Formal Specification of Software

Modal Logic

Bernhard Beckert

UNIVERSITÄT KOBLENZ-LANDAU
Modal Logic

In classical logic, it is only important whether a formula is true.

In modal logic, it is also important in which

- way
- mode
- state

a formula is true.

A formula (a proposition) is

- necessarily / possibly true
- true today / tomorrow
- believed / known
- true before / after an action / the execution of a program
Propositional Modal Logic: Formulas

- The propositional variables \( p \in \text{Var} \) are modal formulas.

- If \( A, B \) are modal formulas, then

\[
\neg A \quad (A \land B) \quad (A \lor B) \quad (A \to B) \quad (A \leftrightarrow B)
\]

\( \Box A \) (read “box \( A \)”, “necessarily \( A \)”)  
\( \Diamond A \) (read “diamond \( A \)”, “possibly \( A \)”)  

are modal formulas.
Informal Interpretations of □

□F means

- F is necessarily true
- F is always true (in future states/words)
- an agent a believes F
- an agent a knows F
- F is true after all possible executions of a program p

Notation

If necessary write

□aF □pF [a]F [p]F

instead of □F
### Informal Interpretations of $\Diamond$

<table>
<thead>
<tr>
<th>$\Box F$</th>
<th>$\Diamond F$ (the same as $\neg \Box \neg F$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$ is necessarily true</td>
<td>$F$ is possibly true</td>
</tr>
<tr>
<td>$F$ is always true</td>
<td>$F$ at least once true</td>
</tr>
<tr>
<td>agent $a$ believes $F$</td>
<td>$F$ is consistent with $a$’s beliefs</td>
</tr>
<tr>
<td>agent $a$ knows $F$</td>
<td>$a$ does not know $\neg F$</td>
</tr>
<tr>
<td>$F$ is true after all possible executions of program $p$</td>
<td>$F$ is true after at least one possible execution of program $p$</td>
</tr>
</tbody>
</table>
Kripke Structures

Given: a propositional signature Var

Definition

A Kripke structure

\[ \mathcal{K} = (S, R, I) \]

consists of

- a non-empty set \( S \) (of worlds / states)
- an accessibility relation \( R \subseteq S \times S \)
- an interpretation \( I : \text{Var} \times S \rightarrow \{ \text{true}, \text{false} \} \)
**Kripke Structures: Example**

**accessibility relation**

**set of states**

---

**Interpretation** $I$
Modal Logic: Semantics

Given: Kripke structure $\mathcal{K} = (S, R, I)$

Valuation

\[ val_{\mathcal{K}}(p)(s) = I(p)(s) \quad \text{for} \quad p \in \text{Var} \]

$val_{\mathcal{K}}$ defined for propositional operators in the same way as $val_I$

\[ val_{\mathcal{K}}(\Box A)(s) = \begin{cases} 
\text{true} & \text{if} \; val_{\mathcal{K}}(A)(s') = \text{true} \text{ for all } s' \in S \text{ with } sR \]
\[ val_{\mathcal{K}}(\Diamond A)(s) = \begin{cases} 
\text{true} & \text{if} \; val_{\mathcal{K}}(A)(s') = \text{true} \text{ for at least one } s' \in S \text{ with } sR \]
\]
Saul Aaron Kripke

Born 1940 in Omaha (US)


Studied at: Harvard, Princeton, Oxford and Rockefeller University

Professor at Princeton University since 1998
Emeritus at Princeton University
Modal Logic: Example for Evaluation

\[(\mathcal{K}, A) \models P \quad (\mathcal{K}, B) \models \neg P \quad (\mathcal{K}, C) \models P \quad (\mathcal{K}, D) \models \neg P\]

\[(\mathcal{K}, A) \models \Box \neg P \quad (\mathcal{K}, B) \models \Box P \quad (\mathcal{K}, C) \models \Box P \quad (\mathcal{K}, D) \models \Box P\]

\[(\mathcal{K}, A) \models \Box \Box P \quad (\mathcal{K}, B) \models \Box \Box P \quad (\mathcal{K}, C) \models \Box \Box \neg P\]
## Modal Logic: Valid Formulas

### Valid

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>□(P → Q) → (□P → □Q)</td>
<td></td>
</tr>
<tr>
<td>□P ∧ □(P → Q)) → □Q</td>
<td></td>
</tr>
<tr>
<td>□P ∨ □Q → □(P ∨ Q)</td>
<td></td>
</tr>
<tr>
<td>□P ∧ □Q ↔ □(P ∧ Q)</td>
<td></td>
</tr>
<tr>
<td>□P ↔ ¬◊¬P</td>
<td></td>
</tr>
<tr>
<td>◊(P ∨ Q) ↔ (◊P ∨ ◊Q)</td>
<td></td>
</tr>
<tr>
<td>◊(P ∧ Q) → (◊P ∧ ◊Q)</td>
<td></td>
</tr>
</tbody>
</table>

### Not valid:

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>□(P ∨ Q) → (□P ∨ □Q)</td>
<td></td>
</tr>
<tr>
<td>◊P ∧ ◊Q → ◊(P ∧ Q)</td>
<td></td>
</tr>
</tbody>
</table>
Not Valid: $\square(P \lor Q) \rightarrow (\square P \lor \square Q)$

$\square(P \lor Q)$
$\neg \square P$
$\neg \square Q$
$\neg(\square P \lor \square Q)$

$\square(P \lor Q) \rightarrow (\square P \lor \square Q)$ **not true in state** $s_1$
Formulas Characterising Properties of $R$

<table>
<thead>
<tr>
<th>Formula</th>
<th>Property of $R$</th>
<th>Formula</th>
<th>Property of $R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\square p \to p$</td>
<td>reflexive</td>
<td>$\square p \to \square \square p$</td>
<td>transitive</td>
</tr>
<tr>
<td>$p \to \Diamond p$</td>
<td>reflexive</td>
<td>$p \to \square \Diamond p$</td>
<td>symmetrical</td>
</tr>
<tr>
<td>$\square \square p \to \square p$</td>
<td>reflexive</td>
<td>$\square \square p \leftrightarrow \square p$</td>
<td>reflexive, transitive</td>
</tr>
<tr>
<td>$\Diamond \Diamond p \to \Diamond p$</td>
<td>reflexive</td>
<td>$\Diamond \Diamond p \leftrightarrow \Diamond p$</td>
<td>reflexive, transitive</td>
</tr>
<tr>
<td>$\square p \to \Diamond \Diamond p$</td>
<td>reflexive</td>
<td>$\Diamond \square p \leftrightarrow \square p$</td>
<td>equivalence relation</td>
</tr>
<tr>
<td>$\Diamond \Diamond \Diamond p \to \Diamond p$</td>
<td>reflexive</td>
<td>$\square \Diamond \Diamond p \leftrightarrow \Diamond p$</td>
<td>equivalence relation</td>
</tr>
</tbody>
</table>
Modal Logic: Valid Formulas

<table>
<thead>
<tr>
<th>□F</th>
<th>□F</th>
<th>□□F</th>
<th>◻F</th>
<th>(□(F → □G) ∧ □G)</th>
<th>◻true</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F is necessarily true</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>agent a knows F</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>agent a believes F</strong></td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>F holds after executing program p</strong></td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>