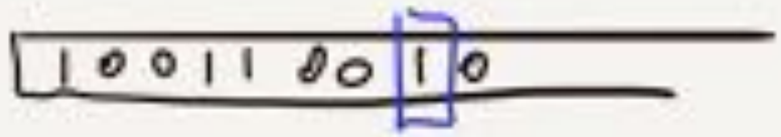
PAST AND PRESENT OF DESCRIPTIVE COMPLEXITY THEORY

Albert Atserias
UPC Barcelona

Workshop on Theor. Found. SAT/SMT Solving
Simons 2021 Spring Programs.

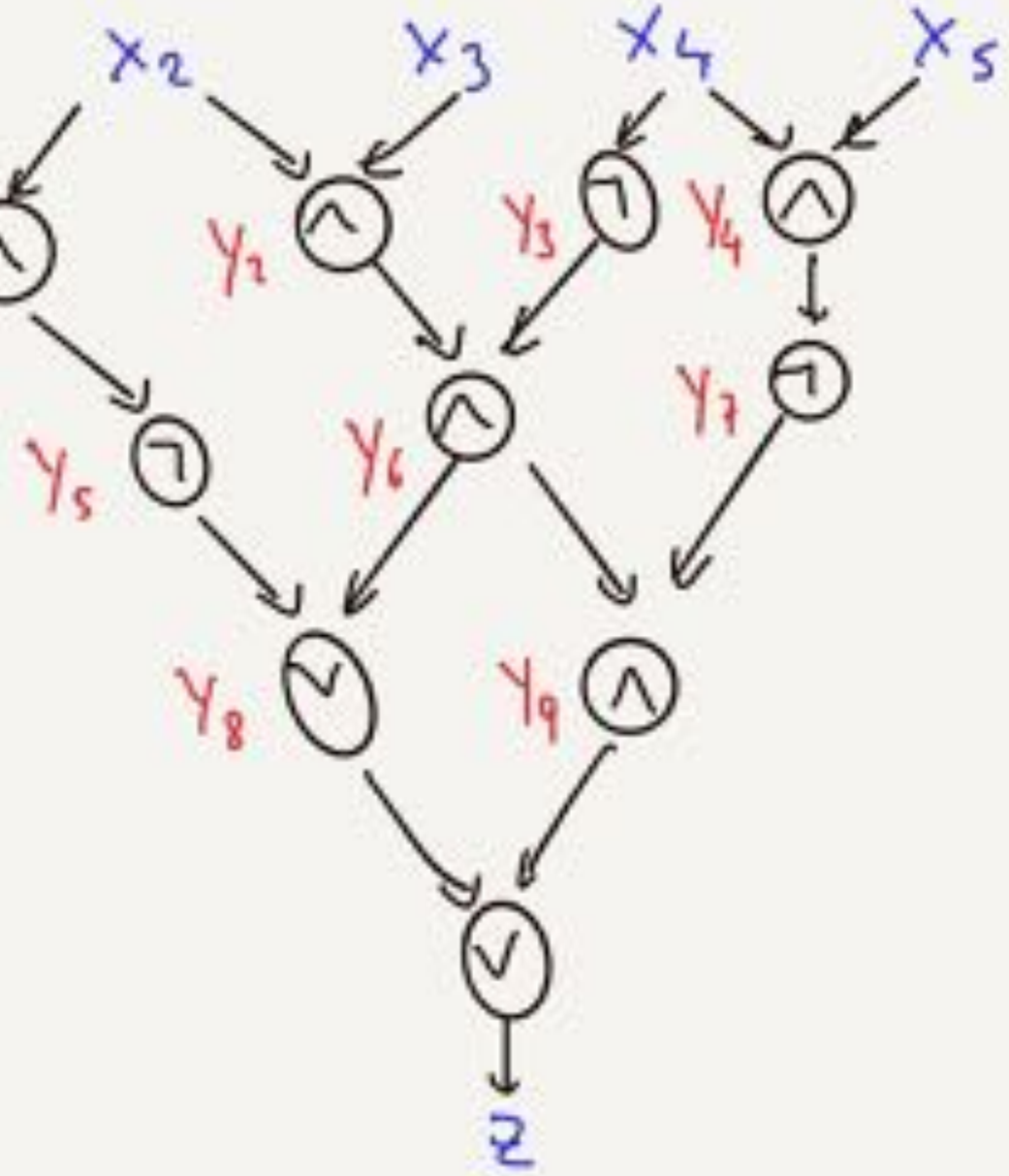
Turing machine [Turing 1936]

tape : 

control :

i: if reading a then
   rewrite by b
   move head left/right
   jump to instruction j

Boolean circuit / Relay & Switching circuit [Shannon 1938]



$y_1 = x_1 \wedge x_2$

$y_2 = x_2 \wedge x_3$

$y_3 = \neg x_4$

$y_4 = x_4 \wedge x_5$

$y_5 = \neg y_1$

$y_6 = y_2 \wedge y_3$

$y_7 = \neg y_4$

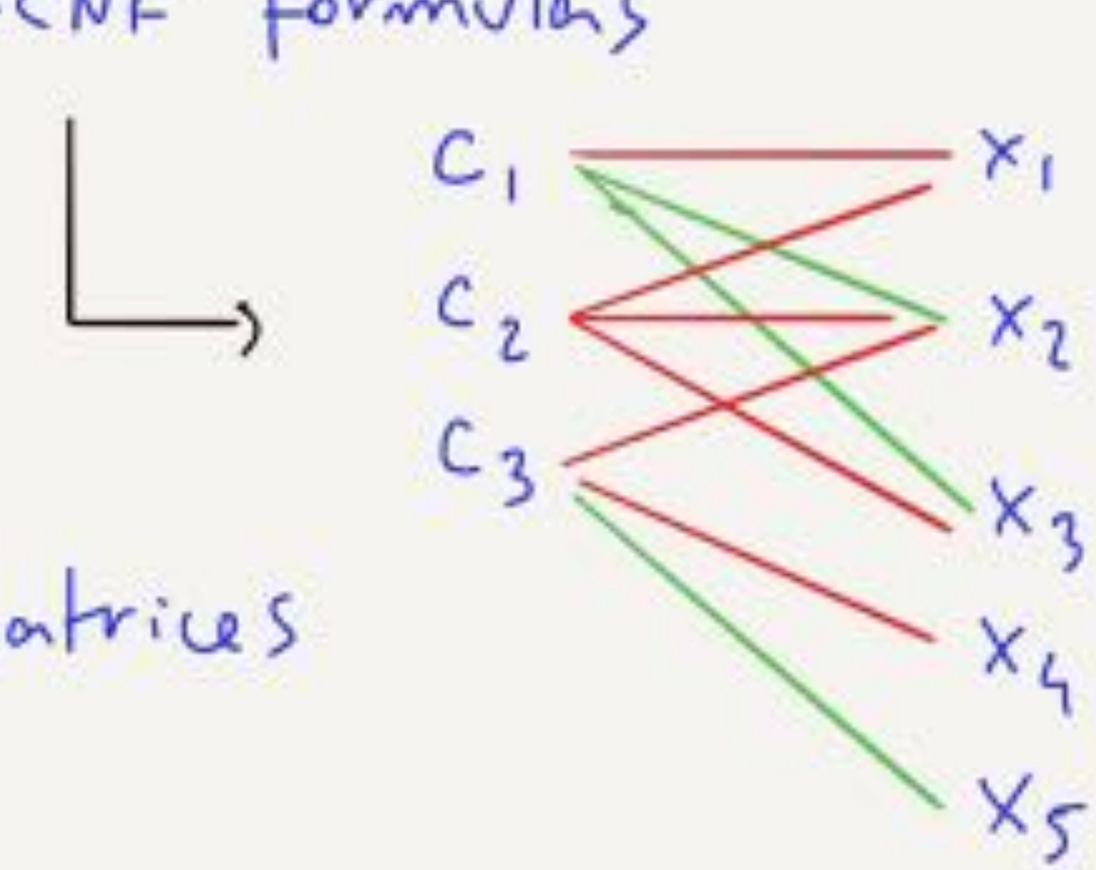
$y_8 = y_5 \vee y_6$

$y_9 = y_6 \wedge y_7$

$z = y_8 \vee y_9$

Structured Data

- graphs, directed, undirected, weighted, ...
- hypergraphs
- trees, partially ordered sets, ...
- k-CNF formulas



- matrices
- ⋮

Relational model of data [Codd 1970]

Relations: $R_1, R_2, \dots, R_m$

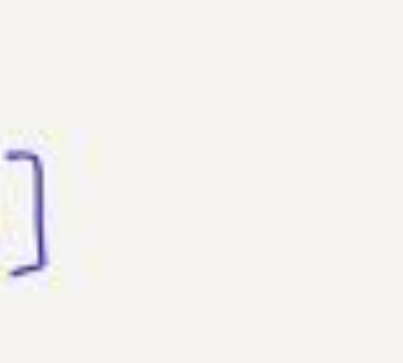
$$R \subseteq D_1 \times D_2 \times \dots \times D_r$$

set of tuples

domains (type of the relation)

Ex:

U	V
a	1
b	1
b	2



E	U	V
	a	1
	b	1
	b	2

Relational algebra [Codd 1972]

Basic operations: $\cap, \cup, -, \times, \sigma, \pi$

$R = S \cap T$ ⟵ intersection

$R = S \cup T$ ⟵ union

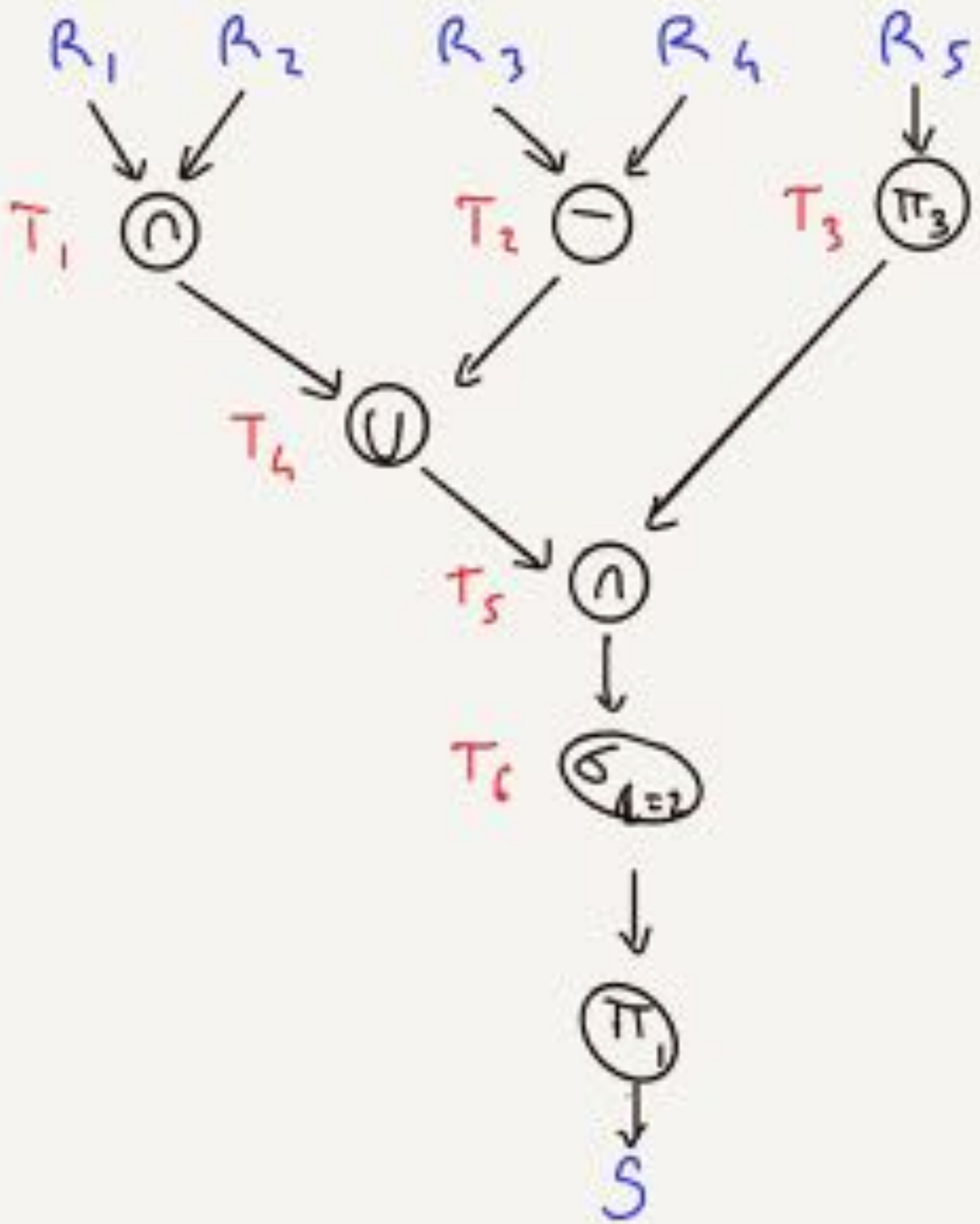
$R = S - T$ ⟵ set difference

$R = S \times T$ ⟵ cartesian product

$R = \sigma_{i=j} T$ ⟵ selection: $\{(t_1, \dots, t_r) \in T : t_i = t_j\}$

$R = \pi_i T$ ⟵ projection: $\{(t_1, \dots t_{i-1}, t_{i+1}, \dots t_n) : (t_1, \dots, t_r) \in T\}$

Relational Algebra Expressions



$T_1 = R_1 \cap R_2$

$T_2 = R_3 - R_4$

$T_3 = \pi_3 R_5$

$T_4 = T_1 \cup T_2$

$T_5 = T_4 \cap T_3$

$T_6 = \sigma_{1=2} T_5$

$S = \pi_1 T_6$

Codd's Theorem

Relational algebra (RA)	≡	First-order logic (FO)
$\cap$	———	$\wedge$
$\cup$	———	$\vee$
$-$	———	$\neg$
$\sigma_{i=j}$	———	$x_i = x_j$
π_i	———	$\exists x_i$

variable-free!
choice-free!

variables that range over the domains.

Uniformity vs. Non-uniformity

A single RA expression / FO formula can serve all inputs of a given type.

Ex: "vertices in a triangle"

$$\phi(x) = \exists y \exists z \, (Exy \wedge Eyz \wedge Ezx)$$

a query (on graphs)

works for all input graphs $G=(V,E)$

Computability with Relations

[Aho - Ullman '1979]

about the model of computation

"The general consensus is that [...] one does not wish to have general Turing machine capabilities"

Principle 1: Order - independance

Principle 2: Isomorphism - invariance

ISO-INVARIANCE

Fagin's Theorem [Fagin 1974]

Iso-invariant
Non-deterministic
Polynomial-time (NP)

≡

Existential
Second-order
Logic (ESO)

a semantic class!

a syntactic class!

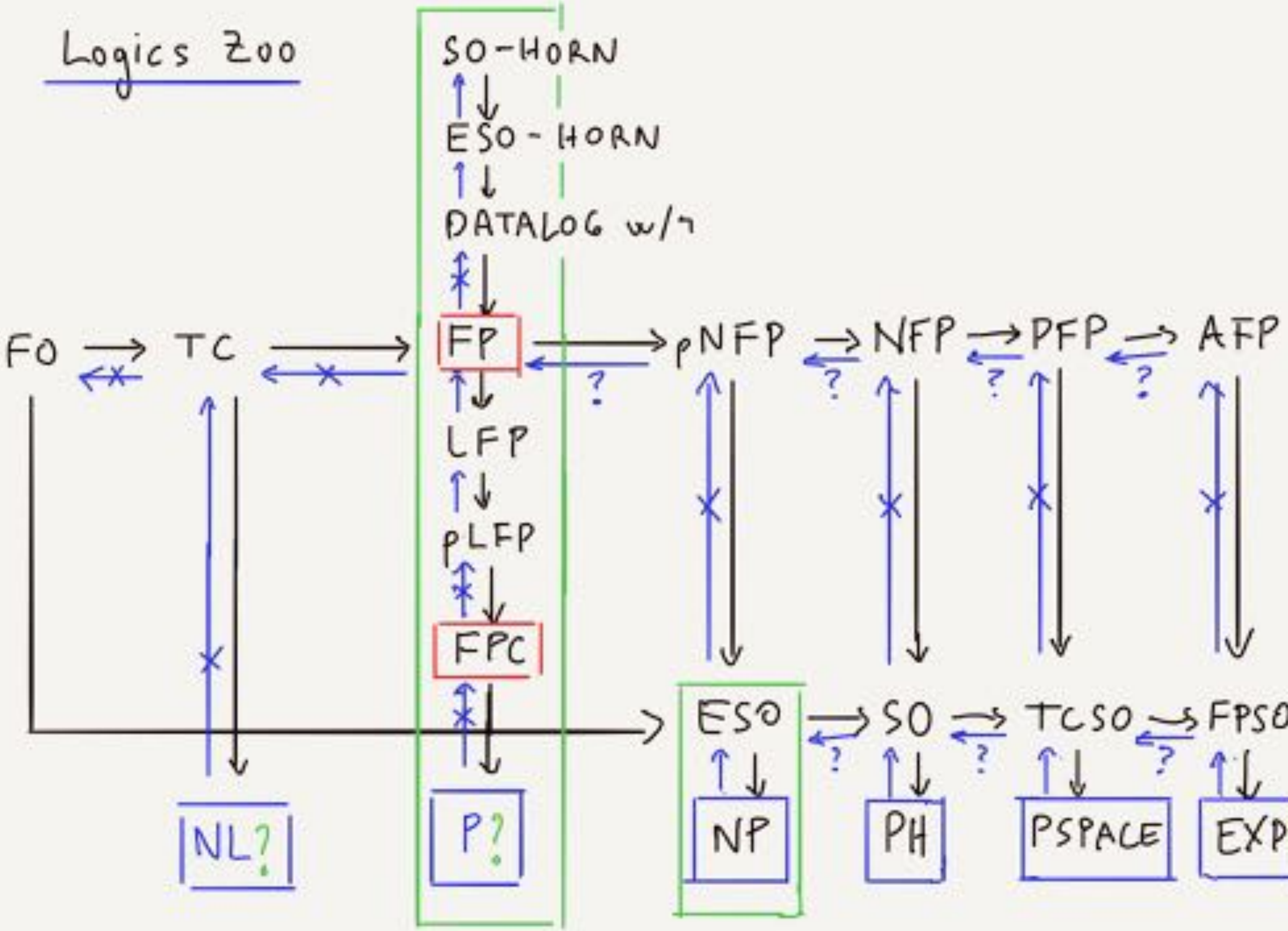
$$\exists \bar{X} \quad \phi(\bar{R}, \bar{X})$$

variables that range over relations!

input relations

FO formula

Logics Zoo


SO-HORN
ESO-HORN
DATALOG w/¬
FO
TC
FP
pNFP
NFP
PFP
AFP
LFP
pLFP
FPC
ESO
SO
TCSO
FPSO
NP
PH
PSPACE
EXP
NL?
P?

FP : Inflationary Fixed-Point Logic

$R_i(\bar{x})$, $x=y$, $\exists x$, $\wedge$, $\vee$, $\neg$, $\boxed{ifp \cdot \psi(X,x)}$

- input
- selection
- projection
- union, inters., neg.
- iteration

```
R = ∅
N = { t : ψ(R,t) }
while (N ≠ R)
    R = N
    N = R ∪ { t : ψ(R,t) }
output R
```

Immerman-Vardi Theorem

IF all domains are ordered and formulas $x < y$ are allowed ⊛

THEN

Deterministic Polynomial-time (P) = Inflationary Fixed-Point Logic (FP)

not a semantic class (by ⊛, iso-invariance trivial)

a syntactic class

Why insist on unordered domains?

1. Fagin's Theorem (NP) doesn't need it.
2. Language for iso-invariant P? [CH'80's]
3. All problems of interest are iso-invariant
4. Choice as a computational resource.
5. Program...

↳ akin to guess bits
random bits
quantum bits

FPC: FP with Counting [Immerman 1986]

FP + counting quantifiers & arithmetic on counters

$\exists^{\geq i} x\,(\psi)$: there exist $\geq i$ many x for which ψ holds.

$\underline{i}$: a variable ranging over a special number domain $[n] = \{0, \cdots, n-1\}$

$i + j = k$
$i \cdot j = k$
] allowed

$\underline{x}$: a variable ranging over one of the domains in input (including $[n]$).

Example: Counting walks in graphs

$A \subseteq V \times V$ ← adjacency matrix of graph

$A \in \{0,1\}^{V \times V}$

$$A^2(u,v) = \sum_{w \in V} A(u,w) \cdot A(w,v)$$

← iso-invariant!

i.e.

$$A^2(u,v) = i \iff \exists^{=i} w \left(A(u,w) \wedge A(w,v) \right)$$

$$A^{2^j}(u,v) = i \iff \left(A^{2^{j-1}}\right)^2 (u,v) = i$$

Expressive power of FPC:

- connectivity queries ← already in FP
- 2-SAT, bounded-width resolution, HORN-SAT, UCP

- perfect matching on bipartite graphs ← [BGS'02]
- matrix non-singularity over finite fields

- determinants and rank of matrices over $\mathbb{Q}$
- solvability of linear systems over $\mathbb{Q}$ ⇐ [H'10]

- linear programming ← [ADH'13]
- perfect matching on general graphs + [AO'18]
- ellipsoid method

Determinants : invariant algorithms

$A \in \mathbb{Q}^{n \times n}$; say 0-1 entries (for simplicity)
$A \subseteq V \times V$; e.g. adjacency matrix

$$\det A = \sum_{\pi \in S_V} (-1)^{ord(\pi)} \prod_{v \in V} A(v, \pi(v))$$

iso-invariant!
invariant under action of S_V (permutations of V)

Is det computable in FPC ?

value in binary as a numeric relation.

Determinants: Invariant Algorithms

Warning: Gaussian elimination NO good

← pivot choice not available.

Alternative : Le Verrier method (in char 0)

$$\det(A - xI) = \sum_{k=0}^{n} c_k \cdot x^k \qquad ; n = |V|$$

$$\begin{cases} c_n = -1 \\ c_{n-j} = \frac{1}{j}\left(c_n \operatorname{tr}(A^j) + c_{n-1}\operatorname{tr}(A^{j-1}) + \dots + c_{n-j+1}\operatorname{tr}(A)\right) \end{cases}$$

FPC !

Linear Programming : Invariant Algorithms

Warnings: Simplex, Karmarkar, Ellipsoid NO good.

Alternative: Iterative refinement of ellipsoid method. [ADH'13]

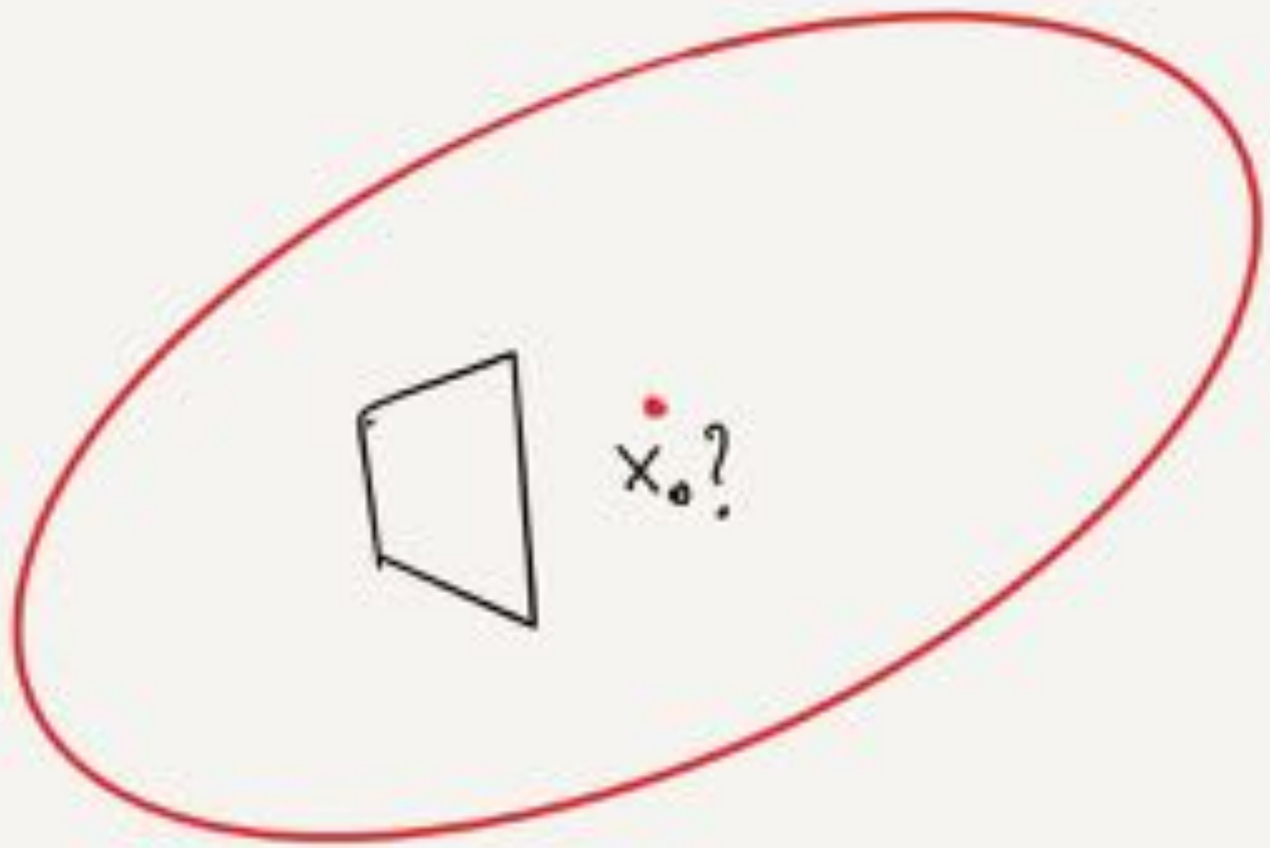
Given

$$Ax \geq b$$

is it feasible?

$$\left[\begin{array}{l} A \subseteq R \times C \times [m] \\ b \subseteq R \times [m] \end{array}\right]$$

bit positions



Linear Programming : Invariant Algorithms

Idea: Solve $\boxed{(A/E_i)(x/E_i) \geq b}$ quotient LP

where:

$$E_0 \geq E_1 \geq E_2 \geq \cdots \geq E_i \geq \cdots \geq Id$$

coarsest equivalence relation on columns of A that respects A (in FPC!)

refinement found by running ellipsoid on $(A/E_0)(x/E_0) \geq b$, and failing.

current refinement.

Each A/E_i has ordered columns:

I.e. IV applies

Directions of future work

PROGRAM

> Prove the Predictions made by Complexity-Theoretic Conjectures, and do so Unconditionally.

[A., Les Houches 2012]

Directions of future work

Ex :

e.g. Unique Games Conjecture

Conjecture X says that problem Y is hard.

Want to show that family Z of algorithms cannot refute conjecture X?

↳ Show that the algorithms in family Z are in FPC.

For hardness of approximation see [AD 2019]

For UGC see J. Tucker-Foltz

Directions of future work

Ex :

Want to prove lower bounds for proof system X?

↳ show that proof-search for X is in FPC.

Upcoming talk by B. Pago

See J. Ochremiak's talk earlier in this series.