Second-Order Logic and Fagin's Theorem

Anna Kontou

ALMA

24 April 2023

Περιεχόμενα

Contents

Second Order Logic

Definition

Second-Order Logic extends first order logic with quantification over relations.

$$\exists X \phi,$$

where X has arity m.

$\exists X \phi$ is true in a structure $\mathcal{A}$ if, and only if, $\mathcal{A}$ can be expanded by an m-ary relation interpreting X to satisfy ϕ.

SO∃ and SO∀

Definition

- *Existential second-order (SO∃) is defined as the restriction of SO that consists of the formula of the form*

$$\exists \boldsymbol{X}_1 \ldots \exists \boldsymbol{X}_n \phi$$

where ϕ is a first-order formula.

SO∃ and SO∀

Definition

- *Existential second-order (SO∃) is defined as the restriction of SO that consists of the formula of the form*

  $$\exists \boldsymbol{X}_1 \ldots \exists \boldsymbol{X}_n \phi$$

  where ϕ is a first-order formula.
- *Universal second-order (SO∀) is defined as the restriction of SO that consists of the formula of the form*

  $$\forall \boldsymbol{X}_1 \ldots \forall \boldsymbol{X}_n \phi$$

  where ϕ is a first-order formula and the second order quantifier prefix consists only of universal quantifiers.

Examples

- *The boolean query SAT in SO∃:*

$$\Phi_{SAT} \equiv (\exists S)(\forall x)(\exists y)\Big(\big(P(x,y) \land S(y)\big) \lor \big(N(x,y) \land \neg S(y)\big)\Big)$$

Examples

- *The boolean query SAT in SO∃:*

$$\Phi_{SAT} \equiv (\exists S)(\forall x)(\exists y)\Big(\big(P(x,y) \land S(y)\big) \lor \big(N(x,y) \land \neg S(y)\big)\Big)$$

- *The boolean query 3-COLOR in SO∃:*

$$\Phi_{3\text{-}COLOR} \equiv (\exists R^1)(\exists Y^1)(\exists B^1)(\forall x)\Big[\big(R(x) \lor Y(x) \lor B(x)\big) \land (\forall y)\Big(E(x,y) \rightarrow \neg\big(R(x) \land R(y)\big) \land \neg\big(Y(x) \land Y(y)\big) \land \neg\big(B(x) \land B(y)\big)\Big)\Big]$$

Contents

Fagin's Theorem

Theorem (1973)

NP is equal to the set of existential, second order boolean queries.

$$NP = SO\exists$$

$SO\exists \subseteq NP$

Proof

- Let $\Phi = (\exists R_1^{r_1}) \dots (\exists R_k^{r_k})\psi$ a second-order existential sentence.
- Let τ be the vocabulary of Φ

Our task is to build an NP machine N such that for all $\mathcal{A} \in STRUC[\tau]$,

$$(\mathcal{A} \models \Phi) \Leftrightarrow \big(N(bin(\mathcal{A})) \downarrow\big)$$

$SO\exists \subseteq NP$

Proof

- Let $\mathcal{A}$ an input structure to N, $n = ||\mathcal{A}||$.
- N machine non-deterministically write down a binary string represantation $R_i, \forall i \in [k]$.
  Check if $(\mathcal{A}, R_1, \ldots, R_k) \models \psi$.
  If yes, then M accepts. Else, rejects.
- N accepts $\mathcal{A}$ iff there is some choice of relations R_i s.t. $(\mathcal{A}, R_1, \ldots, R_k) \models \psi$.
- By FO $\subseteq$ L $\subseteq$ P, we can test if $(\mathcal{A}, R_1, \ldots, R_k) \models \psi$ in NP.

□

$NP \subseteq SO\exists$

Notation

- $\vec{s} = (\vec{s}_1, \ldots, \vec{s}_k)$: *the numbering of tape cells,* $\vec{s} \in \{0, \ldots, n^k - 1\}$
- $\vec{t} = (\vec{t}_1, \ldots, \vec{t}_k)$: *the numbering of machine's computation time,* $\vec{t} \in \{0, \ldots, n^k - 1\}$
- $\Gamma = \{\gamma_0, \ldots, \gamma_g\} = (Q \times \Sigma) \cup \Sigma$: *a listing of the possible contents of a computation cell*

$NP \subseteq SO\exists$

Proof

Our task is to find a second-order sentence $SO\exists\Phi$ such that for a nondeterministic Turing machine N,

$$(\mathcal{A} \models \Phi) \Leftrightarrow \big(N(bin(\mathcal{A}))\downarrow\big)$$

- *Let N be a nondetermenistic Turing machine*
- *Let $n = ||\mathcal{A}||$*
- *Let $k > 0$ such that N uses $n^k - 1$ time and n^k space.*

$NP \subseteq SO\exists$

Proof

Let Φ a second-order sentence:

$$\Phi = (\exists C_1^{2k} \dots C_g^{2k} \Delta^k)\phi$$

where

- *ϕ is a first-order formula*

$NP \subseteq SO\exists$

Proof

Let Φ a second-order sentence:

$$\Phi = (\exists C_1^{2k} \dots C_g^{2k} \Delta^k)\phi$$

where

- ϕ is a first-order formula
- $C_i(\vec{s}, \vec{t})$: computation cell $\vec{s}$ at time $\vec{t}$ contains symbol γ_i

$NP \subseteq SO\exists$

Proof

Let Φ *a second-order sentence:*

$$\Phi = (\exists C_1^{2k} \dots C_g^{2k} \Delta^k)\phi$$

where

- ϕ *is a first-order formula*
- $C_i(\vec{s}, \vec{t})$*: computation cell* $\vec{s}$ *at time* $\vec{t}$ *contains symbol* γ_i
- $\Delta(\vec{t}) = \begin{cases} 1, & \text{if machine makes choice "1" at step } t+1 \\ 0, & \text{otherwise} \end{cases}$

$NP \subseteq SO\exists$

Proof

We define $\phi(\vec{C}, \Delta)$ as the conjuction of four sentences,

$$\phi \equiv \alpha \wedge \beta \wedge \eta \wedge \zeta$$

where

- α: for $\vec{t} = 0$ correctly codes input bin$(\mathcal{A})$

$$\alpha \equiv \cdots \wedge \Big(\vec{t} = 0 = s_1 = \cdots = s_{k-1} \wedge s_k \neq 0 \wedge R_1(s_k) \rightarrow C_1(\vec{s}, \vec{t})\Big)$$

$$\wedge\Big(\vec{t} = 0 = s_1 = \cdots = s_{k-1} \wedge s_k \neq 0 \wedge \neg R_1(s_k) \rightarrow C_0(\vec{s}, \vec{t})\Big) \wedge \cdots$$

NP $\subseteq$ SO$\exists$

Proof

- β: it is never the case that $C_i(\vec{s}, \vec{t})$ and $C_j(\vec{s}, \vec{t})$ both hold for $i \neq j$

$$\beta \equiv \bigwedge_{i,j} \beta_{ij}$$

where

$$\beta_{ij} = \neg\big(C_i(\vec{s}, \vec{t}) \wedge C_j(\vec{s}, \vec{t})\big) \text{ for } i \neq j$$

NP $\subseteq$ SO$\exists$

Proof

- η: *for all time $\vec{t}$, the contents of tape cell $(\vec{s}, \vec{t}+1)$ follows from the contents of cells $(\vec{s}-1, \vec{t}), (\vec{s}, \vec{t})$ and $(\vec{s}+1, \vec{t})$ via the move $\Delta(\vec{t})$ of N*

$$\eta \equiv \eta_0 \wedge \eta_1 \wedge \eta_2$$

where

$\boldsymbol{NP} \subseteq \boldsymbol{SO}\exists$

Proof

$$\eta_1 \equiv (\forall \vec{t}, \vec{s})\Bigg((\vec{t} \neq m\vec{a}x \wedge \vec{0} < \vec{s} < m\vec{a}x) \to \bigwedge_{(\alpha_{-1},\alpha_0,\alpha_1,\delta)\xrightarrow{N} b} \bigg(\Big(\delta = 1 \to \Big(\big(\Delta(\vec{t}) \wedge C_{\alpha_{-1}}(\vec{s}-1,\vec{t}) \wedge C_{\alpha_0}(\vec{s},\vec{t}) \wedge C_{\alpha_1}(\vec{s}+1,\vec{t})\big) \to C_b(\vec{s},\vec{t}+1)\Big)\Big) \wedge \Big(\big(\delta = 0 \to \big((\neg\Delta(\vec{t}) \wedge C_{\alpha_{-1}}(\vec{s}-1,\vec{t}) \wedge C_{\alpha_0}(\vec{s},\vec{t}) \wedge C_{\alpha_1}(\vec{s}+1,\vec{t})) \to C_b(\vec{s},\vec{t}+1)\big)\big)\Big)\bigg)\Bigg)$$

$NP \subseteq SO\exists$

Proof

$$\eta_1 \equiv (\forall \vec{t}, \vec{s})\Bigg((\vec{t} \neq m\vec{a}x \wedge \vec{0} < \vec{s} < m\vec{a}x) \rightarrow \bigwedge_{(\alpha_{-1},\alpha_0,\alpha_1,\delta) \xrightarrow{N} b} \Bigg(\Big(\delta = 1 \rightarrow \Big(\big(\Delta(\vec{t}) \wedge C_{\alpha_{-1}}(\vec{s}-1,\vec{t}) \wedge C_{\alpha_0}(\vec{s},\vec{t}) \wedge C_{\alpha_1}(\vec{s}+1,\vec{t}) \big) \rightarrow C_b(\vec{s},\vec{t}+1) \Big) \Big) \wedge \Big(\big(\delta = 0 \rightarrow \big((\neg\Delta(\vec{t}) \wedge C_{\alpha_{-1}}(\vec{s}-1,\vec{t}) \wedge C_{\alpha_0}(\vec{s},\vec{t}) \wedge C_{\alpha_1}(\vec{s}+1,\vec{t})) \rightarrow C_b(\vec{s},\vec{t}+1) \big) \big) \Big) \Bigg) \Bigg)$$

η_0 ***and*** η_2 ***are the same with*** η_1 ***for*** $\vec{s} = \vec{0}$ ***and*** $\vec{s} = m\vec{a}x$ ***respectively***

$NP \subseteq SO\exists$

Proof

- ζ: for $\vec{t} = n^k - 1$, machine includes the accept state
  - N accepts it, clears its tape, moves all the way to left and enters a unique accept state q_l
  - If $\gamma_l = (q_l, 0) \in \Gamma$, then

$$\zeta \equiv C_l(\vec{0}, m\vec{a}x)$$

□

Contents

NP-Complete Problems

Corollary (Cook)

SAT is NP-complete via first-order reductions.

NP-Complete Problems

Corollary (Cook)

SAT is NP-complete via first-order reductions.

Proof

- *Let $B \in NP$ be any boolean query and $\mathcal{A}$ be any input structure with $n = ||\mathcal{A}||$.*
- *By Fagin's theorem, $B = MOD[\Phi]$ where $\Phi = (\exists S_1^{\alpha_1} \dots \exists S_g^{\alpha_g})(\forall x_1 \dots \forall x_t)\psi(\vec{x})$ with ψ quantifier-free and we assume that $\psi(\vec{x}) = \bigwedge_{j=1}^{r} C_j(\vec{x})$*

$$\mathcal{A} \in B \Leftrightarrow \mathcal{A} \vDash \Phi$$

NP-Complete Problems

Proof

Define the boolean formula $\gamma(\mathcal{A})$ as follows:

- *Boolean variables:*

$$S_i(e_1, \ldots, e_{\alpha_i}) \text{ and } D(e_1, \ldots, e_k),$$

for $i = \{1, \ldots, g\}$ and $e_1, \ldots, e_{\alpha_i} \in |\mathcal{A}|$

- *Clauses:*

$$C_j(\vec{e}),\ j = 1, \ldots r$$

as $\vec{e}$ ranges over all t-tuples from $|\mathcal{A}|$.
In each $C_j(\vec{e})$, there may be some occurrences of numeric or input predicates: $\gamma(\vec{e})$. Replacing each $\gamma(\vec{e})$ by its truth value in $\mathcal{A}$.

NP-Complete Problems

Proof

It is clear from the construction that

$$\mathcal{A} \in \boldsymbol{B} \quad \Leftrightarrow \quad \mathcal{A} \models \Phi \quad \Leftrightarrow \quad \gamma(\mathcal{A}) \in \boldsymbol{SAT}$$

□

NP-Complete Problems

Corollary

3-SAT is NP-complete via first-order reductions.

NP-Complete Problems

Corollary

3-SAT is NP-complete via first-order reductions.

Proof

We show that SAT $\leq_{fo}$ 3-SAT.

- *Let $\mathcal{A} \in STRUCT[\langle P^2, n^2 \rangle]$ be an instance of SAT with $n = ||\mathcal{A}||$.*
- *Each clause c of $\mathcal{A}$ is replaced by $2n$ clauses as follows:*

$$c' \equiv ([x_1]^c \vee d_1) \wedge (\overline{d_1} \vee [x_2]^c \vee d_2) \wedge (\overline{d_2} \vee [x_3]^c \vee d_3) \wedge \cdots$$

$$\wedge(\overline{d_n} \vee [x_1]^c \vee d_{n+1}) \wedge (\overline{d_{n+1}} \vee [x_2]^c \vee d_{n+2}) \wedge \cdots \wedge (\overline{d_{2n-1}} \vee [x_n]^c)$$

NP-Complete Problems

Proof

where

- x_i's are the instance literals
- d_i's are new variables
- $[l]^c$ means the literal l if it occurs in c and false otherwise

$$c \in SAT \Leftrightarrow c' \in SAT$$

and c' is definable in a first order way from c.

□

NP-Complete Problems

Corollary

3-COLOR is NP-complete via first-order reductions.

NP-Complete Problems

Corollary

3-COLOR is NP-complete via first-order reductions.

Proof

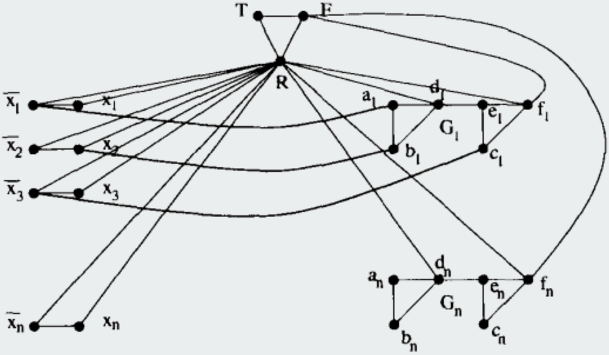
We show that 3-SAT $\leq_{fo}$ 3-COLOR.

- Let $\mathcal{A}$ be an instance of 3-SAT with $n = ||\mathcal{A}||$.
- We construct graph $f(\mathcal{A})$ such that

$$f(\mathcal{A}) \text{ 3-colorable } \Leftrightarrow \mathcal{A} \in \text{3-SAT}$$

NP-Complete Problems

Proof



□

Thanks!

References

- *Immerman, Neil. Descriptive Complexity. 1998. Springer Science & Business Media, 6 Dec. 2012.*
- *Leonid Libkin. Elements of Finite Model Theory. Berlin ; New York, Ny, Springer, 2004.*
- *Κανελλόπουλος, Σ. Γ. 2022. Εφαρμογές της Λογικής σε Αλγορίθμους Μέτρησης [Applications of Logic in Measurement Algorithms]. Εθνικό Μετσόβιο Πολυτεχνείο.*