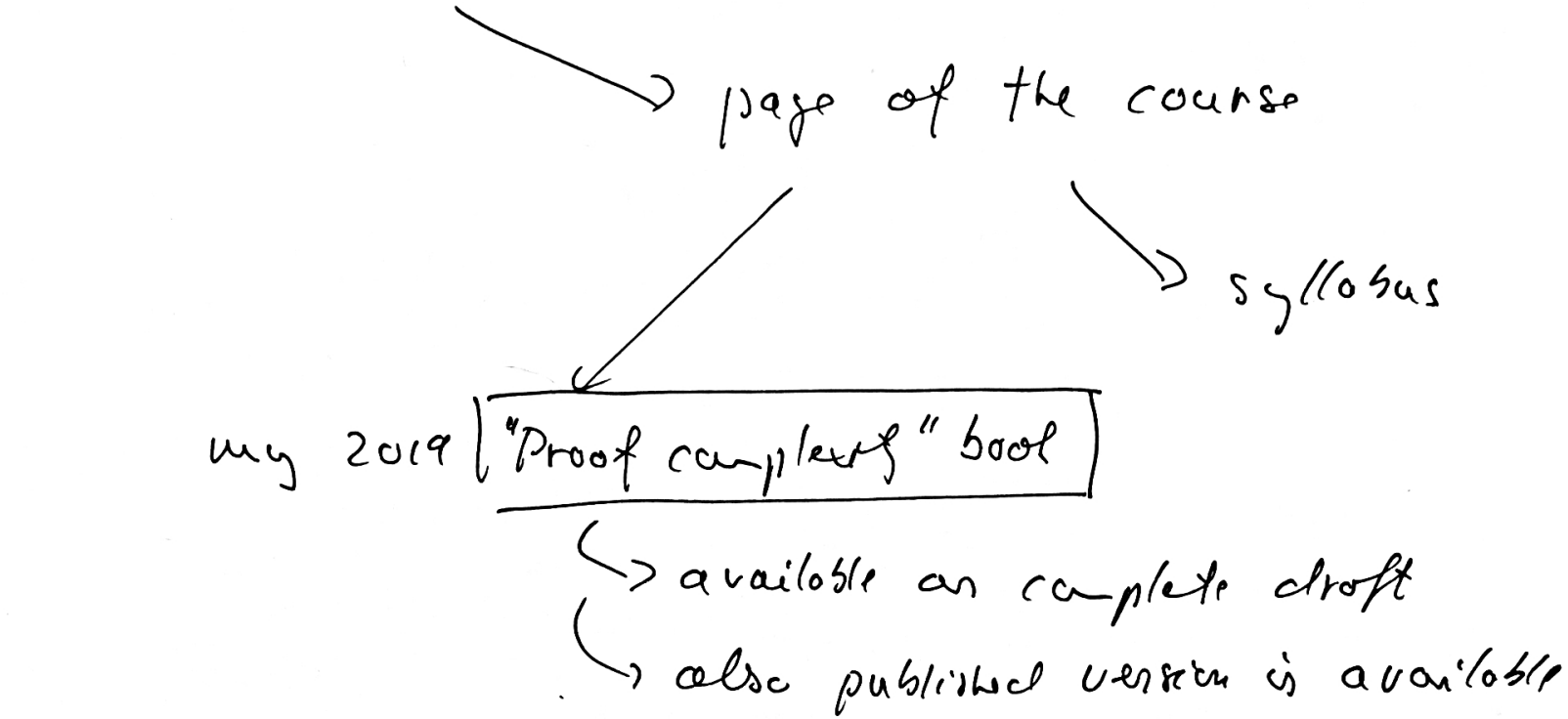
PROOF COMPLEXITY

and

the P vs. NP problem

Lecture 1

on my web page



[Sec. x.y], [Chpt. x], etc ... refers to this book

slides

→ will be online (usually before the lecture)

→ errata will be marked in yellow in the slides

lecture page : - also info about Sections where the resp. material is

- further comments

expected background

- basic propositional & predicate logic (= FO)
  (formulas, logical ax's, inference rules, proofs, ...)
- Turing machines, time complexity, P, NP
  (will also discuss today)

[Secs. 1.1, 1.2, 1.3]

The P vs. NP problem

→ Cook's 1971 paper → SAT problem

→ NP-completeness

early 1960s: Bennett, Cobham,

Samuilya, Edmonds, ...

⋮

notion of [p-time algorithm]

as a model of

feasible alg's

..... → von Neumann

Gauss

("an algorithm is

good karte")

Spectrum problem

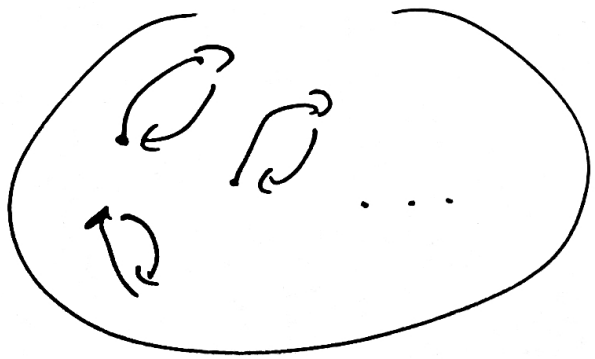
Scholz '52, Asser '53

φ : a FO sentence

$$\boxed{Spec(\varphi)} := \{ n \geq 1 \mid \varphi \text{ has a model of size } n \}$$

(Ex.) $L = \{f(\cdot)\}$

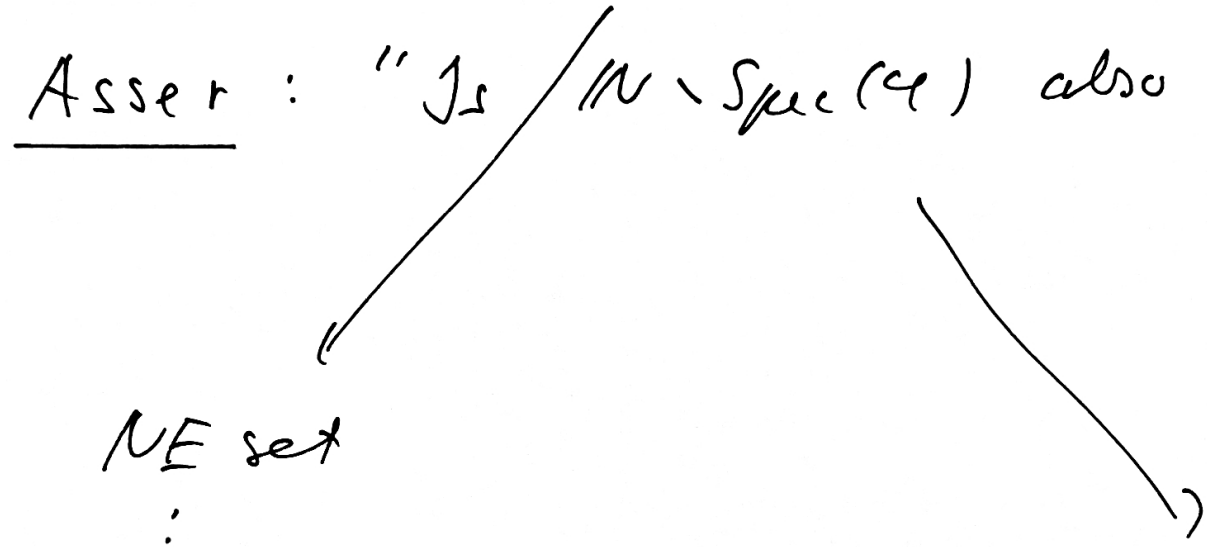
$\varphi : (\forall x\ x \neq f(x)) \wedge (\forall x, x = f(f(x)))$



$Spec(\varphi)$ = even nb's

Scholz : "Characterize Spec(q) sets"

Asser : "Is $\mathbb{N} \setminus Spec(q)$ also a spectrum?"



Fact : $P = NP \Rightarrow NE = coNE$, so showing "expected" $NE \neq coNE$ solves the P/NP problem.

Gödel's 1956 letter to von Neumann

→ discusses the following question:

- input: FO sentence φ, $n \geq 1$
- question: decide if there is a proof of φ (in ZFC, e.g.) with $\leq n$ symbols

Points out: - exhaustive search needs $\approx \exp(n)$ steps

- there is no alg. using just $\approx O(n)$ steps
- Asks: Can there be alg. using $\approx O(n^2)$ steps.
- ↳ YES ⇒ looking for proofs (and doing math) can be automatized

[The letter is online - see the lecture page]

P.

De Morgan l. : $0, 1, \wedge, \vee, \neg$ ... $\bot$ ... $\top$ atoms: $\ldots x_i, y_i \ldots$ $\ldots p_i, q_i \ldots$

SAT := the set of satisfiable fms $\varphi(t_1, \ldots, t_n)$

TAUT := the set of tautologies (= logically valid) fms

Observation: $\varphi \in SAT \iff \neg\varphi \notin TAUT$. □

CNF-formula

$\ldots \wedge (\ldots) \wedge \underbrace{(l_1 \vee l_2 \vee \ldots \vee l_k)}_{\text{a clause}} \wedge (\ldots) \wedge \ldots$

a clause $\overset{\text{def.}}{\Leftrightarrow}$ a disjunction of literals

literal → atom or ¬atom

DNF-formula (dual)

$\ldots \vee \underbrace{(l'_1 \wedge \ldots \wedge l'_k)}_{\text{logical term}} \vee \ldots$

P := the class of languages $L \subseteq \{0,1\}^*$

s.t. there is a (deterministic) Turing machine

M s.t.:

(i) M computes $\mathcal{X}_L$:

$$M(x) = 1 \Leftrightarrow x \in L \,, \quad M(x) = 0 \Leftrightarrow x \notin L$$

(ii) M runs in time $\leq n^{O(1)}$ $(= n^{\text{const}})$

on inputs of size n, $n \geq 1$.

idea: p-time alg's $\approx$ feasible alg's

How accurate this idea is ?

- time n^{100} : "most" steps on inputs $n = 1000$

  ↖ rarely happens - in artificial ex's

- randomized alg's ——→ it is conjectured that they do not solve more than deter. alg's ($P = BPP$)

  ↘ Are there actually random bits in Nature?

- quantum p-time ——→ e.g. Shor's alg. for factoring

  ↘ not clear if they can be built

  so this objection to the thesis p-time = feasible is hypothetical at present

NP : the class of $L \subseteq \{0,1\}^*$ s.t. there are $c \geq 1$ and p-time checkable relation $R(x,y)$ s.t. for all $u \in \{0,1\}^*$:

$$u \in L \iff \exists v \, (|v| \leq |u|^c) \, R(u,v)$$

$\Big\uparrow$

think of v as a "witness to $u \in L$"

Ex: L = composite numbers

v : prime decomposition

the problem: $P =? NP$

obvious: $P \subseteq NP$

facts: (i) $P \subseteq NP \subseteq EXP$ ... det. alg's using time $2^{n^{const}}$

(ii) $P \neq EXP$

Hence either $P \neq NP$ or $NP \neq EXP$ or both.

so

this is expected

p-reduction: $f : \{0,1\}^* \to \{0,1\}^*$

$L_1 \leq_p L_2$: f ... p-time

$$u \in L_1 \iff f(u) \in L_2$$

This is called a computability r. many-one reduction.

More general are Turing reductions: $u \in_? L_1$ can be solved in p-time by a machine which can ask questions $v \in_? L$.

L is NP-complete

$\Updownarrow$ def.

(i) $L \in NP$

(ii) $L' \leq_p L$, for <u>all</u> $L' \in NP$

Informally: L is the hardest problem in NP.

<u>Fact</u>: $P = NP \iff$ some NP-complete $L \in P$

Cook's theorem : SAT is NP-complete

Remarks :

(i) the ∃ of NP-complete problem is a simple consequence of the ∃ of universal T. machine

(ii) Cook used Turing reduction : Karp used many-one red. used by Karp '72

(iii) Levin '73 : ~~the~~ complete NP-record problem

(iv) there are many thousands of NP-compl. problems known

a link with propositional tautologies

coNP := the class of complements of NP-languages

Ex: UNSAT = unsatisfiable formulas

$\varphi \in UNSAT \iff \neg\varphi \in TAUT$

(before we noticed

$\varphi \in SAT \iff \neg\varphi \notin TAUT$)

big difference: switches $\exists$ into $\forall$

Fact: TAUT is coNP-complete

Observation: $NP \neq coNP \iff TAUT \notin NP \Rightarrow P \neq NP$

How it could be that $TAUT \in NP$?

For example: all $\tau \in TAUT$ have p-size propositional proof: $\pi\ (|\pi| \leq |\tau|^c)$

Then:

$\tau \in TAUT \iff \exists \pi\ (|\pi| \leq |\tau|^c)$ "π is a proof of τ"

Hence: we want to show that this cannot happen

a reduction of fms to CNF-fms

distributive De Morgan rules:

$$(a \wedge b) \vee (c \wedge d)$$
$$\Updownarrow$$
$$(a \vee c) \wedge (a \vee d) \wedge (b \vee c) \wedge (b \vee d)$$

applied to

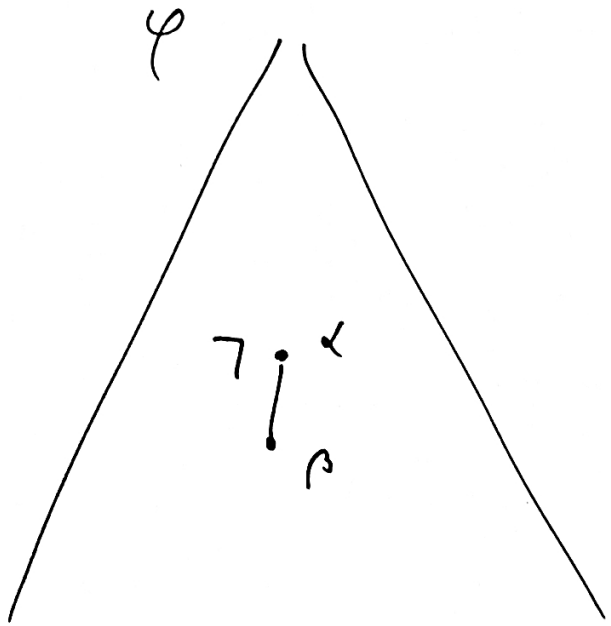
$$(a_1 \wedge b_1) \vee \ldots \vee (a_k \wedge b_k)$$

blow-up the size to $\geq 2^k$. This is bad.

Limited extension (Tseitin '68)

- for each subf α of $\chi(t_1, \ldots, t_n)$ introduce new atm y_α

and 2 or 3 clauses



(22.)

$\alpha = \beta \vee \gamma$ ... analogous

Y_0 .... $\neg Y_0$

Y_1 - - - Y_1

$\alpha = x_{i,\cdot}$ - - - .. $\neg x_{i,\cdot} \vee Y_1$

$\neg Y_1 \vee x_{i,\cdot}$

<u>all new clauses</u> : set $\Gamma(x, y)$

<u>Facts</u> : (i) $|\Gamma| \leq 3 \cdot |\varphi|$

(ii) $\varphi \in SAT \iff \bigwedge_{\gamma \in \Gamma} \gamma \in SAT$

$\gamma \in \Gamma$ — this is CNF

Lemma: $SAT \leq_p CNF\text{-}SAT$

and similarly

$$TAUT \leq_p DNF\text{-}TAUT.$$

□

Hence for proving facts we may, w.l.o.g., restrict to DNF formulas, when convenient.

Eg.

back to Hilbert, ≈ 1920s

↳ Entscheidungsproblem:

"Devise an algorithm deciding the logical validity of FO sentences."

Turing '36, Church '36 : it does not ∃

↳ (Universal) Turing machines

Halting problem

⋮

a transformation:

FO sentences ⇝ propositional fla

general alg. ⇝ p-time alg.

⇓

Entscheid... ⇝ P/NP problem

[Fundamental questions sometimes return ...]

//