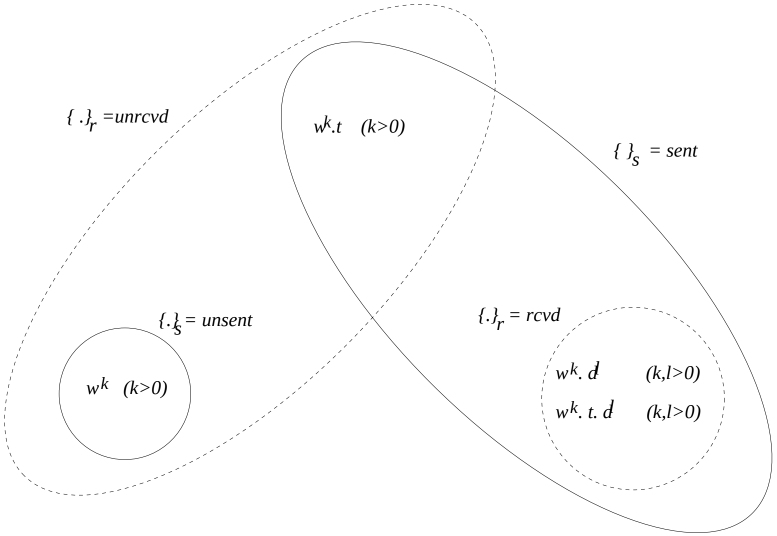
Under the observational view,

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

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

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

Slide 13



Slide 14

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 15

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

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

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

Slide 16

$S5_n$ 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. $\pi : W \times \Phi \rightarrow \{0, 1\}$ is an assignment

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

1. $M, w \models K_i\phi$ if $M, w' \models \phi$ for all $w'\mathcal{K}_iw$
2. $M, w \models C\phi$ if $M, w' \models \phi$ for all $w'\mathcal{K}_Cw$

lide 17

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

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

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

lide 18

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 19

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

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

Slide 20

Progression Structures

A *progression structure* for environment E is a pair $\langle M, \sigma \rangle$ consisting of an $\mathrm{S5}_n$ Kripke structure $M = \langle W, \mathcal{K}_1, \ldots, \mathcal{K}_n, \pi \rangle$ and a *state mapping* $\sigma : W \to S_e$ such that

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

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

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

lide 21

If $P = \langle M, \sigma \rangle$, Write $P, w \models \phi$ if $M, w \models \phi$.

lide 22

The environment E operates on its progression structures by

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

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

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

Slide 23

Proposition: Let $\tau = s_0 \ldots s_k$ be a trace of an environment E and let $\phi \in \mathcal{L}_{\{K_1, \ldots, K_n, C\}}$. Then $\mathcal{I}^{\mathtt{spr}}(E), \tau \models \phi$ iff $P_{E,k}, w_\tau \models \phi$, where $w_\tau = ((\ldots(((s_0, s1), s_2), s_3), \ldots, s_k)$.

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

Slide 24

This means we can check $\mathcal{I}^{\mathtt{spr}}(E), \tau \models \phi$ as follows:

1. Construct $P_{E,0}$,
2. For $i = 1 \ldots k$ construct $P_{E,k} = P_{E,k-1} * E$
3. Check $P_{E,k}, w_\tau \models \phi$ using finite state model checking.

lide 25

References

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