<table>
<thead>
<tr>
<th>Period</th>
<th>Course units</th>
</tr>
</thead>
</table>
| P3     | COMP60332 – Automated Reasoning and Verification  
An introduction to propositional and first-order reasoning, their theoretical basis, systems, and applications in verification |
| P4     | COMP62342 – Ontology Engineering for the Semantic Web  
An application of logic & automated reasoning to making knowledge intensive (web) application smarter |

- semester 2, periods P3/P4
- can be combined with any other theme
- has no pre-requisite themes
- core theme in Semantic Technologies, Data and Knowledge Management and Artificial Intelligence pathways
COMP60332: Automated Reasoning and Verification

Konstantin Korovin and Renate Schmidt

Theme: Ontology Engineering and Automated Reasoning
1. Why Automated Reasoning?
Reasoning is the main ingredient of any intellectual activity.

The main challenge: how to automate the reasoning process.
Reasoning is the main ingredient of any intellectual activity.

The main challenge: how to automate the reasoning process.
Automated Reasoning

- What is Reasoning? **Solving problems** by syntactic manipulations.

Software: Does your program accesses unallocated memory?

Math: Does this equation \((xy)^{-1} = y^{-1}x^{-1}\) hold in all groups?

Knowledge management:
Can we represent and analyse all available knowledge about human body?

Automated reasoning: can we solve all these problems automatically?
Automated Reasoning

- **What is Reasoning?** Solving problems by syntactic manipulations.
  - **Hardware:** Are these two hardware designs equivalent?

Software: Does your program accesses unallocated memory?

Math: Does this equation \((xy)^{-1} = y^{-1}x^{-1}\) hold in all groups?

Knowledge management:
Can we represent and analyse all available knowledge about human body?

Automated reasoning: can we solve all these problems automatically?
Automated Reasoning

- **What is Reasoning?** Solving problems by syntactic manipulations.
  - **Hardware:** Are these two hardware designs equivalent?
    
    ![RTL and SCH diagrams]
    
    **Software:** Does your program accesses unallocated memory?

  **Math:** Does this equation $(xy)^{-1} = y^{-1}x^{-1}$ hold in all groups?

  **Knowledge management:**
  Can we represent and analyse all available knowledge about human body?

  **Automated reasoning:** can we solve all these problems automatically?
What is Reasoning? Solving problems by syntactic manipulations.

Hardware: Are these two hardware designs equivalent?

Software: Does your program accesses unallocated memory?

Math: Does this equation \((xy)^{-1} = y^{-1}x^{-1}\) hold in all groups?

Knowledge management: Can we represent and analyse all available knowledge about human body?

Automated reasoning: can we solve all these problems automatically?
Automated Reasoning

- **What is Reasoning?** Solving problems by syntactic manipulations.
- **Hardware:** Are these two hardware designs equivalent?

Software: Does your program accesses unallocated memory?

Math: Does this equation \((xy)^{-1} = y^{-1}x^{-1}\) hold in all groups?

Knowledge management:
Can we represent and analyse all available knowledge about human body?

Automated reasoning: can we solve all these problems automatically?
Automated Reasoning

- **What is Reasoning?** Solving problems by syntactic manipulations.
  - **Hardware:** Are these two hardware designs equivalent?

![Diagram of hardware designs](image)

- **Software:** Does your program accesses unallocated memory?

- **Math:** Does this equation \((xy)^{-1} = y^{-1}x^{-1}\) hold in all groups?

- **Knowledge management:** Can we represent and analyse all available knowledge about human body?

- **Automated reasoning:** can we solve all these problems automatically?
The scale of automation required

Intel floating arithmetic bug cost $475 million.

Software bugs cost billions.

Intel i7 Haswell-E 2,600,000,000 gates

40 Mil LOC

Major companies: Intel, Microsoft, Airbus, NASA intensively use formal methods.
In mathematics

Erdős discrepancy problem proved by a SAT solver (2014):

12GB proof
In mathematics

Erdős discrepancy problem proved by a SAT solver (2014):

12GB proof

Largest math proof ever: Pythagorean triples problem was proved by a SAT solver (2016)

200TB proof
Applications of automated reasoning

Applications:

- **software and hardware verification:**
  - Intel, Microsoft
- **information management:**
  - biomedical ontologies,
  - semantic Web, databases
- **combinatorial reasoning:**
  - constraint satisfaction, planning, scheduling
- **Internet security**
- **Theorem proving** in mathematics

“It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the past.” McCarthy, 1963.
Manchester: world leading in logic and reasoning

- **Theory:**
  - first-order reasoning
  - resolution, superposition, instantiation, tableaux, linear arithmetic
  - ontology reasoning

- **Applications:**
  - software/hardware verification
  - semantic Web, bio-health
  - multi-agent systems

- **Reasoning systems developed in our School:**
  - **iProver** – an instantiation-based reasoner for first-order logic, won major of awards at CASC championships.
  - **Vampire** – a superposition-based reasoner for first-order logic, won major awards at CASC championships.
  - **MSPASS** – a resolution/superposition based reasoner SPASS extended with reasoning with modal logics.
  - **Fact++** an ontology reasoner: OWL DL.
  - **Pellet** an ontology reasoner: OWL DL.
Manchester: world leading in logic and reasoning

- Theory:
  - first-order reasoning
  - resolution, superposition, instantiation, tableaux, linear arithmetic
  - ontology reasoning

- Applications:
  - software/hardware verification
  - semantic Web, bio-health
  - multi-agent systems

- Reasoning systems developed in our School:
  - iProver – an instantiation-based reasoner for first-order logic won major of awards at CASC championships.
  - Vampire – a superposition-based reasoner for first-order logic, won major awards at CASC championships.
  - MSPASS – a resolution/superposition based reasoner SPASS extended with reasoning with modal logics.
  - Fact++ an ontology reasoner: OWL DL.
  - Pellet an ontology reasoner: OWL DL.
Manchester: world leading in logic and reasoning

- **Theory:**
  - first-order reasoning
  - resolution, superposition, instantiation, tableaux, linear arithmetic
  - ontology reasoning

- **Applications:**
  - software/hardware verification
  - semantic Web, bio-health
  - multi-agent systems

- **Reasoning systems developed in our School:**
  - **iProver** – an instantiation-based reasoner for first-order logic
    won major of awards at CASC championships.
  - **Vampire** – a superposition-based reasoner for first-order logic,
    won major awards at CASC championships.
  - **MSPASS** – a resolution/superposition based reasoner SPASS extended with reasoning with modal logics.
  - **Fact++** an ontology reasoner: OWL DL.
  - **Pellet** an ontology reasoner: OWL DL.
This course is focused on **efficient automated reasoning**.

**Syllabus:**

- **Propositional logic:** syntax, semantics, CNF transformation
- **Propositional reasoning:** DPLL algorithm: unit propagation, backjumping, lemma learning
- **First-order logic:** syntax, semantics, well-founded induction, Skolemization
- **First-order reasoning:** resolution, Bachmair-Ganzinger model construction, completeness, redundancy elimination
- **Applications:** verification of transition systems, LTL, bounded and unbounded model checking
- What is the theory behind efficient SAT solvers?
- How to prove any mathematical theorem using only two rules?
- What is inside a theorem prover?
- How to verify a state transition system?
**COMP60332 – Automated Reasoning and Verification**

This course is focused on **efficient automated reasoning**.

**Syllabus:**

- **Propositional logic**: syntax, semantics, CNF transformation
- **Propositional reasoning**: DPLL algorithm: unit propagation, backjumping, lemma learning

- **First-order logic**: syntax, semantics, well-founded induction, Skolemization
- **First-order reasoning**: resolution, Bachmair-Ganzinger model construction, completeness, redundancy elimination

- **Applications**: verification of transition systems, LTL, bounded and unbounded model checking

- What is the theory behind efficient SAT solvers?
- How to prove any mathematical theorem using only two rules?
- What is inside a theorem prover?
- How to verify a state transition system?
This course is focused on **efficient automated reasoning**.

**Syllabus:**

- **Propositional logic**: syntax, semantics, CNF transformation
- **Propositional reasoning**: DPLL algorithm: unit propagation, backjumping, lemma learning
- **First-order logic**: syntax, semantics, well-founded induction, Skolemization
- **First-order reasoning**: resolution, Bachmair-Ganzinger model construction, completeness, redundancy elimination
- **Applications**: verification of transition systems, LTL, bounded and unbounded model checking
- What is the theory behind efficient SAT solvers?
- How to prove any mathematical theorem using only two rules?
- What is inside a theorem prover?
- How to verify a state transition system?
Assessment

This course is self-contained, no prerequisites but assumes that students are comfortable with basic mathematical notions.

Exam: 50%
Closed book, 2 hours, choose 3 out of 4 questions

Coursework and lab: 50%
Assessed and unassessed exercises: pen and paper
Labwork involving
- SAT solvers
- first-order reasoning systems

Questions? please email:
Konstantin Korovin (room 2.40) Renate Schmidt (room 2.42)
korovin@cs.man.ac.uk schmidt@cs.man.ac.uk