Slide 1

COMP3152/9152
Lecture 3
Semantics for Knowledge and Time
Ron van der Meyden

Reading, FHMV Ch 4

Slide 2

A Model for Runs of a Distributed System

Consider a system for n agents.

For each agent i let L_i be a set of local states of agents i
and let L_e be a set of states of the *environment*

Define the set of *global states* as $\mathcal{G} = L_e \times L_1 \times \ldots \times L_n$, i.e., a global state is a is a tuple $\langle s_e, s_1, \ldots, s_n \rangle$.

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

Slide 3

Runs

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

If $r(m) = (s_e, s_1, \ldots, s_n)$, write $r_i(m)$ for s_i, and $r_e(m)$ for s_e.

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

Slide 4

Distributed Systems

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

Let Φ be a set of propositional constants.

An *interpretation* for $\mathcal{G}$ is a function $\pi : \mathcal{G} \times \Phi \rightarrow \{0, 1\}$.

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

Slide 5

Example: Bit Transmission

A sender S has a bit *bit* that it wants to communicate to a receiver R.

The communications channel is lossy. If a message is sent, it is either delivered immediately (in the next tick of the clock) or lost forever.

So the sender keeps sending until it receives an acknowledgement from the receiver.

Once it receives the message, the receiver keeps sending acknowledgements (forever).

Slide 6

Local states of Sender S: $L_S = \{0, 1, (0, \texttt{ack}), (1, \texttt{ack})\}$

Local states of Receiver R: $L_R = \{\lambda, 0, 1\}$.

Local states of the Environment: L_e: A sequence of pairs of the forms: $(\texttt{sendbit}, \Lambda), (\Lambda, \texttt{sendack}), (\texttt{sendbit}, \texttt{sendack})$, recording the history of actions so far.

Global states: $\mathcal{G} = \{(s_e, s_S, s_R) \mid s_e \in L_e, s_S \in L_S, s_R \in L_R\}$

E.g.:

$(, \langle\rangle, 0, \lambda)$

$(\langle(\texttt{sendbit}, \Lambda)\rangle, 0, \lambda)$ (message lost)

$(\langle(\texttt{sendbit}, \Lambda)(\texttt{sendbit}, \Lambda)\rangle, 0, 0)$ (1st lost, second delivered)

Slide 7

The set of runs of the system is the set $\mathcal{R}$ of sequences $r : \mathbf{N} \to \mathcal{G}$ such that $r(0) = (\langle\rangle, b, \lambda)$, $b \in \{0, 1\}$ and for all $m \geq 0$:

1. If $r(m) = (s, b, \lambda)$ then $r(m+1) = (s \cdot (\texttt{sendbit}, \Lambda), b, \lambda)$ or $r(m+1) = (s \cdot (\texttt{sendbit}, \Lambda), b, b)$ or
2. If $r(m) = (s, b, b)$ then $r(m+1) = (s \cdot (\texttt{sendbit}, \texttt{sendack}), b, b)$ or $r(m+1) = (s \cdot (\texttt{sendbit}, \texttt{sendack}), (b, \texttt{ack}), b)$
3. If $r(m) = (s, (b, \texttt{ack}), b)$ then $r(m+1) = (s \cdot (\Lambda, \texttt{sendack}), (b, ack), b)$

Slide 8

Linear Time Temporal Logic

Extend the language of knowledge and time by the following operators:

$\bigcirc\varphi$ — φ at the next moment of time

$\Box\varphi$ — φ now and at all times in the future

$\Diamond\varphi$ — φ now or at some time in the future

$\phi_1 \;\mathcal{U}\; \phi_2$ – ϕ_1 until ϕ_2, i.e., eventually ϕ_2, and ϕ_1 at all times until then.

Slide 9

Examples:

$\Box(\texttt{rain} \Rightarrow \texttt{wet})$

$\Box(\texttt{dark} - \texttt{clouds} \Rightarrow \bigcirc\,\texttt{rain})$

$\Box(\Diamond\texttt{rain})$
(= it will rain an infinite number of times)

$\Diamond\Box\texttt{dead}$

lide 10

$(\mathcal{I}, r, m) \models p \quad$ if $\pi(r(m))(p) = 1$.

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

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

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

Slide 11

$(\mathcal{I}, r, m) \models \Box\varphi \quad$ if $(\mathcal{I}, r, m') \models \varphi$ for all $m' \geq m$.

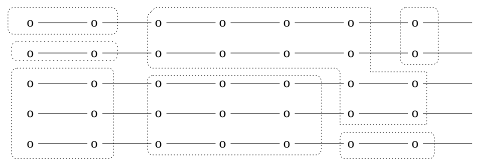
$(\mathcal{I}, r, m) \models \Diamond\varphi \quad$ if $(\mathcal{I}, r, m') \models \varphi$ for some $m' \geq m$.

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

Slide 12

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

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



lide 13

Message Passing Systems

Σ_i - initial states for process i

INT_i - internal actions of i ($\texttt{int}(a,i)$)

MSG - messages μ

Message passing actions of i:

$\texttt{send}(\mu, j, i)$ - i sends message μ to j

$\texttt{receive}(\mu, j, i)$ - i receives message μ from j

lide 14

A *history* for agent i is a sequence consisting of an initial state for i, followed by a sequence of sets of internal and message passing actions of i.

Example:

$s_0\{\texttt{send}(0, R, S)\}\{\texttt{int}(wait, S)\}\{\texttt{send}(0, R, S)\}$
$\{\texttt{int}(wait, S)\}\{\texttt{send}(0, R, S), \texttt{receive}(\texttt{ack}, R, S)\}$

is a history for S

Slide 15

Let $r_i(m)$ be a history for i for all m.

Say that an event e is *in* $r_i(m)$ if it occurs in one of the sets in the sequence $r_i(m)$

Say that event e *occurs in round* m if e is in $r_i(m)$ but not in $r_i(m-1)$.

Slide 16

A system $\mathcal{R}$ is a *message passing system* based on the sets Σ_i, INT_i ($i = 1 \ldots n$) and MSG if for all points (r, m) of $\mathcal{R}$ and agents i:

MP1. $r_i(m)$ is a history over Σ_i, INT_i and MSG

MP2. For every event $\texttt{receive}(\mu, j, i)$ in $r_i(m)$ there is an event $\texttt{send}(\mu, i, j)$ in $r_i(m)$

MP3. $r_i(0)$ is a sequence of length 1, and for all m, $r_i(m+1) = r_i(m)$ or $r_i(m+1) = r_i(m) \cdot X$ where X is a set of events of i.

lide 17

A message passing system is *reliable* if every message is eventually received, i.e.

MP4. If $\texttt{send}(\mu, j, i)$ is in $r_i(m)$ then there exists $m' \geq m$ such that $\texttt{receive}(\mu, i, j)$ is in $r_j(m')$.

lide 18

Asynchronous Message Passing Systems

In an asynchronous system, there are no relationships between the rates of progress of different agents: the next action of any agent could take an arbitrary amount of time to happen (but we still have that a send must occur before a receive).

Formally, say a set V of histories is *prefix-closed* if $h \in V$ and g a prefix of h implies $g \in V$.

Slide 19

Given prefix-closed sets V_1, V_n of histories, let $\mathcal{R}(V_1, \ldots, V_n)$ be the set of all runs r satisfying MP1-MP3 such that for all i and m, we have $r_i(M) \in V_i$.

A system $\mathcal{R}$ is an *asynchronous message passing system* if there exist sets $V_1, \ldots, V_n$ such that $\mathcal{R} = \mathcal{R}(V_1, \ldots, V_n)$.

Slide 20

What does an agent know in an a.m.p. system?

if i receives μ from j, then i knows that j sent μ.

But not...

– what time it is

– how long ago j sent μ

lide 21

Potential Causality (Lamport)

Assumption: each event (send/receive/internal) occurs at most once in a run

For events e, e', define $e \xrightarrow{r} e'$ if either

1. e' is a receive event and e is the corresponding send event
2. for some process i, events e and e' are both in i's history and e is before e', or
3. for some event e'' we have $e \xrightarrow{r} e''$ and $e'' \xrightarrow{r} e'$.

lide 22

Given events e, e', define a proposition $Prec(e, e')$ by

$\pi(r(m))(Prec(e, e')) =$ `true` if e, e' both occur in r by time m and e occurs no later than e' in r.

(Assume environment component of global state records all events in the order they occur.)

Slide 23

Proposition 4.4.3: Let G be all processes, let $\mathcal{R}$ be an a.m.p. system and $\mathcal{I} = (\mathcal{R}, \pi)$ with π as defined above. Then $(\mathcal{I}, r, m) \models D_G(Prec(e, e'))$ iff e, e' have both occurred in r by time m and $e \xrightarrow{r} e'$.

Slide 24

Process Chains

Assume in the following that each event set in a history contains at most one event ...

A run r contains a *process chain* $\langle i_1, i_2, \ldots, i_k \rangle$ in $(r, m \ldots m')$ if there is a causal sequence of events $e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_k$ between times n to n' such that e_i is an event local to processor p_i.

Slide 25

Message Chain Theorems

Theorem: (Chandy and Misra) In asynchronous message passing systems $\mathcal{I}$,
(1) if $\mathcal{I},(r,n) \models \neg K_{i_k}\varphi$ and $\mathcal{I},(r,n') \models K_{i_1}K_{i_2}\dots K_{i_k}\varphi$ then there is a process chain $\langle i_k,\dots,i_1\rangle$ in r in the interval from n to n'.
(2) if $\mathcal{I},(r,n) \models K_{i_1}K_{i_2}\dots K_{i_k}\varphi$ and $\mathcal{I},(r,n') \models \neg K_{i_k}\varphi$
then there is a process chain $\langle i_1,\dots,i_k\rangle$ in r in the interval from n to n'.

Slide 26

Proof of the message chain theorem uses the following:

If $i_1\dots i_k$ is a sequence of agents, define $(r,m) \sim_{i_1\dots i_k} (r',m')$ inductively by

1. $k=1$ and $(r,m) \sim_{i_1} (r',m')$, or
2. $k>1$ and $(r,m) \sim_{i_1} (r'',m'')$ and $(r'',m'') \sim_{i_2\dots i_k} (r',m')$ for some point (r'',m'')

Lemma: Let $\mathcal{R}$ be an a.m.p. system, let $r \in \mathcal{R}$ and let $m < m'$. For all sequences of processes $i_1\dots i_k$, either $(r,m) \sim_{i_1\dots i_k} (r,m')$, or $i_1\dots i_k$ is a process chain in $(r,m\dots m')$.

Slide 27

Application: mutual exclusion protocols

Suppose that the code of a process is divided into a *critical section* and uncritical sections.

A *mutual exclusion* protocol gives a way for processes to communicate that ensures distinct processes are not in their critical sections simultaneously.

$\mathcal{I}$ is a system for mutual exclusion if

$$\mathcal{I} \models \bigwedge_{i\neq j} \neg(\texttt{critical}_i \land \texttt{critical}_j)$$

Slide 28

Note $\texttt{critical}_i$ depends only on i's local state

If $\mathcal{I}$ is an a.m.p. system for mutual exclusion and $m < m'$ and $i \neq j$ and $(\mathcal{I},r,m) \models \texttt{critical}_i$ and $(\mathcal{I},r,m') \models \texttt{critical}_j$ then

$(\mathcal{I},r,m) \models K_iK_j\neg\texttt{critical}_j$ and $(\mathcal{I},r,m') \models \neg K_j\neg\texttt{critical}_j$

So $\langle i,j\rangle$ is a process chain in $(r,m\dots m')$, and at least one message is sent in $(r,m\dots m')$.

Corollary: If processes $i_1,\dots,i_k$ are in their critical sections in that order, at least $k-1$ messages are sent.

lide 29

No gain or loss of common knowledge in a.m.p. systems

Theorem: Suppose $\mathcal{I}$ is an interpreted a.m.p. system and G is a group of processes with $|G| \geq 2$. Then for all formulas ϕ and times $m \geq 0$ we have $(\mathcal{I}, r, m) \models C_G\phi$ iff $(\mathcal{I}, r, 0) \models C_G\phi$.