Introduction to Descriptive Complexity

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 600

1. Complementation and Transitive Closure
   - Normal Form Theorem for FO(LFP)
   - Transitive Closure Operators
   - Normal Form for FO(TC)
   - Logspace is Primitive Recursive
   - NSPACE$[s(n)]$ = co-NSPACE$[s(n)]$
   - Restrictions of SO

Subsection 1

Normal Form Theorem for FO(LFP)

Fixed-Point Hierarchy

- The language FO(LFP) suggests a potential hierarchy of queries.
- At the bottom, we have FO.
- Then LFP[FO], denoting the single application of the least fixed point operator to a first-order formula.
- Next we may apply quantifiers and boolean operations.
- Then another application of fixed point.

  ⋮

Fixed-Point Hierarchy Over Infinite Structures

- Consider the infinite structure consisting of the natural numbers with its ordering relation.
- The first fixed-point level is equal to the set of second-order, universal queries.

Fact

Over the infinite structure $(\mathbb{N}, \leq)$,

$$\text{LFP}[\text{FO}] = \text{SO}\forall.$$

- It follows that over infinite structures, the fixed-point hierarchy is infinite.
- On this basis, Chandra and Harel conjectured that the fixed-point hierarchy for finite structures would also be infinite.

Fixed-Point Hierarchy: Finite Structures

- We have already seen that for finite ordered structures, the fixed-point hierarchy collapses to its first fixed-point level, LFP[FO].
- As we show in this section, this remains true for unordered structures as well, assuming that there are at least two constants
- Without this assumption, the normal form would not be as nice because it would require a quantification outside LFP.
- Finite structures are fundamentally different from infinite structures in this case.
- The crucial difference is that for finite structures, every least fixed point is completed at a fixed stage.
- We are able to detect this completion.
- We then define the negation of a fixed point as the set of tuples that enter only after this last stage, that is, that never enter the fixed point.

The Easy Cases

Lemma

For any class of finite structures, LFP[FO] is closed under quantification, conjunction, disjunction and applications of LFP.

- We show, by induction, that any formula in FO(LFP) that has no negations of fixed points can be written in the form

$$(\mathrm{LFP}\varphi)(\overline{0}),$$

where φ is first-order.

There are four cases to be considered.

The Easy Cases (Case 1)

1. Suppose

$$\Phi_1 \equiv (\mathrm{LFP}_{S^k x_1 \ldots x_k} \varphi)(u_1, \ldots, u_k).$$

Define,

$$\gamma_1(\overline{x}, b) \equiv (b = 0 \wedge R(\overline{u}, 1)) \vee (b = 1 \wedge \varphi(R(*, 1)/S)).$$

Here and below we write "$\varphi(R(*, \overline{v})/S)$" to mean the formula φ with all occurrences of $S(u_1, \ldots, u_k)$ replaced by $R(u_1, \ldots, u_k, \overline{0}, \overline{v})$.

The tuple $\overline{0}$ consists of dummy values in case the arity of R is greater than k plus the arity of $\overline{v}$.

Thus, we have

$$\Phi_1 \equiv (\mathrm{LPF}_{R^{k+1} x_1 \ldots x_k b} \gamma_1)(\overline{0}).$$

The Easy Cases (Case 2)

2. Suppose

$$\Phi_2 \equiv (\mathrm{LFP}_{S^k x_1,\ldots,x_k} \varphi)(\overline{0}) \text{ op } (\mathrm{LFP}_{T^r x_1\ldots x_r} \psi)(\overline{0}),$$

where op $\in \{\wedge, \vee\}$.
Set $a = \max(k, r)$.
Define

$$\begin{array}{rcl} \gamma_2(x_1,\ldots,x_a,b) & \equiv & b = 0 \wedge (R(\overline{0},1) \text{ op } R(\overline{0},2)) \vee \\ & & b = 1 \wedge \varphi(R(*,1)/S) \vee \\ & & b = 2 \wedge \psi(R(*,2)/S). \end{array}$$

γ_2 does the following:

- It simultaneously computes the two fixed points ($b = 1, 2$);
- It applies "op" to them ($b = 0$).

Thus,

$$\Phi_2 \equiv (\mathrm{LFP}_{R^{a+1} x_1\ldots x_a b} \gamma_2)(\overline{0}).$$

The Easy Cases (Case 3)

3. Suppose
$$(Qz)\Phi_3,$$
where $\Phi_3 \equiv (\text{LFP}_{S^k x_1 \ldots x_k} \varphi)(z, \overline{0})$ and $Q \in \{\exists, \forall\}$.

Let
$$\begin{array}{rcl}\gamma_3(x_1, \ldots, x_k, b) & \equiv & b = 0 \wedge (Qz)(R(z, \overline{0}, 1)) \vee \\ & & b = 1 \wedge \varphi(R(*, 1)/S).\end{array}$$

Here, γ_3 does the following:

- It computes the fixed point in parallel for all values of z ($b = 1$);
- Then quantifies the fixed point ($b = 0$).

Thus,
$$\Phi_3 \equiv (\text{LFP}_{R^{k+1} x_1 \ldots x_k b} \gamma_3)(\overline{0}).$$

The Easy Cases (Case 4)

4. Suppose

$$\Phi_4 \equiv (\text{LFP}_{S^k x_1 \dots x_k} \varphi(S, \overline{x}, (\text{LFP}_{T^r y_1 \dots y_r} \psi(S, \overline{y}, T)(\overline{0})))).$$

Let

$$\begin{aligned} \gamma_4(\overline{x}, \overline{y}, b) \quad \equiv \quad & b = 0 \wedge \varphi(R(*, \overline{0}, 0), \overline{x}, R(\overline{0}, *, 1)) \vee \\ & b = 1 \wedge \psi(R(*, \overline{0}, 0), \overline{y}, R(\overline{0}, *, 1)). \end{aligned}$$

In the formal definition of Φ_4, we would:

- First, compute the fixed point T_1 of ψ, with $S = \varnothing$;
- Then, compute a step of φ with $T = T_1$, getting S_1;
- Next, compute the fixed point T_2 of ψ with $S = S_1$;
- ⋮

In the fixed point of γ_4, we concurrently build the least fixed points of φ ($b = 0$) and ψ ($b = 1$).

The Easy Cases (Case 4 Cont'd)

- Note that everything is monotone.

  So, even though the fixed point of γ_4 may be reached earlier, nothing that should not will enter the fixed point.

  More explicitly, it can be proved inductively that, if a tuple $\overline{z}$ is an element of S_t, then it enters the relation $R(*, \overline{0}, 0)$ no later than round tn^r.

  Conversely, if a tuple $\overline{z}$ enters $R(*, \overline{0}, 0)$ at round t', then it is also a member of $S_{t'}$.

  Thus we have

$$\Phi_4 \equiv (\mathrm{LFP}_{R^{k+r+1}x_1 \ldots x_k y_1 \ldots y_r b} \gamma_4)(\overline{0}).$$

Introducing the Stage Comparison Theorem

- The main difference between taking fixed points over finite, rather than infinite, structures is that, for finite structures, the fixed point finishes after a finite number of stages.
- Moschovakis showed that the stages of a fixed point can be identified as the fixed point is being computed.
- Using his Stage Comparison Theorem, we will be able to find a tuple $\overline{m}$ that enters the fixed point at its last state.
- We will then be able to express the fact that a tuple $\overline{t}$ never enters the fixed point, by saying that the stage at which $\overline{t}$ enters is greater than the stage at which $\overline{m}$ enters.

The Rank of a Tuple

- Let $\varphi(x_1 \ldots x_r, R)$ be an R-monotone formula.
- Let $\mathcal{A}$ be a finite structure.
- Each tuple $(a_1 \ldots a_r) \in I_\varphi$ comes in at some stage of the induction.
- Let $|\overline{a}|_\varphi$, the **rank of $\overline{a}$ with respect to φ**, be the step at which $\overline{a}$ enters I_φ,

$$|\overline{a}|_\varphi = \begin{cases} r, & \text{if } \overline{a} \in I_\varphi^r - I_\varphi^{r-1} \\ \infty, & \text{if } \overline{a} \notin I_\varphi \end{cases}$$

- Define the relation $\leq_\varphi$ by setting, for all tuples $\overline{x}$, $\overline{y}$,

$$\overline{x} \leq_\varphi \overline{y} \quad \text{iff} \quad \overline{x} \in I_\varphi \text{ and } |\overline{x}|_\varphi \leq |\overline{y}|_\varphi.$$

- Similarly, $\overline{x} <_\varphi \overline{y}$ means $\overline{x} \in I_\varphi$ and $|\overline{x}|_\varphi < |\overline{y}|_\varphi$.

The Stage Comparison Theorem

Theorem (Stage Comparison Theorem)

Given a monotone formula $\varphi(\overline{x}, R)$, relations $\leq_\varphi$ and $<_\varphi$ can be written as positive fixed points.

- We view $\varphi(\overline{x}, R)$ as $\varphi'(\overline{x}, R, \neg R)$ which is positive in R and $\neg R$.

  We define relations $\leq_\varphi$ and $<_\varphi$ by simultaneous positive inductions.

  We use the abbreviations

$$\begin{array}{rcl} L_{\overline{z}} & = & \{\overline{u} : \overline{u} \leq_\varphi \overline{z}\}; \\ G_{\overline{z}} & = & \{\overline{w} : \overline{z} <_\varphi \overline{w}\}. \end{array}$$

  We will see that, for every $\overline{z}$ already in the fixed point, $L_{\overline{z}}$ and $G_{\overline{z}}$ are complements.

The Stage Comparison Theorem (Cont'd)

- The following are the positive inductive definitions of $\leq_\varphi$ and $<_\varphi$:

$$\begin{array}{rcl} \overline{x} \leq_\varphi \overline{y} & \equiv & \varphi'(\overline{x}, \textbf{false}, \textbf{true}) \vee \\ & & (\exists \overline{z})(\overline{z} <_\varphi \overline{y} \wedge \varphi'(\overline{x}, L_{\overline{z}}, G_{\overline{z}})); \\ \overline{x} <_\varphi \overline{y} & \equiv & (\varphi'(\overline{x}, \textbf{false}, \textbf{true}) \wedge \neg\varphi'(\overline{y}, \textbf{false}, \textbf{true})) \vee \\ & & (\exists \overline{z})(\overline{z} <_\varphi \overline{y} \wedge \varphi'(\overline{x}, L_{\overline{z}}, G_{\overline{z}}) \wedge \neg\varphi'(\overline{y}, \neg G_{\overline{z}}, \neg L_{\overline{z}})). \end{array}$$

Let α and β be the positive first-order formulas in the above definitions of $\leq_\varphi$ and $<_\varphi$.

For $\langle \overline{z}, \overline{z} \rangle \in I_\alpha^r$, let

$$\begin{array}{rcl} L_{\overline{z}}^r & = & \{\overline{u} : \langle \overline{u}, \overline{z} \rangle \in I_\alpha^r\}; \\ G_{\overline{z}}^r & = & \{\overline{w} : \langle \overline{z}, \overline{w} \rangle \in I_\beta^r\}. \end{array}$$

The Stage Comparison Theorem (Cont'd)

- We can show, by induction, that, for all r,

$$\begin{array}{rcl} I_\alpha^r & = & \{\langle \overline{x}, \overline{y} \rangle : |\overline{x}|_\varphi \leq r \text{ and } |\overline{x}|_\varphi \leq |\overline{y}|_\varphi\}; \\ I_\beta^r & = & \{\langle \overline{x}, \overline{y} \rangle : |\overline{x}|_\varphi \leq r \text{ and } |\overline{x}|_\varphi < |\overline{y}|_\varphi\}; \\ L_{\overline{z}}^r & = & \overline{G_{\overline{z}}^r}, \text{ for } \langle \overline{z}, \overline{z} \rangle \in I_\alpha^r. \end{array}$$

  By a previous lemma, we know how to use simultaneous induction given a pair of positive formulas in two relational variables.

  So can combine the definitions of $\leq_\varphi$ and $<_\varphi$ into a single definition.

  This proves the theorem.

The Non-Positive Case

- A corollary of the theorem is that a monotone, but not necessarily positive, inductive definition may be rewritten as a positive one,

$$(\mathrm{LFP}\varphi)(\overline{a}) \equiv (\mathrm{LFP}\alpha)(\overline{a},\overline{a}).$$

Corollary

Let $\varphi(R,\overline{x})$ be monotone, but not necessarily positive in R. Then the least fixed point of φ is expressible as the least fixed point of a positive formula.

Negation of a Fixed Point

- In order to negate fixed points we express the fact that some tuple $\overline{m}$ has maximum possible rank.
- The following formula says that $\overline{m}$ has maximum rank.
- It captures this property by saying that:
  - It is in the fixed point;
  - No tuple enters the fixed point exactly one step after $\overline{m}$.
- So we have

$$\mathrm{MAX}(\overline{m}) \equiv \overline{m} \leq_\varphi \overline{m} \wedge (\forall \overline{x})(\overline{x} \leq_\varphi \overline{m} \vee \neg \varphi'(\overline{x}, \neg G_{\overline{m}}, \neg L_{\overline{m}})).$$

Negation of a Fixed Point (Cont'd)

- We defined

$$\mathrm{MAX}(\overline{m}) \equiv \overline{m} \leq_\varphi \overline{m} \wedge (\forall \overline{x})(\overline{x} \leq_\varphi \overline{m} \vee \neg\varphi'(\overline{x}, \neg G_{\overline{m}}, \neg L_{\overline{m}})).$$

- It now follows that, for any monotone φ, we can express the negation of the fixed point of φ as a positive least fixed point,

$$\neg(\mathrm{LFP}_{R,x_1\ldots x_k}\varphi)(\overline{a}) \equiv (\exists \overline{m})(\mathrm{MAX}(\overline{m}) \wedge \overline{m} <_\varphi \overline{a}).$$

- Combining this with the preceding theorem and the preceding lemma, we get

Theorem

For any class of finite structures, the fixed point hierarchy collapses at its first fixed point level. In symbols,

$$\mathrm{FO(LFP)} = \mathrm{LFP[FO]}.$$

Inflationary Fixed Point Operator

- Let R be a new k-ary relation symbol that occurs not necessarily monotonically in $\varphi(R, x_1, \ldots, x_k)$.
- Define the **inflationary fixed point operator** IFP by

$$\text{IFP}(\varphi(R, \overline{x})) \equiv \text{LFP}(\varphi(R, \overline{x}) \vee R(\overline{x})).$$

- IFP may be applied to any inductive definition - there is no syntactic restriction.
- If φ is monotone, then $\text{IFP}(\varphi) = \text{LFP}(\varphi)$.

Expressibility of Stage Comparison Formulas

- Let
$$\psi(R,\overline{x}) = \varphi(R,\overline{x}) \vee R(\overline{x}).$$
- Whether or not φ is monotone, the following sequence is monotonically increasing and its union is $\text{IFP}(\varphi)$,
$$\varnothing \subseteq \psi(\varnothing) \subseteq \psi^2(\varnothing) \subseteq \psi^3(\varnothing) \subseteq \cdots.$$
- Even though ψ may not be monotone, the monotonicity of the sequence suffices for the proof of the preceding theorem to go through.
- Thus, the stage comparison formulas $\leq_\psi$ and $<_\psi$ are expressible as least fixed points of positive formulas.

Expressive Power of Inflationary Fixed Point Operator

- An immediate corollary of the preceding observations is that FO(LFP) and FO(IFP) have the same expressive power.
- Note that, when using IFP, we do not have to worry about keeping our definitions positive.
- So IFP is usually more convenient than LFP.

Corollary

We have

$$\mathrm{FO}(\mathrm{IFP}) = \mathrm{FO}(\mathrm{LFP}).$$

Subsection 2

Transitive Closure Operators

Transitive Closure Operators

- Let

$$\varphi(x_1, \ldots, x_k, x_1', \ldots, x_k')$$

be a formula of some vocabulary τ with $2k$ free variables.
- The formula φ describes a query I_φ from STRUCT$[\tau]$ to graphs.
- For a structure $\mathcal{A} \in \text{STRUCT}[\tau]$,

$$I_\varphi(\mathcal{A}) = \langle |\mathcal{A}|^k, E \rangle,$$

where

$$E = \{(a_1, \ldots, a_k, a_1', \ldots, a_k') : \mathcal{A} \models \varphi(\overline{a}, \overline{a'})\}.$$

Transitive Closure Operators (Cont'd)

- We write
$$(\mathrm{TC}_{x_1\dots x_k x'_1\dots x'_k}\varphi)$$
to denote the reflexive, transitive closure of binary relation $\varphi(\overline{x},\overline{x}')$.
- We denote by
$$\mathrm{FO}(\mathrm{TC})$$
the closure of first-order logic with arbitrary occurrences of TC.
- We know from a previous proposition and previous theorem that,
$$\mathrm{FO}(\mathrm{TC}) \subseteq \mathrm{FO}[\log n] \subseteq \mathrm{FO}[n^{O(1)}] = \mathrm{FO}(\mathrm{LFP}).$$
- Let FO(pos TC) be the restriction of FO(TC) in which TC never occurs within a negation.

FO(pos TC) = NL

Theorem

FO(pos TC) = NL.

($\subseteq$) With space $\log n$ we can cycle through all the values of x.

So the set of relations computable in NSPACE[$\log n$] is closed under first-order quantifiers, $(\forall x)$ and $(\exists x)$.

Thus, it suffices to show that if $\varphi(\overline{x}, \overline{x}')$ is computable in NSPACE[$\log n$], then so is $(\mathrm{TC}_{\overline{x}\overline{x}'}\varphi)$.

We can test if structure $\mathcal{A}$ satisfies $(\mathrm{TC}_{\overline{x}\overline{x}'}\varphi)(\overline{a}, \overline{a}')$ as follows.

If $\overline{a} = \overline{a}'$, then accept.

Else, guess $\overline{b}$ and check that $\mathcal{A} \models \varphi(\overline{a}, \overline{b})$.

Next, throw away $\overline{a}$ and guess $\overline{c}$, such that $\mathcal{A} \models \varphi(\overline{b}, \overline{c})$.

Repeat until we guess $\overline{z}$, such that $\mathcal{A} \models \varphi(\overline{y}, \overline{z})$ and $\overline{z} = \overline{a}'$.

In this case we accept.

FO(pos TC) = NL (Cont'd)

- The space needed is $3k \log n$ plus the space to check if $\varphi(\overline{x}, \overline{x}')$ holds, where k is the arity of $\overline{x}$.

($\supseteq$) Recall that REACH is complete for NL via first-order reductions.

REACH is expressible in FO(pos TC) as follows:

$$\text{REACH} \equiv (\text{TC}_{xy}(E(x,y)))(s,t).$$

But FO(pos TC) is closed under first-order reductions.

It follows that NL $\subseteq$ FO(pos TC).

Deterministic Transitive Closure

- We next define a deterministic version of transitive closure DTC.
- Given a first order relation $\varphi(\overline{x}, \overline{y})$, define its deterministic reduct

$$\varphi_d(\overline{x}, \overline{y}) \equiv \varphi(\overline{x}, \overline{y}) \wedge [(\forall \overline{z}) \neg \varphi(\overline{x}, \overline{z}) \vee (\overline{y} = \overline{z})].$$

- Thus, $\varphi_d(\overline{x}, \overline{y})$ is true iff $\overline{y}$ is the unique descendent of $\overline{x}$.
- Now define

$$(\mathrm{DTC}\varphi) \equiv (\mathrm{TC}\varphi_d).$$

FO(DTC) = L

Theorem

FO(DTC) = L.

- This proof is similar to the preceding theorem.

  We first show that L contains FO(DTC).

  Suppose $\varphi(x_1,\ldots,x_k,y_1,\ldots,y_k) \in \mathrm{L}$.

  Recall the algorithm "Recognizing REACH_d in L":

  1. $b := s$; $i := 0$; $n := \|G\|$
  2. while $b \neq t \wedge i < n \wedge (\exists! a)(E(b,a))$ do {
  3. $b :=$ the unique a for which $E(b,a)$
  4. $i := i + 1$}
  5. if $b = t$ then accept else reject

  It determines in logspace whether or not $(\mathrm{DTC}\varphi)(\overline{s},\overline{t})$ holds.

  Instead of checking whether there is an edge from $\overline{b}$ to $\overline{a}$, we check that $\varphi(\overline{b},\overline{a})$ holds.

FO(DTC) = L (Converse)

- Conversely, FO(DTC) contains L.

  We know that REACH_d is complete for L via first-order reductions.

  Moreover, FO(DTC) is closed under first-order reductions.

  Thus, it suffices to show that REACH_d is expressible in FO(DTC).

  This is accomplished via

$$\text{REACH}_d \equiv (\text{DTC}_{xy}(E(x,y)))(s,t).$$

Subsection 3

Normal Form for FO(TC)

FO(pos TC) and Transitive Closure

Lemma

In the presence of the successor relation, every formula $\varphi \in$ FO(pos TC) is equivalent to a single application of transitive closure to a quantifier-free formula,

$$\varphi \equiv (\mathrm{TC}\alpha)(0, \max).$$

- By induction on the complexity of φ.

  There are five cases.

  1. Suppose φ is either atomic or the negation of an atomic formula.
     Let u, v be variables not occurring in φ.
     Then
     $$\varphi \Leftrightarrow (\mathrm{TC}_{uv}\varphi)(0, \max).$$

FO(pos TC) and Transitive Closure (Case 2)

2. Suppose

$$\varphi \equiv (\mathrm{TC}_{\overline{xy}}\psi)(\overline{q},\overline{r}).$$

We wish to replace $\overline{q},\overline{r}$ with $\overline{0},\overline{\max}$.

Put

$$\begin{array}{l}\rho(s_1,t_1,\overline{x},s_2,t_2,\overline{y}) \equiv \\ \left[s_1 = 0 \wedge t_1 = 0 \wedge \overline{x} = \overline{0} \wedge s_2 = 0 \wedge t_2 = \max \wedge \overline{y} = \overline{q}\right] \\ \vee[s_1 = 0 \wedge t_1 = \max \wedge s_2 = 0 \wedge t_2 = \max \wedge \psi(\overline{x},\overline{y})] \\ \vee[s_1 = 0 \wedge t_1 = \max \wedge \overline{x} = \overline{r} \wedge s_2 = t_2 = \max \wedge \overline{y} = \overline{\max}].\end{array}$$

Variables s,t split the ρ-path in three stages:

- If $st = 00$, set $\overline{x}$ to $\overline{q}$ and go to next stage.
- If $st = 0\max$, take a ψ step and stay in this stage. When $\overline{r}$ is reached, go to next stage.
- If $st = \max\max$, set $\overline{x} = \max$ and stop.

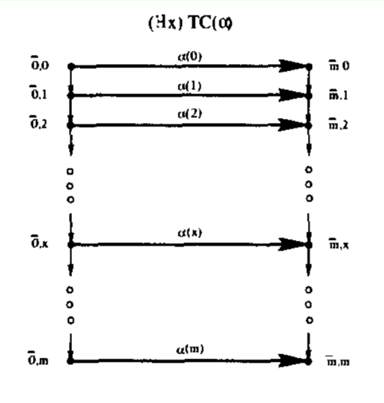
Thus,

$$\varphi \Leftrightarrow (\mathrm{TC}_{s_1t_1\overline{x}s_2t_2\overline{y}}\rho)(\overline{0},\overline{\max}).$$

FO(pos TC) and Transitive Closure (Case 3 Illustration)

3. Suppose

$$\varphi \equiv (\exists x)(\mathrm{TC}_{\overline{uv}}\alpha(x))(\overline{0},\overline{\max}).$$



FO(pos TC) and Transitive Closure (Case 3)

3. Suppose
$$\varphi \equiv (\exists x)(\mathrm{TC}_{\overline{u}\overline{v}}\alpha(x))(\overline{0},\overline{\max}).$$
Here the notation means that the transitive closure is taken over the relation $\alpha(\overline{u},\overline{v})$ and variable x occurs free in α.

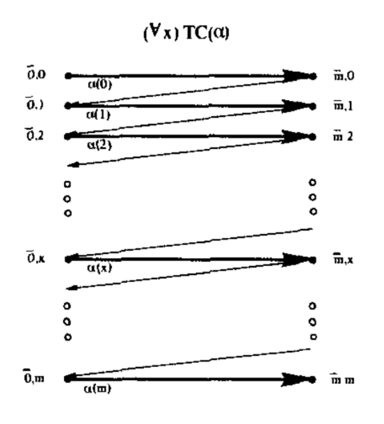
Put
$$\begin{aligned}\chi(\overline{u},x_1,\overline{v},x_2) \quad \equiv \quad & [\overline{u}=\overline{0} \wedge \mathrm{SUC}(x_1,x_2)] \\ & \vee[\alpha(\overline{u},\overline{v};x_1) \wedge x_1 = x_2] \\ & \vee[\overline{u}=\overline{\max} \wedge \overline{v}=\overline{\max} \wedge \mathrm{SUC}(x_1,x_2)].\end{aligned}$$
Formula χ allows a guess of x on the first step.

So
$$\varphi \Leftrightarrow (\mathrm{TC}_{\overline{u}x_1\overline{v}x_2}\chi)(\overline{0},\overline{\max}).$$

FO(pos TC) and Transitive Closure (Case 4 Illustration)

4. Suppose

$$\varphi \equiv (\forall x)(\mathrm{TC}_{\overline{u},\overline{v}}\alpha(x))(\overline{0},\overline{\max}).$$



FO(pos TC) and Transitive Closure (Case 4)

4. Suppose

$$\varphi \equiv (\forall x)(\mathrm{TC}_{\overline{u},\overline{v}}\alpha(x))(\overline{0},\overline{\max}).$$

In this case, we simulate $(\forall x)$ by searching through all x's in order, using SUC.

Put

$$\begin{array}{rcl} \nu(\overline{u},x_1,\overline{v},x_2) & \equiv & [\overline{u} \neq \overline{\max} \wedge \alpha(\overline{u},\overline{v},x_1) \wedge x_1 = x_2] \\ & & \vee[\overline{u} = \overline{\max} \wedge \overline{v} = \overline{0} \wedge \mathrm{SUC}(x_1,x_2)]. \end{array}$$

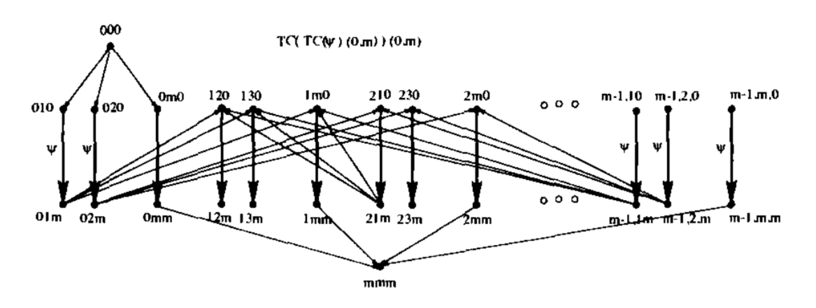
Thus,

$$\varphi \Leftrightarrow (\mathrm{TC}_{\overline{u}x_1\overline{v}x_2}\nu)(\overline{0},\overline{\max}).$$

FO(pos TC) and Transitive Closure (Case 5 Illustration)

5. Suppose

$$\varphi \equiv (\mathrm{TC}_{\overline{uv}}[\mathrm{TC}_{\overline{xy}}\psi](\overline{0},\overline{\max}))(\overline{0},\overline{\max}).$$



FO(pos TC) and Transitive Closure (Case 5)

5. Suppose

$$\varphi \equiv (\mathrm{TC}_{\overline{uv}}[\mathrm{TC}_{\overline{xy}}\psi](\overline{0},\overline{\max}))(\overline{0},\overline{\max}).$$

In this case, formula ψ has free variables $\overline{x},\overline{y},\overline{u},\overline{v}$.

The inner transitive closure is on $\overline{x},\overline{y}$, treating the other variables as parameters.

The outer transitive closure is on $\overline{u},\overline{v}$.

We combine these two TC's into a single transitive closure on δ defined as follows:

$$\begin{array}{l} \delta(\overline{u}_1,\overline{v}_1,\overline{x},\overline{u}_2,\overline{v}_2,\overline{y}) \equiv [\overline{x}=\overline{y}=\overline{0} \wedge \overline{u}_1=\overline{v}_1=\overline{u}_2=\overline{0}] \\ \vee[\overline{x}\neq\overline{\max}\wedge\overline{u}_1\neq\overline{v}_1\wedge\overline{u}_1=\overline{u}_2\wedge\overline{v}_1=\overline{v}_2\wedge\psi(\overline{x},\overline{y};\overline{u}_1,\overline{v}_1)] \\ \vee[\overline{x}=\overline{\max}\wedge\overline{v}_1\neq\overline{\max}\wedge\overline{y}=\overline{0}\wedge\overline{u}_2=\overline{v}_1] \\ \vee[\overline{x}=\overline{\max}\wedge\overline{v}_1=\overline{\max}\wedge\overline{y}=\overline{\max}\wedge\overline{u}_2=\overline{v}_2=\overline{\max}]. \end{array}$$

FO(pos TC) and Transitive Closure (Case 5 Cont'd)

- We claim
$$\varphi \Leftrightarrow (\mathrm{TC}_{\overline{u}_1\overline{v}_1\overline{x}\overline{u}_2\overline{v}_2\overline{y}}\delta)(\overline{0},\overline{\max}).$$
This holds because a δ path consists exactly of a series of $\psi(\cdot,\cdot;u,v)$ paths from $\overline{0}$ to $\overline{\max}$, with u, v fixed.
At the end of any such path we know that $(\mathrm{TC}_{\overline{xy}}\psi(\overline{u},\overline{v}))(\overline{0},\overline{\max})$ holds.
The δ path may now appropriately step from $(\overline{\max},\overline{u},\overline{v})$ to $(\overline{0},\overline{v},\overline{w})$.
That is, it may reach v and begin trying to move from v to w.
- The cases of disjunction and conjunction follow easily from Cases 3 and 4, respectively.

FO(DTC) and Transitive Closure

Lemma

Every formula $\varphi \in \text{FO(DTC)}$ is equivalent to a single application of deterministic transitive closure to a quantifier-free formula,

$$\varphi \equiv (\text{DTC}\alpha)(\overline{0}, \overline{\max}).$$

- We modify the construction in the proof of the lemma so that a deterministic path is never turned into a nondeterministic path.

  The most interesting case is the existential quantifier,

  $$\varphi \equiv (\exists x)(\text{DTC}_{\overline{u},\overline{u}'}\alpha(x))(\overline{0}, \overline{\max}).$$

  Instead of the path finder guessing the correct x, the path:
  - Tries all x's;
  - Goes to $\overline{\max}$ when a correct one is found.

FO(DTC) and Transitive Closure (Cont'd)

- We use the fact that there is a path in an n^k vertex graph iff there is such a path of length at most $n^k - 1$.

  Let $k = \text{arity}(\overline{z}) = \text{arity}(\overline{w}) = \text{arity}(\overline{u}) = \text{arity}(\overline{s})$.

  In the following, we use:
  - Counter $\overline{z}$ to cut off a cycling α-path;
  - $\overline{w}$ to find the α-successor of u, if one exists;
  - $\overline{s}$ to store this α-successor while we check that there are no others.

  We abuse notation and write $\text{SUC}(\overline{z}, \overline{z}')$ to mean that $\overline{z}'$ is the successor of $\overline{z}$ in the lexicographical ordering induced by the successor relation SUC.

FO(DTC) and Transitive Closure (Cont'd)

- Let

$$\chi'(\overline{u},\overline{z},\overline{w},\overline{s},x,\overline{u}',\overline{z}',\overline{w}',\overline{s}',x') \equiv \delta_1 \vee \delta_2 \vee \delta_3 \vee \delta_4 \vee \delta_5 \vee \delta_6 \vee \delta_7,$$

where the meaning of the mutually exclusive δ_i's are as follows:

1. $(\overline{u} = \overline{max})$: Success. So set all primed variables to $\overline{max}$ and halt.
2. $(\overline{z} = \overline{max})$: Failure on x because the counter has overflowed. So set $x' = x + 1$.
3. $(\overline{w} = \overline{max}) \wedge \neg\alpha(\overline{u},\overline{w};x) \wedge \neg\alpha(\overline{u},\overline{s};x)$: Failure on x because there is no α-edge leaving $\overline{u}$. So set $x' = x + 1$.
4. $\alpha(\overline{u},\overline{s};x) \wedge \alpha(\overline{u},\overline{w};x) \wedge \overline{s} \neq \overline{w}$: Failure on x because $\overline{u}$ has more than one α-successor. So set $x' = x + 1$.
5. $(\overline{w} = \overline{max}) \wedge \neg\alpha(\overline{u},\overline{w};x) \wedge \neg\alpha(\overline{u},\overline{s};x)$: Failure on x because there is no α-path leaving $\overline{u}$. So set $x' = x + 1$.
6. $\overline{w} = \overline{max} \wedge (\alpha(\overline{u},\overline{s};x) \oplus \alpha(\overline{u},\overline{w};x))$: There is a unique α-successor of $\overline{u}$. Increment $\overline{z}$ and set $\overline{u}'$ to its successor.
7. $\neg\alpha(\overline{u},\overline{w};x)$: Increment $\overline{w}$ and keep looking for an α-edge leaving $\overline{u}$.

First-Order Definitions of the δ_i's

- For completeness we include the first-order definitions of the δ_i's:
  - $\delta_1 \equiv \overline{u} = \overline{u}' = \overline{z}' = \overline{w}' = \overline{\max} \wedge x' = \max;$
  - $\delta_2 \equiv \overline{u} \neq \overline{\max} \wedge \mathrm{SUC}(x, x') \wedge \overline{z} = \overline{\max} \wedge \overline{u}' = \overline{z}' = \overline{w}' = \overline{0};$
  - $\delta_3 \equiv \overline{u} \neq \overline{\max} \wedge \mathrm{SUC}(x, x') \wedge \overline{z} \neq \overline{\max} \wedge \overline{w} = \overline{\max} \wedge \neg\alpha(\overline{u}, \overline{w}; x) \wedge \neg\alpha(\overline{u}, \overline{s}; x) \wedge \overline{u}' = \overline{z}' = \overline{w}' = \overline{0};$
  - $\delta_4 \equiv \overline{u} \neq \overline{\max} \wedge \mathrm{SUC}(x, x') \wedge \overline{z} \neq \overline{\max} \wedge \alpha(\overline{u}, \overline{s}; x) \wedge \alpha(\overline{u}, \overline{w}; x) \wedge \overline{s} \neq \overline{w} \wedge \overline{u}' = \overline{z}' = \overline{w}' = \overline{0};$
  - $\delta_5 \equiv \overline{u} \neq \overline{\max} \wedge \mathrm{SUC}(x, x') \wedge \overline{z} \neq \overline{\max} \wedge (\overline{w} = \overline{\max}) \wedge \neg\alpha(\overline{u}, \overline{w}; x) \wedge \neg\alpha(\overline{u}, \overline{s}; x) \wedge \overline{u}' = \overline{z}' = \overline{w}' = \overline{0};$
  - $\delta_6 \equiv \overline{u} \neq \overline{\max} \wedge x' = x \wedge \mathrm{SUC}(\overline{z}, \overline{z}') \wedge \overline{w} = \overline{\max} \wedge \alpha(\overline{u}, \overline{u}'; x) \wedge (\alpha(\overline{u}, \overline{s}; x) \oplus \alpha(\overline{u}, \overline{w}; x)) \wedge \overline{w}' = \overline{0};$
  - $\delta_7 \equiv \overline{u} \neq \overline{\max} \wedge \overline{z} \neq \overline{\max} \wedge \neg\alpha(\overline{u}, \overline{w}; x) \wedge \mathrm{SUC}(\overline{w}, \overline{w}') \wedge \overline{u}' = \overline{u} \wedge \overline{z}' = \overline{z} \wedge x' = x.$

  It follows that

$$\varphi \equiv (\mathrm{DTC}\chi)(\overline{0}, \overline{\max}).$$

FO(DTC) and Transitive Closure (Negation)

- The remaining case is negation:

$$\varphi \equiv \neg(\mathrm{DTC}_{x_1 \ldots x_k y_1 \ldots y_k} \psi)(\overline{0}, \overline{\max}).$$

We can handle this case in a similar way to the above case.

We add k-tuples of variables:

  - $\overline{z}, \overline{z}'$ to serve as a counter;
  - $\overline{w}, \overline{w}'$ to run through possible ψ-successors;
  - $\overline{s}, \overline{s}'$ to store the candidate ψ-successor while checking that it is unique.

We start at $\overline{0}$, find a unique ψ-successor of $\overline{x} = \overline{0}$, and increment the counter and repeat.

If we ever get to $\overline{y} = \overline{\max}$, then, instead, we return to $\overline{0}$, i.e., reject.

If ever the counter overflows ($\overline{z} = \overline{\max}$) or there are zero or more than one ψ-successors of $\overline{x}$, then we go to $\overline{\max}$, i.e., accept.

Expressibility of BIT in FO(wo BIT)(DTC)

Proposition

Relation BIT is definable in FO(wo BIT)(DTC). Thus, it is also definable in FO(wo BIT)(TC) and FO(wo BIT)(LFP).

- We first show that PLUS is definable using DTC and SUC.

  We say that there is an α-edge from $\langle x, y \rangle$ to $\langle u, v \rangle$ iff $u = x - 1$ and $v = y + 1$,

  $$\alpha(x, y, u, v) \equiv \mathrm{SUC}(u, x) \wedge \mathrm{SUC}(y, v).$$

  Using transitive closure we get

  $$\mathrm{PLUS}(x, y, z) \equiv (\mathrm{DTC}\alpha)(x, y, 0, z).$$

Expressibility of BIT in FO(wo BIT)(DTC) (Con'd)

- Now define β as follows:

$$\begin{array}{rcl}\beta(w_1, j_1, w_2, j_2) & \equiv & (\exists z(\mathrm{PLUS}(w_2, w_2, z) \wedge (w_1 = z \vee \mathrm{SUC}(z, w_1))) \\ & & \wedge \mathrm{SUC}(j_2, j_1)).\end{array}$$

Note that

$$\beta(w, j, w', j+1) \text{ holds} \quad \text{iff} \quad w' = \left\lfloor \frac{w}{2} \right\rfloor.$$

Let $\mathrm{ODD}(z)$ abbreviate

$$\exists x \exists y(\mathrm{PLUS}(x, x, y) \wedge \mathrm{SUC}(y, z)).$$

It follows that

$$\mathrm{BIT}(w, j) \equiv (\exists z)(\mathrm{ODD}(z) \wedge (\mathrm{DTC}\beta)(w, j, z, 0)).$$

Completeness of REACH_d, REACH for L, NL With SUCC

Corollary

In the presence of the successor relations, problems REACH_d, REACH and REACH_a are complete for L, NL and P, respectively, via quantifier-free reductions.

- The preceding lemma shows how to write any formula in L as a quantifier-free reduction to REACH_d.

  A previous lemma does the same thing for NL and REACH.

  We can define an alternating transitive closure operator ATC that similarly formalizes alternating reachability.

  A similar proof gives the same quantifier-free normal form for FO(ATC).

Subsection 4

Logspace is Primitive Recursive

Initial Functions

- Fix a vocabulary τ, which may include some function symbols.
- Define the **initial functions** to be the following.
  1. **Constant functions**: 0 and max are 0-ary constant functions.
  2. **Successor function**: For each $r > 0$,
     $$\mathrm{SUC}(x_1, \ldots, x_r) = \overline{x} + 1,$$
     the successor of $\overline{x}$ in lexicographic order, and undefined if $\overline{x} = \overline{\max}$.
  3. **Projection functions**: For $\ell > 0$ and $1 \leq i_1 < i_2 < \cdots < i_r \leq \ell$,
     $$\pi^{\ell}_{i_1 \ldots i_r}(x_1, \ldots, x_\ell) = (x_{i_1}, x_{i_2}, \ldots, x_{i_r}).$$
  4. **Input symbols**: For each function or constant symbol in τ we have the corresponding function.
     For each relation symbol, we have the corresponding characteristic function.

Primitive Recursive Functions on Finite Structures

- The initial functions are then closed under the following operations.
  1. **Composition**: If $h_1, \ldots, h_r$ are functions from s-tuples to a_i tuples and g is a function on $(a_1 + a_2 + \cdots + a_r)$-tuples, then the composition of g and $h_1, \ldots, h_r$ is defined by,
  $$g \circ (h_1, \ldots, h_r)(x_1, \ldots, x_s) = g(h_1(\overline{x}), h_2(\overline{x}), \ldots, h_r(\overline{x})).$$
  2. **Primitive recursion**: If g and h are functions of appropriate arity, then the following scheme defines f by primitive recursion from g and h,
  $$\begin{array}{rcl} f(\overline{x}, \overline{0}) & = & g(\overline{x}) \\ f(\overline{x}, \mathrm{SUC}(\overline{t})) & = & h(\overline{x}, \overline{t}, f(\overline{x}, \overline{t})). \end{array}$$
- Define the **primitive recursive functions on finite structures** to be the closure of initial functions under composition and primitive recursion.

Gurevich's Theorem

Theorem

The primitive recursive functions on finite structures are the partial functions computable in logspace.

- For the upperbound, one shows that:
  - REACH is primitive recursive using a previous algorithm;
  - The primitive recursive functions are closed under quantifier-free reductions.

  For the "if" direction, some lemmas are used, asserting the following facts.

Gurevich's Theorem (Sketch of Proof)

- Any boolean combination of primitive recursive predicates is primitive recursive.
- Given a predicate $P(\overline{x})$ and functions $g(\overline{x})$ and $h(\overline{x})$, let $f(\overline{x})$ be defined by

$$f(\overline{x}) = \begin{cases} g(\overline{x}), & \text{if } P(\overline{x}), \\ h(\overline{x}), & \text{otherwise.} \end{cases}$$

  If P and g, h are primitive recursive, then f is primitive recursive.
- A concatenation

$$(f_1(\overline{x}), \ldots, f_m(\overline{x}))$$

  of primitive recursive functions $f_1, \ldots, f_m$ is primitive recursive.
- Let $f_1, \ldots, f_m$ be defined by simultaneous primitive recursion

$$\begin{array}{rcl} f_i(\overline{x}, \overline{0}) & = & g_i(\overline{x}, \overline{0}); \\ f_i(\overline{x}, \text{SUC}(t)) & = & h_i(\overline{x}, \overline{t}, f_1(\overline{x}, \overline{t}), \ldots, f_m(\overline{x}, \overline{t})). \end{array}$$

  If the functions g_i, h_i are primitive recursive, then so are $f_1, \ldots, f_m$.

Gurevich's Theorem (Sketch of Proof Cont'd)

- Let f be a logspace computable function.

  We must show that f is primitive recursive.

  Suppose M is a multihead Turing machine computing f.

  Inputs $(\mathcal{A}, \overline{a})$, with $\|A\| = n$, are presented in some standard way.

  Suppose for simplicity that each basic ℓ-ary predicate $P^{\mathcal{A}}$ is presented on a separate input tape of length n^ℓ, where for each $\overline{x} \in |\mathcal{A}|^\ell$, the truth value of $P^{\mathcal{A}}(\overline{x})$, is coded in cell number $\sum x_i n^{\ell - i}$.

  Suppose, also, that each basic function $f^{\mathcal{A}}$ is presented on a separate tape as the respective graph predicate.

  Finally, the components of $\overline{a}$ are presented in unary notation on separate tapes.

  Then every input tape can be described by a function that is easily definable by cases.

Gurevich's Theorem (Sketch of Proof Cont'd)

- Let $H_1, \ldots, H_m$ be the heads on the input tapes of M.

  For $i = 1, \ldots, m$, let $\mathrm{Sym}_i(\overline{x}_i)$ be the content of cell $\overline{x}_i$ of the tape i.

  There exists a positive integer k, such that, M finds itself in the halting state, say q, at the moment $n^k - 1$.

  Suppose that $\overline{t}$ ranges over n^k.

  $\mathrm{State}(\overline{t})$ is the state of M at moment $\overline{t}$.

  $\mathrm{Head}_i(\overline{t})$ is the position of head H_i at moment $\overline{t}$.

  The functions State and Head_i are defined by simultaneous induction, which uses the compositions

  $$\mathrm{Sym}_1(\mathrm{Head}_1(\overline{t})), \ldots, \mathrm{Sym}_m(\mathrm{Head}_m(\overline{t})).$$

  Under natural assumptions about the output mechanism of M, one defines $\mathrm{Output}(\overline{t})$ by induction from $\mathrm{State}(\overline{t})$.

  Finally, we set $f = \mathrm{Output}(\overline{\mathrm{max}})$.

Subsection 5

NSPACE$[s(n)]$ = co-NSPACE$[s(n)]$

FO(pos TC) = FO(TC)

Theorem

For any class of finite, ordered structures,

$$\mathrm{FO(pos\ TC)} = \mathrm{FO(TC)}.$$

- By a previous lemma, it suffices to show that the relation

$$\neg(\mathrm{TC}_{uu'}E(u,u'))(0,\max),$$

meaning there is no path from 0 to max, is expressible in FO(pos TC). To do this, we count the number of reachable vertices. Fix a graph $G \in \mathrm{STRUC}[\tau_g]$.

FO(pos TC) = FO(TC) (Cont'd)

- As usual, we consider the elements of G both as numbers and as vertices.

  In one setting, as distances, we think of these numbers as ranging from 0 to $n-1$.

  In another setting, as counts of the number of reachable vertices, we have numbers ranging from 1 to n.

  Writing these two sets of numbers as numbers rather than as vertices makes our notation simpler to understand.

FO(pos TC) = FO(TC) (Cont'd)

- Define n_d to be the number of vertices in G that are reachable from 0 in a path of length at most d.

  Given number n_d, we show how to compute number n_{d+1}.

  As a first step, we show that n_d allows us to say in FO(pos TC) that there is no path of length at most d from 0 to a given vertex.

  Claim: The following formulas are expressible in FO(pos TC).

  1. DIST(x, d), meaning that there is a path of length at most d from 0 to x;
  2. NDIST$(x, d; m)$, which, when $m = n_d$, means that there is no path of length at most d from 0 to x.

FO(pos TC) = FO(TC) (Proof of the Claim)

- There is no trouble writing $\text{DIST}(x,d)$ positively,

$$\begin{aligned} \text{DIST}(x,d) &\equiv \text{TC}(\alpha)(0,0,x,d), \text{ where} \\ \alpha(a,i,b,j) &\equiv (E(a,b) \vee a = b) \wedge \text{SUC}(i,j). \end{aligned}$$

We write the formula, $\text{NDIST}(x,d;m) \in \text{FO(posTC)}$ to mean

$$\text{NDIST}(x,d;m) \equiv (\text{There are at least } m \text{ vertices } v) \\ (v \neq x \wedge \text{DIST}(v,d)).$$

It will then follow that, when $m = n_d$, $\text{NDIST}(x,d;m)$ is equivalent to $\neg\text{DIST}(x,d)$.

FO(pos TC) = FO(TC) (Proof of the Claim Cont'd)

- Define edge relation β on pairs of vertices by

$$\begin{aligned}\beta(v,c,v',c') \quad \equiv \quad & 0 \neq x \wedge \mathrm{SUC}(v,v') \\ & \wedge(c = c' \vee (\mathrm{SUC}(c,c') \wedge \mathrm{DIST}(v',d) \wedge v' \neq x)).\end{aligned}$$

  Suppose that c is the number of vertices - not including x - that are at most v and reachable from 0 in at most d steps.
  Then we can take a β-step from $\langle v,c\rangle$ to:
  - $\langle v+1,c\rangle$ guessing that $v+1$ is not reachable from 0 in d steps;
  - $\langle v+1,c+1\rangle$ if we prove that $v+1$ is not equal to x and is reachable from 0 in d steps.

  Thus, there is a path from $\langle 0,1\rangle$ to $\langle v,c\rangle$ iff there are at least c vertices not equal to x and at most v, such that $\mathrm{DIST}(v,d)$:

$$\mathrm{TC}(\beta)(0,1,v,c) \ \Leftrightarrow \ c \leq |\{w : w \leq v \wedge \mathrm{DIST}(w,d)\}|.$$

  NDIST can now be defined by

$$\mathrm{NDIST}(x,d;m) \equiv \mathrm{TC}(\beta)(0,1,\max,m).$$

FO(pos TC) = FO(TC) (Cont'd)

- Using the Claim, we now define the relation $\delta(d, m, d', m')$ so that if $m = n_d$, then $m' = n_{d+1}$.

  We simply cycle through all the vertices, counting how many of them are reachable in $d + 1$ steps:

$$\begin{aligned}\delta(d, m, d', m') &\equiv \mathrm{SUC}(d, d') \wedge \mathrm{TC}(\gamma)(0, 1, \max, m')\\ \gamma(v, c, v', c') &\equiv \mathrm{SUC}(v, v') \wedge ([\mathrm{SUC}(c, c') \wedge \mathrm{DIST}(v', d + 1)] \vee\\ &\qquad [c = c' \wedge (\forall z)(\mathrm{NDIST}(z, d; m) \vee (z \neq v' \wedge \neg E(z, v')))]).\end{aligned}$$

  It follows that formula $\mathrm{TC}(\delta)(0, 1, n - 1, m)$ holds iff $m = n_{n-1}$ is the number of vertices in G that are reachable from 0.

  Using this m, we can express the nonexistence of a path positively as claimed,

$$\neg\mathrm{TC}(E)(0, x) \equiv (\exists m)(\mathrm{TC}(\delta)(0, 1, n - 1, m) \wedge \mathrm{NDIST}(x, n - 1; m)).$$

A Nicer Characterization of NL

Corollary

We have

$$\mathrm{NL} = \mathrm{FO(TC)}.$$

Furthermore every formula $\varphi \in \mathrm{FO(TC)}$ is equivalent to a single application of transitive closure to a quantifier-free formula,

$$\varphi \equiv (\mathrm{TC}\alpha)(\overline{0}, \overline{\mathrm{max}}).$$

Nondeterministic Space and Complements

Corollary

For any $s(n) \geq \log n$,

$$\text{NSPACE}[s(n)] = \text{co-NSPACE}[s(n)].$$

- We have shown that NL = FO(pos TC) = FO(TC).

  Consider any NL property and negate it.

  Then it is still in FO(TC) and, thus, NL.

  It follows that NL = co-NL.

Nondeterministic Space and Complements

- Suppose now that $s(n) \geq \log n$.

  Let N be an NSPACE[$s(n)$] machine.

  Suppose its input w is of length $n = |w|$.

  The computation graph of N on input w has $m = 2^{O(s(n))}$ nodes.

  The question whether N rejects w is the non-reachability problem on this computation graph.

  By the theorem, it is solvable in NSPACE[$\log(m)$].

  That is, it is solvable in NSPACE[$s(n)$].

  Even if we do not know what $s(n)$ is, we can apply the same construction by:
  - Starting with $s = 1$;
  - Incrementing s each time a reachable configuration in the computation graph of size $s + 1$ is found.

Context Sensitive Languages

- Let CSL be the class of context sensitive languages.
- Kuroda showed in 1964 that

$$\text{CSL} = \text{NSPACE}[n].$$

Corollary

The class of context sensitive languages is closed under complementation.

Subsection 6

Restrictions of SO

Horn Formulas

Definition (Horn Formulas)

Let Φ be a second-order formula in prenex form,

$$\Phi \equiv (Q_1 P_1^{a_1}) \cdots (Q_k P_k^{a_k})(\forall \overline{x})\alpha,$$

such that:

- The first-order part of Φ is universal;
- The quantifier-free part α is in conjunctive normal form, i.e., a conjunction of clauses, each of which is a disjunction.

We say that Φ is a **second-order Horn formula** iff the quantifier-free part has at most one positive occurrence of a quantified predicate P_i per clause. Let SO-Horn be the set of boolean queries describable by second-order Horn formulas.

Krom Formulas

Definition (Krom Formulas)

Let Φ be a second-order formula in prenex form,

$$\Phi \equiv (Q_1 P_1^{a_1}) \cdots (Q_k P_k^{a_k})(\forall \overline{x})\alpha,$$

such that:

- The first-order part of Φ is universal;
- The quantifier-free part α is in conjunctive normal form, i.e., a conjunction of clauses, each of which is a disjunction.

Φ is a **second-order Krom formula** iff the quantifier-free part has at most two occurrences of a quantified predicate per clause.
Let SO-Krom be the set of boolean queries describable by second-order Krom formulas.

HORN-SAT and 2-SAT

Proposition

Let HORN-SAT and 2-SAT be the restrictions of the boolean satisfiability problem to Horn and Krom formulas respectively. Then:

1. HORN-SAT is complete for P via quantifier-free reductions.
2. 2-SAT is complete for NL via quantifier-free reductions.

1. One way to see that HORN-SAT is in P is to express it in FO(LFP).

   Inductively, define a variable to be true if it occurs positively in a clause all of whose other variables are true.

   A Horn formula is satisfiable iff this inductively defined assignment satisfies the formula.

HORN-SAT and 2-SAT (Part 1 Cont'd)

- For the inductive definition, let

$$\varphi(R,x) \equiv (\exists c)(P(c,x) \wedge (\forall y.N(c,y))R(y)).$$

Then let $T \equiv (\mathrm{LFP}\varphi)$.

Then,

$$\text{HORN-SAT} \equiv (\forall c)(\exists x)((P(c,x) \wedge T(x)) \vee (N(c,x) \wedge \neg T(x))).$$

We know REACH_a is complete for P via quantifier-free reductions.

So the complementary problem $\overline{\mathrm{REACH}_a}$ is complete for co-P = P.

To show that HORN-SAT is complete for P it therefore suffices to show that

$$\overline{\mathrm{REACH}_a} \leq_{\mathrm{qf}} \text{HORN-SAT}.$$

HORN-SAT and 2-SAT (Part 1 Cont'd)

- We must show that

$$\overline{\text{REACH}_a} \leq_{\text{qf}} \text{HORN-SAT}.$$

The idea of the reduction is simple.

Let G be an alternating graph.

REACH_a remains complete when graphs are restricted to outdegree two.

So we assume that the outdegree of G is two.

Formula $I(G)$ consists of the following clauses:

  - t;
  - $(e \vee \neg f_1 \vee \neg f_2)$ where e is a universal node and has edges to $f_1 \neq f_2$;
  - $(e \vee \neg f_1)$ where there is an edge from e to f_1 and e is existential;
  - $\neg s$.

I is quantifier-free definable.

Moreover, $I(G) \in \text{HORN-SAT}$ iff $G \in \overline{\text{REACH}_a}$.

HORN-SAT and 2-SAT (Part 2)

2. 2-SAT is in NL because a clause with two literals, $\ell_1 \vee \ell_2$, can be understood as two edges in a graph, $\overline{\ell_1} \to \ell_2$, $\overline{\ell_2} \to \ell_1$.

   Let 2-CNF be the set of CNF formulas that have at most two literals per clause.

   Thus, 2-SAT = SAT $\cap$ 2-CNF.

   We can show that a 2-CNF formula φ is satisfiable iff there is no variable x for which there is a path in the corresponding graph from x to $\overline{x}$ and from $\overline{x}$ to x.

   We now write this in FO(TC).

   Suppose $(x, 0)$ encodes literal x and $(x, 1)$ encodes $\overline{x}$.

HORN-SAT and 2-SAT (Part 2 Cont'd)

- Formula δ encodes the edges from literal to literal,

$$\begin{array}{rcl} \text{Occur}(c,x,b) & \equiv & (b=0 \wedge P(c,x)) \vee (b=1 \wedge N(c,x)); \\ \delta(x,b,x',b') & \equiv & (\exists c)(\text{Occur}(c,x,1-b) \\ & & \qquad \wedge\, \text{Occur}(c,x',b') \wedge x \neq x'). \end{array}$$

  PATH is the transitive closure of this edge relation.

$$\begin{array}{rcl} \text{PATH}(u,d,u',d') & \equiv & (\text{TC}_{xbx'b'}\delta)(u,d,u',d'); \\ \text{2-SAT} & \equiv & (\forall x)\neg(\text{PATH}(x,0,x,1) \\ & & \qquad \wedge\, \text{PATH}(x,1,x,0)). \end{array}$$

HORN-SAT and 2-SAT (Part 2 Cont'd)

- We know that $\overline{\text{REACH}}$ is complete for NL via quantifier-free reductions.

  The completeness of 2-SAT will follow when we show that

  $$\overline{\text{REACH}} \leq_{\text{qf}} \text{2-SAT}.$$

  Given a graph G, the boolean formula $I(G)$ will have the following clauses:

  - s;
  - $\neg a \vee b$, when (a, b) is an edge of G;
  - $\neg t$.

  I is quantifier-free definable.

  Moreover, $I(G) \in \text{2-SAT}$ iff $G \in \overline{\text{REACH}}$.

  It follows that HORN-SAT and 2-SAT are complete via quantifier-free reductions.

Using SO-Horn and SO-Krom Formulas

- The proof of the proposition also shows how to express:
  - The negation of the boolean query REACH_a as a SO-Horn formula.

$$\begin{array}{rcl}\overline{\text{REACH}_a} & \equiv & (\exists T^1)(\forall e f_1 f_2)(T(t) \wedge \neg T(s) \wedge \\ & & (T(e) \vee \neg T(f_1) \vee A(e) \vee \neg E(e, f_1)) \wedge \\ & & (T(e) \vee \neg T(f_1) \vee \neg T(f_2) \vee \neg A(e) \vee f_1 = f_2 \\ & & \vee \neg E(e, f_1) \vee \neg E(e, f_2) \vee \neg f_1 \vee \neg f_2)).\end{array}$$

  - The negation of the boolean query REACH as a SO-Krom formula.

$$\begin{array}{rcl}\overline{\text{REACH}} & \equiv & (\exists T^1)(\forall ab)(T(s) \wedge \neg T(t) \\ & & \wedge (T(b) \vee \neg T(a) \vee \neg E(a, b))).\end{array}$$

SO-Horn and SO-Krom Collapse to SO∃ Parts

Lemma

The following equations hold for all sets of structures - finite or infinite:

1. SO-Horn = SO∃-Horn;
2. SO-Krom = SO∃-Krom.

- It suffices to show that Horn or Krom formulas of the form

$$\Psi \equiv (\forall P)(\exists Q_1 \cdots Q_r)(\forall \overline{z})\alpha$$

are equivalent to SO∃-Horn and SO∃-Krom formulas respectively.

SO-Horn Collapses to SO∃ Part (Claim)

- In the Horn case we first observe the following.

  Claim: Consider the Horn formula

$$\Psi \equiv (\forall P)(\exists Q_1 \cdots Q_r)(\forall \overline{z})\alpha.$$

  If Ψ holds for every P that is false on at most one tuple, then Ψ holds (for every P).

  Suppose P has arity k.

  For every k-tuple $\overline{y}$, let $P^{\overline{y}}$ be the predicate that is:
  - False at $\overline{y}$;
  - True at all other points of $|\mathcal{A}|^k$.

  By hypothesis, for all $\overline{y}$, there exist predicates $\overline{Q}^{\overline{y}}$, such that,

$$(\mathcal{A}, P^{\overline{y}}, \overline{Q}^{\overline{y}}) \vDash (\forall \overline{z})\varphi.$$

SO-Horn Collapses to SO∃ Part (Claim Cont'd)

- For every predicate $P \neq |\mathcal{A}|^k$, we construct the predicates

$$Q_i = \bigcap_{\overline{y} \notin P} Q_i^{\overline{y}}.$$

We claim that

$$(\mathcal{A}, P, \overline{Q}) \vDash (\forall \overline{z})\varphi.$$

Suppose, to the contrary, $(\mathcal{A}, P, \overline{Q}) \nvDash (\forall \overline{z})\varphi$.
So there exist:
  - A relation $P \neq |\mathcal{A}|^k$;
  - A clause c of φ;
  - An assignment $\overline{a} : \{z_1, \ldots, z_s\} \to |\mathcal{A}|$,

such that $\mathcal{A} \vDash \neg c(\overline{a}, P, \overline{Q})$.

We show that there exists $\overline{y}$, such that $c(\overline{a}, P^{\overline{y}}, \overline{Q}^{\overline{y}})$ is also false.

Suppose the head of $c(\overline{a})$ is $P(\overline{u})$.

Then we take $\overline{y} = \overline{u} \notin P$.

SO-Horn Collapses to SO∃ Part (Claim Cont'd)

- Suppose the head of $c(\overline{a})$ is $Q_i(\overline{u})$.
  Then choose a $\overline{y} \notin P$, such that $\overline{u} \notin Q_i^{\overline{y}}$.
  Such a $\overline{y}$ must exist because $\overline{u} \notin Q_i$.
  Otherwise (cases where head is empty or an atom $R(\overline{u})$, where R belongs to the vocabulary of $\mathcal{A}$), take an arbitrary $\overline{y} \notin P$.
  The head of $c(\overline{a}, P^{\overline{y}}, \overline{Q}^{\overline{y}})$ is clearly false.
  Note that the atom $P(\overline{y})$ does not occur in the body of $c(\overline{a}, P, \overline{Q})$.
  This is because $\overline{y} \notin P$ and all atoms in the body of $c(\overline{a}, P, \overline{Q})$ are true.
  Indeed, all other atoms of the form $P(\overline{v})$ that might occur in the body of the clause remain true also for $P^{\overline{y}}$.
  Moreover, every atom $Q_i(\overline{v})$ in the body remains also true if Q_i is replaced by $Q_i^{\overline{y}}$ (because $Q_i \subseteq Q_i^{\overline{y}}$).
  This implies that the clause $c(\overline{a}, P^{\overline{y}}, \overline{Q}^{\overline{y}})$ is false.
  Thus, $(\mathcal{A}, P^{\overline{y}}, \overline{Q}^{\overline{y}}) \vDash \neg(\forall \overline{z})\varphi$, contradicting the hypothesis.

SO-Horn Collapses to SO∃ Part (Cont'd)

- By the Claim, we can replace P by either the true relation, or the relation that is true everywhere but on a fixed tuple $\overline{u}$.

  We start with

$$\begin{aligned}\Psi \equiv\ & (\exists Q_1 \cdots Q_r)(\forall \overline{z})\alpha((\overline{x} = \overline{x})/P(\overline{x})) \\ & \wedge (\forall \overline{u})(\exists Q_1 \cdots Q_r)(\forall \overline{z})\alpha((\overline{x} \neq \overline{u})/P(\overline{x})).\end{aligned}$$

  Then, we transform the conjunction into an equivalent formula in SO∃-Horn,

$$\Psi \equiv (\exists Q_1 \cdots Q_r)(\forall \overline{u})(\forall \overline{z})(\alpha(\textbf{true}/P) \wedge \alpha(\neg \overline{u}/P)).$$

SO-Krom Collapse to SO∃ Part

- We now turn to the Krom case.

  We introduce the notation
  $$A \stackrel{\Psi}{\to} B$$
  to mean that:
  - There is a path in the graph determined by Ψ from literal A to literal B;
  - All intermediate literals are existential, i.e., Q-literals.

  We have the following generalization of the satisfaction condition described in the proof of Part 2 of the preceding proposition.

SO-Krom Collapse to SO∃ Part (Claim)

- Claim: A ∀∃-Krom formula

$$\Psi \equiv \forall X_1 \cdots \forall X_m \exists Y_1 \cdots \exists Y_n \varphi(\overline{X}, \overline{Y}),$$

where φ is a Krom formula, is false iff at least one of the following holds:

1. There are distinct ∀ literals X, X' such that $X \stackrel{\Psi}{\to} X'$;
2. There is an ∃ literal Y such that $Y \stackrel{\Psi}{\to} \neg Y$ and $\neg Y \stackrel{\Psi}{\to} Y$.

It follows from this Claim that Ψ is equivalent to the SO∃-Krom formula in which P is replaced by relations that are false at at most two tuples.

This allows using a technique similar to the SO-Horn case.

SO-Krom Collapse to SO∃ Part (Claim Cont'd)

- Any of the two conditions implies that Ψ is false.

  Assume, conversely, that Ψ is false.

  Then, there exists $\varepsilon : \{X_1, \ldots, X_m\} \to \{0, 1\}$, such that

  $$\varphi' \equiv \varphi(\varepsilon, \overline{Y})$$

  is unsatisfiable.

  Suppose $\varphi(\varepsilon, \overline{Q})$ is false because it contains a clause already interpreted false by ε.

  This clause is equivalent to $X \to X'$, for distinct $\forall$ literals X and X'.

  So, in this case, Condition 1 holds.

  Otherwise, by the propositional case, there exists an $\exists$ literal Y, with

  $$\neg Y \stackrel{\varphi'}{\to} Y \quad \text{and} \quad Y \stackrel{\varphi'}{\to} \neg Y.$$

SO-Krom Collapse to SO$\exists$ Part (Claim Cont'd)

- That is, there exists a sequence $Z_0, Z_1, \ldots, Z_\ell$ of $\exists$ literals, such that:
  - $Z_0 = Z_\ell = Y$ and $Z_k = \neg Y$, for some k, $0 < k < \ell$.
  - All implications $Z_i \to Z_{i+1}$ are equivalent to some clause of φ'.

  If $Y \stackrel{\varphi}{\to} \neg Y$ and $\neg Y \stackrel{\varphi}{\to} Y$, then Condition 2 s satisfied.

  Otherwise, take the last implication $Z_i \to Z_{i+1}$ not occurring in φ.

  Then $Z_i = \neg Z_{i+1}$ and φ contains a clause $X \to Z_{i+1}$, where X is a $\forall$ literal, with $\varepsilon(X) = 1$.

  It follows that $X \stackrel{\varphi}{\to} Y$.

  Similarly, we infer that there exists a $\forall$ literal X', such that $\neg X' \stackrel{\varphi}{\to} \neg Y$ and $\varepsilon(\neg X') = 1$.

  In this case $X \stackrel{\varphi}{\to} X'$ and $X \neq X'$, since $\varepsilon(X) = \varepsilon(\neg X') = 1$.

  Hence, Condition 1 is satisfied.

Characterizations of Polynomial Time and Logspace

Theorem

The following equations hold for finite structures that include a successor relation:

1. SO-Horn = P;
2. SO-Krom = NL.

- We have seen that:
  - HORN-SAT is complete for P via quantifier-free reductions;
  - 2-SAT is complete for NL via quantifier-free reductions.

  These, together with the lemma, show that SO-Horn $\subseteq$ P and SO-Krom $\subseteq$ NL.

  We have also seen that SO-Horn and SO-Krom express problems that are complete for P and NL, respectively, via quantifier-free reductions.

  One, therefore, must show that SO-Horn and SO-Krom are closed under quantifier-free reductions.