The Complexity of Finding Paths in Tournaments

Till Tantau

International Computer Schience Institute
Berkeley, California

January 30th, 2004

Outline

Outline

1. Introduction
   - What are Tournaments?
   - What Does “Finding Paths” Mean?
2. Review
   - Standard Complexity Classes
   - Standard Complexity Results on Finding Paths
3. Finding Paths in Tournaments
   - Complexity of: Does a Path Exist?
   - Complexity of: Construct a Path
   - Complexity of: Construct a Shortest Path

Outline

Outline

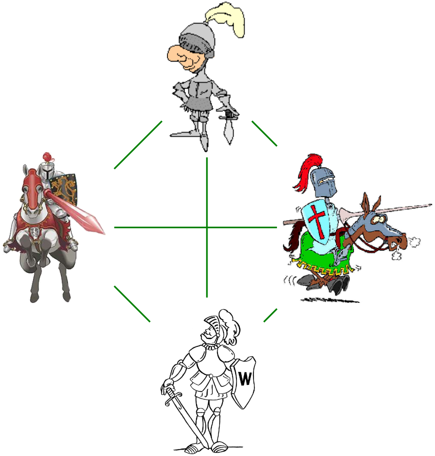
1. Introduction
   - What are Tournaments?
   - What Does “Finding Paths” Mean?
2. Review
   - Standard Complexity Classes
   - Standard Complexity Results on Finding Paths
3. Finding Paths in Tournaments
   - Complexity of: Does a Path Exist?
   - Complexity of: Construct a Path
   - Complexity of: Construct a Shortest Path

Tournaments Consist of Jousts Between Knights

What is a Tournament?

- A group of knights.
- Every two knights have a joust.
- In every joust one knight wins.

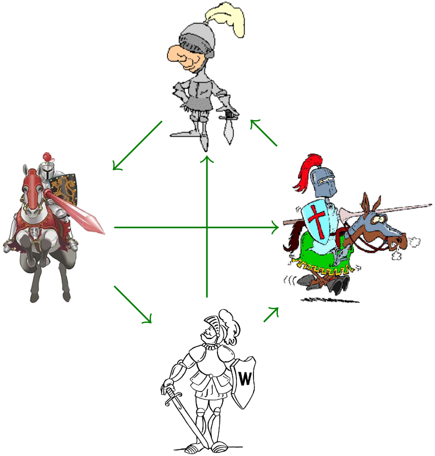
Tournaments Consist of Jousts Between Knights



What is a Tournament?

- A group of knights.
- Every two knights have a joust.
- In every joust one knight wins.

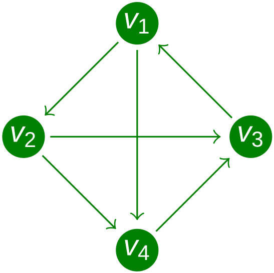
Tournaments Consist of Jousts Between Knights



What is a Tournament?

- A group of knights.
- Every two knights have a joust.
- In every joust one knight wins.

Tournaments are Complete Directed Graphs



Definition

A tournament is a

- directed graph,
- with exactly one edge between any two different vertices,
- without self-loops.

Tournaments Arise Naturally in Different Situations

Applicatins in Ordering Theory

Elements in a set need to be sorted.
The comparison relation may be cyclic, however.

Applicatins in Sociology

Several candidates apply for a position.
Reviewers decide for any two candidates whom they prefer.

AStructural Complexity Theory

A language L is given and a selector function f.
It chooses from any two words the one more likely to be in f.

Tournaments Arise Naturally in Different Situations

Applicatins in Ordering Theory

Elements in a set need to be sorted.
The comparison relation may be cyclic, however.

Applicatins in Sociology

Several candidates apply for a position.
Reviewers decide for any two candidates whom they prefer.

AStructural Complexity Theory

A language L is given and a selector function f.
It chooses from any two words the one more likely to be in f.

Tournaments Arise Naturally in Different Situations

Applicatins in Ordering Theory

Elements in a set need to be sorted.
The comparison relation may be cyclic, however.

Applicatins in Sociology

Several candidates apply for a position.
Reviewers decide for any two candidates whom they prefer.

AStructural Complexity Theory

A language L is given and a selector function f.
It chooses from any two words the one more likely to be in f.

Outline

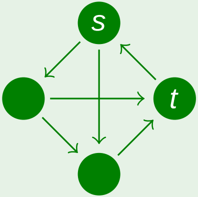
1. Introduction
   - What are Tournaments?
   - What Does “Finding Paths” Mean?
2. Review
   - Standard Complexity Classes
   - Standard Complexity Results on Finding Paths
3. Finding Paths in Tournaments
   - Complexity of: Does a Path Exist?
   - Complexity of: Construct a Path
   - Complexity of: Construct a Shortest Path

“Finding Paths” is Ambiguous

Input for Path Finding Problems

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Example Input



“Finding Paths” is Ambiguous

Input for REACH

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Variants of Path Finding Problems

Reachability Problem: Is there a path from s to t?

Construction Problem: Construct a path from s to t?

Optimization Problem: Construct a shortest path from s to t.

Distance Problem: Is the distance of s and t at most d?

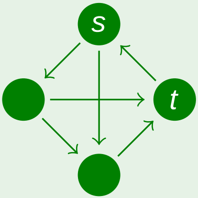
Approximation Problem: Construct a path from s to t of length approximately their distance.

"Finding Paths" is Ambiguous

Input for REACH

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Example Input



Example Output

"Yes"

“Finding Paths” is Ambiguous

Input for the Construction Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Variants of Path Finding Problems

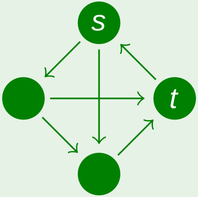
Reachability Problem:	Is there a path from s to t?
Construction Problem:	Construct a path from s to t?
Optimization Problem:	Construct a shortest path from s to t.
Distance Problem:	Is the distance of s and t at most d?
Approximation Problem:	Construct a path from s to t of length approximately their distance.

“Finding Paths” is Ambiguous

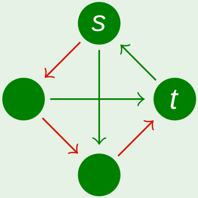
Input for the Construction Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Example Input



Example Output



“Finding Paths” is Ambiguous

Input for the Optimization Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Variants of Path Finding Problems

Reachability Problem: Is there a path from s to t?

Construction Problem: Construct a path from s to t?

Optimization Problem: Construct a shortest path from s to t.

Distance Problem: Is the distance of s and t at most d?

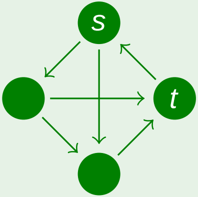
Approximation Problem: Construct a path from s to t of length approximately their distance.

“Finding Paths” is Ambiguous

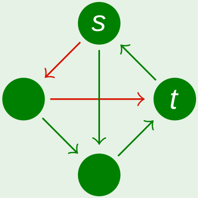
Input for the Optimization Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.

Example Input



Example Output



“Finding Paths” is Ambiguous

Input for DISTANCE

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.
- A maximum distance d.

Variants of Path Finding Problems

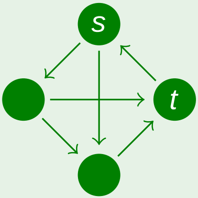
Reachability Problem:	Is there a path from s to t?
Construction Problem:	Construct a path from s to t?
Optimization Problem:	Construct a shortest path from s to t.
Distance Problem:	Is the distance of s and t at most d?
Approximation Problem:	Construct a path from s to t of length approximately their distance.

“Finding Paths” is Ambiguous

Input for DISTANCE

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.
- A maximum distance d.

Example Input



$, d = 2$

Example Output

“Yes”

“Finding Paths” is Ambiguous

Input for the Approximation Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.
- An approximation ratio $r > 1$.

Variants of Path Finding Problems

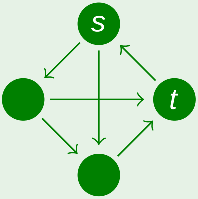
Reachability Problem:	Is there a path from s to t?
Construction Problem:	Construct a path from s to t?
Optimization Problem:	Construct a shortest path from s to t.
Distance Problem:	Is the distance of s and t at most d?
Approximation Problem:	Construct a path from s to t of length approximately their distance.

“Finding Paths” is Ambiguous

Input for the Approximation Problem

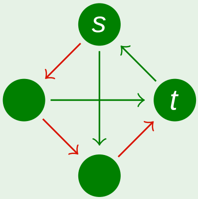
- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.
- An approximation ratio $r > 1$.

Example Input



$, r = 1.5$

Example Output

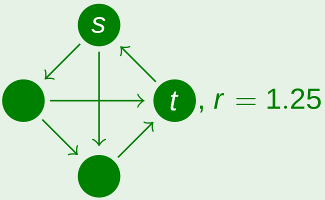


“Finding Paths” is Ambiguous

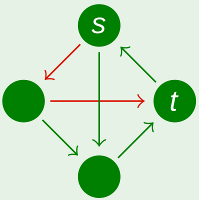
Input for the Approximation Problem

- A graph $G = (V, E)$, a source $s \in V$ and a target $t \in V$.
- An approximation ratio $r > 1$.

Example Input



Example Output



Outline

The Classes L and NL are Defined via Logspace Turing Machines

input tape (read only), n symbols

`3401234*3143223=`

work tape (read/write), $O(\log n)$ symbols

`42`

`10690836937182`

output tape (write only)

The Classes L and NL are Defined via Logspace Turing Machines

input tape (read only), *n* symbols

3401234*3143223=

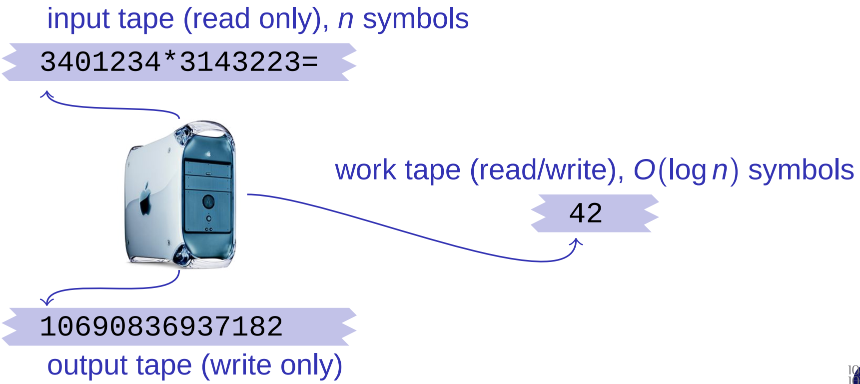
work tape (read/write), $O(\log n)$ symbols

42

10690836937182

output tape (write only)

The Classes L and NL are Defined via Logspace Turing Machines



Logspace Turing Machines Are Quite Powerful

Deterministic logspace machines can compute

- addition, multiplication, and even division
- reductions used in completeness proofs,
- reachability in forests.

Non-deterministic logspace machines can compute

- reachability in graphs,
- non-reachability in graphs,
- satisfiability with two literals per clause.

Logspace Turing Machines Are Quite Powerful

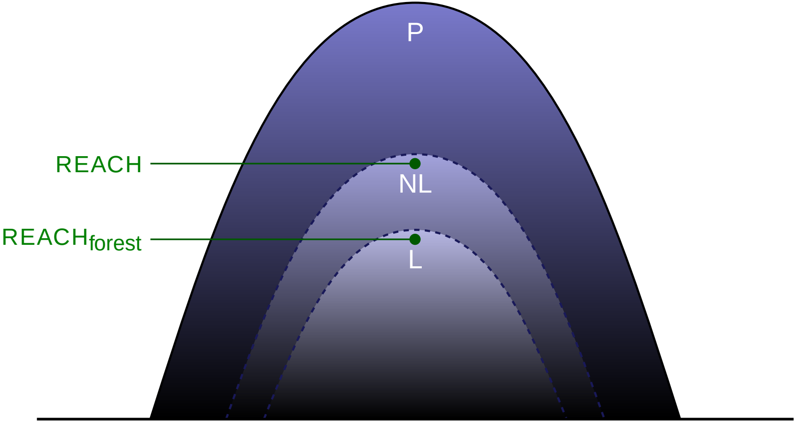
Deterministic logspace machines can compute

- addition, multiplication, and even division
- reductions used in completeness proofs,
- reachability in forests.

Non-deterministic logspace machines can compute

- reachability in graphs,
- non-reachability in graphs,
- satisfiability with two literals per clause.

The Complexity Class Hierarchy



The Circuit Complexity Classes AC^0, NC^1, and NC^2 Limit the Circuit Depth

Circuit Class AC^0

- $O(1)$ depth
- unbounded fan-in

Examples

- ADDITION $\in AC^0$.
- PARITY $\notin AC^0$.

Circuit Class NC^1

- $O(\log n)$ depth
- bounded fan-in

Examples

- PARITY $\in NC^1$.
- MUTIPLY $\in NC^1$.
- DIVIDE $\in NC^1$.

Circuit Class NC^2

- $O(\log^2 n)$ depth
- bounded fan-in

Examples

- NL $\subseteq NC^2$.

The Circuit Complexity Classes AC^0, NC^1, and NC^2 Limit the Circuit Depth

Circuit Class AC^0

- $O(1)$ depth
- unbounded fan-in

Examples

- $\text{ADDITION} \in AC^0$.
- $\text{PARITY} \notin AC^0$.

Circuit Class NC^1

- $O(\log n)$ depth
- bounded fan-in

Examples

- $\text{PARITY} \in NC^1$.
- $\text{MUTIPLY} \in NC^1$.
- $\text{DIVIDE} \in NC^1$.

Circuit Class NC^2

- $O(\log^2 n)$ depth
- bounded fan-in

Examples

- $NL \subseteq NC^2$.

The Circuit Complexity Classes AC^0, NC^1, and NC^2 Limit the Circuit Depth

Circuit Class AC^0

- $O(1)$ depth
- unbounded fan-in

Examples

- $\text{ADDITION} \in AC^0$.
- $\text{PARITY} \notin AC^0$.

Circuit Class NC^1

- $O(\log n)$ depth
- bounded fan-in

Examples

- $\text{PARITY} \in NC^1$.
- $\text{MUTIPLY} \in NC^1$.
- $\text{DIVIDE} \in NC^1$.

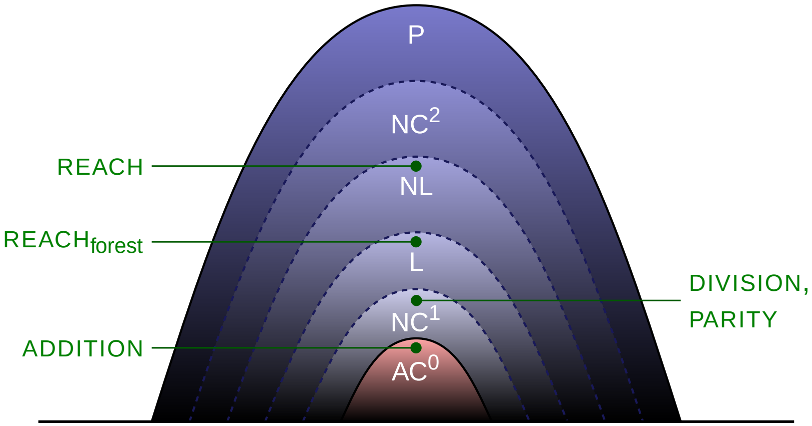
Circuit Class NC^2

- $O(\log^2 n)$ depth
- bounded fan-in

Examples

- $NL \subseteq NC^2$.

The Complexity Class Hierarchy



Outline

1. Introduction
   - What are Tournaments?
   - What Does "Finding Paths" Mean?
2. Review
   - Standard Complexity Classes
   - Standard Complexity Results on Finding Paths
3. Finding Paths in Tournaments
   - Complexity of: Does a Path Exist?
   - Complexity of: Construct a Path
   - Complexity of: Construct a Shortest Path

All Variants of Finding Paths in Directed Graphs Are Equally Difficult

Fact

REACH and DISTANCE are NL-complete.

Corollary

For directed graphs, we can solve

- the reachability problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the construction problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the optimization problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the approximation problem in logspace iff $\mathrm{L} = \mathrm{NL}$.

All Variants of Finding Paths in Directed Graphs Are Equally Difficult

Fact

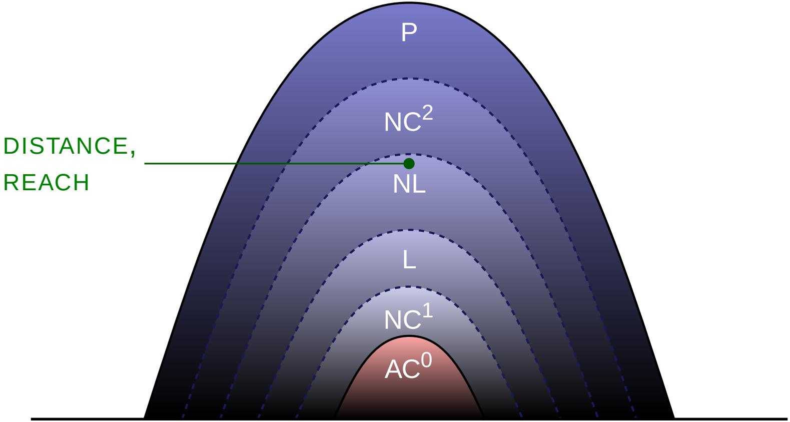
REACH and DISTANCE are NL-complete.

Corollary

For directed graphs, we can solve

- the reachability problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the construction problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the optimization problem in logspace iff $\mathrm{L} = \mathrm{NL}$.
- the approximation problem in logspace iff $\mathrm{L} = \mathrm{NL}$.

The Complexity Class Hierarchy



FindingPaths in Forests and Directed Paths is Easy, But Not Trivial

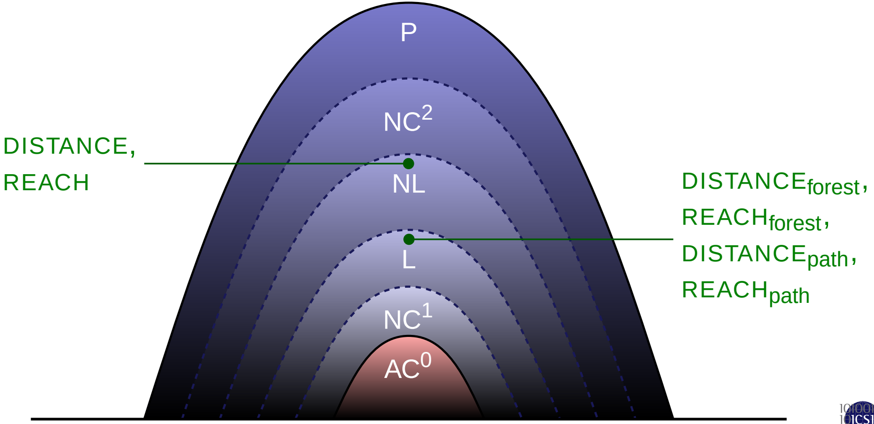
Fact

$\text{REACH}_{\text{forest}}$ and $\text{DISTANCE}_{\text{forest}}$ are L-complete.

Fact

$\text{REACH}_{\text{path}}$ and $\text{DISTANCE}_{\text{path}}$ are L-complete.

The Complexity Class Hierarchy



Outline

1. Introduction
   - What are Tournaments?
   - What Does "Finding Paths" Mean?
2. Review
   - Standard Complexity Classes
   - Standard Complexity Results on Finding Paths
3. Finding Paths in Tournaments
   - Complexity of: Does a Path Exist?
   - Complexity of: Construct a Path
   - Complexity of: Construct a Shortest Path

Definition of the Tournament Reachability Problem

Definition

Let $\text{REACH}_{\text{tourn}}$ contain all triples (T, s, t) such that

- $T = (V, E)$ is a tournament and
- there exists a path from s to t.

The Tournament Reachability Problem is Very Easy

Theorem

$\text{REACH}_{\text{tourn}} \in \text{AC}^0$.

Implications

- The problem is "easier" than REACH and even $\text{REACH}_{\text{path}}$.
- $\text{REACH} \not\leq_{m}^{\text{AC}^0} \text{REACH}_{\text{tourn}}$.

The Tournament Reachability Problem is Very Easy

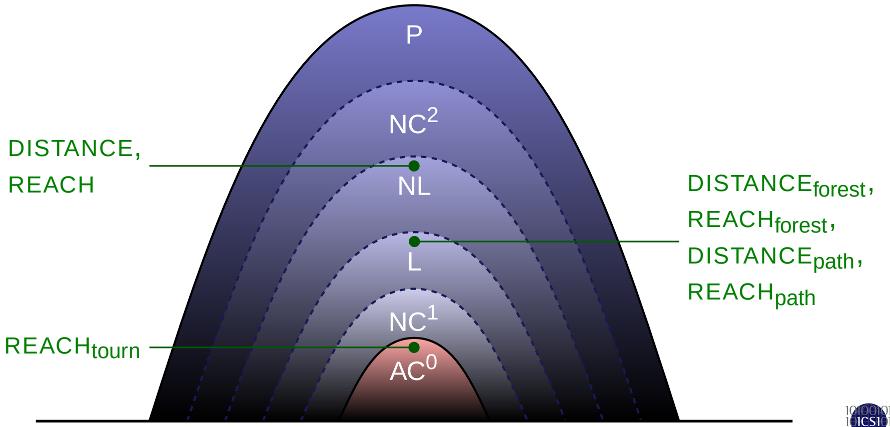
Theorem

$\text{REACH}_{\text{tourn}} \in \text{AC}^0$.

Implications

- The problem is “easier” than REACH and even $\text{REACH}_{\text{path}}$.
- $\text{REACH} \not\leq_{\text{m}}^{\text{AC}^0} \text{REACH}_{\text{tourn}}$.

The Complexity Class Hierarchy



Outline

Finding a Shortest Path Is as Difficult as the Distance Problem

Definition

Let $\text{DISTANCE}_{\text{tourn}}$ contain all tuples (T, s, t, d) such that

- $T = (V, E)$ is a tournament in which
- the distance of s and t is at most d.

The Tournament Distance Problem is Hard

Theorem

$\text{DISTANCE}_{\text{tourn}}$ is NL-complete.

Skip Proof

Corollary

Shortest path in tournaments can be constructed in logarithmic space, iff $L = NL$.

Corollary

$\text{DISTANCE} \leq_m^{AC^0} \text{DISTANCE}_{\text{tourn}}$.

The Tournament Distance Problem is Hard

Theorem

$\text{DISTANCE}_{\text{tourn}}$ is NL-complete.

Skip Proof

Corollary

Shortest path in tournaments can be constructed in logarithmic space, iff $\text{L} = \text{NL}$.

Corollary

$\text{DISTANCE} \leq_m^{\text{AC}^0} \text{DISTANCE}_{\text{tourn}}$.

The Tournament Distance Problem is Hard

Theorem

$\text{DISTANCE}_{\text{tourn}}$ is NL-complete.

Skip Proof

Corollary

Shortest path in tournaments can be constructed in logarithmic space, iff $\text{L} = \text{NL}$.

Corollary

$\text{DISTANCE} \leq_{\text{m}}^{\text{AC}^0} \text{DISTANCE}_{\text{tourn}}$.

Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

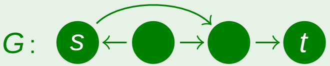
Reduce REACH to DISTANCE$_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in$ DISTANCE$_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

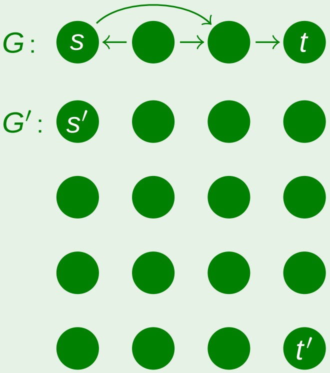
Reduce REACH to DISTANCE$_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in$ DISTANCE$_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

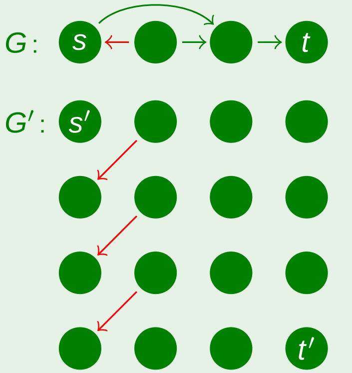
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

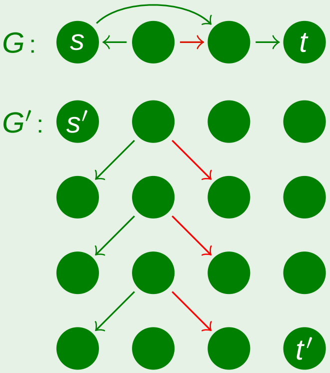
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query:
  $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

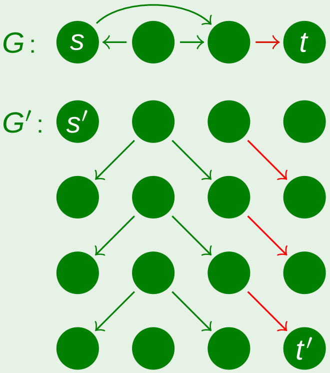
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

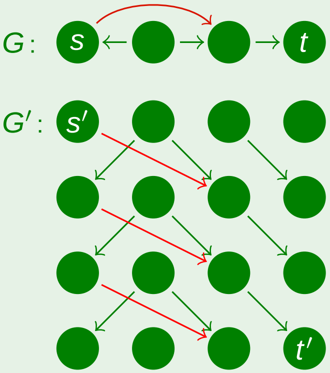
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That DISTANCE$_{\text{tourn}}$ is NL-complete

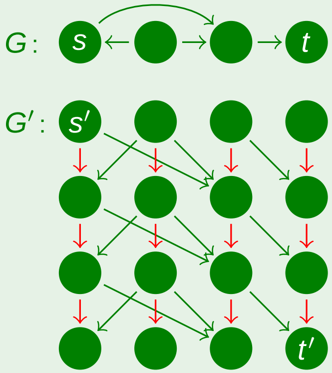
Reduce REACH to DISTANCE$_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in$ DISTANCE$_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\textsc{Distance}_{\text{tourn}}$ is NL-complete

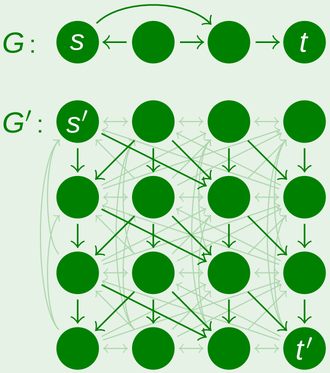
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

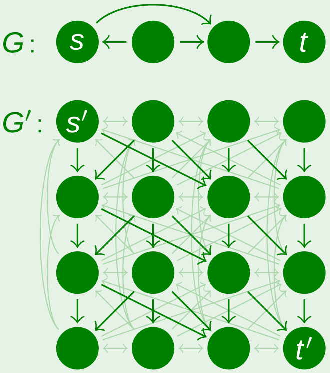
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query:
  $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

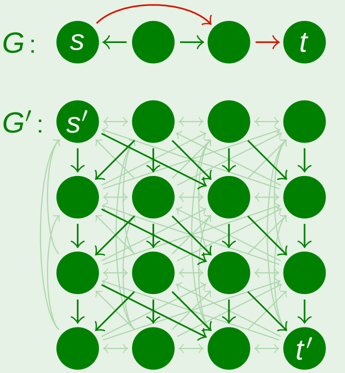
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query:
  $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That DISTANCE$_{\text{tourn}}$ is NL-complete

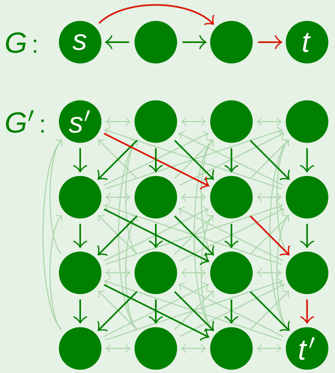
Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query:
  $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

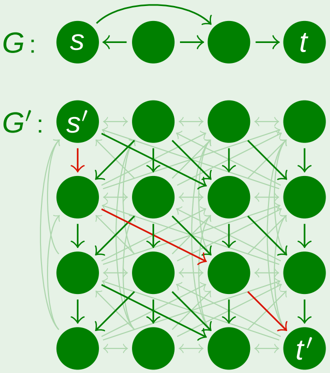
Reduce REACH to DISTANCE$_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query:
  $(G', s', t', 3) \in$ DISTANCE$_{\text{tourn}}$?

Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

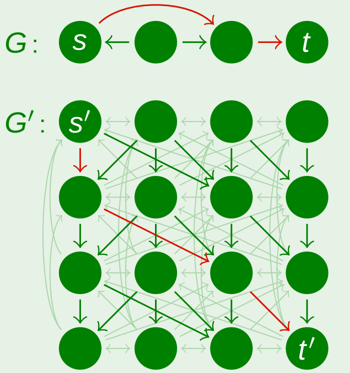
Reduce REACH to DISTANCE$_{\text{tourn}}$

- Is input (G, s, t) in REACH?
- Map G to G'.
- Query: $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}$?

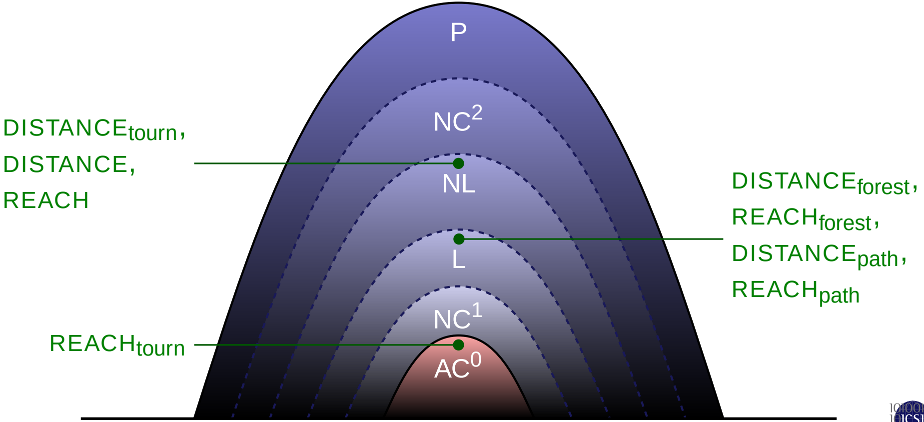
Correctness

- A path in G induces a length-3 path in G'.
- A length-3 path in G' induces a path in G'.

Example



The Complexity Class Hierarchy



Outline

Summary

- First point.
- Second point.
- Third point.

For Further Reading

John Moon.
Topics on Tournaments.
Holt, Rinehart, and Winston, 1968.

Arfst Nickelsen and Till Tantau.
On reachability in graphs with bounded independence number.
In *Proc. of COCOON 2002*, Springer-Verlag, 2002.

Till Tantau
A logspace approximation scheme for the shortest path problem for graphs with bounded independence number.
In *Proc. of STACS 2004*, Springer-Verlag, 2002.
In press.