A notation for programs

Case of
  if $\phi_1$ do action$_1$
  ...
  if $\phi_n$ do action$_n$
end case

means:
repeat forever: nondeterministically choose $i$ such that $\phi_i$ is true, do action$_i$
Knowledge-based programs

Knowledge-based programs for agent $i$ are programs in which the $\phi$ may talk about the knowledge of agent $i$.

E.g., After father says “at least one of you is muddy”, child $i$ behaves as if running the following knowledge-based program:

Case of
  
  if $K_i(\text{muddy}_i) \lor K_i(\neg \text{muddy}_i)$ do Say “Yes”
  
  if $\neg (K_i(\text{muddy}_i) \lor K_i(\neg \text{muddy}_i))$ do Say “No”
end case

A robotics example

A knowledge-based program:

Case of
  
  if $K_r(\text{posn} \in \text{Goal})$ do halt
end case

What is the semantics of such programs? (FHMV book version)

Joint actions

Let $ACT_e$ be a set of actions that the environment can perform.

For each agent $i$, let $ACT_i$ be a set of actions that agent $i$ can perform
(examples: say “yes”, send message $m$ to agent $j$, the “skip/do nothing” action $\Lambda$)

A joint action is a tuple $(a_e, a_1, \ldots, a_n)$ such that $a_e \in ACT_e$ and $a_i \in ACT_i$ for each $i = 1 \ldots n$.

Write $ACT$ for the set of joint actions

Let the set of global states be $G = L_e \times L_1 \times \ldots \times L_n$.

A transition function is a mapping $\tau : ACT \rightarrow (G \rightarrow G)$

if $a$ is a joint action and $s$ is a state, then $\tau(a)(s)$ is the next state when $a$ is performed in state $s$.

Typically, $\tau((\Lambda, \Lambda, \ldots, \Lambda))(s) = s$
Protocols
A protocol for agent $i$ (with local states $L_i$) is a mapping
\[ P_i : L_i \rightarrow \mathcal{P}(ACT_i) \setminus \{\emptyset\} \]
$P_i(l) = \{a_1, \ldots, a_n\}$ means that when in local state $l$, agent $i$’s next action is (nondeterministically) one of $a_1, \ldots, a_n$.
P_i is deterministic if $|P_i(l)| = 1$ for all $l \in L_i$.
A joint protocol is a tuple $(P_1, \ldots, P_n)$ where each $P_i$ is a protocol of agent $i = 1 \ldots n$.

Admissibility Conditions
We sometimes need to express constraints on the set of runs that cannot be captures using just initial states and protocols.
E.g. Every message sent is eventually delivered.
Represent these using a set $\Psi$ of runs, called an admissibility condition.
E.g. $\Psi$ is the set of runs in which all messages sent are eventually delivered.

Contexts
A context is a tuple $\gamma = (P_e, G_0, \tau, \Psi)$ where
1. $P_e : L_e \rightarrow ACT_e$ is a protocol for the environment
2. $G_0 \subseteq G$ is a set of initial global states,
3. $\tau : ACT \rightarrow (G \rightarrow G)$ is a transition function,
4. $\Psi$ is an admissibility condition on runs.

An interpreted context is a pair $(\gamma, \pi)$ where $\pi$ is an interpretation of a set of atomic propositions $Prop$ in the global states $G$ of $\gamma$.

A run $r$ is consistent with a joint protocol $P = (P_1, \ldots, P_n)$ in context $\gamma = (P_e, G_0, \tau, \Psi)$ if
1. $r(0) \in G_0$
2. for all $m \geq 0$ there exists $a_e \in P_e(r_e(m))$ and $a_i \in P_i(r_i(m))$ (for $i = 1 \ldots n$) such that $r(m+1) = \tau((a_e, a_1, \ldots, a_n))(r(m))$
3. $r \in \Psi$

Write $R^{\Psi}(P,\gamma)$ for the set of all runs that are consistent with protocol $P$ in context $\gamma$. 
Let \((\gamma, \pi)\) be an interpreted context and
\(\mathbb{P}_{g_i} = \text{Case of}
\begin{align*}
&\text{if } \phi_1 \text{ do } a_1 \\
&\vdots \\
&\text{if } \phi_n \text{ do } a_n
\end{align*}
end case
a standard program for agent \(i\) such that \(\pi\) is compatible with \(\mathbb{P}_{g_i}\).

Define the protocol \(\mathbb{P}^\pi_{g_i}\) by

\[\mathbb{P}^\pi_{g_i}(l) = \begin{cases} 
\{a_i \mid (\pi, l) \models \phi_i, i = 1 \ldots n\} & \text{if this set is not } \emptyset \\
\{\Lambda\} & \text{otherwise}
\end{cases}\]

For joint programs \(\mathbb{P} = (\mathbb{P}_1, \ldots, \mathbb{P}_n)\),
\[\mathbb{P}^\pi = (\mathbb{P}^\pi_1, \ldots, \mathbb{P}^\pi_n)\]

The interpreted system representing a joint program \(\mathbb{P}\) in the interpreted context \((\gamma, \pi)\) is the system
\[\mathcal{I}^{\text{rep}}(\mathbb{P}, \gamma, \pi) = (\mathcal{R}^{\text{rep}}(\mathbb{P}^\pi, \gamma), \pi)\]

Can we give a similar definition \(\mathbb{P}^\pi\) for knowledge-based programs?
A problem:
to determine the set of runs produced by executing a knowledge based program, we need to know which actions are enabled at each point \((r, m)\).
to decide whether an action \(a_i\) is enabled by a knowledge-based program at \((r, m)\), we need to evaluate the formula \(\phi_i\).
When \(\phi_i\) contains knowledge operators, this means looking at points \((r', m')\) such that \((r', m') \sim_i (r, m)\)
For that, we need to know the set of runs.
This is circular!

Resolution: provide a way of testing whether a given standard program/protocol implements a knowledge-based program.....

Given an interpreted system \(\mathcal{I} = (\mathcal{R}, \pi)\) and a joint knowledge-based program \(\mathbb{P} = (\mathbb{P}_1, \ldots, \mathbb{P}_n)\), define a joint protocol
\[\mathbb{P}^{\mathcal{I}} = (\mathbb{P}^{\mathcal{I}}_1, \ldots, \mathbb{P}^{\mathcal{I}}_n)\], as follows.
A knowledge-based formula $\phi$ is local to agent $i$ according to $\pi$ if it is either a proposition $p$ local to $i$ according to $\pi$, a formula of the form $K_i \phi$, or a boolean combination of these.

Let $\mathcal{I} = (\mathcal{R}, \pi)$ be an interpreted system. If $l$ is a local state of agent $i$ and $\phi$ is a formula local to $i$ according to $\pi$, define $(\mathcal{I}, l) \models \phi$ by:

1. $(\mathcal{I}, l) \models p$ if $(\pi, l) \models p$ (as defined above),
2. $(\mathcal{I}, l) \models K_i \phi$ if $(I, r, m) \models \phi$ for all points $(r, m)$ in $\mathcal{I}$ such that $r_i(m) = l$,
3. (Booleans as usual)

Let $\mathcal{I} = (\mathcal{R}, \pi)$ be an interpreted and

$\text{Pg}_i = \text{Case of}$

if $\phi_1$ do $a_1$

$
\vdots$

if $\phi_n$ do $a_n$

$\text{end case}$

a knowledge-based program for agent $i$ such that each $\phi_j$ is local to $i$ according to $\pi$.

Define the protocol $\text{Pg}_i^{\mathcal{I}}$ by

$$
\text{Pg}_i^{\mathcal{I}}(l) = \begin{cases} 
\{a_i \mid (\mathcal{I}, l) \models \phi_i, i = 1 \ldots n\} & \text{if this set is not } \emptyset \\
\{\Lambda\} & \text{otherwise}
\end{cases}
$$

A system $\mathcal{I}$ represents the knowledge-based program $\text{Pg}$ in an interpreted context $(\gamma, \pi)$ if $\pi$ is compatible with $\text{Pg}$ and $\mathcal{I} = \mathcal{I}^{\text{rep}}(\text{Pg}_{\mathcal{I}}, \gamma, \pi)$.

A standard protocol $P$ is an implementation of the knowledge-based program $\text{Pg}$ in an interpreted context $(\gamma, \pi)$ if $\mathcal{I}_P = \mathcal{I}^{\text{rep}}(P, \gamma, \pi)$ represents $\text{Pg}$ in $(\gamma, \pi)$.

Note this implies $\mathcal{I}_P = \mathcal{I}^{\text{rep}}(\text{Pg}_{\mathcal{I}_P}, \gamma, \pi)$.

A standard program $\text{Pg}_s$ is an implementation of the knowledge-based program $\text{Pg}_k$ in an interpreted context $(\gamma, \pi)$ if the protocol $\text{Pg}_s^{\mathcal{I}}$ is an implementation of $\text{Pg}_k$ in $(\gamma, \pi)$.

Specifications

Let $\phi$ be a formula of the language of knowledge and time

A protocol $P$ satisfies $\phi$ in context $(\gamma, \pi)$ if for all runs $r \in \mathcal{R}^{\text{rep}}(P, \gamma)$, we have

$$
((\mathcal{R}^{\text{rep}}(P, \gamma), r, 0) \models \phi
$$

A (knowledge-based) program $\text{Pg}$ satisfies $\phi$ in $(\gamma, \pi)$ if for all systems $\mathcal{I}$ representing $\text{Pg}$ in $(\gamma, \pi)$, for all runs $r$ of $\mathcal{I}$ we have

$$(\mathcal{I}, r, 0) \models \phi$$
Specification of Robot

Goal = \{2, 3, 4\}
Safety part: □(halted ⇒ posn ∈ Goal)
Liveness part: ◇halted

Implementations of the robot example

Let \( P^A \) be the deterministic protocol that always does \( A \) (never halt)
Let \( \gamma \) be a context in which
- the environment nondeterministically moves the position one step, or none
- the robot’s local state is the sensor reading
- the sensor reading is (nondeterministically) \( posn + x, x ∈ \{-1, 0, 1\} \)

Then
\[ T^{\text{rep}}(P^A, \gamma, \pi) \models K_r(posn ∈ \{2, 3, 4\}) \iff \text{sensor} = 3 \]

Let \( MP \) be the knowledge-based program
\[
\begin{align*}
\text{if } K_r(posn ∈ \{2, 3, 4\}) & \text{ do halt} \\
\end{align*}
\]
Let \( MP_s \) be the standard program
\[
\begin{align*}
\text{if } \text{sensor}=3 & \text{ do halt} \\
\end{align*}
\]
Then \( MP_s \) is an implementation of \( MP \) in \( (\gamma, \pi) \)

\( MP_s \) satisfies the specification □(halted ⇒ posn ∈ \{2, 3, 4\})
But NOT the liveness part: ◇halted

Let \( MP'_s \) be the standard program
\[
\begin{align*}
\text{if } \text{sensor} ∈ \{3, 4, 5\} & \text{ do halt} \\
\end{align*}
\]
Then \( MP'_s \) is also an implementation of \( MP \) in \( (\gamma, \pi) \)

\( MP'_s \) satisfies the safety specification □(halted ⇒ posn ∈ \{2, 3, 4\})
AND the liveness part: ◇halted
A KBP with NO implementations

There are NO implementations of the KBP

\[
\text{if } K_1(\neg \Diamond (\text{bit} = 1)) \quad \text{do} \quad \text{bit} := 1
\]

in a context where the bit is initially 0, agent 1’s local state is the value of bit, and the environment does nothing.

Nonexcluding Admissibility Conditions

The admissibility condition \( \Psi \) of a context \( \gamma = (P_e, G_0, \tau, \Psi) \) is nonexcluding if for every protocol \( P \) and all times \( m \), if \( r : [0, \ldots, m] \rightarrow G \) satisfies

1. \( r(0) \in G_0 \) and
2. for all \( k \in [0, \ldots, m - 1] \) and \( (a_e, a_1, \ldots, a_n) \in P_e(r(k)) \times P_1(r_1(k)) \times \cdots \times P_n(r_n(k)) \) we have \( r(k + 1) = \tau((a_e, a_1, \ldots, a_n))(r(k)) \),

then there exists a run \( r' \in R^{\text{rep}}(P, \gamma) \) such that \( r[0 \ldots m] = r'[0 \ldots m] \).

An interpreted system \( \mathcal{I} \) provides witnesses for \( K_i \phi \) if for every point \((r, m)\) of \( \mathcal{I} \), if \((\mathcal{I}, r, m) \models \neg K_i \phi \) then there exists \( m' \leq m \) such that \((r, m) \sim_i (r', m') \) and \((\mathcal{I}, r', m') \models \neg \phi \).

Lemma: Every synchronous system provides witnesses for every formula \( K_i \phi \).
An interpreted context \((\gamma, \pi)\) provides witnesses for a knowledge-based program \(P_g\) if every interpreted system \(I\) representing \(P_g\) in \((\gamma, \pi)\) provides witnesses for every formula \(K_i\phi\) that appears as a subformula of a test in \(P_g\).

**Theorem:** Let \(P_g\) be a knowledge-based program in which tests do not involve temporal operators, let \(\gamma\) be a nonexcluding context, and assume that the context \((\gamma, \pi)\) provides witnesses for \(P_g\). Then there is a unique interpreted system representing \(P_g\) in \((\gamma, \pi)\).

(There may still be many protocols/programs implementing \(P_g\), but these differ only on unreachable states and syntactic details.)