Lecture 17

NL = coNL

NL = coNL

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in \mathsf{NL}\}$

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in \mathsf{NL}\}$

Example: $\overline{PATH} \in$ coNL.

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$$\overline{PATH} = \{\ \langle G, s, t\rangle \mid G \text{ is a directed graph s.t. there is no path from } s \text{ to } t\}$$

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$$\overline{PATH} = \{\ \langle G, s, t \rangle \mid G \text{ is a directed graph s.t. there is no path from } s \text{ to } t\}$$

Claim: $\overline{PATH} \in \text{NL} \implies \text{NL} = \text{coNL}$

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$$\overline{PATH} = \{\ \langle G, s, t\rangle \mid G \text{ is a directed graph s.t. there is no path from } s \text{ to } t\}$$

Claim: $\overline{PATH} \in$ NL $\implies$ NL = coNL

Proof:

NL $=$ coNL

Definition: coNL $= \{L \mid \overline{L} \in \mathsf{NL}\}$

Example: $\overline{PATH} \in$ coNL.

$$\overline{PATH} = \{ \langle G, s, t\rangle \mid G \text{ is a directed graph s.t. there is no path from } s \text{ to } t\}$$

Claim: $\overline{PATH} \in \mathsf{NL} \implies \mathsf{NL} = \mathsf{coNL}$

Proof: coNL $\subseteq$ NL:

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\Longrightarrow$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ coNL.

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in \text{NL}\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t\rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in \text{NL} \implies \text{NL} = \text{coNL}$

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$.

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\implies$ NL $=$ coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$. ($\because$ $\overline{PATH}$ is **coNL-complete**)

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{\ \langle G, s, t\rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\implies$ NL $=$ coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$. ($\because$ $\overline{PATH}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{PATH}$ and then use **NL** machine of $\overline{PATH}$.

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t\rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\Longrightarrow$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$. ($\because$ $\overline{PATH}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{PATH}$ and then use **NL** machine of $\overline{PATH}$.

NL $\subseteq$ coNL:

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{\textit{PATH}} \in$ coNL.

$\overline{\textit{PATH}} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{\textit{PATH}} \in$ NL $\Longrightarrow$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{\textit{PATH}}$. ($\because$ $\overline{\textit{PATH}}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{\textit{PATH}}$ and then use **NL** machine of $\overline{\textit{PATH}}$.

NL $\subseteq$ coNL:

$L \in$ NL

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{\mathit{PATH}} \in$ coNL.

$\overline{\mathit{PATH}} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{\mathit{PATH}} \in$ NL $\implies$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{\mathit{PATH}}$. ($\because$ $\overline{\mathit{PATH}}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{\mathit{PATH}}$ and then use **NL** machine of $\overline{\mathit{PATH}}$.

NL $\subseteq$ coNL:

$L \in$ NL $\implies \overline{L} \in$ coNL

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{\mathit{PATH}} \in$ coNL.

$\overline{\mathit{PATH}} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{\mathit{PATH}} \in$ NL $\implies$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{\mathit{PATH}}$. ($\because$ $\overline{\mathit{PATH}}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{\mathit{PATH}}$ and then use **NL** machine of $\overline{\mathit{PATH}}$.

NL $\subseteq$ coNL:

$L \in$ NL $\implies \overline{L} \in$ coNL $\implies \overline{L} \in$ NL

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t\rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\implies$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$. ($\because$ $\overline{PATH}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{PATH}$ and then use **NL** machine of $\overline{PATH}$.

NL $\subseteq$ coNL:

$L \in$ NL $\implies \overline{L} \in$ coNL $\implies \overline{L} \in$ NL $\implies L \in$ coNL

NL = coNL

Definition: coNL $= \{L \mid \overline{L} \in$ NL$\}$

Example: $\overline{PATH} \in$ coNL.

$\overline{PATH} = \{ \langle G, s, t \rangle \mid G$ is a directed graph s.t. there is no path from s to $t\}$

Claim: $\overline{PATH} \in$ NL $\implies$ NL = coNL

Proof: coNL $\subseteq$ NL:

Let $L \in$ **coNL**. Then, $L \leq_l \overline{PATH}$. ($\because$ $\overline{PATH}$ is **coNL-complete**)

NL machine for L will first reduce L to $\overline{PATH}$ and then use **NL** machine of $\overline{PATH}$.

NL $\subseteq$ coNL:

$L \in$ **NL** $\implies \overline{L} \in$ **coNL** $\implies \overline{L} \in$ **NL** $\implies L \in$ **coNL**

■

NL = coNL

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in \mathsf{NL}$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in \mathsf{NL}$.

Proof:

NL $=$ coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t\rangle \in \overline{PATH}$ that is verifiable in logspace.

NL $=$ coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: Idea: Make a read-once certificate for $\langle G, s, t\rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in \mathsf{NL}$.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t \rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t \rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t \rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in \mathsf{NL}$.

Proof: Idea: Make a read-once certificate for $\langle G, s, t\rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

Verification: Check $v_1 = s, v_k = v$

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t\rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

Verification: Check $v_1 = s, v_k = v, v_jv_{j+1}$ is an edge, and $k \leq i + 1$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t \rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

Verification is doable in logspace, read-once manner.

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

Verification: Check $v_1 = s, v_k = v, v_j v_{j+1}$ is an edge, and $k \leq i + 1$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in \mathsf{NL}$.

Proof: **Idea:** Make a read-once certificate for $\langle G, s, t \rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

verification is doable in logspace, read-once manner.

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

Verification: Check $v_1 = s$, $v_k = v$, $v_j v_{j+1}$ is an edge, and $k \leq i + 1$.

Certificate: A walk of length at most i from s to v, denoted $path_i(v)$.

NL = coNL

Immerman-Szelepcsényi Theorem: $\overline{PATH} \in$ NL.

Proof: Idea: Make a read-once certificate for $\langle G, s, t\rangle \in \overline{PATH}$ that is verifiable in logspace.

Let C_i = Set of all the vertices of G that have a path of length $\leq i$ from s.

Formally, we want a certificate for $t \notin C_n$.

Q. For a vertex v, how can someone prove that $v \in C_i$?

Verification is doable in logspace, read-once manner.

Proposed solution: A sequence of vertices, $(v_1, v_2, \ldots, v_k)$.

Verification: Check $v_1 = s$, $v_k = v$, $v_j v_{j+1}$ is an edge, and $k \leq i + 1$.

Certificate: A walk of length at most i from s to v, denoted $path_i(v)$.

…

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

Proof of $\mathsf{NL} = \mathsf{coNL}$ ...

Q. For a vertex v, how can someone prove that $v \notin C_i$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1$

Proof of NL = coNL ...

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

Proof of NL = coNL ...

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2 , \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.
- $\forall j$, $v_j \neq v$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.
- $\forall j$, $v_j \neq v$.
- $k = |C_i|$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2 , \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.
- $\forall j$, $v_j \neq v$.
- $k = |C_i|$.
- $v_1 < v_2 < \ldots < v_k$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.
- $\forall j$, $v_j \neq v$.
- $k = |C_i|$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_i|$?

Idea: For every $u \in C_i$, prove that $u \in C_i$ and $u \neq v$.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most i from s to v_j.
- $\forall j$, $v_j \neq v$.
- $k = |C_i|$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Certificate: $v_1 : path_i(v_1), v_2 : path_i(v_2), \ldots, v_{|C_i|} : path_i(v_{|C_i|})$, where $v_j < v_{j+1}$.

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2 , \ldots, v_k : seq_k$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i - 1$ from s to v_j.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2\,, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i - 1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i - 1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.
- $k = |C_{i-1}|$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2\,, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i-1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.
- $k = |C_{i-1}|$.
- $v_1 < v_2 < \ldots < v_k$.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i-1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.
- $k = |C_{i-1}|$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i - 1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.
- $k = |C_{i-1}|$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Certificate: $nopath_i(v)$

Proof of NL = coNL …

Q. For a vertex v, how can someone prove that $v \notin C_i$, if you know $|C_{i-1}|$?

Idea: For every $u \in C_{i-1}$, prove that $u \in C_{i-1}$ and $u \neq v$ and u has no edge to v.

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is a walk of length at most $i - 1$ from s to v_j.
- $\forall j$, $v_j \neq v$ and v_j has no edge to v.
- $k = |C_{i-1}|$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Certificate: $nopath_i(v) = v_1 : path_{i-1}(v_1), \ldots, v_{|C_{i-1}|} : path_{i-1}(v_{|C_{i-1}|})$, where $v_j < v_{j+1}$.

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proof of NL $=$ coNL ...

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1$

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2$

Proof of NL $=$ coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.

Proof of NL = coNL ...

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.
- # of v_js for which $seq_j = path_i(v_j)$ is c.

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.
- # of v_js for which $seq_j = path_i(v_j)$ is c.
- $k = n$.

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.
- # of v_js for which $seq_j = path_i(v_j)$ is c.
- $k = n$.
- $v_1 < v_2 < \ldots < v_k$.

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.
- # of v_js for which $seq_j = path_i(v_j)$ is c.
- $k = n$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Proof of NL = coNL …

Q. How can someone prove that $|C_i| = c$, if you know $|C_{i-1}|$?

Idea: For every $u \in V(G)$, prove that $u \in C_i$ or $u \notin C_i$ (whichever is true).

Proposed Solution: $v_1 : seq_1, v_2 : seq_2, \ldots, v_k : seq_k$, where $v_j \in V(G)$ and seq_j is a sequence of vertices.

Verification: Check whether:

- $\forall j$, seq_j is either $path_i(v_j)$ or $nopath_i(v_j)$.
- # of v_js for which $seq_j = path_i(v_j)$ is c.
- $k = n$.
- $v_1 < v_2 < \ldots < v_k$.

Verification is doable in logspace, read-once manner.

Certificate: $size_i(c) = \ldots\ldots$

Proof of $\mathsf{NL} = \mathsf{coNL}$ ...

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

There exists a read-once logspace verifiable certificate that:

Proof of $\mathsf{NL} = \mathsf{coNL}$ …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

$$size_1(|C_1|)$$

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Certificate that proves the size of C_1 knowing $|C_0|$.

$$size_1(|C_1|)$$

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Certificate that proves the size of C_1 knowing $|C_0|$.

$$size_1(|C_1|),\ size_2(|C_2|)$$

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Certificate that proves the size of C_1 knowing $|C_0|$.

$$size_1(|C_1|),\ size_2(|C_2|),\ \ldots,\ size_{n-1}(|C_{n-1}|)$$

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Certificate that proves the size of C_1 knowing $|C_0|$.

$$size_1(|C_1|),\ size_2(|C_2|),\ \ldots,\ size_{n-1}(|C_{n-1}|),\ nopath_n(t)$$

Proof of NL = coNL …

There exists a read-once logspace verifiable certificate that:

- Proves $v \in C_i$.
- Proves $v \notin C_i$, if $|C_{i-1}|$ is already known. (certificate is $nopath_i(v)$)
- Proves $|C_i| = c$, if $|C_{i-1}|$ is already known. (certificate is $size_i(c)$)

We know $C_0 = \{s\}, |C_0| = 1$.

Final certificate for $\overline{PATH}$:

Certificate that proves the size of C_1 knowing $|C_0|$.

$$size_1(|C_1|), size_2(|C_2|), \ldots, size_{n-1}(|C_{n-1}|), nopath_n(t)$$

Corollary: For every space-constructible function $S(n) \geq \log n$,

$$\text{NSPACE}(S(n)) = \text{coNSPACE}(S(n)).$$