# An Isomorphism between Parameterized Complexity and Classical Complexity, for both Time and Space

Yijia Chen

Fudan University

November 6th, 2015 @ Simons Institute

# Exponential complexity

Find exponential time algorithms for NP-hard problems beating the brute-force search.

## Conjecture (ETH)

*There is no algorithm solving* $3\textsc{Sat}$ *in time* $2^{o(n)}$.

# Parameterized complexity

Find algorithms for NP-hard problems whose superpolynomial behavior is confined in some parameter.

### Conjecture (FPT $\neq$ W[1])

*There is no algorithm solving $k$-Clique in time $f(k) \cdot n^{O(1)}$, i.e., $k$-Clique is not fixed-parameter tractable.*

# The connection

**Theorem (Downey and Fellows, 1999)**
ETH *implies* FPT $\neq$ W[1].

The converse has been a major open problem, and seems hard to prove:

**Theorem (Chen et. al, 2004)**
ETH *implies that* $k$-CLIQUE *is not decidable in time* $f(k) \cdot n^{o(k)}$.

# An equivalence

**Theorem** (*C.* and Grohe, 2007)

FPT $=$ W[1] *if and only if* $k$-CLIQUE *is decidable in time*

$$2^{o(k \cdot \log n)} \cdot n^{O(1)}.$$

**Remark**

*The brute-force algorithm for* $k$-CLIQUE *has running time*

$$n^{k+O(1)} = 2^{k \cdot \log n} n^{O(1)}.$$

# Another equivalence

**Theorem (Cai and Juedes, 2003)**

ETH *fails if and only if the* *miniaturization* *of* 3SAT *is fixed-parameter tractable.*

# Anything Similar in Space Complexity?

# A central problem in classical space complexity

| | |
|---|---|
| STCONN | |
| *Input:* | A directed graph $G$ and $s, t \in V(G)$. |
| *Problem:* | Is there a path from $s$ to $t$ in $G$? |

## Theorem

1. STCONN *is complete for* NL.
2. STCONN *is decidable in space* $O(\log^2 n)$*, i.e., Savitch's Theorem.*

## Question

*Can we decide* STCONN *in space* $o(\log^2 n)$*?*

# Parameterized stConn

| $k$-stConn | |
|---|---|
| *Input:* | A directed graph $G$, $s, t \in V(G)$, and $k \in \mathbb{N}$. |
| *Parameter:* | $k$. |
| *Problem:* | Is there a path from $s$ to $t$ in $G$ of length $\leq k$? |

## Question

*Can we decide $k$-stConn in space*

$$f(k) + O(\log n)?$$

*Equivalently, is $k$-stConn in parameterized logspace?*

The brute-force algorithm decides $k$-stConn in space $k \cdot \log n + O(\log n)$.

Question

$$k\text{-}\textsc{stConn} \in \mathrm{DSPACE}\Big(o(k \cdot \log n) + O(\log n)\Big)?$$

*If so, then $k$-stConn is in parameterized logspace.*

Theorem (Savitch, 1969)

*There is an algorithm deciding $k$-stConn in space*

$$O(\log k \cdot \log n).$$

Note $O(\log k \cdot \log n) \neq o(k \cdot \log n)$ by considering fixed $k$ and $n \to \infty$.

Question

$$k\text{-}\textsc{stConn} \in \mathrm{DSPACE}\Big(o(\log k \cdot \log n) + O(\log n)\Big)?$$

# The space analogy (1)

Theorem (*C.* and Müller, 2014)

$k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(f(k) + O(\log n)\big) \Longrightarrow \textsc{stConn} \in \mathsf{DSPACE}\big(o(\log^2 n)\big)$.

*Recall* $k\text{-}\textsc{Clique} \in \mathsf{DTIME}(f(k)n^{O(1)}) \Longrightarrow 3\textsc{Sat} \in \mathsf{DTIME}(2^{o(n)})$.

Theorem (*C.* , Flum, and Müller, 2015)

$k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(f(k) + o(\log k) \cdot \log n\big) \Longrightarrow \textsc{stConn} \in \mathsf{DSPACE}\big(o(\log^2 n)\big)$.

*Recall* $k\text{-}\textsc{Clique} \in \mathsf{DTIME}(f(k)n^{o(k)}) \Longrightarrow 3\textsc{Sat} \in \mathsf{DTIME}(2^{o(n)})$.

## The space analogy (2)

**Theorem** (*C.* , Flum, and Müller, 2015)

*We have the equivalences:*

$$k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(f(k) + O(\log n)\big)$$
$$\iff k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(o(k \cdot \log n) + O(\log n)\big)$$
$$\iff k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(o(\log k \cdot \log n) + O(\log n)\big)$$

*Recall*

$k\text{-}\textsc{Clique} \in \mathsf{DTIME}(f(k)n^{o(k)}) \iff k\text{-}\textsc{Clique} \in \mathsf{DTIME}\big(2^{o(k \cdot \log n)} n^{O(1)}\big).$

## Easy direct proof for the space case

$$k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(f(k) + O(\log n)\big)$$
$$\iff k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\big(o(k \cdot \log n) + O(\log n)\big) :$$

$(\Longleftarrow)$ $o(k \cdot \log n) \leq f(k) + \log n$.

$(\Longrightarrow)$ An assumed algorithm for $k$-stConn can find a path of length at most

$$d := d(n) = f^{-1}(\log n)$$

in logspace. Then we can modify Savitch's algorithm in such a way that every time we divide the path of length at most $k_i$ into $d$ sub-paths of length at most

$$k_{i+1} := \frac{k_i}{d}.$$

Thus the total space is bounded

$$O\left(\log_d k \cdot \log n\right) = O\left(\frac{\log k}{\log d} \cdot \log n\right) = o(\log k \cdot \log n) = o(k \cdot \log n). \quad \square$$

# Unifying Proofs for

1. FPT $=$ W[1] if and only if $k$-Clique is decidable in time $2^{o(k \cdot \log n)} \cdot n^{O(1)}$.
2. ETH fails if and only if the miniaturization of $k$-Clique is fixed-parameter tractable.
3. $k$-stConn $\in \mathsf{DSPACE}\big(f(k) + O(\log n)\big)$ if and only if $k$-stConn $\in \mathsf{DSPACE}\big(o(k \cdot \log n) + O(\log n)\big)$.

# The Miniaturization Isomorphism

# Parameterization vs. Size Measure

Let $Q \subseteq \Sigma^*$ be a classical problem. A parameterization $\kappa : \Sigma^* \to \mathbb{N}$ and a size measure $\nu : \Sigma^* \to \mathbb{N}$ are both logspace computable functions.

- The parameter $\kappa(x)$ is supposed to be much smaller than $|x|$.
- The size measure $\nu(x)$ is supposed to be the length of an NP-witness of $x$.

## Example

1. $k$-Clique: $\kappa(G, k) := k$ or $\nu(G, k) := k \cdot \log n$.
2. $k$-stConn: $\kappa(G, k) := k$ or $\nu(G, k) := k \cdot \log n$.
3. 3Sat: $\nu(\alpha) := \#\text{var}(\alpha)$ or $\nu(\alpha) := \#\text{clause}(\alpha)$.

# Tractability for time complexity

| | Parameterized Complexity | Classical Complexity |
|---|---|---|
| Tractability | $(Q, \kappa) \in$ FPT | $(Q, \nu) \in$ SUBEXP |
| | i.e., $\mathrm{DTIME}\left(f(\kappa(x))\lvert x\rvert^{O(1)}\right)$ | i.e., $\mathrm{DTIME}\left(2^{o(\nu(x))}\lvert x\rvert^{O(1)}\right)$ |
| Intractability | $(Q, \kappa) \in$ XP | $(Q, \nu) \in$ EXP |
| | i.e., $\mathrm{DTIME}\left(\lvert x\rvert^{f(\kappa(x))}\right)$ | i.e., $\mathrm{DTIME}\left(2^{O(\nu(x))}\lvert x\rvert^{O(1)}\right)$ |

1. EXP: enumerate all NP-witnesses for $x$.
2. SUBEXP: avoid the enumeration.

# Reductions

| | Parameterized Complexity | Classical Complexity |
|---|---|---|
| many-one | fpt-reduction | serf-reduction |
| many-to-many | fpt Turing reduction | serf Turing reduction |

1. FPT and XP are closed under fpt- and fpt Turing reductions.
2. SUBEXP and EXP are closed under serf- and serf Turing reductions.

**Lemma (Impagliazzo, Paturi, and Zane, 2001)**

$(3\textsc{Sat}, \#\mathsf{var}(\alpha))$ *is reducible to* $(3\textsc{Sat}, \#\mathsf{clause}(\alpha))$ *by a* serf *Turing reduction.*

# Tractability for space complexity

| | Parameterized Complexity | Classical Complexity |
|---|---|---|
| Tractability | $(Q, \kappa) \in$ para-L<br>i.e., $\text{DSPACE}\Big(f(\kappa(x)) + O(\log \lvert x \rvert)\Big)$ | $(Q, \nu) \in$ SUBLIN<br>i.e., $\text{DSPACE}\Big(o(\nu(x)) + O(\log \lvert x \rvert)\Big)$ |
| Intractability | $(Q, \kappa) \in$ XL<br>i.e., $\text{DSPACE}\Big(f(\kappa(x)) \log \lvert x \rvert\Big)$ | $(Q, \nu) \in$ LIN<br>i.e., $\text{DSPACE}\Big(O(\nu(x)) + O(\log \lvert x \rvert)\Big)$ |

1. LIN: store NP-witnesses for $x$.
2. SUBLIN: avoid storing NP-witnesses for $x$.

# Reductions

| | Parameterized Complexity | Classical Complexity |
|---|---|---|
| many-one | pl-reduction | slrf-reduction |
| many-to-many | pl Turing reduction | slrf Turing reduction |

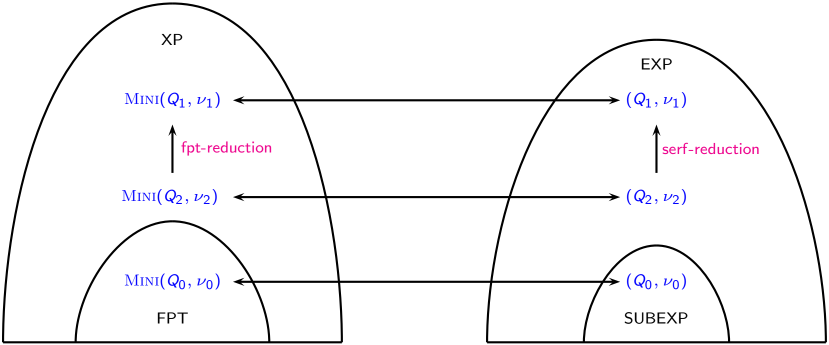
1. para-L and XL are closed under pl- and pl Turing reductions.
2. SUBLIN and LIN are closed under slrf- and slrf Turing reductions.

# The Miniaturization

Let $Q \subseteq \Sigma^*$ be a problem and $\nu$ its size measure.

| | |
|---|---|
| $\textsc{Mini}(Q, \nu)$ | |
| *Input:* | $x \in \Sigma^*$ and $m$ in unary with $m \geq \lvert x \rvert$. |
| *Parameter:* | $\left\lceil \frac{\nu(x)}{\log m} \right\rceil$. |
| *Problem:* | Decide whether $x \in Q$. |

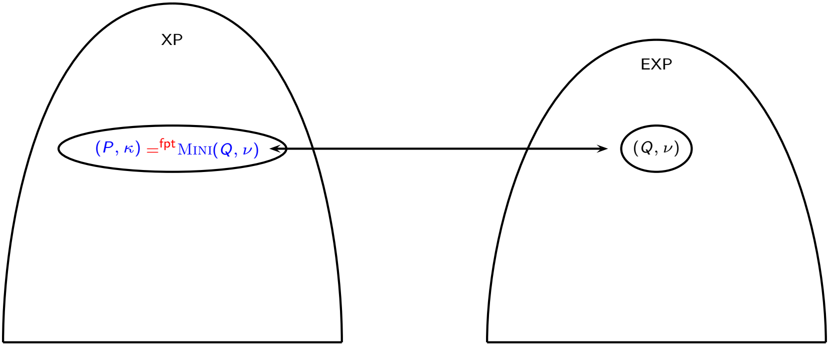
# The Isomorphism for Time Complexity (1)




## Theorem

1. $(Q, \nu) \in \mathsf{SUBEXP} \iff \textsc{Mini}(Q, \nu) \in \mathsf{FPT}$.
2. $(Q, \nu) \in \mathsf{EXP} \iff \textsc{Mini}(Q, \nu) \in \mathsf{XP}$.
3. $(Q_1, \nu_1) \le^{\mathsf{serf}} (Q_2, \nu_2) \iff \textsc{Mini}(Q_1, \nu_1) \le^{\mathsf{fpt}} \textsc{Mini}(Q_2, \nu_2)$.

# The Isomorphism for Time Complexity (2)




**Theorem**

*For any* $(P, \kappa) \in \mathrm{XP}$ *there exists a* $(Q, \nu) \in \mathrm{EXP}$ *such that*

$$(P, \kappa) =^{\mathsf{fpt}} \textsc{Mini}(Q, \nu).$$

# The Isomorphism for Time Complexity (3)

**Theorem**

*For any* $(P, \kappa) \in \mathsf{XP}$ *there exists a* $(Q, \nu) \in \mathsf{EXP}$ *such that*

$$(P, \kappa) =^{\mathsf{fpt}} \textsc{Mini}(Q, \nu).$$

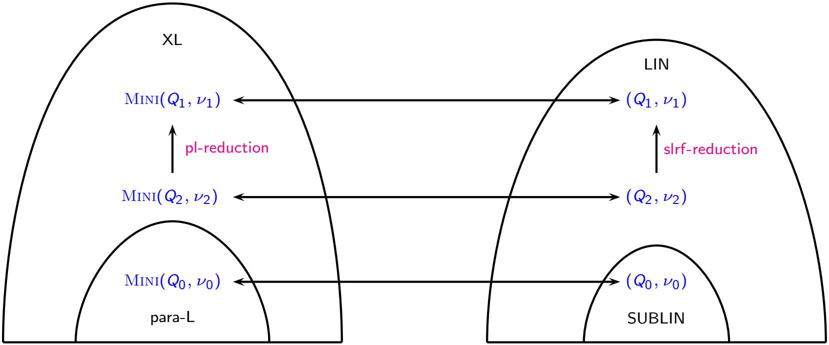
$(Q, \nu) \in \mathsf{EXP}$ constructed in the proof is artificial.

**Theorem**

$$(k\text{-}\textsc{Clique}, k) =^{\mathsf{fpt}} \textsc{Mini}\big(k\text{-}\textsc{Clique}, k \cdot \log n\big).$$

*Hence,* $k\text{-}\textsc{Clique} \in \mathsf{DTIME}\Big(f(k)n^{O(1)}\Big)$ *if and only if* $k\text{-}\textsc{Clique} \in \mathsf{DTIME}\Big(2^{o(k \cdot \log n)} n^{O(1)}\Big)$.

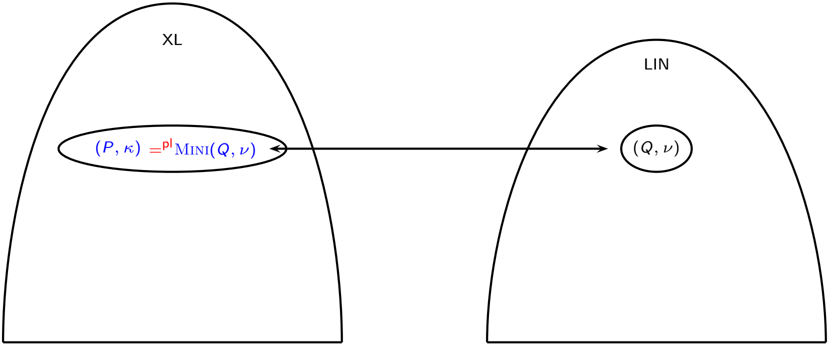
# The Isomorphism for Space Complexity (1)




## Theorem

1. $(Q, \nu) \in \mathsf{SUBLIN} \iff \textsc{Mini}(Q, \nu) \in \mathsf{para\text{-}L}$.
2. $(Q, \nu) \in \mathsf{LIN} \iff \textsc{Mini}(Q, \nu) \in \mathsf{XL}$.
3. $(Q_1, \nu_1) \leq^{\mathsf{slrf}} (Q_2, \nu_2) \iff \textsc{Mini}(Q_1, \nu_1) \leq^{\mathsf{pl}} \textsc{Mini}(Q_2, \nu_2)$.

# The Isomorphism for Space Complexity (2)




**Theorem**

*For any* $(P, \kappa) \in \mathsf{XL}$ *there exists a* $(Q, \nu) \in \mathsf{LIN}$ *such that*

$$(P, \kappa) =^{\mathsf{pl}} \textsc{Mini}(Q, \nu).$$

# The Isomorphism for Space Complexity (3)

**Theorem**

*For any* $(P, \kappa) \in \mathsf{XL}$ *there exists a* $(Q, \nu) \in \mathsf{LIN}$ *such that*

$$(P, \kappa) =^{\mathsf{pl}} \textsc{Mini}(Q, \nu).$$

**Theorem**

$$(k\text{-}\textsc{stConn}, k) =^{\mathsf{pl}} \textsc{Mini}\Big(k\text{-}\textsc{stConn}, k \cdot \log n\Big).$$

*Hence,* $k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\Big(f(k) + O(\log n)\Big)$ *if and only if* $k\text{-}\textsc{stConn} \in \mathsf{DSPACE}\Big(o(k \cdot \log n) + O(\log n)\Big)$*.*

# An application

Many tight bounds under ETH, what about $\textsc{stConn} \notin \mathsf{DSPACE}\big(o(\log^2 n)\big)$?

### Theorem (*C.* , Elberfeld, Flum, and Müller, 2015)

*For every $d \geq 2$ there is an algorithm deciding*

| | |
|---|---|
| *Input:* | A database $\mathcal{A}$ and a Boolean conjunctive query $\varphi$ with $d$ variables. |
| *Parameter:* | $\lvert\varphi\rvert$. |
| *Problem:* | Decide whether $\mathcal{A} \models \varphi$. |

*in space*

$$O(\log |\varphi| \cdot \log |\mathcal{A}|).$$

*Assume* $\textsc{stConn} \notin \mathsf{DSPACE}\big(o(\log^2 n)\big)$. *Then there is no algorithm using space*

$$f(|\varphi|) + o(\log |\varphi|) \cdot \log |\mathcal{A}|.$$

Thank You