SCIENCE
PASSION
TECHNOLOGY

Course no. IND.04033UF (Lecture)
Course no. IND.04034UF (Practicals)

Logic and Computability

Bettina Könighofer
bettina.koenighofer@iaik.tugraz.at

Stefan Pranger
stefan.pranger@iaik.tugraz.at

October 7, 2024

Outline

- Team
- Administrative Information
  - Lecture
  - Practicals
- Outline
- Teaser

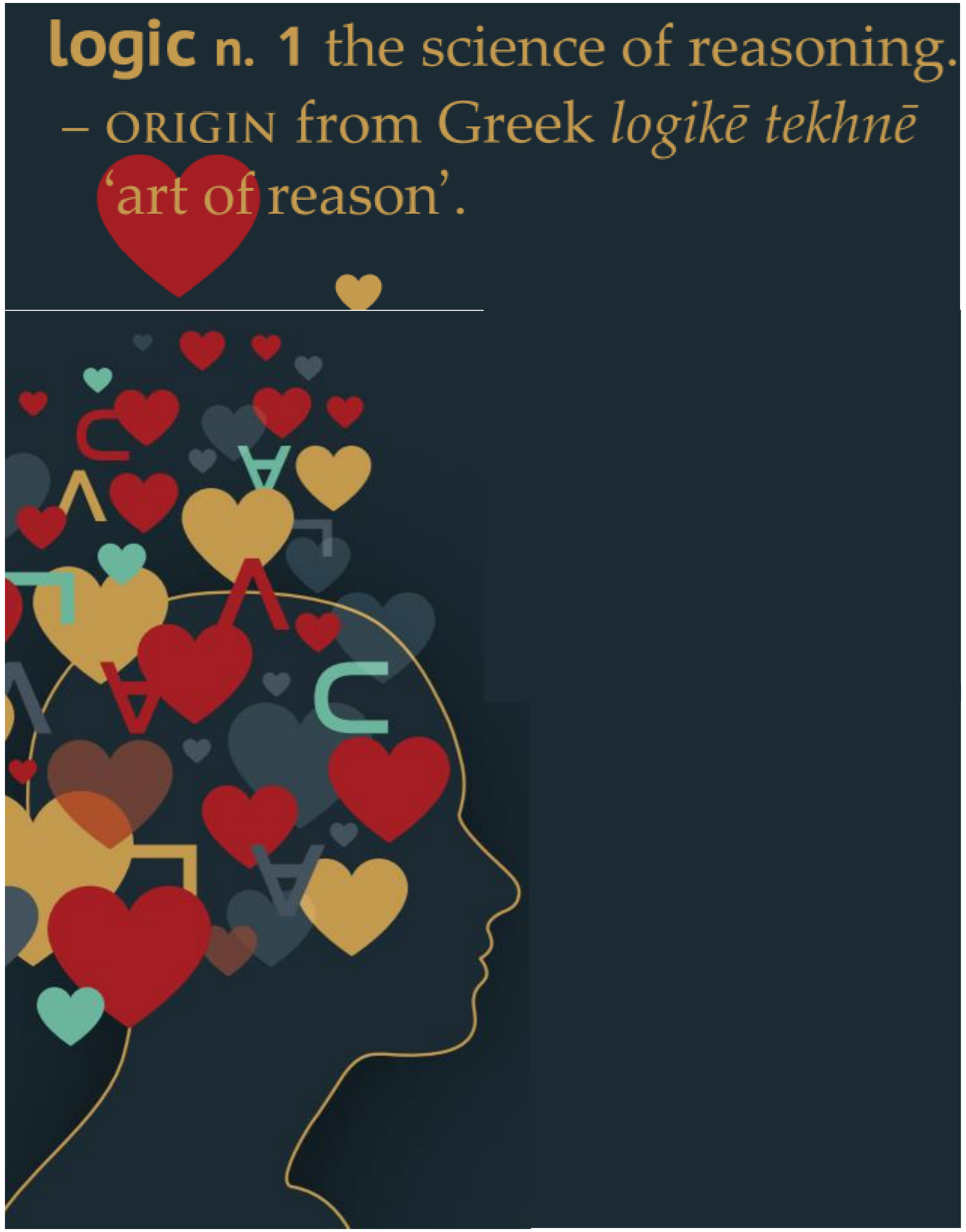
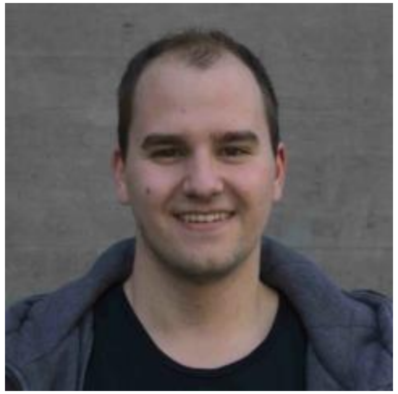
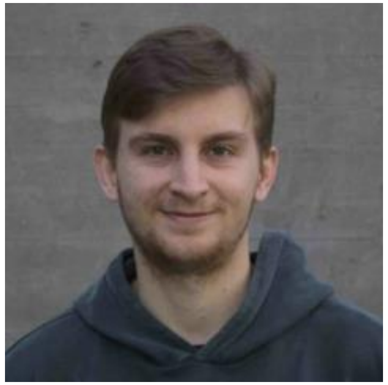


Image from Vienna Summer of Logic http://vsl2014.at/

Teaching Assistants

- Arthur Lippitz
  - arthur.lippitz@student.tugraz.at
- Lukas Schwarz
  - l.schwarz@student.tugraz.at
- Tamim Burgstaller
  - tamim.burgstaller@student.tugraz.at
- Verena Schaffer
  - verena.schaffer@student.tugraz.at
- Matthias Grilz
  - matthias.grilz@student.tugraz.at





Bettina Könighofer

- Assistant Professor at IAIK
- Team: Trusted AI Group

Bettina Könighofer

Filip Cano Cordoba

Stefan Pranger

- Teaching
  - Logic and Computability
  - Model Checking (CS Master)
  - ISW/Bachelor thesis/master thesis (IAIK website)

Our current Bachelor and Master students

Verena Schaffer

Mathias Grilz

Fabian Russold

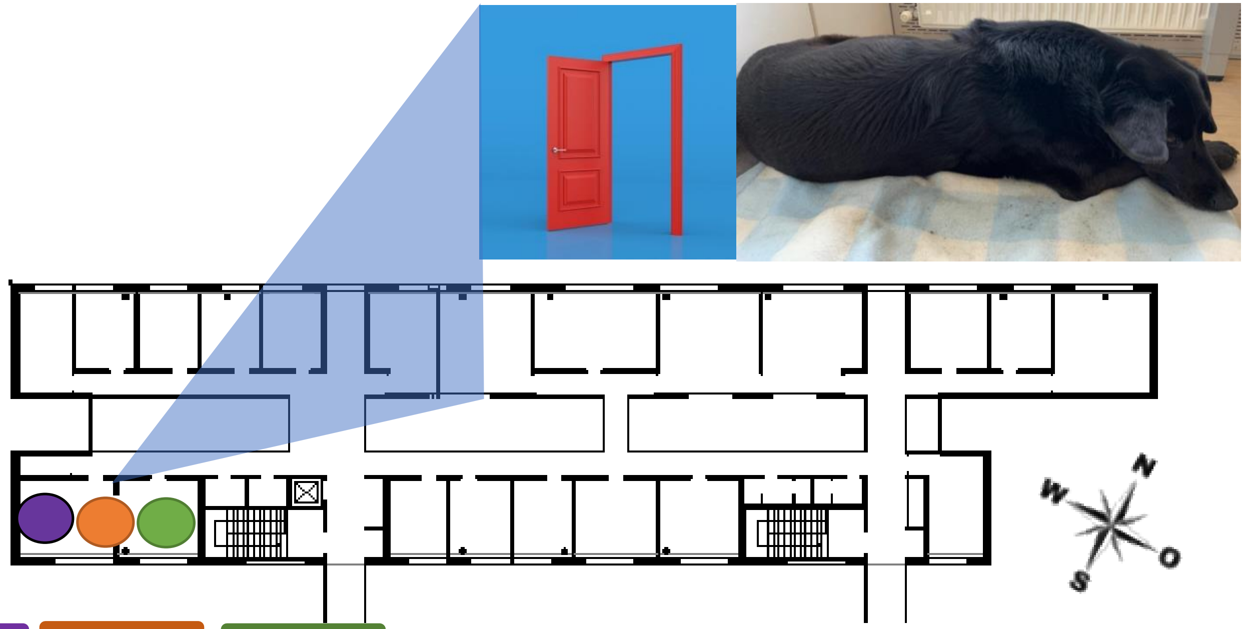
Thomas Knoll

Contact Details

- IAIK, Inffeldgasse 16a/II, Room IF02042
  - Open Door Policy

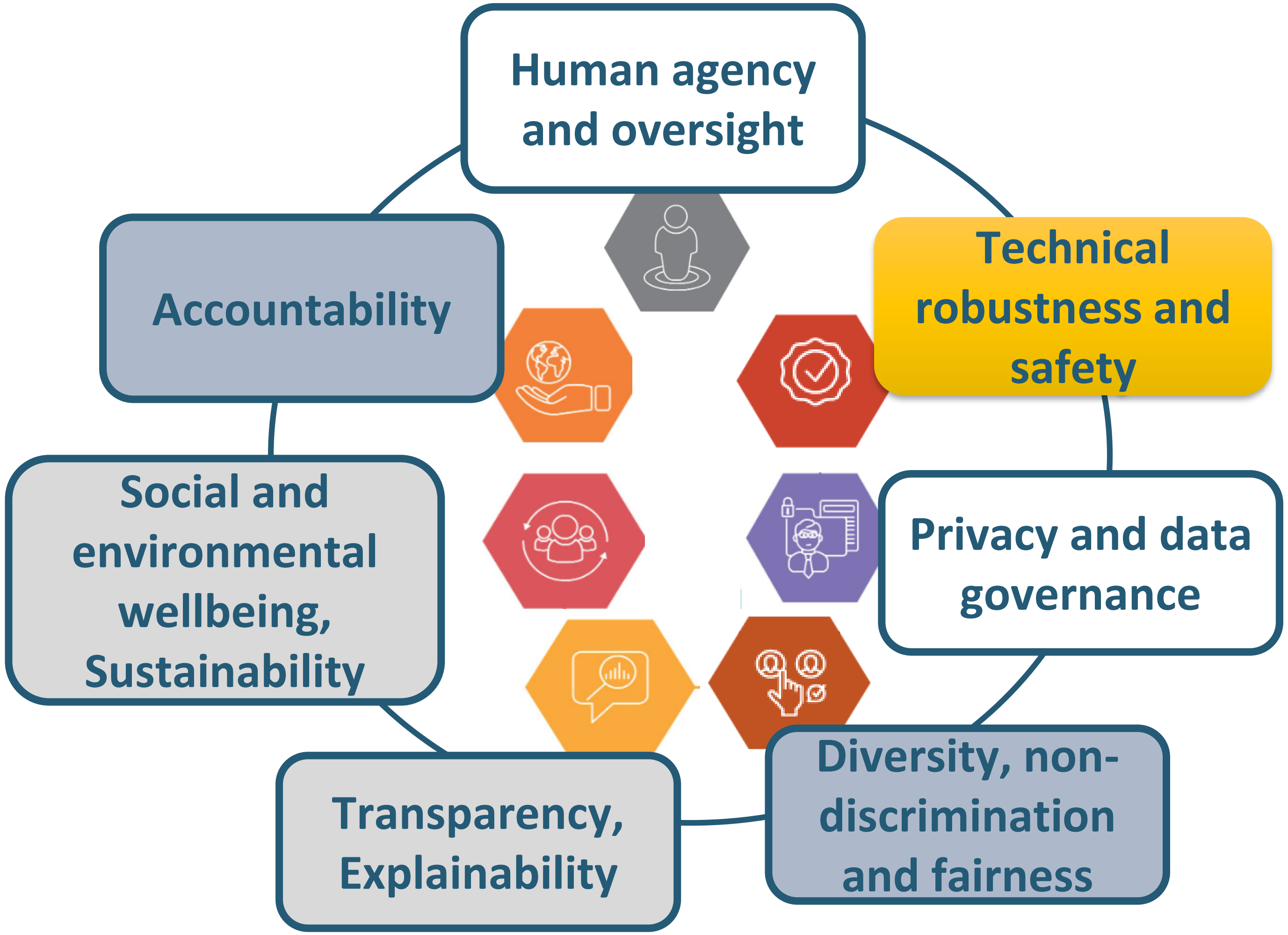
- 0316/873 – 5554
- bettina.koenighofer@iaik.tugraz.at
- stefan.pranger@iaik.tugraz.at

- Discord



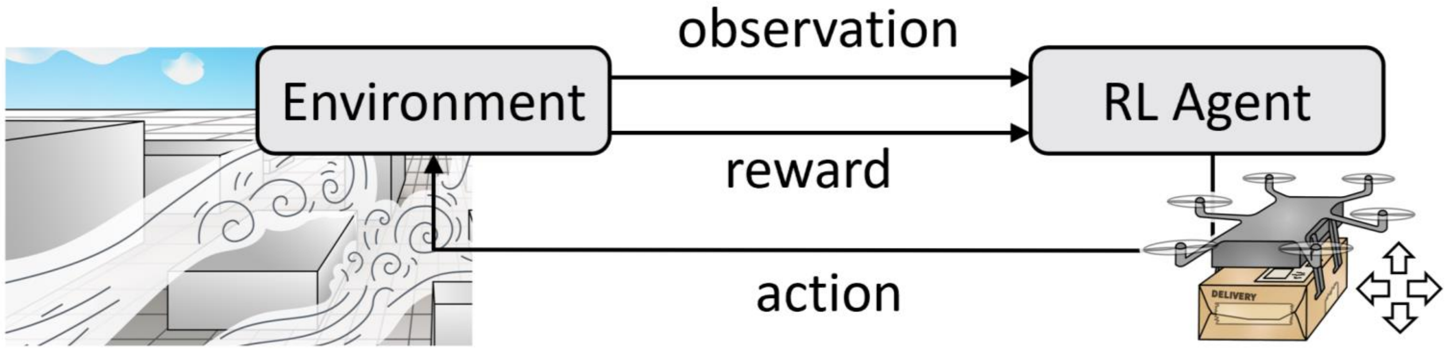
Research: Trustworthiness for AI Systems

- Combine (model-based) **Symbolic AI** and **Machine Learning**



Reinforcement Learning

- RL agent learns optimal policy via trial and error



Find a policy π that maximixes $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$

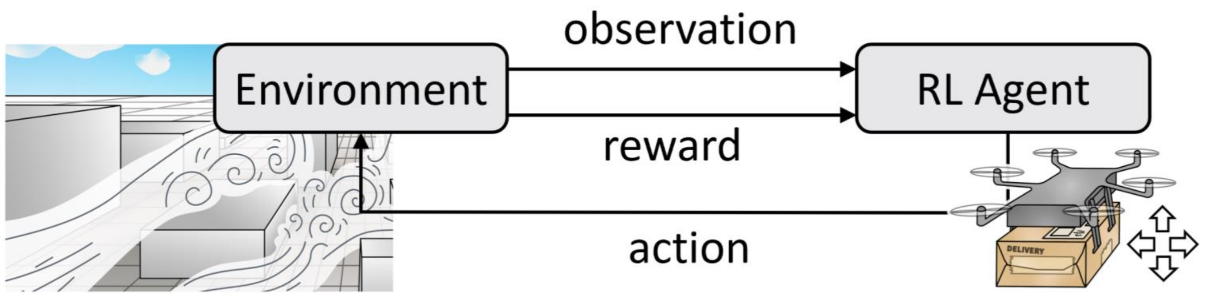
with the discount factor $0 \leq \gamma \leq 1$ and reward R_t at time t

N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Reinforcement Learning

Limitations

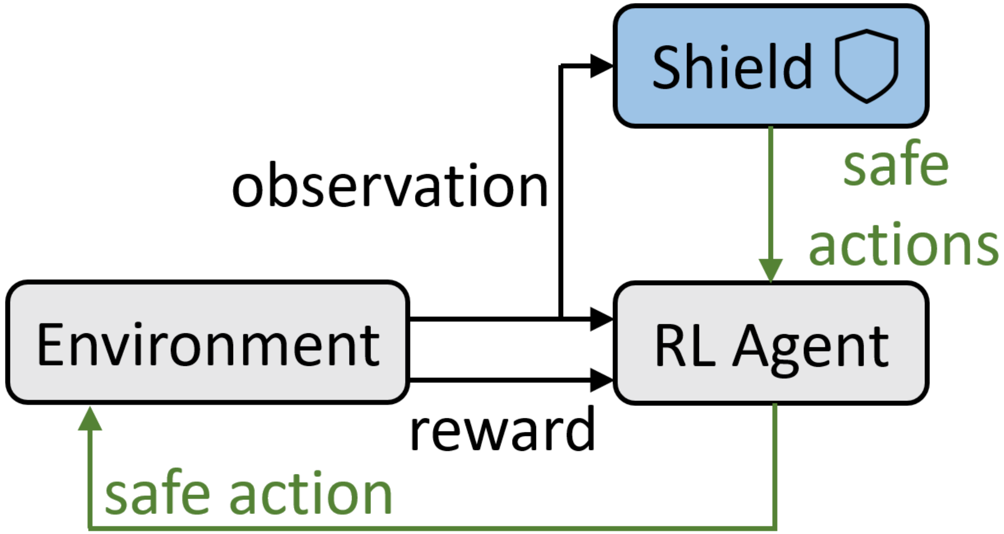
- Exploration is **safety-critical**
- RL is quite **sample inefficient**
- Rewards cannot capture **sophisticated task specifications**

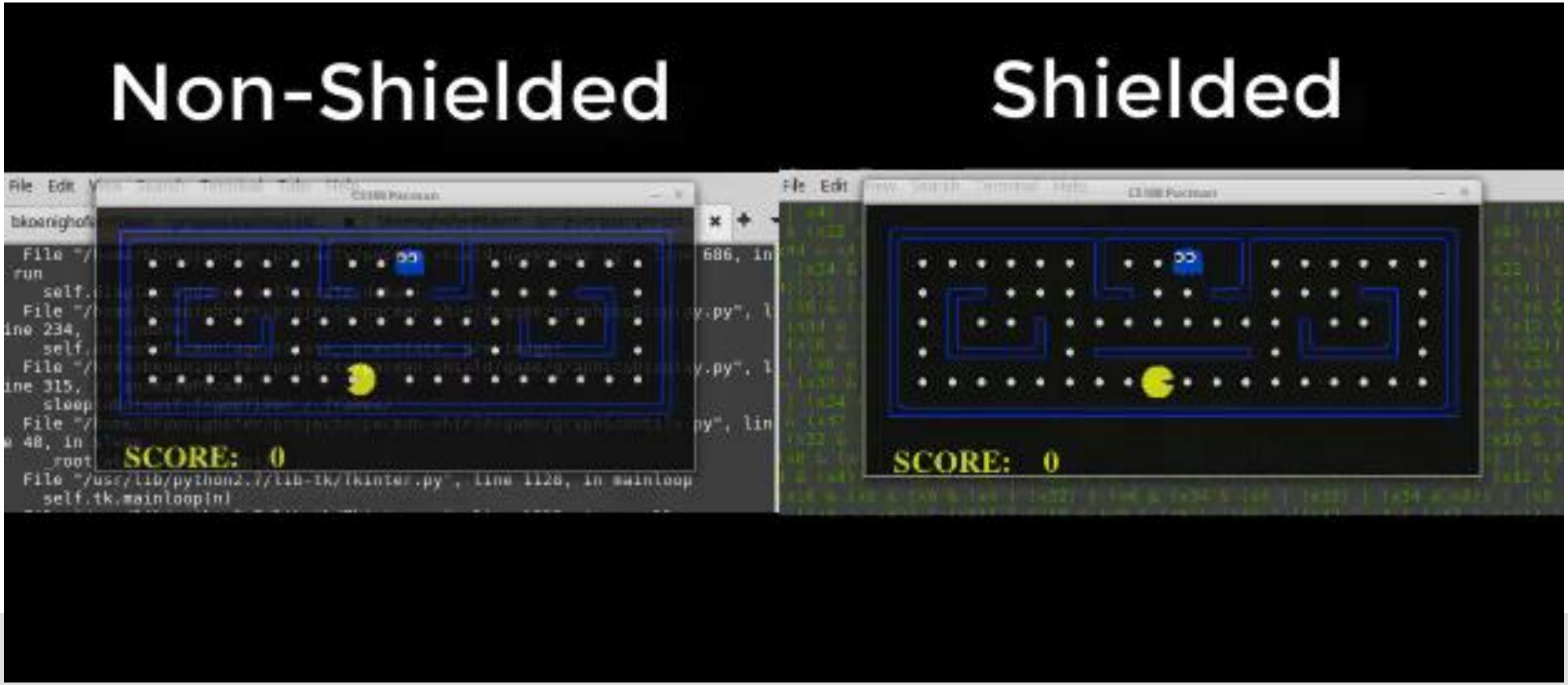


N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Shielded RL

- Mask unsafe actions

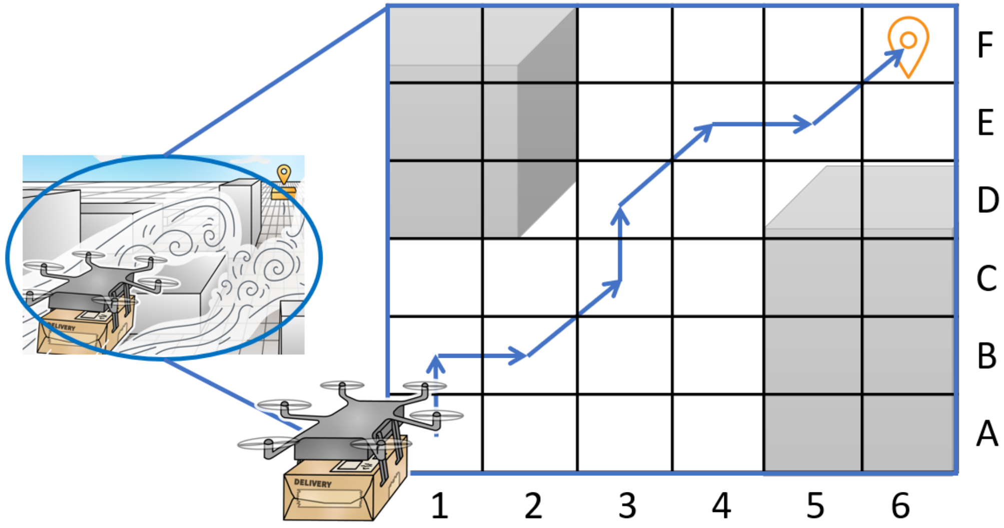




Shield Computation

- Safety specification in probabilistic temporal **logic**

„A crash can only occur with a **probability at most 1%**“



$$Pr_{\leq 0.01}\big(\mathbf{Eventually}\ (Crash)\big)$$

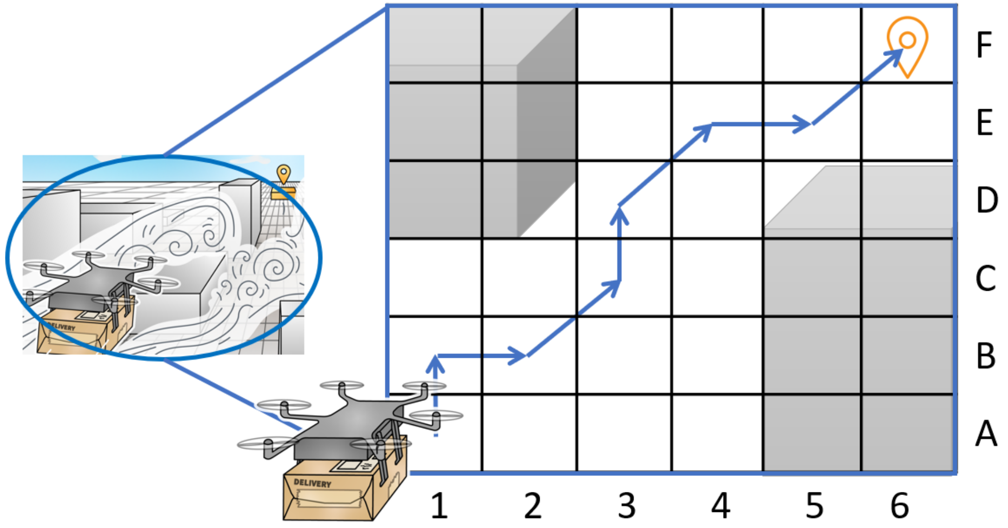
or

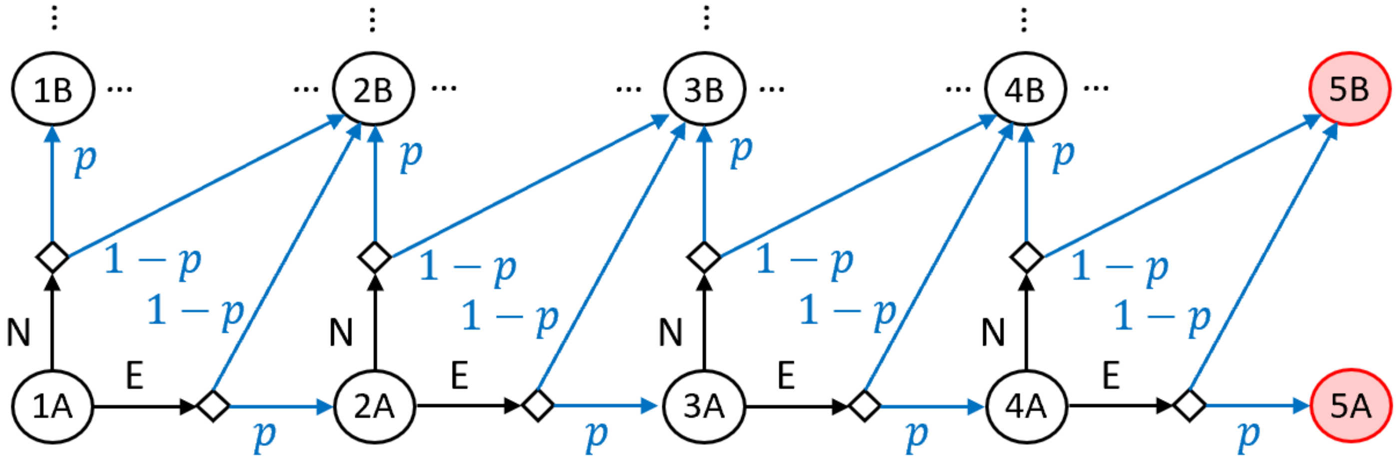
$$Pr_{\geq 0.99}\big(\mathbf{Always}(\neg Crash)\big)$$

N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Shield Computation

- Safety specification in probabilistic temporal **logic**
- Environment modelled as Markov Decision Process (MDP)
- Compute probabilities in MDP
  - Requires solving a dynamic program

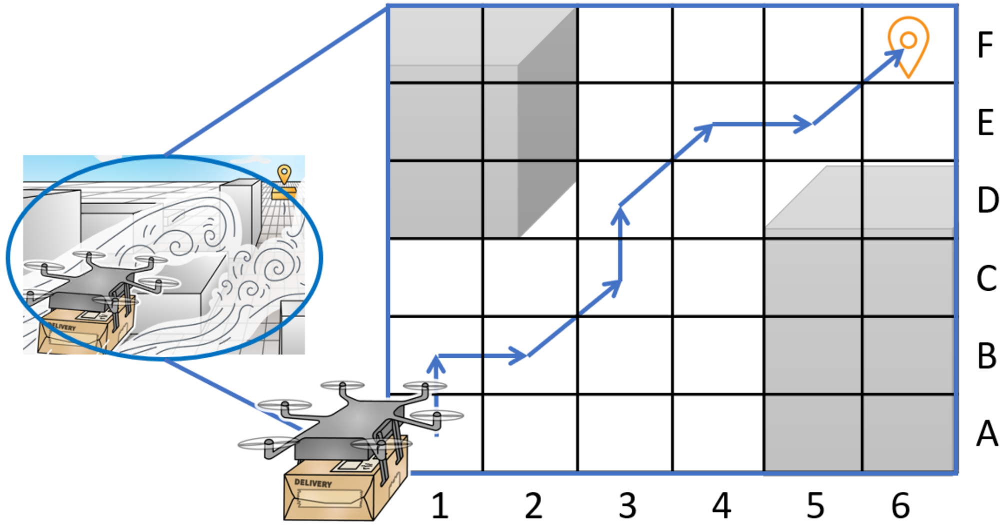


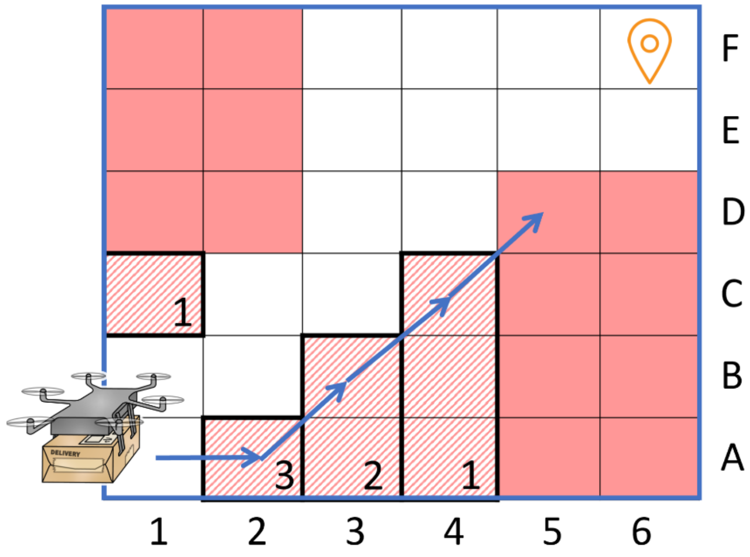


N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Shield Computation

- Safety specification in probabilistic temporal **logic**
- Environment modelled as Markov Decision Process (MDP)
- Compute probabilities in MDP
- Define threshold on allowed risk → **Shield**



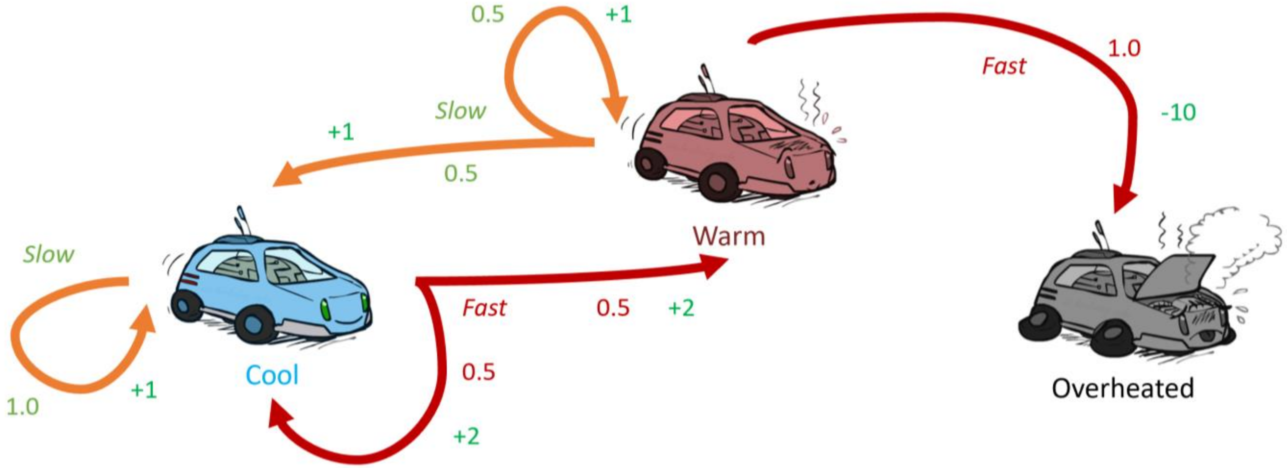


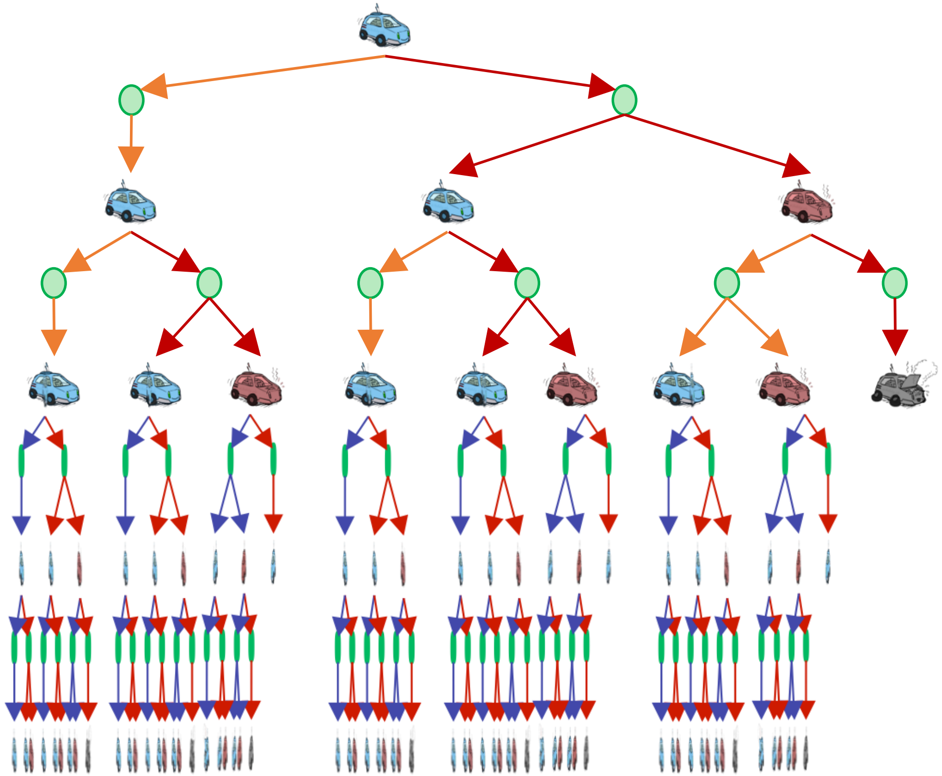
N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Stefan Pranger

- University Assistant at IAIK

- Research
  - Safe Learning in Probabilistic Environments
  - Tool: TEMPEST
  - https://tempest-synthesis.org/



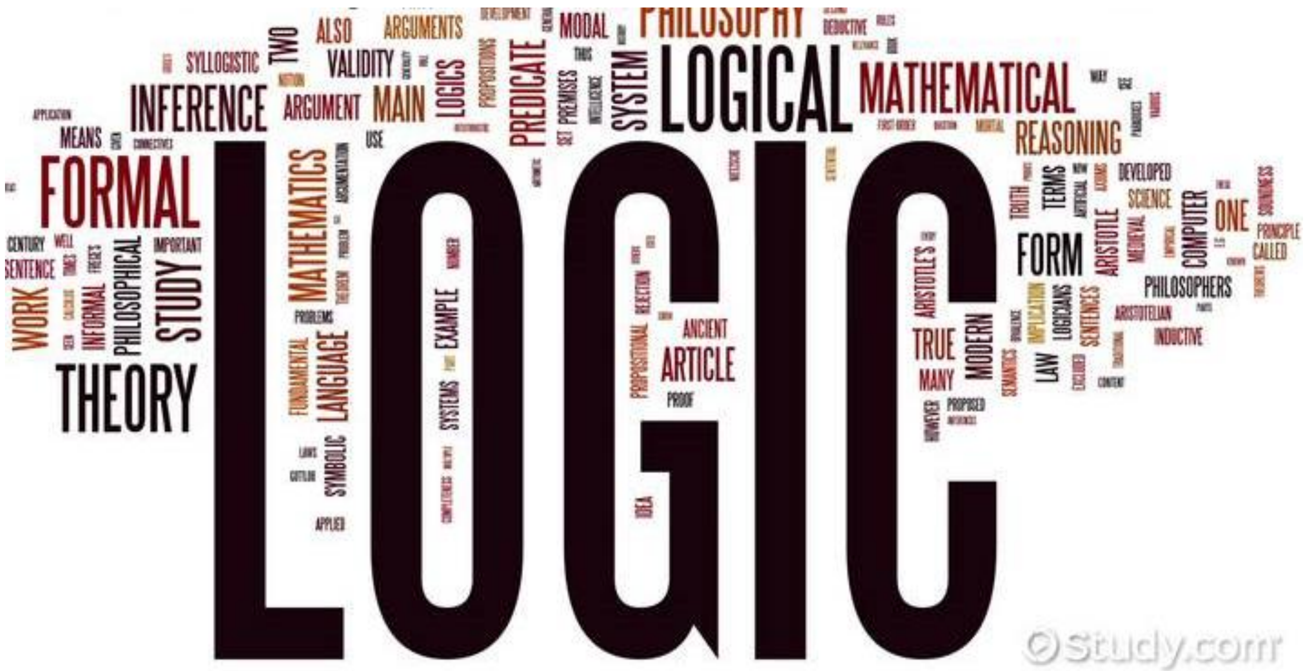


Stefan Pranger

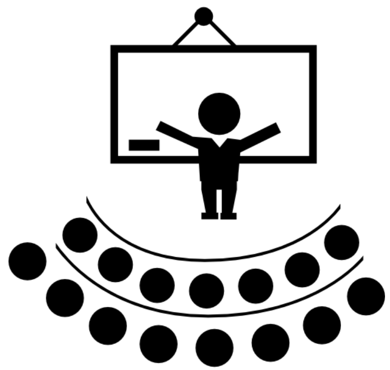
- University Assistant at IAIK
- Research
  - Safe Learning in Probabilistic Environments
  - Tool: TEMPEST
- Teaching
  - Logic and Computability
  - Model Checking
  - Bachelor thesis/master project/master thesis

Outline

- Team
- **Administrative Information**
  - **Lecture**
  - Practicals
- Outline
- Teaser



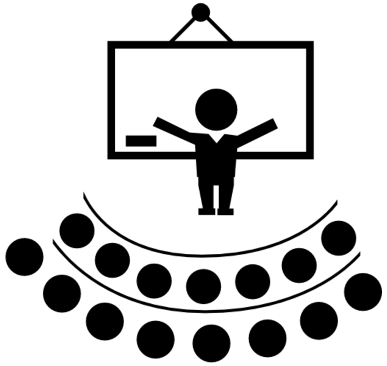
Lecture



- Typically: Monday 12:15pm-1:45pm, HS i12
- With exceptions!

Mo	07.10.2024	10:00	12:00	HS G (NT03128)
Mo	14.10.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	21.10.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Fr	08.11.2024	12:00	14:00	HS i7 (MD01168F)
Mo	11.11.2024	12:00	14:00	HS P2 (PHEG002)
Mo	18.11.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	25.11.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	02.12.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	09.12.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	16.12.2024	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	13.01.2025	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	20.01.2025	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)
Mo	27.01.2025	12:00	14:00	HS i12 "Dynatrace Hörsaal" (ICK1130H)

Lecture

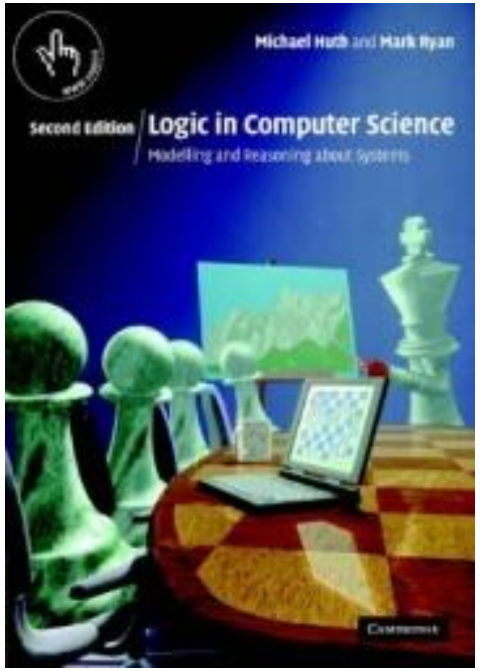


- Very interactive

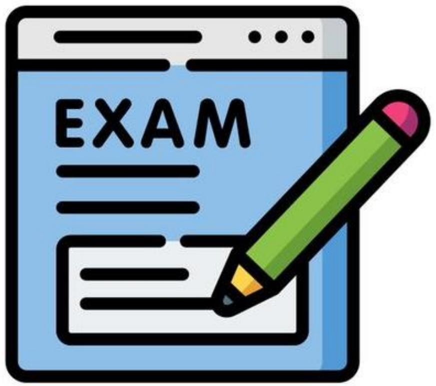
- Solve examples together
  - Bring pen and paper / tablet / coffee
  - Why:
    - Self-control
    - Apply new knowledge immediately

Material

- Course website
  - https://www.iaik.tugraz.at/lc
- Material
  - Slides
  - Lecture Recordings
  - Lecture notes
  - Questionnaire
  - Book
    - Huth and Ryan,
      Logic in Computer Science,
      Cambridge University Press, 2004

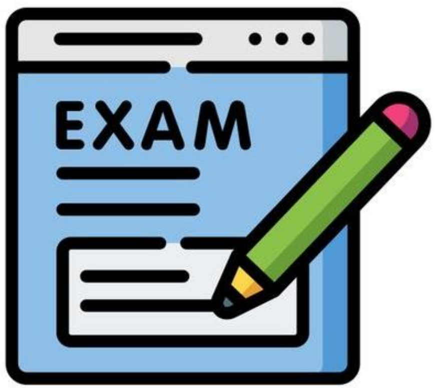


Exam



- Consists only of questions from **questionnaire**
- We will solve examples from questionnaire during class.
- Assignments 1-6 consist of questions from questionnaire.
- You prepare for the exam during
  - the lecture, and
  - the practicals.

Exam



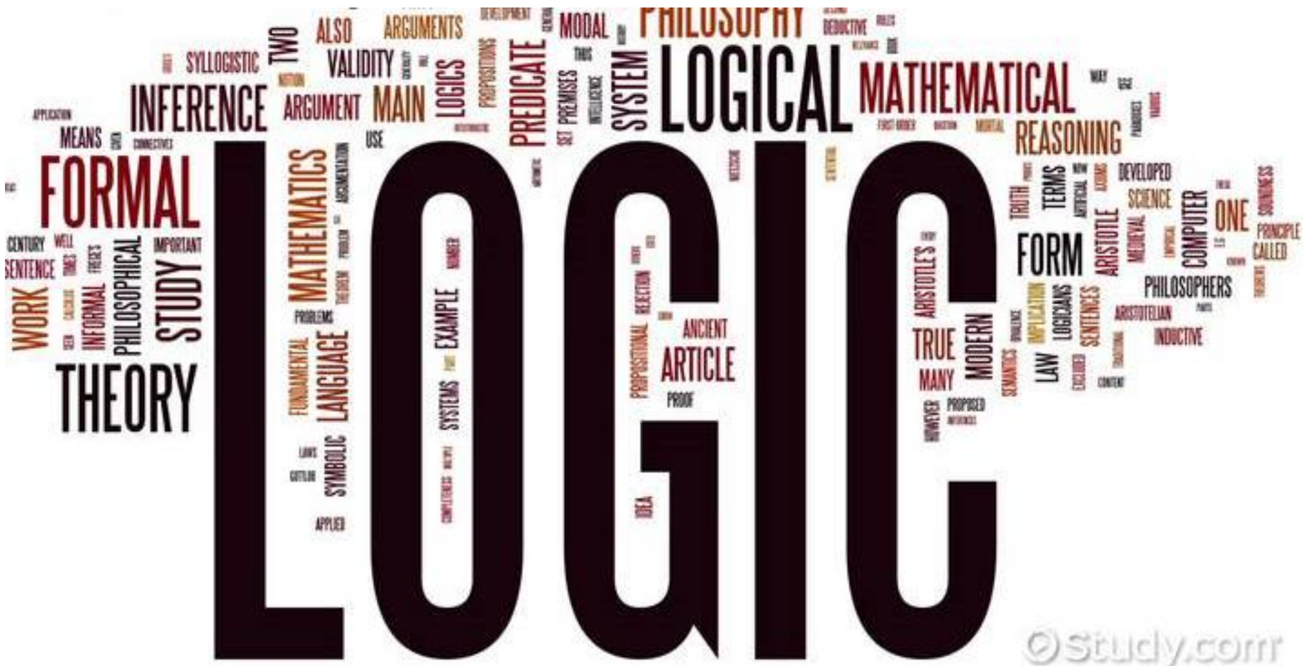
- Written exam at the end of the semester:
  - Friday, 31.01.2025

- Question hour (Training exam):
  - Monday, 27.01.2025

- **Voluntary training evening**
  - Wednesday, 29.01.2025 4pm - open end
  - Students can study for exam. We are there to help.

Outline

- Team
- Administrative Information
  - Lecture
  - **Practicals**
- Outline
- Teaser



Assignments

- 7 Assignments
  - 6 pen-and-paper assignment sheets
  - 1 programming assignment sheets

Number	Topic	Kick-Off	Deadline
1	Natural Deduction for Propositional Logic	2024-10-21	2024-10-27
2	SAT Solving	2024-11-08	2024-11-17
3	Binary Decision Diagrams	2024-11-18	2024-11-24
4	Predicate Logic	2024-11-25	2024-12-01
5	Natural Deduction for Predicate Logic	2024-12-02	2024-12-08
6	Satisfiability Modulo Theory	2025-01-13	2025-01-19
7	Programming Assignment (Z3)	TBA	2025-01-12

Assignments

- Assignment 1-6 – Pen & Paper
  - Tick via TeachCenter
  - Deadline: Sunday 11:59 pm
  - Present in class: Monday 3pm-4pm, or 4pm-5pm
- Assignment 7 – Programming
  - Groups of 2 students
  - Programming exercises handled via git
  - Individual interviews per group

Practical classes

- Students present solutions
- Inability to explain solution or completely wrong solutions lead to point deduction
  - Either 50% or 100% of assignment
  - Minor errors are OK!

Practical classes

- **Attendance is compulsory**
  - Discussion of Pen & Paper exercises

- If you are unable to attend (sickness)
  - E.g., Write an email bettina.koenighofer@iaik.tugraz.at
  - CC your tutor
  - Upload solutions in TeachCenter
  - Replacement interview 1 week later
    - Monday: 2pm, IAIK, Inffeldgasse 16a, 2nd floor

Grading

- Assignment 1-6: 13 points
- Assignment 7: 22 points

- If Points…
  - $\geq$ 87.5: (1) Sehr Gut / Excellent
  - $\geq$ 75.0: (2) Gut / Good
  - $\geq$ 62.5: (3) Befriedigend / Satisfactory
  - $\geq$ 50.0: (4) Genügend / Sufficient
  - $<$ 50.0: (5) Nicht Genügend / Insufficient

Communication

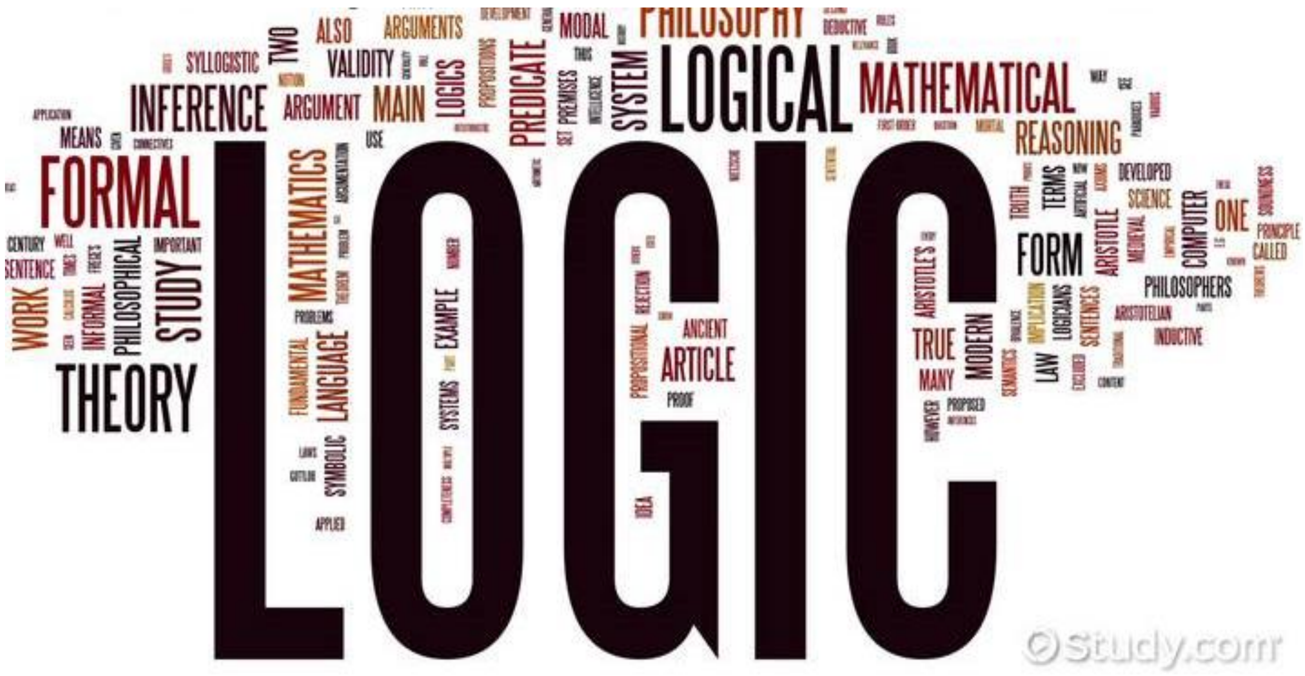
- Discord Server

- E-Mail
  - bettina.koenighofer@iaik.tugraz.at
  - stefan.pranger@iaik.tugraz.at

- Visit us at IAIK – Open door policy

Outline

- Team
- Administrative Information
  - Lecture
  - Practicals
- Outline
- Teaser



Time Line - Topics

Lectures 1 – 5:
Propositional Logic

Lectures 6-11:
Predicate Logic

Exam

Propositional Logic

October → November

- **Syntax & Semantic**
  - How do formulate problems
- **Algorithms to decide satisfiability**
  - Deciding propositional formulas with DPLL (with CDCL)
- **Data structures**
  - Binary Decision Diagrams (BDDs)
- **Natural deduction**
  - Perform proofs
- **Equivalence checking and normal forms**

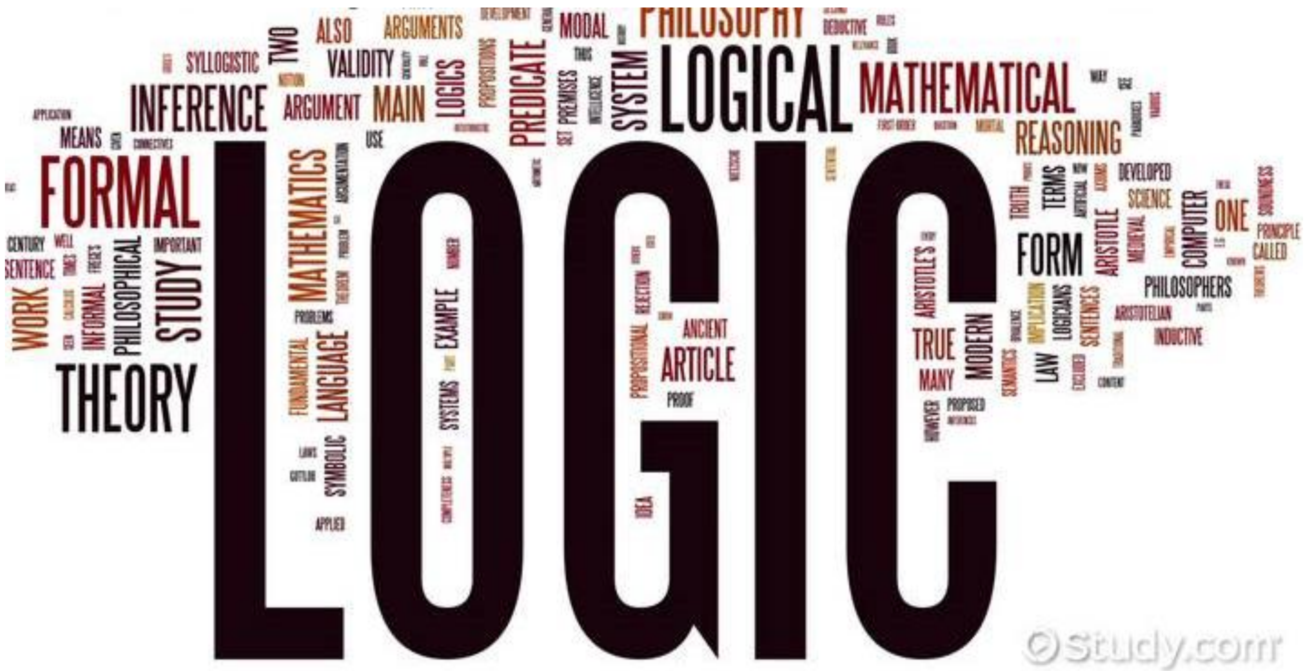
Predicate Logic

December → January

- **Syntax & Semantic**
- **Natural deduction**
  - Perform proofs
- **Satisfiability Modulo Theory (SMT)**
  - Formulas in predicate logic with theories
- **Algorithms to decide satisfiability**
  - Deciding SMT formulas (Eager encoding and DPLL(T))
- **SMT in Practice - Z3**

Outline

- Team
- Administrative Information
  - Lecture
  - Practicals
- Outline
- Teaser



Translate Sentences to Formulas

- *“I like Fridays and I don’t like Mondays.”*

Translate Sentences to Formulas

- *"I like Fridays and I don't like Mondays."*

Sentence that can be true or false

p… I like Fridays

Sentence that can be true or false

q… I like Mondays

$$p \land \neg q$$

Logical Operators

$\land \cdots AND$
$\lor \ \ldots OR$
$\neg \ldots NOT$
$\rightarrow \cdots IMPLICATION$

Translate the Sentences to Formulas

- *“If today is Friday, then tomorrow is Saturday.”*

- *“This lecture is exciting and not boring.”*

Logical Operators

$\wedge \cdots AND$
$\vee \ \ldots OR$
$\neg \ldots NOT$
$\rightarrow \cdots IMPLICATION$

Translate the Sentences to Formulas

- *"If today is Friday, then tomorrow is Saturday."*

  p... today is Friday, q… tomorrow is Saturday

  $$p \to q$$

- *"This lecture is exciting and not boring."*

  p... This lecture is exciting, q… This lecture is boring

  $$p \wedge \neg q$$

Logical Operators

$\wedge \cdots AND$
$\vee \ \ldots OR$
$\neg \ldots NOT$
$\to \cdots IMPLICATION$

Quiz – Translate the Sentences to Formulas

- You can fool some people sometimes.
- You can fool some of the people all the time.
- You can fool some people sometimes but you can‘t fool all the people all the time. [Bob Marley]
- You can fool some of the people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time. [Abraham Lincoln]

A Solution….

- You can fool some people sometimes.
- You can fool some of the people all the time.

A Solution….

$Fool(p, t)$ … returns True if you can fool person p at time t

$\exists x: \varphi$ … returns true if there exists an x that makes φ true
$\forall x: \varphi$ … returns true if forall x that makes φ true

- You can fool some people sometimes.

$$\exists p \in people\ \exists t \in time: Fool(p, t)$$

- You can fool some of the people all the time.

$$\exists p \in people\ \forall t \in time: Fool(p, t)$$

A Solution....

$Fool(p,t)$ … returns True if you can fool person p at time t

$\exists x{:}\,\varphi$ … returns true if there exists an x that makes φ true
$\forall x{:}\,\varphi$ … returns true if forall x that makes φ true

- You can fool some people sometimes but
  you can‘t fool all the people all the time. [Bob Marley]

A Solution....

$Fool(p,t)$ ... returns True if you can fool person p at time t

$\exists x: \varphi$ ... returns true if there exists an x that makes φ true
$\forall x: \varphi$ ... returns true if forall x that makes φ true

- You can fool some people sometimes but you can‘t fool all the people all the time. [Bob Marley]

$$(\exists p \in people\ \exists t \in time: Fool(p,t)) \land$$
$$\neg(\forall x \in people\ \forall t \in time: Fool(p,t))$$

A Solution....

$Fool(p, t)$ ... returns True if you can fool person p at time t

$\exists x: \varphi$ ... returns true if there exists an x that makes φ true
$\forall x: \varphi$ ... returns true if forall x that makes φ true

- You can fool some of the people all of the time,
  and all of the people some of the time,
  but you cannot fool all of the people all of the time.

A Solution….

$Fool(p,t)$ … returns True if you can fool person p at time t

$\exists x\colon \varphi$ … returns true if there exists an x that makes φ true
$\forall x\colon \varphi$ … returns true if forall x that makes φ true

- You can fool some of the people all of the time,
  and all of the people some of the time,
  but you cannot fool all of the people all of the time

$$(\exists p \in people\ \forall t \in time\colon Fool(p,t)) \wedge$$
$$\big(\forall p \in people\ \exists t \in time\colon Fool(p,t)\big) \wedge$$
$$\neg(\forall p \in people\ \forall t \in time\colon Fool(p,t))$$

A Solution….

$Fool(p,t)$ … returns True if you can fool person p at time t

$\exists x{:}\,\varphi$ … returns true if there exists an x that makes φ true
$\forall x{:}\,\varphi$ … returns true if forall x that makes φ true

Now you know some basics of predicate logic ☺

Quiz 2 - Translate the Sentences to Formulas

- *"Always, if there is a request, then there is a grant in the next step."*
- "$grant_1$ *and a* $grant_2$ *are never allowed simultaneously."*
- *"Always, a request will be granted in the next 3 time steps"*
- *"Any request will be granted eventually"*

Temporal Operators

$\mathrm{G} \dots Globally, Always$

$\mathrm{F} \dots Eventually$

$\mathrm{X} \dots Next$

Quiz 2 - Translate the Sentences to Formulas

- *"Always, if there is a request, then there is a grant in the next step."*

p... there is a request, q… there is a grant

$$G(p \rightarrow Xq)$$

Temporal Operators

G ... $Globally, Always$
F ... $Eventually$
X ... $Next$

Quiz 2 - Translate the Sentences to Formulas

- *“$grant_1$ and a $grant_2$ are never allowed simultaneously.”*

p... $grant_1$ is allowed, q… $grant_2$ is allowed

$$G\neg(p \wedge q)$$

Temporal Operators

G ... $Globally, Always$
F ... $Eventually$
X ... $Next$

Quiz 2 - Translate the Sentences to Formulas

- *"Always, a request has to be granted after exactly 3 time steps"*

p... there is a request, q… there is a grant

$$G(p \rightarrow XXXq)$$

Temporal Operators

$\mathrm{G} \ldots Globally, Always$

$\mathrm{F} \ldots Eventually$

$\mathrm{X} \ldots Next$

Quiz 2 - Translate the Sentences to Formulas

- *"Always, a request will be granted in the next 3 time steps"*

p... there is a request, q… there is a grant

$$G(p \rightarrow (q \vee Xq \vee XXq \vee XXXq))$$

Temporal Operators

G ... $Globally, Always$
F ... $Eventually$
X ... $Next$

Quiz 2 - Translate the Sentences to Formulas

- *"Any request is granted eventually"*

p... there is a request, q… there is a grant

$$G(p \rightarrow Fq)$$

Temporal Operators

G … $Globally, Always$
F … $Eventually$
X … $Next$

Quiz 2 - Translate the Sentences to Formulas

Now you know some basics of temporal logic ☺

Temporal Operators

$\mathrm{G} \ldots Globally, Always$

$\mathrm{F} \ldots Eventually$

$\mathrm{X} \ldots Next$

Teaser - SMT

- SMT solvers are **magic**!
- You describe your problem (with a bit of code), the solver finds the answer

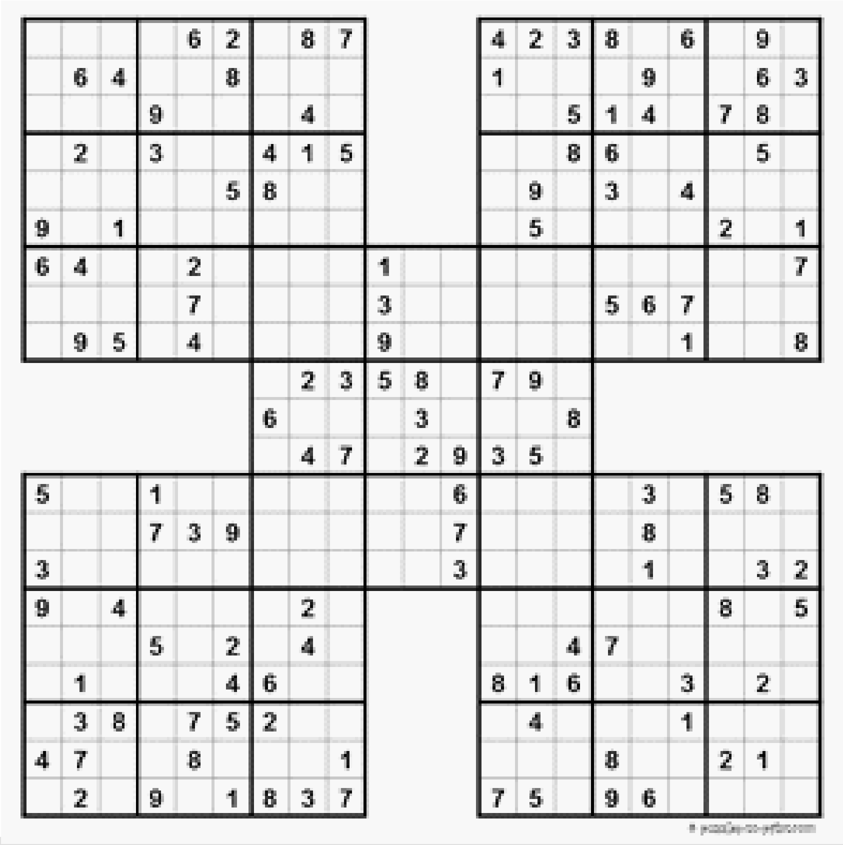
- **Example: Sudoku**
- Total number of possible assignments:
  - $2^{9\times9\times9} = 2^{729} = \mathbf{2.8 \times 10^{219}}$
  - ***How would you solve it?***

3			8		1			2
2		1		3		6		4
			2		4			
8		9				1		6
	6						5	
7		2				4		9
			5		9			
9		4		8		7		5
6			1		7			3

Teaser - SMT

- SMT solvers are **magic**!
- You describe your problem (with a bit of code), the solver finds the answer

- **Example: Samurai Sudoku**
  - *How would you solve it?*



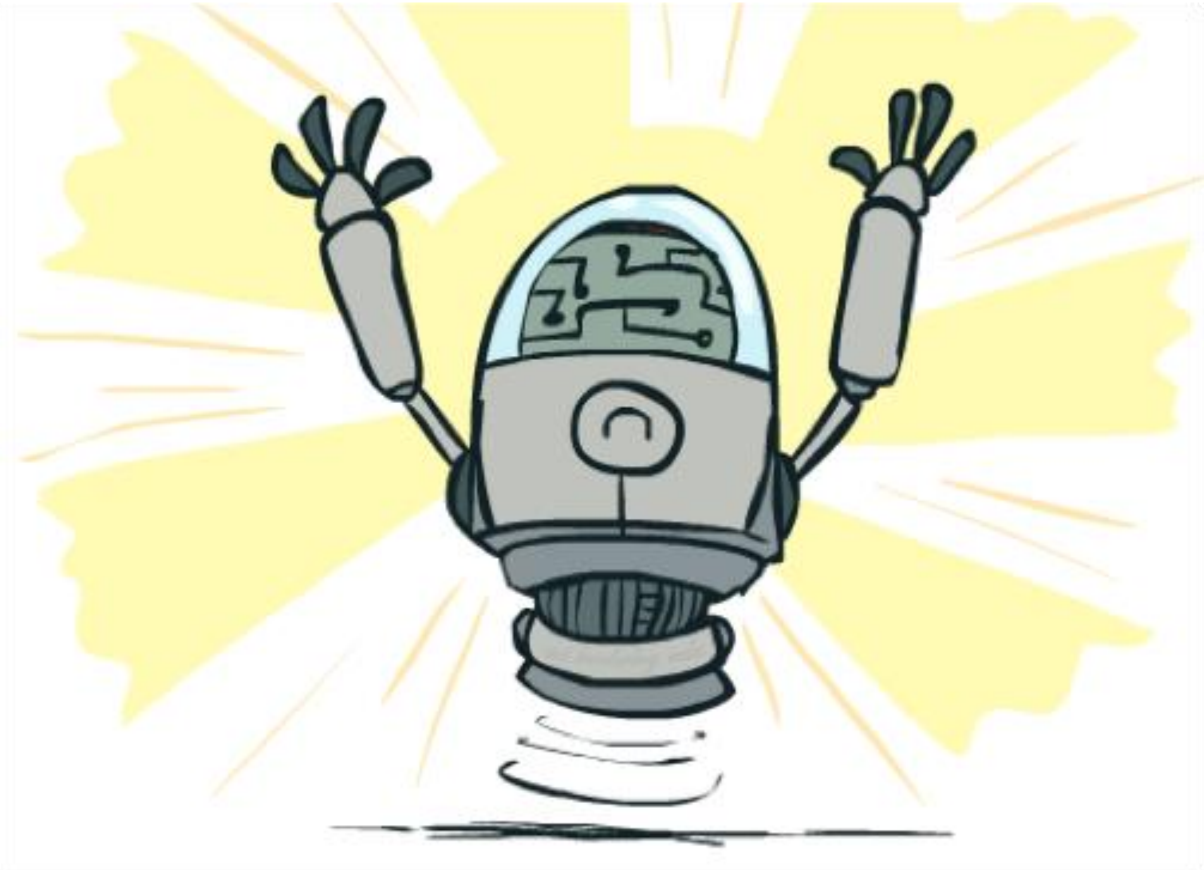
Teaser - SMT

- SMT solvers are **magic**!
- You describe your problem (with a bit of code), the solver finds the answer

- **Example: Sudoku**
- Total number of possible assignments:
  - $2^{9\times 9\times 9} = 2^{729} = 2.8 \times 10^{219}$
  - **Z3 solves a Sudoku in milliseconds without the need to write an algorithm**

3			8		1			2
2		1		3		6		4
			2		4			
8		9				1		6
	6						5	
7		2				4		9
			5		9			
9		4		8		7		5
6			1		7			3

Thank You!
Questions?



Discord