Dynamic Complexity of the Dyck Reachability

Patricia Bouyer-Decitre & Vincent Jugé

CNRS, LSV & ENS Paris-Saclay

25/04/2017

Contents

Dynamic Complexity of Decision Problems

Modulo 3 Decision

- Input: Bit vector $b_1 \cdot b_2 \cdot \ldots \cdot b_n \in \mathbb{F}_3^n$
- Output: **Yes** if $b_1 + b_2 + \ldots + b_n = 0$ — **No** otherwise

Dynamic Complexity of Decision Problems

Modulo 3 Decision

- Input: Bit vector $b_1 \cdot b_2 \cdot \ldots \cdot b_n \in \mathbb{F}_3^n$
- Output: **Yes** if $b_1 + b_2 + \ldots + b_n = 0$ — **No** otherwise

Solving this problem...

- **Static world**: membership in a regular language

Dynamic Complexity of Decision Problems

Modulo 3 Decision

- Input: Bit vector $b_1 \cdot b_2 \cdot \ldots \cdot b_n \in \mathbb{F}_3^n$
- Output: **Yes** if $b_1 + b_2 + \ldots + b_n = 0$ — **No** otherwise

Solving this problem...

- **Static world**: membership in a regular language
- **Dynamic world**: what if some bit b_k changes?
  - Maintain predicates $\mathbf{Aux}_i \equiv (b_1 + b_2 + \ldots + b_n = i)$ for $i \in \mathbb{F}_3$
  - Update the values of $\mathbf{Aux}_0$, $\mathbf{Aux}_1$, $\mathbf{Aux}_2$ when b_k changes
  - Use the new value of $\mathbf{Aux}_0$ and answer the problem

Dynamic Complexity of Decision Problems

Modulo 3 Decision

- Input: Bit vector $b_1 \cdot b_2 \cdot \ldots \cdot b_n \in \mathbb{F}_3^n$
- Output: **Yes** if $b_1 + b_2 + \ldots + b_n = 0$ — **No** otherwise

Solving this problem...

- **Static world**: membership in a regular language
- **Dynamic world**: what if some bit b_k changes?
  - Maintain predicates $\mathbf{Aux}_i \equiv (b_1 + b_2 + \ldots + b_n = i)$ for $i \in \mathbb{F}_3$
  - Update the values of $\mathbf{Aux}_0$, $\mathbf{Aux}_1$, $\mathbf{Aux}_2$ when b_k changes
  - Use the new value of $\mathbf{Aux}_0$ and answer the problem

How complex is it?

- **Static world**: linear time
- **Dynamic world**:
  - **Easy** initial instance $(b_1 = b_2 = \ldots = b_n = 0)$: constant time
  - Each update: constant time

Dynamic Complexity of Decision Problems

Reachability in DAGs

- Input: Directed acyclic graph $G = (V, E)$ & two vertices $s, t \in V$
- Output: **Yes** if $\exists$ path from s to t in G — **No** otherwise

Dynamic Complexity of Decision Problems

Reachability in DAGs

- Input: Directed acyclic graph $G = (V, E)$ & two vertices $s, t \in V$
- Output: **Yes** if $\exists$ path from s to t in G — **No** otherwise

Solving this problem...

- **Static world**: use your favorite graph exploration algorithm
- **Dynamic world**: what if edge $u \rightarrow v$ is inserted/deleted?
  - Maintain a predicate $\mathbf{E}^\star(x, y) \equiv (\exists$ path from x to y in $G)$ for $x, y \in V$
  - Update the values of $\mathbf{E}^\star(x, y)$ when $u \rightarrow v$ is inserted/deleted
  - Use the new value of $\mathbf{E}^\star(s, t)$ and answer the problem

Dynamic Complexity of Decision Problems

Reachability in DAGs

- Input: Directed acyclic graph $G = (V, E)$ & two vertices $s, t \in V$
- Output: **Yes** if $\exists$ path from s to t in G — **No** otherwise

Solving this problem...

- **Static world**: use your favorite graph exploration algorithm
- **Dynamic world**: what if edge $u \to v$ is inserted/deleted?
  - Maintain a predicate $\mathbf{E}^\star(x, y) \equiv (\exists$ path from x to y in $G)$ for $x, y \in V$
  - Update the values of $\mathbf{E}^\star(x, y)$ when $u \to v$ is inserted/deleted
  - Use the new value of $\mathbf{E}^\star(s, t)$ and answer the problem

How complex is it?

- **Static world**: linear time
- **Dynamic world**:
  - **Easy** initial edgeless instance: FO formulas
  - Each update: FO formulas

Dynamic Complexity of Decision Problems

Reachability in DAGs

- Input: Directed acyclic graph $G = (V, E)$ & two vertices $s, t \in V$
- Output: **Yes** if $\exists$ path from s to t in G — **No** otherwise

Solving this problem...

- **Static world**: use your favorite graph exploration algorithm
- **Dynamic world**: what if edge $u \to v$ is inserted/deleted?
  - Maintain a predicate $\mathbf{E}^\star(x, y) \equiv (\exists$ path from x to y in $G)$ for $x, y \in V$
  - Update the values of $\mathbf{E}^\star(x, y)$ when $u \to v$ is inserted/deleted
  - Use the new value of $\mathbf{E}^\star(s, t)$ and answer the problem

How complex is it?

- **Static world**: linear time
- **Dynamic world**:
  - **Easy** initial edgeless instance: FO formulas (**parallel** $\approx$constant time)
  - Each update: FO formulas (**parallel** $\approx$constant time)

FO formulas $\Rightarrow$ parallel $\approx$constant time

$$\phi = \exists x. \forall y. \psi(x, y) \vee \psi(y, x)$$

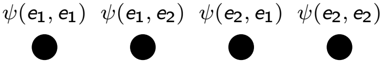
FO formulas ⇒ parallel ≈constant time

$$\phi = \exists x. \forall y. \psi(x, y) \vee \psi(y, x)$$

$\psi(e_1, e_1)$ $\psi(e_1, e_2)$ $\psi(e_2, e_1)$ $\psi(e_2, e_2)$

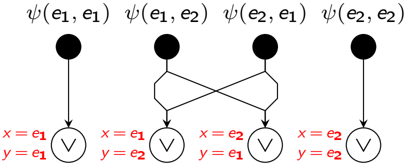
FO formulas ⇒ parallel ≈constant time

$$\phi = \exists x.\forall y.\psi(x, y) \vee \psi(y, x)$$



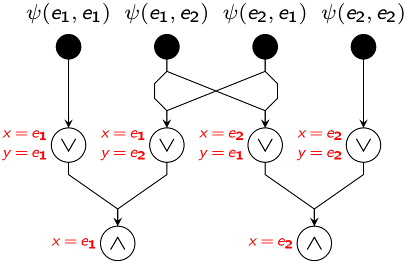
FO formulas ⇒ parallel ≈constant time

$$\phi = \exists x. \forall y. \psi(x,y) \vee \psi(y,x)$$



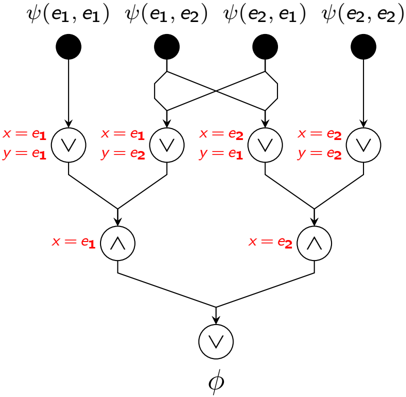
FO formulas ⇒ parallel ≈constant time

$$\phi = \exists x. \forall y. \psi(x, y) \vee \psi(y, x)$$



FO formulas ⇒ parallel ≈constant time

$$\phi = \exists x.\forall y.\psi(x,y) \vee \psi(y,x)$$

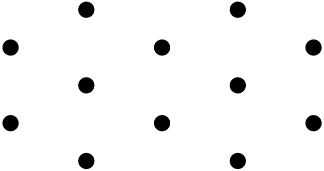


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓

$\mathbf{E}^\star(x, y) \leftarrow (x = y)$

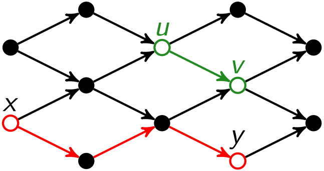


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$

$\mathbf{E}^\star(x, y) \leftarrow \mathbf{E}^\star(x, y)$

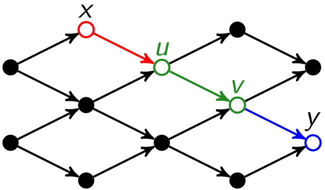


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$: ✓

$$\mathbf{E}^\star(x,y) \leftarrow \mathbf{E}^\star(x,y) \vee \left(\mathbf{E}^\star(x,u) \wedge \mathbf{E}^\star(v,y)\right)$$

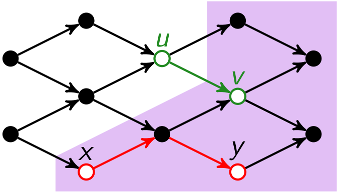


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \rightarrow v$: ✓
- Update after **deleting** the edge $u \rightarrow v$

$$\mathbf{E}^{\star}(x, y) \leftarrow (\mathbf{E}^{\star}(x, y) \wedge \neg \mathbf{E}^{\star}(x, u))$$

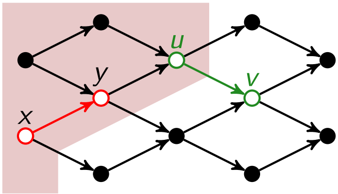


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$: ✓
- Update after **deleting** the edge $u \to v$

$$\mathbf{E}^\star(x,y) \leftarrow (\mathbf{E}^\star(x,y) \wedge \neg \mathbf{E}^\star(x,u)) \vee (\mathbf{E}^\star(x,y) \wedge \mathbf{E}^\star(y,u))$$

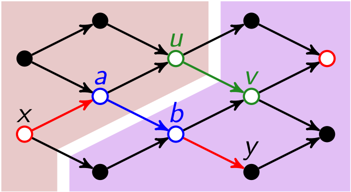


Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$: ✓
- Update after **deleting** the edge $u \to v$: ✓

$$\begin{aligned}\mathbf{E}^\star(x,y) \leftarrow\ & (\mathbf{E}^\star(x,y) \wedge \neg \mathbf{E}^\star(x,u)) \vee \\ & (\mathbf{E}^\star(x,y) \wedge \mathbf{E}^\star(y,u)) \vee \\ & (\exists a. \exists b. \mathbf{E}^\star(x,a) \wedge \mathbf{E}^\star(b,y) \wedge \\ & \ (a \to b) \wedge (a,b) \neq (u,v) \wedge \\ & \ \mathbf{E}^\star(a,u) \wedge \neg \mathbf{E}^\star(b,u))\end{aligned}$$



Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \rightarrow v$: ✓
- Update after **deleting** the edge $u \rightarrow v$: ✓

Definition (Patnaik & Immerman 97, Dong & Su & Topor 93)

A decision problem with updates is in $\mathcal{C}$-**DynFO** if $\exists$ predicates s.t.:

- every predicate can be initialized in $\mathcal{C}$
- every predicate can be updated in FO
- one predicate is the goal predicate

Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$: ✓
- Update after **deleting** the edge $u \to v$: ✓

Definition (Patnaik & Immerman 97, Dong & Su & Topor 93)

A decision problem with updates is in **DynFO** if $\exists$ predicates s.t.:

- every predicate can be initialized in FO
- every predicate can be updated in FO
- one predicate is the goal predicate

Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \to v$: ✓
- Update after **deleting** the edge $u \to v$: ✓

Definition (Patnaik & Immerman 97, Dong & Su & Topor 93)

A decision problem with updates is in **DynFO** if $\exists$ predicates s.t.:

- every predicate can be initialized in FO
- every predicate can be updated in FO
- one predicate is the goal predicate

\$1000000 question:

$P \stackrel{?}{=} NP$

Dynamic Complexity of Decision Problems

Reachability in DAGs with FO formulas

- Initialization (on the edgeless graph): ✓
- Update after **inserting** the edge $u \rightarrow v$: ✓
- Update after **deleting** the edge $u \rightarrow v$: ✓

Definition (Patnaik & Immerman 97, Dong & Su & Topor 93)

A decision problem with updates is in **DynFO** if $\exists$ predicates s.t.:

- every predicate can be initialized in FO
- every predicate can be updated in FO
- one predicate is the goal predicate

\$1000000 question:

P $\stackrel{?}{=}$ NP

1000000 kr. question:

PTime-DynFO $\stackrel{?}{=}$ PTime

Dynamic Complexity of Decision Problems

Some more problems in DynFO

- Reachability in **undirected** graphs (Patnaik & Immerman 97)
- Integer multiplication (Patnaik & Immerman 97)
- **Dyck** reachability in DAGs (Weber & Schwentick 07)
- Context-free language membership (Gelade et al. 08)
- Distance in undirected graphs (Grädel & Siebertz 12)
- Reachability in **directed** graphs (Datta et al. 15)
- **Context-free** reachability in DAGs (Muñoz et al. 16)

Dynamic Complexity of Decision Problems

Some more problems in DynFO

- Reachability in **undirected** graphs (Patnaik & Immerman 97)
- Integer multiplication (Patnaik & Immerman 97)
- **Dyck** reachability in DAGs (Weber & Schwentick 07)
- Context-free language membership (Gelade et al. 08)
- Distance in undirected graphs (Grädel & Siebertz 12)
- Reachability in **directed** graphs (Datta et al. 15)
- **Context-free** reachability in DAGs (Muñoz et al. 16)

Some problems that are **probably not** in PTime-DynFO

- Reachability in 2-player games (Patnaik & Immerman 97)
- **Dyck** reachability in (**un**)**directed** graphs (**Bouyer & Jugé 17**)

Contents

Reachability in 2-Player Games

Reachability in 2-Player Games

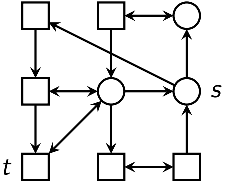
Moving a token on a finite directed graph

- Input: Directed graph $G = (V, E)$, a partition $V_A \uplus V_B = V$, two vertices $s, t \in V$
  - A token is first placed in s
  - Alice controls V_A, Barbara controls V_B
  - Players move the token along edges of G (when they can)
- Alice wins if either:
  - the token reaches a vertex $x \in V_B$ without outgoing edge
  - the token reaches the vertex t
- Output: **Yes** if Alice has a winning strategy — **No** otherwise

Reachability in 2-Player Games

Who wins?

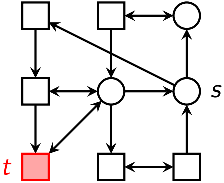
○ Alice's vertices
□ Barbara's vertices



Reachability in 2-Player Games

Who wins?

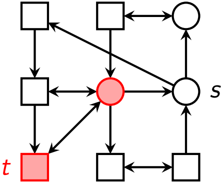
○ Alice's vertices
□ Barbara's vertices
■ ● Alice's winning vertices



Reachability in 2-Player Games

Who wins?

○ Alice's vertices
□ Barbara's vertices
■ ● Alice's winning vertices



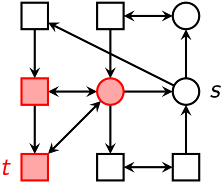
Reachability in 2-Player Games

Who wins?

◯ Alice's vertices

☐ Barbara's vertices

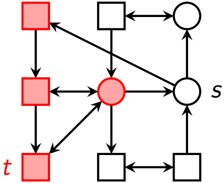
Alice's winning vertices



Reachability in 2-Player Games

Who wins?

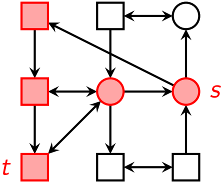
◯ Alice's vertices
☐ Barbara's vertices
■ ● Alice's winning vertices



Reachability in 2-Player Games

Who wins?

Alice !



◯ Alice's vertices

□ Barbara's vertices

Alice's winning vertices

Reachability in 2-Player Games

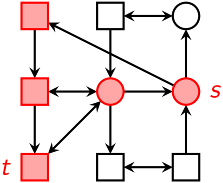
Who wins?

Alice !

○ Alice's vertices

□ Barbara's vertices

■ ● Alice's winning vertices



Theorem (Patnaik & Immerman 97)

Reachability in 2-player games is in PTime-DynFO iff

PTime = PTime-DynFO

Reachability in 2-Player Games

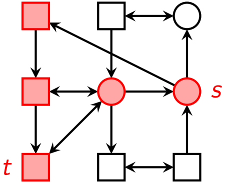
Who wins?

Alice !

○ Alice's vertices

□ Barbara's vertices

■ ● Alice's winning vertices



Theorem (Patnaik & Immerman 97)

Reachability in 2-player games is in PTime-DynFO iff

PTime = PTime-DynFO

PTime-complete
for **LogSpace** reductions

Reachability in 2-Player Games

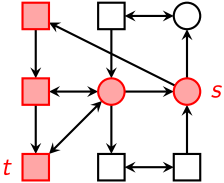
Who wins?

Alice !

○ Alice's vertices

□ Barbara's vertices

■ ● Alice's winning vertices



Theorem (Patnaik & Immerman 97)

Reachability in 2-player games is in PTime-DynFO iff

PTime = PTime-DynFO

PTime-complete for **LogSpace** reductions	PTime-complete for **dynamic-adapted** reductions

Reachability in 2-Player Games

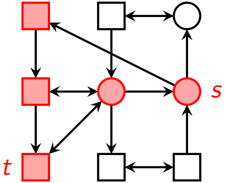
Who wins?

Alice !

○ Alice's vertices

□ Barbara's vertices

■ ● Alice's winning vertices



Theorem (Patnaik & Immerman 97)

Reachability in 2-player games is in PTime-DynFO iff

PTime = PTime-DynFO

PTime-complete for **LogSpace** reductions	PTime-complete$^{\text{dyn}}$ for **dynamic-adapted** reductions

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( ))
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( ))
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( ))
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( ))
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ]))
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]

Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

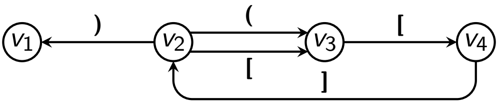
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



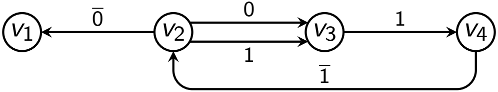
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



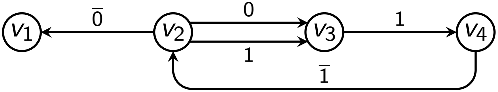
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



v_4

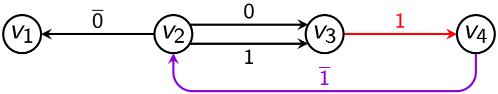
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



$$v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2$$

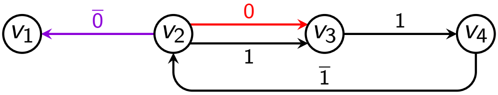
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



$$v_2 \xrightarrow{0} v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2 \xrightarrow{\overline{0}} v_1$$

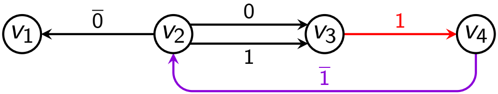
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



$$v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2 \xrightarrow{0} v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2 \xrightarrow{\overline{0}} v_1$$

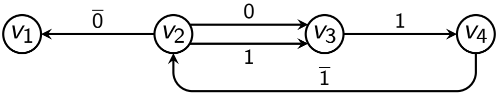
Dyck Reachability

Dyck words = Well-parenthesized words

Are these words Dyck?

- ([( )] ( )): ✓
- ([( ])): ✗
- ([( )] ( )]: ✗

Dyck paths = Paths labeled with Dyck words



$$v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2 \xrightarrow{0} v_3 \xrightarrow{1} v_4 \xrightarrow{\overline{1}} v_2 \xrightarrow{\overline{0}} v_1$$

Theorem (Weber & Schwentick 05)

Computing endpoints of Dyck paths in **acyclic** graphs is in DynFO.

Contents

Dyck Reachability in Directed Graphs is Hard

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

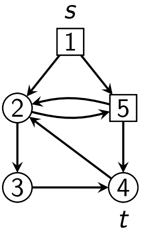
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff

PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



○ Alice

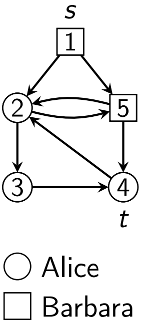
□ Barbara

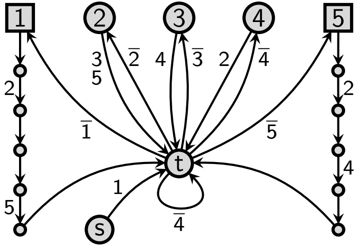
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff
PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



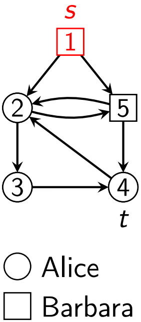


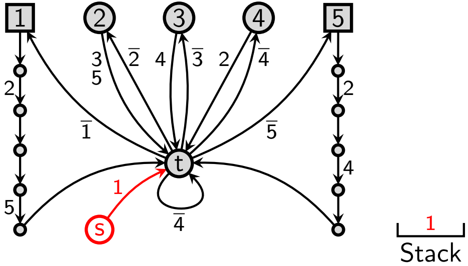
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



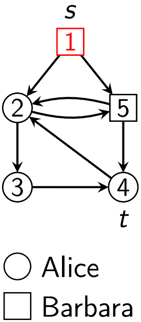


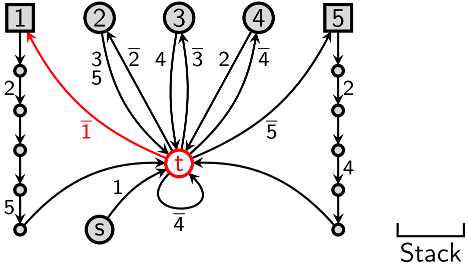
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



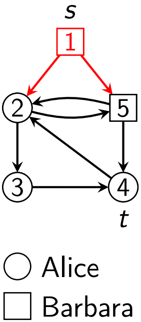


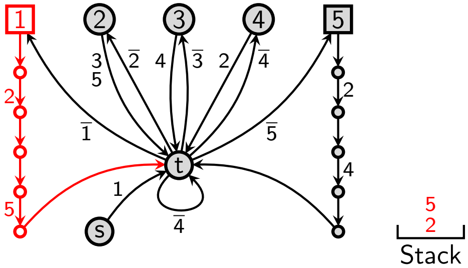
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



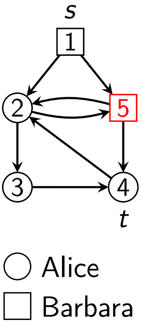


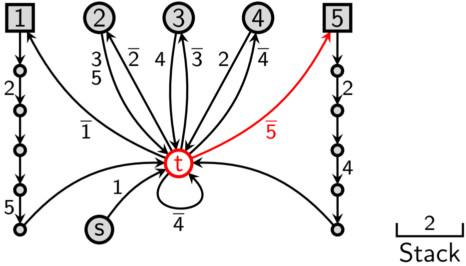
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



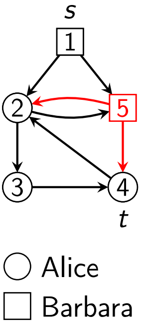


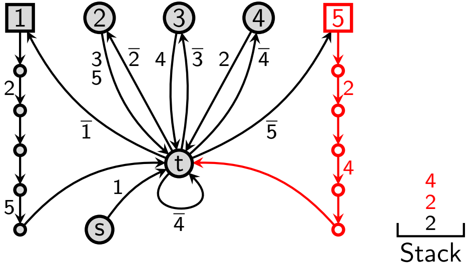
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



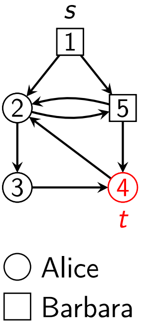


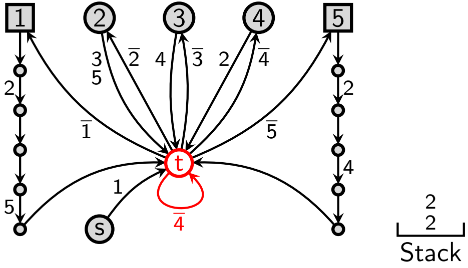
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



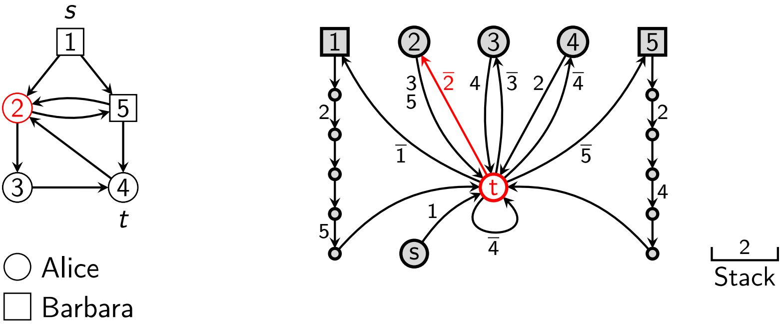


Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!

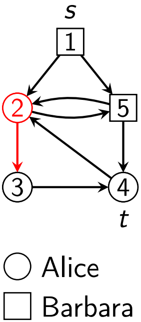


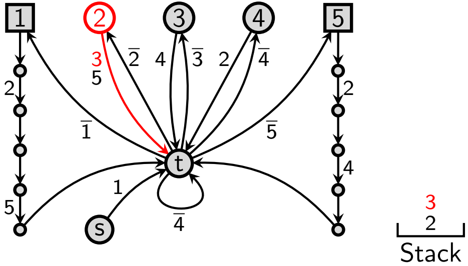
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



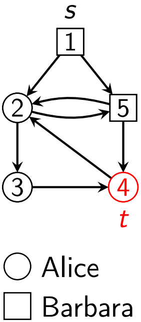


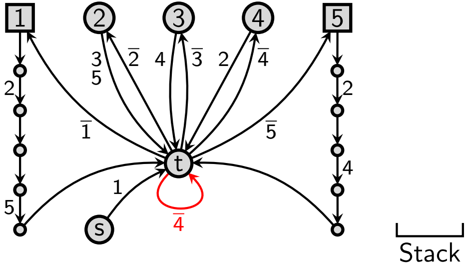
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff
PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!





Until, one day . . .

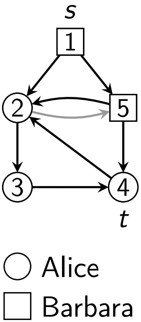
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

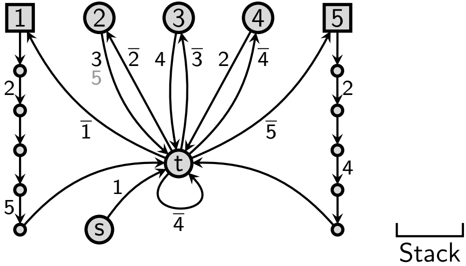
Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff

PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



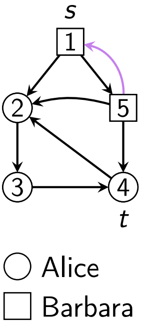


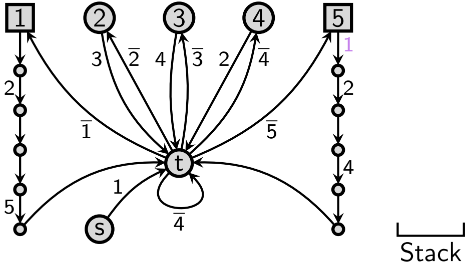
Dyck Reachability in Directed Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!



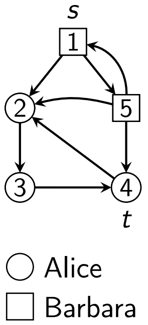


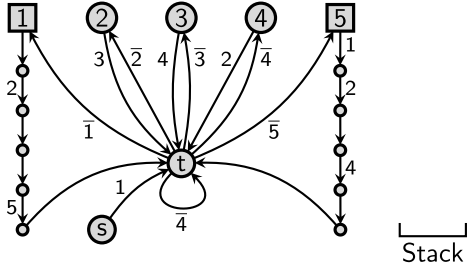
Dyck Reachability in Directed Graphs is PTime-Hard^{dyn}

Theorem #1 (Bouyer & Jugé 17)

2-letter Dyck reachability in directed graphs is in PTime-DynFO iff PTime = PTime-DynFO

Use a **dynamic-adapted** reduction from Reachability in 2-player games!





Use binary label encoding & achieve 2-letter Dyck reachability ☺

Dyck Reachability in Undirected Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

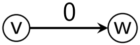
Transform **directed** labeled **edges** into **undirected** labeled **paths**!

Dyck Reachability in Undirected Graphs is PTime-Hard^{dyn}

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!

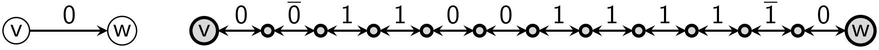


Dyck Reachability in Undirected Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff
PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!

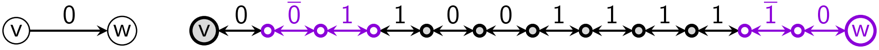


Dyck Reachability in Undirected Graphs is PTime-Hard^{dyn}

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!

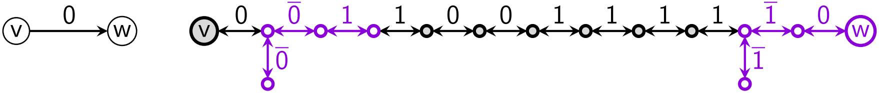


Dyck Reachability in Undirected Graphs is PTime-Hard^{dyn}

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!

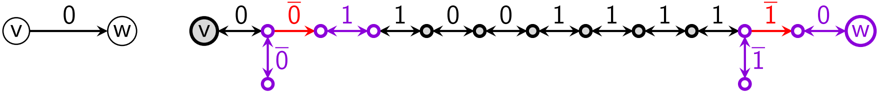


Dyck Reachability in Undirected Graphs is PTime-Hard$^{\text{dyn}}$

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!

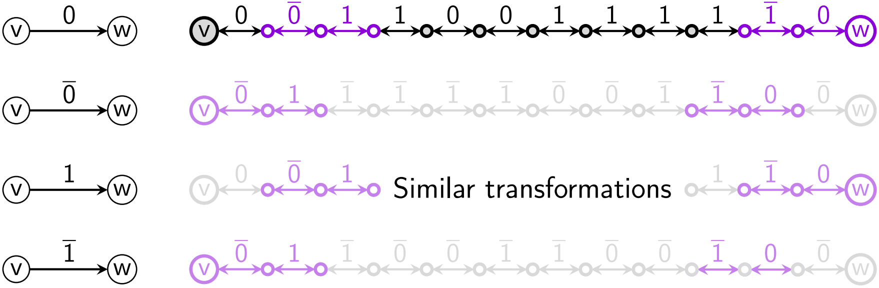


Dyck Reachability in Undirected Graphs is PTime-Hard^{dyn}

Theorem #2 (Bouyer & Jugé 17)

2-letter Dyck reachability in **undirected** graphs is in PTime-DynFO iff PTime = PTime-DynFO

Transform **directed** labeled **edges** into **undirected** labeled **paths**!



And With One Letter Only?

Dynamic complexity of Dyck reachability problems

- With $\geqslant 2$ letters: PTime-complete$^{\text{dyn}}$ in (**un**)**directed** graphs

And With One Letter Only?

Dynamic complexity of Dyck reachability problems

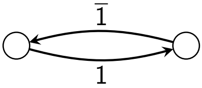
- With $\geqslant 2$ letters: PTime-complete$^{\text{dyn}}$ in (**un**)**directed** graphs
- With 1 letter:
  - in DynFO (and **not** NLogSpace-hard$^{\text{dyn}}$) in **undirected** graphs
  - in NLogSpace in **directed** graphs

Contents

Future work

Some problems to investigate:

- 1-letter Dyck reachability in directed graphs
- Dyck reachability in Cayley graphs



Future work

Some problems to investigate:

- 1-letter Dyck reachability in directed graphs
- Dyck reachability in Cayley graphs