Multi-Agent Systems

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel, Felix Lindner, and Thorsten Engesser

Summer Term 2017

Course outline

1. Introduction
2. Agent-Based Simulation
3. Agent Architectures
4. Beliefs, Desires, Intentions
   - The GOAL Agent Programming Language
   - Introduction to Modal Logics
     - Part I: Graphical Models, Kripke Models
     - Part II: Syntax & Semantics
     - Part III: Model Construction via Tableaux
   - Epistemic Logic
   - BDI Logic
5. Norms and Duties
6. Communication and Argumentation
7. Coordination and Decision Making

Recap

- Last week
  - Kripke models represent specific situations involving Knowledge, Desires, Obligations, ...
  - The language of modal logic can be used to formally talk about Kripke models.
  - Model Checking: Given a formula, is it true in possible world w in Kripke model M?
- Today
  - Beyond specific situations: Automated satisfiability checking.

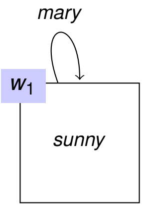
Motivating Example

Consider the personal-assistant robot Alfred. Alfred maintains knowledge about the people he cares for. E.g., Alfred can represent that Mary knows that the sun is shining (and therefore there is no need to tell her about the weather conditions).

Modeling Alfred's Knowledge: Model Checking vs. Theorem Proving

- Traditionally, two approaches can be distinguished (cf., [3] for a discussion):
  - What the agent knows is represented as a Kripke model. Reasoning is modeled as deleting/adding nodes/edges, and model checking.
  - What the agent knows is represented as a set of formulae. Reasoning is modeled as deleting/adding formulae, and theorem proving.

M_0:



$KB = \{K_{mary} sunny\}$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question "Does Tom know it is sunny?"

- Let's consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom} sunny$, thus $M_0, w_1 \not\models \neg K_{tom} sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom} sunny$, thus $M_1, w_1 \not\models \neg K_{tom} sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom} sunny$, thus $M_2, w_1 \models \neg K_{tom} sunny$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question "Does Tom know it is sunny?"

- Let's consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom} sunny$, thus $M_0, w_1 \not\models \neg K_{tom} sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom} sunny$, thus $M_1, w_1 \not\models \neg K_{tom} sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom} sunny$, thus $M_2, w_1 \models \neg K_{tom} sunny$

Absence of Knowledge I

UNI FREIBURG

What about the things Alfred has no knowledge about? How to respond to the question “Does Tom know it is sunny?”

- Let’s consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom}sunny$, thus $M_0, w_1 \not\models \neg K_{tom}sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom}sunny$, thus $M_1, w_1 \not\models \neg K_{tom}sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom}sunny$, thus $M_2, w_1 \models \neg K_{tom}sunny$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question "Does Tom know it is sunny?"

- Let's consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom} sunny$, thus $M_0, w_1 \not\models \neg K_{tom} sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom} sunny$, thus $M_1, w_1 \not\models \neg K_{tom} sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom} sunny$, thus $M_2, w_1 \models \neg K_{tom} sunny$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question “Does Tom know it is sunny?”

- Let’s consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom}sunny$, thus $M_0, w_1 \not\models \neg K_{tom}sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom}sunny$, thus $M_1, w_1 \not\models \neg K_{tom}sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom}sunny$, thus $M_2, w_1 \models \neg K_{tom}sunny$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question “Does Tom know it is sunny?”

- Let’s consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom} sunny$, thus $M_0, w_1 \not\models \neg K_{tom} sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom} sunny$, thus $M_1, w_1 \not\models \neg K_{tom} sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom} sunny$, thus $M_2, w_1 \models \neg K_{tom} sunny$

Absence of Knowledge I

What about the things Alfred has no knowledge about? How to respond to the question “Does Tom know it is sunny?”

- Let’s consider some possibilities:
  - M_0 : Take $R(K_{tom})$ as empty (somewhat illegaly):
    - $M_0, w_1 \models K_{tom} sunny$, thus $M_0, w_1 \not\models \neg K_{tom} sunny$
  - M_1 : Make $R(K_{tom})$ a minimalistic equivalence relation:
    - $M_1, w_1 \models K_{tom} sunny$, thus $M_1, w_1 \not\models \neg K_{tom} sunny$
  - M_2 : Tom does not know whether it is sunny:
    - $M_2, w_1 \not\models K_{tom} sunny$, thus $M_2, w_1 \models \neg K_{tom} sunny$
- Observations
  1. While each of M_0, M_1, M_2 agrees about Mary’s knowledge (of which Alfred is sure), they disagree about Tom’s knowledge (of which Alfred has no information).
  2. Why make a choice? Alfred’s answer should be “Maybe, depends on how the world actually looks like...”
     $\Rightarrow$Consider all possible models.

Absence of Knowledge II

Assume Alfred's knowledge is given by a knowledge base $KB = \{K_{mary} sunny\}$. The formula $K_{mary} sunny$ represents all the possible worlds w in all models M such that $M, w \models K_{mary} sunny$.

- From what Alfred knows, does it follow that Tom knows it is sunny?
  - $KB \models_{\mathbf{S5}_n} K_{tom} sunny$? Answer: No, because there are models in which KB is true and $K_{tom} sunny$ is false (e.g., M_2).
  - $KB \models_{\mathbf{S5}_n} \neg K_{tom} sunny$? Answer: No, because there are models in which KB is true and $\neg K_{tom} sunny$ is false (e.g., M_1).

$\Rightarrow$ It is possible that both a formula and its negation are satisfiable. In this case, none of them is valid, and the agent may answer "Maybe, depends on how the world actually looks like..."

Absence of Knowledge II

Assume Alfred's knowledge is given by a knowledge base $KB = \{K_{mary} sunny\}$. The formula $K_{mary} sunny$ represents all the possible worlds w in all models M such that $M, w \models K_{mary} sunny$.

- From what Alfred knows, does it follow that Tom knows it is sunny?
  - $KB \models_{\mathbf{S5}_n} K_{tom} sunny$? Answer: No, because there are models in which KB is true and $K_{tom} sunny$ is false (e.g., M_2).
  - $KB \models_{\mathbf{S5}_n} \neg K_{tom} sunny$? Answer: No, because there are models in which KB is true and $\neg K_{tom} sunny$ is false (e.g., M_1).

$\Rightarrow$It is possible that both a formula and its negation are satisfiable. In this case, none of them is valid, and the agent may answer "Maybe, depends on how the world actually looks like..."

Absence of Knowledge II

Assume Alfred's knowledge is given by a knowledge base $KB = \{K_{mary} sunny\}$. The formula $K_{mary} sunny$ represents all the possible worlds w in all models M such that $M, w \models K_{mary} sunny$.

- From what Alfred knows, does it follow that Tom knows it is sunny?
  - $KB \models_{\mathbf{S5}_n} K_{tom} sunny$? Answer: No, because there are models in which KB is true and $K_{tom} sunny$ is false (e.g., M_2).
  - $KB \models_{\mathbf{S5}_n} \neg K_{tom} sunny$? Answer: No, because there are models in which KB is true and $\neg K_{tom} sunny$ is false (e.g., M_1).

$\Rightarrow$It is possible that both a formula and its negation are satisfiable. In this case, none of them is valid, and the agent may answer "Maybe, depends on how the world actually looks like..."

Conclusions and Outlook

- Wanted: A procedure to check satisfiability of a modal-logic formula.
  - Can then be used to check validity of a formula by proving its negation unsatisfiable.
- Good news: Satisfiability is decidable for all the modal logics we consider.
- Approach: For a given formula, we will try to construct a Kripke model. If we succeed, the input formula is satisfiable. If we fail, the input formula is unsatisfiable (and thus its negation is valid).
  - Next: Sound, Complete, and Terminating precedure described in [1, 2].

Premodels

Def. Premodel

Given a set of labels L, a premodel is a labelled graph $M = (W, R, V)$ where: W is a non-empty set, $R : L \to 2^{W \times W}$, $V : L \to 2^{W}$.

- Idea
  - First, a premodel is initialized with an input formula whose satisfiability should be proven.
  - Then, rules transform the premodel to other premodels by systematically adding nodes, edges, and formulae.
  - Finally, if no more rules are applicable, a Kripke model can be derived from a premodel iff the input formula is satisfiable.

Rules for Boolean Connectives

- And: If node contains formula $(\varphi \wedge \psi)$ then add φ and ψ.
- NotAnd: If node contains formula $\neg(\varphi \wedge \psi)$ then add $(\neg\varphi \vee \neg\psi)$.
- NotNot: If node contains formula $\neg\neg\varphi$ then add φ.
- NotOr: If node contains formula $\neg(\varphi \vee \psi)$ then add $(\neg\varphi \wedge \neg\psi)$.
- Or: If node contains formula $(\varphi \vee \psi)$ then copy the graph g to g' and add φ to the node in g and ψ to the node in g'.
- Impl: If node contains formula $(\varphi \rightarrow \psi)$ then add $(\neg\varphi \vee \psi)$.
- NotImpl: If node contains formula $\neg(\varphi \rightarrow \psi)$ then add $(\varphi \wedge \neg\psi)$.
- $\perp$: If node contains φ and $\neg\varphi$ then add $\perp$.

Saturation and Openness

- The rules for rewriting the graphs are applied as often as possible.
- A premodel is saturated when no more rule can be applied.
- Premodels with a node containing $\perp$ are called closed; otherwise they are called open.

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain?$

- to show: $\models ((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain$?

- to show: $\models ((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$$\boxed{\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)}$$

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain$?

- to show: $\models ((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$, $((rain \rightarrow wet) \wedge \neg wet)$, $rain$

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain$?

- to show: $\models ((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$,
$((rain \rightarrow wet) \wedge \neg wet)$, $rain$,
$(rain \rightarrow wet)$, $\neg wet$

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain$?

- to show: $\models ((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$\neg((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain)$,
$((rain \rightarrow wet) \land \neg wet)$, $rain$,
$(rain \rightarrow wet)$, $\neg wet$, $\neg rain \lor wet$

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain$?

- to show: $\models ((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$,
$((rain \rightarrow wet) \wedge \neg wet)$, $rain$,
$(rain \rightarrow wet)$, $\neg wet$, $\neg rain$

$\neg((rain \rightarrow wet) \wedge \neg wet) \rightarrow \neg rain)$,
$((rain \rightarrow wet) \wedge \neg wet)$, $rain$,
$(rain \rightarrow wet)$, $\neg wet$, wet

Example I

$\{(rain \rightarrow wet), \neg wet\} \models \neg rain?$

- to show: $\models ((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain)$
- Approach: Assume $\neg((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain)$ is satisfiable, and try to construct the satisfying Kripke model.

$\neg((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain),$
$((rain \rightarrow wet) \land \neg wet),\ rain,$
$(rain \rightarrow wet),\ \neg wet,\ \neg rain,\ \bot$

$\neg((rain \rightarrow wet) \land \neg wet) \rightarrow \neg rain),$
$((rain \rightarrow wet) \land \neg wet),\ rain,$
$(rain \rightarrow wet),\ \neg wet,\ wet,\ \bot$

$\Rightarrow$No open premodel found. Kripke model cannot be constructed. Formula is unsatisfiable. Hence, it's negation is valid (q.e.d).

Rules for Modalities

- $\langle \mathbf{I} \rangle$: If node contains formula $\langle \mathbf{I} \rangle \varphi$ and so far no I-successor contains φ then add an I-labeled edge to a new node that contains φ.
- $[\mathbf{I}]$: If node contains formula $[\mathbf{I}]\varphi$ then add φ to all I-connected nodes (that do not already contain φ).
- $\neg\langle \mathbf{I} \rangle$: If node contains formula $\neg\langle \mathbf{I} \rangle \varphi$ then add $\neg\varphi$ to all I-connected nodes (that do not already contain $\neg\varphi$).
- $\neg[\mathbf{I}]$: If node contains formula $\neg[\mathbf{I}]\varphi$ and so far no I-successor contains $\neg\varphi$ then add an I-labeled edge to a new node that contains $\neg\varphi$.

Example II

- to show: $\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes}$ is **K**-satisfiable.

$\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes}$

Example II

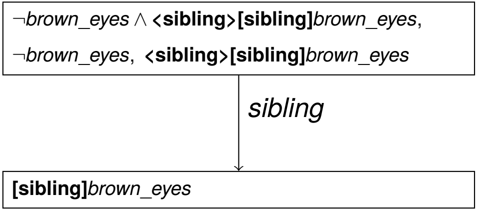
- to show: $\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes}$ is **K**-satisfiable.

$$\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes},$$

$$\neg \mathit{brown\_eyes}, \ \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes}$$

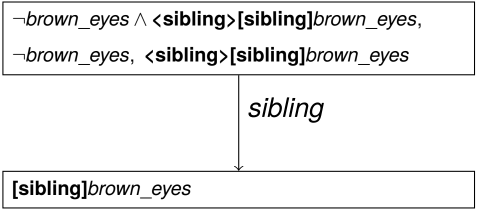
Example II

- to show: $\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}]\mathit{brown\_eyes}$ is **K**-satisfiable.



Example II

- to show: $\neg \textit{brown\_eyes} \wedge \langle \textbf{sibling} \rangle [\textbf{sibling}] \textit{brown\_eyes}$ is **K**-satisfiable.



- A Kripke model can be derived: $M = (W, R, V)$ with $W = \{w_1, w_2\}$, $R(\textit{sibling}) = \{(w_1, w_2)\}$, $V(\textit{brown\_eyes}) = \{\}$. Indeed $M, w_1 \models \neg \textit{brown\_eyes} \wedge \langle \textbf{sibling} \rangle [\textbf{sibling}] \textit{brown\_eyes}$.
- Problem: The relation **sibling** should be symmetric.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle \mathbf{I} \rangle p \wedge [\mathbf{I}]\langle \mathbf{I} \rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains $[\mathbf{I}]$ then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle \mathbf{I} \rangle p \wedge [\mathbf{I}]\langle \mathbf{I} \rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains $[\mathbf{I}]$ then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle \mathbf{I} \rangle p \wedge [\mathbf{I}]\langle \mathbf{I} \rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains $[\mathbf{I}]$ then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle\mathbf{I}\rangle p \land [\mathbf{I}]\langle\mathbf{I}\rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains $[\mathbf{I}]$ then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle \mathbf{I} \rangle p \wedge [\mathbf{I}]\langle \mathbf{I} \rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains $[\mathbf{I}]$ then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no I-edge to itself then add one.
- symmetric (B): If there is an I-edge then add an I-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is I-connected to a second node which is I-connected to a third node then add an I-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle\mathbf{I}\rangle p \wedge [\mathbf{I}]\langle\mathbf{I}\rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no I-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no I-successor and contains $[\mathbf{I}]$ then add a new I-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no I-edge to itself then add one.
- symmetric (B): If there is an I-edge then add an I-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is I-connected to a second node which is I-connected to a third node then add an I-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle\mathbf{I}\rangle p \wedge [\mathbf{I}]\langle\mathbf{I}\rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no I-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no I-successor and contains $[\mathbf{I}]$ then add a new I-successor. After the premodel is built, add reflexive edges to leaf nodes.

Rules for Frame Classes

If $R(\mathbf{I})$ is supposed to be ...

- reflexive (T): If node has no *I*-edge to itself then add one.
- symmetric (B): If there is an *I*-edge then add an *I*-edge heading in the opposite direction (if non-existent yet).
- transitive (4): If a first node is *I*-connected to a second node which is *I*-connected to a third node then add an *I*-edge from the first to the third (if none exist yet).
  - Problem: E.g., $\langle \mathbf{I} \rangle p \wedge [\mathbf{I}]\langle \mathbf{I} \rangle p$ will create an infinite premodel.
  - Solution: Check if the new node is equal to parent node. If yes, do not expand new node further.
- serial (D)
  - First attempt: If node has no *I*-successor then add one. (Problem: Won't terminate.)
  - Better: If node has no *I*-successor and contains **[I]** then add a new *I*-successor. After the premodel is built, add reflexive edges to leaf nodes.

Example II Revisited

- to show: ¬*brown_eyes* ∧ **<sibling>[sibling]***brown_eyes* is not **KB**-satisfiable.

¬*brown_eyes* ∧ **<sibling>[sibling]***brown_eyes*

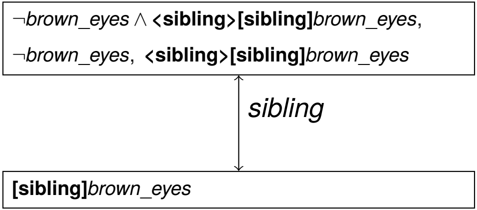
Example II Revisited

- to show: $\neg \mathit{brown\_eyes} \land \langle\mathbf{sibling}\rangle[\mathbf{sibling}]\mathit{brown\_eyes}$ is not **KB**-satisfiable.

$\neg \mathit{brown\_eyes} \land \langle\mathbf{sibling}\rangle[\mathbf{sibling}]\mathit{brown\_eyes}$,
$\neg \mathit{brown\_eyes}$, $\langle\mathbf{sibling}\rangle[\mathbf{sibling}]\mathit{brown\_eyes}$

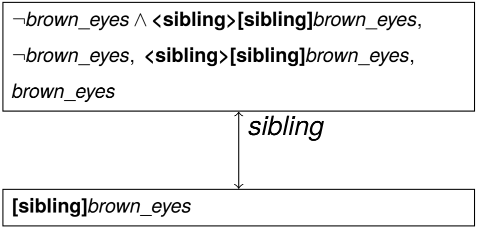
Example II Revisited

- to show: ¬*brown_eyes* ∧ **<sibling>[sibling]***brown_eyes* is not **KB**-satisfiable.



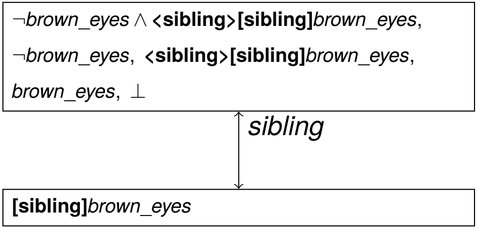
Example II Revisited

- to show: ¬*brown_eyes* ∧ **<sibling>[sibling]***brown_eyes* is not **KB**-satisfiable.



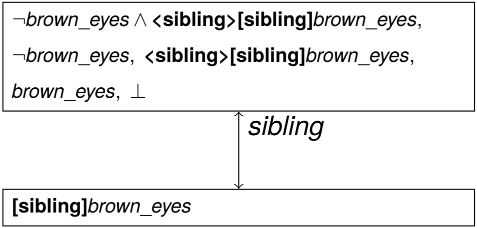
Example II Revisited

- to show: $\neg \mathit{brown\_eyes} \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] \mathit{brown\_eyes}$ is not **KB**-satisfiable.



Example II Revisited

- to show: $\neg brown\_eyes \wedge \langle \mathbf{sibling} \rangle [\mathbf{sibling}] brown\_eyes$ is not **KB**-satisfiable.



- No Kripke model can be derived. $\Rightarrow$The formula is unsatisfiable in **KB**, hence its negation is **KB**-valid (q.e.d).

Remark: Multiple Modalities

- Different modalities can be mixed. E.g., the approach also works for $\mathbf{S5}_n$ (multi-agent knowledge), which we will have a closer look on next time. E.g., also $K_{mary}K_{tom}p \rightarrow K_{mary}p$ is valid in $\mathbf{S5}_n$.
- However, in general one has to mind undesired interactions. E.g., mixing the epistemic modality K (**S5**) and the deontic modality O (**KD**) yields the validity $\models_{\mathbf{S5} \otimes \mathbf{KD}} OKp \rightarrow Op$, which says that only obligatory facts must be known.

Course outline

1. Introduction
2. Agent-Based Simulation
3. Agent Architectures
4. Beliefs, Desires, Intentions
   - The GOAL Agent Programming Language
   - Introduction to Modal Logics
   - Epistemic Logic
   - BDI Logic
5. Norms and Duties
6. Communication and Argumentation
7. Coordination and Decision Making

Literature I

- B. Said, Graph rewriting for model construction in modal logic, PhD Thesis, Université Paul Sabatier – Toulouse III, 2010. Available Online: `https://tel.archives-ouvertes.fr/tel-00466115/`
- O. Gasquet, A. Herzig, B. Said, F. Schwarzentruber, Kripke's Worlds — An Introduction to Modal Logics via Tableaux, Springer, ISBN 978-3-7643-8503-3, 2014.
- J. Y. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. Artificial Intelligence and Mathematical Theory of Computation, 212:151–176, 1991.