DTU Course 02156 Logical Systems and Logic Programming (2021)

Week	Date	Main Topics (Prolog Programming in All Lessons)
35 #01	31/8	Course Prerequisites & Tutorial on Logical Systems and Logic Programming
36 #02	7/9	Chapter 1 - Introduction (Prolog Note)
37 #03	14/9	Chapter 2 - Propositional Logic: Formulas, Models, Tableaux
38 #04	21/9	Chapter 3 - Propositional Logic: Deductive Systems
39 #05	28/9	"Isabelle" - Propositional Logic: Sequent Calculus Verifier (SeCaV)
40 #06	5/10	Chapter 4 - Propositional Logic: Resolution
41 #07	12/10	Chapter 7 - First-Order Logic: Formulas, Models, Tableaux
42		(Autumn Vacation)
43 #08	26/10	Chapter 8 - First-Order Logic: Deductive Systems
44 #09	2/11	"Isabelle" - First-Order Logic: Sequent Calculus Verifier (SeCaV)
45 #10	9/11	Chapter 9 - First-Order Logic: Terms and Normal Forms
46 #11	16/11	Chapter 10 - First-Order Logic: Resolution
47 #12	23/11	Chapter 11 - First-Order Logic: Logic Programming
48 #13	30/11	Chapter 12 - First-Order Logic: Undecidability and Model Theory & Course Evaluation

Responsible: Associate Professor Jørgen Villadsen <jovi@dtu.dk>

Assignments & Exam **MUST BE SOLVED INDIVIDUALLY**

Assignment-1 Deadline Sunday 26/9 (Available Wednesday 15/9)

Assignment-2 Deadline Sunday 10/10 (Available Wednesday 29/9)

Assignment-3 Deadline Sunday 31/10 (Available Wednesday 13/10)

Assignment-4 Deadline Sunday 14/11 (Available Wednesday 3/11)

Assignment-5 Deadline Thursday 2/12 (Available Wednesday 17/11)

Written Exam Tuesday 14/12 (2 Hours / No Computer / All Notes Allowed)

The mandatory assignments and the written exam are evaluated as a whole – even if you do well in the mandatory assignments then you still must do decent in the written exam in order to pass the course!

A TEACHER MUST IMMEDIATELY REPORT ANY SUSPICION OF CHEATING TO THE STUDY ADMINISTRATION FOR FURTHER ACTIONS

SeCaV Sequent Calculus Verifier

SeCaV formalizes first-order logic with constants and functions

SeCaV verifies one-sided sequent calculus proofs

SeCaV uses the Isabelle proof assistant

SeCaV is a tool for teaching logic

Jørgen Villadsen

Asta Halkjær From

Alexander Birch Jensen

Anders Schlichtkrull

Agenda — Week #5

Motivation — Logic — Is it important...???

Isabelle Proof Assistant

Sequent Calculus Verifier (SeCaV)

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ "Hilbert's program" to show the consistency of mathematics

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ "Hilbert's program" to show the consistency of mathematics

Bertrand Russell 1872-1970

⋆ Simple type theory as first foundation of mathematics

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ "Hilbert's program" to show the consistency of mathematics

Bertrand Russell 1872-1970

⋆ Simple type theory as first foundation of mathematics

Kurt Gödel 1906-1978

⋆ Greatest logician – Incompleteness – Paranoid and refused to eat

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ “Hilbert's program” to show the consistency of mathematics

Bertrand Russell 1872-1970

⋆ Simple type theory as first foundation of mathematics

Kurt Gödel 1906-1978

⋆ Greatest logician – Incompleteness – Paranoid and refused to eat

Jacques Herbrand 1908-1931

⋆ Automated theorem proving – Died while mountain-climbing

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ "Hilbert's program" to show the consistency of mathematics

Bertrand Russell 1872-1970

⋆ Simple type theory as first foundation of mathematics

Kurt Gödel 1906-1978

⋆ Greatest logician – Incompleteness – Paranoid and refused to eat

Jacques Herbrand 1908-1931

⋆ Automated theorem proving – Died while mountain-climbing

Gerhard Gentzen 1909-1945

⋆ Starved to death in prison when arrested due to his Nazi loyalties

Motivation — Logical Systems & Great Logicians

Gottlob Frege 1848-1925

⋆ Predicate logic and naive set theory as first symbolic logics

David Hilbert 1862-1943

⋆ "Hilbert's program" to show the consistency of mathematics

Bertrand Russell 1872-1970

⋆ Simple type theory as first foundation of mathematics

Kurt Gödel 1906-1978

⋆ Greatest logician – Incompleteness – Paranoid and refused to eat

Jacques Herbrand 1908-1931

⋆ Automated theorem proving – Died while mountain-climbing

Gerhard Gentzen 1909-1945

⋆ Starved to death in prison when arrested due to his Nazi loyalties

Alan Turing 1912-1954

⋆ Computability – Homosexual – Ate apple with cyanide – Suicide?

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

- ICLP: International Conference on Logic Programming

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

- ICLP: International Conference on Logic Programming
- TABLEAUX: Analytic Tableaux and Related Methods

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

- ICLP: International Conference on Logic Programming
- TABLEAUX: Analytic Tableaux and Related Methods
- UNIF: International Workshop on Unification

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

- ICLP: International Conference on Logic Programming
- TABLEAUX: Analytic Tableaux and Related Methods
- UNIF: International Workshop on Unification

Oxford, UK, 6-19 July 2018

http://www.floc2018.org/

Motivation — Federated Logic Conference (FLoC)

FLoC is a large research event in computer science (every 4 years)

1996 New York, USA	1999 Trento, Italy	2002 Copenhagen
2006 Seattle, USA	2010 Edinburgh, UK	2014 Vienna, Austria

Logicians: Informatics, Mathematics, Linguistics, Philosophy, etc.

- ICLP: International Conference on Logic Programming
- TABLEAUX: Analytic Tableaux and Related Methods
- UNIF: International Workshop on Unification

Oxford, UK, 6-19 July 2018

`http://www.floc2018.org/`

More than 2000 participants...

Isabelle Proof Assistant

Isabelle is a generic proof assistant http://isabelle.in.tum.de/

The Archive of Formal Proofs is a collection of proof libraries, examples, and larger scientific developments, mechanically checked in the theorem prover Isabelle http://www.isa-afp.org/

The world's first operating-system kernel with an end-to-end proof of implementation correctness and security enforcement is now open source http://sel4.systems/

480000 lines of Isabelle source files = 9 Gutenberg bibles

Tableaux are used...

```
theory Scratch imports Main begin

theorem "p \<or> \<not>p" by blast

end
```

Wikipedia

Sequent calculus is, in essence, a style of formal logical argumentation where every line of a proof is a conditional tautology (called a sequent by Gerhard Gentzen) instead of an unconditional tautology.

Wikipedia

Sequent calculus is, in essence, a style of formal logical argumentation where every line of a proof is a conditional tautology (called a sequent by Gerhard Gentzen) instead of an unconditional tautology.

Each conditional tautology is inferred from other conditional tautologies on earlier lines in a formal argument according to rules and procedures of inference, giving a better approximation to the style of natural deduction used by mathematicians than David Hilbert's earlier style of formal logic where every line was an unconditional tautology.

The Gentzen System — Sequent Calculus

Tableaux turned "tree upside down and signs reversed":

$\neg((p \lor q) \to (q \lor p))$

$p \lor q, \neg(q \lor p)$

$p \lor q, \neg q, \neg p$

L R

$p, \neg q, \neg p$

$\times$

R

$q, \neg q, \neg p$

$\times$

1.	$\vdash \neg p, q, p$	Axiom
2.	$\vdash \neg q, q, p$	Axiom
3.	$\vdash \neg(p \lor q), q, p$	$\beta\lor$, 1, 2
4.	$\vdash \neg(p \lor q), (q \lor p)$	$\alpha\lor$, 3
5.	$\vdash (p \lor q) \to (q \lor p)$	$\alpha \to$, 4

The Gentzen System — Sequent Calculus

Tableaux turned "tree upside down and signs reversed":

$\neg((p \vee q) \rightarrow (q \vee p))$
$p \vee q, \neg(q \vee p)$
$p \vee q, \neg q, \neg p$

$\overline{\text{L} \qquad \text{R}}$

$p, \neg q, \neg p$
$\times$

R

$q, \neg q, \neg p$
$\times$

1.	$\vdash \neg p, q, p$	Axiom
2.	$\vdash \neg q, q, p$	Axiom
3.	$\vdash \neg(p \vee q), q, p$	$\beta\vee$, 1, 2
4.	$\vdash \neg(p \vee q), (q \vee p)$	$\alpha\vee$, 3
5.	$\vdash (p \vee q) \rightarrow (q \vee p)$	$\alpha \rightarrow$, 4

Hence $\vdash (A \vee B) \rightarrow (B \vee A)$

The Gentzen System — Sequent Calculus

Tableaux turned "tree upside down and signs reversed":

$\neg((p \vee q) \rightarrow (q \vee p))$
$p \vee q, \neg(q \vee p)$
$p \vee q, \neg q, \neg p$

$\overline{\text{L} \qquad \text{R}}$

$p, \neg q, \neg p$
$\times$

R

$q, \neg q, \neg p$
$\times$

1.	$\vdash \neg p, q, p$	Axiom
2.	$\vdash \neg q, q, p$	Axiom
3.	$\vdash \neg(p \vee q), q, p$	$\beta\vee$, 1, 2
4.	$\vdash \neg(p \vee q), (q \vee p)$	$\alpha\vee$, 3
5.	$\vdash (p \vee q) \rightarrow (q \vee p)$	$\alpha \rightarrow$, 4

Hence $\vdash (A \vee B) \rightarrow (B \vee A)$

Soundness and Completeness Theorem: $\models A$ **iff** $\vdash A$

The Gentzen System — Sequent Calculus

Tableaux turned "tree upside down and signs reversed":

$\neg((p \vee q) \rightarrow (q \vee p))$
$p \vee q, \neg(q \vee p)$
$p \vee q, \neg q, \neg p$

$\overline{\text{L} \qquad \text{R}}$

$p, \neg q, \neg p$
$\times$

R

$q, \neg q, \neg p$
$\times$

1.	$\vdash \neg p, q, p$	Axiom
2.	$\vdash \neg q, q, p$	Axiom
3.	$\vdash \neg(p \vee q), q, p$	$\beta\vee$, 1, 2
4.	$\vdash \neg(p \vee q), (q \vee p)$	$\alpha\vee$, 3
5.	$\vdash (p \vee q) \rightarrow (q \vee p)$	$\alpha \rightarrow$, 4

Hence $\vdash (A \vee B) \rightarrow (B \vee A)$

Soundness and Completeness Theorem: $\models A$ **iff** $\vdash A$

Note: Use implicit set formation for conclusions in the system $\mathcal{G}$