Modal Logic

Midlands Graduate School in Foundations of CS, April 2003

Natasha Alechina
School of Computer Science & IT
University of Nottingham
nza@cs.nott.ac.uk

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?

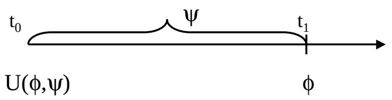
- Modality is a connective which takes a formula (or formulas) and produces a new formula with a new meaning.
- Just as $\neg$ is a connective which takes a formula ϕ and produces a new formula $\neg\phi$, or $\rightarrow$ takes ϕ and ψ and produces a formula $\phi\rightarrow\psi$.
- The only difference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

Examples (unary modalities)

- $\Box\phi$: "it is necessary that ϕ"
- $\Diamond\phi$: "it is possible that ϕ"
- $G\phi$: "always in the future, ϕ will be true"
- $F\phi$: "at some point in the future, ϕ will be true"
- $P\phi$: "at some point in the past, ϕ was true"
- $K_i\phi$: "agent i knows that ϕ"
- $B_i\phi$: "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

- $\phi\rightarrow\psi$ (intuitionistic implication)
- $U(\phi,\psi)$: "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={p_1, p_2 ...}
- Boolean connectives $\neg$ and $\rightarrow$ ($\wedge$, $\vee$, and $\leftrightarrow$ are definable)
- (Unary) modality $\Box$ ($\Diamond$ is definable)

Well-formed formula ϕ :

$\phi := p \in Prop \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \Box\phi$

Just for completeness...

- $\phi_1 \vee \phi_2 \ := \neg \phi_1 \rightarrow \phi_2$
- $\phi_1 \wedge \phi_2 \ := \neg(\phi_1 \rightarrow \neg \phi_2)$
- $\phi_1 \leftrightarrow \phi_2 \ := (\phi_1 \rightarrow \phi_2) \wedge (\phi_2 \rightarrow \phi_1)$
- $\Diamond\phi \ := \neg \ \Box\neg\phi$

Basic modal logic:models

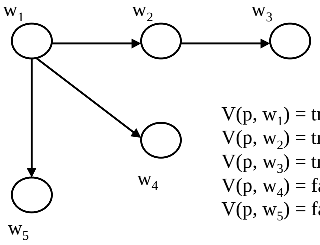
Kripke structures (possible worlds structures) are models of basic modal logic.

A Kripke structure is a triple M = (W,R,V) where

- W is a non-empty set (possible Worlds)
- R $\subseteq$ W x W is an accessibility Relation
- V: (Prop x W) $\rightarrow$ {true, false} is a Valuation function.

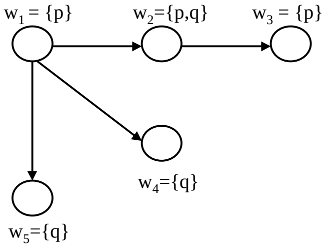
This is just a graph (W,R) with a function V which tells which propositional variables are true at which vertices.

Example



V(p, w_1) = true, V(q, w_1) = false
V(p, w_2) = true, V(q, w_2) = true
V(p, w_3) = true, V(q, w_3) = false
V(p, w_4) = false, V(q, w_4) = true
V(p, w_5) = false, V(q, w_5) = true

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:

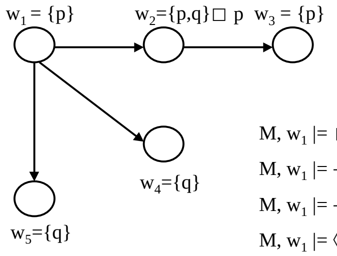
M, w |= p iff V(p,w) = true;

M,w |= $\neg\phi$ iff M,w $|\neq \phi$;

M,w |= $\phi \rightarrow \psi$ iff either M,w $|\neq \phi$ or M,w |= ψ;

M,w |= $\Box\phi$ iff for all v accessible from w (for all v such that R(w,v)), M,v |= ϕ

Example



M, w_1 |= $\Box q$

M, w_1 |= $\neg\Box p$

M, w_1 |= $\neg\Box\neg p$

M, w_1 |= $\Diamond p$

M, w_1 |= $\Diamond \Box p$

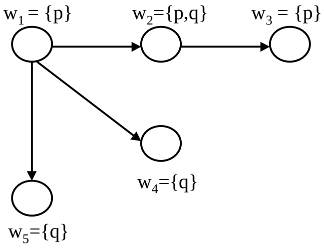
Validity (and satisfiability)

- A formula ϕ is true in a model M if it is satisfied in all of M's worlds
- A formula ϕ 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).

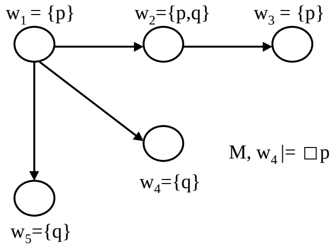
Examples

- $\Box p \rightarrow \Box p$ 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 $\{\Box p, \neg p\}$ is satisfiable in some worlds).

Example



Example



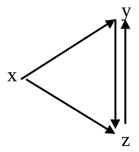
M, w_4 |= $\Box p \wedge \neg p$

Aside

- To make $\Box p \rightarrow p$ valid, need to require that R is reflexive.
- Then if $M,w \not\models p$, from $R(w,w)$ also $M,w \not\models \Box p$.
- 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

- $\Diamond p \rightarrow \Box\Diamond p$
- $\forall x \forall y \forall z\, (R(x,y) \wedge R(x,z) \rightarrow R(y,z))$



What can you express in basic modal logic?

Useful intuition:

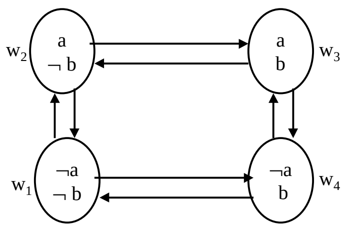
- possible worlds are states in a computation,
- R is the transition relation,
- V tells us which properties are true of which state.

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?

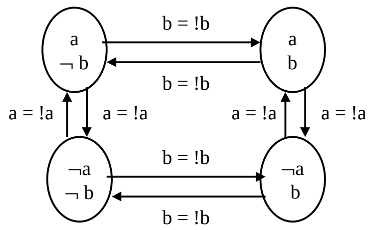
- $\Diamond\neg a \wedge \Diamond a$
- $\Diamond\neg b \wedge \Diamond b$
- $a \wedge b \rightarrow \Box(\neg a \vee \neg b)$
- $a \wedge b \rightarrow \Diamond\Diamond\,(a \wedge b)$

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 w_1 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



Propositional dynamic logic

- Instead of one accessibility relation R, structures have many acessibility 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] ϕ (always after executing a=!a, ϕ holds) and <a=!a>ϕ (it is possible by executing a=!a to arrive in a state where ϕ holds).
  - $a \rightarrow \langle b = !b \rangle\, a$ and $a \rightarrow [b = !b]\, a$
  - $a \rightarrow [a = !a] \neg a$
  - $a \wedge b \rightarrow [a = !a](\neg a \wedge b)$
- So we can axiomatise pre- and post-conditions of actions.

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 |= [i] ϕ iff for all v with $R_i(w,v)$, M,v |= ϕ

M, w |= <i> ϕ iff for some v with $R_i(w,v)$, M,v |= ϕ

- As before, <i> ϕ is definable as $\neg$ [i] $\neg$ ϕ .

- Action i is deterministic: $\langle i \rangle \phi \rightarrow [i] \phi$

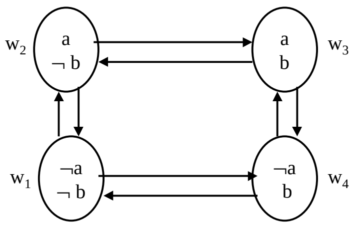
Propositional dynamic logic contd.

- 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
  - $a \wedge b \rightarrow [(a = !a) \cup (b = !b)] (\neg a \vee \neg b)$: 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

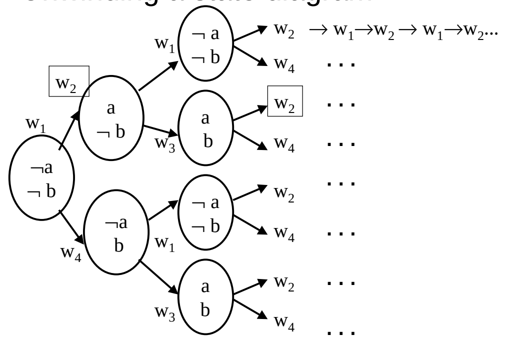
Talking about traces

- 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



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
  - AGϕ : on all paths starting here in all states ϕ holds
  - EG ϕ: there is a path starting here along which ϕ holds
  - AF ϕ : on all paths starting here ϕ holds at least once
  - EF ϕ: there is a path starting here where ϕ holds at least once
  - AGF ϕ: on all paths starting here there is always a state ahead where ϕ holds (ϕ holds infinitely often).

Computation tree logic CTL*

- Useful for expressing safety and liveness properties and verifying whether some protocol satisfies them
- More details in lecture on model checking

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 □ we have modal operator K for Knows. Truth definition the same as for □.
- 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).

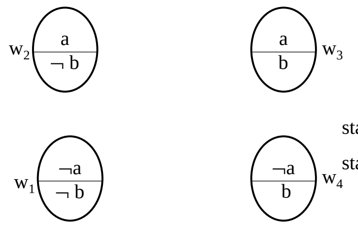
Intuition

- Suppose we have agents A and B with local states *state(A)* and *state(B)*. The global state of the system (possible world w) consists of w_A = *state(A)* at w, w_B = *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 |= $K_A\,\phi$ if in all states v where $v_A = w_A$, M,v |= ϕ.

- Somehow A manages to correlate its state with ϕ. It only goes into state w_A when ϕ is true and never when ϕ is false.

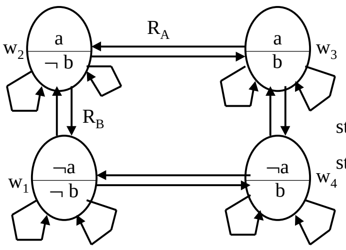
Epistemic accessibility relations



state(A) = {a}

state(B) = {b}

Epistemic accessibility relations



state(A) = {a}

state(B) = {b}

M,w_1 |= K_A (K_B b $\vee$ K_B $\neg$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.