Revisiting Synthesis for One-Counter Automata

Ritam Raha

(Joint work with Guillermo A. Perez)

MOVEP 2020
1. One-Counter Automata (Parametric) and Synthesis Problem

2. Previous Approach

3. Approach with Alternating Two Way Automata for a subclass

4. Approach with Partial Observation Games
One-Counter Automata

n=5
n= \text{max}(0,n-5)
if n=0:
  while (n<10):
    n=n+1
    n= n+100
else:
  return false
One-Counter Automata

```
1 n=5
2 n= max(0,n-5)
3 if n=0:
4     while (n<10):
5         n=n+1
6         n= n+100
7 else:
8     return false
```

▶ Configurations: \((q, c), c \geq 0;\)
One-Counter Automata

```
1 n=5
2 n= max(0,n-x)
3 if n=0:
    4      while (n<10):
    5          n=n+1
    6      n= n+100
4 else:
    5      return false
```

- **Configurations:** \((q, c), c \geq 0;\)
One-Counter Automata

\[ X = \{ x_1, \ldots, x_n \} \]

5 \rightarrow -x_1 \rightarrow 0 \rightarrow \geq x_2 \rightarrow +100

false
One-Counter Automata

\[ X = \{ x_1, \ldots, x_n \} \]

\[ 5 - x_1 = 0 \geq x_2 + 100 \geq 1 \]

Definition (\textit{S}OCA\textit{P})

\( \mathcal{A} = (Q, T, \delta, q_{\text{in}}, X) \quad \delta : T \rightarrow Op \)

\( Op := \)
One-Counter Automata

\[ X = \{ x_1, \ldots, x_n \} \]

\[ 5 - x_1 = 0 \geq x_2 + 100 \geq 1 \]

**Definition (OCA)**

\[ \mathcal{A} = (Q, T, \delta, q_{\text{in}}, X) \quad \delta : T \rightarrow \text{Op} \]

\text{Op} :=

- \( \text{CU} := \{ +a : a \in \mathbb{Z} \} \)
- \( \text{CT} := \{ = 0, \geq a, = a : a \in \mathbb{Z} \} \)
One-Counter Automata

Definition ($\langle S \rangle$OCA$\langle P \rangle$)

\[ A = (Q, T, \delta, q_{in}, X) \] \hspace{0.5cm} \delta : T \rightarrow Op

\[ Op := \]

- $CU := \{ +a : a \in \mathbb{Z} \}$
- $CT := \{ = 0, \geq a, = a : a \in \mathbb{Z} \}$
- $PU := \{ +x, -x : x \in X \}$
- $PT := \{ = x, \geq x : x \in X \}$

\[ X = \{ x_1, \ldots, x_n \} \]
One-Counter Automata

\[ X = \{x_1, \ldots, x_n\} \]

**Definition (<S>OCA<P>)**

\[ \mathcal{A} = (Q, T, \delta, q_{in}, X) \quad \delta : T \rightarrow Op \]

\[ Op := \]

- \( CU := \{+a : a \in \mathbb{Z}\} \)
- \( PU := \{+x, -x : x \in X\} \)
- \( CT := \{=0, \geq a, =a : a \in \mathbb{Z}\} \)
- \( PT := \{=x, \geq x : x \in X\} \)

Non-parametric: \( X = \emptyset \)
## One-Counter Automata

<table>
<thead>
<tr>
<th>Models</th>
<th>$CU$</th>
<th>$PU$</th>
<th>$ZT$</th>
<th>$PT$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Parametric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOCA</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Parametric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCAPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>${-1, 0, 1}$</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SOCAP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Decision Problems

Non-Parametric:

Parametric:
Decision Problems

Non-Parametric:

\[
\exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f
\]

Reach

[NP-complete] (Haase-Kreutzer-Joel-Worrell '09)

Parametric:

\[
\exists V(X) \text{ such that for all infinite } \rho, (q_{in}, 0) \xrightarrow{\rho} Vq_f
\]

[in NEXPTIME] (Haase-Kreutzer-Joel-Worrell '09)

SynthReach
Decision Problems

Non-Parametric:

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

[NP-complete]
(Haase-Kreutzer-Joel-Worrell ’09)

Parametric:

\[ \exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} Vq_f \]

[In NEXPTIME]
(Haase-Kreutzer-Joel-Worrell ’09)
Decision Problems

Non-Parametric:

∃ρ such that

(q_{in}, 0) \xrightarrow{\rho} q_f

[NP-complete]
(Haase-Kreutzer-Joel-Worrell '09)

Parametric:

∃V(X) such that ∃ρ,

(q_{in}, 0) \xrightarrow{\rho} V q_f
**Decision Problems**

**Non-Parametric:**

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

*Reach*

[**NP-complete**]

(Haase-Kreutzer-Joel-Worrell '09)

**Parametric:**

\[ \exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]

*Par-Reach*

[in **NEXPTIME**]

(Haase-Kreutzer-Joel-Worrell '09)
**Decision Problems**

**Non-Parametric:**

\[
\exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f
\]

[**NP-complete**]
(Haase-Kreutzer-Joel-Worrell ’09)

**Parametric:**

\[
\exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f
\]

[**in NEXPTIME**]
(Haase-Kreutzer-Joel-Worrell ’09)
Decision Problems

Non-Parametric:

**REACH**
\[
\exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f
\]

[NP-complete]
(Haase-Kreutzer-Joel-Worrell ’09)

**UNIVREACH**
For all infinite \(\rho\),
\[
(q_{in}, 0) \xrightarrow{\rho} q_f
\]

Parametric:

**PAR-REACH**
\[
\exists V(X) \text{ such that } \exists \rho,
(q_{in}, 0) \xrightarrow{\rho} V q_f
\]

[in NEXPTIME]
(Haase-Kreutzer-Joel-Worrell ’09)

**SYNTHREACH**
\[
\exists V(X) \text{ such that for all}
\text{ infinite } \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f
\]
Decision Problems

Non-Parametric:

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

[Non-Parametric]
(Reach)
(Haase-Kreutzer-Joel-Worrell ’09)

\[ (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[UnivReach]

Parametric:

\[ \exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[SynthReach]

\[ \exists V(X) \text{ such that for all infinite } \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[Par-Reach]

(Haase-Kreutzer-Joel-Worrell ’09)
UnivReach for SOCA

\[ Op = CU \cup ZT; \quad \text{UnivReach: all infinite paths reach } q_f; \]
\( Op = CU \cup ZT; \quad \text{UnivReach: all infinite paths reach } q_f; \)

\( \neg \text{Reach} \equiv \text{two Reach queries} \)
UnivReach for SOCA

\[ Op = CU \cup ZT; \quad \text{UnivReach: all infinite paths reach } q_f; \]

\[ \neg \text{Reach} := \text{two Reach queries (coNP)}; \]
UnivReach for SOCA

$Op = CU \cup ZT$; \hspace{1cm} \textbf{UnivReach}: all infinite paths reach $q_f$;

$\neg \text{Reach} := \text{two Reach queries (coNP)}$;

Hardness from reduction from co-subsetsum.

**Proposition**

The \textit{UnivReach} problem for SOCA is coNP-complete.
### Decision Problems

**Non-Parametric:**

**REACH**

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

[NP-complete]

(Haase-Kreutzer-Joel-Worrell ’09)

**UNIVREACH**

For all infinite \( \rho \),

\[ (q_{in}, 0) \xrightarrow{\rho} q_f \]

**Parametric:**

**PAR-REACH**

\[ \exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[in NEXPTIME]

(Haase-Kreutzer-Joel-Worrell ’09)

**SYNTHREACH**

\[ \exists V(X) \text{ such that for all } \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]
Decision Problems

Non-Parametric:

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

[NP-complete]
(Haase-Kreutzer-Joel-Worrell '09)

Parametric:

\[ \exists V(X) \text{ such that } \exists \rho, \quad (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[in NEXPTIME]
(Haase-Kreutzer-Joel-Worrell '09)

UnivReach

For all infinite \( \rho \),
\[ (q_{in}, 0) \xrightarrow{\rho} q_f \]

[coNP-complete]
Decision Problems

Non-Parametric:

\[ \exists \rho \text{ such that } (q_{in}, 0) \xrightarrow{\rho} q_f \]

[NP-complete] (Haase-Kreutzer-Joel-Worrell '09)

Parametric:

\[ \exists V(X) \text{ such that } \exists \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]

[in NEXPTIME] (Haase-Kreutzer-Joel-Worrell '09)

UniReach

For all infinite \( \rho \),

\[ (q_{in}, 0) \xrightarrow{\rho} q_f \]

[coNP-complete]

SynthReach

\[ \exists V(X) \text{ such that for all } \text{infinite } \rho, (q_{in}, 0) \xrightarrow{\rho} V q_f \]
Presburger arithmetic (PA) := $\langle \mathbb{N}, 0, 1, +, < \rangle$
Presburger arithmetic (PA) := $\langle \mathbb{N}, 0, 1, +, \prec \rangle$

Presburger arithmetic with Divisibility (PAD) := PA + $|$ 

(a \mid b \iff \exists c \in \mathbb{Z} : b = ac)$

Theorem (Lechner-Ouaknine-Worrell'15)

EPAD is decidable in NEXPTIME.
Presburger arithmetic (PA) := \langle \mathbb{N}, 0, 1, +, < \rangle

Presburger arithmetic with Divisibility (PAD) := PA + |
\(a \mid b \iff \exists c \in \mathbb{Z} : b = ac\)

Theorem (Lechner-Ouaknine-Worrell’15)

EPAD is decidable in NEXPTIME.
Outline

1. One-Counter Automata (Parametric) and Synthesis Problem

2. Previous Approach

3. Approach with Alternating Two Way Automata for a subclass

4. Approach with Partial Observation Games
∀∃_R PAD & Undecidability

∀∃_R PAD := ∀z_1 ... ∀z_n ∃x_1 ... ∃x_m ϕ(⃗x,⃗z)
∀∃_R PAD & Undecidability

∀∃_R PAD := ∀z_1 ... ∀z_n ∃x_1 ... ∃x_m φ(⃗x, ⃗z)(f(⃗z) | g(⃗x, ⃗z)).
∀∃_R PAD & Undecidability

- ∀∃_R PAD := ∀z_1 \ldots ∀z_n ∃x_1 \ldots ∃x_m \varphi(\vec{x}, \vec{z})(f(\vec{z}) \mid g(\vec{x}, \vec{z})).

- ∀∃_R PAD^+ := ∀∃_R PAD with \neg not allowed before divisibility.
∀∃_R \text{PAD} & \text{Undecidability}

\begin{itemize}
  \item \( \forall∃_R \text{PAD} := \forall z_1 \ldots \forall z_n \exists x_1 \ldots \exists x_m \varphi(\vec{x}, \vec{z})(f(\vec{z}) | g(\vec{x}, \vec{z})). \)
  \item \( \forall∃_R \text{PAD}^+ := \forall∃_R \text{PAD} \text{ with } \neg \text{ not allowed before divisibility.} \)
\end{itemize}

\textbf{Claim(Lechner’15)}

\textbf{SYNTHREACH} for SOCAP is decidable by a reduction to \( \forall∃_R \text{PAD}^+ \).
∀∃_R PAD & Undecidability

∀∃_R PAD := ∀z_1 ... ∀z_n ∃x_1 ... ∃x_m \varphi(\vec{x}, \vec{z})(f(\vec{z}) | g(\vec{x}, \vec{z})).

∀∃_R PAD^+ := ∀∃_R PAD with \neg not allowed before divisibility.

Claim(Lechner’15)

SYNTHREACH for SOCAP is decidable by a reduction to ∀∃_R PAD^+.

∀∃_R PAD ≡ ∀∃_R PAD^+
∀∃_R PAD & Undecidability

- ∀∃_R PAD := ∀z_1 ... ∀z_n ∃x_1 ... ∃x_m ϕ(⃗x, ⃗z)(f(⃗z) | g(⃗x, ⃗z)).
- ∀∃_R PAD^+ := ∀∃_R PAD with ¬ not allowed before divisibility.

Claim(Lechner’15)

**SynthReach** for SOCAP is decidable by a reduction to ∀∃_R PAD^+.

∀∃_R PAD ≡ ∀∃_R PAD^+

**Idea:** We have to rewrite ¬(a | b)
∀∃_R PAD & Undecidability

▶ ∀∃_R PAD := ∀z_1 \ldots ∀z_n \exists x_1 \ldots \exists x_m \varphi(\vec{x}, \vec{z})(f(\vec{z}) \mid g(\vec{x}, \vec{z})).

▶ ∀∃_R PAD^+ := ∀∃_R PAD with \neg not allowed before divisibility.

Claim(Lechner’15)

SYNTHREACH for SOCAP is decidable by a reduction to ∀∃_R PAD^+.

∀∃_R PAD \equiv ∀∃_R PAD^+

Idea: We have to rewrite \neg(a \mid b)

\neg(a \mid b) \equiv b = aq + r \ \text{where,} \ 0 < r < b.
Using techniques from Bozga-Iosif’05 and Lechner’15:

\( \forall \exists_R^{+} \text{PAD} \iff \forall \exists_R \text{PAD} \)
∀∃_R^{PAD} & Undecidability

Using techniques from Bozga-Iosif’05 and Lechner’15:

∀∃_R^{PAD^+} ↔ ∀∃_R^{PAD} → LCM
Using techniques from Bozga-Iosif’05 and Lechner’15:

\[ \forall \exists_R \text{PAD}^+ \iff \forall \exists_R \text{PAD} \implies \text{LCM} \implies \text{Square} \]
∀∃_R PAD \& Undecidability

Using techniques from Bozga-Iosif’05 and Lechner’15:

∀∃_R PAD^+ \iff ∀∃_R PAD \implies LCM \implies Square \implies Multiplication
Using techniques from Bozga-Iosif’05 and Lechner’15:

\( \forall \exists_R \text{PAD}^+ \iff \forall \exists_R \text{PAD} \longrightarrow \text{LCM} \longrightarrow \text{Square} \longrightarrow \text{Multiplication} \)

Undecidable!!
Decision Problems

Non-Parametric:

\[ \begin{align*}
& \text{Reach} \\
& \exists \rho \text{ such that } \ (q_{in}, 0)^\rho \rightarrow q_f \\
& [\text{NP-complete}] \\
& (\text{Haase-Kreutzer-Joel-Worrell '09})
\end{align*} \]

\[ \begin{align*}
& \text{UnivReach} \\
& \text{For all} \text{ infinite } \rho, \\
& \ (q_{in}, 0)^\rho \rightarrow q_f \\
& [\text{coNP-complete}]
\end{align*} \]

Parametric:

\[ \begin{align*}
& \text{Par-Reach} \\
& \exists V(X) \text{ such that } \exists \rho, \\
& \ (q_{in}, 0)^\rho \rightarrow V q_f \\
& [\text{in NEXPTIME}] \\
& (\text{Haase-Kreutzer-Joel-Worrell '09})
\end{align*} \]

\[ \begin{align*}
& \text{SynthReach} \\
& \exists V(X) \text{ such that for all} \text{ infinite } \rho, \\
& \ (q_{in}, 0)^\rho \rightarrow V q_f
\end{align*} \]
1. One-Counter Automata (Parametric) and Synthesis Problem

2. Previous Approach

3. Approach with Alternating Two Way Automata for a subclass

4. Approach with Partial Observation Games
Alternating Two Way Automaton

Theorem (Serre'06)
The non-emptiness problem for A2As is in PSPACE.

\[ w = ab^\omega \]
Alternating Two Way Automaton

The non-emptiness problem for A2As is in PSPACE.

\[ w = ab^\omega \]
Alternating Two Way Automaton

\[ w = ab^\omega \]

Theorem (Serre'06)
The non-emptiness problem for A2As is in PSPACE.
Alternating Two Way Automaton

\( w = ab^\omega \)

Theorem (Serre’06)

The non-emptiness problem for A2As is in PSPACE.
Alternating Two Way Automaton

The non-emptiness problem for A2As is in PSPACE.

\[ w = ab^\omega \]
The non-emptiness problem for A2As is in PSPACE.
Alternating Two Way Automaton

The non-emptiness problem for A2As is in PSPACE.
Proposition (Similar Idea to Bollig-Quaas-Sangnier'19)

For every OCAPT $\mathcal{A}$ ⇒ an A2A $\mathcal{T}$ of poly-size such that, it accepts words corresponding to valuations for which $\text{SynthReach}$ is true.

Idea:

▶ Encode valuations as parameter word
▶ For every operation, we build an A2A
▶ Accept the reaching runs

$Op = \{-1, 0, +1\} \cup ZT \cup PT$
From OCAPT to A2A

\[ Op = \{-1, 0, +1\} \cup \mathbb{Z}_T \cup PT \]

**Proposition (Similar Idea to Bollig-Quaas-Sangnier’19)**

For every OCAPT \( \mathcal{A} \Rightarrow \) an A2A \( J \) of poly-size such that, it accepts words corresponding to valuations for which \( \text{SYNTHREACH} \) is true.
From OCAPT to A2A

Op = \{-1, 0, +1\} \cup ZT \cup PT

Proposition (Similar Idea to Bollig-Quaas-Sangnier’19)

For every OCAPT \( \mathcal{A} \Rightarrow \) an A2A \( \mathcal{T} \) of poly-size such that, it accepts words corresponding to valuations for which SynthReach is true.

Idea:

▶ Encode valuations as parameter word
▶ For every operation, we build an A2A
▶ Accept the reaching runs
From OCAPT to A2A

\[ \text{Op} = \{-1, 0, +1\} \cup ZT \cup PT \]

**Proposition (Similar Idea to Bollig-Quaas-Sangnier’19)**

*For every OCAPT \( \mathcal{A} \) ⇒ an A2A \( \mathcal{T} \) of poly-size such that, it accepts words corresponding to valuations for which SYNTHREACH is true.*

Idea:

- Encode valuations as parameter word
From OCAPT to A2A

\[ Op = \{-1, 0, +1\} \cup ZT \cup PT \]

**Proposition (Similar Idea to Bollig-Quaas-Sangnier’19)**

For every OCAPT \( \mathcal{A} \Rightarrow \) an A2A \( \mathcal{T} \) of poly-size such that, it accepts words corresponding to valuations for which SYNTHREACH is true.

**Idea:**

- Encode valuations as parameter word
- For every operation, we build an A2A
From OCAPT to A2A

\[ Op = \{-1, 0, +1\} \cup \mathbb{Z}_T \cup PT \]

**Proposition (Similar Idea to Bollig-Quaas-Sangnier’19)**

For every OCAPT \( \mathcal{A} \Rightarrow an \ A2A \mathcal{T} \) of poly-size such that, it accepts **words corresponding to valuations for which SYNTHREACH is true.**

Idea:
- Encode valuations as parameter word
- For every operation, we build an A2A
- Accept the reaching runs
Valuation to Words:
Valuation to Words:

\[ V : X \rightarrow \mathbb{N}; \quad \Sigma = X \cup \{\square\}; \]
Valuation to Words:

\[ V : X \to \mathbb{N}; \quad \Sigma = X \cup \{\Box\}; \]

\[ \Box \]
Valuation to Words:

$V : X \rightarrow \mathbb{N}; \quad \Sigma = X \cup \{\square\};$

$\square x_1 \square x_2 \square \omega$
Valuation to Words:

\[ V : X \rightarrow \mathbb{N}; \quad \Sigma = X \cup \{\Box\}; \]

\[ \Box \Box \Box x_1 \Box x_2 \Box \omega \Rightarrow x_1 \mapsto 2, x_2 \mapsto 3; \]
Valuation to Words:

$$V : X \rightarrow \mathbb{N}$$;

$$\Sigma = X \cup \{\Box\};$$

$$\Box \Box \Box x_1 \Box x_2 \Box \omega \Rightarrow x_1 \mapsto 2, x_2 \mapsto 3;$$

*Op to A2A:*
Valuation to Words:

\[ V : X \rightarrow \mathbb{N}; \quad \Sigma = X \cup \{\square\}; \]

\[ \square\square x_1 \square x_2 \square^\omega \Rightarrow x_1 \mapsto 2, x_2 \mapsto 3; \]

*Op to A2A:*
Valuation to Words:

\[ V : X \rightarrow \mathbb{N}; \quad \Sigma = X \cup \{\square\}; \]

\[ \square \square x_1 \square x_2 \square \omega \Rightarrow x_1 \mapsto 2, x_2 \mapsto 3; \]

**Op to A2A:**

Accept reaching runs:

\[ q \rightarrow \lor \rightarrow \land \rightarrow q' \xrightarrow{\text{simulation}} q_f \xrightarrow{\text{true}} \]

\[ q \rightarrow \lor \rightarrow \land \rightarrow \text{violation} \rightarrow \text{validation} \]
• Encoding Equality Test:
• Encoding Equality Test:

\[ q \lor (q' \land 0) \lor (X \setminus \{x\}, +1) \]

\[ X \setminus \{x\}, +1 \]

\[ x \]

\[ q' \]

\[ 0 \]

\[ +1 \]

\[ X \setminus \{x\}, +1 \]

\[ true \]
• Encoding Decrement:
• Encoding Decrement:
We want to check all paths:
We want to check all paths:

\[ q^{op_1} \xrightarrow{} q_1 \Rightarrow (q, \square, \mathcal{T}_1) \]
We want to check all paths:

$q \xrightarrow{op_1} q_1 \Rightarrow (q, \Box, \mathcal{T}_1)$

$q \xrightarrow{op_2} q_2 \Rightarrow (q, \Box, \mathcal{T}_2)$
We want to check all paths:

\[ q \xrightarrow{\text{op}_1} q_1 \Rightarrow (q, \square, \mathcal{T}_1) \]

\[ q \xrightarrow{\text{op}_2} q_2 \Rightarrow (q, \square, \mathcal{T}_2) \]

\[(q, \square, \mathcal{T}_1 \land \mathcal{T}_2) \in \mathcal{T}\]
We check if the word is valid.
We check if the word is valid.
We gradually build A2A for every op.
We check if the word is valid.
We gradually build A2A for every op.
Finally we have the global A2A.
We check if the word is valid.
We gradually build A2A for every op.
Finally we have the global A2A.
We check if the word is valid.
We gradually build A2A for every op.
Finally we have the global A2A.

PSPACE!!
We check if the word is valid.

We gradually build A2A for every op.

Finally we have the global A2A.

PSPACE!!

For an A2A $\mathcal{T} \Rightarrow$ an NBA $\mathcal{B}$ of exponential size accepting same language!!
We check if the word is valid.

We gradually build A2A for every op.

Finally we have the global A2A.

PSPACE!!

For an A2A \( \mathcal{T} \Rightarrow \) an NBA \( \mathcal{B} \) of exponential size accepting same language!!

Lemma

If SYNTHREACH is true then we can bound the valuation to be exponential and thus poly size in succinct encoding.
We check if the word is valid.

We gradually build A2A for every op.

Finally we have the global A2A.

PSPACE!!

For an A2A $T \Rightarrow$ an NBA $B$ of exponential size accepting same language!!

**Lemma**

*If SYNTHREACH is true then we can bound the valuation to be exponential and thus poly size in succinct encoding.*

Guess the valuation
We check if the word is valid.
We gradually build A2A for every op.
Finally we have the global A2A.

PSPACE!!

For an A2A $\mathcal{T} \Rightarrow$ an NBA $\mathcal{B}$ of exponential size accepting same language!!

Lemma

If $\text{SYNTHREACH}$ is true then we can bound the valuation to be exponential and thus poly size in succinct encoding.

Guess the valuation $\Rightarrow \text{UNIVREACH}$ of SOCA
We check if the word is valid.
We gradually build A2A for every op.
Finally we have the global A2A.

PSPACE!!

For an A2A $\mathcal{T} \Rightarrow$ an NBA $\mathcal{B}$ of exponential size accepting same language!!

**Lemma**

*If $\text{SYNTHREACH}$ is true then we can bound the valuation to be exponential and thus poly size in succinct encoding.*

Guess the valuation $\Rightarrow$ $\text{UNIVREACH}$ of SOCA $\Rightarrow$ $\text{NP}^{\text{NP}}$!!
1. One-Counter Automata (Parametric) and Synthesis Problem

2. Previous Approach

3. Approach with Alternating Two Way Automata for a subclass

4. Approach with Partial Observation Games
Partial Observation Energy Game

- Chooses an action
- Keeps the energy level positive always at infinite play
- Resolves Non-determinism
- Makes the energy level negative at some point
Partial Observation Energy Game

A Partial Observation Energy Game is a type of game where the players take actions and try to keep the energy level positive at all times. The game starts with an initial state and the players choose actions. The resulting states and their associated rewards are shown in the diagram.

- Action 'a' leads to states 'a,4', 'a,1', and 'a,-2'.
- Action 'b' leads to states 'b,-5' and 'b,0'.
- The initial state is labeled with 'Σ,0'.

The game resolves non-determinism at some point, leading to an end state.

'Ritam Raha 24/26'
Partial Observation Energy Game

- Chooses an action
- Resolves Non-determinism
Partial Observation Energy Game

- Chooses an action
- Keeps the energy level positive always at infinite play
- Resolves Non-determinism
- Makes the energy level negative at some point
Partial Observation Energy Game

- Chooses an action
- Keeps the energy level positive always at infinite play
- Resolves Non-determinism
- Makes the energy level negative at some point
Lemma

For every SOCAP $\mathcal{A} \Rightarrow$ an POEG $\mathcal{G}$ such that for all valuations $V$ there exists a reaching run of $\mathcal{A}$ iff Eve has a winning strategy in $\mathcal{G}$. 

Idea:

▶ Adam chooses valuation and Eve has to simulate reaching run for that.

▶ We create gadgets in the game simulating $\text{Op}$. 

▶ Using partial observation, Adam can force Eve to defeat in a gadget if she cheats. 

Theorem

SynthReach problem for SOCAP is decidable.
SOCAP to POEG

Lemma

For every SOCAP $\mathcal{A} \Rightarrow$ an POEG $\mathcal{G}$ such that for all valuations $V$ there exists a reaching run of $\mathcal{A}$ iff Eve has a winning strategy in $\mathcal{G}$.

Idea:

Adam chooses valuation and Eve has to simulate reaching run for that.
Lemma

For every SOCAP $\mathcal{A} \Rightarrow$ an POEG $\mathcal{G}$ such that for all valuations $V$ there exists a reaching run of $\mathcal{A}$ iff Eve has a winning strategy in $\mathcal{G}$.

Idea:

- Adam chooses valuation and Eve has to simulate reaching run for that.
- We create gadgets in the game simulating $Op$. 
SOCAP to POEG

Lemma

For every SOCAP $\mathcal{A}$ there exists an POEG $\mathcal{G}$ such that for all valuations $V$ there exists a reaching run of $\mathcal{A}$ iff Eve has a winning strategy in $\mathcal{G}$.

Idea:

- Adam chooses valuation and Eve has to simulate reaching run for that.
- We create gadgets in the game simulating $Op$.
- Using partial observation, Adam can force Eve to defeat in a gadget if she cheats.
Lemma

For every SOCAP $\mathcal{A} \Rightarrow$ an POEG $\mathcal{G}$ such that for all valuations $V$ there exists a reaching run of $\mathcal{A}$ iff Eve has a winning strategy in $\mathcal{G}$.

Idea:

- Adam chooses valuation and Eve has to simulate reaching run for that.
- We create gadgets in the game simulating $Op$.
- Using partial observation, Adam can force Eve to defeat in a gadget if she cheats.

Theorem

$\text{SYNTHREACH problem for SOCAP is decidable.}$
Decision Problems

Non-Parametric:

Reach

$\exists \rho$ such that

$(q_{in}, 0) \xrightarrow{\rho} q_f$

[NP-complete]

UnivReach

For all infinite $\rho$,

$(q_{in}, 0) \xrightarrow{\rho} q_f$

[coNP-complete]

Parametric:

Par-Reach

$\exists V(X)$ such that $\exists \rho$,

$(q_{in}, 0) \xrightarrow{\rho} V q_f$

[in NEXPTIME]

SynthReach

$\exists V(X)$ such that for all infinite $\rho$,

$(q_{in}, 0) \xrightarrow{\rho} V q_f$

[Decidable]

Ritam Raha 26/26
Clarified the decidability status for the synthesis problem for OCA.

If parameters are only allowed on tests, the problem is in $\text{NP}^{\text{NP}}$.

In full generality, for SOCA it is decidable.
Clarified the decidability status for the synthesis problem for OCA.

If parameters are only allowed on tests, the problem is in $\text{NP}^\text{NP}$.

In full generality, for SOCA it is decidable.

Approach with Multi Energy Game with resets and transfers- Better Complexity?