DAT060
2016-09-02
LV 1, Lecture 2

1 Propositional Logic

$\wedge$ - conjunction (and)
$\vee$ - disjunction (or)
$\rightarrow$ - implication (if – then)
$\bot$ - absurdity (can never be true)
$\neg$ - negation (not)
ϕ_1,ϕ_2,... $\vdash \psi$ - we can derive ψ from ϕ_1,ϕ_2,...

1.1 Formulas

Formulas are built from atomic formulas (atomic formulas usually represented by lower case letters: p_i,q_j, ...) by using the connectives $\wedge$, $\vee$, $\rightarrow$, $\bot$, and $\neg$. For instance: $(p \rightarrow q) \vee (q \wedge r)$.

1.2 Natural deduction (Gentzen)

There are two kinds of rules: an introduction rule, and an elimination rule.

1.2.1 $\wedge$ - introduction

If we have concluded ϕ and ψ separately, conjunction allows to conclude $\phi \wedge \psi$.

$$\frac{\phi \qquad \psi}{\phi \wedge \psi} \wedge i$$

1.2.2 $\wedge$ - elimination

You can eliminate or introduce each connectives given certain rules. The rules for 'and-elimination' are:

$$\frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad\qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

The rule $\wedge e_1$says that if you have a proof of $\phi \wedge \psi$, you can apply this rule and get a proof of ϕ. $\wedge e_2$ is analogous for ψ.

Example Show that $\phi \wedge \psi \vdash \psi \wedge \phi$

1	$\phi \wedge \psi$	premise
2	ψ	$\wedge e_2$ 1
3	ϕ	$\wedge e_1$ 1
4	$\psi \wedge \phi$	$\wedge$i 2, 3

1.2.3 $\rightarrow$ - introduction

If we assume ϕ and eventually show ψ, we can introduce an implication.

$$\begin{array}{c} [\phi] \\ \vdots \\ \psi \\ \hline \phi \rightarrow \psi \end{array} \rightarrow i$$

1.2.4 $\rightarrow$ - elimination

$$\frac{\phi \qquad \phi \rightarrow \psi}{\psi} \rightarrow e$$

Example Show that $\phi \rightarrow (\psi_1 \wedge \psi_2) \vdash \phi \rightarrow \psi_1$

1	$\phi \rightarrow (\psi_1 \wedge \psi_2)$	premise
2	ϕ	assumption
3	$\psi_1 \wedge \psi_2$	$\rightarrow$e 2,1
4	ψ_1	$\wedge e_1$ 3
5	$\phi \rightarrow \psi_1$	$\rightarrow$i 2–4

1.2.5 $\perp$- introduction

You can never introduce an absurdity, hence there is no rule to do it.

1.2.6 $\perp$ - elimination

The rule of $\perp$ - elimination states that if you have an absurdity you can deduce anything.

$$\frac{\perp}{\phi} \perp \text{e}$$

1.2.7 $\neg\neg$ - introduction and elimination

"It is raining" is the same thing as "it is not not raining".

$$\frac{\neg\neg\phi}{\phi}\ \neg\neg e \qquad\qquad \frac{\phi}{\neg\neg\phi}\ \neg\neg i$$

Example $\phi \vdash \neg\neg\ \phi$ (i.e.) $\phi \vdash (\phi \to \bot) \to \bot$

1	ϕ	premise
2	$\quad \phi \to \bot$	assumption
3	$\quad \bot$	$\to$e 1, 2
4	$(\phi \to \bot) \to \bot$	$\to$i 2–3

Example $\phi \wedge \neg\phi \vdash \psi$

1	$\phi \wedge \neg\phi$	premise
2	ϕ	$\wedge e_1$ 1
3	$\phi \to \bot$	$\wedge e_2$ 1 (definition of $\neg\phi$ is $\phi \to \bot$)
4	$\bot$	$\to$e 2, 3
5	ψ	$\bot$e 4

1.2.8 $\vee$ - introduction

$$\frac{\phi}{\phi \vee \psi}\ \vee i_1 \qquad\qquad \frac{\psi}{\phi \vee \psi}\ \vee i_2$$

1.2.9 $\vee$ - elimination

$$\frac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots\\ \chi\end{array} \qquad \begin{array}{c}[\psi]\\ \vdots\\ \chi\end{array}}{\chi}\ \vee e$$

Example $\phi \vee \psi \vdash \psi \vee \phi$

1	$\phi \vee \psi$	premise
2	$\quad \phi$	assumption
3	$\quad \psi \vee \phi$	$\vee i_2$ 2
4	$\quad \psi$	assumption
5	$\quad \psi \vee \phi$	$\vee i_1$ 4
6	$\psi \vee \phi$	$\vee$e 1, 2–3, 4–5

1.2.10 Proof by Contradiction

PBC says that if we from $\neg\phi$ obtain a contradiction, then we are entitled to deduce ϕ. ***Note*** *that Jan defines* $\neg p$ *as* $p \to \bot$.

$$\begin{array}{c} [\neg\phi] \\ \vdots \\ \dfrac{\bot}{\phi}\, PBC \end{array}$$

Example $\neg\neg\phi \vdash \phi$

1	$\neg\neg\phi$	premise
2	$\quad \neg\phi$	assumption
3	$\quad \bot$	$\to$e 2, 1
4	ϕ	PBC 2–3

DAT060
2016-09-02
LV 1, Lecture 3

1 Exercises

1.1 Express the following sentences, in English, using propositional logic.

1.1.1 "Today it will rain or shine, but not both."

p: It will rain.
q: It will shine.

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

1.1.2 "If the barometer falls, it will either rain or snow."

p: Barometer falls.
q: It will rain.
r: It will snow.

$p \rightarrow (q \vee r)$

Note that "or" is usually meant as xor: $p \rightarrow (q \vee r) \wedge \neg (q \wedge r)$

1.2 Recall conventions about tightness

Ranked in order:

1. $\neg$ binds the tightest
2. $\wedge$, $\vee$ binds equally tight
3. $\rightarrow$ is right associative: $p \rightarrow q \rightarrow r$ is $(p \rightarrow (q \rightarrow r))$

1.2.1 $(p \rightarrow q) \wedge \neg (r \vee p \rightarrow q)$

Should really be: $((p \rightarrow q) \wedge (\neg ((r \vee p) \rightarrow q)))$

1.2.2 $(\mathbf{p} \to \mathbf{q}) \to \mathbf{r} \to \mathbf{s} \vee \mathbf{t}$

Sholud really be: ((p → q) → (r → (s ∨ t)))

1.3 Show that the following sequents are valid

1.3.1 $\mathbf{p} \wedge \mathbf{q} \vdash \mathbf{q} \wedge \mathbf{p}$

1	$p \wedge q$	premise
2	q	$\wedge e_2$ 1
3	p	$\wedge e_1$ 1
4	$q \wedge p$	$\wedge$i 2–3

1.3.2 $(\mathbf{p} \wedge \mathbf{q}) \wedge \mathbf{r}, \mathbf{s} \wedge \mathbf{t} \vdash \mathbf{q} \wedge \mathbf{s}$

1	$(p \wedge q) \wedge r$	premise
2	$s \wedge t$	premise
3	$p \wedge q$	$\wedge e_1$ 1
4	q	$\wedge e_2$ 3
5	s	$\wedge e_1$ 2
6	$q \wedge s$	$\wedge$i 4, 5

1.3.3 $\mathbf{p} \to (\mathbf{p} \to \mathbf{q}), \mathbf{p} \vdash \mathbf{q}$

1	$p \to (p \to q)$	premise
2	p	premise
3	$p \to q$	→e 1, 2
4	q	→e 3, 2

1.3.4 $\mathbf{p} \to \mathbf{q}, \mathbf{q} \to \mathbf{r} \vdash \mathbf{p} \to \mathbf{r}$

1	$p \to q$	premise
2	$q \to r$	premise
3	p	assumption
4	q	→e 1, 3
5	r	→e 2, 4
6	$p \to r$	→i 3–5

1.3.5 $\vdash \mathbf{p} \to (\mathbf{q} \to \mathbf{p})$

1	p	assumption
2	$\quad q$	assumption
3	$\quad p$	copy 1
4	$q \to p$	$\to$i 2–3
5	$p \to (q \to p)$	$\to$i 1–4

1.3.6 $\mathbf{p} \wedge \mathbf{q} \vdash \mathbf{p} \vee \mathbf{q}$

1	$p \wedge q$	premise
2	p	$\wedge e_1$ 1
3	$p \vee q$	$\vee i_1$ 2

1.3.7 $(\mathbf{p} \vee (\mathbf{q} \to \mathbf{p})) \wedge \mathbf{q} \vdash \mathbf{p}$

1	$(p \vee (q \to p)) \wedge q$	premise
2	$p \vee (q \to p)$	$\wedge e_1$ 1
3	q	$\wedge e_2$ 1
4	$\quad p$	assumption
5	$\quad q \to p$	assumption
6	$\quad p$	$\to$e 5, 3
7	p	$\vee$e 2, 4, 5–6

1.3.8 $\mathbf{p \to r,\ q \to s \vdash p \vee q \to r \vee s}$

1	$p \to r$	premise
2	$q \to s$	premise
3	$\quad p \vee q$	assumption
4	$\quad\quad p$	assumption
5	$\quad\quad r$	$\to$e 1, 4
6	$\quad\quad r \vee s$	$\vee i_1$ 5
7	$\quad\quad q$	assumption
8	$\quad\quad s$	$\to$e 2, 7
9	$\quad\quad r \vee s$	$\vee i_2$ 8
10	$\quad r \vee s$	$\vee$e 3, 4–6, 7–9
11	$p \vee q \to r \vee s$	$\to$i 3–10

1.3.9 $\mathbf{\neg\, p \vdash p \to q}$

1	$p \to \bot$	premise
2	$\quad p$	assume
3	$\quad \bot$	$\to$e 1, 2
4	$\quad q$	$\bot$e 3
5	$p \to q$	$\to$i 2–4

1.3.10 $\mathbf{\neg(p \to q) \vdash q \to p}$

1	$\neg(p \to q)$	premise
2	$\quad q$	assumption
3	$\quad\quad p$	assumption
4	$\quad\quad q$	copy 2
5	$\quad p \to q$	$\to$i 3–4
6	$\quad \bot$	$\to$e 1, 5
7	$\quad p$	$\bot$e 6
8	$q \to p$	$\to$i 2–7

1.4 Show: $\vdash \phi \vee \neg\phi$ (Law of excluded middle)

1	$\neg(\phi \vee \neg\phi)$	assumption
2	ϕ	assumption
3	$\phi \vee \neg\phi$	$\vee i_1$ 2
4	$\bot$	$\to$e 1, 3
5	$\phi \to \bot$	$\to$i 2–4
6	$\phi \vee \neg\phi$	$\vee i_2$ 5
7	$\bot$	$\to$e 1, 6
8	$\phi \vee \neg\phi$	PBC 1–7

1.4.1 $\neg\mathbf{p} \to \mathbf{p} \vdash \mathbf{p}$

1	$\neg p \to p$	premise
2	$\neg p$	assumption
3	p	$\to$e 1, 3
4	$\bot$	$\to$e 2, 4
5	p	PBC 2–4

1.4.2 $\mathbf{p} \vee \mathbf{q} \vdash (\mathbf{p} \to \mathbf{q}) \to \mathbf{q}$

1	$p \vee q$	premise
2	$p \to q$	assumption
3	p	assumption
4	q	$\to$e 2, 3
5	q	assumption
6	q	$\vee$e 1, 3–4, 5
7	$(p \to q) \to q$	$\to$i 2–6

1.4.3 $(\mathbf{p \to q}) \to \mathbf{q} \vdash \mathbf{p} \vee \mathbf{q}$

1	$(p \to q) \to q$	premise
2	$\neg(p \vee q)$	assume
3	p	assume
4	$p \vee q$	$\vee i_1$ 3
5	$\bot$	$\to$e 2, 4
6	q	$\bot$e 5
7	$p \to q$	$\to$i 3–6
8	q	$\to$e 1, 7
9	$p \vee q$	$\vee i_2$ 8
10	$\bot$	$\to$e 2, 9
11	$p \vee q$	PBC 2–10

DAT060
2016-09-06
LV 2, Lecture 1

1 Inductive Definitions

1.1 Example: The set of natural numbers ($\mathbb{N}$)

Assume that we have got two constructors: 0 and a successor of 0.
We also have rules:

1. $0 \in \mathbb{N}$
2. if m $\in \mathbb{N}$ then succ(m) $\in \mathbb{N}$

2 Definition by recursion

2.1 Example: The factorial function, n!

Given n!, fac(n), we can define the function as:

fac(0)=1
fac(succ(n))=succ(n)*fac(n)

3 Inductive definition

Inductive definition of the set F, of propositional formulas (Def 1.27, p 32).

- Atoms, including $\perp$, are formulas. I.e. elements of F.
- If $\phi \in$ F and $\psi \in$F then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ and $(\neg\phi)$ are elements of F.

3.1 Example: Definition by recursion of the function par()

par() computes the number of parantheses in an expression.

1. par(ϕ) = 0 (if ϕ is an atomic formula)

2. par(composite expression)

   (a) $\text{par}((\phi \wedge \psi)) = \text{par}(\phi) + \text{par}(\psi) + 2$
   (b) $\text{par}((\phi \vee \psi)) = \text{par}(\phi) + \text{par}(\psi) + 2$
   (c) $\text{par}((\phi \rightarrow \psi)) = \text{par}(\phi) + \text{par}(\psi) + 2$
   (d) $\text{par}((\neg\phi)) = \text{par}(\phi) + 2$

4 Semantics

Semantics can be expressed in various ways.

- Truth values
- Pruportions defined by laying down what ... Constructivism(?)

4.1 Truth Tables

ϕ	ψ	$\phi \wedge \psi$	$\phi \vee \psi$	$\phi \rightarrow \psi$	$\neg\phi$
T	T	T	T	T	F
T	F	F	T	F	F
F	T	F	T	T	T
F	F	F	F	T	T

4.1.1 Definition: How to compute the truth table (Def 1.28, p 37)

You can view each row in a truth table as a line describing something about the world. You are assigning either T or F to an atom.

A valuation v is a function from the set of atoms to the set of truth values:

v : {p, q, r, ..., p_1, p_2, ...} →{T, F}

v can be extended to the set F of propositional formulas by recursion on F using the truth table. So we extend v to every propositional formula.

v: F →{T, F}

- v is already defined on the atoms, and we put v(⊥)=F
- $\text{v}(\phi \wedge \psi) = \begin{cases} T & if\ v(\phi) = v(\psi) = T \\ F & otherwise \end{cases}$
- $\text{v}(\phi \vee \psi) = \begin{cases} F & if\ v(\phi) = v(\psi) = F \\ T & otherwise \end{cases}$

- $\mathrm{v}(\phi \to \psi) = \begin{cases} F & if\ v(\phi) = T\, and\, v(\psi) = F \\ T & otherwise \end{cases}$

We will occasionally use the notation $[|\phi|]_v$ for v(ϕ). Semantic brackets by Dana Scott.

4.2 Soundness

The implication from right to left: $\Leftarrow$, is called soundness. It can be proved by induction on the length of the proof of ψ from ϕ_1, ..., ϕ_n. (The proof can be found in the book.)

4.3 Completeness

The implication from left to right: $\Rightarrow$, is called completeness. (The proof can be found in the book.)

4.4 Definition 1.34, p 46

If for all evalutations in which all $\phi_1, ..., \phi_n$ evaluates to T, ψ evaluates to T as well. We say that $\phi_{1,...,}\phi_n \models \psi$ holds and call $\models$ the semantic entailment notation.

Alternative notation:

$[|\phi_1|]_v = ... = [|\phi_n|]_v$= T, then $[|\psi|]_v$=T

4.4.1 Example

p $\wedge$ q $\models$ p

p, q $\models$ p

p $\vee$ q $\not\models$ p (p=F and q=T)

4.5 The Constructive Semantics{} of Propositional Logic in terms of proofs

says that if:

we can prove	then
$\phi \wedge \psi$	we can prove ϕ and prove ψ
$\phi \vee \psi$	we can prove ϕ or we can prove ψ
$\phi \to \psi$	we can give a method which to each proof of ϕ gives o proof of ψ
$\bot$	nothing comes as a proof of $\bot$

What about $\phi \vee \neg\phi$ (Law of Exclude the Middle - LEM)? If it is true constructively, we should be able to prove either ϕ or $\neg\phi$. This has not been done, yet.

DAT060
2016-09-09
LV 2, Lecture 2

1 Propositional Logic

Semantics:

- The meaning of a proposition is its truth value. (T or F)
- A proposition is given by laying down what comes as a proof of it. (Constructive) ($\phi \vee \neg\phi$, does not hold here)

Definition A valuation v (a model) is a function from the atoms to the set of truth values. I.e.: v : {p,q,r,...,p_1, p_2, ...} $\rightarrow$ {T, F}. v can be extended to all formulas by the truth tables using recursion. Note that v(ϕ) is the same as $[|\phi|]_v$.

Definition ϕ_1,...,$\phi_n \models \psi$ tells us that for all valuations v, we have that iff\end{figure}
\begin{figure} $[|\phi_1|]_v$,...,$[|\phi_n|]_v$ = T then $[|\psi|]_v$ = T.

Definition Sondness says that: If ϕ_1,...,$\phi_n \vdash \psi$ then $\phi_1, ..., \phi_n \models \psi$.

Definition Completeness (the other way around) says that: If ϕ_1,...,$\phi_n \models \psi$ then $\phi_1, ..., \phi_n \vdash \psi$.

Definition If: $\models \psi$ then we say that ψ is a tautology, or logical truth.

Note: $\phi_1, ..., \phi_n \models \psi$ iff $\models (\phi_1, ..., \phi_n) \rightarrow \psi$

1.1 F_ϕ

Each formula ϕ containing the atoms p_1,...,p_n gives a function F_ϕ.

F_ϕ : {T, F}^{n} $\rightarrow$ {T, F}

where $\{T, F\}^n = \{(A_1, ..., A_n) \mid A_i \in \{T, F\}, 1 \leq i \leq n\}$

defined by $F_\phi(A_1, ..., A_n) = [|\phi|]_v$

where $v(p_1)=A_1$, $v(p_2)=A_2$, ..., $v(p_n) = A_n$

There are $2^{(2^n)}$functions in $\{T, F\}^n \rightarrow \{T, F\}$. Can each function in $\{T, F\}^n \rightarrow \{T, F\}$ be expressed by a formula? Yes!

1.1.1 Proof by Example

Given an arbitrary function Fun, we want to express Fun using our standard connectives.

p_1	p_2	p_3	Fun	Matching expr:
T	T	T	F	$\neg p_1 \vee \neg p_2 \vee \neg p_3$
T	T	F	T	
T	F	T	T	
T	F	F	F	$\neg p_1 \vee p_2 \vee p_3$
F	T	T	T	
F	T	F	F	$p_1 \vee \neg p_2 \vee p_3$
F	F	T	T	
F	F	F	T	

Fun is expressed by the conjunctions of the three expr above (conjuctive normal form):

$(\neg p_1 \vee \neg p_2 \vee \neg p_3) \wedge (\neg p_1 \vee p_2 \vee p_3) \wedge (p_1 \vee \neg p_2 \vee p_3)$.

1.2 Connectives

Each formula of propositional logic is equivalent to a formula which only contains the connectives: $\wedge$, $\vee$, $\neg$. $\{\wedge, \vee, \neg\}$ is called the complete set of connectives.

$\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$, hence $\{\vee, \neg\}$ is complete
$\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$, hence $\{\wedge, \neg\}$ is complete

p	q	$p \uparrow q$ (nand)	$p \downarrow q$ (nor)	$\neg p \equiv p \downarrow p$	$p \vee q \equiv \neg(p \downarrow q) \equiv (p \downarrow q) \downarrow (p \downarrow q)$
T	T	F	F	F	T
T	F	T	F	F	T
F	T	T	F	T	T
F	F	T	T	T	F

1.3 Types and Propositions

∧i and xi

$$\frac{\phi \qquad \psi}{\phi \wedge \psi} \wedge i$$

$$\frac{a \in A \qquad b \in B}{< a, b > \in AxB} \, xi$$

∧e and xe

$$\frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

$$\frac{c \in AxB}{fst(c) \in A} \, xe_1 \qquad \frac{c \in AxB}{snd(c) \in B} \, xe_2$$

$$\begin{cases} \text{fst(<a,b>)=a} \\ \text{snd(<a,b>)=b} \end{cases}$$

∨i and +i

$$\frac{\phi}{\phi \vee \psi} \vee i_1 \qquad \frac{\psi}{\phi \vee \psi} \vee i_2$$

$$\frac{a \in A}{inl(a) \in A + B} + i_1 \qquad \frac{b \in B}{inr(x) \in A + B} + i_2$$

∨e and +-elimination

$$\frac{\phi \vee \psi \qquad \begin{matrix}[\phi] \\ \vdots \\ \chi\end{matrix} \qquad \begin{matrix}[\psi] \\ \vdots \\ \chi\end{matrix}}{\chi} \vee \text{e} \qquad \frac{C \in A + B \qquad \begin{matrix}[x \in A] \\ \vdots \\ d(x) \in C\end{matrix} \qquad \begin{matrix}[y \in B] \\ \vdots \\ e(y) \in C\end{matrix}}{when(c, d, e) \in C} + \text{e}$$

$$\begin{cases} \text{when (inl(a),d,e)=d(a)} \\ \text{when (inr(b),d,e)=e(b)} \end{cases}$$

→i

$$\frac{\begin{array}{c}[\phi]\\ \vdots\\ \psi\end{array}}{\phi \to \psi} \to i \qquad \frac{\begin{array}{c}[x \in A]\\ \vdots\\ b(x) \in B\end{array}}{\lambda x\ b(x) \in A \to B} \to i$$

→e

$$\frac{\phi \qquad \phi \to \psi}{\psi} \to e \qquad \frac{a \in A \qquad f \in A \to B}{apply(f, a) \in B} \to e$$

apply($\lambda x\ b(x), a$) = b(a)

DAT060
LV 1, Lecture 3
Assignment 1

Problem 1

(1) If the sun shines, Emmy and Kurt eat ice cream.

p: the sun shines
q: emmy eats ice cream
r: kurt eats ice cream

$p \rightarrow q \wedge r$

Alternative Solutions:

p: the sun shines
q: E and K eat ice cream

$p \rightarrow q$
—
p: if the sun shines, E and K eat ice cream.

p

(2) Exactly one out of Ada, Haskell, and Bertrand doesn't like cats.

p: ada likes cats
q: haskell likes cats
r: bertrand likes cats

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

Alternative Solutions:

p: Ada does not like cats.
q: Haskell does not like cats.

r: Berthrand does not like cats.

$(\text{p} \land \neg\text{q} \land \neg\text{r}) \lor (\neg\text{p} \land \text{q} \land \neg\text{r}) \lor (\neg\text{p} \land \neg\text{q} \land \text{r})$

Problem 2

1) $\mathbf{p \land (q \land r) \vdash (p \land q) \land r}$

1	$p \land (q \land r)$	premise
2	p	$\land e_1$ 1
3	$q \land r$	$\land e_2$ 1
4	q	$\land e_1$ 3
5	r	$\land e_2$ 3
6	$p \land q$	$\land$i 2, 4
7	$(p \land q) \land r$	$\land$i 6, 5

2) $\mathbf{(p \to r) \lor (q \to r) \vdash p \land q \to r}$

1	$(p \to r) \lor (q \to r)$	premise
2	$p \land q$	assumption
3	p	$\land e_1$ 2
4	q	$\land e_2$ 2
5	$p \to r$	assumption
6	r	$\to$e 3, 5
7	$q \to r$	assumption
8	r	$\to$e 4, 7
9	r	$\lor$e 1, 5–6, 7–8
10	$p \land q \to r$	$\to$i 2–9

3) $\mathbf{p \to \neg p,\ \neg p \to p \vdash \bot}$

1	$p \to \neg p$	premise
2	$\neg p \to p$	premise
3	$\quad p$	assumption
4	$\quad \neg p$	$\to$e 1, 3
5	$\quad \bot$	$\neg$e 3, 4
6	$\neg p$	$\neg$i 3–5
7	$\quad \neg p$	assumption
8	$\quad p$	$\to$e 2, 6
9	$\quad \bot$	$\neg$e 6, 7
10	$\neg\neg p$	$\neg$i 7–9
11	$\bot$	$\neg$e 10, 6

Apparently you can solve this with a flashy lemma-macro. I did not get that down.

4) $\mathbf{\neg(p \wedge q) \vdash \neg p \vee \neg q}$

1	$(p \wedge q) \to \bot$	premise
2	$p \vee \neg p$	LEM
3	$\quad p$	assumption
4	$\quad q \vee \neg q$	LEM
5	$\quad\quad q$	assumption
6	$\quad\quad p \wedge q$	$\wedge$i 3, 5
7	$\quad\quad \bot$	$\neg$e 6, 1
8	$\quad\quad \neg p \vee \neg q$	$\bot$e
9	$\quad\quad \neg q$	assumption
10	$\quad\quad \neg p \vee \neg q$	$\vee i1$ 9
11	$\quad \neg p \vee \neg q$	$\vee$e 4, 5–8, 9–10
12	$\quad \neg p$	assumption
13	$\quad \neg p \vee \neg q$	$\vee i_1$ 12
14	$\neg p \vee \neg q$	$\vee$e 2, 3–11, 12–13

Alternative Solution:

1	$\neg(p \wedge q)$	premise
2	$\quad \neg(\neg p \vee \neg q)$	assumption
3	$\quad\quad \neg p$	assumption
4	$\quad\quad \neg p \vee \neg q$	$\vee i_1$ 3
5	$\quad\quad \bot$	$\neg$e 4, 2
6	$\quad p$	PBC 3–5
7	$\quad\quad \neg q$	assumption
8	$\quad\quad \neg p \vee \neg q$	$\vee i_2$ 8
9	$\quad\quad \bot$	$\neg$e 8, 2
10	$\quad q$	PBC 7–9
11	$\quad p \wedge q$	$\wedge$i 6,10
12	$\quad \bot$	$\neg$e 11, 1
13	$\neg p \vee \neg q$	PBC 2–12

Problem 3 $(p \to q) \to p \vdash p$

1	$(p \to q) \to p$	premise
2	$p \vee \neg p$	LEM
3	$\quad p$	assumption
4	$\quad \neg p$	assumption
5	$\quad\quad p$	assumption
6	$\quad\quad \bot$	$\neg$e 4, 5
7	$\quad\quad q$	$\bot$e 6
8	$\quad p \to q$	$\to$i 5–7
9	$\quad p$	$\to$e 8, 1
10	p	$\vee$e 2, 3–3, 4–9

Alternative Solution:

1	$(p \to q) \to p$	premise
2	$\neg p$	assumption
3	p	assumption
4	$\bot$	$\neg$e 3, 2
5	q	$\bot$e 4
6	$p \to q$	$\to$i 3–5
7	p	$\to$e 6, 1
8	$\bot$	$\neg$e 7, 2
9	p	PBC 2–8

Assume Pierce's law instead of LEM.

$\vdash$ p $\vee$ $\neg$p
Assume $\neg$(p $\vee$ $\neg$p) $\vdash$ $\neg$p, then we must show that $\bot$ is implied by p. However, if we have p then p $\vee$ $\neg$p. This in turn implies $\bot$.

Next assignment

Exercise 1.4.2 d) (p 82)

Give the truth table for: p∧q→p∨q

p	q	p∧q	p∨q	p∧q → p∨q
T	T	T	T	T
T	F	F	T	T
F	T	F	T	T
F	F	F	F	T

Conjunctive Normal Form: p ∨ ¬p

Exercise 1.4.2 c)

Give the truth table for: p ∨ (¬(q ∧ (r → q)))

p	q	r	r → q	q ∧ (r → q)	¬(q ∧ (r → q))	p ∨ (¬(q ∧(r → q)))
T	T	T	T	T	F	T
T	T	F	T	T	F	T
T	F	T	F	F	T	T
T	F	F	T	F	T	T
F	**T**	**T**	T	T	F	**F**
F	**T**	**F**	T	T	F	**F**
F	F	T	F	F	T	T
F	F	F	T	F	T	T

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

¬p∧q∧r
¬p∧q∧¬r
—
¬((¬p ∧ q ∧ r) ∨ (¬p ∧ q ∧ ¬r)) <=> (p ∨¬q ∨ ¬r) ∧ (p ∨ ¬q ∨ r)

Exercise 1.4.12 a) (p 86)

Show that: $\neg p \vee (q \rightarrow p) \vdash \neg p \wedge q$, is not valid.

We use the soundness theorem: $\phi \vdash \psi \Rightarrow \phi \models \psi$. ($\phi \models \psi \Leftrightarrow \forall$ valuations, val(ϕ)=T => val(ψ)=T)

We strive towards finding a valuation s.t. val(ϕ)=T and val(ψ)=F

Attempt q=F: $\neg p \vee (F \rightarrow p) \vdash \neg p \wedge F$

$F \rightarrow p$ is true for all p, that means that LHS true and RHS, we disproved the expression!

This also works if p=T.

Exercise 1.4.12 b)

$\neg r \rightarrow (p \vee q)$, $r \wedge \neg q \vdash r \rightarrow q$

RHS = F iff r = T, q = F

LHS = $F \rightarrow (p \vee F)$, $T \wedge T$, it checks out for all values of p. We proved that the secuent is invalid.

Exercise 1.5.3 (p 87)

$\neg p \equiv p \rightarrow \bot$

Show that: $\{\rightarrow, \bot\}$ is adeqvate (with those two you can derive all other connectives).

$\phi \wedge \psi \equiv \neg\neg(\phi \wedge \psi) \equiv$
$\neg(\phi \wedge \psi \rightarrow \bot) \equiv$
$\neg(\phi \rightarrow \psi \rightarrow \bot) \equiv$
$(\phi \rightarrow \psi \rightarrow \bot) \rightarrow \bot$

DAT060
2016-09-13
LV 3, Lecture 1

1 Propositional Logic

So far we have talked about:

- Atoms: p, q, r, ...
- Connectives: $\wedge$, $\vee$, $\neg$, ...
- Proof system
  - Natural deduction
- Semantics
  - Classical semantics in terms of truth values.
  - Constructive semantics in terms of proofs.
- Identification of proposition and types:
  - $\phi \wedge \psi$ A x B
  - $\phi \vee \psi$ A + B
  - $\phi \rightarrow \psi$ A $\rightarrow$ B
  - $\bot$ $\emptyset$

2 Predicate Logic (Syllogism)

All dog have four legs.
Caro is a dogs.
Caro has four legs.

Predicate logic is about (propositional logic) + (quantifiers: $\exists$, $\forall$).

D(x): x is a dog.
F(x): x has four legs.

c: Caro

$\forall x$(D(x) $\rightarrow$ F(x))
D(c)
F(c)

2.1 The Dog Language

D and F are unary predicates (take one argument). C is an individual constant.

We extend this by: m(x), a unary function symbol which we interpret as "the mother of x", as well as O(x,y), a binary function which we interpret as "x is older than y".

$\forall x$(D(x) $\rightarrow$ O(m(x), x)))

2.2 Arithmetic

= (equality) binary predicate
+ (addition) binary function symbol
* (multiplication) binary function symbol
succ (successor) unary function symbol
0 (zero) individual constant

2.2.1 Examples

$\forall x \forall y$(x+y = y+x)

2.3 The Language of Set Theory

Contains only two predicates (= and $\in$) both binary. There are no individual constants, nor any function symbols.

$\exists x \forall y(\neg(y \in x))$ expresses the empty set.
$\forall x \forall y(x = y \leftrightarrow \forall z(z \in x \leftrightarrow z \in y))$ expresses the extensionality axiom.

$0 = \emptyset$
$1 = \{\emptyset\}$
$2 = \{\emptyset, \{\emptyset\}\}$
$n = \{0, \ldots, n\text{-}1\}$

2.4 Predicate Logic as a Formal Language (p. 99)

A vocabulary (P, F, C) consists of:

- P: set of predicate symbols.

- F: set of function symbols.
- C: set of constant symbols.

2.4.1 Vocabularies

The dog language has the vocabulary ({D, F, O}, {m}, {c}).
The arithmetic language has vocabulary ({=}, {succ, +, *}, {0})
Set theory has the vocabulary ({$\in$, =}, $\emptyset$, $\emptyset$)

2.4.2 Terms (p. 99)

The set of terms for a vocabulary is inductively defined by:

i) Any variable is a term.

ii) Any individual constant is a term.

iii) If t_1, ..., t_n are terms and f $\in$ F with arity n > 0, then f(t_1, ..., t_n) is a term.

Examples of Terms

Dog Language
x
c
m(c)
m(m(c))

Arithmetic
0
x
succ(x)

Set Theory
x, y, z, ... (no other terms)

2.4.3 Formulas (p. 100)

i) If P is an n-ary predicate symbol and t_1, ..., t_n are terms, then P(t_1, ..., t_n) is an atomic formula.

ii) The set of formulas is inductively defined by:

- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ and ψ are formulas, then $\phi \wedge \psi$, $\phi \vee \psi$, and $\phi \rightarrow \psi$ are formulas.
- If ϕ is a formula and x is a variable, then $\forall x\phi$ and $\exists x\phi$ are formulas.

2.4.4 Bound ond Free Variables

$\forall x(P(x) \rightarrow \exists z Q(x,y,z)) \rightarrow Q(x,y,z)$

P(x) is bound, because of the quantifier $\forall x$ (which in itself is bound). The x and z in Q(x,y,z) is also bound (also because of $\forall x$ and $\exists z$). y, however, is free! In the second Q(x,y,z) all variables are free (no quantifier).

Substition is notated as: $\phi[t/x]$ denotes the result of substituting t for all free occurances of x in ϕ. **Note:** No free variable in t must become bound in $\phi[t/x]$!

DAT060
2016-09-16
LV 3, Lecture 2

1 Free/Bound variables and substitutions

Given a formula $\phi := \exists x(\neg(x = y))$ and the substitutional operator $\phi[x/y]{:=}\exists x(\neg(x = x))$ we are given that we have something that is not equal to itselsf. This is not allowed, since x is not free for y, in ϕ.

2 Rules for Natural Deduction

All the rules for propositional logic still holds, but we add some additional rules for the quantifiers: $\forall$ and $\exists$.

2.1 $\forall$-elimination

$$\frac{\forall x\ \phi}{\phi[t/x]}\ \forall e$$

With the restriction that t must be free for x in ϕ.

Example (Dog language)**:**
$\forall x(D(x) \to L(x),\ D(c) \vdash L(c))$

1	$\forall x(D(x) \to L(x))$	premise
2	$D(c)$	premise
3	$D(c) \to L(c)$	$\forall e$ 1
4	$L(c)$	$\to e$ 2, 3

2.2 ∀-introduction

Introduce the "completely arbitrary" variable x_0.

$$\begin{array}{c} [x_0] \\ \vdots \\ \phi[x_0/x] \\ \hline \forall x\ \phi \end{array} \forall i$$

x_0 is a fresh variable, i.e. it occurs not outside the box, nor in ϕ. (Imagine a box around the nominator and the x_0.)

Example: Show that $\forall x\, P(x) \vdash \forall x\, (P(x) \vee Q(x))$

1	$\forall x\ P(x)$	premise
2	$\quad x_0\ P(x_0)$	$\forall e$ 1
3	$\quad P(x_0) \vee Q(x_0)$	$\vee i_1$ 2
4	$\forall x\ (P(x) \vee Q(x))$	$\forall i$ 2–3

Example: Show that $\vdash \forall x\, (P(x) \wedge Q(x) \to P(x))$

1	$\quad x_0$	Fresh Variable
2	$\quad\quad P(x)Q(x)$	assumption
3	$\quad\quad P(x_0)$	$\wedge e_1$ 2
4	$\quad P(x_0) \wedge Q(x_0) \to P(x_0)$	$\to i$ 2–3
5	$\forall x\ (P(x_0) \wedge Q(x_0) \to P(x_0))$	$\forall i$ 1–4

2.3 Equality

$$\frac{}{t = t} = i \qquad\qquad \frac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} = e$$

Example: Commutativity ($t_1 = t_2 \vdash t_2 = t_1$)

Put $\phi := (x = t_1)$ and use $=$-elimination.
Then $\phi[t_1/x] := t_1 = t_1$
$\phi[t_2/x] := t_2 = t_1$

$$\frac{t_1 = t_2 \qquad \dfrac{}{t_1 = t_1} = i}{t_2 = t_1} = e$$

2.4 ∃-Introduction

$$\frac{\phi[t/x]}{\exists x\ \phi}\ \exists i$$

Example: $\mathrm{P(c)} \vdash \exists x\, P(x)$

1	$P(c)$	premise
2	$\exists x\ P(x)$	$\exists i$ 1

Example: $\forall x\, \phi \vdash \exists x\, \phi$

1	$\forall x\ \phi$	premise
2	$\phi[x/x]$	$\forall e$ 1
3	$\exists x\ \phi$	$\exists i$ 2

2.5 ∃-Elimination

$$\frac{\exists x\ \phi \qquad \begin{array}{c}[x_0 \quad \phi[x_0/x]] \\ \vdots \\ \chi\end{array}}{\chi}\ \exists e$$

Example: $\forall x\, (P(x) \to Q(x)),\ \exists x\, P(x) \vdash \exists x\, Q(x)$

1	$\forall x\ (P(x) \to Q(x))$	premise
2	$\exists x\ P(x)$	premise
3	$\overline{x_0 \quad P(x_0)}$	assumption
4	$P(x_0) \to Q(x_0)$	$\forall e$ 1
5	$Q(x_0)$	$\to e$ 3,4
6	$\exists x\ Q(x)$	$\exists i$ 5
7	$\exists x\ Q(x)$	$\exists e$ 2, 3–6

Example: $\forall x\, (\neg P(x)) \vdash \neg \exists x\, P(x)$

1	$\forall x\ (\neg P(x))$	premise
2	$\quad \exists x\ P(x)$	assumption
3	$\quad\quad x_0 \quad P(x_0)$	assumption
4	$\quad\quad \neg P(x_0)$	$\forall e$ 1
5	$\quad\quad \bot$	$\neg e$ 3, 4
6	$\quad \bot$	$\exists e$ 2, 3–5
7	$\neg\exists x\ P(x)$	$\rightarrow i$ 2–6

Theorem 2.13:
$\vdash\ \exists x\,\phi \leftrightarrow \neg(\forall x\,\neg\phi)$
$\vdash\ \forall x\,\phi \leftrightarrow \neg(\exists x\,\neg\phi)$
Note: Both proofs uses PBC.

DAT060
LV 3, Lecture 3
Assignment 2

1 Next Assignment

1.1 Syllogism

No human is immortal.
Socrates is human.
Socrates is not immortal.

A sentence is built up by:
Fix signature $\sum$ (symbols that we can use in our formula):
- function symbols - s
- predicate symbols - h, i
- for each symbol its arity (s: 0-arity, h: 1-arity, i: 1-arity)

No x ϕ[x] $\forall x \neg\phi$

$\forall x$ h(x) $\rightarrow \neg$i(x)
h(s)
$\neg$i(s)

1	$\forall x\, h(x) \rightarrow \neg i(x)$	premise
2	$h(s)$	premise
3	$h(s) \rightarrow \neg i(s)$	$\forall e$ 1
4	$\neg i(s)$	$\rightarrow$e 2, 3

1.2 $\neg\forall x\,\phi \vdash \exists x\,\neg\phi$

1	$\neg\forall x\,\phi$	premise
2	$\quad\neg\exists x\,\neg\phi$	assumption
3	$\quad\forall x\,\neg\neg\phi$	(*2) 2
4	$\quad\forall x\,\phi$	(*3) 3
5	$\quad\bot$	$\to$e 4, 1
6	$\exists x\,\neg\phi$	PBC

(*1):$\neg\forall x\,\phi \vdash \exists x\,\neg\phi$
(*2): $\neg\exists x\,\psi \vdash \forall x\,\neg\psi$
(*3): $\phi \leftrightarrow \psi$, $\forall x\,\phi \vdash \forall x\,\psi$
(*4): $\forall x\,\phi \vdash \exists x\,\phi$
(*5): $\forall x\,\phi \vdash \forall y\,\phi[y/x]$
(*6): $\chi[y/y] = \chi$
(*7): $\chi[y/x][z/y] = \chi[z/x]$

Proof of (*2):

1	$\forall x\,\phi \leftrightarrow \psi$	premise
2	$\forall x\,\phi$	premise [$\forall x\,\psi$]
3	$\quad x_0 \quad \phi[x_0/x]$	$\forall$e 2
4	$\quad\phi[x_0/x] \leftrightarrow \psi[x_0/x]$	$\forall$e 1
5	$\quad\phi[x_0/x] \to \psi[x_0/x]$	$\wedge e_1$ 4
6	$\quad\psi[x_0/x]$	$\to$e 3, 5
7	$\forall x\,\psi$	$\forall$i 3, 6

Proof of (*4):

1	$\forall x\,\phi$	premise
2	$\phi[x_0/x]$	$\forall$e 1 [for x_0]
3	$\exists x\,\phi$	$\exists$i 2

Proof of (*5):

1	$\forall x\,\psi$	premise
2	$\quad y \quad \psi[y/x]$	$\forall$e 1 [for y]
3	$\forall y\,\psi[y/x]$	$\forall$i 2–2 [on the unknown y]

1.3 Extra

$\forall x\, P(x) \vdash \forall x\, P(g(x))$

Predicate: P
Function symbol: g

1	$\forall x\, P(x)$	premise
2	$\vert\ x_0$	
3	$\vert\ P(g(x_0))$	$\forall$e 1, [g(x_0)/x]
4	$\forall x\, P(g(x))$	$\forall$i 2–3

2 Previous Assignment

2.1 Problem 1

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

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

CNF is given by:
$(\neg p \vee q) \wedge (p \vee \neg q) \wedge (p \vee q)$

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

p	q	r	$r \wedge \neg p$	$q \rightarrow r \wedge \neg p$	$p \vee (q \rightarrow r \wedge \neg p)$	$\neg r$	$\neg r \rightarrow p \vee (q \rightarrow r \wedge p)$
T	T	T	F	F	T	F	T
T	T	F	F	F	T	T	T
T	F	T	F	T	T	F	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	F	T
F	T	F	F	F	F	T	F
F	F	T	T	T	T	F	T
F	F	F	F	T	T	T	T

CNF is given by:
$(p \vee \neg q \vee r)$

2.2 Problem 2

The CNF for the table is:
$(\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge$
$(p \vee q \vee \neg r) \wedge (p \vee q \vee r)$

2.3 Problem 3

a) $p \rightarrow q \vdash p \vee q$

if p=q=F, the LHS evaluate to T while the RHS evaluate to F.

b) $p \rightarrow q \vee r \vdash (p \rightarrow q) \wedge (p \rightarrow r)$

The RHS evaluates to False if p=T and q (inclusive) or r = F. The LHS evaluates to True for all values of p, q, r except: p=T, q=r=F.

Hence:

- p=T, q=T, r=F
- p=T, q=F, r=T

both prooves that the sequent is invalid.

c) $\vdash (p \wedge q) \vee (p \rightarrow q)$

The RHS evaluates to False iff p=T and q=F.

2.4 Problem 4

The truth table looks like:

ϕ	ψ	$\phi \odot \psi$
T	T	F
T	F	T
F	T	T
F	F	T

a)
The operand $\odot$ has the same behaviour as the operation nand (not and, $\uparrow$). Using only our well known connectives, $\phi \odot \psi \equiv \neg(\phi \wedge \psi)$.

b)
In class we said that $\{\vee, \wedge, \neg\}$ is a complete set of connectives. (We also said that $\{\wedge/\vee, \neg\}$ is a complete set, but I'll show the former anyway.)

p	q	p$\wedge$q	(p $\odot$ q) $\odot$ (p $\odot$ q)	p $\vee$ q	(p $\odot$ p) $\odot$ (q $\odot$ q)	$\neg$p	p $\odot$ p	p $\rightarrow$ q	p $\odot$ (q $\odot$ q)
T	T	T	T	T	T	F	F	T	T
T	F	F	F	T	T	F	F	F	F
F	T	F	F	T	T	T	T	T	T
F	F	F	F	F	F	T	T	T	T

DAT060
LV 4, Lecture1

1 Models (Interpretations)

A mode for propositional logic is a valuation assigning a truth value to each atom. A model corresponds to a row in a truth table for a particular formula.

A formula is true in the model if its truth value is T.

(Def 2.14 p. 124)
A model M of a vocabulary {P, F, C} of predicate logic consists of:

1. A non empty set A, the universe of concrete values;
2. for each constant C, an element C^M in A
3. for each f ∈ F with arity n > 0, a function f^M: $A^n \rightarrow A$
4. for each p ∈P with arity n ≥ 0, a subset $p^M \subseteq A^n$ of n-tuples over A.

Example: A model of the dog language

Vocabulary:

P = {D, L, H} with arities 1, 1, 2.
F = {m} with arity 1
C = {c}

The intended interpretation M:

1. A is the set of all mammals.
2. c^M= Caro
3. m^M is the function which to each mammal gives its mother.
4. Predicates:
   - D^M= {x ∈ A | x is a dog}
   - L^M = {x ∈ A | x has four legs}
   - H^M= {(x,y) ∈ A^2 | x is heavier than y}

Usage:
is $\forall x\,(D(x) \to L(x))$ true in M?
No, there are many amputated dog legs out there.

is $\exists y \forall x (D(x) \to H(y, x))$ true in M?
Choose y to be an elephant, then the formula is true in M.

Example 2: Another model M' of the dog language

1. A = {0, 1}
2. $\mathrm{c}^{M'} = 0$
3. $\mathrm{m}^{M'}$ is the function defined by $\begin{cases} m^{M'}(0) = 0 \\ m^{M'}(1) = 0 \end{cases}$
4. Predicates:
   - $\mathrm{D}^{M'} = \{0\}$
   - $\mathrm{L}^{M''} = \{1\}$
   - $\mathrm{H}^{M'} = \emptyset$

Usage:
is $\forall x\,(D(x) \to L(x))$ true in M'?
No, let x=0, then D(x) is true but L(x) is false.

is $\exists y \forall x (D(x) \to H(y, x))$ true in M'?
Nothing can satisfy $\mathrm{H}^{M'}$ in M' since $\mathrm{H}^{M'}$ is the empty set. Therefore the expression is false for all objects in A, since $\mathrm{D}^{M'}$ is true for 0.

Example 3: Arithmetic

Vocabulary

P = {=}
F = {+, *, succ}
C = {0}

Let N be the standard model of arithmetic:

1. A = $\mathbb{N}$ = {0, 1, 2, ...}
2. $0^N = 0$
3. $+^N$ is addition, $*^N$ is multiplication, succ^N is the successor
4. $=^N$ is the equality between natural numbers

Usage:
$\forall x(\exists y\,(x = 2 * y) \vee \exists y\,(x = 2 * y + 1))$, where 1 = succ(0) and 2 = succ(succ(0))
This is true. Every number in $\mathbb{N}$ is either even or odd.

Example 4: A non intended model M for the vocabulary of arithmetic.

1. A is the set of mammals
2. 0^M is Caro
3. $+^M$(u, v) = $*^M$(u, v) = succ$^M(u)$= Caro for all u, v ∈ A
4. $=^M$ is interpreted as equality between mammals

Note: *ϕ_1, ..., $\phi_n \models \psi$ means that for all models in which ϕ_1, ..., ϕ_n is true, ψ is true. ϕ_1, ..., $\phi_n \models \psi$ iff ϕ_1, ..., $\phi_n \vdash \psi$. So, ϕ_1, ..., $\phi_n \vdash \psi \Rightarrow \phi_1$, ..., $\phi_n \models \psi$. The reversed is also true: ϕ_1, ..., $\phi_n \models \psi \Rightarrow \phi_1$, ..., $\phi_n \vdash \psi$.*

Is there a set of formulas Ax s.t: Ax $\vdash \phi$ iff ϕ is true in the standard model N of arithmetic?
No! That is what Gödel's incompleteness theorem shows.

2 Tarski's Definition of Truth for Formalized Languages

2.1 Def 2.17 p. 127

An environment for a univrse A is a function l: var → A from the set var of variables to A. l[x → a] is the environment which maps x to a and any other variable y to l(y).

2.2 Def 2.18 p.128

Let a vocabulary (P, F, C) be given and a model M for the vocabulary. We want to define what it means for a formula ϕ to be true in M given an environment l. We write this as M $\models_l \phi$ or $[|\phi|]^M_l$.

Let t be a term. We define $[|t|]^M_l$ by induction on t.

1. t is a variable x, and $[|x|]^M_l = l(x)$
2. t is some c^C and $[|c|]^M_l = c^M$
3. If $f \in$ F of arity n and t_1, ..., t_n are terms then: $[|f(t_1, ..., t_n)|]^M_l = f^M([|t_1|]^M_l, ..., [|t_n|]^M_l)$ so for each term t $[|t|]^M_l \in$ A.

DAT060
LV 4, Lecture 2

1 Tarski's Truth Definition

See previous lecture, def 2.8 for the whole definition.

We will define M $\models_l \phi$, meaning that ϕ is true in M given the environment l.

1. ϕ is P(t_1, ..., t_n) (the atomic case)
   M $\models_l$ P(t_1, ..., t_n) holds if $([|t_1|]^m_l$, ..., $[|t_n|]^m_l) \in P^M$
2. Propositional symbols:
   M $\models_l \phi_1 \wedge \phi_2$ holds if M $\models_l \phi_1$ and M $\models_l \phi_2$ holds
   M $\models_l \neg\phi_1$ holds if M $\models_l \phi_1$ does not hold
   .
   .
   .
3. The universal quantifier:
   M $\models_l \forall x\, \phi$ holds if for all a $\in \forall$, M $\models_{l[x\to a]} \phi$ holds
4. The existential quantifier:
   M $\models_l \exists x\, \phi$ holds if there exist an a $\in$ A s.t. M $\models_{l[x\to a]} \phi$ holds

1.1 Example

The sentence "It is raining." is true iff it is raining.

To each formula ϕ you can associate a unique natural number $\ulcorner\phi\urcorner$, the Gödel number of the formula.

Tarski's undefineability theorem of truth says that there is no formula T s.t. N$\models$T($\ulcorner\phi\urcorner$) $\Leftrightarrow$ N $\models \phi$ (N is the standard model of arithmetic).

2 Def 2.20

Let Γ be a set of sentences (closed formulas w/o free variables). $\Gamma \models \phi$ holds if, for all models, if all formulas in Γ are true in the model, then ϕ is true. (ϕ is also a sentence)

3 Theorem

If Γ is consistent ($\Gamma \nvdash \bot$) then Γ has a model.

4 Gödel's Incompleteness Theorem

An axiom system is a set of formulas (Ax) s.t. it is decidable if a formula is in Ax or not. "You recognize an axiom when you see it." Let N be the standard model of arithmetic. Gödel then states that there is no axiom system such that:

$$N \models \phi \Leftrightarrow Ax \vdash \phi$$

4.1 Idea of Proof

There is a formula G_{Ax}, in arithmetic, which intuitively says: "This formula cannot be proved from Ax." From this, one can prove: $Ax \nvdash G_{Ax}$ and $Ax \nvdash \neg G_{Ax}$.

Note that N $\models$ G_{Ax} holds!

5 Show that $\exists x\,(P(x) \to Q(x)) \nvdash \forall x\,(P(x) \to Q(x))$

Soundness tells us that: $\Gamma \vdash \phi \Rightarrow \Gamma \models \phi$. To show that $\Gamma \nvdash \phi$ we will give a model M such that M $\models \psi$ for all $\psi \in \Gamma$ and M $\not\models \psi$.

A = {0, 1} (the domain)
P^M = {0}
Q^M = {$\emptyset$}

M $\models \exists x\,(P(x) \to Q(x))$ $\Leftrightarrow$ There exists a $\in$ A s.t. M $\models_{l[x\to a]}$P(x) $\to$ Q(x). $\Leftrightarrow$ There exists a $\in$ A s.t. (M $\models_{l[x\to a]}$ P(x) $\Rightarrow$ M $\models_{l[x\to a]}$ Q(x)).

Choose a to be 1.

M $\models \forall x\,(P(x) \to Q(x))$ $\Leftrightarrow$ For all a $\in$ A, M $\models_{l[x\to a]}$ $P(x) \to Q(x)$

Choose a to be 0, then M $\models_{l[x\to a]}$ $P(x)$ holds, but M $\models_{l[x\to a]}$ $Q(x)$ does not. Q is always false, due to being the empty set. So $M \not\models \forall x\,(P(x) \to Q(x))$.

DAT060
LV 4, Lecture 3
Assignment 3

1 $\mathbf{a = b, P(a,a) \vdash P(a,b)}$

1	$a = b$	premise
2	$P(a, a)$	premise
3	$P(a, b)$	=e, t=a, u=b, $\phi = P(a, x)$

2 $\forall x\, f(g(x)) = x \vdash \forall x \forall y\, g(x) = g(y) \to x = y$

1	$\forall x\, f(g(x)) = x$	premise
2	$\quad x_0$	
3	$\quad\quad y_0$	
4	$\quad\quad\quad g(x_0) = g(y_0)$	assumption
5	$\quad\quad\quad f(g(x_0)) = f(g(x_0))$	=i $f(g(x_0))$
6	$\quad\quad\quad f(g(x_0)) = f(g(y_0))$	=e 4, 5 $(t = g(x_0), u = g(y_0), \phi = f(g(x_0)) = f(x)$
7	$\quad\quad\quad f(g(x_0)) = x_0$	$\forall$e 1 (x_0)
8	$\quad\quad\quad f(g(y_0)) = y_0$	$\forall$e 1 (y_0)
9	$\quad\quad\quad x_0 = f(g(y_0))$	=e 7, 6 $(t = f(g(x_0)), u = x_0, \phi = x = f(g(y_0)))$
10	$\quad\quad\quad x_0 = y_0$	=e 8, 9 $(t = f(g(y_0)), u = y_0, \phi = (x_0 = x))$
11	$\quad\quad g(x_0) = g(y_0) \to x_0 = y_0$	$\to$i 4–10
12	$\quad \forall y\, g(x_0) = g(y) \to x_0 = y$	$\forall$i 3–11
13	$\forall x \forall y\, g(x) = g(y) \to x = y$	$\forall$i 2–12

Beware of the direction in the elimination rule. If you have t=u and $\phi[t/x]$ you end up with $\phi[u/x]$.

3 Problem 2.4.3

Given that: P is a binary predicate, M has carrier A, $\mathrm{P}^M \subseteq AxA$
Note: $\neg P(x,x) \equiv P(x,x) \to \bot$

i) $M \models_l \forall x\, \neg P(x,x)$
iff for all a ∈ A, $M \underset{l[x\to a]}{\models} \neg P(x,x)$
iff for all a ∈ A, $M \underset{l[x\to a]}{\models} P(x,x) \Rightarrow (M \underset{l[x\to a]}{\models} \bot)$
iff for all a ∈ A, (a,a) ∈ P^M ⇒ False
iff for all a ∈ A, (a,a) ∉ $\mathrm{P}^M \square$

$\mathrm{P}^M = \emptyset$ and (a,a) ∉ ∅

ii) N$\not\models_l \forall x\, \neg P(x,x)$ N has carrier B, $\mathrm{P}^N \subseteq BxB$
(N$\models_l \forall x\, \neg P(x,x)) \Rightarrow$F
(forall a ∈ B, (a,a) ∉ P^N)

Let B = {0}, $\mathrm{P}^N = \{(0,0)\}$

(0,0) ∈ P^N contradicts (forall a ∈ B, (a,a) ∉ P^N) hence we have provet what we wanted.

4 Problem 11 a)

Γ:=$\{\forall x\, \neg S(x,x),\ \exists(x)\, P(x),\ \forall x \exists y\, S(x,y),\ \forall x\, (P(x) \to \exists y\, S(y,x))\}$
Show that Γ is consistent, i.e. show M for all $\phi \in \Gamma$, M$\models \phi$.

A = {0, 1}
$\mathrm{P}^M = \{0\}$
$\mathrm{S}^M = \{(0,1),(1,0)\}$

i) M$\models \forall x\, \neg S(x,x)$ can be shown by: Let a ∈ A, (a,a)$\notin \mathrm{S}^M \Rightarrow \bot$. This has two cases:

1. a = 0, (0,0) ∉ S^M
2. a = 1, (1,1) ∉ S^M

ii) M $\models \exists x\, P(x)$
iff exists a ∈ A, a ∈ P^M
Let a = 0, then 0 ∈ P^M

iii) M$\models \forall x \exists y\, S(x,y)$
forall a ∈ A, exists b ∈ A, (a,b) ∈ S^M

1. a = 0, let b = 1, (0,1) $\in S^M$
2. a = 1, let b = 0, (1,0) $\in S^M$

Assignment 3

Problem 2

a) $\forall x \forall y\, R(x, y) \vdash \forall x\, R(x, x)$

1	$\forall x \forall y\, R(x, y)$	premise
2	$\quad x_0$	
3	$\quad\quad \forall y\, R(x_0, y)$	$\forall e(1, x_0)$
4	$\quad\quad R(x_0, x_0)$	$\forall e(3, x_0)$
5	$\forall x\, R(x, x)$	$\forall i$ 2–4

b) $\forall x\, (P(x) \to S) \vdash (\exists x\, P(x)) \to S$

1	$\forall x (P(x) \to S)$	premise
2	$\quad \exists x\, P(x)$	assumption
3	$\quad\quad x_0$	
4	$\quad\quad P(x_0)$	assumption
5	$\quad\quad P(x_0) \to S$	$\forall e(1, x_0)$
6	$\quad\quad S$	$\to$e 4, 5
7	$\quad S$	$\exists e$ 2, 3–6
8	$\exists x\, (P(x)) \to S$	$\to i$ 2–7

c) $\exists x \forall y\, R(x, y) \vdash \forall y \exists x\, R(x, y)$

1	$\exists x \forall y\, R(x, y)$	premise
2	$\quad y_0$	
3	$\quad\quad x_0$	
4	$\quad\quad \forall y\, R(x_0, y)$	assumption
5	$\quad\quad\quad R(x_0, y_0)$	$\forall e(4, y_0)$
6	$\quad\quad \exists x\, R(x, y_0)$	$\exists i(5, x_0)$
7	$\quad \exists x\, R(x, y_0)$	$\exists e$ 1, 3–6
8	$\forall y \exists x\, R(x, y)$	$\forall i$ 2–7

d) $\exists x\,(P(x) \lor Q(x)),\ \forall x\,(Q(x) \to F(x)) \vdash \exists x\,(P(x) \lor F(x))$

1	$\exists x\,(P(x) \lor Q(x))$	premise
2	$\forall x\,(Q(x) \to F(x))$	premise
3	$\quad x_0$	
4	$\quad P(x_0) \lor Q(x_0)$	assumption
5	$\quad Q(x_0) \to F(x_0)$	$\forall e$ 2
6	$\quad\quad Q(x_0)$	assumption
7	$\quad\quad F(x_0)$	$\to$e 5, 6
8	$\quad\quad P(x_0) \lor F(x_0)$	$\lor i_2$ 7
9	$\quad\quad P(x_0)$	assumption
10	$\quad\quad P(x_0) \lor F(x_0)$	$\lor i_1$ 9
11	$\quad P(x_0) \lor F(x_0)$	$\lor e$ 4, 6–8, 9–10
12	$\quad \exists x\,(P(x) \lor F(x))$	$\exists i$ 11
13	$\exists x\,(P(x) \lor F(x))$	$\exists e$ 1, 3–12

Problem 3

$\exists x\,(P(x) \to \forall y\,P(y))$

Solution 1

(1):$\forall y\,P(y) \vdash \forall x\,(P(x) \to \forall y\,P(y)) \vdash \exists x\,(P(x) \to \forall y\,P(y))$
(2):$\neg\forall y\,P(y) \vdash \exists y\,\neg P(y) \vdash \exists y\,(P(y) \to \forall y\,P(y))$

1	$\forall y\ P(y) \lor \neg\forall y\ P(y)$	LEM
2	$\quad \forall y\ P(y)$	assumption
3	$\quad \exists x\ (P(x) \to \forall y P(y))$	(1)
4	$\quad \neg\forall y\ P(y)$	assumption
5	$\quad \exists x\ (P(x) \to \forall y\ P(y))$	(2)
6	$\exists x\ (P(x) \to \forall y\ P(y))$	$\lor$e (1, 2–3, 4–5)

\;

1	$\forall y\ P(y)$	premise
2	$\quad P(x_0)$	assumption
3	$\quad \forall y\ P(y)$	copy(1)
4	$P(x_0) \to \forall y\ P(y)$	$\to$i (2–3)
5	$\exists x\ (P(x) \to \forall y\ P(y))$	$\exists(4, x_0)$

1	$\neg\forall y\ P(y)$	premise
2	$\exists y\ \neg P(y)$	lemma from last time
3	$\quad \neg P(y_0)$	assumption
4	$\quad\quad P(y_0)$	assumption
5	$\quad\quad \bot$	$\neg$e
6	$\quad\quad \forall y\ P(y)$	$\bot$e
7	$\quad P(y_0) \to \forall y\ P(y)$	$\to$i 4–6
8	$\quad \exists x\ (P(x) \to \forall y\ P(y))$	$\exists i(7, x_0)$
9	$\exists\ (P(x) \to \forall y\ P(y))$	$\exists e(3 - -8, y_0)$

Solution 2 (What happened here?)

1	$\forall y\ P(y) \lor \neg\forall y\ P(y)$	LEM
2	$\forall y\ P(y) \lor \exists y\ \neg P(y)$	
3	$\exists y \forall y\ P(y) \lor \neg P(y)$	weird lemma
4	$\exists y\ P(y) \to \forall x P(x)$	also weird lemma

Solution 3

1	$\neg\exists x\ (P(x) \to \forall y\ P(y))$	assumption
2	$x_0 \quad P(x_0)$	assumption
3	y_0	
4	$\neg P(y_0)$	assumption
5	$P(y_0)$	assumption
6	$\bot$	$\neg$e 4, 5
7	$\forall y\ P(y)$	$\bot$ i 6
8	$P(y_0) \to \forall y\ P(y)$	$\to$i 5–7
9	$\exists x\ (P(x) \to \forall y\ P(y))$	$\exists$i 8
10	$\bot$	$\neg$e 1, 9
11	$\neg\neg P(y_0)$	$\neg$i 4–10
12	$P(y_0)$	$\neg\neg$e 11
13	$\forall y\ P(y)$	$\forall$i 3–12
14	$P(x_0) \to \forall y\ P(y)$	$\to$i 2–13
15	$\exists x\ (P(x) \to \forall y\ P(y))$	$\exists$i 14
16	$\bot$	$\neg$e 1, 15
17	$\neg\neg\exists x\ (P(x) \to \forall y\ P(y))$	$\neg i$ 1–16
18	$\exists x\ (P(x) \to \forall y\ P(y))$	$\neg\neg$e 17

DAT060
LV 5, Lecture 1

1 First order formula - ϕ

if ϕ has no free variables we say that ϕ is a sentence.
ϕ is valid in $\mathcal{M}$ if $\mathcal{M} \models \phi$.

In order to define this, we need $\mathcal{M} \models_l \phi$.
if ϕ has no free variables: l: var $\to$ A
if ϕ is a sentence this does not depend on l ($\mathcal{M} \models \phi$)

ϕ is a tautology if $\models \phi$, it means that ϕ is valid in all possible models: "$\forall \mathcal{M}\, \mathcal{M} \models \phi$" which is a very complex notion since we quantify over *all* possible models! We do however also say that $\vdash \phi$ is easier to prove, and $\models \phi \Leftrightarrow \vdash \phi$. $\Rightarrow$ is the completness, $\Leftarrow$ is the soundness.

2 First order theory - T (in a given language)

T is a *set* of some sentences.

$\mathcal{M} \models T$ means that $\mathcal{M} \models \phi$ for all ϕ in T.
T $\models \psi$ if for all models $\mathcal{M}$, $\mathcal{M} \models T \to \mathcal{M} \vdash \psi$
If T = $\{\phi_1,..., \phi_n\}$ (finite) ϕ_1, ..., $\phi_n \models \psi$, $\phi_1,..., \phi_n \vdash \psi$
we can derive ψ with hypotheses $\phi_1,..., \phi_n$.

In general, T will be infinite and T $\vdash \psi$ will mean that we can find ϕ_1, ..., $\phi_n \in$ T s.t. $\phi_1,..., \phi_n \vdash \psi$.

We still have T $\models \psi \Leftrightarrow$ T $\vdash \psi$ even if T is infinite!

2.1 Theory of equivalence relations

$\mathcal{F} = \emptyset$
$\mathcal{P} = \{R\}$ (R is a relation symbol of arity 2)

A model, $\mathcal{M}$, of this language is a set A and $R^{\mathcal{M}} \subseteq A^2$. This model can

be seen as a graph!

Example
A is a set with 4 elements {a,b,c,d} with connections between: a and b, a and c, c and d.
T has two sentences: $\phi_1 : \forall x\, R(x,x)$ and $\phi_2 : \forall x \forall y \forall z\, ((R(x,z) \wedge R(y,z)) \rightarrow R(x,y))$.

T = $\{\phi_1, \phi_2\}$

From this we can derive: T $\vdash \phi_3$, $\phi_3 : \forall x \forall y (R(x,y) \rightarrow R(y,x))$. I.e. $\phi_1, \phi_2 \vdash \phi_3$. We would prove $\phi_1, \phi_2 \vdash \phi_3$ by showing: $\phi_1, \phi_2 \models \phi_3$.

Example (Transitive)
T $\vdash \phi_4$, $\phi_4 : \forall x \forall y \forall z\, ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$

if $\mathcal{M} \models T$ then $\mathcal{M} \models \phi_4$

Civen elements a, b, c $\in$ A we see that a is connceted to b which is turn is connected to c. We want to show that a is connected to c. We know that $\mathcal{M} \models \phi_3$ so we know that c is connected to b. Using ϕ_2 we can show that a is connected to c.

We have seen that T $\vdash \phi_3$ and T $\vdash \phi_4$, so given T'=$\{\phi_1, \phi_3, \phi_4\}$ we should be able to show that T' $\vdash \phi_2$. T and T' satisfy: T' $\vdash \psi$ if $\psi \in$ T, T $\vdash \phi$ if $\phi \vdash$ T. Hence, T $\vdash \phi \Leftrightarrow$T' $\vdash \phi$ for any formula ϕ of the language. T and T' are different axiomatizations of the same theory.

2.2 Theory of order relations

$\mathcal{F} = \emptyset$
$\mathcal{P} = \{R\}$ (R is a relation symbol of arity 2

Preorder: $\phi_1 : \forall x\, R(x,x)$, ϕ_4: $\forall x \forall y \forall z\, ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$

Poset: $\phi_5 : \forall x \forall y\, (R(x,y) \wedge R(y,x)) \rightarrow x = y$

$\mathbb{N}$, $\mathrm{R}^{\mathcal{M}}(p,q)$ $p \leq q$
$\mathbb{Z}$
$\mathbb{Q}$
$\mathbb{R}$
All models of ϕ_1, ϕ_4, ϕ_5

($\mathbb{N}$,$\leq$), ($\mathbb{Z}$,$\leq$), ($\mathbb{Q}$,$\leq$), ($\mathbb{R}$,$\leq$)

A = $\mathbb{N}$ $\mathrm{R}^{\mathcal{M}}$ is $p \leq q$

Do we have:

1. $(\mathbb{N},\leq) \models \phi \Leftrightarrow (\mathbb{Z},\leq) \models \phi$
2. $(\mathbb{Z},\leq) \models \phi \Leftrightarrow (\mathbb{Q},\leq) \models \phi$
3. $(\mathbb{Q},\leq) \models \phi \Leftrightarrow (\mathbb{R},\leq) \models \phi$

For 2. we ask ourselves:
Can we express the difference between $(\mathbb{Z},\leq)$ and $(\mathbb{Q},\leq)$ in a first-order way?

$(\mathbb{Q},\leq)$: $\forall x \forall y\, R(x,y) \wedge x \neq y \rightarrow \exists z\, (R(x,z) \wedge R(z,y))$. This holds for $\mathbb{Q}$ but not for $(\mathbb{Z},\leq)$. This is because if we have to rational numbers, we can always squeeze in a number in between them. This is not true for natural or whole numbers.

For 1. we ask:
Can we find $(\mathbb{N},\leq) \models \phi$ and $(\mathbb{Z},\leq) \nvDash \phi$?

$\psi = \exists x \forall y\, \neg(R(y,x) \wedge x \neq y)$
0 for all n, we cannot have $n \leq 0 \wedge n \neq 0$ so $(\mathbb{N},\leq) \models \phi$
for the model $(\mathbb{Z},\leq) \models \phi$ for any n, n-1 $\leq$ n, n-1 $\neq$ n

For 3:
One can show that 3 holds since there is no way to make a difference between the rational and the real numbers using a "first order way".

All these models satisfy $\phi_6 = \forall x \forall y\, R(x,y) \vee R(y,x)$ (linnearity).

We now ask: $\phi_1, \phi_4, \phi_5 \models \phi_6$?
One can show that this is not the case, but: $\phi_1, \phi_4, \phi_5 \nvDash \phi_6$ and in order to show this one must give a model of ϕ_1, ϕ_4, ϕ_5 that is not a model of ϕ_6. A model that does this could be a model that relate every element to itself.

A = $\mathbb{N}$, (a,b)$\in R^{\mathcal{M}'}$ a=b

$\mathcal{M}' \models \phi_1 \wedge \phi_4 \wedge \phi_5$ but $\mathcal{M}' \nvDash \phi_6$

By soundness $\phi_1, \phi_4, \phi_5 \nvdash \phi_6$. There is no derivation of ϕ_6 from ϕ_1, ϕ_4, ϕ_5.

Is ϕ derivable from T? I.e. T $\vdash \phi \Leftrightarrow$ T $\models \phi$? Is there an algorithm for solving this question? This is called the *decision problem,* and there is no algorithm to solve this.

2.3 Theories about arithmetic

- Theory of 0 and n+1 (complete)
- Presburger arithmetic 0, n+1, n+m (in complete)
- Peano arithmetic 0, n+1, n+m, n*m (not decidable)

2.3.1 When is T decidable?

We have an algorithm which decides. Given an input (a sentence) ϕ we either get $T \vdash \phi$ or $T \nvdash \phi$.

2.3.2 Theory of 0 and n+1

Language: zero (constant), S(x) (functional symbol of arity 1), no relations
Model: a = zero$^{\mathcal{M}} \in$ M, f = S$^{\mathcal{M}}$: A $\to$ A. A, a $\in$ A, f: A $\to$ A,
$\mathbb{N}$, 0, f: $\mathbb{N} \to \mathbb{N}$, n $\longmapsto$ n+1 is a particular model

$(\mathbb{N}, 0, S) \models \phi$

Can I find a theory T s.t. $T \vdash \phi \Leftrightarrow (\mathbb{N}, 0, S) \models \phi$?
We have two axioms:
$\phi_1 = \forall x\, zero \neq S(x)$
$\phi_2 = \forall x \forall y\, S(x) = S(y) \to x = y$
These formulas are valid in ($\mathbb{N}$, 0, S)

ϕ_1 tells us that a is not in the image of f
ϕ_2 tells us that f: A $\to$ A is injective

$T_0 = \{\phi_1, \phi_2\}$
Exercise:
Give a model of T_0 that is not a model of $\delta_1 = \forall x\, x \neq S(x)$, $\delta_2 = \forall x\, x \neq S(S(x))$, $\delta_3 = \forall x\, x \neq S(S(S(x)))$, ... We are missing $\psi = \forall x\, (x = 0 \vee \exists y\, (x = S(y)))$.

We have $T \vdash \phi \Leftrightarrow (\mathbb{N}, 0, S) \models \phi$.

Furthermore one can write an algorithm which decides $T \vdash \phi$. We can replace "thinking" by computations. We can do the same for addition.

Logic in Computer Science

For a given language $\mathcal{F}, \mathcal{P}$, a *first-order theory* is a set T of sentences (closed formulae) in this given language. The elements of T are also called *axioms* of T.

A model of T is a model $\mathcal{M}$ of the given language such that $\mathcal{M} \models \psi$ for all sentences ψ in T.

$T \vdash \varphi$ means that we can find $\psi_1, \ldots, \psi_n$ in T such that $\psi_1, \ldots, \psi_n \vdash \varphi$.
$T \models \varphi$ means that $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$ of T.

The generalized form of *soundness* is that $T \vdash \varphi$ implies $T \models \varphi$ and *completness* is that $T \models \varphi$ implies $T \vdash \varphi$.

If T is a finite set $\psi_1, \ldots, \psi_n$ this follows from the usual statement of soundness ($\vdash \delta$ implies $\models \delta$) and completness ($\models \delta$ implies $\vdash \delta$). Indeed, in this case, we have $T \vdash \varphi$ iff $\vdash (\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi$ and $T \models \varphi$ iff $\models (\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi$.

Theory of equivalence relations

The language is $\mathcal{P} = \{E\}$, binary relation, and $\mathcal{F} = \emptyset$. The axioms are

$$\forall x.\ E(x,x) \qquad\qquad \forall x\ y\ z.\ (E(x,z) \wedge E(y,z)) \to E(x,y)$$

We can then show $T \vdash \forall x\ y. E(x,y) \to E(y,x)$ and $T \vdash \forall x\ y\ z.\ (E(x,y) \wedge E(y,z)) \to E(x,z)$.

Theory about orders

The theory of *strict order*. The language is $\mathcal{P} = \{R\}$, binary relation, and $\mathcal{F} = \emptyset$. The axioms are

$$\forall x. \neg R(x,x) \qquad\qquad \forall x\ y\ z.\ (R(x,y) \wedge R(y,z)) \to R(x,z)$$

We can add equality and get the theory T_{lin} of *linear* orders

$$\forall x\ y.\ (x \neq y) \to (R(x,y) \vee R(y,x))$$

Models are given by the usual order on $\mathbb{N}, \mathbb{Q}, \mathbb{R}$. The model of rationals $(\mathbb{Q}, <)$ also satisfies

$$\psi_1 = \forall x. \exists y.\ R(x,y) \qquad \psi_2 = \forall x. \exists y.\ R(y,x) \qquad \psi_3 = \forall x\ y.\ R(x,y) \to \exists z.\ R(x,z) \wedge R(z,y)$$

It can be shown that we have $(\mathbb{Q}, <) \models \varphi$ iff $(\mathbb{R}, <) \models \varphi$ iff $T_{lin}, \psi_1, \psi_2, \psi_3 \vdash \varphi$ and furthermore, there is an algorithm to decide whether $(\mathbb{Q}, <) \models \varphi$ holds or not.

The theory of *preorder* has for axioms

$$\forall x. R(x,x) \qquad\qquad \forall x\ y\ z.\ (R(x,y) \wedge R(y,z)) \to R(x,z)$$

and for the theory of *poset* is this theory together with the antisymmetry

$$\forall x\ y.\ (R(x,y) \wedge R(y,x)) \to x = y$$

A poset is *linear* if it also satisfies the axiom

$$\forall x\ y.\ R(x,y) \vee R(y,x)$$

$(\mathbb{Q}, \leqslant)$ and $(\mathbb{R}, \leqslant)$ are two linear posets that are not isomorphic but they satisfy the same first-order formula. Furthermore we can decide whether $(\mathbb{Q}, \leqslant) \vdash \varphi$ holds or not.

Theory about arithmetic

The language is $\mathcal{F} = \{\mathsf{zero}, \mathrm{S}\}$ and $\mathcal{P} = \emptyset$, but we have equality.

The first theory T_0 is

$$\forall x.\mathsf{zero} \neq \mathrm{S}(x) \qquad \forall x\ y.\mathrm{S}(x) = \mathrm{S}(y) \to x = y$$

A model of this theory is a set A with a constant $a \in A$ and a function $f \in A \to A$ such that f is injective and a is not in the image of f.

A particular model $\mathbb{N}$ is given by the set of natural numbers and $0 \in \mathbb{N}$ and the successor function s on $\mathbb{N}$.

The formulae $\delta_1 = \forall x.x \neq \mathrm{S}(x)$, $\delta_2 = \forall x.x \neq \mathrm{S}(\mathrm{S}(x)), \ldots$ are not provable in T_0 but are valid in the model $(\mathbb{N}, 0, \mathrm{s})$. The formula $\psi = \forall x.x = 0 \vee \exists y.(x = \mathrm{S}(y))$ is not provable in $T_0, \delta_1, \delta_2, \ldots$ but is also valid in the model $(\mathbb{N}, 0, \mathrm{s})$. We can look at the possible shape of the models of $T_0, \delta_1, \delta_2, \ldots$ Such a model is a disjoint union of copies of $\mathbb{N}$ and $\mathbb{Z}$ and it there are several copies of $\mathbb{N}$ the formula ψ will not be satisfied.

It can be shown that we have $(\mathbb{N}, 0, \mathrm{s}) \models \varphi$ iff $T_0, \delta_1, \delta_2, \ldots, \psi \vdash \varphi$ and furthermore, there is an algorithm to decide $(\mathbb{N}, 0, \mathrm{s}) \models \varphi$. The models of $T_0, \delta_1, \delta_2, \ldots, \psi$ consist of *one* copy of $\mathbb{N}$ and zero or several copies of $\mathbb{Z}$

Presburger arithmetic

We add the binary function symbol $(+)$ and add to T_0 the axioms

$$\forall x.\ x + \mathsf{zero} = x \qquad \forall x\ y.\ x + \mathrm{S}(y) = \mathrm{S}(x + y)$$

and the induction schema

$$\forall y_1 \ldots y_m.\varphi(y_1, \ldots, y_m, \mathsf{zero}) \wedge \forall x.(\varphi(y_1, \ldots, y_m, x) \to \varphi(y_1, \ldots, y_m, \mathrm{S}(x))) \to \forall z.\varphi(y_1, \ldots, y_m, z)$$

The resulting theory PrA is called *Presburger arithmetic*. It can be shown that $(\mathbb{N}, 0, \mathrm{s}, +) \models \varphi$ iff $PrA \vdash \varphi$ and there is an algorithm to decide $(\mathbb{N}, 0, \mathrm{s}, +) \models \varphi$.

Peano arithmetic

We add the binary function symbol $(\cdot)$ and add to PrA the axioms for multiplication

$$\forall x.\ x \cdot \mathsf{zero} = \mathsf{zero} \qquad \forall x\ y.\ x \cdot \mathrm{S}(y) = x \cdot y + x$$

with the induction schema, where the formula $\varphi(y_1, \ldots, y_m, x)$ can also used multiplication. The resulting theory PA is called *Peano arithmetic*. It has been shown by Gödel that PA is *incomplete*: there is a formula φ such that $(\mathbb{N}, 0, \mathrm{s}, +, \cdot) \models \varphi$ but we don't have $PA \vdash \varphi$.

Furthermore $(\mathbb{N}, 0, \mathrm{s}, +, \cdot) \models \varphi$ is undecidable (there is no algorithm to decide $\mathbb{N} \models \varphi$) and there is *no* effective way to enumerate all sentences φ valid in the model $(\mathbb{N}, 0, \mathrm{s}, +, \cdot)$.

The decision problem

The *decision problem* (Hilbert-Ackermann 1928) is the problem of deciding if a sentence in a given language is provable or not.

More generally the problem is to decide if we have $\psi_1, \ldots, \psi_n \vdash \varphi$ or not.

There are special cases where this problem has a positive answer.

A general method is to apply the following remark: we have $\psi_1, \ldots, \psi_n \vdash \varphi$ iff the following theory $\psi_1, \ldots, \psi_n, \neg\varphi$ has *no* models. This follows from soundness and completeness.

Bernays-Schönfinkel decidable case

This is the particular case where $\mathcal{F}$ has only *constant* symbols and all formulae $\psi_1, \ldots, \psi_n, \varphi$ are of the form $\forall y_1 \ldots y_m.\delta$ or $\exists y_1 \ldots y_m.\delta$ where δ is quantifier-free.

In this case the following algorithm, that I illustrate on some examples, gives a way to decide whether $\psi_1, \ldots, \psi_n, \neg\varphi$ has a model or not. (If it has a model, it always has a *finite* model.) In this way, we decide whether $\psi_1, \ldots, \psi_n \vdash \varphi$ holds or not.

We take the example

$$T_1 = \exists x.(P(x) \wedge \neg M(x)), \exists y.(M(y) \wedge \neg S(y)), \forall z.(\neg P(z) \vee S(z))$$

The first step is to eliminate the existential quantifiers by introducing constants

$$T_2 = P(a) \wedge \neg M(a),\ M(b) \wedge \neg S(b), \forall z.(\neg P(z) \vee S(z))$$

It should be clear that T_1 has a model iff T_2 has a model.

The second step is to eliminate the universal quantifiers by instantiating on all constants

$$T_3 = P(a) \wedge \neg M(a),\ M(b) \wedge \neg S(b),\ \neg P(a) \vee S(a),\ \neg P(b) \vee S(b)$$

In this way we find a model with two elements $P(a), \neg M(a), S(a), M(b), \neg S(b), \neg P(b)$.

This implies that $\exists x.(P(x) \wedge \neg M(x)), \exists y.(M(y) \wedge \neg S(y)) \vdash \exists z.(P(z) \wedge \neg S(z))$ is *not* valid.

Other examples

$\forall x.\neg R(x,x) \vdash \forall x\ y.(R(x,y) \rightarrow \neg R(y,x)$ is not valid since we find a model of

$$T_1 = \forall x.\neg R(x,x),\ \exists x\ y.\ R(x,y) \wedge R(y,x)$$

by eliminating existentials

$$T_2 = \forall x.\neg R(x,x),\ R(a,b) \wedge R(b,a)$$

and then universals

$$T_3 = \neg R(a,a),\ \neg R(b,b),\ R(a,b) \wedge R(b,a)$$

and we get a counter-model with two elements.

On the other hand $\forall x\ y.(R(x,y) \rightarrow \neg R(y,x) \vdash \neg R(x,x)$ is valid, since if we try to find a model of

$$T_1 = \forall x\ y.(R(x,y) \rightarrow \neg R(y,x)),\ \exists x.R(x,x)$$

by eliminating existentials

$$T_2 = \forall x\ y.(R(x,y) \rightarrow \neg R(y,x)),\ R(a,a)$$

and then universals

$$T_3 = R(a,a) \rightarrow \neg R(a,a),\ R(a,a)$$

we should have $R(a,a)$ and $\neg R(a,a)$ and we cannot find a counter-model.

Theory of cyclic order

(Not covered in the lecture, but a nice example of a theory and of the use of the Bernays-Schönfinkel algorithm.)

A *cyclic order* is a way to arrange a set of objects in a circle (examples: seven days in a week, twelve notes in the chromatic scale, ...). The language is $\mathcal{P} = \{S\}$ which is a *ternary* predicate symbol and the first 3 axioms are

$$\psi_1 = \forall x\ y\ z.S(x,y,z) \to S(y,z,x) \qquad \psi_2 = \forall x\ y\ z.S(x,y,z) \to \neg S(x,z,y)$$

$$\psi_3 = \forall x\ y\ z\ t.(S(x,y,z) \wedge S(x,z,t)) \to S(x,y,t)$$

One can then use the Bernays-Schönfinkel algorithm to show automatically that these axioms are *independent*: we don't have $\psi_1, \psi_2 \vdash \psi_3$ or $\psi_2, \psi_3 \vdash \psi_1$ or $\psi_3, \psi_1 \vdash \psi_2$.

The last axiom of the theory of cyclic order uses equality

$$\psi_4 = \forall x\ y\ z.(x \neq y \wedge y \neq z \wedge z \neq x) \to S(x,y,z) \vee S(x,z,y)$$

The extension of the Bernays-Schönfinkel algorithm to equality is possible by axiomatising the equality relation. (This was first done by Ramsey, 1928, by another method.)

DAT060
LV 5, Lecture 2

1 First-Order Theory (Cont.) $\mathbb{N}$= {0, 1, 2, ...}

Given:

One constant: zero

One function symbol: S(x)

We can describe $\mathbb{N}$ using a model for this language. We have to give a meaning to the symbols of this language. A model is made from a set (A) elements in that set ($a \in A$) and functions (f: $A \to A$).

1.1 Example

M_1: A = $\mathbb{N}$, a = 0, f: n $\mapsto$ n+1
M_2: A = $\mathbb{Q}$, a = 1, f: x $\mapsto$ n+1
M_3: A = [0, 1], a = 1, f: x $\mapsto \frac{x}{2}$
M_4: A =]0 ,1], a = 1, f: x $\mapsto \frac{x}{2}$ *]0, 1] = {r / 0 < r ≤ 1}*

What we want is a theory T in this language s.t. $T \vdash \phi \longleftrightarrow M_1 \models \phi$. In particular $M_1 \models$ T.

Wy may ask more: M_1 is the only model of T.

This is not possible as such. If M_1 is 0, 1, 2, 3, 4 using S(x) as the succ function between each number, We can always take a "copy" of M_1 and call it M'$_1$ but instead of A = $\mathbb{N}$ we use A = {1, 2, 3, ...}, a = 1, f: n $\mapsto$ n+1. $M_1 \models \phi \longleftrightarrow$ M'$_1 \models \phi$.

M"$_1$: 1, $\frac{1}{2}$, $\frac{1}{4}$, ... we can have A = {1, $\frac{1}{2}$, $\frac{1}{4}$, ...}, a=1, f: x$\mapsto \frac{x}{2}$.

M"$_1 \models \phi \longleftrightarrow$ M'$_1 \models \phi$

First-order logic can only describe a "structure" and not talk abou the "nature" of elements. It is an abstract data type. We can ask instead: Any model $M \models T$ is a "copy" (isomorphic) of M_1. Even this is not possible.

1.2 Example

T_0: $\begin{cases} \forall x \, \text{zero} \neq \text{S(x)} \\ \forall xy \, (S(x) = S(y) \to x = y) \end{cases}$

$M_1 \models T_0$

$M_2 \not\models T_0$ because $M_2 \not\models \forall x(zero \neq S(x))$
r = -1 $\in \mathbb{Q}$, r+1 = 0

$M_3 \models T_0$

$M_4 \models T_0$

1.3 T_0 is not "complete"

$M_1 \models \forall x(x \neq S(x))$
But $T_0 \not\vdash \forall x(x \neq S(x))$, we prove such things via soundness.

We give a model of T_0 which is not a model of $T_0 \not\vdash \forall x(x \neq S(x))$. M_3 is such a model. $M_3 \models T_0$ but $M_3 \not\models \forall x(x = S(x))$. "$\frac{0}{2}$ is still zero." $M_3 \models \exists x(\text{x=S(x)})$, $0 \in [0, 1]$, $\frac{0}{2} = 0$.

In order to try to make T_0 a complete theory T we must:

$$T = \begin{cases} T_0 \\ \forall x(x \neq S(x)), \forall x(x \neq S(S(x))), ... \\ \forall x(x = zero \vee \exists y(x = S(y))) \end{cases}$$

$M_4 \not\models \forall x(x = zero \vee \exists y(x = S(y)))$
zero^{M_4}=1, $\frac{1}{2} < \frac{3}{4} \leq 1$, $\frac{3}{4}$ is in]0, 1] is not zero^{M_4}, but we cannot find r $\in$]0, 1].
$\text{S}^{M_4}(\text{r}) = \frac{3}{4}$, $\frac{r}{2} = \frac{3}{4}$.

1.4 Theorem: $\mathbf{T} \vdash \phi \longleftrightarrow \mathbf{M}_1 \models \phi$

But T has other models (not the "same" as M_1)

Let M_5: A = $\mathbb{N} \cup \{r \in \mathbb{R} | r \, is \, irrational\} = \mathbb{N} \cup \mathbb{R} \backslash \mathbb{Q}$
$\text{zero}^{M_5} = 0$, $\text{S}^{M_5} \begin{cases} n \in \mathbb{N} \mapsto n + 1 \\ x \in \mathbb{R} \backslash \mathbb{Q} \mapsto \frac{x}{2} \end{cases}$
$M_5 \models$ T, we can describe all models of T.

1.5 Another way to complete T_0

We do this by adding "induction". We add: T' = T_0+ all induction sentences. $\phi(zero) \wedge \forall x(\phi(x) \rightarrow \phi(S(x))) \rightarrow \forall x\, \phi(x)$ for all formula $\phi(x)$.

1.5.1 Example

T' $\vdash \forall x(x \neq S(x))$
Let us write $\psi(x) = x \neq S(x)$. $T_0 \vdash \psi$(zero) = zero $\neq$S(zero)
$T_0 \vdash \forall x(\psi(x) \rightarrow \psi(S(x)))$
$T_0 \vdash \forall x(x \neq S(x) \rightarrow S(x) \neq S(S(x)))$

2 Presburger Arithmetic: PrA

We have: zero, S(x), add (arity 2)
And the following axiom: T_0 + all induction sentences + $\forall x(add(x, zero) = x)$ + $\forall xy(add(x, S(y)) = S(add(x, y)))$

A model of PrA is given by a model M where:
A = $\mathbb{N}$
$\text{zero}^M = 0$
$\text{S}^M = \text{n} \mapsto \text{n+1}$
add^M(n,m) = n+m

2.1 Completeness proof

One can show that: PrA $\vdash \phi \longleftrightarrow$ M $\models \phi$
Furthermore, we can write an algorithm which decides PrA $\vdash \phi$.

2.2 Notes

Completeness of a theory (complete theory) for a <u>given model</u> M_1 (T $\vdash \phi \longleftrightarrow$ $M_1 \models \phi$). The completeness theorem states that: T $\vdash \phi \longleftrightarrow$ <u>For all</u> models M, M $\models$ T $\rightarrow$ M $\models \phi$.

If we have T $\vdash \phi \longleftrightarrow M_1 \models \phi$ then we have: $\forall \phi(T \vdash \phi \, or \, T \vdash \neg\phi)$, because M_1 $\models \phi$ or $M_1 \not\models \neg\phi$.

T_0 not complete, because: $T_0 \nvdash \forall x(x \neq S(x))$, $T_0 \nvdash \neg\forall x(x \neq x = S(x))$.

If a theory is called decidable it means that we can write an algorithm which decides: T $\vdash \phi$ or T $\nvdash \phi$.

3 Peano Arithmetic

PA = PrA +
mul(x,y), function symbol of arity 2
$\forall x(mult(x, zero)) = zero$
$\forall xy(mult(x, S(y)) = add(mult(x, y), x))$

PA $\vdash$ "$\forall xyz\,(x+1)^{\mathbf{3}} + (y+1)^{\mathbf{3}} \neq (z+1)^{\mathbf{3}}$"

Gödel showed that PA is not complete.

Given a model M:
A = $\mathbb{N}$
$\text{zero}^M = 0$
$\text{S}^M(\text{u}) = \text{u}+1$
$\text{add}^M(\text{x,y}) = \text{x}+\text{y}$
$\text{mul}^M(\text{x,y}) = \text{x}*\text{y}$

His incompleteness theorem showed that: PA $\vdash \phi \nleftrightarrow$ M $\models \phi$ and there is no way to complete the theory PA. We cannot find in an "effective" way T $\supseteq$ PA which is complete. There is no algorithm which decides whether M $\models \phi$ or not.

In particular one can find ϕ s.t. PA $\nvdash \phi$ but M $\models \phi$.

4 Decision Problem

Given a language and a sentence ϕ is this language, can we decide if $\vdash \phi$ (which is the same as $\models \phi$)? In other words, can we write an algorithm that given the input ϕ returns yes if $\vdash \phi$ and no if $\nvdash \phi$.

The answer is no. There is no such algorithm in general. It is however possible is some special cases!

4.1 A more general problem

Given ϕ_1, ..., ϕ_n can we prove ϕ_1, ..., $\phi_n \vdash \phi$?

The first idea is to use completeness/soundness. ϕ_1, ..., $\phi_n \vdash \phi$ holds iff ϕ_1, ..., ϕ_n, $\neg\phi \vdash \bot$. Which is the same as showing that ϕ_1, ..., ϕ_n, $\neg\phi$ has no model.

4.1.1 Special case

No function symbols.
We can have constants.
All sentences are on the form: $\forall x_1$,...,x_n ϕ or $\exists x_1, ..., x_n$ ϕ, where ϕ is quantifier free.

Example 1
Given: R(x,y), we ask if: $\forall xy(\underline{R(x,y) \to \neg R(y,x)}) \vdash \forall x \neg R(x,x)$ holds. *(The underlined part is the quantifier free part.)* We solve this in an automatic way using the remark under 4.1.

Has $\forall xy(R(x,y) \to \neg R(y,x))$, $\exists x R(x,x)$ a model?
We use $\neg \forall x \psi(x) \longleftrightarrow \exists x \neg \psi(x)$ and $\neg\neg\psi \longleftrightarrow \psi$.

1. Replace $\exists x$ by introducing a constant.

Has $\forall xy(R(x,y) \to R(y,x))$, $R(a,a)$ a model?
We get a theory with only universal formulae and finitely many constants.

2. Eliminate $\forall x$

Has $R(a,a) \to \neg R(a,a)$, $R(a,a)$ a model?
Theory without quantifiers purely a propositional problem.

In this case: $R(a,a) = 1$ should be true, but in the above expr we get that $R(a,a) = 0$ so there is no possible model!

Example 2
$\forall x \neg R(x,x) \overset{?}{\vdash} \forall xy(R(x,y) \to \neg R(y,x))$
$\forall x \neg R(x,x), \exists xy(R(x,y) \wedge R(y,x))$?

1. $\forall x \neg R(x,x)$, $R(a,b) \wedge R(b,a)$
2. $\neg R(a,a)$, $\neg R(b,b)$, $R(a,b), R(b,a)$

We found a counter-model!

A = {a,b}
R(a,b) = R(b,a) = 1 R(a,a)=R(b,b)=0

DAT060
LV 5, Lecture 3

Problem 1

Signature = $\sum$ ={True, False, $\vdash$, $\models$}
Domain = D = {algorithms A(x), formulas f}
An algorithm decides a relation: decides(A,R) := $\forall i(A(i) = True \longleftrightarrow R(i))$
T = {$\forall\phi(\vdash \phi \rightarrow \models \phi)$, $\forall\phi(\models \phi \rightarrow \vdash \phi)$, $\neg decidable(\models)$}

We are only interested in the relation $\vdash$ hence we change the relation R to $\vdash$: decides(A, $\vdash$) := $\forall\phi(A(\phi) = True \longleftrightarrow \vdash (\phi))$.
decidable($\vdash$) := $\exists A$ decides(A, $\vdash$) (note that we quantify over function symbols A)

We want to show that: $\vdash \neg$decidable($\vdash$)

We prove a negation by assuming the negation.

1	$\mathit{decidable}(\vdash)$	assumption ($\exists A\ \forall \phi(A(\phi)) = \text{True} \leftrightarrow\vdash \phi$)
2	$\forall \phi(A_0(\phi) = \text{True} \leftrightarrow\vdash \phi)$	assumption
3	ϕ	
4	$A_0(\phi_0) = \text{True} \leftrightarrow\vdash \phi_0$	$\forall e$ (2, ϕ_0)
5	$A_0(\phi_0) = \text{True} \rightarrow \phi_0$	$\wedge e_1$ 4
6	$A_0(\phi_0) = \text{True} \leftarrow \phi_0$	$\wedge e_2$ 5
7	$A_0(\phi_0) = \text{True}$	assumption
8	$\vdash \phi_0$	$\rightarrow$e 7, 5
9	$\vdash \phi_0 \rightarrow\models \phi_0$	soundness
10	$\models \phi_0$	$\rightarrow$e 8, 9
11	$A_0(\phi_0) = \text{True} \rightarrow\models \phi_0$	$\rightarrow$i 7–10
12	$\models \phi_0$	assumption
13	$\models \phi_0 \rightarrow\vdash \phi_0$	completeness
14	$\vdash \phi_0$	$\rightarrow$e 12, 13
15	$A_0(\phi_0) = \text{True}$	$\rightarrow$e 14, 6
16	$\models \phi_0 \rightarrow A_0(\phi_0) = \text{True}$	$\rightarrow$i 12-15
17	$A_0(\phi_0) = \text{True} \leftrightarrow\models \phi_0$	$\wedge$i 11, 16
18	$\forall \phi\ A_0(\phi) = \text{True} \leftrightarrow\models \phi$	$\forall$i (3–16, ϕ_0)
19	$\exists A\ \forall \phi(A(\phi) = \text{True} \leftrightarrow\models \phi)$	$\exists i$ (18, A_0)
20	$\mathit{decidable}(\models)$	$\exists e$ (1,2, 2–19, A_0)
21	$\neg \mathit{decidable}(\models)$	undecidability
22	$\bot$	$\neg e$ 20, 21
23	$\neg \mathit{decidable}(\vdash)$	$\neg$i 1–22

What went wrong on the submissions?

p∧q→r is parsed as (p∧q)→r and $\exists x\,\phi \to \psi$ is parsed as $\exists x\,(\phi) \to \psi$

If you in a proof write: $\neg\forall y\,P(y)$ you cannot eliminate $\forall y$ before you have gotten rid of the negation.

If you have an implication: P(x) → S, $\exists x(P(x))$→S then this is wrong. You should write $\exists x(P(x) \to S)$!

If you have:

1	$\exists x\,P(x)$	
2	$P(x_0)$	
3	.	
4	.	
5	.	
6	$Q(x_0)$	
7	$Q(x_0)$	$\exists e$ 0,x_0

This is wrong...

Last Week's Assignment

Problem 1

a) $\forall x\, f(f(x)) = f(x),\ f(b) = c \vdash c = f(c)$

1	$\forall x\ f(f(x)) = f(x)$	premise
2	$f(b) = c$	premise
3	$f(f(b)) = f(b)$	$\forall x$e 1
4	$f(c) = c$	=e 2, 3
5	$c = f(c)$	symmetry

symmetry: t=u $\vdash$ u=t

1	$t = u$	premise
2	$t = t$	=i
3	$\phi[u/x]$	=e 1, 2 ($\phi[u/x] = (u = t)$)

b) $\forall x \forall y\, (x = g(y) \to f(x) = y) \vdash \forall x\, f(g(x)) = x$

1	$\forall x \forall y\ (x = g(y) \to f(x) = y)$	premise
2	$\vert\ x_0$	
3	$\vert\ \forall y\ (g(x_0) = g(y) \to f(g(x_0)) = y)$	$\forall x$e 1 $g(x_0)$
4	$\vert\ g(x_0) = g(x_0) \to f(g(x_0)) = x_0$	$\forall y$e 3 x_0
5	$\vert\ g(x_0) = g(x_0)$	=i $g(x_0)$
6	$\vert\ f(g(x_0)) = x_0$	$\to$e 4, 5
7	$\forall x\ f(g(x)) = x$	$\forall x$i 2–6

Problem 2

$\phi = \forall x \exists y \exists z\, (P(x, y) \land P(y, z) \land (\forall w\, (P(w, x) \to P(w, z))))$
a) $\mathcal{M}_0$: A = $\mathbb{N}$ with $\mathcal{P}^{\mathcal{M}_0} := \{(m, n) | m < n \text{ and } m, n \in \mathbb{N}\}$

$\mathcal{M}_0 \models \phi \Leftrightarrow$ For all n $\in \mathbb{N}$, exists a, b $\in \mathbb{N}$, n $<$ a $\land$ a $<$ b $\land$ for all c, if c $<$ a then c $<$ b.

For an arbitrary n $\in \mathbb{N}$, let a = n+1 and b = n+2. Then n $<$ a $\Leftrightarrow$ n $<$ n+1 and a $<$ b $\Leftrightarrow$ n+1 $<$ n+2. We also have that for all c, if c $<$ a then c $<$ b $\Leftrightarrow$ for all c, c $<$ n+1 then c $<$ n+2 (by transitivity).

b) $\mathcal{M}_1$: A = $\mathbb{N}$ with $\mathcal{P}^{\mathcal{M}_1} := \{(m, 2m) | m \in \mathbb{N}\}$

$\mathcal{M}_1 \models \phi \Leftrightarrow$ For all c $\in \mathbb{N}$, exists a, b $\in \mathbb{N}$, (c,a) $\in \mathrm{P}^{\mathcal{M}_1}$, (a,b) $\in \mathrm{P}^{\mathcal{M}_1}$, for all d $\in \mathbb{N}$ if ((d,c) $\in \mathrm{P}^{\mathcal{M}_1}$) then (d,b) $\in \mathrm{P}^{\mathcal{M}_1}$

(c, a) $\in \mathrm{P}^{\mathcal{M}_1} \Leftrightarrow$ a = 2c
(a, b) $\in \mathrm{P}^{\mathcal{M}_1} \Leftrightarrow$ b = 2a
last part $\Leftrightarrow$ for all d $\in \mathbb{N}$ c = 2d and b=2d

His example was wacko. Well, the model does not satisfy the formula. You can look at my handin for an example.

Problem 3

a) $\forall x \exists y\, R(x, y) \vdash \exists y \forall x\, R(x, y)$

We can reuse the model $\mathcal{M}_0$ with an extension saying that $\mathrm{R}^{\mathcal{M}_0} = P^{\mathcal{M}_0} = \{(x, y) | x < y,\, x,\, y \in \mathbb{N}\}$ from Problem 2 to show that $\mathcal{M}_0 \models \phi$.

Given any number x in $\mathbb{N}$ you can always find a y s.t. x $<$ y. However, given a number y $\in \mathbb{N}$ it is not the case that y $<$ (all possible numbers in $\mathbb{N}$).

The sequent is invalid.

b) $\forall x\, (P(x) \vee Q(x)) \vdash \forall x\, P(x) \vee \forall x\, Q(x)$

A := {0,1}
$P^{\mathcal{M}}$:= {0}
$Q^{\mathcal{M}}$:= {1}

$\forall x\, (P(x) \vee Q(x))$ is obviously true, since either $P^{\mathcal{M}}$ or $Q^{\mathcal{M}}$ will be true for x=0 and x=1. In the second case, $\forall x\, P(x)$ is not true, since $P^{\mathcal{M}}(1)$ is not true. $\forall x\, Q(x)$ is also not true for all x, since $Q^{\mathcal{M}}(0)$ is false.

c) $\forall x \exists y\,(P(x) \to Q(y)) \vdash \forall x\,(P(x) \to \exists y\,Q(y))$

1	$\forall x \exists y\ P(x) \to Q(y)$	premise
2	$\quad x_0$	
3	$\quad \exists y\ (P(x_0) \to Q(y))$	$\forall x$e 1
4	$\quad\quad P(x_0)$	assumption
5	$\quad\quad P(x_0) \to Q(y_0)$	assumption
6	$\quad\quad\quad Q(y_0)$	$\to$e 4, 5
7	$\quad\quad\quad \exists y\ Q(y)$	$\exists y$i 6
8	$\quad\quad \exists y\ Q(y)$	$\exists y$e 3, 5–7
9	$\quad P(x_0) \to \exists y\ Q(y)$	$\to$i 4–8
10	$\forall x\ (P(x) \to \exists y\ Q(y))$	$\forall x$i 2–9

d) $\forall x\, R(x,x),\ \forall x \forall y\,(R(x,y) \to R(y,x)) \vdash \forall x \forall y \forall z\,(R(x,y) \wedge R(y,z) \to R(x,z))$

The formula claims that if a relation R is reflexive and symmetrical, it must also be transitive.

Let A :={1,2,3}
Let $R^{\mathcal{M}}$ = AxA \ {(1,3),(3,1)}

DAT060
LV 6, Lecture 1

Decision Problem

ϕ_1, ..., $\phi_n \vdash \phi \longleftrightarrow \phi_1$, ..., $\phi_n \vdash \neg\phi$ has no models.

$\phi_1 = \forall x\, \neg R(x, x)$
$\phi_2 = \forall x \forall y\, R(x, y) \rightarrow \neg R(y, x)$
$\phi = \forall x \forall y \forall z\, R(x, y) \wedge R(y, z) \rightarrow R(x, z)$

Problem

$\phi_1, \phi_2 \overset{?}{\vdash} \phi$

$\phi_1, \phi_2, \neg\phi$
$\neg\phi \longleftrightarrow \exists x \exists y \exists z\, R(x, y) \wedge R(y, z) \wedge \neg R(x, z)$

1) Introduce a, b, c (constants) $\phi_1, \phi_2, R(a, b), R(b, c), \neg R(a, c), \neg R(b, a), \neg R(c, b)$
2) ϕ_1:$\neg R(a, a)$, $\neg R(b, b)$, $\neg R(c, c)$
ϕ_2: $\neg R(b, a)$, $\neg R(c, b)$

A finite counter model:

	a	b	c
a	0	1	0
b	0	0	1
c	?	0	0

Answer: $\phi_1, \phi_2 \nvdash \phi$

For the expression ϕ_1, ..., $\phi_n, \neg\phi$ we either get a finite counter model, or we are unable to find a counter model at all. However, this does not work with function symbols...

Function Symbols

$\mathcal{F}$ =$\{a, f(x)\}$
$\mathcal{P} = \{R\}$
$\phi_1, \phi \overset{?}{\vdash} \exists x \neg R(x, f(x))$

Look for a counter model: T =$\phi_1, \phi, \forall x\, R(x, f(x))$
Switch ϕ_1 to a constant a: $\neg R(a, a)$
ϕ will not give anything.

$\forall x\, R(x, f(x))$ will give $R(a, f(a))$. This leads to: $f(a) \neq a$ in any model.
We also have $R(f(a), f(f(a))) = R(f(a), f^2(a))$ and $\neg R(f(a), f(a))$.
By ϕ we also have: $R(a, f^{\mathbf{2}}(a))$.
This far we have obtained: $f(a) \neq a$, $f(a) \neq f^{\mathbf{2}}$(a), a$\neq$f$^{\mathbf{2}}$(a) and we can go on like this...

Any model of T has to be infinite. a, $f(a)$, $f^2(a)$, $f^3(a)$, ... all have to be different. This is not a finite counter model. However, there is an infinite counter model: $\mathcal{M}$ with $\overline{\text{A}} = \mathbb{N}$, a$^{\mathcal{M}} = 0$, f$^{\mathcal{M}}(n) = n + 1$, $P^{\mathcal{M}}(p, q)$ means $p < q$.

In general, a counter model may have to be infinite. The same will hold with "complex" sentences (not purely universal sentences).

Complex Example

$\phi_1, \phi \overset{?}{\vdash} \exists x \forall y \neg R(x, y)$

T' = ϕ_1, ϕ, $\forall x \exists y\, R(x, y)$

any model of T' has to be infinite! The model is the same as in the previous example.

This indicates that the decision problem in general is difficult. The question we ask is: "Is there an algorithm, deciding ϕ_1, ..., $\phi_n \overset{?}{\vdash} \phi$ in general?" There is no such algorithm...

Proof that there is no algorithm

We are going to encode problems that we know are not decidable. Such a problem is the Halting Problem, which we will encode in first-order logic.

Computer Model
Register Machine

Given a programming unit (list of intsructions) and a finite number of registers $\{R1, R2, R3\}$ to which the PU has access. Each register contains a natural number. We also have a list L_1, ..., L_n of instructions which do operations on the registers.

Example

If we have two registers: x and y, we start with x = n and y = 0.

Our program looks like:

L_1: y ←x
L_2: if x == zero goto L_5 else x = x-1
L_3: y = y+1
L_4: goto L_2
L_5: STOP

Test run: x = 2, y = 0

	x	y
L_1	2	0
L_2	2	2
L_3	1	2
L_4	1	3
L_2	1	3
L_3	0	3
L_4	0	4
L_2	0	4
L_5	X	X

In general, if we start with x = n, y = 0 the machine will stop with x = 0 and y = 2n. In this way, we can represent any program with a finite number of registers and instructions. The halting problem is in general undecidable.

Example

A program that does not terminate is:

L_1: y ← x
L_2: goto L_1
L_3: STOP

It will loop inf.

The Halting Problem

Given a program and a starting state (x = 0, y = 0) will this program stop?

Halting Problem in Predicate Logic

We represent each line by a predicate symbol. Since a program is finite, we will use a finite number of predicate symbols. The arity of these symbols is the number of registers. (All the predicate symbols have the same arity.) We also have 0 and Succ(x) = S(x). The program will be represented by a finite number of sentences: ϕ_1, ..., ϕ_n.

The question whether the program halts will be represented by $\mathrm{T} \vdash \exists x \exists y\, P_5(x, y)$ (where P_5 is the predicate symbol for L_5 in the example.

This will show that ϕ_1, ..., $\phi_n \overset{?}{\vdash} \phi$ is "in general not solvable".

Program

Let the theory T be:

L_1: $\forall x \forall y\, P_1(x, y) \rightarrow P_2(x, x)$
L_2: $\forall y\, P_2(0, y) \rightarrow P_5(0, y)) \wedge \forall x \forall y\, P_2(S(x), y) \rightarrow P_3(x, y))$
L_3 : $\forall x \forall y (P_3(x, y) \rightarrow P_4(x, S(y)))$
L_4: $\forall x \forall y (P_4(x, y) \rightarrow P_2(x, y))$

If we start with the state: T, $P_1(n, 0) \vdash P_5(0, 2n)$ in general T, $P_1(n, 0) \vdash \exists x \exists y\, P_5(x, y)$.

Program 2

Let the theory T' be:

L_1: $\forall x \forall y\, P_1(x, y) \rightarrow P_2(x, x)$
L_2: $\forall x \forall y (P_2(x, y) \rightarrow P_1(x, y))$

If we start with the state: T', $P_1(0,0) \nvdash \exists x \exists y\, P_3(x, y)$, we can only deduce $P_2(0, 0)$.

Decision Problem

In general we have k registers (r_1, ..., r_k) and in general we can represent the instructions below in first-order logic.

- $\mathrm{r}_i \leftarrow \mathrm{r}_j$
- $\mathrm{r}_i = \mathrm{r}_i + 1$
- goto L_p
- if $\mathrm{r}_i == 0$ then goto L_p else $\mathrm{r}_i = \mathrm{r}_i$ - 1

Yet another encoding of an undecidable problem

The problem is to decide if we can solve an equation $P(x_1, ..., x_n) = Q(x_1, ..., x_n)$ where $x_1, ..., x_n$ are natural numbers, and P, Q are polynomials with natural number coefficients. There is no such algorithm.

The theory T: 0, S(x), add, mul.

$\forall x$ add(x,0) = x
$\forall x \forall y$ add(x,S(y)) = S(add(x,y))
$\forall x$ mul(x,0) = 0
$\forall x \forall y$ mul(x,S(y)) = add(mul(x,y)x)

The question is: Can we solve $x^3 = y^2 + 2$? Or: $T \overset{?}{\vdash} \exists x \exists y$ mul(mul(x, x), x) = S(S(mul(y, y)))

This will hold iff we cand find natural numbers p and q s.t. $p^3 = q^2 + 2$.

Restrictions of Predicate Logic

Graphs

Where R(x,y) means that x is connected to y. A model $\mathcal{M}$ is a graph. The question is: Can we express reachability, in predicate logic? I.e. are two given elements connected by a path?

Language: R(x,y), a, b
Theorem: There is no sentence ϕ s.t. $\mathcal{M} \models \phi \longleftrightarrow a^{\mathcal{M}}\, b^{\mathcal{M}}$ are connected by a path.

Proof: The completeness theorem and sondness theorem, generalized.
Recall that a theory T is a set of sentences, which can be infinite. $T \vdash \phi$ means that we can find $\phi_1, ..., \phi_n \in T$ s.t. $\phi_1, ..., \phi_n \vdash \phi$. $T \models \phi$ means that $\forall \mathcal{M}$ $\mathcal{M} \models T \rightarrow \mathcal{M} \models \phi$ means that $\mathcal{M} \models \psi$ for all $\psi \in T$.

$T \vdash \phi \longleftrightarrow T \models \phi$
$T \nvdash \phi \longleftrightarrow T \not\models \phi$
$T \nvdash \bot \longleftrightarrow T \not\models \bot$
T is consistent $\longleftrightarrow$ T has a model

Compactness Theorem

T has a model $\longleftrightarrow$ for all finite subsets $\phi_1, ..., \phi_n$ of T, $\phi_1, ..., \phi_n$ have a model.

Proof:
T is consistent $\longleftrightarrow$ All finite subsets of T is consistent

Using first-order logic we can express that $\mathrm{a}^{\mathcal{M}}$ and $\mathrm{b}^{\mathcal{M}}$ are connected by a path of length ≤ 1.

δ_1: $a = b \vee R(a, b)$
δ_2: $\delta_1 \vee \exists x\, R(a, x) \wedge R(x, b)$
δ_3: $\delta_2 \vee \exists x_1 \exists x_2\, R(a, x_1) \wedge R(x_1, x_2) \wedge R(x_2, b)$
.
.
.
$\mathcal{M} \models \delta_k \longleftrightarrow \mathrm{a}^{\mathcal{M}}$, $\mathrm{b}^{\mathcal{M}}$ connected by a path of length $\leq$ k

Proof of the theorem:
Assume we have a such a sentence ϕ and consider the theory. This will be infinite.

ϕ, $\neg\delta_1$, $\neg\delta_2$, $\neg\delta_3$, ...

The theory is infinite.

By the compactness theorem it should have a model. If we for instance take ϕ, $\neg\delta_3$, $\neg\delta_{17}$ it should have a model (any finite subset should have one). Hence, the theory should have a model: $\mathcal{M}$. $\mathrm{a}^{\mathcal{M}}$, $\mathrm{b}^{\mathcal{M}}$ should be connected, but this is a contradiction since we have stated that we have no path of length 1, 2, 3, ...

There is therefore no formula ϕ such that $\mathcal{M} \models \phi$ and we cannot prove reachability in predicate logic.

DAT060
LV 6, Lecture 2

Problem

$\mathit{\Gamma} = \{\forall x.S(x,x),\ \forall x \forall y.(S(x,y) \rightarrow x = y)\}$

If $\mathcal{M} \models \Gamma$ and so $\mathrm{S}^{\mathcal{M}} \subseteq A^{\mathbf{2}}$. What is $\mathrm{S}^{\mathcal{M}}$?

$\mathcal{M} \models \forall x.S(x,x) \Leftrightarrow$ for all a $\in$ A, (a,a) $\in \mathrm{S}^{\mathcal{M}} \Leftrightarrow \{$(a,a) | a $\in$ A$\} \subseteq \mathrm{S}^{\mathcal{M}}$
$\mathcal{M} \models \forall x \forall y.(S(x,y) \rightarrow x = y) \Leftrightarrow$ for all a, b $\in$ A, if (a,b) $\in \mathrm{S}^{\mathcal{M}}$ then a = b $\Leftrightarrow$ $\mathrm{S}^{\mathcal{M}} \subseteq \{(a,b) | a = b,\ a,b \in A\} = \{(a,a) | a \in A)\}$

This leads to: $\{(a,a) | a \in A\} \subseteq S^{\mathcal{M}} \subseteq \{(a,a) | a \in A\}$

Problem 2.4.11 d)

Is $\{\phi = \exists x.S(x,x),\ \psi = \forall x \forall y.(S(x,y) \rightarrow x = y)\}$ consistent?

The set is consistent iff there is a model $\mathcal{M}$ s.t. $\mathcal{M} \models \phi$ and $\mathcal{M} \models \psi$.

Let $\mathcal{M}$ have A = $\mathbb{N}$, $\mathrm{S}^{\mathcal{M}} = \{(a,a) | a \in \mathbb{N}\}$, $\mathcal{M} \models \psi$ (which we saw above).
$\mathcal{M} \models \phi \Leftrightarrow$ exists a $\in \mathbb{N}$, (a,a) $\in \mathrm{S}^{\mathcal{M}}$. In order to prove this we can let a = 0. Then (0,0) $\in \mathrm{S}^{\mathcal{M}}$.

Problem

Let $\Gamma = \{\phi_1 = \forall x.\neg S(x,x),\ \phi_2 = \forall x \forall y \forall z.(S(x,y) \wedge S(y,z) \rightarrow S(x,z)),\ \phi_3 = \forall x \exists y.S(x,y)\}$

a) Show that Γ is satisfiable (holds for at least one model).
We take $\mathcal{M}$ given by A = $\mathbb{N}$ and $\mathrm{S}^{\mathcal{M}} = \{(m,n) | m < n,\ m,n \in \mathbb{N}\}$

b) Show that any model of Γ is infinite. I.e. if $\mathcal{M} \models \Gamma$, with carrier A, then |A| is infinite.

By def. $A \neq \emptyset$, so $a_0 \in A$.
Because of ϕ_3 there is $a_1 \in A$ such that $(a_0, a_1) \in S^{\mathcal{M}}$.
$(a_0, a_1) \in S^{\mathcal{M}}$, $(a_1, a_2) \in S^{\mathcal{M}}$, $(a_2, a_3) \in S^{\mathcal{M}}$, ...
(a_i) for each $i \in \mathbb{N}$ s.t. $(a_i, a_{i+1}) \in S^{\mathcal{M}}$

We want to show that $a_i \neq a_j$ for $i < j$. We use contradiction.

- Assume $a_i = a_j$
- $(a_i, a_{i+1}) \in S^{\mathcal{M}}$, $(a_{i+1}, a_{i+2}) \in S^{\mathcal{M}}$, ..., $(a_{j-1}, a_j) \in S^{\mathcal{M}}$

From these two points we can use that $\mathcal{M} \models \phi_2$ to conclude that $(a_i, a_j) \in S^{\mathcal{M}}$.
Using the assumption we see that $(a_i, a_i) \in S^{\mathcal{M}}$ but this contradicts $\mathcal{M} \models \phi$.

In the end, $|A|$ is infinite.

c) For each $n \geq 2$, define a sentence ϕ_n s.t. $\mathcal{M} \models \phi_n \Leftrightarrow \mathcal{M}$ has at least n elements.
$\phi_n \equiv \exists x_1 \exists x_2$, ..., $\exists x_n$, $\neg(x_1 = x_2) \wedge \neg(x_2 = x_3) \wedge$ ... which we can write as:

$$\phi_n \equiv \exists x_1 \exists x_2, ..., \exists x_n, \bigwedge_{1 \leq i \leq j \leq n} \neg(x_i = x_j)$$

d) Show for any $n \geq 2$, $\Gamma \vdash \phi_n$ is valid.
By completeness we know that if: for all $\mathcal{M}$, $\mathcal{M} \vdash \Gamma$ then $\mathcal{M} \models \phi$. This in turn implies that $\Gamma \vdash \phi_n$.

Show that for all $\mathcal{M}$, $\mathcal{M} \models \Gamma$ then $\mathcal{M} \models \phi_n$. However, $\mathcal{M} \models \Gamma \Rightarrow \mathcal{M}$ is infinite $\Rightarrow \mathcal{M}$ has at least n elements $\Rightarrow \mathcal{M} \models \phi_n$.

Compactness Theorem
For each set of formulas Γ, if all finite subsets of Γ are satisfiable, then Γ is satisfiable.
$\Gamma = \{\forall x \forall y.(x = y \rightarrow \neg(x = y)), \forall x.x = x\}$
$\Delta = \{\forall x.x = x\}$

There is no set of formulas T, s.t. for every $\mathcal{M}$, $\mathcal{M} \models T \Leftrightarrow \mathcal{M}$ is finite.

Let $Q = T \cup \{\phi_n \mid n \in \mathbb{N}\}$

If $\mathcal{M} \models Q \Rightarrow (\mathcal{M} \models T$ and $\mathcal{M} \models \{\phi_n \mid n \in \mathbb{N}\}) \Leftrightarrow \mathcal{M}$ is finite and $\mathcal{M}$ is infinite. This is clearlya contradiction! There is no $\mathcal{M}$ s.t. $\mathcal{M} \models Q$.

If we want to show that there exist an $\mathcal{M}$ and that $\mathcal{M} \models Q$ we can use the compactness theorem. We can use it to show that it is enough to show that (for all $\Delta \subseteq Q$, $\Delta_{\text{is finite}} \Rightarrow \Delta_{\text{satisfiable}}$). For an arbitrary $\Delta \subseteq Q, \Delta_{\text{finite}}$ show that

there exists $\mathcal{M}$, $\mathcal{M} \models \Delta$.

$\Delta \subseteq$ Q =T $\cup \{\phi_n \mid$ n $\in \mathbb{N}\}$
$\Delta = \Delta_1 \cup \Delta_2$, $\Delta_1 \subseteq$ T, $\Delta_2 \subseteq \{\phi_n | n \in \mathbb{N}\}$
$\Delta_2 = \{\phi_i | i \in B, B \subseteq \mathbb{N}\}$ (B is finite)
$\mathcal{M} \models \Delta_2 \Leftrightarrow \mathcal{M}$ has at least max(B) elements

We pick $\mathcal{M}$ with domain A = {0, 1, ..., max(B)}. We have shown that $\mathcal{M} \models \Delta_2$, we need to show that $\mathcal{M} \models \Delta_1$.

$\Delta_1 \subseteq$ T, hence we show instead that $\mathcal{M} \models$ T.

$\mathcal{M} \models$ T $\Leftrightarrow \mathcal{M}$ is finite

$\mathcal{M}$ is finite, since |A| is max(B) which is finite!

$\mathcal{M} \models \Delta_1$, $\mathcal{M} \models \Delta_2 \Rightarrow \mathcal{M} \models \Delta \Rightarrow \Delta$ is satisfiable.

We did hower say that there is no $\mathcal{M}$, $\mathcal{M} \models Q$ so there is some sort of contradiction somewhere. I didn't really understand.

DAT060
LV 6, Lecture 3

Compactness Theorem

T has a model if all finite subset of T has a model.

A relation is a well-founded relation: a $\rightarrow$ b (a,b) $\in S^{\mathcal{M}}$, S $\subseteq A^2$, if it has no infinite paths. An infinite path would be $a_0 \rightarrow a_1 \rightarrow ...$, $a_n \in S$, $\forall n.(a_n, a_{n+1}) \in S$.

Remark: if we have a$\rightarrow$a we have a$\rightarrow$a$\rightarrow$a$\rightarrow$a$\rightarrow$...

Theorem

There is no sentence ϕ such that $\mathcal{M} \models \phi \longleftrightarrow R^{\mathcal{M}}$ is well founded.

Proof: by contradiction
Assume ϕ.
Add to the language, constants: a_0, a_1, a_2, ...
Consider T = ϕ, $R(a_0, a_1)$, $R(a_1, a_2)$, ...

We claim that T has a model (but this will be a contradiction).

By the compactness theorem we know that all finite subsets of T has a finite model. A model of T will look like ${a_0}^{\mathcal{M}} \rightarrow {a_1}^{\mathcal{M}} \rightarrow {a_2}^{\mathcal{M}} \rightarrow ...$, so it is not well fonded. This is a contradiction with $\phi \in$ T, $\mathcal{M} \models \phi$.

Example

$\phi = \forall x \forall y. \neg R(x, y)$
$\mathcal{M} \models \phi \rightarrow \mathcal{R}^{\mathcal{M}}$ is well-founded : a $\rightarrow$ b

$\phi = \forall x \forall y \forall z. \neg(R(x, y) \wedge R(y, z))$

We can state that there is no path of length $>$ k for a fixed k.
We cannot state that there is no infine path.

Temporal Logic

LTL and CTL are extensions of propositional logic. There are no quantifiers(!) instead we have modal operators.

3 Traditions in Logic

- Model Theory: $\models$
- Proof Theory: $\vdash$
- Algebraic Logic: $\phi \leftrightarrow \psi$, $\phi \rightarrow \psi$

A model in algebraic logic

α : PVar $\rightarrow$ {0,1}
$\alpha \models \phi$ by induction on ϕ
$\alpha \models$ P α(P) = 1
$\alpha \models \neg\phi$ not $\alpha \models \phi$
$\alpha \models \phi \wedge \psi$ $\alpha \models \phi$ and $\alpha \models \psi$
$\alpha \models \phi \vee \psi$ $\alpha \models \phi$ or $\alpha \models \psi$

Note: PVar is a propositional variable.

Proof Theory

ϕ_1, ..., $\phi_n \vdash \phi$, what is the derevation of ϕ from ϕ_1, ..., ϕ_n?

$\vdash \phi \leftrightarrow \models \phi$

Linear Temporal Logic - LTL

The idea: The truth value of a propositional variable is not simply 0 or 1 but an infinite sequence of 0s and 1s. A model α: PVar $\rightarrow \{0,1\}^{\mathbb{N}}$.

Motivation: Circuit analysis!

Givan a Flip-Flop (latch) you need to consider the time (clock). In this case we need to be able to represent:

$$\begin{cases} r(0) = 0 \\ r(t+1) = \neg(p(t) \wedge s(t)) \\ s(t+1) = \neg(q(t) \wedge r(t)) \end{cases}$$

we introduce the modal operations:

- X_p (X_p)(t) "=" p(t+1) next

- G_p $(G_p)(t)$ "=" $\forall x \geq t$ p(x) globally
- F_p $(F_p)(t)$ "=" $\exists x \geq t$ p(x) finally

We will have:
$G_p \longleftrightarrow p \wedge XG_p$
$F_p \longleftrightarrow p \vee XF_p$

Syntax Definition

$\phi \equiv p|\phi \wedge \phi|\phi \vee \phi|\phi \rightarrow \psi|\neg\phi|X_\phi|F_\phi|G_\phi$

Model Definition

$\alpha \models \phi$
α : PVar $\rightarrow \{0,1\}^{\mathbb{N}}$

$\models \phi$
$\forall\alpha(\alpha \models \phi)$

If we have the variables: p, q, r, a model in propositional logic is a sequence of three boolean values: p = 0, q = 1, r = 0 from which ϕ can be calculated.

In LTL a model for three variables, p, q, r, is given by time as an additional element. How do we then determine if $\alpha \models \phi$ holds?

time	0	1	2	3	4
p	0	0	0	0	0
q	1	0	1	0	1
r	1	1	1	0	0

$\alpha \not\models$ p, $\alpha \models$ q, $\alpha \models$ r

$\alpha \models$ p $\equiv \alpha$p(0) = 1
$\alpha \models \neg\phi \equiv$ not $\alpha \models \phi$
$\alpha \models \phi \wedge \psi \equiv \alpha \models \phi$ and $\alpha \models \psi$
$\alpha \models \phi \vee \psi \equiv \alpha \models \phi$ or $\alpha \models \psi$
$\alpha \models X_\phi \equiv \alpha' \models \phi$
$\alpha \models G_\phi \equiv$ for all k ≥ 0 $\alpha^{(k)} \models \phi$
$\alpha \models F_\phi \equiv$ exists k ≥ 0 $\alpha^{(k)} \models \phi$

We define $\alpha \models \phi$ for a given α: PVar $\rightarrow \{0,1\}^{\mathbb{N}}$.

Given α we can define α': PVar $\rightarrow \{0,1\}^{\mathbb{N}}$, p $\in$ PVar, α' p(n) = α p(n+1)

α'

	0	1	2	3
p	0	0	0	0
q	0	1	0	1
r	1	1	0	0

α': p $\longmapsto$ n $\mapsto$ α p(n+1)

α": p $\longmapsto$ n $\mapsto$ α p(n+2)
$\alpha^{(3)}$: p $\longmapsto$ n $\mapsto$ α p(n+3)
α^k: p $\longmapsto$ n $\mapsto$ α p(n+k)

Example

α

	0	1	2	.
p	0	1	1	.

$\alpha \not\models$ p, $\alpha' \models p$, $\alpha'' \models$ p, $\alpha^{(3)} \models$ p, ...
$\alpha \not\models \mathrm{G}_p$, $\alpha' \models \mathrm{G}_p$, $\alpha \models \mathrm{XG}_p$
$\alpha \models \mathrm{F}_p$ since $\alpha' \models$ p

Theorem

If we have ψ s.t. $\models \psi \to \phi \wedge X\psi$ then $\models \psi \to G\phi$

Proof:
Assume $\models \psi \to \phi \wedge X\psi$.
$\alpha \models \psi$, show $\alpha \models G\phi$

$\alpha \models \psi$ so $\alpha \models \phi \wedge X$, so $\alpha \models \phi$ and $\alpha \models X\psi$

$\forall\beta(\beta \models \psi \to \beta \models \phi \wedge X\psi)$ so $\alpha' \models \psi \to \alpha' \models \phi \wedge X\psi$ so $\alpha' \models \phi$ and $\alpha' \models X\psi$ and $\alpha^{(2)} \models \psi$

Remark

Gϕ is the greatest solution of the equation in ψ: $\psi = \phi \wedge X\psi$. It means that:

1. it is a solution $\mathrm{G}\phi = \phi \wedge X\psi$
2. if ψ is a another solution $\psi \leq G\phi$ and $\models \psi \to G\phi$

Fϕ is the least solution of the equation in ψ: $\psi = \phi \vee X\psi$

We can define the relation $\phi \leq \psi$ on LTL formula. $\phi \leq \psi$ means $\models \phi \to \psi$. This relation satisfies $\phi \leq \phi$, $\frac{\phi \leq \psi \quad \psi \leq \delta}{\phi \leq \delta}$ (if $\phi \leq \psi$ and $\psi \leq \delta$ then $\phi \leq \delta$).

Induction Principle

For any formula ϕ we have:

$\models (G(\phi \to Xp) \wedge \phi) \to G\phi$

This is because: $G(\phi \to X\phi)$ means "always if ϕ holds then ϕ will hold next" and ϕ if ϕ holds at the start. Hence, ϕ must hold - always.

Formally

Given $\alpha \models G(\phi \to X\phi)$ and $\alpha \models \phi$ then $\alpha \models G\phi$ for all k: $\alpha^{(k)} \models \phi \to X\phi$ and $\alpha^{(k)} \models \phi \to \alpha^{(k+1)} \models \phi$ so we can show $\alpha^{(k)} \models \phi$ by induction on k.

How α is given in practice

We want to study a system which will be represented by a "transition system" (S, $\to$, L) where:

S is a finite set of "states"

$\to$ is a binary relation on S that satisfies: $\forall a \exists b. a \to b$

L is a function, L: S $\to$ PVar $\to$ {0,1}

A path in (S, $\to$, L) is a sequence $\pi = s_0 \to s_1 \to ....$ Any path will define a model α.

DAT060
LV 7, Lecture 1

Linear Temporal Logic

F$\phi \longleftrightarrow \phi \vee$ XFϕ
Fϕ the least solution of $\psi \longleftrightarrow \phi \vee$ Xψ.
Fϕ is a solution
if $\psi \longleftrightarrow \phi \vee$ Xψ we have F$\phi \rightarrow \psi$

The equation $\psi \longleftrightarrow \phi \vee$ Xψ has a greatest solution.
If we take ψ = True then X True $\longleftrightarrow$ True.

The relation $\models \phi_1 \rightarrow \phi_2$ is reflexive and transitive.
True $\models \phi \rightarrow$ True

If we have $\psi = \phi \vee$ X$\psi = \phi \vee$ X($\phi \vee$ Xψ)=$\phi \vee$ X$\phi \vee$ X^2 $\psi = \phi \vee$ X$\phi \vee$ X$^2\phi \vee$ X$^3\psi$ = ...

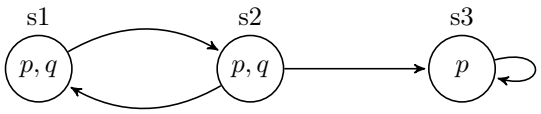
$\psi \longleftrightarrow \phi \wedge$ Xψ has a least solution: ψ = False and a greatest solution: ψ = Gϕ.

Intuitively Fϕ= $\phi \vee$ X $\phi \vee$ X$^2\phi \vee$... and Gϕ= $\phi \wedge$ X$\phi \wedge$ X$^2\phi$ ... but in our language we cannot write infinite formulas.

What does $\models \phi$ mean?

A model $\mathcal{M} = (S, \rightarrow, L)$ where S, $\rightarrow$ is a transition system (graph) where $\forall s \exists t. s \rightarrow t$. L: S $\rightarrow$ PVar $\rightarrow$ {0,1}.

Example



	p	q
s1	1	1
s2	1	1
s3	1	0

Here, $\mathcal{M} \models Gp$ but $\mathcal{M} \not\models Gq$. $\mathcal{M} \not\models Gq$ because $s_3 \to s_3 \to ...$ is a path in $\mathcal{M}$. $\mathcal{M} \not\models Fq$ because $s_1 \to s_2 \to s_1 \to ...$

Definition

A path $\sigma = s_0 \to s_1 \to s_2 \to ...$ is a function $\sigma : \mathbb{N} \to S$ s.t. $\forall n.\sigma n \to \sigma(n+1)$. We define $\sigma \models \phi$ by induction on ϕ. $\sigma \models p$ means $L(\sigma 0)p = 1$. $\sigma = s_1 \to s_2 \to s_1 \to s_2$. $\sigma \models Xp$ means $\sigma' \models \phi$ where $\sigma' = \sigma 1 \to \sigma 2 \to \sigma 2 \to ...$ $\sigma' n = \sigma(n+1)$ or more generally $\sigma^{(k)} n = \sigma(n+k)$.

We define $\mathcal{M} \models \phi$ for all paths σ of $\mathcal{M}$ we have $\sigma \models \phi$. $\models \phi$ means: for all models $\mathcal{M}$ we have $\mathcal{M} \models \phi$.

New Operator

$\psi, \phi ::= ...|F\phi|G\phi|\phi U \psi$
$\sigma \models \phi \vee \psi$ means $\exists k.\sigma^{(k)} \models \psi \wedge \forall l < k\, \sigma^{(l)} \models \phi$
$\phi U \psi \longleftrightarrow \psi \vee (\phi \wedge X(\phi U \psi))$

The least solution of the equation $\delta \leftrightarrow \psi \vee (\phi \wedge X\delta)$

If we have $\sigma \models \phi$ U ψ we can have $\sigma \models \phi$ and $\sigma \models \psi$.

Path

A path $s_1 \to s_2 \to s_3 \to s_3 \to ...$ and another path $s_1 \to s_2 \to s_1 \to s_2 \to ...$

Propositional Logic Likeliness

For propositional logic a model is just L: Pvar $\to$ {0,1}.

Model Checking

Given as input: $\mathcal{M} = (S, \to, L)$ and ϕ we get as output either a yes ($\forall \sigma.\sigma \models \phi$) or a no (a counter example that shows $\sigma \not\models \phi$).

Traffic Light Example

G(red $\rightarrow$ red U (yellow $\wedge$ X(yellow U green))
G(red $\rightarrow$ X(red U (yellow $\wedge$ X(yellow U green))))

In order to avoid multiyle lights at the same time we need to add G(red $\longleftrightarrow$ ($\neg$yellow $\wedge$ $\neg$green)).

Examples

Prove that: $\models G(\phi \rightarrow \psi) \rightarrow (G\phi \rightarrow G\psi)$

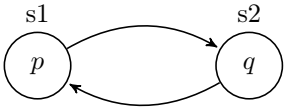
We have to prove that for all model $\mathcal{M}$ and for all path σ in $\mathcal{M}$ we have $\sigma \models G(\phi \rightarrow \psi) \rightarrow (G\phi \rightarrow G\psi)$. This means that if $\sigma \models G(\phi \rightarrow \psi)$ and $\sigma \models G\phi$ we have $\sigma \models G\psi$.

This is similar to: $\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x) \vdash \forall x.Q(x)$

Corresponding to $\exists x.(P(x) \vee Q(x)) \longleftrightarrow \exists x.P(x) \vee \exists x.Q(x)$ we have F($\phi \vee \psi$) $\leftrightarrow$ (F$\phi \vee$ Fψ).

$\exists x.(P(x) \wedge Q(x)) \rightarrow \exists x.P(x) \wedge \exists x.Q(x)$ but not $\leftarrow$, similarly we have F($\phi \wedge \psi$) $\rightarrow$ (F$\phi \wedge$ Fψ). In order to show F$\phi \wedge$ F$\psi \nrightarrow$ F($\phi \wedge \psi$) we must show that $\models F\phi \wedge F\psi \rightarrow F(\phi \wedge \psi)$. We have to find $\mathcal{M}, \sigma, \phi, \psi$ s.t. $\sigma \not\models F\phi \wedge F\psi \rightarrow F(\phi \wedge \psi)$ which means $\sigma \models F\phi$, $\sigma \models F\psi$, $\sigma \not\models F(\phi \wedge \psi)$.

For instance:



	p	q
p	1	0
q	0	1

$\sigma = s_1 \rightarrow s_2 \rightarrow s_2 \rightarrow ...$ which shows $\sigma \models G(\neg(p \wedge q))$.

GFϕ means "ϕ holds infinitely often".
$\sigma \models GF\phi$ means "$\forall k.\exists l \geq k.\sigma^{(l)} \models \phi$
$\sigma \models G(F\phi)$ means "$\forall k.\sigma^{(k)} \models F\phi$, $\forall k.\exists k'.\sigma x^{(k+k')} \models \phi$.

FGψ means "eventually ψ will always hold".
We have $\models (FG\phi) \rightarrow (GF\phi)$, but $\not\models (GF\phi) \rightarrow (FG\phi)$

In our example we have $\sigma = s_1 \rightarrow s_2 \rightarrow s_2 \rightarrow ...$ where p holds every other state (infinitiely often). However, it does not ever hold globally.

Special Case for deciding $\mathcal{M} \models \phi$

If $\phi = FGp_1 \vee FGp_2 \vee ... \vee FGp_n$ we have $\neg\phi \longleftrightarrow GF(\neg p_1) \wedge ... \wedge GF(\neg p_n)$. Instead of proving $\forall\sigma.\sigma \models \phi$ we show $\exists\sigma.\sigma \models \neg\phi$.

The problem in general is that there are infinitely many possible paths.

Idea of Algorithm

1. Look at the strongly connected components of this graph (S,$\rightarrow$). SCC is an equivalence class for the relation $\sim$.
2. Look at all non trivial SCCs. An SCC is trivial if it has "one point no path".

Main Remark

Any infinite path for $\mathcal{M}$ will eventually stay in one non trivial SCC. Hence, we can find $\sigma \models GF(\neg p_1) \wedge ... \wedge GF(\neg p_n)$ iff there is an SCC where $\neg p_1, ...\neg p_n$ holds for some state of this SCC. Conversely, if there is an infinte path $\sigma \models GF(\neg p_1) \wedge ... \wedge GF(\neg p_n)$ eventually this path will stay in an SCC where we have $\neg p_1, \neg p_2, ..., \neg p_n$.

Hamiltonian Cycle Example

In genera, to dedice $\mathcal{M} \models \phi$ has to be "complicated". Coding of the Hamiltonian Path Problem.

Given a graph G=(V,E), can we find $v_{\sigma 1} \rightarrow v_{\sigma 2} \rightarrow v_{\sigma 3} \rightarrow ... \rightarrow v_{\sigma n}$ where $\sigma 1, ..., \sigma n$ is a permutation of 1,...,n. Can we find a path in G which visits each vertex exactly once?

Solve this with LTL

S=V $\cup$ {b} and s $\rightarrow$ t means s,t $\in$ V or t=b.

Atomic formulae: p_v for each v $\in$ V, with L(b) $p_v = 0$ and $\begin{cases} L(v)p_{v'} = 1 & v = v' \\ L(v)p_{v'} = 0 & v \neq v' \end{cases}$.

$\psi = \bigvee_{v \in V} G(\neg p) \vee GFp_v \wedge XF(p_v))$

$\sigma \models \psi$ means either σ does not visit v or σ visits v twice. This means that $\mathcal{M} \models \psi \leftrightarrow$ no hamiltonian cycle. Any algorithm for deciding $\mathcal{M} \stackrel{?}{\models} \phi$ has to be atleast $\mathcal{NPC}$.

DAT060
LV 7, Lecture 2

Computational Tree Logic

Syntax

state: $\psi, \phi := p|\neg\phi|\psi \vee \phi|\psi \wedge \phi|A\alpha|E\alpha$
path: $\alpha, \beta := X\phi|F\phi|G\phi|\phi \cup \psi$

$A\alpha$ means "for all path" α holds on this path.
$E\alpha$ means "there exists a path" s.t. α holds on this path.

Model

Same as the LTL model.

A model $\mathcal{M}$=(S, $\rightarrow$, L)
S is a finite set, an element of S is called a state.
$\rightarrow$ is a binary relation on S s.t. $\forall s \exists t. s \rightarrow t$. (S, $\rightarrow$) is a transition system.
L:S $\rightarrow$ PVar $\rightarrow$ {0,1} or L: $S\,x\,Pvar \rightarrow \{0\}$

Semantics

$\mathcal{M} \models \phi$ where $\mathcal{M}$ is a model and ϕ is a CTL formula.

$\mathcal{M}, s \models \phi$, $s \in S$ state of $\mathcal{M}$
$\sigma \models \alpha$, σ is a path of $\mathcal{M}$ and α is a path formula

$s \models p \leftrightarrow L(S, p) = 1$
$s \models \neg\phi \leftrightarrow not\, s \models \phi$
$s \models \phi \vee \psi \leftrightarrow s \models \phi\, or\, s \models \psi$
$s \models \phi \wedge \psi \leftrightarrow s \models \phi\, and\, s \models \psi$
$s \models A\alpha \leftrightarrow$for all paths σ starting from s, s.t. $\sigma \models \alpha$
$s \models E\alpha \leftrightarrow$ there exists a path σ starting from s, s.t. $\sigma \models \alpha$
$\sigma \models \alpha$ where $\sigma = \sigma_0, \sigma_1, \sigma_2, ...$ is a path in $\mathcal{M}$ and $\sigma_0, \sigma_1, \sigma_2, ...$ are elements of S.

$\sigma \models X\phi \leftrightarrow \sigma_1 \models \phi$

$\sigma \models G\phi \leftrightarrow \forall k.\sigma_k \models \phi$
$\sigma \models F\phi \leftrightarrow \exists k.\sigma_k \models \phi$
$\sigma \models \phi \cup \psi \leftrightarrow \exists k.\sigma_k \models \psi \wedge \forall l < k.\sigma_l \models \phi$ (we can have k = 0, in which case $\sigma_0 \models \psi$)

$\models \phi$ means $\forall \mathcal{M}.\mathcal{M} \models \phi$
$\models \phi \leftrightarrow \psi$ means $\forall \mathcal{M}.\mathcal{M} \models \phi \leftrightarrow \psi$, $\mathcal{M} \models \phi \leftrightarrow \mathcal{M} \models \psi$
$\models \phi \rightarrow \psi$ means $\forall \mathcal{M}.\mathcal{M} \models \phi \rightarrow \mathcal{M} \models \psi$

$AG\phi \leftrightarrow \phi \wedge AX(AG\phi)$
$AF\phi \leftrightarrow \phi \vee AX(AF\phi)$
$EG\phi \leftrightarrow \phi \wedge EX(EG\phi)$
$EF\phi \leftrightarrow \phi \vee EX(EF\phi)$
$A(\phi U\psi) \leftrightarrow \psi \vee (\phi \wedge AX(\phi U\psi))$
$E(\phi U\psi) \leftrightarrow \psi \vee (\phi \wedge EX(\phi U\psi))$

So $AF\phi$ is a solution of the equation $\psi \leftrightarrow \phi \vee AX\psi$. This is the least solution of this equation.

Assume that we have $\mathcal{M}$ s.t. $\mathcal{M} \models \psi \leftrightarrow \phi \vee AX\psi$. We show $\mathcal{M} \models AF\phi \rightarrow \psi$. However, if we have $\mathcal{M} \models \neg\psi \rightarrow \neg AF\phi$, which we will prove instead.

Assume we take a state s of $\mathcal{M}$ and we assume that s$\models \neg\psi$. We then show from this: $s \models \neg AF\phi$. In order to do this we build a path $\sigma = s \rightarrow s1 \rightarrow s2 \rightarrow ...$ such that $s \not\models \phi$, $s1 \not\models \phi$, ...

From our assumption we get that: $s \models \psi \leftrightarrow s \models \phi \vee AX\psi$. We know that $s \models \neg\psi \leftrightarrow s \not\models \psi$ so we have $s \not\models \phi \vee AX\psi$ so $s \not\models \phi$ and $s \not\models AX\psi$. Hence, $s1 \not\models \psi$ which leads to $s1 \not\models \phi$. We have $s2 \not\models \psi$ which ultimately leads to $s2 \rightarrow s3 \rightarrow s4 \rightarrow ...$ where no state satisfies ϕ.

Model Checking Algorithm

Given $\mathcal{M}, \phi$ we compute $SAT(\phi) \subseteq S$. $SAT(\phi) = \{s \in S | s \models \phi\}$ will be an invariant of this algorithm. In the end we get that $\mathcal{M} \models \phi \leftrightarrow SAT(\phi) = S$.

Furthermore if $SAT(\phi) \neq S$ any $s \in S \setminus SAT(\phi)$ will satisfy $s \not\models \phi$.

What is used in a crucial way is that S is a finite set. This means that this will only be an algorithm for *model checking*. Given $\mathcal{M}$ we can decide $\mathcal{M} \models \phi$.

(It is also possible to decide $\models \phi \leftrightarrow \forall \mathcal{M}.\mathcal{M} \models \phi$, but we will not talk about this.)

Idea of Algorithm

$SAT(p) = \{s \in S | s \models p\} = \{s \in S | L\, s\, p = 1\}$
$SAT(\neg\phi) = S \setminus SAT(\phi)$
$SAT(\phi \wedge \phi) = SAT(\phi) \cap SAT(\psi)$
$SAT(\phi \vee \psi) = SAT(\phi) \cup SAT(\psi)$
$SAT(\phi \to \psi) = SAT(\neg\phi \vee \psi) = SAT(\neg\phi) \cup SAT(\psi) = (S \setminus SAT(\phi)) \cup SAT(\psi)$

Given $\mathcal{M} = (S, \to, L)$ we define $N(s) \subseteq S$ where $N(s) = \{s' \in S | s \to s'\}$.
We also consider $Pow(S)$, the set of all subsets of S. $Pow(S)$ is a partially ordered set (poset) for the inclusion relation $I, J \in Pow(S)$, $I \subseteq J$.

We have two operations:

$pre_\forall : Pow(S) \to Pow(S)$, $I \mapsto \{s \in S | N(s) \subseteq I\}$ (if $s \to s'$ then $s' \in I$)
$pre_\exists : Pow(S) \to Pow(S)$, $I \mapsto \{s \in S | N(s) \cap I \neq \phi\}$ (there exists $s' \in I$, s.t. $s \to s'$)

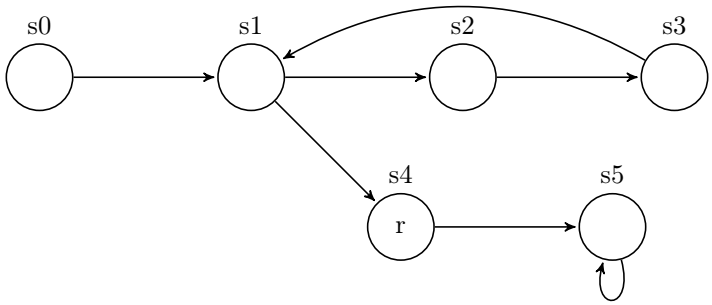
$SAT(AF\phi)$
$Y_0 := SAT(\phi)$, hence $Y_0 \subseteq SAT(AF\phi)$
$Y_1 := Y_0 \cup pre_\forall(Y_0)$, add all states for which all following states satisfies ϕ
$Y_2 := Y_1 \cup pre_\forall(Y_1)$, ...
.
.
$Y_{n+1} := Y_n \cup pre_\forall(Y_n)$, therefore $Y_n \subseteq SAT(AF\phi)$
In this way we build an increasing sequence of subsets of S: $Y_0 \subseteq Y_1 \subseteq Y_2 ...$
S is finite, hence we have to stop eventually and we will get $Y_{n+1} = Y_n$.
Then we have $Y_n = Y_{n+1} = Y_{n+2}$ and the solution is $Y_n = SAT(AF\phi)$.

$SAT(EF\phi)$
$X = SAT(EF\phi)$
$Y_0 := SAT(\phi)$
$Y_{n+1} := Y_n \cup pre_\exists(Y_n)$
$s \in X$ means that we have a path from s into $SAT(\phi)$. We want to collect all states such that there is some path that goes into $SAT(\phi)$.
$s \in Y_0$ means that we have a path of length 0 into $SAT(\phi)$.
$s \in Y_1$ means that we have a path of length ≤ 1 into $SAT(\phi)$.
$s \in Y_n$ means that we have a path of length $\leq n$ into $SAT(\phi)$.
The solution will be $Y_n = SAT(EF\phi)$.

$SAT(EG\phi)$
We can use that $\neg EG\phi \leftrightarrow AF(\neg\phi)$
$SAT(EG\phi) = S \setminus SAT(AF(\neg\phi))$

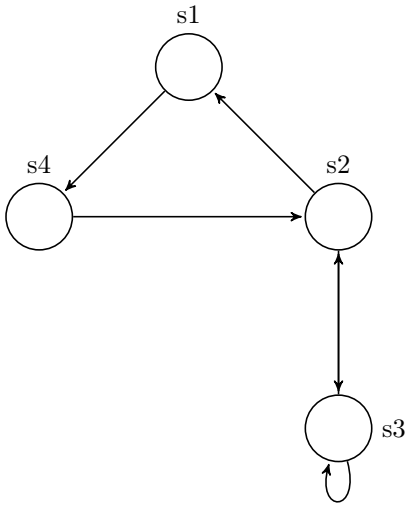
$SAT(AG\phi)$
$SAT(AG\phi) = S \setminus SAT(EF(\neg\phi))$

Examples



$SAT(EFr) = S \setminus \{s5\}$
$Y_0 := SAT(r) = s4$
$Y_1 := Y_0 \cup pre_\exists(Y_0) = s4, s1$
$Y_2 := Y_1 \cup pre_\exists(Y_1) = s4, s1, s0, s3$
$Y_3 := Y_2 \cup pre_\exists(Y_2) = s4, s1, s0, s3$,s2
$Y_4 := Y_3 \cup pre_\exists(Y_3) = s4, s1, s0, s3, s2$

Microwave



$\mathcal{M}=$

	close	starting	cooking
s1	0	0	0
s2	0	1	0
s3	1	0	0
s4	1	1	1

$\mathcal{M} \models AG(start \rightarrow AFcooking)$?
$\mathcal{M} \models AG(start \wedge close \rightarrow AFcooking)$?
$SAT(AFcooking)$?

$N(s2) = \{s1, s3\}$
$N(s2) \nsubseteq Y_0$
$Y_0 := SAT(cooking) = s3$
$Y_1 := Y_0 \cup pre_\forall(Y_0) = s3 = Y_0$
$Y_2 := Y_1 \cup pre_\forall(Y_1) = Y_0$
$SAT(AFcooking) = s3$

$\mathcal{M} \overset{?}{\models} start \rightarrow AFcooking$
$SAT(start) \nsubseteq SAT(AFcooking)$so $\mathcal{M} \not\models start \rightarrow AFcooking$

$\mathcal{M} \models start \wedge close \rightarrow AFcooking$
Yes! $SAT(start \wedge close) = s3 \subseteq SAT(AFcooking)$

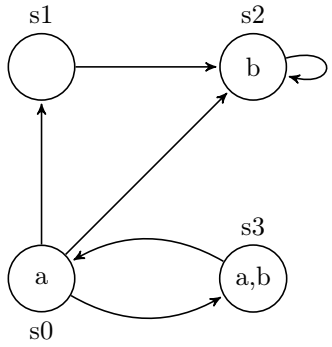
This is an example of fixed-point solution: $F.Pow(S) \rightarrow Pow(S)$, $I \mapsto SAT(\phi) \cup pre_\forall(I)$ where $J = SAT(AF\phi)$ is the least fixed-point of this operation. $F(J) = J$ and $F(I) = I \rightarrow J \subseteq I$.

Look in the book: Chapter 3.7.

DAT060
LV 7, Lecture 3

Example

Fix $\mathcal{M}$
$\mathcal{M} = (\{s0, s1, s2, s3\}, \{(s0, s1), ...\}, L)$
$\Sigma = \{a, b\}$



$O \models_{v(O)=s0} Ga \Leftrightarrow \forall i.i \models_v a$
(i): $\exists v$ s.t. $\forall i.i \models a$?
We assign $v := (s0, s3, s0, s3)$
(ii): $\forall v$ s.t. $\forall i.i \models a$?
This does not hold. E.g. $v := (s0, s2, s2, ...)$

$O \models_{v(O)=s0} GFb \Leftrightarrow \forall i \models_v Fb \Leftrightarrow \forall i \exists j \geq i.j \models_v b$
(i): Yes. $v := (s0, s1, s2, s2...)$
We see that if $i_0 \geq 2$ then $v(i_0) = s2$ and in particular $v(i_0)(b) = True$.
If $i_0 = 0||1$ then $v(2) = s_2$ and $v(2)(b) = True$
(ii): $\forall v \forall i \exists j \geq i.j \models_v b$
Bevisa med Natural Deduction

Equality

$\neg AG\phi \equiv EF\neg\phi \in CTL$
1) $\neg A\psi \equiv E\neg\psi$
2) $\neg G\psi \equiv F\neg\psi$

So, $\neg AG\psi \equiv E\neg G\psi \equiv EF\neg\psi$

Proof of 1

$\forall s.s \models \neg A\psi \leftrightarrow s \models E\neg\psi$,
$s_0 \models \neg A\psi \Leftrightarrow s_0 \not\models A\psi \Leftrightarrow \exists v(v(0) = s_0 \wedge \not\models_v \neg\psi) \Leftrightarrow Ev.(v(0) = s_0 \wedge \models_v \neg\psi) \Leftrightarrow$
$s_0 \models E\neg\psi$

Proof of 2

$\forall i.i \models_v \neg G\psi' \leftrightarrow i \models F\neg\psi'$
$i_0.i_0 \models_v \neg G\psi' \Leftrightarrow i_0 \not\models_v \psi' \Leftrightarrow \exists j \geq i_0.j \not\models \psi' \Leftrightarrow \exists j \geq i_0.j \models \neg\psi' \Leftrightarrow i_0 \models_v F\neg\psi'$

Assignment 5

Problem 1

$\exists\mathcal{M}.\mathcal{M} \models\phi = SAT(\phi)$

In order to show that SAT is undecidable you assume that SAT is decidable. This means that there exists an algorithm A .s.t $\forall\phi.A(\phi) = True \Leftrightarrow SAT(\phi)$. We construct an algorithm B that takes as input: ϕ and outputs: $\neg A(\neg\phi)$.

We show that $\forall\phi.B(\phi) = True \Leftrightarrow \forall\mathcal{M}.\mathcal{M} \models \phi$.

If $B(\phi) = True \Leftrightarrow A(\neg\phi) = False \Leftrightarrow \forall\mathcal{M}.\mathcal{M} \not\models \neg\phi \Leftrightarrow \forall\mathcal{M}.\mathcal{M} \models \phi \Leftrightarrow \phi\, valid$, hence B decides validity. But validity is undecidable so this is a contradiction. Hence, we can conclude that SAT is undecidable.

Problem 2

$\phi \equiv \forall x.P(x, f(x)) \wedge \exists x\forall y.P(x, y)$

A model that satisfies ϕ:$\mathcal{M}_{sat} := (A, P^{\mathcal{M}} = A^2, f^{\mathcal{M}}(x) = x)$

A model that do not satisfies ϕ: $\mathcal{M}_{not} := (\{0, 1\}, P^{\mathcal{M}} := \{(0, 1), (1,0)\}, f^{\mathcal{M}}(x) = \bar{x})$

Problem 3

$\forall x(P(x) \to \neg Q(x)) \vdash \neg\exists x(P(x) \wedge Q(x))$

1	$\forall x(P(x) \to \neg Q(x))$	premise
2	$\exists x(P(x) \wedge Q(x))$	assumption
3	$x0\ P(x_0) \wedge Q(x_0)$	assumption
4	$P(x_0)$	$\wedge e_1$
5	$P(x_0) \to \neg Q(x_0)$	
6	$\neg Q(x_0)$	$\to$e
7	$Q(x_0)$	$\wedge e_2$
8	$\bot$	$\neg$e
9	$\bot$	$\exists e$ 2, 3–8
10	$\neg\exists x(P(x) \wedge Q(x))$	$\neg$i 2–9

Problem 5

$\mathcal{M} \models \phi \Leftrightarrow |\mathcal{M}| = 2$
$\phi_{\geq 2} := \exists x \exists y.x \neq y$
$\mathcal{M} \models \phi \Leftrightarrow |\mathcal{M}| \geq 2$
$\phi_{\geq 3} := \exists x \exists y \exists z.x \neq y \wedge y \neq z \wedge x \neq z$
$\phi_{\geq n} := \exists \bar{x} \bigwedge_{i,j\, i \neq j} x_i \neq x_j$
$\Gamma = \{\phi_{\geq n} | n \in \mathbb{N}\}$
$|\mathcal{M}| = n \wedge \mathcal{M} \models - \Rightarrow \mathcal{M} \models \phi_{\geq n+1} \Rightarrow \bot$

$\mathcal{M} \models \Gamma \Leftrightarrow |\mathcal{M}| = \infty$

$\mathcal{M} \models \wedge\phi \Leftrightarrow |\mathcal{M}| = \infty$
$\psi := \neg \bigwedge_{\phi \in \Gamma} \phi\, \psi \notin PL = \bigvee_{\phi \in \Gamma} \neg\phi$
$\mathcal{M} \models \psi \Leftrightarrow \neg|\mathcal{M}| = \infty \Leftrightarrow |\mathcal{M}| \leq \infty$

Logic in Computer Science

For a given language $\mathcal{F}, \mathcal{P}$, a *first-order theory* is a set T of sentences (closed formulae) in this given language. The elements of T are also called *axioms* of T.

A model of T is a model $\mathcal{M}$ of the given language such that $\mathcal{M} \models \psi$ for all sentences ψ in T.

$T \vdash \varphi$ means that we can find $\psi_1, \ldots, \psi_n$ in T such that $\psi_1, \ldots, \psi_n \vdash \varphi$.
$T \models \varphi$ means that $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$ of T.

The generalized form of *soundness* is that $T \vdash \varphi$ implies $T \models \varphi$ and *completness* is that $T \models \varphi$ implies $T \vdash \varphi$.

If T is a finite set $\psi_1, \ldots, \psi_n$ this follows from the usual statement of soundness ($\vdash \delta$ implies $\models \delta$) and completness ($\models \delta$ implies $\vdash \delta$). Indeed, in this case, we have $T \vdash \varphi$ iff $\vdash (\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi$ and $T \models \varphi$ iff $\models (\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi$.

Compactness Theorem

Theorem 0.1 *A theory has a model iff any of its finite subtheory has a model*

Application 1: non-standard model

We recall that the theory of Peano arithmetic PA is a theory for the language $\mathcal{F} = \{\mathsf{zero}, \mathrm{S}, +, \cdot\}$ and with no predicate symbol apart from equality. We add the special constant u with the axioms

$$u \neq \mathsf{zero},\ u \neq \mathrm{S}(\mathsf{zero}),\ u \neq \mathrm{S}(\mathrm{S}(\mathsf{zero})),\ \ldots$$

By the Compactness Theorem, this theory has a model. The domain of this model has to contain an element which is different from the semantics of zero, $\mathrm{S}(\mathsf{zero})$, $\mathrm{S}(\mathrm{S}(\mathsf{zero})), \ldots$ This is a *non standard* model of arithmetic.

Application 2: transitive closure is not first-order definable

In the language with one binary relation symbol R and two constant a, b, we can state

Theorem 0.2 *There is no formula φ such that $\mathcal{M} \models \varphi$ iff there is a path from $a^{\mathcal{M}}$ to $b^{\mathcal{M}}$*

Indeed, if there was such a formula φ then the theory $\varphi,\ \neg\delta_0,\ \neg\delta_1, \ldots$ would be consistent, by the Compactness Theorem, where δ_0 is $a = b$ and δ_{n+1} is $\delta_n \vee \exists z_1\ \ldots\ z_n.R(a, z_1) \wedge \cdots \wedge R(z_n, b)$. But this is a contraction.

Application 3: to be well-founded is not first-order definable

We recall that a relation S is well-founded iff there is no infinite sequence $x_0,\ x_1, \ldots$ such that $S(x_0, x_1),\ S(x_1, x_2), \ldots$. In the language with one binary relation symbol R we can state

Theorem 0.3 *There is no formula φ such that $\mathcal{M} \models \varphi$ iff $R^{\mathcal{M}}$ is well-founded.*

We add to the language infinitely many constants $a_0,\ a_1,\ a_2,\ \ldots$ and, if there is such a formula φ, we consider the theory

$$\varphi, R(a_0, a_1),\ R(a_1, a_2),\ R(a_2, a_3), \ldots$$

By the Compactness Theorem, this theory has a model, which is a contradiction.

Three traditions in logic

Before starting the presentation of Linear Temporal Logic, I started to recall the 3 traditions in logic, that are important for propositional logic (and temporal logics)

1. model theory
2. proof theory
3. algebraic logic

We present this in the case of propositional logic, where the syntax is

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \to \varphi$$

where p ranges over atoms. We can then define $\psi_0 \vee \psi_1 = \neg\psi_0 \to \psi_1$ and $\psi_0 \wedge \psi_1 = \neg(\psi_0 \to \neg\psi_1)$.

Model Theory

In the model theoretic approach, we start by defining what is a model α which is a function from the atomic formulae to $\{0, 1\}$. We then define $\alpha \models \varphi$ by induction on φ.

We write $\models \varphi$ iff $\alpha \models \varphi$ for all models α.

Proof Theory

In the proof theoretic approach, we define when φ is *derivable*, notation $\vdash \varphi$, and more generally, when φ is derivable from hypotheses $\psi_1, \ldots, \psi_k$, notation $\psi_1, \ldots, \psi_k \vdash \varphi$.

In this course, we presented this following the notion of *natural deduction*.

Another way to present the notion of derivability is via the so-called notion of *Hilbert-style* proof system (which was actually already in Frege). It consists in giving some axioms and to say that φ is derivable iff we can build a derivation tree using as the only derivation rule the *modus-ponens rule*

$$\frac{\psi \qquad \psi \to \delta}{\delta}$$

and the leaves are axioms.

For instance, for proposition a possible axiom system is the given by the 3 axiom schemas

- $\varphi \to \psi \to \varphi$
- $(\varphi \to \psi \to \delta) \to (\varphi \to \psi) \to \varphi \to \delta$
- $(\neg\varphi \to \psi) \to (\neg\varphi \to \neg\psi) \to \varphi$

With this presentation it is not at all obvious that, e.g. $p \to p$ is derivable!

Both presentations are actually equivalent, and we have $\vdash \varphi$ iff $\models \varphi$.

Algebraic logic

An important remark if that, if we define $\varphi \equiv \psi$ by $\alpha \models \varphi$ iff $\alpha \models \psi$ (or equivalently $\vdash \varphi \to \psi$ and $\vdash \psi \to \varphi$), then we have the rules

$$\frac{\varphi \equiv \psi}{\neg\varphi \equiv \neg\psi} \qquad \frac{\varphi_0 \equiv \psi_0 \qquad \varphi_1 \equiv \psi_1}{\varphi_0 \to \varphi_1 \equiv \psi_0 \to \psi_1}$$

It is then natural to write simply $\varphi = \psi$ instead of $\varphi \equiv \psi$ and to consider that we have two operations (negation and implication). It is also natural to write $\varphi \leqslant \psi$ instead of $\vdash \varphi \to \psi$.

We see then the set of formulae as a set equipped with some operations, satisfying some algebraic laws (e.g. $1 = p \to p$). The relation $\leqslant$ is a poset relation.

This was the view of logic coming from Boole (1815-1864). One can consider more generally algebras satisfying the same laws as the one of proposition formulae, and these are called Boolean algebras. In term of the operations $\neg, \vee$, one possible list of equational axioms for Boolean algebra is

$$x \vee (y \vee z) = (x \vee y) \vee z \qquad x \vee y = y \vee x \qquad x \vee 1 = 1 \qquad x \vee 0 = x$$

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z \qquad x \wedge y = y \wedge x \qquad x \wedge 1 = x \qquad x \wedge 0 = 0$$

$$\neg(x \vee y) = \neg x \wedge (\neg y) \qquad 1 = \neg\, 0 \qquad 0 = \neg 1 \qquad \neg(\neg x) = x$$

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \qquad x \wedge (x \vee y) = x$$

In the algebraic approach, we can consider more general algebras than the algebras of propositional formulae.

In this approach, a natural question is how to solve equations. For instance, it can be shown (exercise) that the equation in x

$$(x \wedge b) \vee (\neg x \wedge (a \vee \neg b)) = 1$$

has exactly the solution $x = b \wedge (\neg a \vee u)$ where u is arbitrary.

For propositional logic, these three approaches, model theoretic, proof theoretic and algebraic are equivalent, but they provide very different intuitions.

Logic in Computer Science

Model checking

A *transition system* (or Kripke frame) is a triple (S, R, L) where S is a finite set of states, $R(s,t)$ a binary relation on S such that

$$\forall s \ \exists t \quad R(s,t)$$

and L is a labelling function, so that $L\ s$ gives a value 0 or 1 to each atom.

A *path* or *behavior* or *possible run of a program* for this transition system is an infinite sequences of state $\pi = \pi_0, \pi_1, \pi_2, \ldots$ such that $R(\pi_n, \pi_{n+1})$ for all n.

To such a path, we can associate a model α of LTL by taking $\alpha\ p\ n = L\ \pi_n\ p$ and we define $\pi \models \varphi$ to mean $\alpha \models \varphi$. (This is equivalent to the definition presented in the book.)

We define $(S, R, L) \models \psi$ to mean $\pi \models \psi$ for all path π of (S, R, L).

A *model-checker* for LTL is an algorithm deciding $(S, R, L) \models \psi$.

Example of a LTL model-checking problem

It is possible to encode the *Hamiltonian Path Problem* as a LTL model-checking problem. The Hamiltonian Path Problem is the following problem: given a graph (V, G) to decide if there is a way to enumerate V as a sequence of vertices $v_1, \ldots, v_n$ (where each vertex appears exactly once) and such that $G(v_1, v_2), \ldots, G(v_{n-1}, v_n)$. This is a well-known NP-complete problem.

For this reduction, we introduce the atoms p_v for each v in V and define the following transition system. We take S to be $V \cup \{b\}$ where b is not in V and add new edges $R(v, b)$ for all v in V and $R(b, b)$, and $R(v, v')$ if $G(v, v')$. We then have

$$\forall s \ \exists t \quad R(s,t)$$

The labelling function is defined by taking $L\ b\ p_v = 0$ and $L\ v'\ p_v = 1$ if $v = v'$ and $L\ v'\ p_v = 0$ if $v \neq v'$.

The following formula ψ is then such that $(S, R, L) \models \psi$ iff the Hamiltonian Path Problem has *not* a solution

$$\psi = \bigvee_{v \in V} (G(\neg p_v) \vee F(p_v \wedge XF(p_v)))$$

Indeed this implies that for any path π, there exists v such that either π does not visit v or π visits v twice.

Logic in Computer Science

CTL: some corrections

For a model M we don't have that $M \models \varphi \leftrightarrow \psi$ is equivalent to $M \models \varphi \leftrightarrow M \models \psi$ (exercise: find a counter-example). What we have is that $M \models \varphi \leftrightarrow \psi$ is equivalent to $s \models \varphi \leftrightarrow s \models \psi$ for all states of M.

Similarly to have $M \models \varphi \rightarrow \psi$ is the same as having $s \models \varphi \rightarrow s \models \psi$ for all states of M which is *not* the same (exercise) as $M \models \varphi \rightarrow M \models \psi$.

Because of this, we don't have in general

$$\models (\varphi \rightarrow EX\varphi) \rightarrow (\varphi \rightarrow EG\varphi)$$

but what we have is that, *if* $M \models \varphi \rightarrow EX\varphi$ *then* $M \models \varphi \rightarrow EG\varphi$