Logic (PHIL 2080, COMP 2620, COMP 6262)
Chapter: Introduction to Logic

Pascal Bercher

Planning & Optimization

Yoshihiro Maruyama

Logic
Intelligent Agents

College of Engineering and Computer Science
the Australian National University (ANU)

23 & 25 February 2021

Australian National University

Organizational Matters

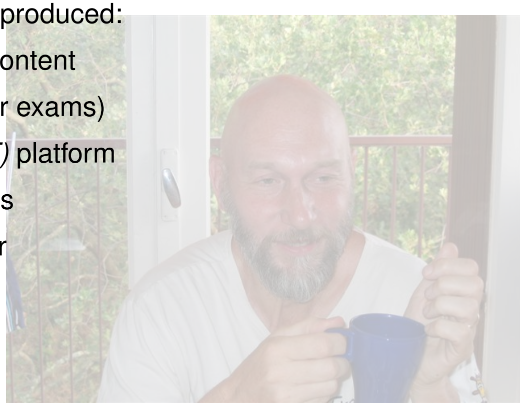
Team: The Lecturers, 1/3

Prof. Dr. John Slaney (2011 – 2020, now retired)

`http://users.cecs.anu.edu.au/~jks/`

We inherited his course; he produced:

- Course structure and content
- Most exercises (also for exams)
- The *Logic for Fun (L4F)* platform
- The online course notes
- The plagiarism scanner



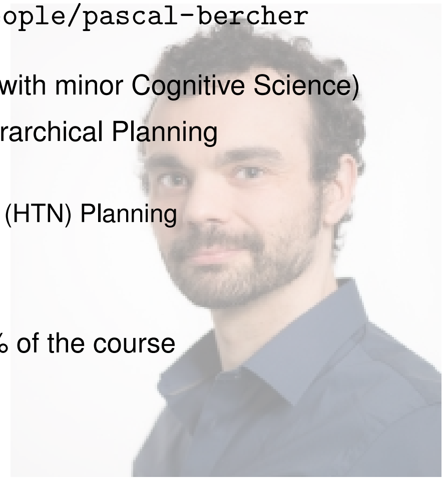
Team: The Lecturers, 2/3

Dr. Pascal Bercher since 2021

`https://cecs.anu.edu.au/people/pascal-bercher`

- *Studies:* Computer Science (with minor Cognitive Science)
- *PhD:* Computer Science: Hierarchical Planning
- *Research:*
  - Hierarchical Task Network (HTN) Planning
  - Heuristic Search
  - Complexity Theory

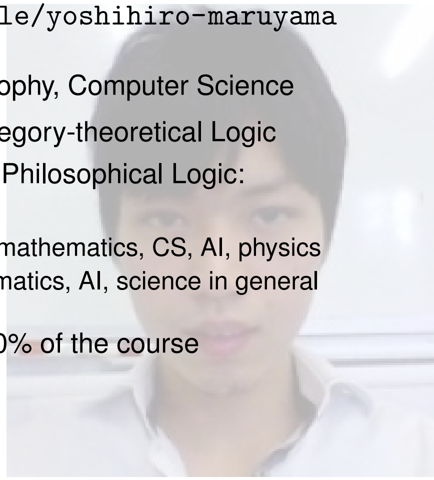
→ Pascal will teach the first 50% of the course



Team: The Lecturers, 3/3

Dr. Yoshihiro Maruyama since 2021

`https://cs.anu.edu.au/people/yoshihiro-maruyama`



- *Studies:* Mathematics, Philosophy, Computer Science
- *PhD:* Computer Science: Category-theoretical Logic
- *Research:* Mathematical and Philosophical Logic:
  - category-theoretical logic
  - categorical foundations of mathematics, CS, AI, physics
  - philosophy of logic, mathematics, AI, science in general

→ Yoshi will teach the second 50% of the course

Team: The Tutors

- See Wattle for the complete list and contact info.
- What do they do?
  - Give the tutorials/workshops
  - Answer your questions (via Wattle forum)
  - (Co-)Mark the homework, assignments, and exam

Various: Appointments/Dates

- Lectures:
  - Online live, recordings available via Wattle/Echo360
  - 2 per week, (approx.) 50 minutes each, see Wattle for dates
- Workshops/tutorials:
  - Only online via Zoom: please activate video and bring a headset
  - Once per week, 120 minutes each, see Wattle for dates
  - We'll do both tutorial-like "standard" exercises as well as workshop-like modeling tasks
- Appointments:
  - Only in exceptional cases. If required ask for appointment via mail
  - Otherwise ask all questions via the forum or your tutor

Various: Exercises and Exam

- Homework every two weeks:
  - Standard exercises (do proofs) or modeling tasks
  - Get corrected by tutors, marks are just FYI, they do *not* count towards the exam/course mark
  - Collaboration (up to 3 people) is strongly encouraged, but please don't hand in the same results several times
- Three Assignments:
  - 1 related to formal proofs, 1 via Logic for Fun (L4F) and 1 essay
  - Each assignment counts 15% of the final mark
  - Any form of cheating will be escalated and has serious consequences
  - Deadlines: Are strict, no exceptions (unless you have a *serious* reason, backed up by medical certificates, for example)
- Exam
  - Will be online, 3 hours
  - Counts 55% of final mark

Various: Course Material

- Slides (see Wattle)
- Online book "Logic Notes" (`http://users.cecs.anu.edu.au/~jks/LogicNotes/index.html`)
- Our modeling tool "Logic for Fun (L4F)" (`https://l4f.cecs.anu.edu.au/`)
- Online forum! (Set Wattle reminders accordingly!)
- For further reading, see books:
  - G. Restall. *Logic: An Introduction*. Ed. by J. Shand. Routledge, 2005 (Well-suited for Philosophy students)
  - D. van Dalen. *Logic and Structure*. Springer, 2012 (Well-suited for Computer Science and Mathematics students)

Various: Feedback/Corrections

- Nobody is perfect!
- Did you find an error in the slides? (Even just a typeo!)
- Do you have an idea on how to improve the slides?
  - More content? Less content?
  - Adding a specific example?
  - Adding a specific explanation?
  - Explaining a specific error many make?

→ Let us know! Drop us/the respective lecturer an email!

Introduction to Logic

Motivation: Motivations

See presentation by Yoshi!

Introduction: What is Logic?

- Logic is a mathematical discipline, and the basis of many related fields, most notably computer science.
- Logic is the Science of Reasoning:
  - Good/correct reasoning vs. bad/wrong reasoning
  - Making (and reasoning about) valid arguments
  - ⇒ See Monty Python sketch "argument clinic" (e.g., `https://www.dailymotion.com/video/x2hwqn9`)

Introduction: What's an argument?

Example:

- All footballers are bipeds
- Socrates it a footballer

} premises

- Thus, Socrates is a biped

} conclusion

→ This is a valid argument

Arguments consist of premises and a conclusion.

Introduction: What's an Argument?

Another Example:

- All cats are insects
- Snoopy is a cat

} premises

- Thus, Snoopy is an insect

} conclusion

→ This is also a valid argument!
  - Although everything was wrong!
  - All premises and the conclusion!

→ But we don't care, since it has a valid *form*. We exploit this form, and abstract from the content to reason about the conclusions.

Introduction: What's an Argument?

Our final Example:

- All logicians are rational
- Restall[1] is rational

} premises

- Thus, Restall is a logician

} conclusion

→ Interestingly, this is an invalid (wrong!) argument!
  - Although everything was right!
  - All premises and the conclusion!

→ *Wrong form*: The conclusion did not *follow* from the premises.

[1] Greg Restall, professor of logic at the University of Melbourne, author of the best-known book on substructural logic and editor in chief of the Australasian Journal of Logic, is presumably a logician if anyone is.

Introduction: Forms of Arguments

Valid arguments have, e.g., the following form:

- All *A*s are *B*s;
- *x* is an *A*;
- Therefore, *x* is a *B*.

The example with Restall did not work because it used a wrong form:

- All *A*s are *B*s;
- *x* is an *B*;
- Therefore, *x* is an *A*.

Introduction: Valid Arguments

- An argument is considered valid, when ever the conclusion logically follows from the premises.
- “Logically follows” abstracts away from the number of “intermediate steps” that are required so that the conclusion becomes “obvious”.
- For example, if we take all axioms of some mathematical system as the premises and one of it’s (valid) theorems/propositions as its conclusion, this forms a valid argument – no matter how ingenious the theorem is!
- Thus, showing that an argument is actually valid is hard!
- We will break down arguments into a sequence of arguments, so that every conclusion “follows in one step” from its premises.
- This will be done via *natural deduction* (next chapter and couple of weeks!), based on *sequents*, which represent arguments.

Sequents

Introduction to Sequents

Our convention:

- Letters from the *end* of the alphabet: set
- . . . *beginning* . . . : single object of the kind that’s in the set

This represents sequent: $X : A$

- X is a set of formulae
- A is a single formula

This represents a “good” sequent: $X \vdash A$

- Read it: A follows (logically) from X
- For example, “the query A follows from the data X”

Definition of Sequents

- $\vdash$ is a relation, so how is it defined?
- When are we allowed to write $X \vdash A$ (instead of just $X : A$)?

When the following the properties hold:

- If A is in X, then $X \vdash A$ (reflexivity)
- If $X \vdash A$, then $X \cup Y \vdash A$ for all Y (monotonicity)
- If $X \vdash A$ and $Y, A \vdash B$, then $X, Y \vdash B$ (transitivity)
  - $X, A \vdash B$ is short for $X \cup \{A\} \vdash B$ and (*also called* cut)
  - $X, Y \vdash A$ is short for $X \cup Y \vdash A$

When these criteria hold, we also say it's a *consequence relation*.

The Purpose of Sequents

- Sequent are used to express *valid* arguments! (This will be formally covered in week 4.)
- Recall that an argument (in the sense of logic) is valid even if the "step" from the premises to the conclusion is far from obvious.
- To "prove" that a sequent is valid we will derive a sequence of sequents, by only "slightly" manipulating the involved formulae.
- More details will be given later when we study *natural deduction*.

Propositional Calculus

Introduction

Logic is about making statements: $\overbrace{\underbrace{\text{Socrates}}_{\text{constant}}\ \underbrace{\text{is a goat}}_{\text{predicate}}}^{\text{(natural language) sentence}}$

What's a predicate?

- It will formally be defined later, when we deal with *predicate logics*
- A predicate relates various objects (constants) to each other, e.g.:
  - isGoat(Socrates) or isGoat(Goat)
  - isFootballer(Socrates)
  - Kicks(Socrates,Goat) or Kicks(Socrates,Ball)
- Actually, predicates do not exist in *propositional logics*, where we have *propositions* instead, e.g.,
  - SocratesIsGoat or GoatIsGoat
  - SocratesIsFootballer
  - SocratesKicksGoat or SocratesKicksBall
  - $\rightarrow$ Since this is way too long, we usually just write p, q, r, etc.

Basic Definitions: Terminology

Atoms refer to any "atomic" truth statement:

- true (denoted by $\top$, T, or 1)
- false (denoted by $\bot$, F, or 0)
- any propositional symbol (denoted by $p, q, r, \ldots$)

What's missing for *non-atomic* statements? Connectives!

- Socrates is a goat, ...
  - because ...
  - although ...
  - until ...
  - and ...
  - or ...
- It is not true that ...
  - Socrates is a goat
  - ...

Basic Definitions: Syntax of Connectives

Which connectors do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$
- ... or ...: $\vee$ e.g., $(\bot \vee \top)$ or $(p \vee (q \wedge r))$
- if ..., then ...: $\rightarrow$ e.g., $(p \rightarrow q)$ or $((p \wedge q) \rightarrow (p \vee q))$
  also: ... implies ...
- ... if and only if ...: $\leftrightarrow$ e.g., $(p \leftrightarrow q)$ or $((p \wedge q) \leftrightarrow (q \wedge p))$
- not ...: $\neg$ e.g., $((\neg p) \rightarrow q)$ or $\neg(p \rightarrow q)$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics will be covered *formally* in a later section, so this is just a "teaser" for *some* of the connectives.
- Our connectives act like functions, assigning any input pattern a new output (just like logical gates in "technical computer science" / "computer architecture").
- Examples:
  - $(\neg p)$ inverts p's truth value: $\top$ is switched to $\bot$, and vice versa.
  - $(p \wedge q)$ is true if and only if both p and q are true.
  - $(p \vee q)$ is true if and only if at least one of p and q are true.

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics will be covered *formally* in a later section, so this is just a "teaser" for *some* of the connectives.
- Our connectives act like functions, assigning any input pattern a new output (just like logical gates in "technical computer science" / "computer architecture").
- Examples: (expressed as *truth tables*)

p	$\neg$
$\bot$	$\top$
$\top$	$\bot$

p	q	$\wedge$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

p	q	$\vee$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics will be covered *formally* in a later section, so this is just a "teaser" for *some* of the connectives.
- Our connectives act like functions, assigning any input pattern a new output (just like logical gates in "technical computer science" / "computer architecture").
- Examples: (expressed as *truth tables*)

p	$\neg$
0	1
1	0

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

Basic Definitions: Semantics of Connectives in *Natural Language*

Note that natural language does not always translate 1-to-1 to logics:

- "Jane and Jill went up the hill": says more than just WentUpHill(Jane) $\wedge$ WentUpHill(Jill), because it means that they went there *together*.
- "One false move and I will shoot!" Does not mean $Move(You) \wedge Shoot(I)$, but $Move(You) \rightarrow Shoot(I)$
- Funnily, "Don't move or I shoot": Is not meant to mean $\neg Move(You) \vee Shoot(I)$, but also means $Move(You) \rightarrow Shoot(I)$, but both are equivalent.

Modeling "the real world" (based on natural language) is not easy, and partially covered in some exercises of Logic For Fun.

Basic Definitions: Precedence of Connectives (Syntax Simplification)

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
- Second-Lowest: $\rightarrow$ e.g., $p \rightarrow \neg q \leftrightarrow r \equiv (p \rightarrow (\neg q)) \leftrightarrow r$
- Lowest: $\leftrightarrow$ e.g., $\neg p \vee q \leftrightarrow q \wedge r \equiv ((\neg p) \vee q) \leftrightarrow (q \wedge r)$

We reduce parenthesis to simplify and avoid confusion by exploiting:

- *precedence*, e.g., we write $(p \wedge \neg q \wedge r) \rightarrow (p \vee \neg q \vee r)$ instead of $(((p \wedge (\neg q)) \wedge r) \rightarrow (p \vee ((\neg q) \vee r)))$, and
- *associativity*, e.g., we write $p \wedge q \wedge r$ instead of $(p \wedge (q \wedge r))$

Connective Scopes and Main Connective: Connective Scopes

- Every connective has a *scope*.
- "[The scope of a connective] is defined to be the shortest formula or subformula in which that occurrence lies." (Logic Notes)
- Examples: In the formula $\neg(p \wedge q) \rightarrow ((p \vee r) \rightarrow \neg s)$
  - ... the scope of its first $\neg$ is $(p \wedge q)$
  - ... the scope of its second $\neg$ is s

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- "[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula." (Logic Notes)
- Examples: The main connective of ...
  - $\neg(p \land q) \to ((p \lor r) \to \neg s)$ is: the first $\to$
  - $(p \land q) \lor r$ is: $\lor$
  - What's the main connective of $(p \land q) \lor r \lor (q \to r)$? Recall that "$(p \land q) \lor r \lor (q \to r)$" is only syntactic sugar!
    - It was either $((p \land q) \lor r) \lor (q \to r)$ [then, it's the right $\lor$],
    - or it was $(p \land q) \lor (r \lor (q \to r))$ [then, it's the left $\lor$]
- Why is it important to identify the main connective? Because we will manipulate sequents based on their main connective!

Connective Scopes and Main Connective: Type of Formula

The main connective dictates the type of a formula:

- if main connective is $\neg$, formula is a *negation*
- ... $\land$, ... *conjunction*
- ... $\lor$, ... *disjunction*
- ... $\to$, ... *implication*
- ... $\leftrightarrow$, ... *double-implication*

Substitution: Substitutions of Formulae

What is a substitution?

- "Formula A is a substitution instance of formula B if and only if A results from B by substitution of formulas for sentence letters." (Logic Notes) – (Definitions is specific to *propositional* logic.)

Non-Example:

- "Original" formula: $q \lor q$
- The formula $(p \land q) \lor \neg r$ is *not a* substitution instance of it (because the left part had to be the same as the right)

Organizational Matters Introduction to Logic Sequents Propositional Calculus Summary

Substitution: Substitutions of Formulae

What is a substitution?

- "Formula A is a substitution instance of formula B if and only if A results from B by substitution of formulas for sentence letters." (Logic Notes) – (Definitions is specific to *propositional* logic.)

Australian National University Pascal Bercher and Yoshihiro Maruyama 32.34

Summary

Content of this Lecture

- Organizational Matters
- *Sequents*, as a way to express valid arguments
- Introduction to *Propositional Logic*
  - Its syntax and (a preview to) its semantics
  - What connectives exist (and which don't)
  - How to identify the type of a formula (e.g., negation, conjunction, disjunction, implication, double-implication)
  - What's a substitution

→ The entire Logic Notes section "Introduction"