Inductive Definability & Finite-Variable Logics: From Logic to Computer Science

Phokion G. Kolaitis

IBM Almaden & UC Santa Cruz

dedicated to

Yiannis N. Moschovakis

Definability circa 1931

"Mathematicians, in general, do not like to deal with the notion definability; their attitude towards this notion is one of distrust and reserve. The reasons for this aversion are quite understandable."

"Without doubt the notion of definability as usually conceived is of a metamathematical origin. I believe that I have found a general method which allows us a rigorous metamathematical definition of this notion."

Definability circca 1931

"… by analyzing this [metamathematical] definition, it proves possible … to replace it by one formulated exclusively in mathematical terms. Under this new definition, the notion of definability does not differ from other mathematical notions and need not arouse either fears or doubts; it can be discussed entirely within the domain of normal mathematical reasoning."

On Definable Sets of Real Numbers

Alfred Tarski, 1931

Definability circa 1980

"Beyond that, what he (the mathematician) needs to read this book is patience and a basic interest in the central problem of descriptive set theory and definability theory in general:

to find and study the characteristic properties of definable objects."

Descriptive Set Theory (About This Book)

Yiannis N. Moschovakis, 1980

Inductive Definability

- **First-order definability:** the study of the relations *explicitly* definable by first-order formulas on a structure.

- **Inductive definability:** the study of the relations *inductively* definable by first-order formulas on a structure.

- **Motivation:** Augment first-order logic with recursive constructs.

- **Example:** Graphs **G** = (V,E)
  - The transitive closure T of E is not first-order definable
  - Recursive specification of transitive closure:

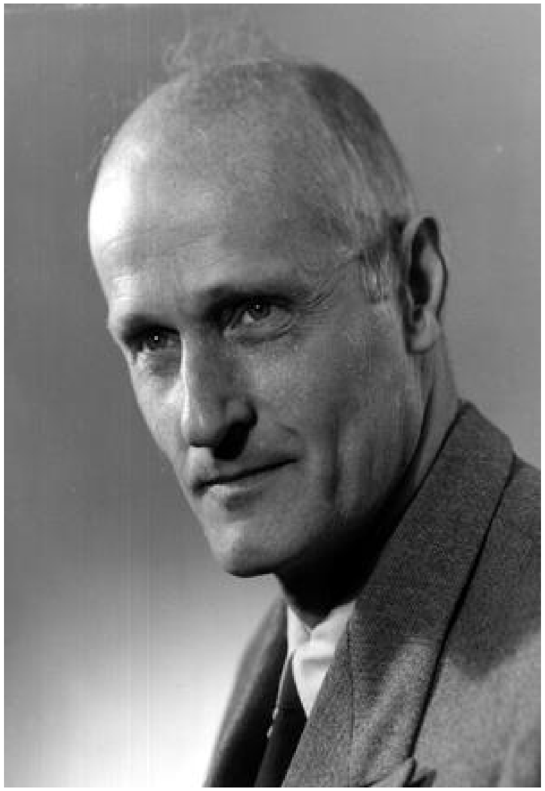
$$T(x,y) \Leftrightarrow (E(x,y) \vee \exists z\, (E(x,z) \wedge T(z,y))$$

Inductive Definability: A Brief History

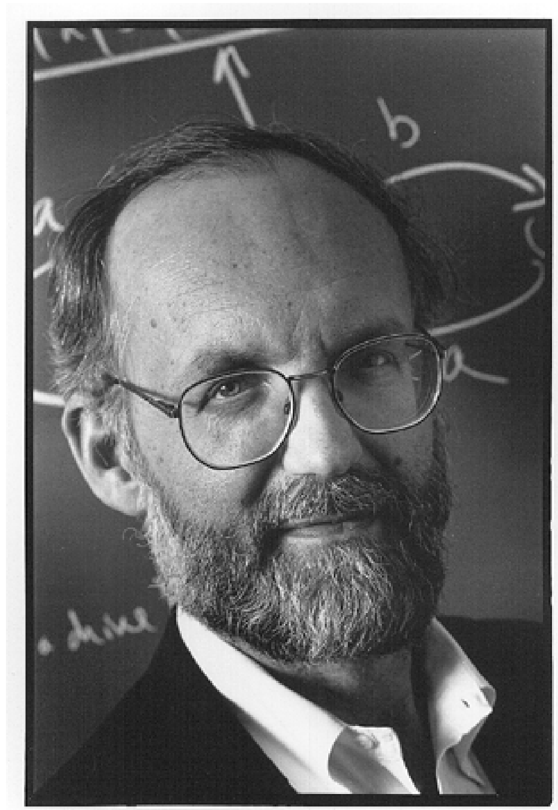
- **Hyperarithmetic Theory:** 1944-1961
  Kleene and Spector
  Study of inductively definable relations on **N** = (N, +, $\times$)

- **Abstract Recursion Theory:** late 1960s onward
  Aczel, Barwise, Gandy, Moschovakis, …
  Study of notions of computability on infinite structures
  (ordinals, admissible sets, …)

- **Inductive Definability on Abstract Structures**
  Y.N. Moschovakis' monograph:
  Elementary Induction on Abstract Structures, 1974



Stephen C. Kleene



K. Jon Barwise

Robin O. Gandy

Least Fixed-Points of First-Order Formulas

- Vocabulary σ, first-order formula $\phi(x_1,\ldots,x_k,T)$ over $\sigma \cup \{T\}$
- On every structure A over σ, it gives rise to an operator
  - Φ: $\mathbf{P}(A^k) \rightarrow \mathbf{P}(A^k)$, where
  - $\Phi(T) = \{(a_1, \ldots,a_k)\colon \mathbf{A} \models \phi(a_1, \ldots,a_k,T)\}$
- **Transfinite iteration** of Φ
  - $\phi^1 = \Phi(\emptyset)$
  - $\phi^\alpha = \Phi(\bigcup_{\beta<\alpha} \phi^\beta)$
- If $\phi(x_1,\ldots,x_k,S)$ is positive in T, then Φ is monotone in T
  - $\phi^1 \subseteq \phi^2 \subseteq \ldots \subseteq \phi^\alpha \subseteq \phi^{\alpha+1} \subseteq \ldots$
- **Tarski-Knaster Theorem**: Φ has a least fixed-point ϕ^∞ (the smallest T such that $T = \Phi(T)$). Moreover,

$$\phi^\infty = \bigcup_\alpha \phi^\alpha$$

Examples

- **Transitive Closure**
  - $\phi(x,y,T) \equiv E(x,y) \vee \exists z\ (E(x,z) \wedge T(z,y))$
  - $\phi^n(x,y) \equiv$ "there is a path of length $\leq$ n from x to y"
  - $\phi^\infty(x,y) \equiv$ "there is a path from x to y"
- **Well-Founded Part**
  - $\psi(x,T) \equiv \forall y\ (E(y,x) \rightarrow T(y))$
  - $\psi^1(x) \equiv$ in-degree(x) = 0
  - $\psi^2(x) \equiv \forall y\ (E(y,x) \rightarrow$ in-degree(y) = 0)
  - $\psi^\infty(x) \equiv$ no infinite descending chain through x
    $E(x,y_1), E(y_1,y_2), \ldots, E(y_n,y_{n+1}), \ldots$

Systems of Positive First-Order Formulas

Systems of Positive First-Order Formulas

- $ODD(x,y) \equiv E(x,y) \lor \exists z\, (E(x,z) \land EVEN(z,y))$
  $EVEN(x,y) \equiv \exists z\, (E(x,z) \land ODD(z,y))$

  Simultaneous Inductive Definitions

- $ODD^{\infty}(x,y) \equiv$ "there is a path of odd length from x to y"
  $EVEN^{\infty}(x,y) \equiv$ "there is a path of even length from x to y"

Least Fixed-Point Logic LFP

- **Definition:**
  - Least Fixed-Point Logic LFP: least fixed-points of systems of positive first-order formulas
  - If **A** = (A, R_1, …, R_m) is a structure, then
    LFP[**A**] = Collection of all LFP-definable relations on **A**
- **Fact:** For every structure **A** = (A, R_1, …, R_m),

$$\mathrm{FO}[\mathbf{A}] \subseteq \mathrm{LFP}[\mathbf{A}] \subseteq \Pi^1_1(\mathbf{A})$$

Least Fixed-Point Logic

- **Theorem (Kleene – Spector):** On **N** = (N, +, $\times$),

  $$\mathrm{LFP}[\mathbf{N}] = \Pi^1_1(\mathbf{N})$$

  Moreover, LFP[**N**] is not closed under complements.

- **Note:** "Constructive" characterization of universal second-order definable relations on **N** = (N, +, $\times$).

- **Theorem (Moschovakis):** If **A** = (A, R_1, …, R_m) is a countable structure with a first-order definable coding apparatus, then

  $$\mathrm{LFP}[\mathbf{A}] = \Pi^1_1(\mathbf{A}).$$

  Moreover, LFP[**A**] is not closed under complements.

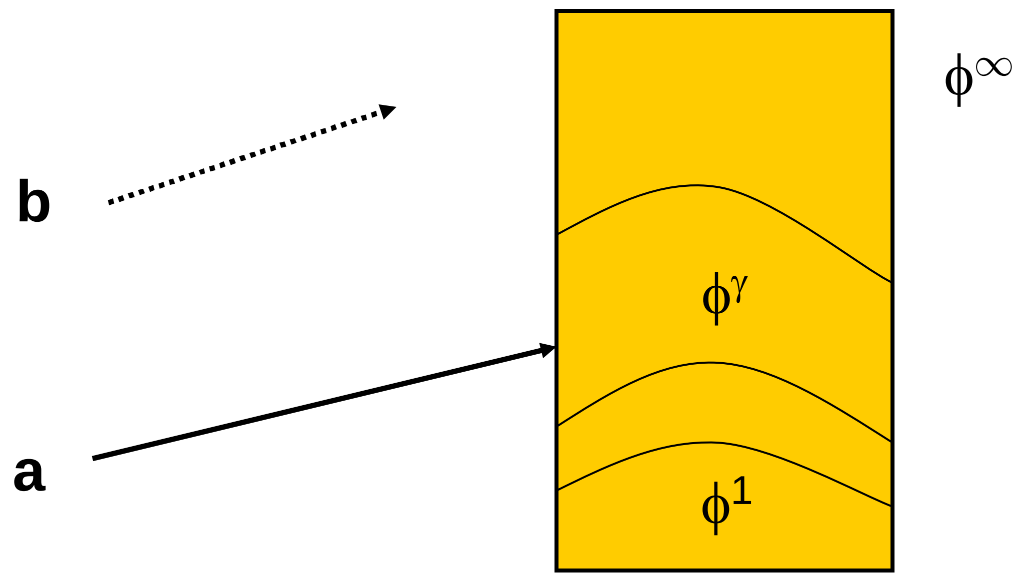
Stage Comparison Relations

- **Definition:** $\phi(\mathbf{x},T)$ positive in T first-order formula
  Stage Comparison Relations on $\mathbf{A} = (A, R_1, \ldots, R_m)$:
  - $\mathbf{a} \prec_\phi \mathbf{b} \Leftrightarrow$ **a** enters ϕ^∞ before **b**
  - $\mathbf{a} \preceq_\phi \mathbf{b} \Leftrightarrow$ **a** enters ϕ^∞ no later than **b**

b need not be in ϕ^∞

$\mathbf{a} \prec_\phi \mathbf{b}$

a must be in ϕ^∞



Stage Comparison Relations

- **Example:** $\phi(x,y,T) \equiv E(x,y) \vee \exists z\,(E(x,z) \wedge T(z,y))$
  - $\prec_\phi$ and $\preceq_\phi$ are the distance comparison queries on E:
  - $(a,c) \preceq_\phi (b,d) \Leftrightarrow \text{distance}(a,c) \leq \text{distance}(b,d)$

- **Stage Comparison Theorem (Moschovakis):**

  For every positive first-order formula $\phi(\mathbf{x},T)$ and every structure $\mathbf{A} = (A, R_1, \ldots, R_m)$, the stage comparison relations $\prec_\phi$ and $\preceq_\phi$ are LFP-definable on **A**.

Finite-Variable Infinitary Logics

- **Definition:** Infinitary Logic $L_{\infty\omega}$

  FO-logic + infinitary disjunctions $\bigvee\Psi$ and $\bigwedge\Psi$.

- **Definition: (Barwise – 1975)**

  - $L^k_{\infty\omega}$ is the collection of all $L_{\infty\omega}$-formulas with at most k distinct variables (variables may be reused), $k \geq 1$.

  - $L^\omega_{\infty\omega} = \bigcup_k L^k_{\infty\omega}$

LFP and Finite-Variable Infinitary Logic

- **Fact:** For every $n \geq 1$, there is a FO^3-formula $\psi^n(x,y)$ expressing the property:
  "there is a path of length at most n from x to y"
  - $\psi^1(x,y) \equiv E(x,y)$
  - $\psi^{n+1}(x,y) \equiv \exists z\, (E(x,z) \wedge \exists x\, (x=z \wedge \psi^n(x,y))$
- **Theorem (Barwise - 1975):**
  - On every structure $\mathbf{A} = (A, R_1, \ldots, R_m)$,
    $$LFP[\mathbf{A}] \subseteq L^{\omega}_{\infty\omega}[\mathbf{A}].$$
  - $L^k_{\infty\omega}$-definability can be analyzed via k-pebble games, i.e., families of partial isomorphisms with back-&-forth properties up to k (also Immerman – 1981).

Local vs. Global Inductive Definability

- **Local Inductive Definability:** In Moschovakis' monograph, the study of inductive definability takes place on an arbitrary, but fixed, infinite structure.

- **Global Inductive Definability:** Results in local inductive definability often hold uniformly for classes of structures (and with the same proof).

- **Sample Result:** The inductive definitions of the stage comparison relations $\prec_\phi$ and $\preceq_\phi$ depend only on the formula ϕ, not on the structure **A**.

Logic and Computer Science

- The study of abstract recursion theory and inductive definability on fixed infinite structures waned in the 1980s.

However,

- During the past 30 years, there has been an extensive and continuous interaction between logic and computer science.
- Global inductive definability and finite-variable logics have featured prominently in this interaction:
  - Computational Complexity
  - Finite Model Theory
  - Relational Database Theory
  - Constraint Satisfaction
  - …

Queries on Finite Structures

- **F**: the class of all finite structures **A** = (A, R_1, …, R_m) over σ
- **C**: a subclass of F closed under isomorphisms

- **Definition: Chandra & Harel – 1980**
  - A k-ary query on **C** is a function Q on **C** such that
    - For every **A** in **C**, we have that Q(**A**) $\subseteq A^k$
    - Q is preserved under isomorphisms:

    If h: **A** $\rightarrow$ **B** is an isomorphism, then Q(**B**) = h(Q(**A**)).

  - A Boolean query on **C** is a function Q: **C** $\rightarrow$ {0,1} that is preserved under isomorphisms.

Examples of Queries on Graphs

- **Transitive Closure**: Is there a path from a to b?
  - **G**=(V,E) $\rightarrow$ T(**G**), where

    T(**G**) = {(a,b): there is a path from a to b}
  - **Transitive Closure** is a binary query

- **3-Colorability:** Is **G** a 3-colorable graph?
  - $$Q(\mathbf{G}) = \begin{cases} 1 & \text{if } \mathbf{G} \text{ is 3-colorable} \\ 0 & \text{if } \mathbf{G} \text{ is not 3-colorable} \end{cases}$$
  - **3-Colorability** is a Boolean query

Global Definability on Finite Structures

- **Definition:** Let L be a logic, **C** a class of finite structures, and Q a query on **C**.

  Q is L-definable on **C** if there is a formula $\phi(x_1, \ldots, x_k)$ of L such that for every structure **A** in **C**,

  $Q(\mathbf{A}) = \{(a_1,\ldots,a_k): \mathbf{A} \models \phi(a_1, \ldots, a_k)\}$

- **Notation:**

  L[**C**] = class of all L-definable queries on **C**

LFP on Finite Structures

Proposition: For every class **C** of finite structures,

$$\mathrm{FO}[\mathbf{C}] \subseteq \mathrm{LFP}[\mathbf{C}] \subseteq \mathrm{PTIME}[\mathbf{C}]$$

Proof: Let $\phi(x_1, \ldots, x_k, T)$ be a positive FO-formula.

For every finite structure $\mathbf{A} = (A, R_1, \ldots, R_m)$, we have that $\mathbf{A} \models \phi^\infty = \phi^s$, for some $s \leq |A|^k$, because

$$\phi^1 \subseteq \phi^2 \subseteq \ldots \subseteq \phi^n \subseteq \ldots \subseteq A^k.$$

Proposition: On the class **F** of all finite graphs,

$$\mathrm{FO}[\mathbf{F}] \subsetneq \mathrm{LFP}[\mathbf{F}] \subsetneq \mathrm{PTIME}[\mathbf{F}]$$

Proof:

- **Transitive Closure** Query $\in \mathrm{LFP}[\mathbf{F}] \setminus \mathrm{FO}[\mathbf{F}]$
- **Even Cardinality** Query $\in \mathrm{PTIME}[\mathbf{F}] \setminus \mathrm{LFP}[\mathbf{F}]$

LFP on Ordered Finite Structures

Theorem: Immerman – Vardi, 1982

If **C** is a class of ordered finite structures **A** = $(A, <, R_1, \ldots, R_m)$, then LFP[**C**] = PTIME[**C**].

Open Problem: Gurevich, 1988

- Is there is a logic for PTIME?
- More precisely, let **F** be the class of all finite structures **A** = $(A, R_1, \ldots, R_m)$. Is there a logic L such that

  L[**F**] = PTIME[**F**]?

LFP on Finite Structures

- **Note:** Recall that LFP(**N**) is not closed under complements.
- **Theorem: Immerman, 1982**

  Let **F** be the class of all finite structures **A** = $(A, R_1, \ldots, R_m)$.

  Then LFP(**F**) is closed under complements

  Hint of Proof: Use the **Stage Comparison Theorem**

  - Show that Max_ϕ is LFP(**F**)-definable, where

    $\mathrm{Max}_\phi(\mathbf{A}) = \{\mathbf{a}: \mathbf{a}$ enters ϕ^∞ at the last stage of $\phi\}$
  - Note that if **A** is finite, then $\mathrm{Max}_\phi(\mathbf{A}) \neq \emptyset$
  - Hence, for every finite **A**,

    $\mathbf{b} \notin \phi^\infty \Leftrightarrow \exists\, \mathbf{a}\, (\mathbf{a} \in \mathrm{Max}_\phi \wedge \mathbf{a} \prec_\phi \mathbf{b})$

Finite-Variable Logics on Finite Structures

- $L_{\infty\omega}$ is uninteresting on classes of finite structures as it can express every query. In contrast,
- $L^{\omega}_{\infty\omega}$ turns out to be interesting and useful.
  $L^{\omega}_{\infty\omega}$ has been extensively studied in finite model theory.
- **Fact:** On the class **F** of all finite structures,
  $$\mathrm{FO}[\mathbf{F}] \subsetneq \mathrm{LFP}[\mathbf{F}] \subsetneq L^{\omega}_{\infty\omega}[\mathbf{F}].$$
- The k-pebble games for $L^{k}_{\infty\omega}$, $k \geq 1$, have been used as a tool to study the expressive power of LFP on classes of finite structures: inexpressibility results for $L^{\omega}_{\infty\omega}$ imply inexpressibility results for LFP.
- Structural results for $L^{\omega}_{\infty\omega}$ yield similar structural results for LFP.

Logic & Asymptotic Probabilities

- **Notation:**
  - Q: Boolean query on the class **F** of all finite structures
  - $\mathbf{F}_n$: Class of finite structures of cardinality n
  - μ_n: Probability measure on $\mathbf{F}_n$, $n \geq 1$
  - $\mu_n(Q)$ = Probability of Q on $\mathbf{F}_n$ with respect to μ_n, $n \geq 1$.
- **Definition:** Asymptotic probability of query Q

  $\mu(Q) = \lim_{n \to \infty} \mu_n(Q)$, provided the limit exists
- **Examples:** For the uniform measure μ on finite graphs **G**:
  - μ(**G** contains a $\triangle$) = 1.
  - μ(**G** is connected) = 1.
  - μ(**G** is 3-colorable) = 0.
  - μ(**G** has even cardinality) does not exist.

0-1 Laws in Finite Model Theory

- **Definition:** L a logic, μ_n a probabilty measure on $\mathbf{F}_n$, $n \geq 1$.
  L has a 0-1 law with respect to μ_n, $n \geq 1$, if
  $$\mu(Q) = 0 \text{ or } \mu(Q) = 1.$$
  for every L-definable query Q on **F**.

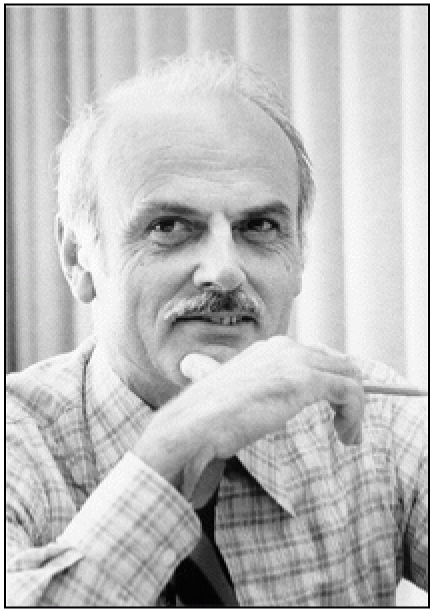
- **Theorem:** With respect to the uniform measure on **F**:
  - FO has a 0-1 law (Glebskii et al.,1969 - Fagin, 1972).
  - LFP has a 0-1 law (Blass, Gurevich, Kozen, 1985)
  - $L^{\omega}_{\infty\omega}$ has a 0-1 law (K .. & Vardi, 1990).

- **Fact:** $L_{\omega_1\omega}$ does not have a 0-1 law.

Relational Databases

E.F. Codd, 1970-1971

- **Relational Database:**
  Collection $(R_1, \ldots, R_m)$ of finite relations
- Relational database $\sim$ Finite structure
  $\mathbf{A} = (A, R_1, \ldots, R_{\mathbf{m}})$
- **Relational Query Languages:**
  - **Relational Algebra**:
    operations $\pi, \sigma, \times, \cup, \setminus$
  - **Relational Calculus:**
    (safe) first-order logic
- **SQL:** The standard commercial database query language based on relational algebra and relational calculus.



E.F. Codd

Datalog

- **Theorem: Aho-Ullman, 1979**

  SQL cannot express the **Transitive Closure** query.

- **Definition**: **Chandra-Harel, 1982**

  A Datalog program is a function-free and negation-free Prolog program.

- **Example:** Datalog program for **Transitive Closure**

  T(x,y) :- E(x,y);

  T(x,y) :- E(x,z), T(z,y).

Datalog and Least Fixed-Point Logic

- **Fact:** For a query Q, the following are equivalent:
  - Q is definable by a Datalog program.
  - Q is definable by a system of existential, entirely positive first-order formulas.
- **Example:**
  - System of existential, entirely positive first-order formulas:

$$ODD(x,y) \equiv E(x,y) \vee \exists z\,(E(x,z) \wedge EVEN(z,y))$$

$$EVEN(x,y) \equiv \exists z\,(E(x,z) \wedge ODD(z,y))$$

  - Datalog program:

```
ODD(x,y)   :-  E(x,y);
ODD(x,y)   :-  E(x,z), EVEN(z,y);
EVEN(x,y)  :-  E(x,z), ODD(z,y).
```

Datalog and Least Fixed-Point Logic

- **Fact:** On the class **F** of all finite structures **A** = $(A, R_1, \ldots, R_m)$,
  Datalog[**F**] $\subsetneq$ LFP[**F**] $\subsetneq$ PTIME[**F**].

- **Theorem:**
  Datalog can express PTIME-complete queries.
  Proof:
  - Datalog can express the **Path Systems** query
    S = (F, A, R), where $A \subseteq F$ and $R \subseteq F^3$.
  - Datalog program for **Path Systems** query:
    T(x) :- A(x);
    T(x) :- R(x,y,z), T(y), T(z).
  - **Cook, 1974: Path Systems** is a PTIME-complete query.

Datalog: Theory and Practice

- 1985-1995: in-depth study of Datalog and its variants.
- Little impact on commercial database systems. However, SQL: 1999 standard supports linear Datalog.
- **Transitive Closure** in SQL:1999

```
with recursive FLY(origin,destination) as
        (select origin, destination
         from NonSTOP
       union
         select NonSTOP.origin, FLY.destination
         from   NonSTOP, FLY
         where NonSTOP.destination = FLY.destination)
select  destination
from    FLY
where origin = 'Athens'
```

Constraint Satisfaction

- **Constraint Satisfaction Problem (CSP):**
  Given a set V of variables, a set D of values, and a set C of constraints, is there an assignment of variables to values such that all constraints in C are satisfied?

- **CSP** is a fundamental and ubiquitous problem in computer science. Special cases of **CSP** include:
  - Boolean Satisfiability
  - Graph Colorability
  - Relational Join Evaluation
  - Scene Recognition in machine vision
  - Belief Revision
  - ...

CSP and the Homomorphism Problem

- **Thesis: Feder & Vardi, 1993**
  CSP can be formalized as the **Homomorphism Problem:**
  Given two finite structures **A** = $(A, R_1, \ldots, R_m)$ and
  B = $(B, P_1, \ldots, P_m)$, is there a homomorphism from **A** to **B**?

- **Definition:** Homomorphism h: **A** $\rightarrow$ **B**
  If $(a_1, \ldots, a_k) \in R_i$, then $(h(a_1), \ldots, h(a_k)) \in P_i$

- **Example:** The following are equivalent for a graph **G**:
  - **G** is 3-colorable
  - There is a homomorphism from **G** to $\triangle$.

Computational Complexity of CSP

- **Fact:** **CSP** is NP-complete.
- **Definition:** **CSP**(**C**,**D**) is the restriction of **CSP** to classes **C** and **D**: Given **A** $\in$ **C** and **B** $\in$ **D**, is there a homomorphism from **A** to **B**?
- **Research Program:**
  - Islands of Tractability of **CSP**:
    For which classes **C** and **D**, is **CSP**(**C**,**D**) in PTIME?
  - Unifying Explanations: Are there any unifying explanations for the tractability of **CSP**(**C**,**D**) for various **C** and **D**?
- **Fact (Feder & Vardi, 1993):** Expressibility in Datalog is a unifying explanation for numerous islands of tractability of **CSP**.

Treewidth

- **Fact:** Many intractable algorithmic problems on arbitrary graphs are solvable in polynomial time on trees.
- **Question:** Can the concept of tree be relaxed to a "*tree-like*" concept, while maintaining good algorithmic behavior?
- **Answer (Robertson and Seymour):** Bounded Treewidth
- **"Definition":** The treewidth of a graph **G**, denoted tw(**G**), is a positive integer that measures how much "*tree-like*" **G** is.
- **Examples:**
  - tw(**T**) = 1, for every tree **T**
  - tw(**C**) = 2, for every cycle **C**.
  - tw($\mathbf{K}_k$) = k-1, where K_k is the complete graph with k nodes

Bounded Treewidth and CSP

- **Definition:** $\mathbf{T}(k)$ = Class of finite structures **B** with tw(**B**) < k.

- **Theorem: Dechter & Pearl, 1989**
  CSP($\mathbf{T}$(k), **F**) is in PTIME, for each k.

- **Theorem: Dalmau, K .., Vardi, 2002**
  - ¬ **CSP**($\mathbf{T}$(k), {**B**}) is definable in k-Datalog, for each k and **B**
  - **CSP**($\mathbf{T}$(k), **F**) is LFP-definable, for every k.
  - Different polynomial-time algorithm for **CSP**($\mathbf{T}$(k), **F**): determine who wins the existential k-pebble game.

Finite-Variable Logics

- **Definition:** If **A** is a finite structure, then $Q^{\mathbf{A}}$ is an existential positive first-order sentence describing the positive atomic diagram of **A**.

- **Example:** If $\mathbf{C}_4$ is the 4-cycle, then $Q^{\mathbf{C}_4}$ is

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \big(E(x_1,x_2) \wedge E(x_2,x_3) \wedge E(x_3,x_4) \wedge E(x_4,x_1)\big)$$

- **Definition:** L^k is the class of all first-order variables with at most k distinct variables built from atomic formulas, $\wedge$, and $\exists$.

- **Example:** $Q^{\mathbf{C}_4}$ is logically equivalent to the L^3-sentence:

$$\exists x_1 \exists x_2 \exists x_3 \big(E(x_1,x_2) \wedge E(x_2, x_3) \wedge \exists x_2 \, (E(x_3,x_2) \wedge E(x_2, x_1))\big)$$

Treewidth and Finite-Variable Logics

- **Theorem: Dalmau, K .., Vardi, 2002**

  For every k$\geq$ 2 and every finite structure **A**, the following are equivalent
  - $Q^{\mathbf{A}}$ is logically equivalent to some L^k-sentence.
  - **A** is homomorphically equivalent to a structure **B** in $\mathbf{T}$(k).
  - core(**A**) $\in$ $\mathbf{T}$(k).

- **Conclusion:** The combinatorial concept of treewidth can be characterized in terms of definability in finite-variable logics.

Synopsis

- Inductive definability and finite-variable logics were originally studied on infinite structures.
- Inductive definability and finite-variable logics turned out to have numerous uses in several different areas in the interface between logic and computer science, including:
  - computational complexity
  - database theory
  - finite model theory
  - constraint satisfaction.

Yiannis N. Moschovakis as Advisor

- Each time I was stuck on a problem:

  "You go home now and think about it."

- When I was attempting naïve approaches:

  "You ***cannot*** solve a hard problem by reformulating it."

- When I was about to defend my Ph.D. thesis and was whining that the results were rather trivial:

  "This happens to everyone. Ten years from now, you will look back and say: how smart was I then!"

  He was, of course, quite right.