LECTURE 5

→ TOWARDS LOWER BOUNDS

NEED <u>"HARD" FLAS</u>

↳ $\tau_n, n \geq 1, |\tau_n| \nearrow \infty$

→ "RANDOM" DNFs : $D_1 \vee \ldots \ldots \vee D_k$

↳ whp ARE TAUTOLOGIES : DEPENDS ON THE RELATION OF nb. of vars / k / $|D_i|$

<u>NOT VERY USEFULL</u> : LOWER BOUNDS FOR THEM COME SECOND...

NEED FLAS τ_n THAT "DESCRIBE" SOME MATH.

FACT

→ in ZFC YOU ALSO DO NOT SHOW INDEPENDENCE FOR "RANDOM" SENTENCE

→ THEN WE CAN TRY TO "MODEL" A FALSIFYING ASSIGNMENT AND USE IT TO SHOW THAT NO SHORT PRF CAN RULE IT OUT

PIGEONHOLE PRINCIPLE $PHP_n(f)$:

"NO $f : [n+1] \xrightarrow[1\text{-to-}1]{} [n]$ EXISTS"

=

onto PHP_n : NO BIJECTION $f : [n+1] \twoheadrightarrow [n]$ EXISTS

—

$[n] := \{1, \ldots, n\}$

[Sec. 1.5]

PROPOSITIONAL VERSION

ATOMS : p_{ij} , $i \in [n+1]$, $j \in [n]$: REPRE'S "$f(i)=j$"

THE FLA :

$$\neg \left[\bigwedge_i \bigvee_j p_{ij} \wedge \bigwedge_i \bigwedge_{j \neq j'} (\neg p_{ij} \vee \neg p_{ij'}) \wedge \bigwedge_{i \neq i'} \bigwedge_j (\neg p_{ij} \vee \neg p_{i'j}) \right]$$

↑ f defined everywhere

dom(f) = [n+1]

↑ it is a FUNCTION

↑ [THIS COND. is NOT NEEDED, in FACT]

↑ it is 1-to-1

$\neg PHP_n$ AS AN UNSAT SET OF CLAUSES

- $\{p_{i1}, \ldots, p_{in}\}$, ONE FOR EACH $i \in [n+1]$
- $\{\neg p_{ij}, \neg p_{ij'}\}$, —"— $i \in [n+1]$, $j \neq j' \in [n]$
- $\{\neg p_{ij}, \neg p_{i'j}\}$, —"— $i \neq i' \in [n+1]$, $j \in [n]$

- $\{p_{1j}, \ldots, p_{n+1,j}\}$, ONE FOR EACH $j \in [n]$

↖ THIS FORCES f TO BE ONTO.

$\neg ontoPHP_n$

OUR GOAL

- SHOW THAT $\mathrm{Onto}\,PHP_n$ IS NOT REFUTABLE IN A FREGE SYSTEM WHEN ALL FLAS HAVE <u>BOUNDED DEPTH</u>
  - ↳ a AC^0-FREGE

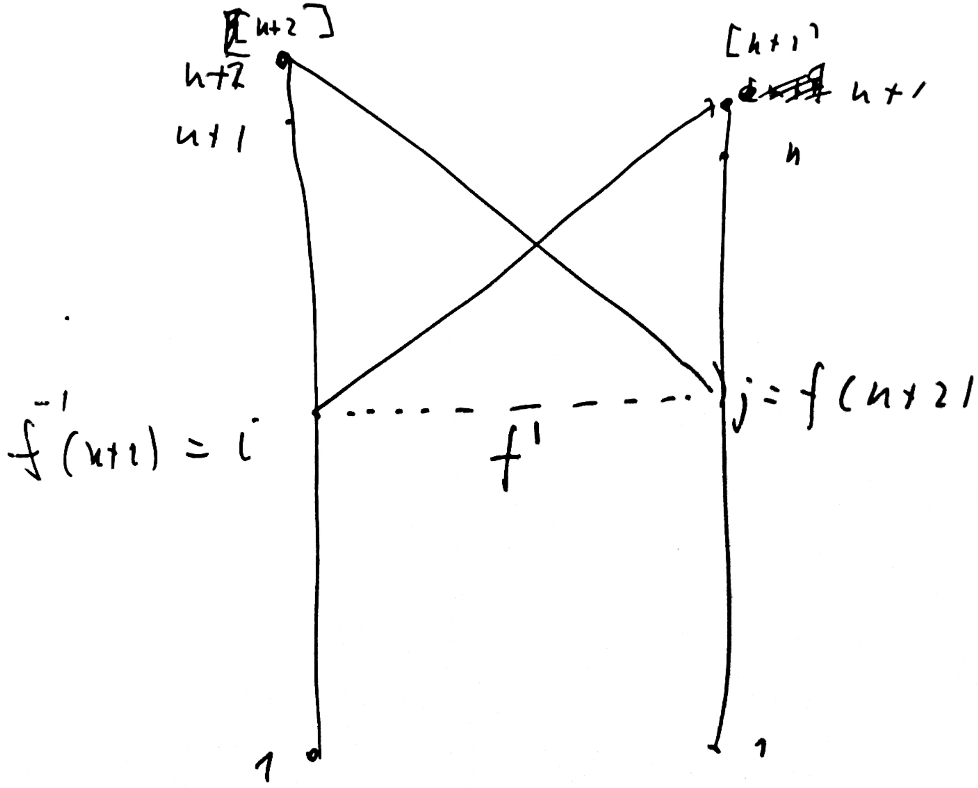
- FIRST: LOWER BOUNDS FOR R^* (tree-like R) AND SOME EXT'S.

AS WARM-UP LET'S LOOK AT <u>UPPER</u> BOUNDS:

FO-proof of $\mathrm{ontoPHP}_n(f)$ BY IND ON n:

(1) $n = 1$ ..... NO $f : [2] \longleftrightarrow [1]$: EXHAUSTIVE SEARCH

(2) $\neg \mathrm{onto} PHP_{n+1}(f) \longrightarrow \exists f' \neg \mathrm{onto} PHP_n(f')$



<u>OTHERWISE</u> $f' = f$

TO SIMULATE THE DEF. OF f' FROM f IN PROP. LOGIC:

$$p'_{ij} := p_{ij} \vee (p_{i,n+1} \wedge p_{n+1,j})$$

$i \in [n+1]$

$j \in [n]$

- 1 new atom $\Leftarrow$ 3 old atoms
- in THE IND $n+1 \to n \to n-1 \to n-2 \to \dots (n-t)$

  This would grow EXPONENTIALLY: $\sim 3^{t+1}$

SO FOR $t = n$ $\sim 3^n$. BAD!

ALTERNATIVE: INTRODUCE (FOR FREGE SYSTEM)

NEW "EXTENSION RULE":

= EF = EXTENDED FREGE

Secs 2.4 + 10.3

if $\theta_1, \ldots, \theta_i$

is a proof we can intro as θ_{i+1} a fla of the form:

$$q \equiv \eta$$

WHERE:
- q does not occur in $\theta_1, \ldots, \theta_i$
- — " — in η
- q — " — in the target fla we want to prove.

q: ABBREVIATES η, can be used in $\theta_{i+2}, \ldots$

ANOTHER NON-PROP. PROOF

USE COUNTING TO SHOW THAT IF f IS A BIJECTION THEN:

(i) $|f(x)| = |x|$, ANY FINITE x

(ii) $|[n+1]| \neq [n]$

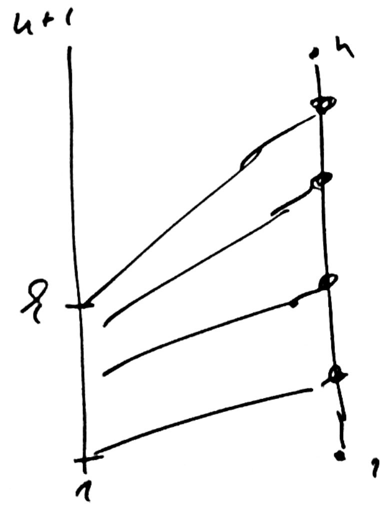
BUT HOW TO COUNT IN PROP. LOGIC?

COUNTING FLAS

$C_{n,k}(p_1, \ldots, p_n)$ $\quad 0 \le k \le n, \ 1 \le n$

$$C_{n,k}(p_1, \ldots, p_n) = 1 \iff |\{p_i \mid p_i = 1\}| = k$$

THEN PROVE (IN FREGE):



(i) $C_{n,k}\left(\bigvee_{i \in [k]} p_{i1}, \ldots, \bigvee_{i \in [k]} p_{in}\right)$

BY IND ON k, AND

(ii) $C_{n,n}(q_1, \ldots, q_n) \models \bigwedge_{j \le n} q_j$.

I.E. NO ROOM FOR j S.T. $p_{n+1,j}$.

[Sec. 10.2]

FACT (BUSS)

THERE ARE DE MORGAN FLAS $C_{n,k}(p_1,\dots,p_n)$ S.T. FREGE SYSTEMS IN DE MORGAN LANGUAGE PROVE:

(a) $C_{n,0}(\bar{p}) \equiv \bigwedge_j \neg p_j$

(b) ~~$C_{n,k}(\bar{p}) \equiv [C_{n+1,k+1}(\bar{p},q) \equiv$~~

$$C_{n,k+1}(\bar{p}) \equiv [(C_{n-1,k+1}(p_1,\dots,p_{n-1}) \wedge \neg p_n) \vee (C_{n-1,k}(p_1,\dots,p_{n-1}) \wedge p_n)]$$

BY PROOFS OF SIZE POLY(n).

IN PARTICULAR, $|C_{n,k}| \leq \text{poly}(n)$.

[Secs. 11.2 + 11.3]

SUMMARY: LOWER BOUNDS FOR $\text{onto}PHP_n$ CAN BE TRUE ONLY FOR SYSTEM $\not\geq$ FREGE.

OUR FIRST TARGET:

R^* = tree-like R
(RESOLUTION)

I.E. WE WILL CONSIDER <u>REFUTATIONS</u> OF $\neg\text{onto}PHP_n$

[Sec. 13.1]

$\mathcal{C}$: A SET OF (INITIAL) CLAUSES

UNSATISFIABLE

⇓

SEARCH PROBLEM <u>SEARCH ($\mathcal{C}$)</u>

- Given: TRUTH ASSIGN. $a \in \{0,1\}^*$ to ATOMS

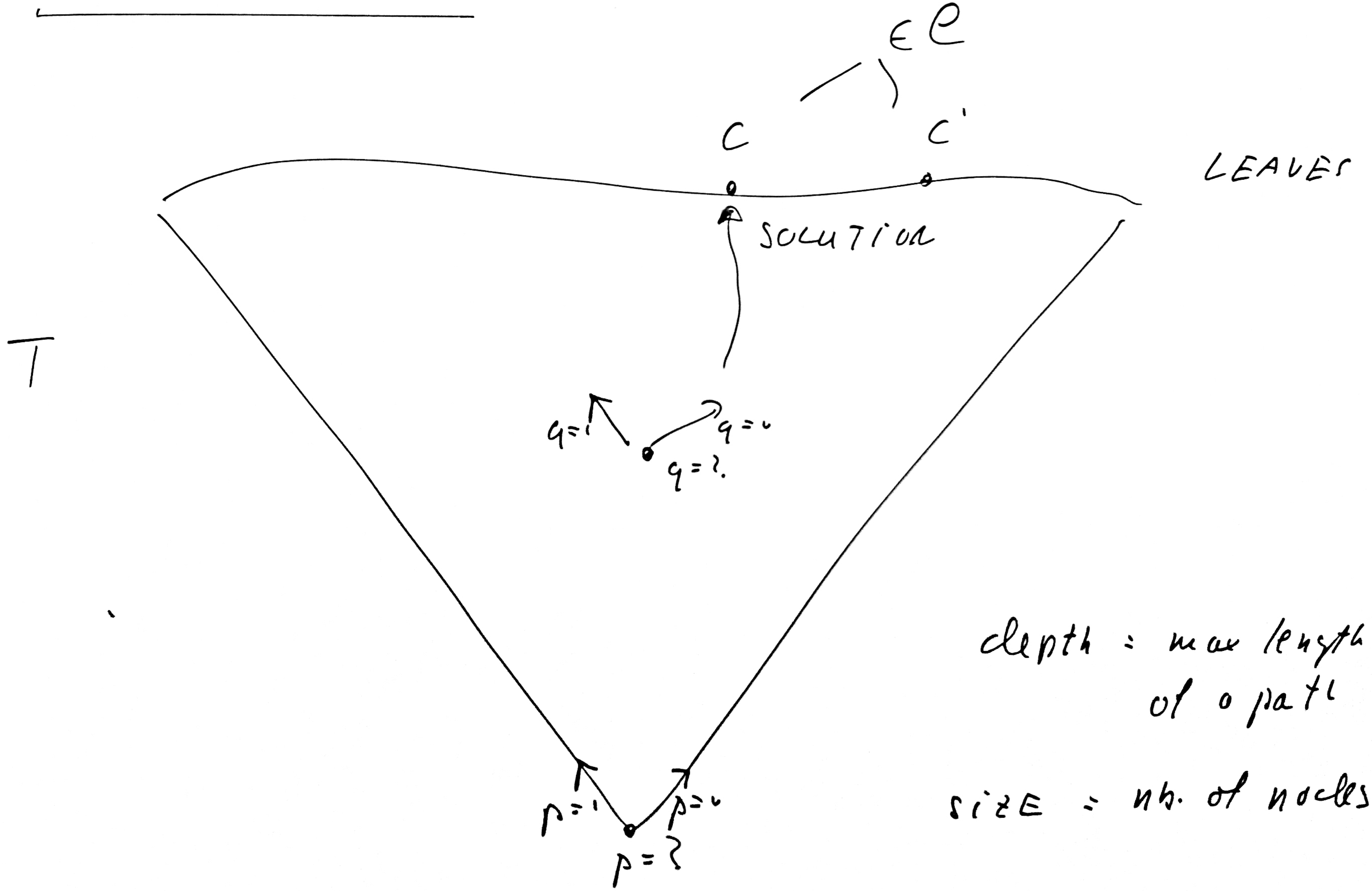
- TASK: FIND a CLAUSE $C \in \mathcal{C}$ S.T.

$$C(a) = 0$$

<u>L. BOUND IDEA:</u> PROOF ⇝ COMPUT. DEVICE TO SOLVE

⇝ PROVE L. BOUNDS FOR THAT

DECISION TREES



depth : max length of a path

size : nb. of nodes

LEMMA: ASSUME THERE IS AN R^*-REFUT. π OF $\mathcal{C}$ WITH PROOF-TREE T_π. THEN THERE IS A DECISION TREE T SOLVING SEARCH($\mathcal{C}$) WHOSE UNDERLYING TREE IS T_π.

PRF: THIS IS THE $R^* \Leftrightarrow$ DPLL RELATION WE DISCUSSED IN ~~SECTIO~~ LECTURE 4.

□

COROLLARY: IF EVERY DEC. TREE SOLVING SEARCH($\mathcal{C}$) MUST HAVE THE DEPTH $\geq D$ OR THE SIZE $\geq S$, SAME IS TRUE ABOUT R^*-PROOFS.

□

LEMMA: EVERY DEC. TREE T SOLVING SEARCH($\neg$onto-PHP_n) MUST HAVE THE DEPTH $\geq n$.

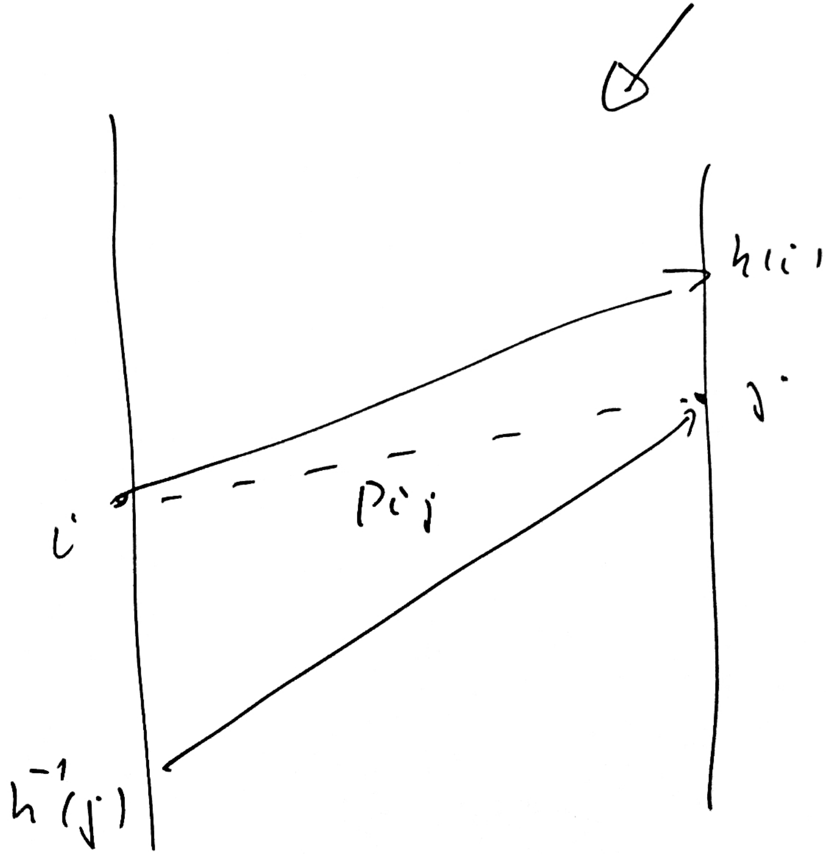
PRF:

WE WANT TO ANSWER T's QUERIES WITHOUT REVEALING WHICH $C \in \neg$onto-PHP_n FAILS, I.E. THE ANSWERS CANNOT EXPLICITLY FALSIFY ANY CLAUSE OF $\neg$onto-PHP_n.

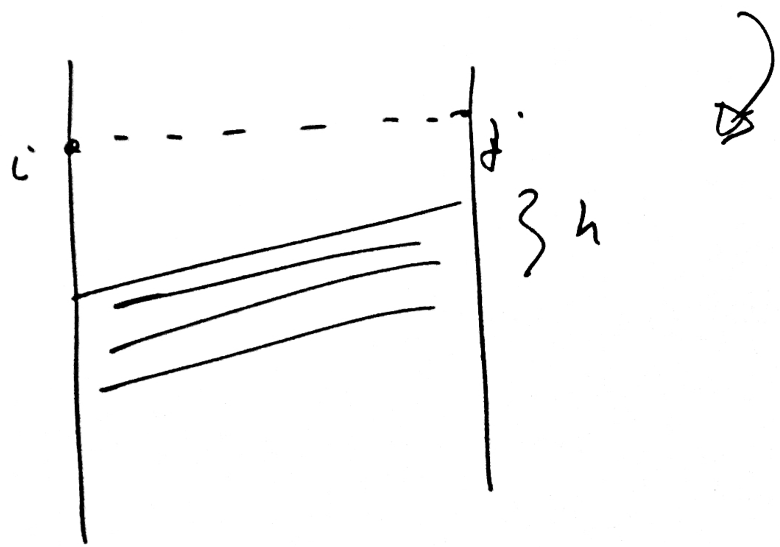
FOR PARTIAL 1-to-1 $h: \subseteq [n+1] \longrightarrow [n]$

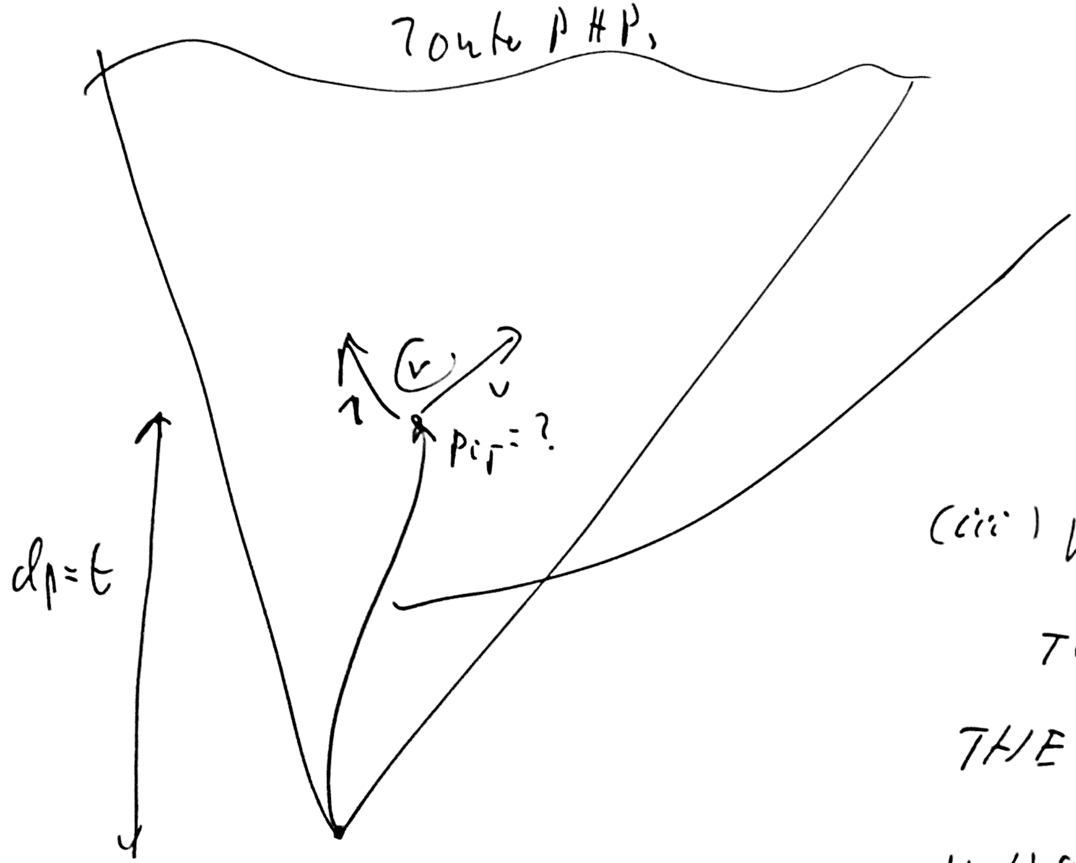
DEFINE: $h \models p_{ij} = 1 \iff i \in dom(h) \wedge h(i) = j$

$h \Vdash p_{ij} = 0 \iff (i \in dom(h) \wedge h(i) \neq j)$
$\vee\ (j \in rng(h) \wedge h^{-1}(j) \neq i)$



NOTE: IT MAY BE THAT
$h \not\Vdash p_{ij} = 1$ and $h \not\Vdash p_{ij} = 0$





BUILD $h_0 := \emptyset \subseteq h_1 \subseteq \dots \subseteq h_t$

(i) $|h_i| \le i$

(ii) $h_i : \subseteq [n+1] \xrightarrow[1-1]{} [n]$

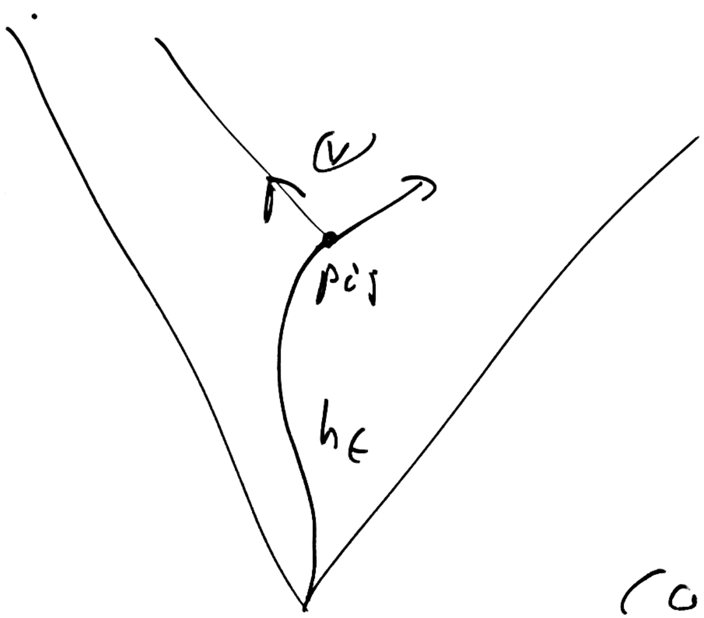
(iii) h_t DETERMINES THE PATH TO v IN THE SENSE THAT THE PATH USES EDGES WHOSE VALUES ARE ~~FORCED FALSE~~ FORCED TRUE BY h_t

~~(iii)~~

↖ $p_{ij} = 1$ ... $h_t \Vdash p_{ij} = 1$

↗ $p_{ij} = 0$ .... $h_t \Vdash p_{ij} = 0$

IND. STEP $t \to t+1$



Case 1: $h_t \cup \{(i,j)\}$ is a partial 1-to-1
$\Rightarrow h_{t+1} :=$ — " — AND GO
ALONG THE EDGE $p_{ij} = 1$.

Case 2: NO, $h_{t+1} := h_t$ AND
USE EDGE $p_{ij} = 0$.

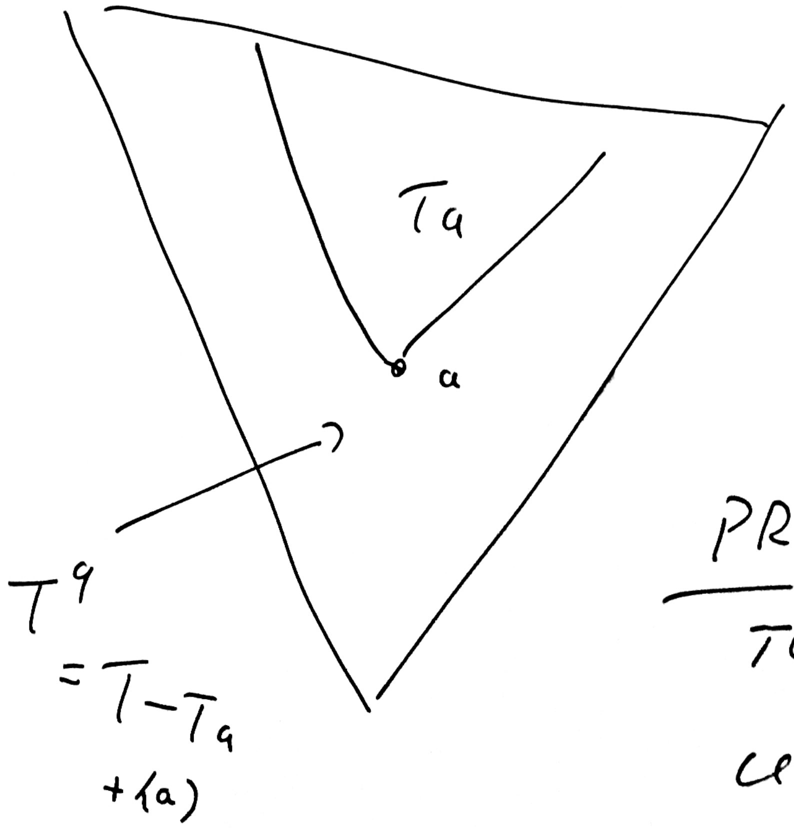
WHEN h_t LEADS TO A LEAVE THERE HAS TO BE
A $C \in \neg onto PHP_n$ FORCED FALSE BY h_t
$\Rightarrow |h_t| \geq n$ AS h_t HAS TO OCCUPY ALL OF $[n]$.

□

WHAT ABOUT SIZE?

SPIRA'S LEMMA: [L.1.1.4] IN ANY BINARY TREE T THERE IS A MID-POINT a S.T.

$$\frac{1}{3}|T| \leq |T^a|, |T_a| \leq 2/3\,|T|. \quad (*)$$

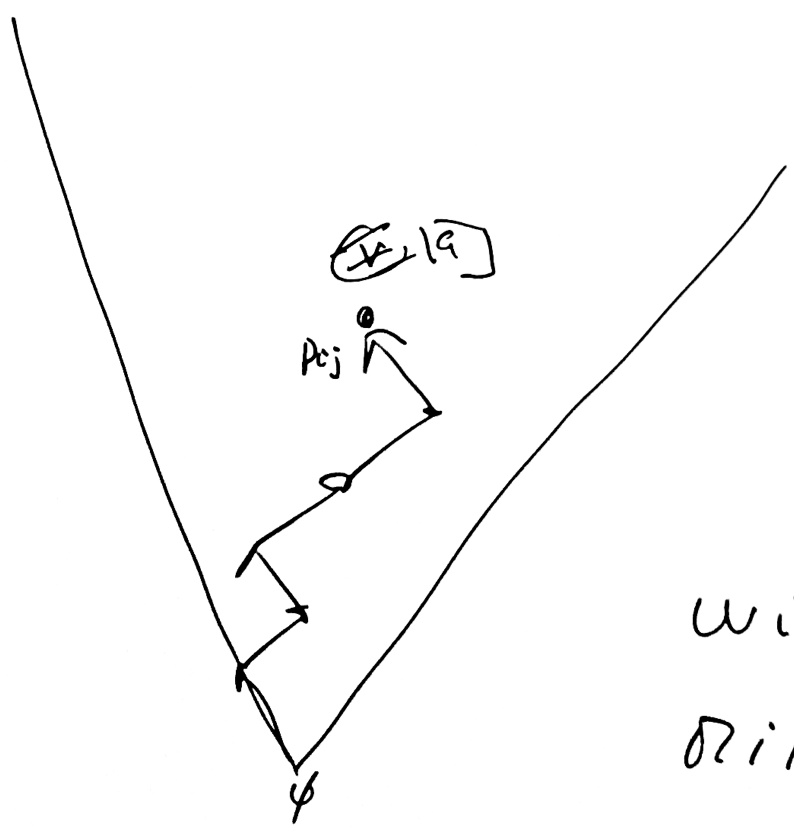


PRF: WALK FROM THE ROOT ALWAYS TOWARDS THE BIGGER SUBTREE UNTIL (*) BECOMES TRUE.

□

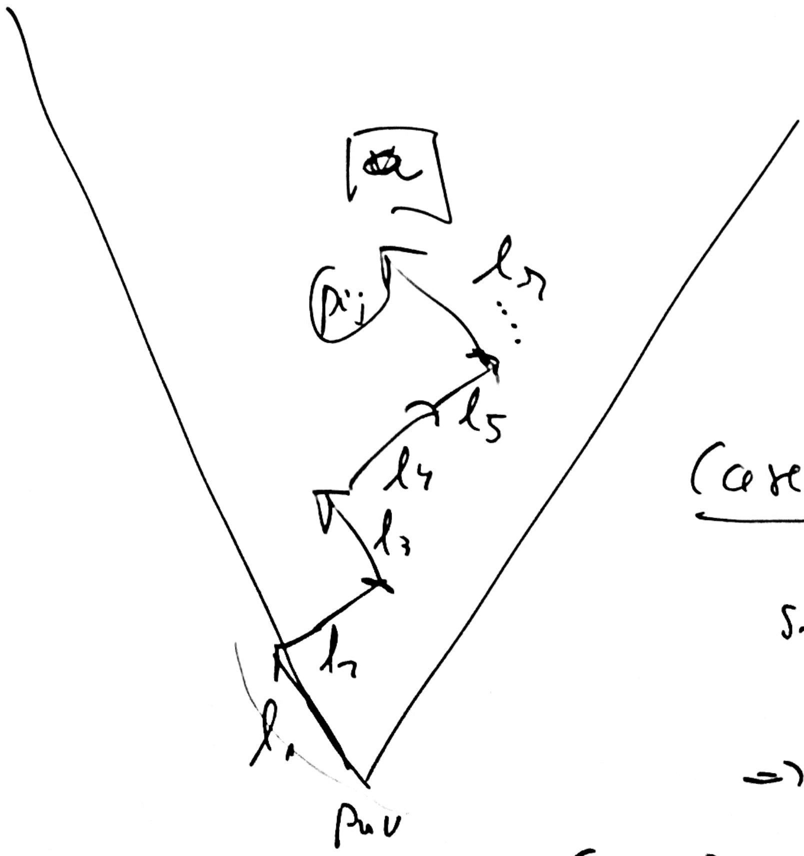
LEMMA: EVERY DEC. TREE T SOLVING SEARCH($\neg$ onto-PHP$_n$) MUST HAVE THE SIZE $\geq \left(\frac{3}{2}\right)^n$.

PRF:



IDEA: INSTEAD OFF BUILDING THE PATH AND THE MAPS h_t **bottom-up** WE WILL WALK ON SPIRA'S MID-POINTS.

STEP t (have h_t)



$l_1 \cdots p_{uv}$, if edge $p_{uv}=1$

$\cdots \neg p_{uv}$, if $\cdots$ $p_{uv}=0$

a : mid-point

ASK: Is $\{\neg l_1, \ldots, \neg l_k\}$ true?

Case 1: h_t CAN BE EXTENDED to h_{t+1}

s.t. $h_{t+1} \Vdash \neg l_j = 1$, some l_j.

$\Rightarrow$ W.L.O.G. $|h_{t+1} \setminus h_t| \leq 1$.

Case 2: NOT, so $h_t \Vdash \neg l_j = 0$, all l_j

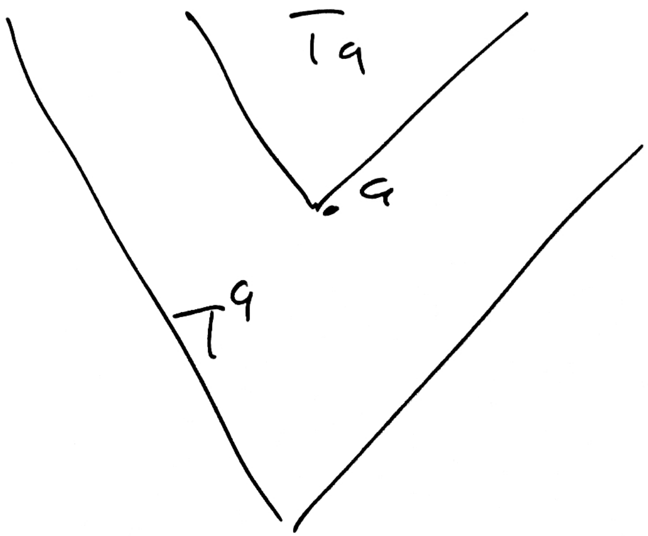
PUT $h_{t+1} := h_t$.

CLAIM : In Case 1 each assignement extending THE ONE DETERMINED By h_{t+1} WILL AVOID a.

In Case 2 it will force the PATH TO a.

⇒ Case 1 : We can restrict to tree T^a.

Case 2 : — " — T_a



⇒ size goes to 2/3.

□ Lemma.

THIS CONSTRUCTION CAN BE INTERPRETED AS:

LEMMA: IF THERE IS A SIZE S DEC. TREE T SOLVING SEARCH($\mathcal{C}$) THEN THERE IS ANOTHER TREE T^* S.T.:

(i) T^* SOLVES SEARCH($\mathcal{C}$) TOO.

(ii) $\mathrm{depth}(T^*) \leq \log_{3/2}(\mathrm{size}(T))$

(iii) T^* BRANCHES UPON QUESTIONS: "Is clause C true?"

$\square$

REMARK (THINK IT THROUGH)

FOR Tout PHP_n WE COULD ALLOW SUCH T^* to ask for TRUTH VALUES OF k-DNFs :

$$D_1 \vee \dots \vee D_\ell$$

$$D_i = l_{i_1} \wedge \dots \wedge l_{i_k}$$

- FORCING D_i true needs to EXTEND h_t BY $\le k$ PAIRS.

$\Rightarrow$ L. BOUND CHANGES FROM $(3/2)^n$ TO $(3/2)^{n/k}$. STILL BIG IF, E.G., $k \sim \log n$ or $\sim \sqrt{n}$.

NEXT 2-3 LECTURES WILL BE DEVOTED TO A PROOF OF SIZE LOWER BOUND FOR THE

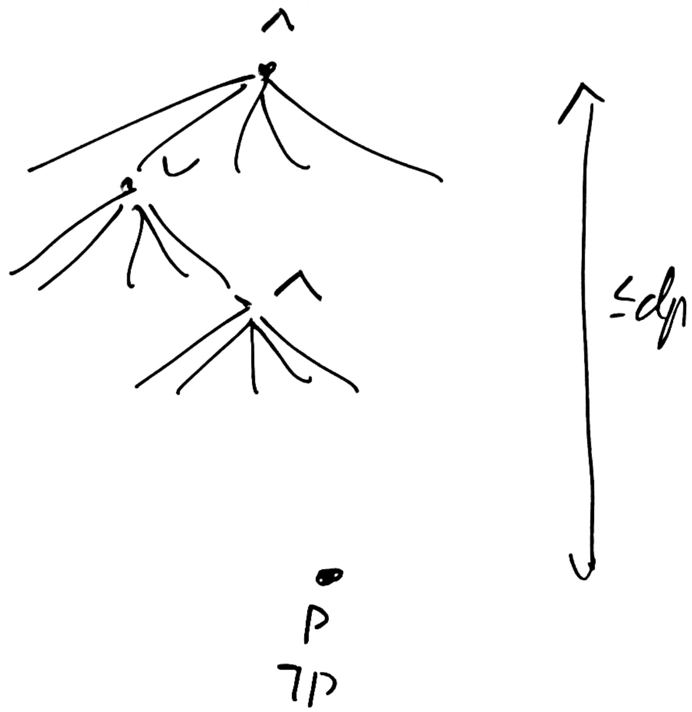
AC^0-Frege system F_d (any fixed $d \geq 1$).

↳ OPERATES WITH FORMULAS OF DEPTH $\leq d$ ONLY

FACT: FOR ANY DeMORGAN LANG. F THERE IS $d \geq 2$ S.T. F_d IS COMPLETE FOR DNFs (AS IS PHP_n).

DEF. OF DEPTH

PICTURE



[Sec. 15.1]

BUT WE HAVE ONLY BINARY

$\vee, \wedge$

LANG. SIMPLIFICATION : $0, 1, \neg, \vee$

$dp(\text{atom}) = dp(0) = dp(1) := 0$

$dp(\neg A) := 1 + dp(A)$

$dp(A \vee B) :=$

	A	B (starting)
$\max(dp A, dp B)$	$\vee$	$\vee$
$\max(1 + dp A, 1 + dp B)$	$\neg$	$\neg$
$\max(1 + dp A, dp B)$	$\neg$	$\vee$
$\max(dp A, 1 + dp B)$	$\vee$	$\neg$

THAT is : $dp(A) = \max$ nb. of ALTERNATIONS OF $\neg/\vee$ WHEN GOING TO SMALLER SUBFLAS.