Undirected ST-Connectivity in Log-Space

Omer Reingold

Presented by: Thang N. Dinh
CISE, University of Florida,
Fall, 2010

Undirected s-t connectivity(USTCON)

- Is *t* reachable from *s* in a undirected graph G?
- BFS, DFS take *O(n)* space
- USTCON $\in$ NL (nondeterministic log-space Turing machine)

SL Complexity class

- SL = problems log-space reducible to USTCON
- Defined by Papadimitriou in 1982 via Symmetric Turing machine
- L $\subseteq$ SL $\subseteq$ NL$\subseteq$ L^2 (SPACE($\log^2 n$))
- Problems in SL:
  - Vertex-disjoint paths: are there *k* vertex-disoint paths between *s* and *t*?
  - Is a graph a bipartite?
  - Is there a cycle containing a given edge?
  - Exclusive or 2-satisfiability
  - …

Sequential of Works

- Savitch's Theorem

  USTCON $\in L^2$
- Nisan, Szemeredi and Wigderson 1989

  USTCON $\in L^{3/2}$
- Armoni, Ta-Shma, Wigderson, and S. Zhou 2000

  USTCON $\in L^{4/3}$
- Trifonov 2005

  USTCON $\in$ SPACE(O(log n log log n))

Important Results

- Omer Reingold paper **L = SL $\subseteq$ NL**
- All problems in SL are now in L.
- Showing that a problems are not in L, by showing that such no log-space reduction from it to USTCON exist

- *Best paper award STOC 2005 - Godel's prize 2009*
- *Trifonov got Best student paper award in STOC 2005*

Outline

I. Introduction

- Space Complexity
- Connectivity is NL-complete
- Savitch's Theorem ($NL \subseteq L^2$)
- Immerman's theorem (NL=coNL)

II. Transforming to (N,D, λ)-graph

- Expander Graph
- Transforming into an expander

Space Complexity

Motivation

Complexity classes correspond to bounds on resources

One such resource is space: the number of tape cells a TM uses when solving a problem

Space Complexity Classes

For any function $f:N \to N$, we define:

$SPACE(f(n))=\{ L : L$ is decidable by a deterministic $O(f(n))$ space TM$\}$

$NSPACE(f(n))=\{ L : L$ is decidable by a non-deterministic $O(f(n))$ space TM$\}$

Low Space Classes

<u>Definitions (logarithmic space classes):</u>

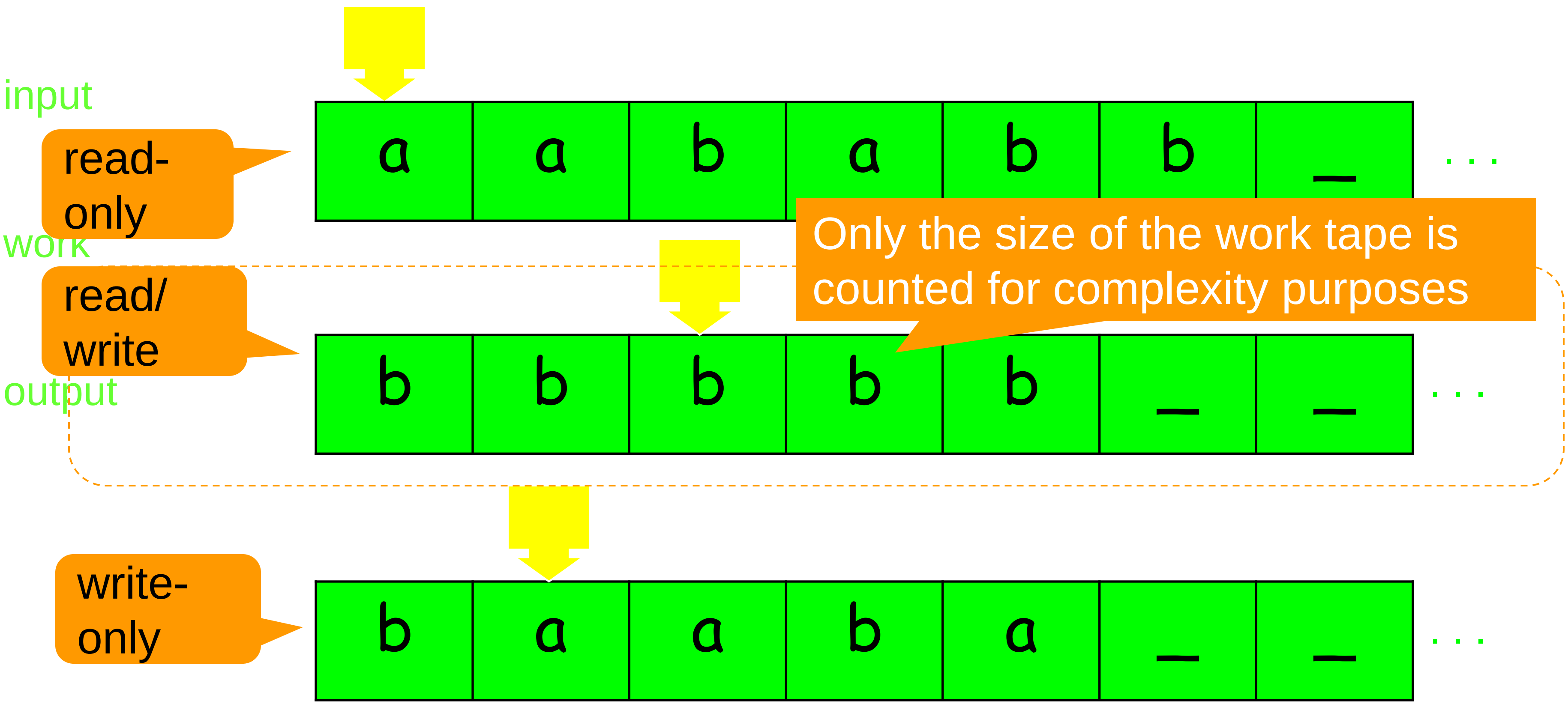
- L = SPACE(logn)
- NL = NSPACE(logn)

Problem!

How can a TM use only logn space if the input itself takes n cells?!

!?

3Tape Machines



Example

Question: How much space would a TM that decides $\{a^n b^n \mid n>0\}$ require?

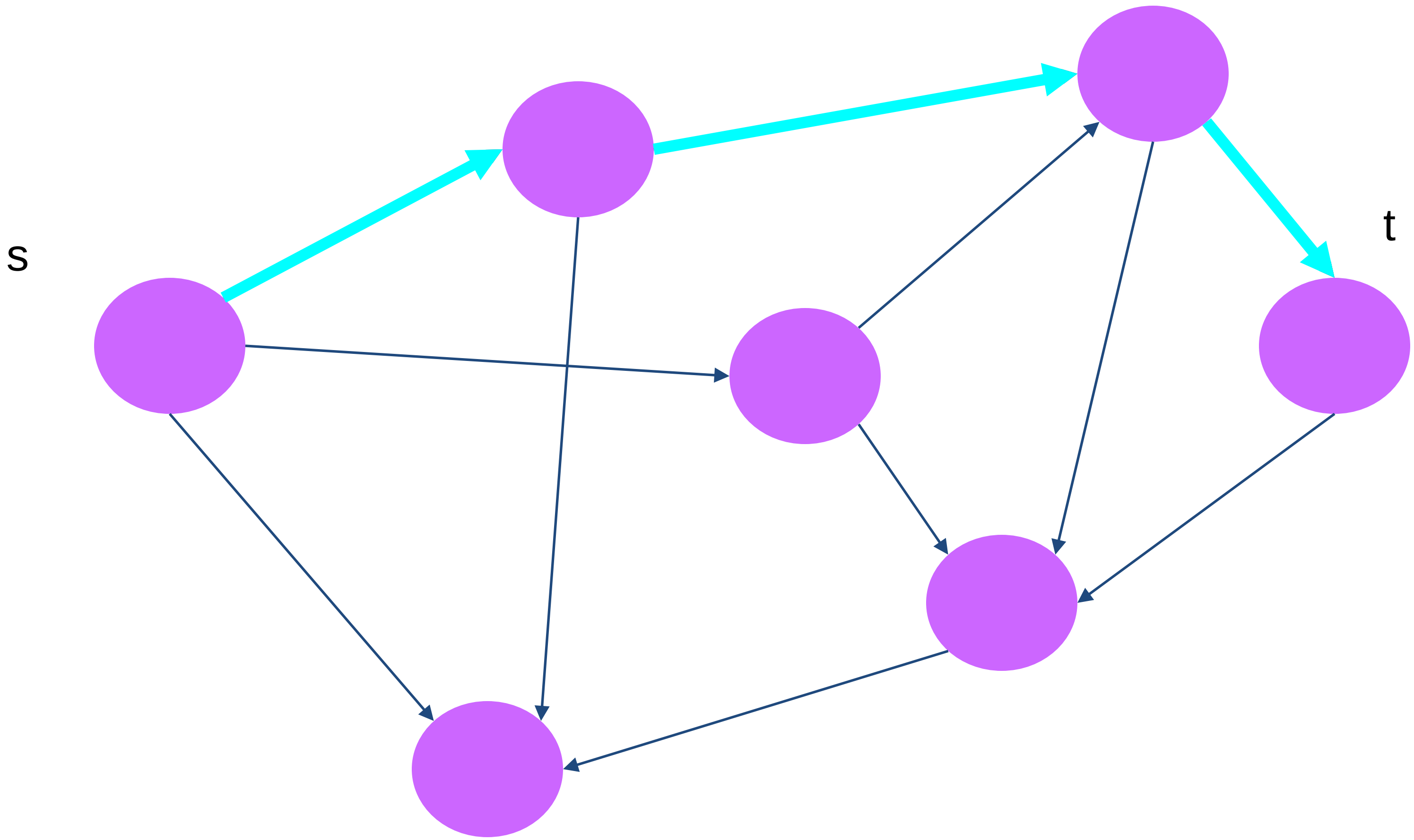
Note: to count up to n, we need $\log n$ bits

Graph Connectivity

CONN

- Instance: a *directed* graph $G=(V,E)$ and two vertices $s,t \in V$
- Problem: To decide if there is a path from s to t in G?

Graph Connectivity



CONN is in NL

- Start at s
- For $i = 1, .., |V|$ {
  - Non-deterministically choose a neighbor and jump to it
  - Accept if you get to t

  }
- If you got here – reject!

> • Counting up to $|V|$ requires $\log|V|$ space
> • Storing the current position requires $\log|V|$ space

Configurations

Which objects determine the configuration of a TM of the new type?

- The content of the work tape
- The machine's state
- The head position on the input tape
- The head position on the work tape
- The head position on the output tape

If the TM uses logarithmic space, there are polynomially many configurations

Log-Space Reductions

Definition:

A is log-space reducible to B, written $A \leq_L B$,

if there exists a log space TM M that, given input w, outputs f(w) s.t.

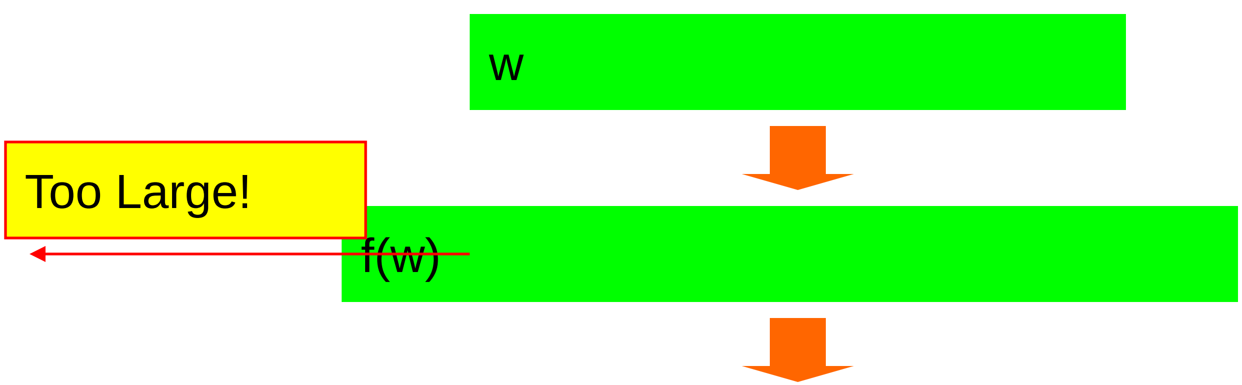
$$w \in A \text{ iff } f(w) \in B$$

the reduction

Do Log-Space Reductions Imply what they should?

Suppose $A_1 \leq_L A_2$ and $A_2 \in L$; how to construct a log space TM which decides A_1?

Wrong Solution:



Use the TM for A_2 to decide if $f(w) \in A_2$

Log-Space reductions

Claim: if

1. $A_1 \leq_L A_2$ – f is the log-space reduction
2. $A_2 \in L$ – M is a log-space machine for A_2

Then, A_1 is in L

Proof: on input x, in or not-in A_1:

Simulate M and

whenever M reads the i^{th} symbol of its input tape

run f on x and wait for the i^{th} bit to be outputted

NL Completeness

Definition:

A language B is NL-Complete if

1. $B \in NL$
2. For every $A \in NL$, $A \leq_L B$.

If (2) holds, B is NL-hard

Savitch's Theorem

Theorem:

$\forall S(n) \geq \log(n)$

$NSPACE(S(n)) \subseteq SPACE(S(n)^2)$

Proof:

First we'll prove $NL \subseteq SPACE(\log^2 n)$

then, show this implies the general case

Savitch's Theorem

Theorem:

$NSPACE(\log n) \subseteq SPACE(\log^2 n)$

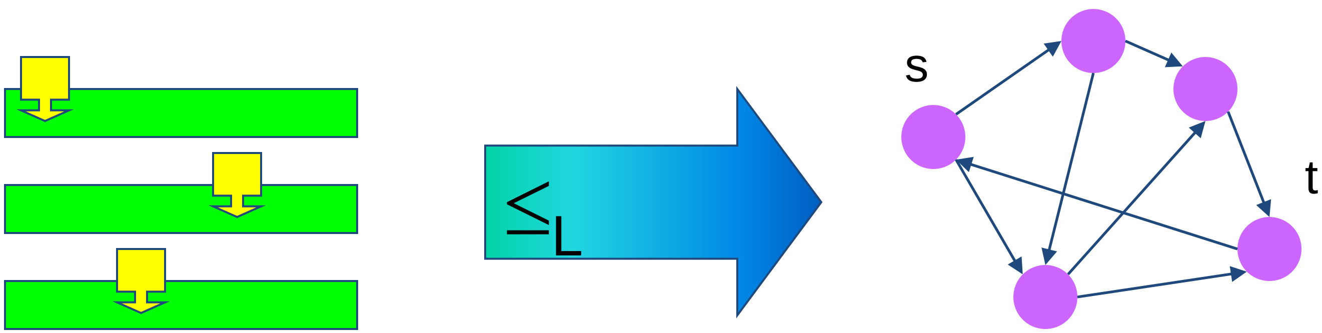
Proof:

1. First prove CONN is NL-complete (under log-space reductions)
2. Then show an algorithm for CONN that uses $\log^2 n$ space

CONN is NL-Complete

Theorem: CONN is NL-Complete

Proof: by the following reduction:

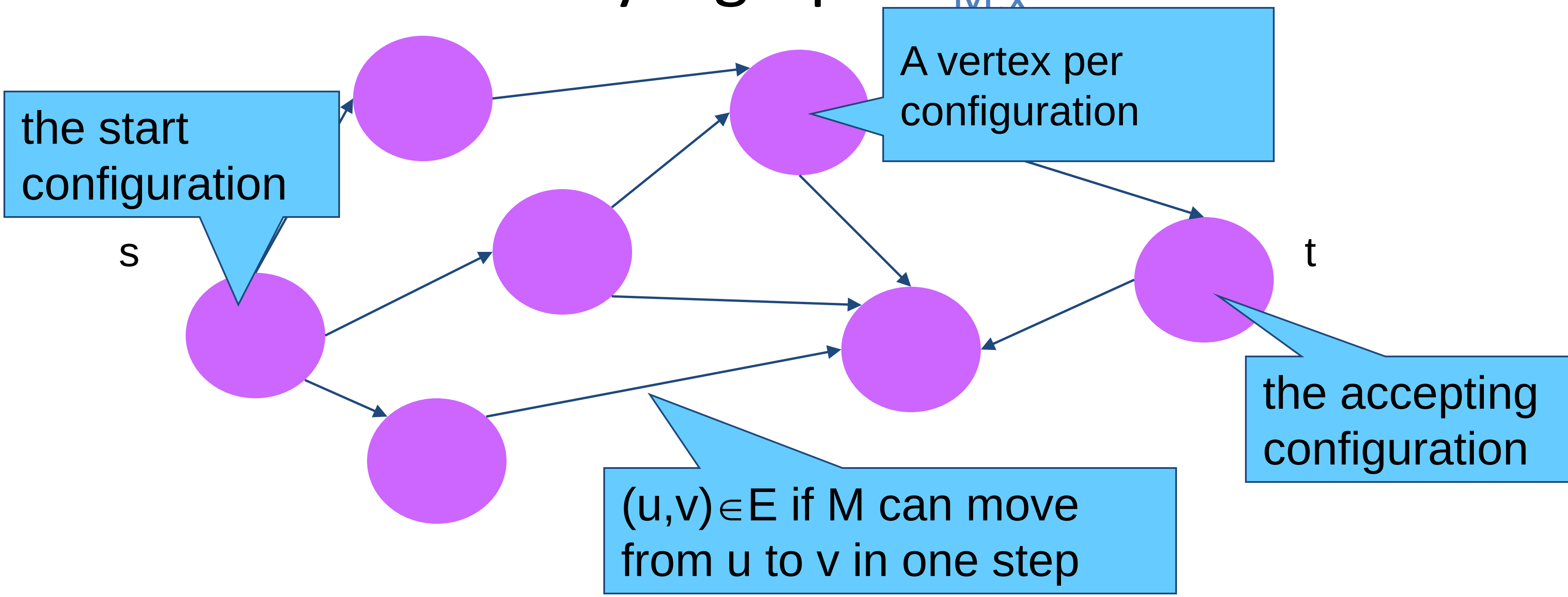


“Does M accept x?”

“Is there a path from s to t?”

Configurations Graph

A Computation of a NTM M on an input x can be described by a graph $G_{M,x}$:



CONN is NL-Complete

Corollary: CONN is NL-Complete

Proof: We've shown CONN is in NL. We've also presented a reduction from any NL language to CONN which is computable in log space (Why?) ■

A Byproduct

Claim: $NL \subseteq P$

Proof:

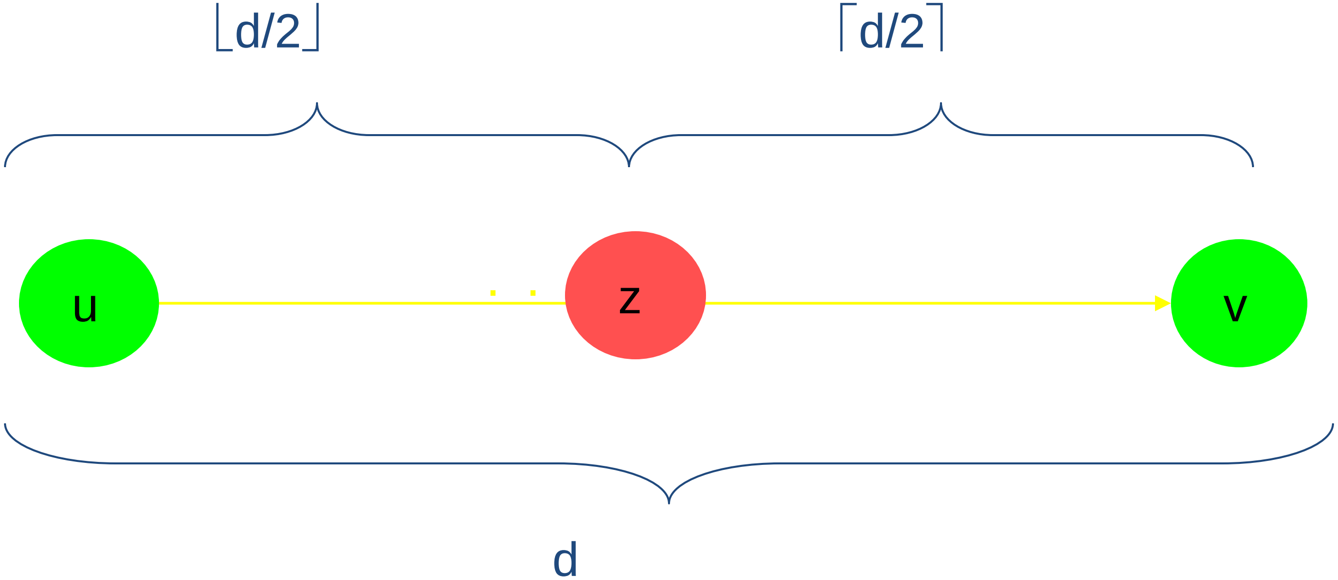
- Any NL language is log-space reducible to CONN
- Thus, any NL language is poly-time reducible to CONN
- CONN is in P
- Thus any NL language is in P. ■

What Next?

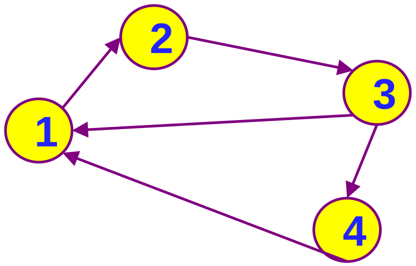
We need to show CONN can be decided by a deterministic TM in $O(\log^2 n)$ space.

The Trick

“ “Is there a vertex z, so there is a path from u to z of size $\lfloor d/2 \rfloor$ and one from z to v of size $\lceil d/2 \rceil$?”



Example of Savitch's algorithm




```
boolean PATH(a,b,d) {
    if there is an edge from a to b then
        return TRUE
    else {
        if (d=1) return FALSE
        for every vertex v (not a,b)  {
            if PATH(a,v, ⌈d/2⌉) and
                PATH(v,b, ⌊d/2⌋) then
                return TRUE
        }
        return FALSE
    }
}
```

(a,b,c)=Is there a path from a to b, that takes no more than c steps.

(1,4,3) TRUE

$3Log_2(d)$

$O(\log^2 n)$ Space DTM

Claim: There is a deterministic TM which decides CONN in $O(\log^2 n)$ space.

Proof:

To solve CONN, we invoke PATH(s,t,|V|)

The space complexity:

$S(n)=S(n/2)+O(\log n)=O(\log^2 n)$ ■

Conclusion

Theorem:

$$NSPACE(\log n) \subseteq SPACE(\log^2 n)$$

How about the general case $NSPACE(S(n)) \subseteq SPACE(S^2(n))$?

Formally

$s_i(n)$ can be computed with space $s_i(n)$

Claim: For any two space constructible functions $s_1(n), s_2(n) \geq \log n$, $f(n) \geq n$:

simulation overhead

$$NSPACE(s_1(n)) \subseteq SPACE(s_2(n))$$

$$\Downarrow$$

$$NSPACE(s_1(f(n))) \subseteq SPACE(s_2(f(n)))$$

E.g $NSPACE(n) \subseteq SPACE(n^2) \Rightarrow NSPACE(n^2) \subseteq SPACE(n^4)$

Savitch: Generalized Version

Theorem (Savitch):

$\forall S(n) \geq \log(n)$

$\text{NSPACE}(S(n)) \subseteq \text{SPACE}(S(n)^2)$

Corollary

Corollary: PSPACE = NPSPACE

Proof:

Clearly, PSPACE$\subseteq$NPSPACE.

By Savitch's theorem, NPSPACE$\subseteq$PSPACE. ■

NON-CONN

- Clearly, NON-CONN is coNL-Complete.

(Because CONN is NL-Complete. See coNP)

- If we'll show it is also in NL, then NL=coNL.

(Again, see the coNP)

An Algorithm for NON-CONN

We'll see a log space algorithm for counting reachability

1. Count how many vertices are reachable from s.
2. Take out t and count again.
3. Accept if the two numbers are the same.

Immerman's Theorem

Theorem[Immerman/Szelepcsenyi]: NL=coNL

Proof:

(1) NON-CONN is NL-Complete

(2) NON-CONN $\in$ NL

Hence, NL=coNL. ■

Corollary

Corollary:

$\forall s(n) \geq \log(n)$, NSPACE(s(n))=coNSPACE(s(n))

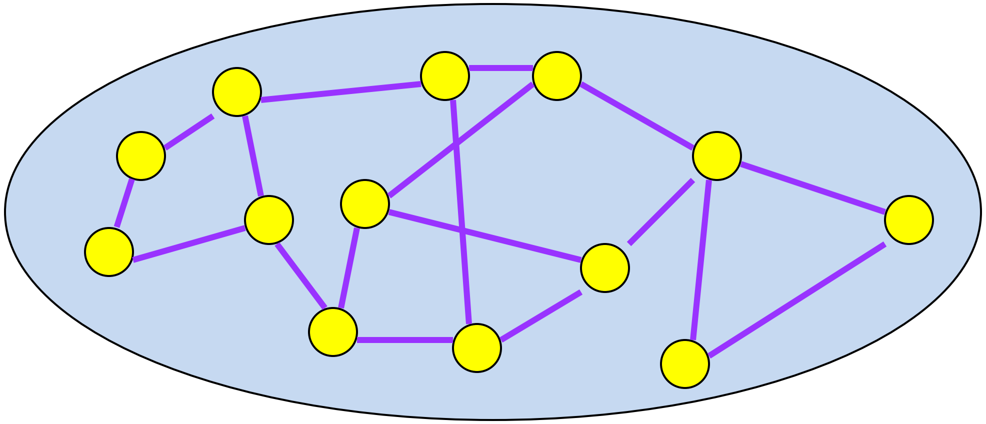
Summary

- Connectivity is NL-complete
- Savitch's Theorem (USTCONN $\in$NL $\subseteq$ L^2)
- Immerman's theorem (NL=coNL)

USTCONN is in log-space

- Low diameter property of Expander
- Connectivity Amplification
  - Powering
  - Zig-Zag product
- Log-space algorithm for USTCONN

Expander



- **Combinatorial:** no small cuts, high connectivity

- **Probabilistic:** rapid convergence of random walk

- **Algebraic:** small second eigenvalue

Theorem. [Cheeger, Buser, Tanner, Alon-Milman, Alon, Jerrum-Sinclair,…]:
All properties are equivalent!

Expander Graph

- D-regular graphs : Every vertex has degree D
- A (N, D, λ) is a D-regular graph of N vertices where $\lambda(G) \leq \lambda$
- $G = (N, D, \lambda)$ is an expander iff the spectral gap $1 - \lambda > 0$
- $G = (N, D, \lambda)$ is an expander if there exists $\varepsilon > 0$ such that for any set S, $|S| < \frac{1}{2} N$, at least $(1+\varepsilon)|S|$ vertices of G are connected to some vertex in S

Normalized Adjacency Matrix M^n

- G: D-regular undirected graph

$$M^n(v_i, v_j) = M(v_i, v_j)/D$$

M (D=3)

	v_1	v_2	v_3	v_4
v_1	1	0	2	0
v_2	0	1	1	1
v_3	2	1	0	0
v_4	0	1	0	2

M^n

	v_1	v_2	v_3	v_4
v_1	1/3	0	2/3	0
v_2	0	1/3	1/3	1/3
v_3	2/3	1/3	0	0
v_4	0	1/3	0	2/3

(N,D,λ)-graph

- $1_N=(1,\ldots,1)$ is an eigenvector of M^n with eigenvalue 1 since $M^n * 1_N = 1 * 1_N$
- $|\lambda| <= 1$ for any other eigenvalue λ of M^n
  - $\lambda(G)$: the second largest eigenvalue of M^n
  - (N,D,λ)-graph: a D-regular graph G on N vertices such that $\lambda(G) <= \lambda$.

Diameter of a (N,D, λ)-graph

Lemma: Diameter of a (N,D, λ) is bounded by O(log N)

Proof:

- Pick any vertex s, let l=O(logN). Then at least $(1+\varepsilon)^l >= N/2$ vertices are at distance at most l to s.
- Pick any two vertices s and t, then at least one vertex is of distance at most l from both s and t →a path of length at most 2 l between any two vertices.

ST-Connectivity for (N,D, λ)-graph

- Can be determined in space O(Log D*logN)
- Enumerate all paths from s of length O(log N)
- Memory: log D for remembering an edge in the path, and at most O(log N) edges for a path.

Rotation Map

- For D-regular undirected graph G,

  Rot_G: [N]x[D]→[N]x[D]

	v_1	v_2	v_3	v_4
v_1	1	0	*1*	1
v_2	0	0	**1**	0
v_3	*1*	**1**	0	1
v_4	1	0	1	1

Rot_G(v1,2)=(v3,1)

Rot_G(v2,1)=(v3,2)

Sketch of the algorithm

- Transform input graph into a D-regular non-bipartite graph (How?)
- Amplify the connectivity without increasing too much the degree

Transforming G to an (N,D, λ)-graph

- Idea: increase the connectivity of G by powering G
- Challenge: keeping degree be constant by using zig-zag expander.

Powering

- G: an (N,D,λ)-graph G by rotation map Rot_G. The t'th power G^t of G is:

$$Rot_G(v_0,(a_1,\ldots,a_t))=(v_t,(b_1,\ldots b_t)).$$

i.e., there is path v_0-a_1-b_1-v_1-a_2…v_{t-1}-a_t-b_t-v_t.

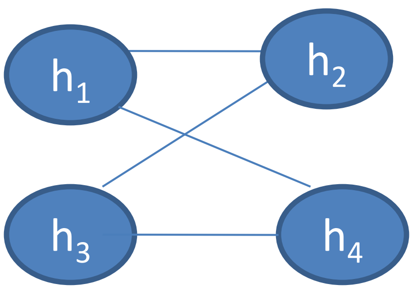
G: an (N,D,λ)-graph ⇒ G^t:an (N,D^t,λ^t)-graph

In rotation map notation, this means that

[illegible]

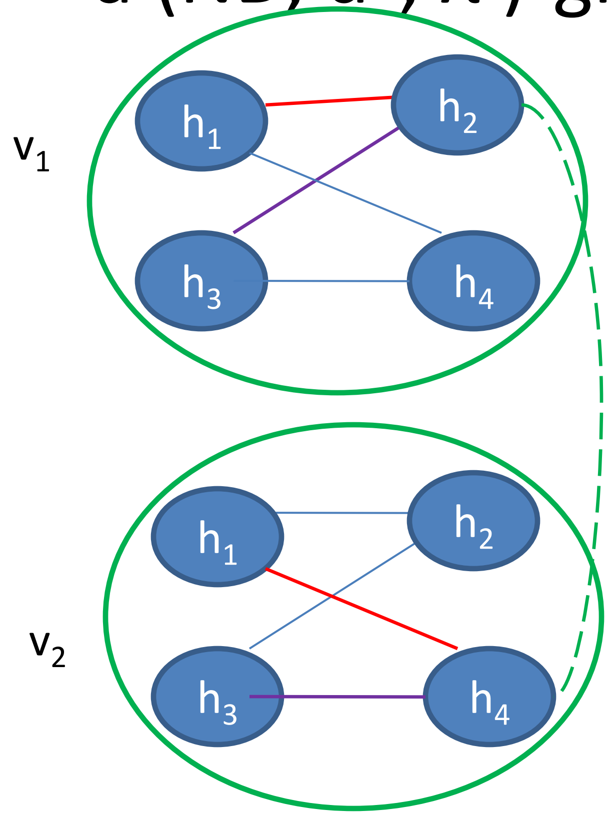
Zig-zag Graph Product

H: an (D, d, α)-graph

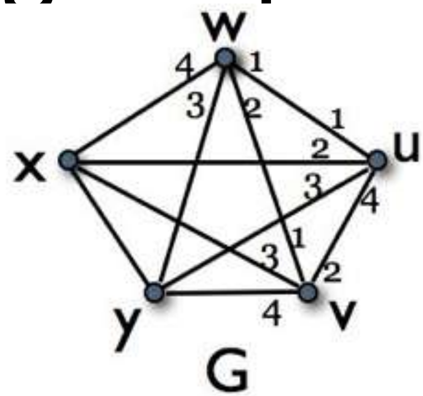


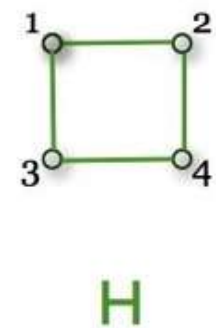
G: an (N, D, λ)-graph

a (ND, d^2, λ')-graph



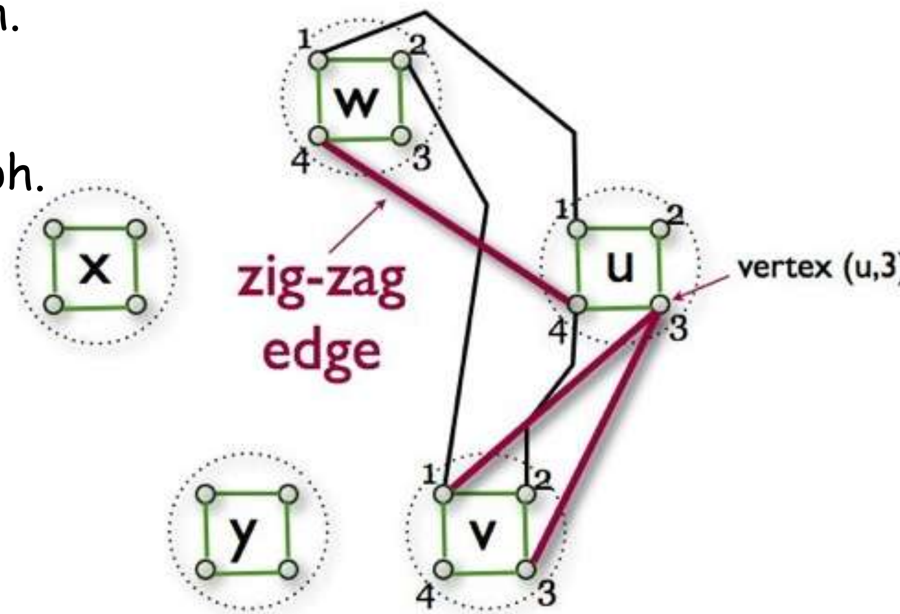
Zig-Zag Graph Product





G an (n,m,α)-graph. H an (m,d,β) -graph.

Theorem. $G \textcircled{z} H$ is an $(nm,d^2,\alpha+\beta)$-graph.



Main Transformation

- Input: H: a $(D^{16},D,1/2)$-graph and
  G: a (N,D^{16},λ)-graph where $\lambda \leq 1-1/(DN^2)$

- Processing: for i=1 to $l=O(\log N)$ do
  $$G_i=(G_{i-1} \textcircled{z} H)^8$$
- Output: G_l=: a $(N^{poly}, D^{16}, 1/2)$-graph

USTCONN is in log-space

- Theorem 1. USTCONN $\in$ L
- Theorem 2. SL = L

Questions

Thanks you for listening!