What is modal logic?

- Variety of different systems
- Difficult to give a definition which fits all of them
- Superficial answer: a logic which has a modality or several modalities in it

What is a modality?

- Modality is a connective which takes a formula (or formulas) and produces a new formula with a new meaning.
- Just as ¬ is a connective which takes a formula φ and produces a new formula ¬φ, or → takes φ and ψ and produces a formula φ→ψ.
- The only difference is that in classical logic, the truth value of ¬φ is uniquely determined by the value of φ, and the value of φ→ψ is a function of the values of φ and ψ.
- Modalities are not truth-functional.

Examples (unary modalities)

- □φ: “it is necessary that φ”
- ◊φ: “it is possible that φ”
- Gφ: “always in the future, φ will be true”
- Fφ: “at some point in the future, φ will be true”
- Pφ: “at some point in the past, φ was true”
- K_iφ: “agent i knows that φ”
- B_iφ: “agent i believes that φ”
- [prog]φ: “after any execution of the program prog, the state satisfies property φ”
- <prog>φ: “there is an execution of the program prog, which results in a state satisfying property φ”

Binary modalities

- φ → ψ (intuitionistic implication)
- U(φ,ψ): “until φ becomes true, ψ holds”

The plan (for today)

- Define basic modal logic
- Describe various systems of modal logic
- Explain how they are used
- Try to explain what they have in common

After the break:

- Completeness and decidability of basic modal logic
The plan (for tomorrow)

- Bisimulation
- Processes
- Propositional dynamic logic (PDL)
- Computation tree logic (CTL*)
- Model checking.

Basic modal logic: the language

Alphabet:
- A set of propositional variables Prop={p₁, p₂, ...}
- Boolean connectives ¬ and → (∧, ∨, and ↔ are definable)
- (Unary) modality □ (◊ is definable)

Well-formed formula φ:
φ := p ∈ Prop | ¬φ | φ₁ → φ₂ | !φ₁

Just for completeness...

φ₁ ∨ φ₂ := ¬¬φ₁ → φ₂
φ₁ ∧ φ₂ := ¬(φ₁ → ¬φ₂)
φ₁ ↔ φ₂ := (φ₁ → φ₂) ∧ (φ₂ → φ₁)
◊φ := ¬□¬φ

Basic modal logic: models

Kripke structures (possible worlds structures) are models of basic modal logic.
A Kripke structure is a triple M = (W,R,V) where
- W is a non-empty set (possible Worlds)
- R ⊆ W x W is an accessibility Relation
- V: (Prop x W) → {true, false} is a Valuation function.

This is just a graph (W,R) with a function V which tells which propositional variables are true at which vertices.

Example

Example
Basic modal logic: meaning of formulas

Given $M = (W, R, V)$ and $w \in W$, we define what does it mean for a formula to be true (satisfied) in a world $w$ of a model $M$:

$M, w \models p$ iff $V(p, w) = \text{true}$;
$M, w \models \neg \phi$ iff $M, w \not\models \phi$;
$M, w \models \theta \rightarrow \psi$ iff either $M, w \not\models \theta$ or $M, w \models \psi$;
$M, w \models \Box \phi$ iff for all $v$ accessible from $w$ (for all $v$ such that $R(w, v)$), $M, v \models \phi$.

Example

$w_1 = \{p\}$
$w_2 = \{p, q\}$
$w_3 = \{p\}$
$w_4 = \{q\}$
$w_5 = \{q\}$

$M, w_1 \models \Box q$
$M, w_1 \models \neg \Box p$
$M, w_1 \models \neg \Box \neg p$
$M, w_1 \models \Diamond p$

Validity (and satisfiability)

- A formula $\phi$ is true in a model $M$ if it is satisfied in all of $M$'s worlds.
- A formula $\phi$ is valid if it is true in all models.
- A formula is satisfiable if its negation is not valid (if it is satisfied in at least one world of one model).

Examples

- $\Box p \rightarrow \Box p$ is valid (just an example of a propositional tautology).
- $\Box (p \rightarrow p)$ is valid (because $p \rightarrow p$ is true in all accessible worlds, wherever you are).
- $\Box p \rightarrow p$ is not valid (the set $\{\Box p, \neg \Box p\}$ is satisfiable in some worlds).

Example

$w_1 = \{p\}$
$w_2 = \{p, q\}$
$w_3 = \{p\}$
$w_4 = \{q\}$
$w_5 = \{q\}$

$M, w_1 \models \Box p \land \neg p$
Aside

- To make \( \Box p \rightarrow p \) valid, need to require that \( R \) is reflexive.
- Then if \( M,w \models p \), from \( R(w,w) \) also \( M,w \models \Box p \).
- Other correspondencies:
  - \( \Box p \rightarrow \Box \Box p \) corresponds to transitivity of \( R \) (easier to see in \( \Diamond \) form, \( \Diamond \Diamond p \rightarrow \Diamond p \); if you can get somewhere in two steps, you can get there is one step).
  - \( \Box p \rightarrow \Diamond p \) corresponds to seriality of \( R \) (for every world there is an accessible world)
  - \( p \rightarrow \Box \Diamond p \) corresponds to symmetry
  - \( \Diamond p \rightarrow \Box \Diamond p \) corresponds to \( R \) being euclidean

Euclidean relation

- \( \Diamond p \rightarrow \Diamond \Diamond p \)
- \( \forall x \forall y (R(x,y) \land R(x,z) \rightarrow R(y,z)) \)

What can you express in basic modal logic?

Useful intuition:
- possible worlds are states in a computation,
- \( R \) is the transition relation,
- \( V \) tells us which properties are true of which state.

Let’s see what we can express in basic modal logic - this will also allow us to motivate more complicated systems.

Running example

Suppose we have two processes/agents A and B.
- Each has a local boolean variable (A has a, B has b).
- All they are doing is: flip the value of their variable; sleep for a bit; then flip the value back again.
- We assume that their actions are interleaved (not executed simultaneously).

Running example

What can we say about this system in basic modal logic?

- \( \Diamond a \land \Diamond a \)
- \( \Diamond b \land \Diamond b \)
- \( a \land b \rightarrow \Diamond (\neg a \lor \neg b) \)
- \( a \land b \rightarrow \Box (a \land b) \)

Basically, which states we can reach and in how many steps.
What we cannot say

- Can’t say something is “reachable” in principle: have to say “reachable in n steps”.
- Can’t say which action (by which process) will get us to which state.
- Can’t say “there is an execution trace starting at w₁ where b is always false”.
- Can’t say what agent A “knows” about agent B (this would make more sense if A and B were trying to communicate and make sure messages were received etc.)

Adding actions

![Diagram showing actions a and b with implications and negations.]

Propositional dynamic logic

- Instead of one accessibility relation $R$, structures have many accessibility relations. Each $R_i$ corresponds to some statement (atomic action) $i$, for example $a = !a$.
- Corresponding modalities are $[i]$ and $<i>$, for example $[a = !a] \phi$ (always after executing $a = !a$, $\phi$ holds) and $<a = !a> \phi$ (it is possible by executing $a = !a$ to arrive in a state where $\phi$ holds).
- $\neg a \rightarrow <b = !b> a$ and $a \rightarrow <b = !b> a$
- $a \wedge b \rightarrow ([a = !a] \cup [b = !b]) (\neg a \lor \neg b):$ if $a$ and $b$ hold, then after executing $a = !a$ or $b = !b$, either $a$ is false or $b$ is false.
- $a \rightarrow ((b = !b)^* a):$ if $a$ holds, then after 0 or finitely many iterations of $b = !b$, $a$ still holds.

Propositional dynamic logic contd.

- In addition to atomic statements, we can use composition $;$, union $\cup$ and iteration $^*$ to form new program modalities.
- For example,
  - $a \rightarrow [(a = !a);(a = !a)] a$: if $a$ holds, then after executing $a = !a$ twice, $a$ holds again.
  - $a \wedge b \rightarrow [(a = !a) \cup (b = !b)] (\neg a \lor \neg b):$ if $a$ and $b$ hold, then after executing $a = !a$ or $b = !b$, either $a$ is false or $b$ is false.
  - $a \rightarrow [(b = !b)^* a]:$ if $a$ holds, then after 0 or finitely many iterations of $b = !b$, $a$ still holds.

Multimodal logic

- In general, in multimodal logic (including PDL) each accessibility relation $R_i$ gives rise to modalities $[i]$ and $<i>$ with the following truth definitions:
  - $M,w \models [i] \phi$ iff for all $v$ with $R_i(w,v)$, $M,v \models \phi$
  - $M,w \models <i> \phi$ iff for some $v$ with $R_i(w,v)$, $M,v \models \phi$
- As before, $<i> \phi$ is definable as $\neg [i] \neg \phi$.
- Action $i$ is deterministic: $<i> \phi \rightarrow [i] \phi$

Talking about traces

- However, PDL cannot express the fact for example that there is a particular execution trace where process B does not have a chance to run at all.
Unwinding a state diagram

Computation tree logic CTL*

- Talks about computation trees as above
- Choose initial state and unwind a Kripke structure into a tree
- Can quantify over paths and say things like
  - $\text{AG} \phi$: on all paths starting here in all states $\phi$ holds
  - $\text{EG} \phi$: there is a path starting here along which $\phi$ holds
  - $\text{AF} \phi$: on all paths starting here $\phi$ holds at least once
  - $\text{EF} \phi$: there is a path starting here where $\phi$ holds at least once
  - $\text{AGF} \phi$: on all paths starting here there is always a state ahead where $\phi$ holds ($\phi$ holds infinitely often).

Reasoning about knowledge

- What if we want to verify a communication protocol...
- Agent A sends agent B a message and B sends A an acknowledgement
- Now A knows that B has received the message, but A does not know whether B knows that A knows that B has received the message ... etc.
- Classical examples involve muddy children/wise men and Byzantine generals.

Basic epistemic logic

- Instead of $\Box$ we have modal operator $K$ for Knows. Truth definition the same as for $\Box$.
- Usually we consider several agents, so we have a multimodal logic: several operators $K_i$, each interpreted using accessibility relation $R_i$.
- Each accessibility relation $R_i$ is assumed to be an equivalence relation (reflexive, transitive and symmetric).
Intuition

- Suppose we have agents A and B with local states $\text{state}(A)$ and $\text{state}(B)$. The global state of the system (possible world $w$) consists of $w_A = \text{state}(A)$ at $w$, $w_B = \text{state}(B)$ at $w$, and perhaps some more variables.
- Then two global states $w$ and $v$ are connected by $R_A$ if $w_A = v_A$.
- $M,w \models K_A \phi$ if in all states $v$ where $v_A = w_A$, $M,v \models \phi$.
- Somehow A manages to correlate its state with $\phi$. It only goes into state $w_A$ when $\phi$ is true and never when $\phi$ is false.

Epistemic accessibility relations

- $\neg a \quad \neg b$
- $a \quad \neg b$
- $\neg a \quad \neg b$
- $\neg a \quad \neg b$
- $w_1 \quad w_2 \quad w_3 \quad w_4$
- $\text{state}(A) = \{a\}$
- $\neg a \quad \neg b$
- $\neg a \quad \neg b$
- $\neg a \quad \neg b$
- $\text{state}(B) = \{b\}$

What do these logics have in common

- All these logics talk about graphs.
- They talk about them from a 'local' point of view: what can we see from a given point? Quantifiers (for all... exists...) are restricted by edge relation or path; we quantify not over all points in the structure, but over ones accessible from a given point.
- On technical level, unlike say first order logic, all those logics are decidable. They can express fewer things but this means that they are easier to reason with.