Counting Complexity

Computation and Reasoning Laboratory

Graduate course
Spring semester 2022

Overview

1. Descriptive complexity
   - The class NP
   - The class #P

Overview

1. Descriptive complexity
   - The class NP
   - The class #P

Edge Existence

- Does $G = (V, E)$ have an edge?
- G can be seen as a structure of a first-order (**FO**) language with only one binary relation symbol, E.
- $G \models \exists x \exists y E(x, y)$.

Vertex cover of size k

Does $G = (V, E)$ have a vertex cover of size k?

$$G \models (\exists W \subseteq V)\Big[|W| \leq k \wedge (\forall x, y \in V)\big[E(x, y) \rightarrow (x \in W \vee y \in W)\big]\Big]$$

Vertex cover of size k

Does $G = (V, E)$ have a vertex cover of size k?

$$G \models (\exists W \subseteq V)\Big[|W| \leq k \wedge (\forall x, y \in V)\big[E(x, y) \rightarrow (x \in W \vee y \in W)\big]\Big]$$

We quantified over sets!

Descriptive complexity

The computational complexity of a problem can be understood as the richness of the language needed to specify the problem.

"Edge Existence" is easier than "Has a Vertex Cover of size k" since the formula $\exists x \exists y E(x, y)$ is **FO** whereas the formula

$$G \models (\exists W \subseteq V)\Big[|W| \leq k \wedge (\forall x, y \in V)\big[E(x, y) \rightarrow (x \in W \vee y \in W)\big]\Big]$$

is **∃SO**.

Finite relational structures

The input to any computational problem can be seen as a finite relational structure.

Let $\tau = \langle P^{a_1}, R^{a_2}, Q^{a_3}, \ldots \rangle$. A structure over τ looks like:

$$\mathcal{A} = \langle A, P^{\mathcal{A}}, R^{\mathcal{A}}, Q^{\mathcal{A}}, \ldots \rangle.$$

Finite relational structures

The input to any computational problem can be seen as a finite relational structure.

Let $\tau = \langle P^{a_1}, R^{a_2}, Q^{a_3}, \ldots \rangle$. A structure over τ looks like:

$$\mathcal{A} = \langle A, P^{\mathcal{A}}, R^{\mathcal{A}}, Q^{\mathcal{A}}, \ldots \rangle.$$

$$\mathrm{STRUCT}(\tau) = \{\mathcal{B} \mid \mathcal{B} \text{ is a finite structure over } \tau\}.$$

Strings as relational structures

A string with 5 characters can be seen as a relational structure:

Position	4	3	2	1	0
String	0	1	0	0	1

Vocabulary $\langle S^1, \leq^2 \rangle$

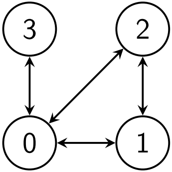
$$\mathcal{A} = \langle A, S^{\mathcal{A}}, \leq^{\mathcal{A}} \rangle, \text{ where}$$

$$A = \{0, 1, 2, 3, 4\},\ S^{\mathcal{A}} = \{0, 3\},\ \leq^{\mathcal{A}} = \{(0, 1), (0, 2), \ldots\}$$

For example,

$$\mathcal{A} \models \exists u, v \Big[\neg S(u) \wedge \neg S(v) \wedge \neg \exists w (v < w < u) \Big].$$

Graphs as relational structures



Vocabulary $\tau = \langle E^2 \rangle$

$\mathcal{G} = \langle V, E \rangle$, where

$$V = \{0, 1, 2, 3\}, E = \{(0, 1), (1, 0), \ldots\}$$

$$\mathcal{G} \models (\forall x, y)\Big[\neg E(x, x) \wedge \big(E(x, y) \leftrightarrow E(y, x)\big)\Big]$$

Propositional formulas as relational structures

A formula in conjunctive normal form.

$$\phi = (x_1 \vee x_2 \vee \neg x_3 \vee x_5) \wedge (x_4 \vee \neg x_2)$$

Vocabulary $\langle C^1, P^2, N^2 \rangle$

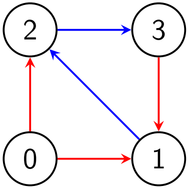
$\mathcal{A}_\phi = \langle A_\phi, C, P, N \rangle$, where

$A = \{x_1, x_2, x_3, x_4, x_5, c_1, c_2\},\ C = \{c_1, c_2\},$

$P = \{(c_1, x_1), (c_1, x_2), (c_1, x_5), (c_2, x_4)\},\ N = \{(c_1, x_3), (c_2, x_2)\}$

$$\mathcal{A}_\phi \models (\exists S)(\forall c)(\exists v)\big[C(c) \rightarrow \big(P(c,v) \wedge S(v)\big) \vee \big(N(c,v) \wedge \neg S(v)\big)\big]$$

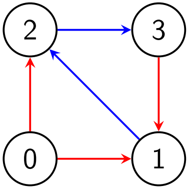
Binary Encoding of a Structure



$$\mathcal{G} = \langle V, E, R \rangle$$
$$|V| = 4$$
$$E = \{(1,2),(2,3)\}$$
$$R = \{(0,1),(0,2),(3,1)\}$$

The binary encoding of $\mathcal{G}$ is:

Binary Encoding of a Structure

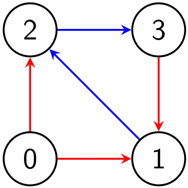


$$\mathcal{G} = \langle V, E, R \rangle$$
$$|V| = 4$$
$$E = \{(1,2),(2,3)\}$$
$$R = \{(0,1),(0,2),(3,1)\}$$

The binary encoding of $\mathcal{G}$ is:

$$\text{bin}(G) = \overbrace{0000001000010100}^{E}\overbrace{0110000000000100}^{R}.$$

Binary Encoding of a Structure



$$\mathcal{G} = \langle V, E, R \rangle$$
$$|V| = 4$$
$$E = \{(1,2),(2,3)\}$$
$$R = \{(0,1),(0,2),(3,1)\}$$

The binary encoding of $\mathcal{G}$ is:

$$\text{bin}(G) = \overbrace{0000001000010100}^{E}\overbrace{0110000000000100}^{R}.$$

It holds that $|\text{bin}(G)| = 2n^2$.

Fagin's theorem

Theorem (Fagin 1973)

∃SO captures NP: For any language L, $L \in$ NP iff it is definable by an existential second-order sentence.

Fagin's theorem

In other words, $L \in \mathrm{NP}$ if there is a formula $\phi(\overrightarrow{T})$ with relation symbols from $\overrightarrow{T} \cup \tau$ such that

$$\mathcal{A} \in L \Leftrightarrow \mathcal{A} \models \exists \overrightarrow{T} \phi(\overrightarrow{T}).$$

where $\mathcal{A}$ is an ordered finite structure over the vocabulary τ.

Fagin's theorem

In other words, $L \in \mathrm{NP}$ if there is a formula $\phi(\overrightarrow{T})$ with relation symbols from $\overrightarrow{T} \cup \tau$ such that

$$\mathcal{A} \in L \Leftrightarrow \mathcal{A} \models \exists \overrightarrow{T} \phi(\overrightarrow{T}).$$

where $\mathcal{A}$ is an ordered finite structure over the vocabulary τ.

Proof idea.
∃SO $\subseteq$ NP: For every **∃SO** formula ϕ, there is an NPTM M that nondeterministically chooses relations $\overrightarrow{S} = (S_1, ..., S_k)$ and verifies whether $\mathcal{A} \models \phi(\overrightarrow{T} / \overrightarrow{S})$ in polynomial time.

Fagin's theorem

In other words, $L \in \mathrm{NP}$ if there is a formula $\phi(\overrightarrow{T})$ with relation symbols from $\overrightarrow{T} \cup \tau$ such that

$$\mathcal{A} \in L \Leftrightarrow \mathcal{A} \models \exists \overrightarrow{T} \phi(\overrightarrow{T}).$$

where $\mathcal{A}$ is an ordered finite structure over the vocabulary τ.

Proof idea.
∃SO $\subseteq$ NP: For every **∃SO** formula ϕ, there is an NPTM M that nondeterministically chooses relations $\overrightarrow{S} = (S_1, ..., S_k)$ and verifies whether $\mathcal{A} \models \phi(\overrightarrow{T}/\overrightarrow{S})$ in polynomial time.

NP $\subseteq$ **∃SO**: The existence of an accepting computation path of an NPTM M can be expressed in **∃SO**.

$$\mathrm{bin}(\mathcal{A}) \models \exists \overrightarrow{T} \phi(\overrightarrow{T}) \text{ iff } M \text{ has an accepting path on input } \mathcal{A}$$

where $\overrightarrow{T}$ encodes the accepting computation. □