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
Applications of Logics in MAS

- Specification
  - The intended behavior of a MAS can be specified using a logical specification language. The concrete program is derived from the specification (manually, in most cases).

- Verification
  - Once a program $P$ is built, one wishes to be able to prove that it behaves according to its specification $\varphi_P$, i.e., $P \models \varphi_P$.

- Agent programming
  - Agents themselves can be realized deductive reasoners: What an agent knows is represented as formulae of a formal language. The agent can reason about these formulae to derive new formulae, or to determine what to do next.
Model Checking

Definition

Model checking is an automated technique that, given a finite-state model of a system and a formal property, systematically checks whether this property holds for (a given state in) that model.

- Model of the system $\Rightarrow$ How the system actually behaves.
- Formal properties $\Rightarrow$ How the system should behave.
  - Safety: something bad never happen
  - Liveness: something good eventually happens
  - Fairness: if something may happen frequently, it will happen
Runtime Verification

Definition

Runtime verification is the discipline of computer science that deals with the study, development, and application of those verification techniques that allow checking whether a run of a system under scrutiny satisfies or violates a given correctness property.

⇒ Testing using formal methods.
Requirements

- **Question**: Does a given BDI agent act right (viz., according to some specified properties)?
- **Required**
  - Representation of the agent’s execution.
  - Language to specify the wanted properties.
  - Algorithm to check if some given properties hold in some representation of an execution.
Ingredients:

- Action and time
- Belief and preference
- Definition of intention

---

1The following notations are according to Meyer, Broersen, Herzig (2015). They slightly deviate from the original notations in Cohen, Levesque (1990).
A BDI Kripke model is a tuple $M = (\mathcal{W}, R, B, P, V)$, where:

- $\mathcal{W}$ is a set of possible worlds.
- $R : I \times A \rightarrow \mathcal{W} \times \mathcal{W}$
  - Accessibility relations $R_i : \alpha \subseteq \mathcal{W} \times \mathcal{W}$ for each action $i : \alpha$.
  - $(\mathcal{W}, R)$ is a linear transition system.
- $B : I \rightarrow \mathcal{W} \times \mathcal{W}$
  - Accessibility relations $B_i \subseteq \mathcal{W} \times \mathcal{W}$ for each agent $i$.
  - Every $B_i$ is serial, transitive, Euclidean ($\textbf{KD45}$).
- $P : I \rightarrow \mathcal{W} \times \mathcal{W}$
  - Accessibility relations $P_i \subseteq B_i \subseteq \mathcal{W} \times \mathcal{W}$ for each agent $i$.
  - Every $P_i$ is serial ($\textbf{KD}$).
- $V : P \rightarrow 2^\mathcal{W}$ maps atomic propositions to their extension $V(p) \subseteq \mathcal{W}$. 
Actions: Example I

1: wakeUp

1: goUni

1: goLect

atUni

atLecture

asleep

awake

awake

atUni

awake

asleep
Actions: Operators

- $M, w \models \text{Happ}_{i: \alpha} \varphi$ iff. there is a $w' \in W$ s.th. $(w, w') \in R_{i: \alpha}$ and $M, w' \models \varphi$ (⇒ diamond operator).

- $M, w \models \text{IfHapp}_{i: \alpha} \varphi$ iff. $M, w \models \neg \text{Happ}_{i: \alpha} \neg \varphi$ (⇒ box operator).

- $M, w \models \exists \alpha \text{Happ}_{i: \alpha} \varphi$ iff. there are agent $i$, action type $\alpha$ and $w'$ s.th. $(w, w') \in R_{i: \alpha}$ and $M, w' \models \varphi$.

- Linearity is characterised by the axiom
  \[(\text{Happ}_{i: \alpha} \top \land \text{Happ}_{j: \alpha'} \varphi) \rightarrow \text{IfHapp}_{i: \alpha} \varphi.\] ("If $i : \alpha$ is executable and $j : \alpha'$ brings about $\varphi$, then also $i : \alpha$ brings about $\varphi$.")
- $M, w_1 \models \text{Happ}_{1: \text{wakeUp}} \text{awake}$
- $M, w_2 \models \exists \alpha \text{Happ}_{1: \alpha} \exists \beta \text{Happ}_{1: \beta} \text{atLecture}$
- $M, w \models X \varphi$ iff. $M, w' \models \varphi$ for some $w'$ s.th. $(w, w') \in R_{i: \alpha}$ for some $i : \alpha$.
- $M, w \models F \varphi$ iff. $M, w \models \varphi$ or $M, w \models XF \varphi$.
- $M, w \models G \varphi$ iff. $M, w \models \neg F \neg \varphi$.
- $M, w \models \psi U \varphi$ iff. $M, w \models \varphi$ or $(M, w \models \psi$ and $M, w' \models \psi U \varphi$) for some $w'$ s.th. $(w, w') \in R_{i: \alpha}$ for some $i : \alpha$. 
Time: Example

\[ M, w_1 \models X(awakeUatLecture) \]
\[ M, w_1 \models atSleep \land XFatSleep \]
\[ M, w_1 \models G(atSleep \leftrightarrow \neg awake) \]
\[ M, w_1 \models F \exists \alpha \text{Happ}_{1: \alpha} atLecture \]
Belief and Preference

- \( M, w \models Bel_i \phi \) iff. for all \( w' \) s.th. \((w, w') \in B_i: M, w' \models \phi \).
  - \( Know_i \phi \overset{\text{def}}{=} \phi \land Bel_i \phi \).
- \( M, w \models Pref_i \phi \) iff. for all \( w' \) s.th. \((w, w') \in P_i: M, w' \models \phi \).
  - In the original \( Pref \) is called \( Goal \). Some authors call it \( Choice \). It is meant to be a “chosen desire” (consistent!).

Properties

- For \( Bel_i \) all properties for \( KD45 \) operators.
- For \( Pref_i \) all properties for \( KD \) operators.
- \( \models Bel_i \phi \rightarrow Pref_i \phi \) (Realism)
- \( \models (Pref_i \phi \land Bel_i (\phi \rightarrow \psi)) \rightarrow Pref_i \psi \).
Belief and Preference: Example
Preferences alone are too weak

Because of realism, all believed propositions are preferred propositions. But it only makes sense for an agent to adopt some goal $\phi$ if $\phi$ is believed to be false (compare to the GOAL programming language).
Agent $i$ has the **achievement goal** that $\varphi$ iff $i$ prefers that $\varphi$ is eventually true and believes that $\varphi$ is currently false:

$$\text{AGoal}_i \varphi \overset{\text{def}}{=} \text{Pref}_i F \varphi \land \text{Bel}_i \neg \varphi$$

**Example**

In the Netflix-vs.-Lecture dilemma:

- $M, w_1 \not\models \text{AGoal}_1(\text{asleep})$
- $M, w_1 \models \text{AGoal}_1(\text{watching})$
Achievement Goal: Properties

\[\models A\text{Goal}_i \neg \varphi \rightarrow \neg A\text{Goal}_i \varphi.\]

Check that \(A\text{Goal}_i \neg \varphi \land A\text{Goal}_i \varphi\) is unsatisfiable, because the achievement goal that \(\neg \varphi\) implies to believe \(\varphi\), and the achievement goal that \(\varphi\) implies to believe \(\neg \varphi\). This contradicts axiom D \((B_{ei} \varphi \rightarrow \neg B_{ei} \neg \varphi)\).

\[\not\models A\text{Goal}_i (\varphi \land \psi) \rightarrow A\text{Goal}_i \varphi \land A\text{Goal}_i \psi\] (for exercise).

\[\not\models A\text{Goal}_i \varphi \land A\text{Goal}_i \psi \rightarrow A\text{Goal}_i (\varphi \land \psi).\]

\[\not\models A\text{Goal}_i (\varphi \lor \psi) \rightarrow A\text{Goal}_i \varphi \lor A\text{Goal}_i \psi.\]

\[\not\models A\text{Goal}_i \varphi \lor A\text{Goal}_i \psi \rightarrow A\text{Goal}_i (\varphi \lor \psi).\]
\[ \not \models A\text{Goal}_i \varphi \land A\text{Goal}_i \psi \rightarrow A\text{Goal}_i (\varphi \land \psi) \]

“Lisa has the goal to listen to the lecture and she has the goal to have dinner” vs. “Lisa has the goal to listen to the lecture and to have dinner”

\[ M, w_1 \models A\text{Goal}_1 (at\text{Lecture}) \land A\text{Goal}_1 (have\text{Lunch}) \]

\[ M, w_1 \not\models A\text{Goal}_1 (at\text{Lecture} \land have\text{Lunch}) \]
\[ \not \models A\text{Goal}_i(\varphi \lor \psi) \rightarrow A\text{Goal}_i\varphi \lor A\text{Goal}_i\psi. \]

“Paul asks Lisa whether she likes him.” (Paul does not prefer any of the two possible answers.)

\[ M, w_1 \models A\text{Goal}_p(Know_p\text{like} \lor Know_p\text{dislike}) \]
\[ M, w_1 \not\models A\text{Goal}_p(Know_p\text{like}) \]
\[ M, w_1 \not\models A\text{Goal}_p(Know_p\text{dislike}) \]
\[ \not \models A\text{Goal}_i \varphi \lor A\text{Goal}_i \psi \rightarrow A\text{Goal}_i(\varphi \lor \psi) \]

- \( M, w_1 \models A\text{Goal}_1(\text{haveLunch}) \)
- \( M, w_1 \not\models A\text{Goal}_1(\text{atLecture}) \)
  - Reason: \( M, w_1 \not\models Bel_i \neg \text{atLecture} \)
- \( M, w_1 \not\models A\text{Goal}_1(\text{atLecture} \lor \text{haveLunch}) \)
  - Reason: \( M, w_1 \not\models Bel_i(\neg(\text{atLecture} \lor \text{haveLunch})) \)
Agents can change their preferences whenever they like: Lack of commitment!

- $M, w_1 \models \text{AGoal}_1(\text{haveLunch})$
- $M, w_2 \models \neg\text{AGoal}_1(\text{haveLunch})$
The Nell problem

Say a problem solver is confronted with the classic situation of a heroine, called Nell, having been tied to the tracks while a train approaches. The problem solver, called Dudley, knows that “If Nell is going to be mashed, I must remove her from the tracks.” When Dudley deduces that he must do something, he looks for, and eventually executes, a plan for doing it. This will involve finding out where Nell is, and making a navigation plan to get to her location. Assume that he knows where she is, and he is not too far away; then the fact that the plan will be carried out will be added to Dudley’s world model. Dudley must have some kind of database consistency maintainer to make sure that the plan is deleted if it is no longer necessary. Unfortunately, as soon as an apparently successful plan is added to the world model, the consistency maintainer will notice that “Nell is going to be mashed” is no longer true. But that removes any justification for the plan, so it goes too. But that means “Nell is going to be mashed” is no longer contradictory, so it comes back in. And so forth.
Cohen & Levesque’s intentions involve commitment. Having a commitment means having a persistent goal, viz., a goal the agent only abandons if s(he) comes to believe that the goal is fulfilled or unreachable. This is called single-minded commitment.

Other forms of commitment:

- **Blind commitment**: The agent maintains its intention until it is actually achieved.
- **Open-minded commitment**: The agent maintains its intention as long as it is still believed possible. It may e.g. be rendered impossible by adapting new intentions.
Agent $i$ has the **persistent goal** that $\varphi$ iff $i$ has the achievement goal that $\varphi$ and will keep that goal until it is either fulfilled or believed to be out of reach:

$$PGoal_i \varphi \overset{\text{def}}{=} AGoal_i \varphi \land (AGoal_i \varphi)U(Bel_i \varphi \lor Bel_i G\neg \varphi)$$

- $M, w_1 \models PGoal_i(atLecture)$
Agent $i$ has the intention that $\varphi$ iff $i$ has the persistent goal that $\varphi$ and believes that (s)he can achieve $\varphi$ by an action.

$$\text{Intend}_i \varphi \overset{\text{def}}{=} P\text{Goal}_i \varphi \land B\text{el}_i F \exists \alpha \text{Happ}_i: \alpha \varphi$$

Intending is acting! An agent $1$ cannot intend that some other agent $2$ does something. However, $1$ may intend to make $2$ do something.

Viz., $\text{Intend}_1 \text{Happ}_{2: \text{act}} \top$ expands to $P\text{Goal}_1 \text{Happ}_{2: \text{act}} \top \land B\text{el}_1 F \exists \alpha \text{Happ}_i: \alpha \text{Happ}_2: \alpha \text{Happ}_{2: \text{act}} \top$
\[
\neg (\text{Intend}_i \varphi \land \text{Bel}_i G (\varphi \rightarrow \psi)) \rightarrow \text{Intend}_i \psi.
\]

**Proof**

We provide a model for \( \text{Intend}_i \varphi \land \text{Bel}_i G (\varphi \rightarrow \psi) \land \neg \text{Intend}_i \psi \):

John intends to go to the dentist. He believes that going to the dentist always implies pain. At the dentist, John gets some painkiller.
Dentist Example

\[
M, w_1 \models \text{Intend}_l(treatment) \land \text{Bel}_l G(treatment \rightarrow pain), \text{ but:}
\]
\[
M, w_2 \not\models \text{AGoal}_l(pain), \text{ thus:}
\]
\[
M, w_1 \not\models \text{PGoal}_l(pain), \text{ thus:}
\]
\[
M, w_1 \not\models \text{Intend}_l(pain)
\]
Remark: Runtime Verification

Sketch

1. Observe the execution of the system to be verified (e.g., log state of the environment, mental state of the agents, the agents’ actions).
2. Represent the execution log using the semantics of Cohen & Levesque.
3. Model check representation against the agents’ specification, e.g.:
   - $G(\text{goldNear} \rightarrow \text{Intend}(\text{hasGold}))$
   - $G(\text{Bel}(\text{goldNear}) \rightarrow \text{Intend}(\text{hasGold}))$
   - $G(\text{battLow} \rightarrow \text{Intend}(\neg\text{battLow}))$
4. Find time points where the specification evaluates false \(\Rightarrow\) Fault detection.
Recap and Outlook

- We have studied an integrated logical framework that captures many aspects of agent behavior taking belief and knowledge, preferences, goals, and intentions into account, as well as how these mental attitudes change through time as progressed by actions.

- Next time, we’ll look at another important notion, obligations and permissions, and we’ll briefly discuss a practical framework (BOID) that deals with decision making in light of conflicts between beliefs, desires, intentions, and obligations.
Course outline

1. Introduction
2. Agent-Based Simulation
3. Agent Architectures
4. Beliefs, Desires, Intentions
5. Norms and Duties
6. Communication and Argumentation
7. Coordination and Decision Making

