Dynamic Epistemic Logic

Chapter 2 - Multi agent S5
Axiomatisation and Common knowledge

2.3 Axiomatisation

1- Semantic derivation of valid formulas via Kripke models.
2- Syntactic derivation of valid formulas via axioms.

Modal logic K:
- all instantiations of propositional tautologies (Prop)
- \( K_a(\phi \rightarrow \psi) \rightarrow (K_a\phi \rightarrow K_a\psi) \) (K)
- From \( \phi \land \phi \rightarrow \psi \), we can infer \( \psi \) (MP, modus ponens)
- From \( \phi \), we can infer \( K_a\phi \) (Nec, necessitation)

**Definition 1** Derivation

Let \( \mathcal{X} \) be an arbitrary axiomatisation with axioms \( \Lambda_{x_1}, \ldots, \Lambda_{x_n} \) and rules \( R_{u_1}, \ldots, R_{u_m} \), where each rule is of the form "From \( \phi_1, \ldots, \phi_l \), infer \( \phi_j \)." Then a derivation of a formula \( \phi \) with \( \mathcal{X} \) is a finite sequence \( \phi_1, \ldots, \phi_m \) of formulas such that:

1) \( \phi_m = \phi \) and
2) every \( \phi_i \) in the sequence is:
   a) either an instance of one of the axioms
   b) or else the result of the application of one of the rules to formulas in the sequence that appear before \( \phi_i \)

If there is a derivation for \( \phi \) in \( \mathcal{X} \), then we write \( \vdash_{\mathcal{X}} \phi \) or \( \vdash \phi \) if \( \mathcal{X} \) is clear.

We say that \( \phi \) is a theorem of \( \mathcal{X} \).

Logic K is only (arbitrary) Kripke models, including models where \( R_i \) not necessarily reflect knowledge. E.g model \( \mathcal{M} \)

\[
\begin{array}{c}
\text{w}_1 : p \\
\text{a} \\
\text{w}_2 : \neg p
\end{array}
\]

\( (\mathcal{M}, w_1) \models p \) but, \( (\mathcal{M}, w_2) \) models \( K_a \neg p \)
We would like a logic where something like \( \neg(p \land K_a \neg p) \) is a theorem. Semantically, we solved this by requiring epistemic models to have reflexive accessibility relations (among other requirements). Syntactically, add axiom \( K_a \phi \rightarrow \phi \).

**Additional axioms for S5:**

\[
K_a \phi \rightarrow K_a K_a \phi \quad (4, \text{positive introspection})
\]

\[
\neg K_a \phi \rightarrow K_a \neg K_a \phi \quad (5, \text{negative introspection})
\]

**Theorem 1** Axiom system \( K \) is sound and complete w.r.t. the class \( \mathcal{K} \) of all Kripke models, i.e. for every formula \( \phi \) in \( L_K \), we have that \( \vdash K \phi \iff K \models \phi \).

Similarly, \( \vdash S_5 \phi \iff S_5 \models \phi \). ("you can derive \( \phi \) in \( S_5 \) iff \( \phi \) is valid in all epistemic Kripke models")

### 2.4 Common knowledge

Group notions of knowledge:

Recall \( E_B \phi \). \( E_B \) satisfies axiom T, but not positive introspection.

\( E_B \phi \rightarrow E_B E_B \phi \) is not valid. E.g if agents a and b are both (separately) told that \( p \) is true, \( E_a b p \) is true but not \( E_a b E_a b p \).

So, how to model that everybody knows that everybody knows that... that \( p \)?

The common knowledge operator!

For \( B \subseteq A \), \( C_B \phi \equiv \bigwedge_{n=0} E^n_B \phi \), where \( E_B \phi = E_B E_B ... E_B \phi \).

**Definition 2** By language \( L_{KC} \), we refer to the language defined just like \( L_K \), but with the additional \( C \) modality. For \( a \in A, B \subseteq A, p \in P \), we define:

\[
\phi := \phi \land \phi \land ... \land K_a \phi \land C_B \phi
\]

**Semantics:** As before, using (epistemic) Kripke models.

**Definition 3** Let \( M = \langle S, R, V \rangle \) be a Kripke models with agents \( A \) and \( B \subseteq A \). Then \( R_{E_B} = \bigvee_{b \in B} R_b \).

The transitive closure of a relation \( R \) is the smallest relation \( R^+ \) s.t. :

1. \( R \subseteq R^+ \)

2. \( \forall x, y, z \text{ if } (x, y) \in R^+ \text{ and } (y, z) \in R^+ \text{ then also } (x, z) \in R^+ \)

If additionally, \( (x, x) \in R^+ \forall x \), then \( R^+ \) is the reflexive-transitive closure of \( R \).

**Definition 4** Let \( P \) be a set of atomic propositions, \( A \) a set of agents and \( M = \langle S, R, V \rangle \) an epistemic model and \( B \subseteq A \). Then the truth of an \( L_{K,C} \) formula \( \phi \) in \( \langle M, s \rangle \) is defined as for \( L_K \), with an additional clause for common knowledge.

\( \langle M, s \rangle \models C_B \phi \iff \langle M, t \rangle \models \phi \forall t \in S \text{ with } (s, t) \in E^\ast (C_B) = R^C_B \).
Example 1 $\mathcal{M}, w \models C_{ab} p$

$\mathcal{M}, w \not\models C_{abc} p$

![Figure 2: Example 1](image)

Additional axioms for common knowledge:

$C_B (\phi \rightarrow \psi) \rightarrow (C_B \phi \rightarrow C_B \psi)$ (Dist)

$C_B \phi \rightarrow (\phi \land E_B C_B \phi)$ (Mix)

$C_B (\phi \rightarrow E_B \phi) \rightarrow (\phi \rightarrow C_B \phi)$ (Ind)

From $\phi$, infer $C_B \phi$ (Nec)

Together with S5 axioms and rules: sound and complete w.r.t. epistemic models with common knowledge.

2.5 Model checking

Local MC for $L_K C$ formulas: Given a finite Kripke model $\mathcal{M} = (S, R, V)$, an $L_K C$ formulas $\phi$ and a state $s$, determine whether $s$ satisfies $\phi$:

you only care about state $s$. The rest of $S$ may be given only implicitly.

Global MC for $L_K C$ formulas: Given a finite Kripke model $\mathcal{M}_K C$, an $L_K C$ formula $\phi$, determine the set of states where $\phi$ is satisfied.

We care about all states.

Especially easy if $S$ is given explicitly.

Algorithmically often done semantically.

Idea: For all subformulas $\psi$ of $\phi$, determine the sets of states where $\psi$ is true, inductively from small to large subformulas.

Definition 5 Subformulas

Let $\phi$ be a formula in the $L_K C$ language. Then the set of subf($\phi$) of subformulas is
defined recursively as follows:

\[ \text{subf}(p) = p \text{ for atomic propositions } p \in P \]
\[ \text{subf}(\neg \phi) = \{ \neg \phi \} \cup \text{subf}(\phi) \]
\[ \text{subf}(\phi \lor \psi) = \{ \phi \lor \psi \} \cup \text{subf}(\phi) \cup \text{subf}(\psi) \]
\[ \text{subf}(K_a \phi) = \{ K_a \phi \} \cup \text{subf}(\phi) \]
\[ \text{subf}(C_B \phi) = \{ C_B \phi \} \cup \text{subf}(\phi) \]

If \( \psi \in \text{subf}(\phi) \setminus \{ \phi \} \) then \( \psi \) is called a proper subformula of \( \phi \).

**Definition 6** Let \( a \) be an agent and \( S' \subseteq S \). Then the strong preimage of \( S \); w.r.t \( a \) is:

\[ \text{spreimg}_a(S) = \{ s \in S | \text{for } s' \in S \text{ with } (s, s') \in R_a: s' \in S' \} \]

**Notation:**
Let \( \llbracket \phi \rrbracket = \{ s \in S | s \models \phi \} \) be the set of states where \( \phi \) is true.

**MC algorithm**
Let \( M = (S, R, V) \) be an (epistemic) Kripke model and \( \phi \in L_KC \) a formula. Let \( \phi_1, \ldots, \phi_n \) be the subformulas of \( \phi \) ordered from small to large. Then:
Algorithm 1 Model checking

switch $\phi_i$ do
    case $p$
        $\llbracket p \rrbracket := V(p)$
    case $\neg \phi'$
        $\llbracket \phi_i \rrbracket := S \setminus \llbracket \phi' \rrbracket$
    case $\phi' \lor \phi''$
        $\llbracket \phi_i \rrbracket := \llbracket \phi' \rrbracket \cup \llbracket \phi'' \rrbracket$
    case $\phi' \land \phi''$
        $\llbracket \phi_i \rrbracket := \llbracket \phi' \rrbracket \cap \llbracket \phi'' \rrbracket$
    case $K_a \phi'$
        $\llbracket \phi_i \rrbracket := \text{spreimg}_a(\llbracket \phi' \rrbracket)$
    case $C_a \phi'$
        Let $S_1 = \llbracket \phi' \rrbracket$
        Let $S_2 = S_1 \cap \bigcap_{b \in B} \text{spreimg}(S_1)$
        $j := 1$
        while $S_j \neq S_{j+1}$ do
            $j := j + 1$
            $S_{j+1} := S_j \cap \bigcap_{b \in B} \text{spreimg}(S_j)$
        end while
        Then $\llbracket \phi_i \rrbracket := S_{j+1}$
end switch
Intuition behind the $C_B \phi'$ case:

$$[[E_B \Phi']] = [[[E_B^{+1} \Phi'] = [C_B \phi']]]$$

Example 2 $[[\neg B_b (K_a p \land q)]]$?

$[p] = \{S_1, S_2, S_3, S_5, S_6\}$

$[q] = \{S_2, S_3, S_4, S_5, S_6\}$

$[K_a p] = \{S_1, S_2, S_3\}$

$[K_a \land q] = \{S_2, S_3\}$

$[K_b (K_a p \land q)]] = \emptyset$

$[[\neg B_b (K_a p \land q)]] = \{S_1, S_2, S_3, S_4, S_5, S_6\}$

Figure 3: Example 2
Example 3 $\llbracket C_{ab}p \rrbracket$?

$\llbracket p \rrbracket = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\} = S_1$

$S_2 = S_1 \cap (\text{spreimg}_a(S_1) \cap \text{spreimg}_a(S_2))$

$= S_1 \cap (S_1 \cap \{s_1, \ldots, s_6\})$

$= \{s_1, \ldots, s_6\}$

$S_3 = \ldots = \{s_1, \ldots, s_5\}$

$S_4 = S_3 = \llbracket C_{ab}p \rrbracket = \{s_1, \ldots, s_5\}$