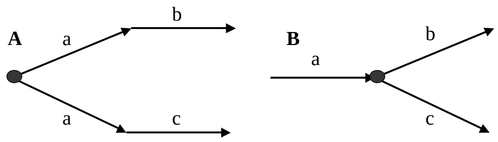
Modal logic and first order logic

- Modal logic: local view of the structure (“where can I get by following links from here”).
- First order logic: global view of the structure (can see everything, quantifiers do not follow edges).
- Meaningful equivalence between structures:
  - first order logic: (partial) isomorphism
  - modal logic: two structures should have the same “edge-following behaviour”

Equivalence between transition systems

- Trace equivalence?

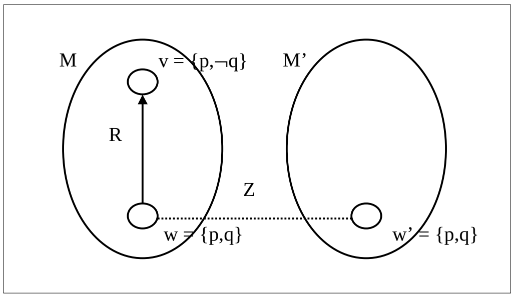


choice between b and c is made at different points

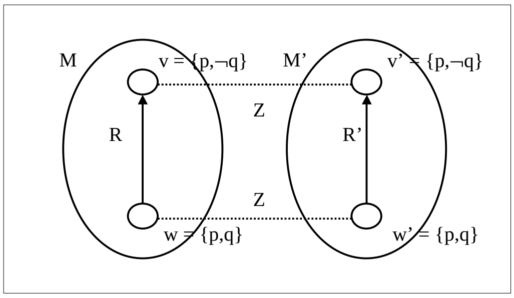
Bisimilation

- Let M = (W,R,V) and M’ = (W’,R’,V’) be to Kripke structures. A non-empty binary relation Z is called a *bisimulation* between M and M’ if the following conditions are satisfied:
  - if Z(w,w’) then w and w’ satisfy the same propositional letters
  - if Z(w,w’) and R(w,v) in M, then there exists v’ in M’ such that R’(w’,v’) and Z(v,v’) (the forth condition)
  - if Z(w,w’) and R’(w’,v’) in M’, then there exists v in M such that R(w,v) and Z(v,v’) (the back condition)

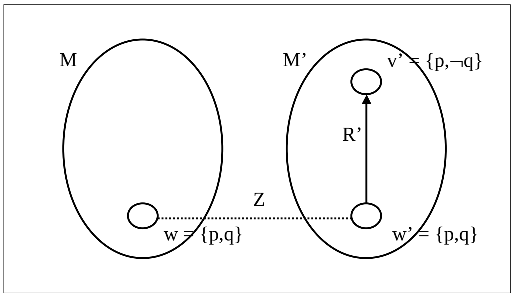
In pictures: forth condition



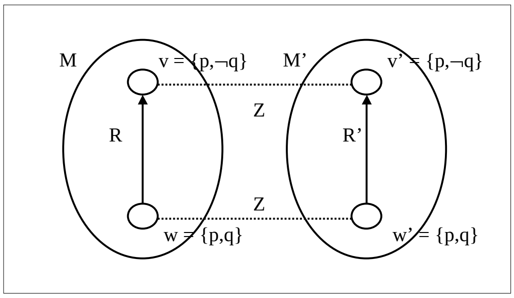
In pictures: forth condition



In pictures: back condition



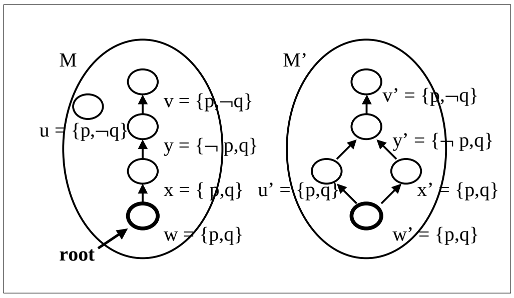
In pictures: back condition



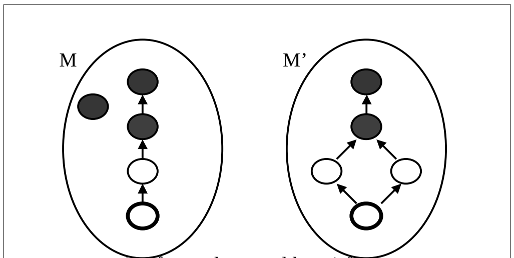
Bisimilar structures

- We shall call M and M’ bisimilar if there is a non-empty bisimulation relation between them.
- Sometimes we consider *rooted structures:* structures where there is a distinguished root/initial state of the system.
- For rooted structures, M and M’ are bisimilar if there is a bisimulation Z such that (root of M, root of M’) are in Z.
- Let us look at rooted examples.

Examples: are M and M’ bisimilar?

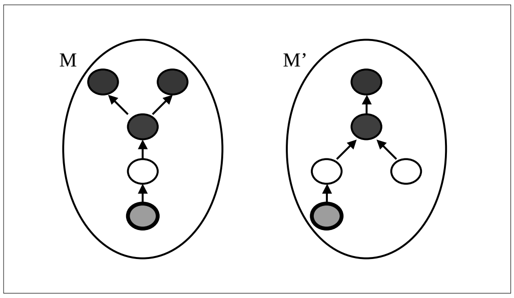


Examples: are M and M’ bisimilar?

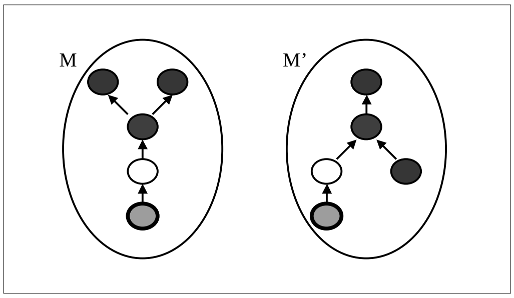


easier if we colour worlds satisfying same propositional variables the same colour!

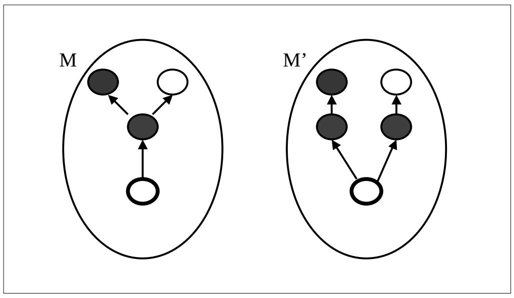
Are M and M’ bisimilar?



Are M and M’ bisimilar?



Are M and M’ bisimilar?



Bisimulation and modal logic

- Suppose M and M’ are to Kripke structures and Z is a bisimulation between M and M such that Z(w,w’).
- Theorem: for every formula ϕ of basic modal logic,

$M,w \models \phi$ iff $M',w' \models \phi$

- In other words, modal logic cannot distinguish bisimilar structures (can describe them up to bisimulation equivalence).

Proof

$M,w \models \phi$ iff $M',w' \models \phi$ if $Z(w,w')$

- Proof by induction on ϕ.
  - ϕ is a propositional variable: from Z(w,w’) w and w’ satisfy the same propositional variables
  - ϕ is $\neg\psi$: $M,w \models \neg\psi$ iff $M,w \not\models \psi$ iff (by the inductive hypothesis) $M',w' \not\models \psi$ iff $M,w \models \neg\psi$
  - ϕ is $\psi\to\chi$: $M,w \models \psi\to\chi$ iff $M,w \not\models \psi$ or $M,w \models \chi$ iff (by the inductive hypothesis) $M',w' \not\models \psi$or $M',w' \models \chi$ iff $M',w' \models \psi\to\chi$.

Proof continued

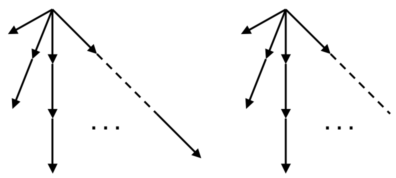
$M,w \models \phi$ iff $M',w' \models \phi$ if $Z(w,w')$

- Proof by induction on ϕ.
  - ϕ is $[]\psi$: $M,w \models []\psi$ iff for all v such that R(w,v), $M,v \models \psi$. Suppose $M,w \models []\psi$ and $M',w' \not\models []\psi$. Then there is a world v’ in M’ such that R’(w’,v’) and $M',v' \not\models \psi$. By the *back* condition, there is a v in M such that R(w,v) and Z(v,v’). By the inductive hypothesis, $M,v \not\models \psi$. So $M,w \not\models []\psi$: a contradiction.
  - Similarly for $M',w' \models []\psi$ and $M,w \not\models []\psi$.

Reverse?

- Modal logic does not distiguish bisimilar structures.
- Is the reverse true: if two structures are indistinguishable by modal formulas, they are bisimilar?
- This is true for finite structures. If two finite structures satisfy the same modal formulas, then they are bisimilar.

Counterexample for the infinite case



a branch of length n for every n

a branch of length n for every n, plus an infinite branch

Some other illuminating properties

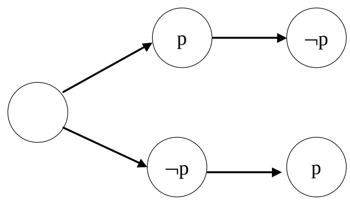
- Define modal depth of ϕ as the maximal depth of nesting of modalities in ϕ.
- For example,

$\Diamond(p \wedge [\,]\, \neg p) \wedge \Diamond(\neg p \wedge [\,]\, p) \wedge [\,]((p \wedge [\,]\neg p) \vee (\neg p \wedge [\,]\, p))$

has modal depth 2.

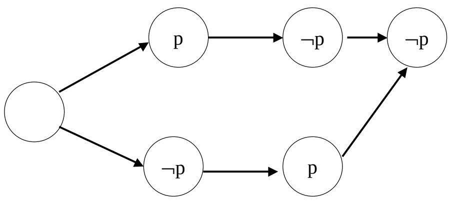
- A formula of modal depth n can't see further than n steps from the current world. If we change something about the worlds accessible in more than n steps, the truth value of formula will not change.

Example



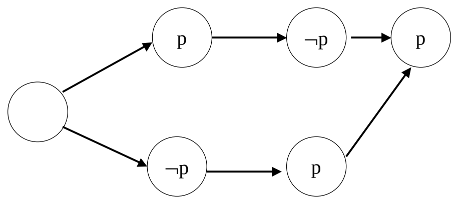
$\Diamond(p \wedge [\,]\neg p) \wedge \Diamond(\neg p \wedge [\,]p) \wedge [\,]((p \wedge [\,]\neg p) \vee (\neg p \wedge [\,]p))$

Example



$\Diamond(p \wedge [\,]\neg p) \wedge \Diamond(\neg p \wedge [\,]p) \wedge [\,]((p \wedge [\,]\neg p) \vee (\neg p \wedge [\,]p))$

Example

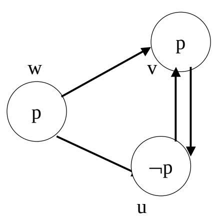


$\Diamond(p \wedge [\,]\neg p) \wedge \Diamond(\neg p \wedge [\,]p) \wedge [\,]((p \wedge [\,]\neg p) \vee (\neg p \wedge [\,]p))$

Illuminating properties continued

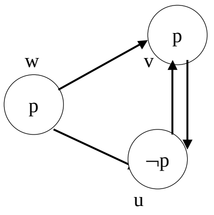
- Tree model: every satisfiable formula has a tree model.
- We obtain it by *unravelling* the original model into a tree (and proving that it is bisimilar to the original model).
- By our previous result, we already know that every satisfiable formula has a finite model.
- If we unravel a finite cyclic model, we get an infinite tree.
- However, we know that the formula only cares about worlds accessible in n steps, where n is the formula's modal depth.
- So we can chop the infinite tree after n levels.
- Each satisfiable formula has a finite tree model.

Example: unravelling

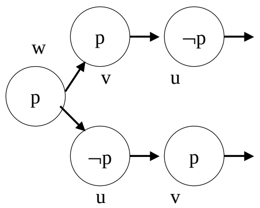


Example: unravelling

- Original structure

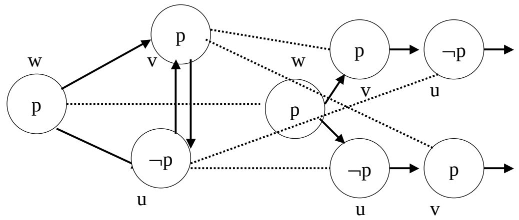


Unravelled structure



Example: unravelling

- Each world is bisimilar to all copies of itself in the tree



Bisimilarity checking algorithm

- Suppose we are given two rooted graphs (W’,R’,w’) and (W’’,R’’,w’’) (ignore V for the moment). We want to check if they are bisimilar with respect to back and forth conditions, that is if there is a bisimulation Z between them such that Z(w’,w’’).
- This is basically the same as saying: is there a bisimulation-induced equivalence relation on the set of vertices of the graph (W,R) = (W’$\cup$W’’, R’ $\cup$ R’’) such that w’ and w’’ are in the same equivalence class?

Relational coarsest partition

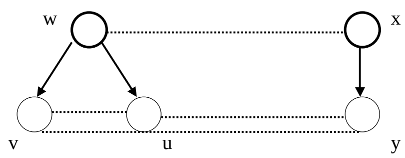
- Let V, V’ $\subseteq$ W
- Define $R^{-1}(V)$ as {w in W: for some v in V, R(w,v)}
- V is *stable* with respect to V’ if either V $\subseteq$ R^{-1}(V’) or V$\cap R^{-1}$(V’) = $\emptyset$.
- Intuitively, V’ is `a kind of vertices’ and to be an equivalence class, V should either only see vertices in V’ or not see any vertices in V’. Otherwise vertices in V are not equivalent with respect to seeing vertices in V’.
- If V is not stable with respect to V’, say V’ *splits* V.
- We need to find a coarsest stable partition of W, i.e. such that none of the equivalence classes splits another.

Relational coarsest partition algorithm

- Start with any partition of Q of W (for example, Q={W,$\emptyset$}).
- Repeat until the resulting partition is stable:
  - find a set S which is a union of some of the equivalence classes in Q, such that S splits some of the classes in Q
  - replace Q = {Q_1,…,Q_n} with

{$Q_1 \cap R^{-1}$(S), Q_1-R^{-1}(S),…, $Q_n \cap R^{-1}$(S), Q_n-R^{-1}(S)}

Example



Q = { W, $\emptyset$} splitter: W; R^{-1}(W) ={w,x}

Q = {{w,x}, {v,u,y} }

now stable

Exercise

- The algorithm above ignores propositional variables.
- Write a version which does not ignore variables (checks for bisimulation satisfying all three conditions).