Knowledge Representation & Reasoning

Lecture 1: Introduction

dr Anna Maria Radzikowska

Warsaw University of Technology
Faculty of Mathematics and Information Science
Building MiNI PW, room 504

E-mail: Anna.Radzikowska@pw.edu.pl

Warsaw 2025

Outline

1. Course regulations
2. Checkpoints
3. Office hours
4. Literature
5. Course overview
6. Reasoning about Actions – basic issues
   - Applications of RAC
   - Ontologies of actions
   - Main Problems in RAC

Course Regulations

Course regulations

1. The course consists of lectures (30 hours) and laboratories (30 hours).
2. ATTENDANCE:
   - Lectures – not compulsory;
   - Laboratories - 4 checkpoints are compulsory for everybody, exact dates are fixed and given on next slides.
3. EVALUATION:
   - Test (quiz) – max. 20 points,
   - Laboratories - max. 20 points (submission in September max. 15 points).
   - Requirements – to pass the course **at least 11 points** from laboratory task **and at least 11 points** from test.

Course regulations

1. The course consists of lectures (30 hours) and laboratories (30 hours).
2. ATTENDANCE:
   - Lectures – not compulsory;
   - Laboratories - 4 checkpoints are compulsory for everybody, exact dates are fixed and given on next slides.
3. EVALUATION:
   - Test (quiz) – max. 20 points,
   - Laboratories - max. 20 points (submission in September max. 15 points).
   - Requirements – to pass the course **at least 11 points** from laboratory task **and at least 11 points** from test.

Course regulations

1. The course consists of lectures (30 hours) and laboratories (30 hours).
2. ATTENDANCE:
   - Lectures – not compulsory;
   - Laboratories - 4 checkpoints are compulsory for everybody, exact dates are fixed and given on next slides.
3. EVALUATION:
   - Test (quiz) – max. 20 points,
   - Laboratories - max. 20 points (submission in September max. 15 points).
   - Requirements – to pass the course **at least 11 points** from laboratory task **and at least 11 points** from test.

Course regulations

1. The course consists of lectures (30 hours) and laboratories (30 hours).
2. ATTENDANCE:
   - Lectures – not compulsory;
   - Laboratories - 4 checkpoints are compulsory for everybody, exact dates are fixed and given on next slides.
3. EVALUATION:
   - Test (quiz) – max. 20 points,
   - Laboratories - max. 20 points (submission in September max. 15 points).
   - Requirements – to pass the course **at least 11 points** from laboratory task **and at least 11 points** from test.

Course regulations

1. The course consists of lectures (30 hours) and laboratories (30 hours).
2. ATTENDANCE:
   - Lectures – not compulsory;
   - Laboratories - 4 checkpoints are compulsory for everybody, exact dates are fixed and given on next slides.
3. EVALUATION:
   - Test (quiz) – max. 20 points,
   - Laboratories - max. 20 points (submission in September max. 15 points).
   - Requirements – to pass the course **at least 11 points** from laboratory task **and** **at least 11 points** from test.

Course regulations (cont.)

4. LABORATORIES:
   - Students prepare 1 project task in teams of 3-4 members, teams draw projects from among those prepared by the course instructor, no changes are allowed during semester.
   - Each task has 3 main subtasks: a theoretical part, an implementation (application), and a testing part.
   - Each student participates in at least 2 parts, testing part is obligatory for everybody.
   - Results of each part should be presented during checkpoints and submitted in the form of printed and electronical documents.
   - Each part is evaluated separately; one delay of a week is allowed, father delays causes loss of 10% points per week.

Course regulations (cont.)

4. Laboratories:
   - Students prepare 1 project task in teams of 3-4 members, teams draw projects from among those prepared by the course instructor, no changes are allowed during semester.
   - Each task has 3 main subtasks: a theoretical part, an implementation (application), and a testing part.
   - Each student participates in at least 2 parts, testing part is obligatory for everybody.
   - Results of each part should be presented during checkpoints and submitted in the form of printed and electronical documents.
   - Each part is evaluated separately; one delay of a week is allowed, father delays causes loss of 10% points per week.

Course regulations (cont.)

4. Laboratories:
   - Students prepare 1 project task in teams of 3-4 members, teams draw projects from among those prepared by the course instructor, no changes are allowed during semester.
   - Each task has 3 main subtasks: a theoretical part, an implementation (application), and a testing part.
   - Each student participates in at least 2 parts, testing part is obligatory for everybody.
   - Results of each part should be presented during checkpoints and submitted in the form of printed and electronical documents.
   - Each part is evaluated separately; one delay of a week is allowed, father delays causes loss of 10% points per week.

Course regulations (cont.)

4. Laboratories:
   - Students prepare 1 project task in teams of 3-4 members, teams draw projects from among those prepared by the course instructor, no changes are allowed during semester.
   - Each task has 3 main subtasks: a theoretical part, an implementation (application), and a testing part.
   - Each student participates in at least 2 parts, testing part is obligatory for everybody.
   - Results of each part should be presented during checkpoints and submitted in the form of printed and electronical documents.
   - Each part is evaluated separately; one delay of a week is allowed, father delays causes loss of 10% points per week.

Course regulations (cont.)

4. LABORATORIES:
   - Students prepare 1 project task in teams of 3-4 members, teams draw projects from among those prepared by the course instructor, no changes are allowed during semester.
   - Each task has 3 main subtasks: a theoretical part, an implementation (application), and a testing part.
   - Each student participates in at least 2 parts, testing part is obligatory for everybody.
   - Results of each part should be presented during checkpoints and submitted in the form of printed and electronical documents.
   - Each part is evaluated separately; one delay of a week is allowed, father delays causes loss of 10% points per week.

Course regulations (cont.)

- Partial evaluation:
  - theoretical part: **max. 8p**;
  - application part: **max. 8p**;
  - testing part: **max. 4p**.
- Overall evaluation depends on the quality and individual contribution.
  - For theoretical/application part:
    - Basis: $B = n \cdot 16$ points (n – number of team members who participate in the respective part).
    - Evaluation: $P\%$.
    - $b = \frac{P}{100} \cdot B$ points to be distributed.
    - Individual contribution: $k\%$.
    - Individual evaluation: $\frac{k}{100} \cdot b$ points.
  - Testing part is evaluated separately for each team member.

Course regulations (cont.)

- Partial evaluation:
    - theoretical part: **max. 8p**;
    - application part: **max. 8p**;
    - testing part: **max. 4p**.
- Overall evaluation depends on the quality and individual contribution.
    - For theoretical/application part:
        - Basis: $B = n \cdot 16$ points (n – number of team members who participate in the respective part).
        - Evaluation: $P\%$.
        - $b = \frac{P}{100} \cdot B$ points to be distributed.
        - Individual contribution: $k\%$.
        - Individual evaluation: $\frac{k}{100} \cdot b$ points.
    - Testing part is evaluated separately for each team member.

Course regulations (cont.)

- Final submission:
  - Presentation and sumbission (exe file) of the prepared application.
  - Documentation:
    - Detailed technical description.
    - User guide.
    - Testing part.
    - Detailed contribution of each team member: description and percentage.

  The overall evaluation is fixed after the whole project is submitted and accepted.

- Final submission:
  - Presentation and sumbission (exe file) of the prepared application.
  - Documentation:
    - Detailed technical description.
    - User guide.
    - Testing part.
    - Detailed contribution of each team member: description and percentage.

  The overall evaluation is fixed after the whole project is submitted and accepted.

Course regulations (cont.)

- Final submission:
  - Presentation and sumbission (exe file) of the prepared application.
  - Documentation:
    - Detailed technical description.
    - User guide.
    - Testing part.
    - Detailed contribution of each team member: description and percentage.

  The overall evaluation is fixed after the whole project is submitted and accepted.

Course regulations (cont.)

5. TEST:
   - Test has only a written part, compulsory for all students.
   - It is a multiple choice quiz with 10 questions, each one has 4 options from among which 0, 1, 2, 3, or 4 ones are correct. The task is to mark all correct options and leave unmarked all incorrect ones. Then you receive 2 points, otherwise 0 points.
6. Lectures and current information are available at:
   - MS Teams, team **KRR 2025**
   - https://pages/mini.pw.edu.pl/~radzikowskaa/Courses/KRR 2025

Course regulations (cont.)

5. TEST:
   - Test has only a written part, compulsory for all students.
   - It is a multiple choice quiz with 10 questions, each one has 4 options from among which 0, 1, 2, 3, or 4 ones are correct. The task is to mark all correct options and leave unmarked all incorrect ones. Then you receive 2 points, otherwise 0 points.
6. Lectures and current information are available at:
   - MS Teams, team **KRR 2025**
   - https://pages/mini.pw.edu.pl/∼radzikowskaa/Courses/KRR 2025

Course regulations (cont.)

5. Final grades:
   - 23–25 points C (3.0)
   - 26–29 points C$\frac{1}{2}$ (3.5)
   - 30–33 points B (4.0)
   - 34–37 points B$\frac{1}{2}$ (4.5)
   - 38–40 points A (5.0)

Checkpoints

Checkpoint 1 – 20.03.2025

Progress in preparing a theoretical part

- Task 1: 15:00 – 15:15
- Task 2: 15:15 – 15:30
- Task 3: 15:30 – 15:45

Checkpoint 2 – 24.04.2025

Submission of the theoretical part

- Task 1: 14:30 – 15:00
- Task 2: 15:00 – 15:30
- Task 3: 15:30 – 16:00

Checkpoint 3 – 15.05.2025

Progress in application

- Task 1: 15:00 – 15:15
- Task 2: 15:15 – 15:30
- Task 3: 15:30 – 15:45

Checkpoint 4 – 5.06.2025

Project submission

- Task 1: 14:30 – 15:00
- Task 2: 15:00 – 15:30
- Task 3: 15:30 – 16:00

Office hours

Office hours

Tuesday 13-14, **on request!** If you need a meeting, please let me know **a couple of days before**.

Literature

Literature

1. Basic: presentations of lectures.
2. Additional:
   - R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Reasoning about Knowledge, The MIT Press, 1995.
   - R. Brachman, H. Levesque, Knowledge Representation and Reasoning. Morgan Kaufmann, 2004.
   - E. Mueller, Commonsense reasoning. Morgan Kaufmann Publishers, 2005.
   - E. Sandewall, Feature and Fluents: A Systematic Approach to the Representation of Knowledge of Dynamical Systems, Oxford University Press, 1994.
   - Proceedings of conferences *Principles of Knowledge Representation and Reasoning.*

Literature

1. Basic: presentations of lectures.
2. Additional:
   - R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Reasoning about Knowledge, The MIT Press, 1995.
   - R. Brachman, H. Levesque, Knowledge Representation and Reasoning. Morgan Kaufmann, 2004.
   - E. Mueller, Commonsense reasoning. Morgan Kaufmann Publishers, 2005.
   - E. Sandewall, Feature and Fluents: A Systematic Approach to the Representation of Knowledge of Dynamical Systems, Oxford University Press, 1994.
   - Proceedings of conferences *Principles of Knowledge Representation and Reasoning.*

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Course overview

- Reasoning about Actions and Changes.
- Foundations to Classical Logic
  - Automated Theorem Proving – Resolution principle
- Models and types of knowledge
- Default Logic
- Rough Sets and Learning from Examples.

Reasoning about Actions

Reasoning about Actions and Changes

Dynamic system

A **dynamic system** (DS) is viewed as

- a collection of objects, together with their properties, and
- a collection of actions which, while performed, change properties of objects (in consequence, the state of the world).

Main task

Define reasoning methods that allow for deriving conclusions about necessary/possible results of performing actions.

Reasoning about Actions (RAC)

The above task is the main objective in **Reasoning about Actions** (RAC).

Reasoning about Actions and Changes

Dynamic system

A **dynamic system** (DS) is viewed as

- a collection of objects, together with their properties, and
- a collection of actions which, while performed, change properties of objects (in consequence, the state of the world).

Main task

Define reasoning methods that allow for deriving conclusions about necessary/possible results of performing actions.

Reasoning about Actions (RAC)

The above task is the main objective in **Reasoning about Actions** (RAC).

Reasoning about Actions and Changes

Dynamic system

A **dynamic system** (DS) is viewed as

- a collection of objects, together with their properties, and
- a collection of actions which, while performed, change properties of objects (in consequence, the state of the world).

Main task

Define reasoning methods that allow for deriving conclusions about necessary/possible results of performing actions.

Reasoning about Actions (RAC)

The above task is the main objective in **Reasoning about Actions** (RAC).

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Applications of RAC

- AI systems that operate in real words.
- Design of autonomous agents (involving robotics, MAS).
- Planning tasks.
- Model–based diagnosis.
- Automatic control of complex systems.
- ...

Ontologies of actions

- **Effects of actions:**
  - deterministic (only one outcome), non-deterministic (at least two different outcomes)
  - certain, typical (preferred)
  - direct, indirect
  - occurring immediately after an action ends; delayed effects
  - all (some) effects are known.

Ontologies of actions

- **Effects of actions:**
  - deterministic (only one outcome), non-deterministic (at least two different outcomes)
  - certain, typical (preferred)
  - direct, indirect
  - occurring immediately after an action ends; delayed effects
  - all (some) effects are known.

Ontologies of actions

- **Effects of actions:**
  - deterministic (only one outcome), non-deterministic (at least two different outcomes)
  - certain, typical (preferred)
  - direct, indirect
  - occurring immediately after an action ends; delayed effects
  - all (some) effects are known.

Ontologies of actions

- **Effects of actions:**
  - deterministic (only one outcome), non-deterministic (at least two different outcomes)
  - certain, typical (preferred)
  - direct, indirect
  - occurring immediately after an action ends; delayed effects
  - all (some) effects are known.

Ontologies of actions

- **Effects of actions:**
  - deterministic (only one outcome), non-deterministic (at least two different outcomes)
  - certain, typical (preferred)
  - direct, indirect
  - occurring immediately after an action ends; delayed effects
  - all (some) effects are known.

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**
  - if do not hold, the effects of an action are unknown (the effects are empty).

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**
  - if do not hold, the effects of an action are unknown (the effects are empty).

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**
  - if do not hold, the effects of an action are unknown (the effects are empty).

Ontologies of actions (cont.)

- **Course of actions:**
  - 1-step actions, actions with durations
  - sequential actions, concurrent actions
  - hierarchical occurrences (actions/subactions), non-hierarchical occurrences.
- **Preconditions of actions:**
  - if do not hold, the effects of an action are unknown (the effects are empty).

Main problems in RAC

Frame problem (McCarthy & Hayes, 1969)

The difficulty is that of indicating and inferring all those things that do not change when actions are performed.

Ramification problem (Finger, 1987)

Concerns the problem of concisely representing indirect effects of actions (propagation of changes). It is usually unreasonable to explicity enumerate all of the consequences of actions

Qualification problem (McCarthy, 1977)

The number of preconditions of actions is usually immense and it is unreasonable (if ever possible) to explicitly enumerate and check all of (even most unlikely) possibilities.

Main problems in RAC

Frame problem (McCarthy & Hayes, 1969)

The difficulty is that of indicating and inferring all those things that do not change when actions are performed.

Ramification problem (Finger, 1987)

Concerns the problem of concisely representing indirect effects of actions (propagation of changes). It is usually unreasonable to explicity enumerate all of the consequences of actions

Qualification problem (McCarthy, 1977)

The number of preconditions of actions is usually immense and it is unreasonable (if ever possible) to explicitly enumerate and check all of (even most unlikely) possibilities.

Main problems in RAC

Frame problem (McCarthy & Hayes, 1969)

The difficulty is that of indicating and inferring all those things that do not change when actions are performed.

Ramification problem (Finger, 1987)

Concerns the problem of concisely representing indirect effects of actions (propagation of changes). It is usually unreasonable to explicity enumerate all of the consequences of actions

Qualification problem (McCarthy, 1977)

The number of preconditions of actions is usually immense and it is unreasonable (if ever possible) to explicitly enumerate and check all of (even most unlikely) possibilities.

Frame problem

Yale Shooting Problem

There is a shooter Bill and a turkey Fred. Initially Fred is alive and Bill has an unloaded gun. Bill can perform two actions: loading the gun (LOAD) *and shooting the gun* (SHOOT)*. Loading the gun makes the gun loaded, while shooting the gun makes it unloaded and, in addition, Fred in not alive anymore, provided that the gun was loaded.*

Natural conclusions:

- after loading the gun it is loaded and Fred is still alive
- after shooting the gun it is unloaded and Fred is not alive.

However, in classical logic we cannot infer that after loading the gun Fred is still alive!

Frame problem

Yale Shooting Problem

There is a shooter Bill and a turkey Fred. Initially Fred is alive and Bill has an unloaded gun. Bill can perform two actions: loading the gun (LOAD) *and shooting the gun* (SHOOT)*. Loading the gun makes the gun loaded, while shooting the gun makes it unloaded and, in addition, Fred in not alive anymore, provided that the gun was loaded.*

Natural conclusions:

- after loading the gun it is loaded and Fred is still alive
- after shooting the gun it is unloaded and Fred is not alive.

However, in classical logic we cannot infer that after loading the gun Fred is still alive!

Frame problem

Yale Shooting Problem

There is a shooter Bill and a turkey Fred. Initially Fred is alive and Bill has an unloaded gun. Bill can perform two actions: loading the gun (LOAD) *and shooting the gun* (SHOOT)*. Loading the gun makes the gun loaded, while shooting the gun makes it unloaded and, in addition, Fred in not alive anymore, provided that the gun was loaded.*

Natural conclusions:

- after loading the gun it is loaded and Fred is still alive
- after shooting the gun it is unloaded and Fred is not alive.

However, in classical logic we cannot infer that after loading the gun Fred is still alive!

Ramification problem

Modification of YSP

There is a shooter Bill and a turkey Fred. Whenever Fred is walking, it is alive. Assume that the gun is loaded. Shooting the gun makes it unloaded and, if it was loaded before shooting, Fred is no longer alive

Natural conclusion:

After shooting the gun Fred is no longer alive and, on addition, it does not walk!

In other words, "*Fred is not walking*" is the indirect effect of the shooting action!

Ramification problem

Modification of YSP

There is a shooter Bill and a turkey Fred. Whenever Fred is walking, it is alive. Assume that the gun is loaded. Shooting the gun makes it unloaded and, if it was loaded before shooting, Fred is no longer alive

Natural conclusion:

After shooting the gun Fred is no longer alive and, on addition, it does not walk!

In other words, "*Fred is not walking*" is the indirect effect of the shooting action!

Ramification problem

Modification of YSP

There is a shooter Bill and a turkey Fred. Whenever Fred is walking, it is alive. Assume that the gun is loaded. Shooting the gun makes it unloaded and, if it was loaded before shooting, Fred is no longer alive

Natural conclusion:

After shooting the gun Fred is no longer alive and, on addition, it does not walk!

In other words, "*Fred is not walking*" is the indirect effect of the shooting action!

Qualification problem

Potato in the tailpipe problem, McCarthy 1977

We want to start a car. Natural preconditions to do that are:

- *the key must be turned in the ignition,*
- *there must be gas in the tank,*
- *the battery must be connected,*
- *the wiring must be intact,*
- $\ldots$

Also, there need not be a potato in the tailpipe!

Therefore, it is hardly possible (or practical) to check all of unlikely qualifications each time we are interested in using the car.

Qualification problem

Potato in the tailpipe problem, McCarthy 1977

We want to start a car. Natural preconditions to do that are:

- *the key must be turned in the ignition,*
- *there must be gas in the tank,*
- *the battery must be connected,*
- *the wiring must be intact,*
- *...*

Also, there need not be a potato in the tailpipe!

Therefore, it is hardly possible (or practical) to check all of unlikely qualifications each time we are interested in using the car.

Qualification problem

Potato in the tailpipe problem, McCarthy 1977

We want to start a car. Natural preconditions to do that are:

- *the key must be turned in the ignition,*
- *there must be gas in the tank,*
- *the battery must be connected,*
- *the wiring must be intact,*
- *...*

Also, there need not be a potato in the tailpipe!

Therefore, it is hardly possible (or practical) to check all of unlikely qualifications each time we are interested in using the car.

Thank you for your attention!

Questions are welcome.