Logics for Computation

Lecture #2: To Pee or not to Pee?

Carlos Areces and Patrick Blackburn

{carlos.areces,patrick.blackburn}@loria.fr

INRIA Nancy Grand Est
Nancy, France

ESSLLI 2008 - Hamburg - Germany

The Story So Far

- We introduced the concept of logic (indeed, of logics) from a broad semantic (or Tarskian, or model-theoretic) perspective.
- We introduced the key unifying notion of the course: relational structures.
- We introduced the three major themes that we are interested in:
  - Inference:
    - Model checking.
    - Satisfiability checking (a.k.a. Model building).
    - Validity checking.
  - Expressivity.
  - Computation.

What do we do Today?

- We introduce the simplest logic discussed in this course, one you are probably familiar with: Propositional Logic (PL).
- We introduce/review the language, the familiar truth functional semantics, and solve a simple problem with PL.
- We then discuss inference in more detail: we see that model checking in PL is (very) easy. Moreover, we see that models can be elegantly built using tableaux method, hence we solve satisfiability checking task too.
- We then show that any method for solving the satisfiability task also solves the validity task — hence our tableaux method solves the validity task too.
- We conclude with brief remarks on expressivity and computability.

Propositional Logic: Syntax

The language of propositional logic is simple. We have the following basic symbols:

Propositional symbols:	$p, q, r, p_1, p_2, p_3, \ldots$
Logical symbols:	$\top, \bot, \neg, \vee, \wedge, \rightarrow, \leftrightarrow$
Grouping symbols:	(,)

PL is sometimes called Boolean logic (after the pioneering English logician George Boole) and the symbols $\top$, $\bot$, $\neg$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ are often called Boolean connectives.

Propositional Logic: Syntax

We then say that $\top$, $\bot$ and any propositional symbol are formulas (or well-formed formulas, or wff). These single-symbol formulas are often called atomic formulas.

We then construct complex formulas (or compound formulas) in accordance with the following recursive definition:

- If ϕ and ψ are formulas then so are $\neg\psi$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$, and $(\varphi \leftrightarrow \psi)$.
- Nothing else is a formula.

That's the official syntax — but we often simplify the bracketing. For example we would typically write $((p \wedge q) \rightarrow r)$ as $(p \wedge q) \rightarrow r$.

Propositional Logic: Semantics

- The semantics is also straightforward. A model for this language is simply an assignment V of true (T) or false (F) to the propositional symbols. So this is a very simple conception of model; sometimes V is called an assignment.
- Thus the models for PL are not relational structures. Or at least, so it seems. Actually, there is a natural way to view PL in terms of relational structures — but that can wait.

Propositional Logic: Semantics

So: V determines the truth values of the propositional formulas. We then determine whether other formulas are true or false with respect to V by using the following rules. Note: iff is short for if and only if:

$\top$		is always true
$\bot$		is always false
$\neg\varphi$ is true	iff	φ is false
$\varphi \vee \psi$ is true	iff	either φ or ψ (or both) are
$\varphi \wedge \psi$ is true	iff	both φ and ψ are
$\varphi \rightarrow \psi$ is true	iff	either φ is false or ψ is true
$\varphi \leftrightarrow \psi$ is true	iff	φ and ψ are both true, or they are both false

Truth Tables

A couple of remarks. First, you may may have seen the previous truth definition in the guise of a "truth table".

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	T	T
F	T	T	T	F	F	F
T	F	F	T	F	T	F
F	F	T	F	F	T	T

Truth tables are a conceptually simple (if tedious) way of working with PL.

We don't need all these connectives

Secondly, as many of you will know, many connectives can be defined in terms of others. We don't need them all.

- $\bot$ is $\neg\top$
- $\top$ is $p \vee \neg p$.
- $\varphi \rightarrow \psi$ is $\neg\varphi \vee \psi$.
- $\varphi \wedge \psi$ is $\neg(\neg\varphi \vee \neg\psi)$.
- $\varphi \vee \psi$ is $\neg(\neg\varphi \wedge \neg\psi)$.

A set of logical symbols that can define all the others is called truth functionally complete. For example, $\{\neg, \wedge\}$, $\{\neg, \vee\}$, and $\{\rightarrow, \bot\}$ are truth functionally complete sets. The set $\{\vee, \wedge\}$ is not.

Fundamental Semantic Concepts

And now we come to the key semantic concepts that we mentioned in the previous lecture:

- If a model V makes a formula φ true we say V satisfies φ, or φ is true in V, and write $V \models \varphi$.
- If it is possible to find some model V that makes φ true, then we say φ is satisfiable.
- If φ is true, no matter what what model V we use, then we say that φ is valid and write $\models \varphi$.

A Diplomatic Problem

A Diplomatic Problem

You are chief of protocol for the embassy ball. The crown prince instructs you either to invite Peru or to exclude Qatar. The queen asks you to invite either Qatar or Romania or both. The king, in a spiteful mood, wants to snub either Romania or Peru or both. Who do you invite?

A Diplomatic Problem

You are chief of protocol for the embassy ball. The crown prince instructs you either to invite Peru or to exclude Qatar. The queen asks you to invite either Qatar or Romania or both. The king, in a spiteful mood, wants to snub either Romania or Peru or both. Who do you invite?

- Can we model this using just propositional logic?

A Diplomatic Problem

You are chief of protocol for the embassy ball. The crown prince instructs you either to invite Peru or to exclude Qatar. The queen asks you to invite either Qatar or Romania or both. The king, in a spiteful mood, wants to snub either Romania or Peru or both. Who do you invite?

- Can we model this using just propositional logic?
- And what do we gain by doing that?

A Diplomatic Problem

You are chief of protocol for the embassy ball. The crown prince instructs you either to invite Peru or to exclude Qatar. The queen asks you to invite either Qatar or Romania or both. The king, in a spiteful mood, wants to snub either Romania or Peru or both. Who do you invite?

- Can we model this using just propositional logic?
- And what do we gain by doing that?
- What kind of questions can we "ask" our model?

Formalizing the Diplomatic Problem

Formalizing the Diplomatic Problem

- Three propositional symbols

Formalizing the Diplomatic Problem

- Three propositional symbols

$$P \quad \equiv \quad \text{invite Peru}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$P \quad \equiv \quad \text{invite Peru} \qquad\qquad \neg P \quad \equiv \quad \text{exclude Peru}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcl@{\qquad}lcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & & & \end{array}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcl@{\qquad}lcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \end{array}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$
\begin{array}{lcl@{\qquad}lcl}
P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\
Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\
R & \equiv & \text{invite Romania} & & &
\end{array}
$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcl@{\qquad}lcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\ R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania} \end{array}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcllcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\ R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania} \end{array}$$

- The problem can be formalized as

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcllcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\ R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania} \end{array}$$

- The problem can be formalized as

prince: $P \vee \neg Q \quad \equiv \quad$ invite Peru or exclude Qatar

Formalizing the Diplomatic Problem

- Three propositional symbols

$$
\begin{array}{lcl@{\qquad}lcl}
P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\
Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\
R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania}
\end{array}
$$

- The problem can be formalized as

$$
\begin{array}{rcl}
\text{prince: } P \vee \neg Q & \equiv & \text{invite Peru or exclude Qatar} \\
\text{queen: } Q \vee R & \equiv & \text{invite Qatar or Romania or both}
\end{array}
$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$\begin{array}{lcl@{\qquad}lcl} P & \equiv & \text{invite Peru} & \neg P & \equiv & \text{exclude Peru} \\ Q & \equiv & \text{invite Qatar} & \neg Q & \equiv & \text{exclude Qatar} \\ R & \equiv & \text{invite Romania} & \neg R & \equiv & \text{exclude Romania} \end{array}$$

- The problem can be formalized as

$$\begin{array}{rcl} \text{prince: } P \vee \neg Q & \equiv & \text{invite Peru or exclude Qatar} \\ \text{queen: } Q \vee R & \equiv & \text{invite Qatar or Romania or both} \\ \text{king: } \neg R \vee \neg P & \equiv & \text{snub Romania or Peru or both} \end{array}$$

Formalizing the Diplomatic Problem

- Three propositional symbols

$$
\begin{array}{lclclcl}
P & \equiv & \text{invite Peru} & & \neg P & \equiv & \text{exclude Peru} \\
Q & \equiv & \text{invite Qatar} & & \neg Q & \equiv & \text{exclude Qatar} \\
R & \equiv & \text{invite Romania} & & \neg R & \equiv & \text{exclude Romania}
\end{array}
$$

- The problem can be formalized as

$$
\begin{array}{rcl}
\text{prince: } P \vee \neg Q & \equiv & \text{invite Peru or exclude Qatar} \\
\text{queen: } Q \vee R & \equiv & \text{invite Qatar or Romania or both} \\
\text{king: } \neg R \vee \neg P & \equiv & \text{snub Romania or Peru or both}
\end{array}
$$

- Let $\Sigma = (P \vee \neg Q)\ \&\ (Q \vee R)\ \&\ (\neg R \vee \neg P)$. Solving the problem amounts to seeing whether Σ has a model (that is, whether it is possible to make all three formulas in Σ simultaneously true).

Solving the Diplomatic Problem: informal reasoning

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R$$

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\frac{\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R}{P \vee R}$$

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\frac{\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R}{P \vee R}$$

- That is, one consequence of satisfying the prince and the queen is that we must invite Peru or Romania (or both).

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\frac{\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R}{P \vee R}$$

- That is, one consequence of satisfying the prince and the queen is that we must invite Peru or Romania (or both).
- So, is Σ satisfiable?

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\frac{\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R}{P \vee R}$$

- That is, one consequence of satisfying the prince and the queen is that we must invite Peru or Romania (or both).
- So, is Σ satisfiable? Yes, 2 out of 8 possible truth assignments satisfy Σ

$$\begin{array}{lll} P = \text{true} & Q = \text{true} & R = \text{false} \\ P = \text{false} & Q = \text{false} & R = \text{true} \end{array}$$

Solving the Diplomatic Problem: informal reasoning

- What can we deduce from Σ?

$$\frac{\text{prince: } P \vee \neg Q \qquad \text{queen: } Q \vee R}{P \vee R}$$

- That is, one consequence of satisfying the prince and the queen is that we must invite Peru or Romania (or both).
- So, is Σ satisfiable? Yes, 2 out of 8 possible truth assignments satisfy Σ

$$\begin{array}{lll} P = \text{true} & Q = \text{true} & R = \text{false} \\ P = \text{false} & Q = \text{false} & R = \text{true} \end{array}$$

So either invite Peru and Qatar and not Romania
or invite Romania and not Peru and not Qatar

Solving the Diplomatic Problem 2: truth tables

Solving the Diplomatic Problem 2: truth tables

- Is there a way of computing this solution?

Solving the Diplomatic Problem 2: truth tables

- Is there a way of computing this solution?
- Yes. Use truth tables.

Solving the Diplomatic Problem 2: truth tables

- Is there a way of computing this solution?
- Yes. Use truth tables.

P	Q	R	$P \vee \neg Q$	$Q \vee R$	$\neg R \vee \neg P$	Σ

Solving the Diplomatic Problem 2: truth tables

- Is there a way of computing this solution?
- Yes. Use truth tables.

P	Q	R	$P \vee \neg Q$	$Q \vee R$	$\neg R \vee \neg P$	Σ
T	T	T	T	T	F	F
T	T	F	T	T	T	T
T	F	T	T	T	F	F
T	F	F	T	F	T	F
F	T	T	F	T	T	F
F	T	F	F	T	T	F
F	F	T	T	T	T	T
F	F	F	T	F	T	F

Solving the Diplomatic Problem 2: truth tables

- Is there a way of computing this solution?
- Yes. Use truth tables.

P	Q	R	$P \vee \neg Q$	$Q \vee R$	$\neg R \vee \neg P$	Σ
T	T	T	T	T	F	F
T	T	F	T	T	T	T
T	F	T	T	T	F	F
T	F	F	T	F	T	F
F	T	T	F	T	T	F
F	T	F	F	T	T	F
F	F	T	T	T	T	T
F	F	F	T	F	T	F

- This works — but it's about as exciting as watching paint dry. And may take considerably longer; truth tables are 2^n in the number of propositional symbols. There could be a lot of rice on the chessboard before we're finished ...

Inference in PL

So it's time to look at inference in PL more systematically. We'll look at all three tasks:

- Model checking
- Satisfiability building
- Validity checking

Inference in PL

So it's time to look at inference in PL more systematically. We'll look at all three tasks:

- Model checking (Easy! As in very!)
- Satisfiability building
- Validity checking

Inference in PL

So it's time to look at inference in PL more systematically. We'll look at all three tasks:

- Model checking (Easy! As in very!)
- Satisfiability building (We'll use tableaux to build models)
- Validity checking

Inference in PL

So it's time to look at inference in PL more systematically. We'll look at all three tasks:

- Model checking (Easy! As in very!)
- Satisfiability building (We'll use tableaux to build models)
- Validity checking (We'll see that tableaux can be used to determine validity)

Inference in PL

So it's time to look at inference in PL more systematically. We'll look at all three tasks:

- Model checking (Easy! As in very!)
- Satisfiability building (We'll use tableaux to build models)
- Validity checking (We'll see that tableaux can be used to determine validity)

Our discussion – and in particular, the tableaux method and the relationship between satisfiability checking and validity checking — will play an important role in subsequent lectures.

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

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

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

$$((p \wedge \neg q) \vee q) \rightarrow q$$
$$((\top \wedge \neg \bot) \vee \bot) \rightarrow \bot$$

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \land \neg q) \lor q) \to q$ is true in this model or not. We could just fill in the row of the truth table.

$$((p \land \neg q) \lor q) \to q$$
$$((\top \land \neg\bot) \lor \bot) \to \bot$$
$$((\top \land \top) \lor \bot) \to \bot$$

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

$$((p \wedge \neg q) \vee q) \rightarrow q$$
$$((\top \wedge \neg \bot) \vee \bot) \rightarrow \bot$$
$$((\top \wedge \top) \vee \bot) \rightarrow \bot$$
$$(\top \vee \bot) \rightarrow \bot$$

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

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

$$((\top \wedge \neg \bot) \vee \bot) \rightarrow \bot$$

$$((\top \wedge \top) \vee \bot) \rightarrow \bot$$

$$(\top \vee \bot) \rightarrow \bot$$

$$\top \rightarrow \bot$$

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

$$((p \wedge \neg q) \vee q) \rightarrow q$$
$$((\top \wedge \neg\bot) \vee \bot) \rightarrow \bot$$
$$((\top \wedge \top) \vee \bot) \rightarrow \bot$$
$$(\top \vee \bot) \rightarrow \bot$$
$$\top \rightarrow \bot$$
$$\bot$$

Model Checking

Suppose we are given a model V in which p is true and q is false, and we are asked to check whether $((p \wedge \neg q) \vee q) \rightarrow q$ is true in this model or not. We could just fill in the row of the truth table.

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

$$((\top \wedge \neg \bot) \vee \bot) \rightarrow \bot$$

$$((\top \wedge \top) \vee \bot) \rightarrow \bot$$

$$(\top \vee \bot) \rightarrow \bot$$

$$\top \rightarrow \bot$$

$$\bot$$

If you think of $\top$ as 1 and $\bot$ as 0 you can easily implement this in hardware (an idea that goes back to Shannon's Master's thesis).

So let's turn to satisfiability checking . . .

- We'll use tableaux to perform this task.
- A tableaux is essentially a tree-like data structure that records attempts to build a model.
- Tableaux are built by applying rules to an input formula. These rules systematically tear the formula to detect all possible ways of building a model.
- Each branch of a tableaux records one way of trying to build a model. Some branches ("closed branches") don't lead to models. Others branching ("open branches") do.
- The best way to learn is via an example. . .

Tableaux for PL

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

Tableaux for PI

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \ (\neg\neg)$$

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

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

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

$$\begin{array}{c}(\neg(p \wedge q) \wedge \neg\neg r) \wedge p \\ \neg(p \wedge q) \wedge \neg\neg r \\ p\end{array}$$

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

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

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

$$\begin{array}{c}
(\neg(p \wedge q) \wedge \neg\neg r) \wedge p \\
\neg(p \wedge q) \wedge \neg\neg r \\
p \\
\neg(p \wedge q) \\
\neg\neg r \\
r
\end{array}$$

Tableaux for Pl

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

$$\begin{array}{ccc}
 & (\neg(p \wedge q) \wedge \neg\neg r) \wedge p & \\
 & \neg(p \wedge q) \wedge \neg\neg r & \\
 & p & \\
 & \neg(p \wedge q) & \\
 & \neg\neg r & \\
 & r & \\
\neg p & & \neg q
\end{array}$$

Tableaux for PI

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

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

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

$$p$$

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

$$\neg\neg r$$

$$r$$

$\neg p$ $\qquad\qquad\qquad\qquad$ $\neg q$

Contradiction!!!

Tableaux for PL

Let's see if we can build a model for $(\neg(p \wedge q) \wedge \neg\neg r) \wedge p$.

Rules for $\neg$ and $\wedge$

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \ (\neg\neg)$$

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

$\neg p$ | $\neg q$

Contradiction!!! | *Model*

Satisfiability and Validity are Dual

- A formula φ is valid iff $\neg\varphi$ is not satisfiable.
- A consequence of this observation is: if we have a method for solving the satisfiability problem (that is, if we have an algorithm for building models) then we have a way of solving the validity problem.
- Why? Because: to test whether φ is valid, simply give $\neg\varphi$ to the algorithm for solving satisfiability. If it can't satisfy it, then φ is valid.
- Well, we have an algorithm for satisfiability (namely the tableaux method), so let's put this observation to work.

Validity via Tableaux

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Validity via Tableaux

Let's show that $(p \land q) \rightarrow p$ is valid

Rules for $\rightarrow$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

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

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi\\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

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

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

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

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi \\ \neg\psi\end{array}}\ (\neg \rightarrow)$$

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

$$p \wedge q$$

$$\neg p$$

$$p$$

$$q$$

Contradiction!!!

Validity via Tableaux

Let's show that $(p \wedge q) \rightarrow p$ is valid

Rules for $\rightarrow$

$$\frac{\varphi \rightarrow \psi}{\neg\varphi \qquad\qquad \psi}\ (\rightarrow)$$

$$\frac{\neg(\varphi \rightarrow \psi)}{\begin{array}{c}\varphi\\ \neg\psi\end{array}}\ (\neg\rightarrow)$$

$$\begin{array}{c}\neg((p \wedge q) \rightarrow p)\\ p \wedge q\\ \neg p\\ p\\ q\end{array}$$

Contradiction!!!

It is impossible to apply any more rules, and there are no open branches. Hence no model exists for the input $\neg\varphi$. Hence φ is valid.

Expressivity

- As we remarked earlier, it is possible to think about the semantics of PL in terms of relational structure.
- This gives us a way of comparing the expressivity of PL with the more powerful logics we shall study later.
- The idea is simple: think of PL as a way of talking about one element (!) relational structures of the form $\langle \{d\}, \{P_n\} \rangle$.
- That is, we have one individual, and one property for every propositional letter p_n (think of each P_n as a colour — we are covering the individual with coloured dots).
- This way of thinking about PL semantics is equivalent to the truth conditional semantics. Can you see why?
- That is, PL validity is completely determined by one element relational structures! Measured this way, its expressivity is low.

Computability

- We haven't directly said much about computability, but it should be clear that PL is a "computable logic".
- For a start, model checking is clearly computationally straightforward — it's linear in the length of the input formula.
- And checking satisfiability (and hence validity) is clearly computable too. The truth table method shows that we can do it in 2^n steps, where n is the number of propositional symbols in the input formal.
- Can we do better that 2^n steps? In particular, can the tableaux method do satisfiability/validity checking more efficiently.
- Sadly, it seems the answer is no.

What we Covered in this Lecture

- We introduced PL and dealt with all three inference tasks.
- Model checking was simple (linear-time algorithm) and satisfiability/validity checking turn out to be solvable using the same model building algorithm.
- We discussed a model building algorithm that looked more elegant than truth tables; it is more elegant, and often more efficient, but too turns out to share the same 2^n upper bound.

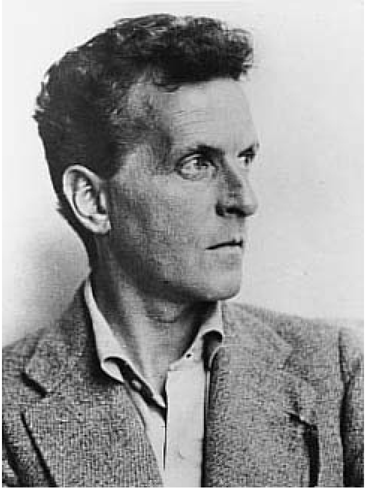
Relevant Bibliography

- One of the main founders of PL was George Boole (1815–1864), mathematician and philosopher.
- His book "An Investigation of the Laws of Thought" was one of the first mathematical treatments of logic, and one of the most important conceptual advances in logic from the Aristotelian syllogistic.

1815–1864

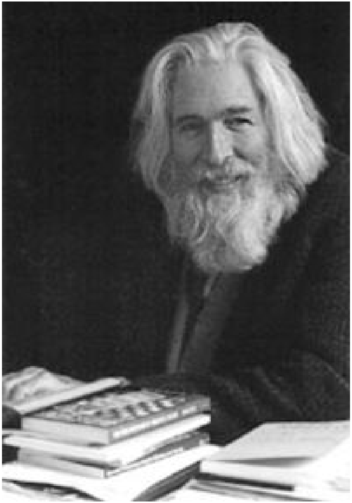
Relevant Bibliography

- The truth table method was pioneered by the philosopher Ludwig Wittgenstein (1889–1951) in his first famous philosophical work, the "Tractatus Logico-Philosophicus".
- He used the method in support of his celebrated "picture" theory of meaning.



Relevant Bibliography

- The tableaux method in the form presented here is due to Raymond Smullyan, logician, magician, and puzzle-supremo.
- His classic exposition of the method is in his book "First-Order Logic" (1968) , which remains one of the best technical expositions of the subject.
- A better book for beginners is Graham Priest's "An introduction to non-classical logic" (2001), Cambridge University Press.



The Next Lecture

How many Angels can Dance on the Head of a Pin?