Todays programme: Propositional Logic

- *Familiarity* with basic *terminology* of logics
  - Syntax, logical connectives
  - Semantics: models, truth, validity, logical consequence
  - Proof systems: deductions, deductive consequence, theorems
  - Soundness and completeness
- *Describe* propositional logic and some of its basic *properties*
  - Semantics: truth tables
  - Axiomatic proof system AL and its deductions
  - Soundness and completeness of AL

Program Fac

```
y := 1; z := 0;

while ¬(z = x) do
    z := z + 1;
    y := y * z
```


Program Specification

```
{ x > 0}
y := 1; z := 0;

while ¬(z = x) do
    z := z + 1;
    y := y * z
{ y = x! }
```


Program Verification

```
{ x > 0}
y := 1; z := 0;
{ y = z! }                         1 = 0!
while ¬ (z = x) do
    { y = z! ∧ ¬(x =z) }
    z := z + 1;
    { y×z = z! }
    y := y * z
    { y = z! }
{ y = x! }            ∀ x,y,z. ( (y = z! ∧ ¬ ¬(z = x) ) → (y = x!) )
```

Predicate Logic

- Sten kan ikke flyve og morlille kan ikke flyve
  ergo er morlille en sten!
- $\forall$x. (St(x) $\rightarrow \neg$ Fl(x)), $\neg$Fl(morlille)
  $|\neq$ St(morlille)

- Fugle kan flyve og piphans er en fugl
  ergo kan piphans flyve!

$\forall$ $\forall$x. (Bi(x) $\rightarrow$ Fl(x)), Bi(piphans)
  $|=$ Fl(piphans)

Propositional Logic

- Hvis det er tirsdag er der dBerLog undervisning, og der er dBerLog undervisning
  ergo det er tirsdag!
- Tir $\rightarrow$ dBL, dBL $\not\models$ Tir

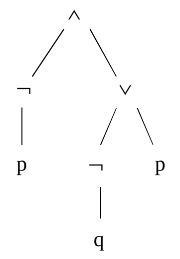
- Hvis det er tirsdag er der dBerLog undervisning, og der ikke dBerLog undervisning
  ergo det er ikke tirsdag!
- Tir $\rightarrow$ dBL, $\neg$ dBL $|=$ $\neg$ Tir

Propositional Logic - syntax

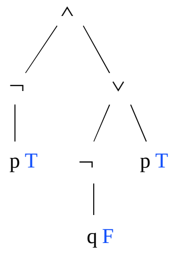
- Propositional variables
  p, q, r, ...

- Propositional formulas
  A ::= F | T | p | q | r |...
  $\neg A$ | $A \vee A$ | $A \wedge A$ | $A \rightarrow A$

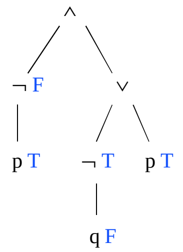
Propositional logic - semantics, example



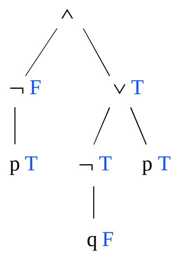
Propositional logic - semantics, example



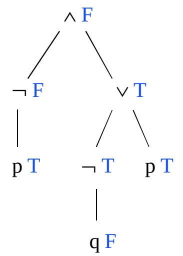
Propositional logic - semantics, example



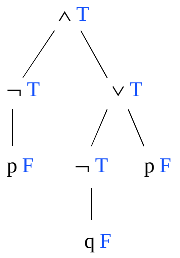
Propositional logic - semantics, example



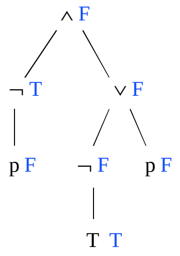
Propositional logic - semantics, example



Propositional logic - semantics, example



Propositional logic - semantics, example



Propositional Logic - semantics, formally

- Semantics of a variable is a value from {T, F}
- Semantics of a formula over $p_1, p_2,.., p_n$ is a function
  $((p_1 \rightarrow \{T, F\}) \times \ldots \times ((p_n \rightarrow \{T, F\})) \quad \rightarrow \quad \{T, F\}$
  defined by the following truth tables

Propositional logic - truth tables ∨, ∧

A	B	A ∨ B
T	T	T
T	F	T
F	T	T
F	F	F

A	B	A ∧ B
T	T	T
T	F	F
F	T	F
F	F	F

Propositional logic - truth tables →, ¬

A	B	A → B
T	T	T
T	F	F
F	T	T
F	F	T

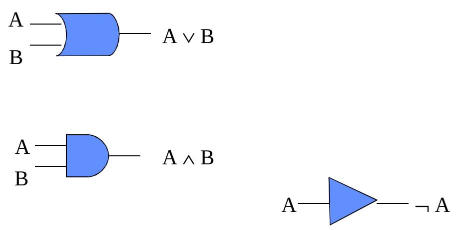
A	¬A
T	F
F	T

T	T
F	F

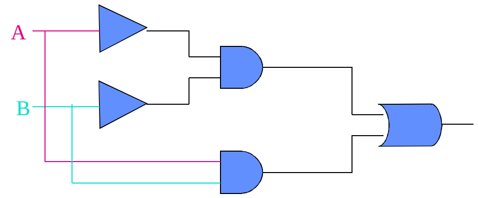
Propositional logic - semantics, example

p	q	¬p∧(¬ q∨p)
T	T	F
T	F	F
F	T	F
F	F	T

Logical Circuits - building blocks



Logical Circuits - example



Semantic consequence and tautology - definitions

- $A_1 \ldots A_n \models B$ (B is a *logical consequence* of $A_1, \ldots A_n$)
  iff
  B evaluates to T whenever $A_1, A_2, \ldots A_n$ evaluate to T

  Examples $p \vee q, \neg p \models q$ $p, p \to q \models p \wedge q$

- A is said to be *valid* (or a *tautology*) iff $\models A$

  Examples $\models p \vee \neg p$ $\models (p \to q) \vee p$

Logical equivalence

- **Definition**
  Two formulae A and B are said to be *logically equivalent*, $A \equiv B$, iff they define the same truth table

  Examples
  $A \equiv \neg \neg A$
  $A \wedge B \equiv \neg (\neg A \vee \neg B)$

Expressibility

- **Exercise**
  All truth tables can be expressed in propositional logic

- **Theorem**
  All truth tables can be expressed by the operators $\neg$ and $\to$

  Proof
  $A \vee B \equiv (\neg A) \to B$
  $A \wedge B \equiv \neg (\neg A \vee \neg B)$

Expressibility

- **Exercise**: all truth tables can be expressed in propositional logic

- **Theorem**
  All truth tables can be expressed by the operator | (Sheffer stroke / nand-gate) defined by the following truth table

A	B	A | B
T	T	F
T	F	T
F	T	T
F	F	T

Todays programme: Propositional Logic

- *Familiarity* with basic *terminology* of logics
  - Syntax, logical connectives
  - Semantics: models, truth, validity, logical consequence
  - Proof systems: deductions, deductive consequence, theorems
  - Soundness and completeness
- *Describe* propositional logic and some of its basic *properties*
  - Semantics: truth tables
  - Axiomatic proof system AL and its deductions
  - Soundness and completeness of AL

Axiomatic Proof System - definition

- A set of *well formed formulae (*A, B ,.. ∈ *)* wff
- A set of *axioms* Ax ⊆ wff
- A set of *deduction rules*:

$$\frac{A_1,\ A_2,\ \ldots\ldots,\ A_n}{B} \qquad \begin{matrix}\text{(premises)}\\ \text{(consequence)}\end{matrix}$$

Axiomatic Proof System AL for Propositional Logic

- A set of *well formed formulae (*A, B ,.. ∈ *)* wff
  - well formed formulae of propositional logic over ¬ and →
- A set of *axioms* Ax ⊆ wff
  - Ax1 $A \rightarrow (B \rightarrow A)$
  - Ax2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
  - Ax3 $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$
- A set of *deduction rules*:
  - Modus ponens MP:

$$\frac{A,\quad A \rightarrow B}{B}$$

AL deduction - example

1.	$A \rightarrow ((B \rightarrow A) \rightarrow A)$	Ax1
2.	$(A\rightarrow((B\rightarrow A)\rightarrow A)) \rightarrow ((A\rightarrow(B\rightarrow A))\rightarrow(A\rightarrow A))$	Ax2
3.	$(A\rightarrow(B\rightarrow A))\rightarrow(A\rightarrow A)$	MP 1, 2
4.	$A \rightarrow (B \rightarrow A)$	Ax1
5.	$A \rightarrow A$	MP 3, 4

Axiomatic Proof System - Deduction

- A *deduction* is a sequence of well formed formulae
  $A_1, A_2, \ldots, A_n$
  such that for all i, $n \geq i \geq 1$, either:
  (a) A_i is an axiom *instance* or
  (b) A_i is a hypothesis (from a set of formulae H)
  (c) A_i is derived by an deduction rule using formulae A_j where $j < i$ as premises
- A_n is a *deductive consequence* of H, $H \vdash A_n$
  where H is the set of hypotheses used in the deduction
- A is a *theorem* iff $\emptyset \vdash A$ (notation: $\vdash A$)

AL deduction - example TI

1.	$B \to C$	Hyp
2.	$(B\to C) \to (A \to (B\to C)$	Ax1
3.	$A \to (B\to C)$	MP 1, 2
4.	$A \to (B \to C) \to (A\to B) \to (A\to C)$	Ax2
5.	$(A\to B) \to (A\to C)$	MP 3, 4
6.	$A \to B$	Hyp
7.	$A \to C$	MP 5, 6

Conclude: $\{A \to B, B \to C\} \vdash A \to C$

Meta theorems

4.1	$\vdash A \to A$	
4.2	$\vdash \neg A \to (A \to B)$	
4.5	$\vdash (B \to C) \to ((A \to B\} \to (A \to C))$	(TI)
4.8	$\vdash \neg\neg A \to A$	
4.9	$\vdash (\neg A \to A) \to A$	
4.10	$\vdash A \to (\neg B \to \neg (A \to B)$	
4.17	$\vdash (B \to A) \to ((\neg B \to A) \to A)$	

Deduction Theorem for AL

- **Theorem 4.6**
  If $H \cup \{A\} \vdash B$ then $H \vdash A \to B$
- **Theorem 4.7**
  If $H \vdash A \to B$ then $H \cup \{A\} \vdash B$

Todays programme: Propositional Logic

- *Familiarity* with basic *terminology* of logics
  - Syntax, logical connectives
  - Semantics: models, truth, validity, logical consequence
  - Proof systems: deductions, deductive consequence, theorems
  - Soundness and completeness
- *Describe* propositional logic and some of its basic *properties*
  - Semantics: truth tables
  - Axiomatic proof system AL and its deductions
  - Soundness and completeness of AL

Proofs and semantics - fundamental definitions

- An axiomatic proof system for |- is said to be *sound* for |= iff for all formulae A:

  if |- A then |= A
- An axiomatic proof system for |- is said to be *complete* for |= iff for all formulae A:

  if |= A then |- A
- An axiomatic proof system for |- is said to be *consistent* iff for all formulae A it is not the case that (|- A and |- ¬A)

Prop. Logic - soundness and completeness

- **Theorem**

  The axiomatic proof system AL for propositional logic is *sound* and *complete*!

Soundness proof

- **Theorem 4.11**

  For all wff's A, if |- A then |= A
- *Proof*: Induction in lengths of proofs

  Induction hypothesis:

  M(k) = for all proofs of |- A with a proof of length ≤ k, it is the case that |= A

Completeness proof I

- **Theorem 4.19**
  For all wff's A, if $|= A$ then $|- A$

- *Lemma I*
  Let A be a formula with atoms $\{p_1, p_2,..p_n\}$. Let l be a line in A's truth table, and let $\underline{p}_i$ be p_i if the entry of p_i in line l is T, otherwise $\underline{p}_i$ is $\neg p_i$. Then

  $\underline{p}_1, \underline{p}_2,.., \underline{p}_n \vdash A$ is provable if the entry for A in line l is T

  $\underline{p}_1, \underline{p}_2,.., \underline{p}_n \vdash \neg A$ is provable if the entry for A in line l is F

Meta theorems

4.1 $\vdash A \rightarrow A$
4.2 $\vdash \neg A \rightarrow (A \rightarrow B)$
4.5 $\vdash (B \rightarrow C) \rightarrow ((A \rightarrow B\} \rightarrow (A \rightarrow C))$ (TI)
4.8 $\vdash \neg\neg A \rightarrow A$
4.9 $\vdash (\neg A \rightarrow A) \rightarrow A$
4.10 $\vdash A \rightarrow (\neg B \rightarrow \neg (A \rightarrow B)$
4.17 $\vdash (B \rightarrow A) \rightarrow ((\neg B \rightarrow A) \rightarrow A)$

Completeness proof II

- **Theorem 4.19**
  For all wff's A, if $|= A$ then $|- A$

- *Lemma II*
  Let A be a valid formula with atoms $\{p_1, p_2,..p_n\}$. From the two deductive consequences from *Lemma I*

  $\underline{p}_1, \underline{p}_2,..,\underline{p}_{n-1}, p_n \vdash A$

  $\underline{p}_1, \underline{p}_2,..,\underline{p}_{n-1}, \neg p_n \vdash A$

  we can construct the deductive consequence

  $\underline{p}_1, \underline{p}_2,.., \underline{p}_{n-1} \vdash A$

Deduction Theorem for AL

- **Theorem 4.6**
  If $H \cup \{A\} \vdash B$ then $H \vdash A \rightarrow B$

- **Theorem 4.7**
  If $H \vdash A \rightarrow B$ then $H \cup \{A\} \vdash B$

Meta theorems

4.1 |- $A \rightarrow A$

4.2 |- $\neg A \rightarrow (A \rightarrow B)$

4.5 |- $(B \rightarrow C) \rightarrow ((A \rightarrow B\} \rightarrow (A \rightarrow C))$ (TI)

4.8 |- $\neg\neg A \rightarrow A$

4.9 |- $(\neg A \rightarrow A) \rightarrow A$

4.10 |- $A \rightarrow (\neg B \rightarrow \neg (A \rightarrow B)$

4.17 |- $(B \rightarrow A) \rightarrow ((\neg B \rightarrow A) \rightarrow A)$

Validity in propositional logic

- Validity problem for propositional logic:
  Given a propositional logic formula A,
  is A valid, i.e. |= A?
- **Theorem**
  The validity problem for popositional logic is decidable
- **Proof**
  Easy - construct truth table!
- **Corollary**:
  The set of valid formulas in propositional logic is recursive

Exercises

- *Describe* the semantics of propositional logic
  - Kelly page 14: 1.10 (i)-(ii): expressibility of nor and nand
  - Kelly page 25: 1 (i)-(v), 2, 5 (i)-(ii), 6, 7: truth tables
- *Describe* and *construct* deductions in AL
  - Kelly page 92-93: 2 (i)-(ii), 3 (iv)-(v)
- *Analyze* proof of completeness of AL
  - Kelly page 90: 4.11

dBerLog exam 2007

- Oral exam
- 20 minutes - without preparation time
- Grading (12-scale)
- Internal examiners

- Two questions:
  - Computability
  - Logic

dBerLog Compulsory Assignments 2007

- Write manuscripts for a 15 minutes exam presentation for each of the two exam questions: Computability and Logic
- 2-3 pages each
- dBerLog curriculum follows from dBerLog home page - Weekly Schedules (will appear under Final Exam later)

- First assignment: Computability
- Hand in to your tutor no later than Wednesday September 26!

dBerLog Compulsory Assignments 2007

- Your assignment contains at least:
  - an outline of the presentation
  - brief argumentation for choices made
  - indications of levels in the dBerLog learning taxonomy:
    - to be *familiar* with the basic *terminology* for computability and logic
    - to *describe* basic computability classes and fundamental logics
    - to *describe* basic *properties* of computability classes and logics
    - to *explain* constructive/algorithmic approaches to computability classes and logics
    - to *analyse* and to *prove* properties of computability classes and logics

Next week: Predicate Logic

- Sten kan ikke flyve og morlille kan ikke flyve
  ergo er morlille en sten!
- $\forall x. (St(x) \rightarrow \neg Fl(x)), \quad \neg Fl(morlille)$
  $\not\models \quad St(morlille)$

- Fugle kan flyve og piphans er en fugl
  ergo kan piphans flyve!

∀ $\forall x. (Bi(x) \rightarrow Fl(x)), \quad Bi(piphans)$
$\models \quad Fl(piphans)$

Prolog

Predicate logic

$\forall x. (Bi(x) \rightarrow Fl(x)), \quad Bi(piphans)$
$\models \quad Fl(piphans)$

Prolog

Fl(X) :- Bi(X).
Bi (piphans).

FL(piphans)?