Counting of Teams in First-Order Team Logics

Juha Kontinen

November 30, 2018

Introduction

- This is joint work with A. Haak, F. Müller, H. Vollmer, and F. Yang.
- The question of the power of counting arises in propositional and predicate logic in a number of contexts. We do extend logics by counting constructs but consider functions arising form counting tuple and relations satisfying a fixed formula.
- A fundamental counting problem on propositional formulas, $\#\mathrm{SAT}$, counts the number of satisfying assignments of a given formula. It is complete for Valiant's class #P that counts accepting paths of nondeterministic polynomial-time Turing machines.
- The class #P is the counting analogue of NP corresponding to existential second order logic, where the quantified relation encodes accepting computation paths of NP-machines. Hence, if we define $\#\mathsf{FO}^{\mathrm{rel}}$ to count accepting assignments to free relational variables in FO-formulae, we obtain $\#\mathsf{FO}^{\mathrm{rel}} = \#\mathsf{P}$. [Saluja et al., 95].

Introduction cont.

- We consider a different model-theoretic way to study counting processes using team-based logics. Here, formulae with free variables are evaluated not for a single assignment to these variables but for *sets* of such assignments.
- We define $\#\mathrm{FO}^{\mathrm{team}}$ to be the class of functions counting teams that satisfy a given FO-formula, and similarly for extensions of FO by various dependencies in team semantics.

Preliminaries

- Formulae of first-order logic (FO) are defined by the following grammar:

  $$\varphi ::= \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi \mid R(\bar{t}) \mid \neg R(\bar{t}) \mid t_1 = t_2 \mid \neg t_1 = t_2$$

- We consider finite ordered structures with a finite vocabulary σ consisting of relation and constant symbols.
- The class of such σ-structures is denoted by STRUC[σ]. We write $\mathrm{enc}_\sigma(\mathcal{A})$ for the binary encoding of a σ-structure $\mathcal{A}$.

Basics of team semantics

For a team X, a structure $\mathcal{A}$, and $\phi \in \mathrm{FO}$, $\mathcal{A} \models_X \phi$ is def. by:

- $\mathcal{A} \vDash_X \alpha$ for α a literal, iff for all $s \in X$, $\mathcal{A} \vDash_s \alpha$.
- $\mathcal{A} \models_X \varphi \vee \psi$, iff there are teams $Y, Z \subseteq X$ s.t. $Y \cup Z = X$, $\mathcal{A} \models_Y \varphi$ and $\mathcal{A} \models_Z \psi$.
- $\mathcal{A} \models_X \varphi \wedge \psi$, iff $\mathcal{A} \models_X \varphi$ and $\mathcal{A} \models_X \psi$.
- $\mathcal{A} \models_X \exists x \varphi$, iff there exists a function $F \colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$, s.t. $\mathcal{A} \models_{X[F/x]} \varphi$.
- $\mathcal{A} \models_X \forall x \varphi$, iff $\mathcal{A} \models_{X[A/x]} \varphi$.

Team semantics cont.

- A sentence φ is *true* in $\mathcal{A}$, written $\mathcal{A} \models \varphi$, if $\mathcal{A} \models_{\{\emptyset\}} \varphi$.
- First-order formulae φ are flat, i.e., $\mathcal{A} \vDash_X \varphi$, iff $\mathcal{A} \vDash_s \varphi$ for all $s \in X$.

The semantics of the relevant dependency atoms are defined by:

- $\mathcal{A} \models_X =\!(\overline{x}, y)$, iff for all $s, s' \in X$, if $s(\overline{x}) = s'(\overline{x})$, then $s(y) = s'(y)$.
- $\mathcal{A} \models_X \overline{x} \perp_{\overline{y}} \overline{z}$, iff for all $s, s' \in X$ such that $s(\overline{y}) = s'(\overline{y})$, there exists $s'' \in X$ such that $s''(\overline{y}) = s(\overline{y})$, $s''(\overline{x}) = s(\overline{x})$ and $s''(z) = s'(z)$.
- $\mathcal{A} \models_X \overline{x} \subseteq \overline{y}$, iff for all $s \in X$ there is $s' \in X$ such that $s(\overline{x}) = s'(\overline{y})$.

Dependence, independence, and inclusion logic

We recall some basic properties of $\mathrm{FO}(=(\dots))$, $\mathrm{FO}(\subseteq)$, and $\mathrm{FO}(\perp)$:

- Formulae of $\mathrm{FO}(=(\dots))$ are *closed downwards*, i.e., $\mathcal{A} \models_X \varphi$ and $Y \subseteq X$ imply $\mathcal{A} \models_Y \varphi$.
- formulae of $\mathrm{FO}(\subseteq)$ are *closed under unions*, i.e., $\mathcal{A} \models_X \varphi$ and $\mathcal{A} \models_Y \varphi$ imply $\mathcal{A} \models_{X \cup Y} \varphi$.
- Formulae of any of these logics have the *empty team property*, i.e., $\mathcal{A} \models_\emptyset \varphi$ always holds.

Expressive power of team based logics

Recall that existential second-order logic (Σ^1_1) consists of formulas of the form $\exists R_1 \ldots \exists R_k \varphi$, where φ is a first-order formula.

Theorem

1. *For every σ-formula φ of* FO($\perp$)*, there is an $\sigma(R)$-sentence $\psi(R)$ of Σ^1_1 such that for all σ-structures $\mathcal{A}$ and teams $X \neq \emptyset$,*

$$\mathcal{A} \models_X \varphi \iff (\mathcal{A}, rel(X)) \models \psi(R), \tag{1}$$

   and vice versa.

2. *The above holds for* FO($=(\ldots)$) *as well, except that in both directions for* FO($=(\ldots)$) *the relation symbol R is assumed to occur only negatively in the sentence $\psi(R)$.*
3. *For any sentence $\varphi \in$* FO($\subseteq$)*, there exists an equivalent sentence ψ of positive greatest fixed point logic (*posGFP*) and vice versa.*

Propositional and quantified Boolean formulae

- We use CNF to denote propositional formulae in conjunctive normal form and k-CNF to denote the class such formulae where each clause contains at most k literals.
- A formula in CNF is the class DualHorn, if each of its clauses contains at most one negative literal.
- For a class C of formulae, we denote by Σ_1-C the class of quantified Boolean formulae in prenex normal form with only existential quantifiers where the quantifier-free part is is an element of C.
- With C^+(resp. C^-) we denote the class of formulae in C whose free variables occur only positively (resp. negatively).

Counting problems and classes

Definition

A function $f\colon \{0,1\}^* \to \mathbb{N}$ is in $\#\mathrm{P}$, if there is a non-deterministic TM M such that for all inputs $x \in \{0,1\}^*$:

$$f(x) = \text{the number of acc. computation paths of } M \text{ with } x.$$

This definition can be generalized as follows.

Definition

Let $\mathcal{C}$ be a complexity class. A function $f\colon \{0,1\}^* \to \mathbb{N}$ is in $\# \cdot \mathcal{C}$, if there is a language $L \in \mathcal{C}$ and a polynomial p s.t. for all $x \in \{0,1\}^*$:

$$f(x) = |\{y \mid |y| \leq p(|x|) \text{ and } (x,y) \in L\}|.$$

Now $\#\mathrm{P} = \# \cdot \mathrm{P}$, and $\#\mathrm{P} \subseteq \# \cdot \mathrm{NP}$.

Logically defined counting classes

Definition

A function $f\colon \{0,1\}^* \to \mathbb{N}$ is in $\#\mathsf{FO}^{\mathrm{rel}}$, if there exists:

- a vocabulary σ with a built-in linear order $\leq$,
- an FO[σ]-formula $\varphi(R_1, \ldots, R_k, x_1, \ldots, x_\ell)$ with free variables $R_1, \ldots, R_k$ and $x_1, \ldots, x_\ell$,

s.t. for all σ-structures $\mathcal{A}$,

$$f(\mathrm{enc}_\sigma(\mathcal{A})) = |\{(S_1, \ldots, S_k, c_1, \ldots, c_\ell) \mid \mathcal{A} \models \varphi(S_1, \ldots, S_k, c_1, \ldots, c_\ell\}|.$$

If the input of f is not of this form, we assume f takes the value 0.

Logically defined counting classes

In the same fashion, subclasses of $\#\mathsf{FO}^{\mathrm{rel}}$, such as $\#\Sigma_k^{\mathrm{rel}}$ and $\#\Pi_k^{\mathrm{rel}}$ for arbitrary k, are defined by assuming that the formula φ in the above definition is in the corresponding fragments Σ_k and Π_k.

Theorem (Saluja et al., 95)

$\#\Sigma_0^{\mathrm{rel}} = \#\Pi_0^{\mathrm{rel}} \subset \#\Sigma_1^{\mathrm{rel}} \subset \#\Pi_1^{\mathrm{rel}} \subset \#\Sigma_2^{\mathrm{rel}} \subset \#\Pi_2^{\mathrm{rel}} = \#\mathsf{FO}^{\mathrm{rel}} = \#\mathsf{P}$.

Furthermore, it was shown that $\#\Sigma_0^{\mathrm{rel}} \subseteq \mathrm{FP}$.

A word on reductions

- Let f and h be counting problems. We say that f is *parsimoniously* reducible to h if there is a polynomial-time computable function g such for all inputs x, $f(x) = h(g(x))$,
- f is *Turing* reducible to h if $f \in \mathrm{FP}^h$,
- f is *metrically* reducible to h if there are polynomial-time computable functions g_1, g_2 such for all inputs x, $f(x) = g_2(h(g_1(x)), x)$. Note that metric reductions are thus Turing reductions with one oracle query.

For $\mathcal{F}$ a class of quantified Boolean formulae, $\#\mathcal{F}$ is the function that, given a formula $\varphi \in \mathcal{F}$, outputs the number of satisfying assignments of φ. The functions $\#\mathrm{SAT}$ and #3-CNF, are complete for #P under parsimonious reductions, while, e.g., $\#\mathrm{DNF}$ and #2-CNF$^+$ are complete for #P under metric reductions.

Example: #CNF vs. $\#\mathrm{DNF}$

We briefly explain why $\#\mathrm{DNF}$ is #P-complete by reducing #CNF to it.

Suppose a CNF-formula $\psi(p_1, ..., p_n)$ is given as an input. Now the number of satisfying assignments for ψ can be computed as follows.

1. Transform $\neg\psi$ to DNF-form and calculate the number k of satisfying assignments of it,
2. Output $2^n - k$.

Note that if this reduction could be made parsimonious, then the decision problem CNF would reduce to DNF which is unlikely.

Counting of teams

Definition

For any set A of atoms, $\#\mathrm{FO}(A)^{\mathrm{team}}$ is the class of all functions $f\colon \{0,1\}^* \to \mathbb{N}$ for which there exists,

- vocabulary σ with a built-in linear order $\leq$,
- a $\mathrm{FO}(A)$-formula $\varphi(\overline{x})$ over σ with a tuple of free first-order variables $\overline{x}$,

s.t. for all σ-structures $\mathcal{A}$,

$$f(\mathrm{enc}_\sigma(\mathcal{A})) = |\{X \mid \mathcal{A} \models_X \varphi(\overline{x})\}|,$$

or for all σ-structures $\mathcal{A}$,

$$f(\mathrm{enc}_\sigma(\mathcal{A})) = |\{X \mid X \neq \emptyset \text{ and } \mathcal{A} \models_X \varphi(\overline{x})\}|.$$

We denote by f_φ and f^*_φ the functions defined by φ in this way, respectively.

A Characterization of $\# \cdot \mathrm{NP}$

Theorem

For any set A of NP*-definable generalized atoms,* $\#\mathrm{FO}(A)^{\mathrm{team}} \subseteq \# \cdot \mathrm{NP}$.

Theorem

$\# \cdot \mathrm{NP} = \#\Sigma_1^1 = \#\mathrm{FO}(\perp)^{\mathrm{team}}$

Proof.

First note that

$$\#\Sigma_1^1 = \#\mathrm{FO}(\perp)^{\mathrm{team}}, \tag{2}$$

since any $\phi(R) \in \Sigma_1^1$ with a k-ary relation symbol R can be easily turned into a sentence $\phi'(R')$ for some $(k+1)$-ary R' such that ϕ and ϕ' define the same functions and $\phi'(R')$ is only satisfied by non-empty relations. Hence it suffices to show $\# \cdot \mathrm{NP} \subseteq \#\Sigma_1^1$ which is straightforward by Fagin's theorem.

□

The class $\#\mathrm{FO}(=(\dots))$,

Due to downward closure of $\mathrm{FO}(=(\dots))$ we cannot expect $\#\mathrm{FO}(=(\dots))$ to be equal to $\#\mathrm{P}$ or $\# \cdot \mathrm{NP}$. Still, we can show that the class contains a complete problem for $\# \cdot \mathrm{NP}$.

We first show that the function Σ_1-3CNF$^-$ is $\# \cdot \mathrm{NP}$-complete, and place it in the class $\#\mathrm{FO}(=(\dots))$.

Theorem

$\#\Sigma_1$-CNF$^-$ *is* $\# \cdot \mathrm{NP}$ *complete under Turing reductions.*

Axiomatizing $\#\Sigma_1$-CNF$^-$

In first-order logic we encode Σ_1-3CNF$^-$-formulae as structures over the vocabulary

$$\tau_{\Sigma_1\text{-3CNF}^-} = (P^1, Q^1, C_0^3, C_1^3, C_2^3, C_3^3).$$

Let $\varphi \in \Sigma_1$-3CNF$^-$. φ is encoded by the structure
$\mathcal{A} = (A, P^{\mathcal{A}}, Q^{\mathcal{A}}, C_0^{\mathcal{A}}, C_1^{\mathcal{A}}, C_2^{\mathcal{A}}, C_3^{\mathcal{A}})$ with $A = \mathrm{Vars}(\varphi)$.

- $P(x)$: x is a free variable in φ
- $Q(x)$: x is a bound variable in φ
- $C_i(x_1, x_2, x_3)$: there is $0 \leq j \leq i$ such that $P(x_k)$ for $1 \leq k \leq j$, and $\bigvee_{\ell=1}^{i} \neg x_\ell \vee \bigvee_{\ell=i+1}^{3} x_\ell$ is a clause in φ

Theorem

$\#\Sigma_1$-3CNF$^-$ is definable in Σ_1-formula $\phi(R)$ in which R appears only negatively and also in $\#\mathrm{FO}(=(\dots))^{\mathrm{team}}$.

The class $\#\mathrm{FO}(\subseteq)^{\mathrm{team}}$

Theorem

$\#\mathrm{FO}(\subseteq)^{\mathrm{team}} \subseteq \#\mathrm{P}$.

Lemma

Let σ be a vocabulary and $\varphi(\overline{x}) \in \mathrm{FO}(\subseteq)$ a formula over σ. Then the language $L := \{w \mid f^*_\varphi(w) > 0\}$ is in P.

Corollary

If $\mathrm{P} \neq \mathrm{NP}$, *then* $\#\mathrm{FO}(\subseteq)^{\mathrm{team}} \neq \#\mathrm{P}$.

Placing a #P-complete problem in $\#\mathrm{FO}(\subseteq)^{\mathrm{team}}$

Theorem

$\#2\text{-CNF}^+ \in \#\mathrm{FO}(\subseteq)^{\mathrm{team}}$

Proof.

1. Let $\varphi(x_1, \ldots, x_n) = \bigwedge D_i$, where $D_i = \ell_{i,1} \vee \ell_{i,2}$ and $\ell_{i,j} \in \{x_1, \ldots, x_n\}$,
2. $\varphi(x_1, \ldots, x_n)$ is encoded by $\mathcal{A} = (\{x_1, \ldots, x_n\}, D^{\mathcal{A}})$ with $(x, y) \in D^{\mathcal{A}}$ iff the clause $x \vee y$ occurs in φ.
3. The number of teams X with domain $\{t\}$ satisfying γ def. by

$$\mathcal{A} \models_X \forall x \forall y (\neg D(x, y) \vee x \subseteq t \vee y \subseteq t)$$

is equal to the number of satisfying assignments of φ, hence $\#2\text{-CNF}^+ = f^*_\gamma$.

□

Complete problems

For the theorems below, we restrict attention to functions counting also the empty team, i.e., the functions f^*_φ are omitted.

Theorem

$\#\Sigma_1$-CNF$^-$ *is complete (hard) for* $\#\mathrm{FO}(=(\dots))^{\mathrm{team}}$ *under parsimonious reductions.*

Theorem

$\#\Sigma_1$-DualHorn *is complete (hard) for* $\#\mathrm{FO}(\subseteq)^{\mathrm{team}}$ *under parsimonious reductions.*

Conclusion

- We determined the complexity of dependence, independence and inclusion logic.
- Same kind on analysis can be made for other logics with team semantics.
- Can smaller counting classes defined, e.g., by arithmetic circuits be logically characterized in this framework?