Computing Linear Arithmetic Representation of Reachability Relation of One-counter Automata

Authors: **Xie Li**, Taolue Chen, Zhilin Wu and Mingji Xia

SETTA 2020: Guangzhou, China, November 24-28, 2020

ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences

Birkbeck UNIVERSITY OF LONDON

中国科学院大学 University of Chinese Academy of Sciences

Overview

- Introduction to One-counter Automata(OCA) and its Reachability Relation.

Overview

- Introduction to One-counter Automata(OCA) and its Reachability Relation.
- Computing the Reachability Relation of OCA.

Overview

- Introduction to One-counter Automata(OCA) and its Reachability Relation.
- Computing the Reachability Relation of OCA.
- Tool OCAReach and Experimental Results.

What is One-counter Automata(OCA)

- DFA with a **counter** c where c is a **non-negative** integer.

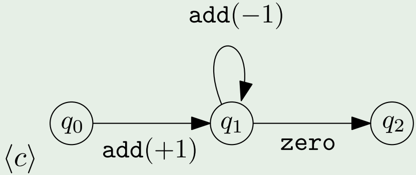
What is One-counter Automata(OCA)

- DFA with a **counter** c where c is a **non-negative** integer.
- Transitions: $q \xrightarrow{\mathtt{Op}} q'$ where $\mathtt{Op} \in \{\mathtt{add}(+1), \mathtt{add}(-1), \mathtt{zero}\}$

What is One-counter Automata(OCA)

- DFA with a **counter** c where c is a **non-negative** integer.
- Transitions: $q \xrightarrow{\texttt{Op}} q'$ where $\texttt{Op} \in \{\texttt{add}(+1), \texttt{add}(-1), \texttt{zero}\}$

Example (OCA)



Semantic of OCA

Semantic of OCA: A transition system.

- Configuration: (q, c).
- Transitions of configurations corresponds to the transitions in the OCA.

$$(q_1, c_1) \rightarrow_{\mathcal{A}} (q_2, c_2)$$

if $q_1 \xrightarrow{\texttt{add}(+1)} q_2$ in the OCA and $c_1 + 1 = c2$ and $\texttt{add}(-1)$ vice versa, or
if $q_1 \xrightarrow{\texttt{zero}} q_2$ and $c_1 = c_2 = 0$.
$c_1, c_2 \geq 0$

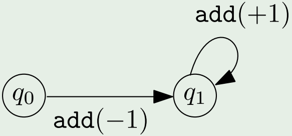
Reachability Problem

Reachability Problem of OCA: whether $(q_s, c_s) \to_{\mathcal{A}}^{*} (q_t, c_t)$

Reachability Problem

Reachability Problem of OCA: whether $(q_s, c_s) \rightarrow^*_{\mathcal{A}} (q_t, c_t)$

Example



Due to the non-negative requirement,
$(q_1, 1)$ is not reachable from $(q_0, 0)$

Reachability Relation

Reachability Problem of OCA: whether $(q_s, c_s) \to_{\mathcal{A}}^{*} (q_t, c_t)$
Instead of using concrete values c_s and c_t, we use variables x_s and x_t for the reachability relation.

Reachability Relation

Reachability Problem of OCA: whether $(q_s, c_s) \rightarrow^*_{\mathcal{A}} (q_t, c_t)$
Instead of using concrete values c_s and c_t, we use variables x_s and x_t for the reachability relation.

Definition (Reachability Relation of OCA)

A reachability relation of an OCA $\mathcal{A}$ from state q_s to q_t is a set $R_{\mathcal{A},q_s,q_t} \subseteq \mathbb{N} \times \mathbb{N}$.

$$\forall (c_s, c_t) \in R_{\mathcal{A},q_s,q_t}.(q_s, c_s) \rightarrow^*_{\mathcal{A}} (q_t, c_t)$$

Reachability Relation

Reachability Problem of OCA: whether $(q_s, c_s) \rightarrow^*_{\mathcal{A}} (q_t, c_t)$
Instead of using concrete values c_s and c_t, we use variables x_s and x_t for the reachability relation.

Definition (Reachability Relation of OCA)

A reachability relation of an OCA $\mathcal{A}$ from state q_s to q_t is a set $R_{\mathcal{A},q_s,q_t} \subseteq \mathbb{N} \times \mathbb{N}$.

$$\forall (c_s, c_t) \in R_{\mathcal{A},q_s,q_t}.(q_s, c_s) \rightarrow^*_{\mathcal{A}} (q_t, c_t)$$

Goal: use a Presburger Arithmetic (PA) formula $\phi(x_s, x_t)_{\mathcal{A},q_s,q_t}$ to represent this relation.

Weighted Directed Graph, Path and Flow

- An OCA can be regarded as a weighted directed graph $G_{\mathcal{A}} = (V, E, \eta)$.

Weighted Directed Graph, Path and Flow

- An OCA can be regarded as a weighted directed graph $G_{\mathcal{A}} = (V, E, \eta)$.
- **Path**: a sequence of vertices $v_0 \cdot v_1 \cdots v_k$ where $(v_i, v_{i+1}) \in E$.

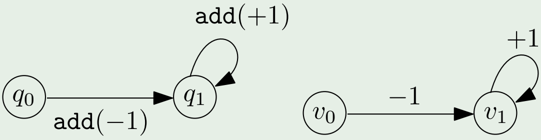
Weighted Directed Graph, Path and Flow

- An OCA can be regarded as a weighted directed graph $G_{\mathcal{A}} = (V, E, \eta)$.
- **Path**: a sequence of vertices $v_0 \cdot v_1 \cdots v_k$ where $(v_i, v_{i+1}) \in E$.
  - Drop of path

Weighted Directed Graph, Path and Flow

- An OCA can be regarded as a weighted directed graph $G_{\mathcal{A}} = (V, E, \eta)$.
- **Path**: a sequence of vertices $v_0 \cdot v_1 \cdots v_k$ where $(v_i, v_{i+1}) \in E$.
  - Drop of path
- **Flow**: a function $f : E \to \mathbb{N}$.

Example



- path: $v_0 \cdot v_1 \cdot v_1 \cdot v_1$
- drop: -1

Path Flow and Support

- Support: edge-induced subgraph of flow.
- s-t **path flow**: the flow corresponds to a path.
    - Requirements of flow of each vertex.
    - Connectivity of the support.

Path Flow and Support

- Support: edge-induced subgraph of flow.
- s-t **path flow**: the flow corresponds to a path.
  - Requirements of flow of each vertex.
  - Connectivity of the support.

$$\phi^{pf}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{f_{e \in E}} := weight(f) = x_t - x_s \wedge$$

$$\textbf{if } (s = t) \textbf{ then foreach } v \in V : \texttt{num}(\text{in-flows})_v = \texttt{num}(\text{out-flows})_v$$

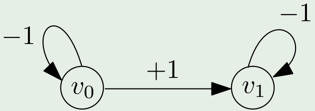
$$\textbf{else foreach } v \in V - \{s, t\} : (\texttt{num}(\text{in-flows})_v = \texttt{num}(\text{out-flows})_v \wedge$$

$$\texttt{num}(\text{in-flows})_s = \texttt{num}(\text{out-flows})_s - 1 \wedge$$

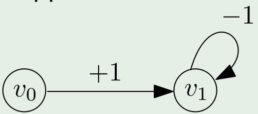
$$\texttt{num}(\text{in-flows})_t = \texttt{num}(\text{out-flows})_t + 1)$$

Example of Path Flow

Example



- Support:



- Path: $v_0 \cdot v_1 \cdot v_1 \cdot v_1$
- Pathflow: $f(v_0, v_0) = 0$
  $f(v_0, v_1) = 1$
  $f(v_1, v_1) = 2$
- Weight: $weight(f) = \Sigma_{e \in E} f(e) \cdot \eta(e)$

The Difficulty of Computing the Reachability Relation

NON-NEGATIVE

The Difficulty of Computing the Reachability Relation

NON-NEGATIVE

If we do not require the non-negative of counter.

- Existence of a path flow.

The Difficulty of Computing the Reachability Relation

NON-NEGATIVE

If we do not require the non-negative of counter.

- Existence of a path flow.

$$\phi_{G_{\mathcal{A}},s,t}(x_s, x_t)_{f_{e\in E}}$$

Non-negative implies the constraint: everywhere along the path, the counter need to be non-negative.

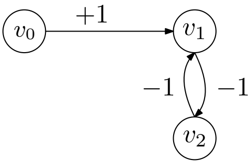
Certificates of the Reachability

Use **path flow** as certificate of OCA reachability problem.

Certificates of the Reachability

Use **path flow** as certificate of OCA reachability problem.

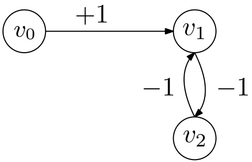
- Type-1 Certificate:
  - Flow is a path flow.
  - No positive cycle.
  - Path flow has edge decompositions (which implies **non-negative**).



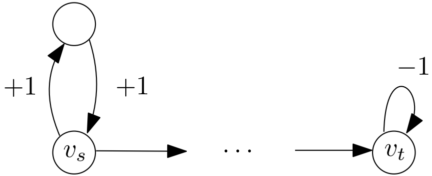
Certificates of the Reachability

Use **path flow** as certificate of OCA reachability problem.

- Type-1 Certificate:
  - Flow is a path flow.
  - No positive cycle.
  - Path flow has edge decompositions (which implies **non-negative**).



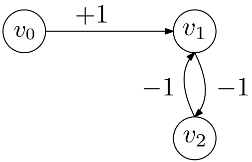
- Type-3 Certificate:



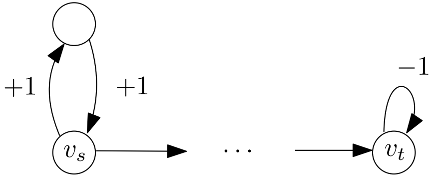
Certificates of the Reachability

Use **path flow** as certificate of OCA reachability problem.

- Type-1 Certificate:
  - Flow is a path flow.
  - No positive cycle.
  - Path flow has edge decompositions (which implies **non-negative**).



- Type-3 Certificate:



- Type-2 Certificate: Dual of type-1 certificate at the end.

Edge Decomposition

Definition (Edge Decomposition)

Edge decomposition of a path flow f is

Edge Decomposition

Definition (Edge Decomposition)

Edge decomposition of a path flow f is

- $f = \Sigma_{i \in |E|} f_i + f_{e_i}$ where f_i is also a path flow, $e_i \in E$.

Edge Decomposition

Definition (Edge Decomposition)

Edge decomposition of a path flow f is

- $f = \Sigma_{i \in |E|} f_i + f_{e_i}$ where f_i is also a path flow, $e_i \in E$.
- e_i does not appear in support of f_j where $j > i$.

Edge Decomposition

Definition (Edge Decomposition)

Edge decomposition of a path flow f is

- $f = \Sigma_{i \in |E|} f_i + f_{e_i}$ where f_i is also a path flow, $e_i \in E$.
- e_i does not appear in support of f_j where $j > i$.
- $weight(f_i) + weight(e_i) \geq 0$ for all i.

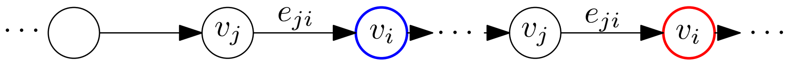
Edge Decomposition

Definition (Edge Decomposition)

Edge decomposition of a path flow f is

- $f = \Sigma_{i \in |E|} f_i + f_{e_i}$ where f_i is also a path flow, $e_i \in E$.
- e_i does not appear in support of f_j where $j > i$.
- $weight(f_i) + weight(e_i) \geq 0$ for all i.

This definition implies the non-negative requirement of Type-1 certificate.



Decidability of Reachability of OCA

Theorem (Haase)

The reachability problem of OCA can be solved iff we can find a certificate that is of the form

$$(\textit{Type-1})^{n_1}(\textit{Type-3})^{n_3}(\textit{Type-2})^{n_2}$$

where $n_i \in \{0, 1\}$

Use PA Formula to Represent Type-3 Reachability Relation

$$\phi_{G_{\mathcal{A}},s,t}(x_s, x_t) = \exists (f_e)_{e\in E}.\phi^{T1RC} \vee \phi^{T2RC} \vee \phi^{T3RC} \vee \cdots$$

Type-3 certificate:

$$\phi^{T3RC}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{(f_{e,3})_{e\in E}}$$

- Positive Cycle at q_s.
- Existence of a q_s-q_t path flow.
- Negative Cycle at q_t.

Use PA Formula to Represent Type-1 Reachability Relation

$$\phi^{T1RC}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{(f_{e,1})_{e\in E}} := \exists (idx_e, sum_e)_{e\in E}$$

Use PA Formula to Represent Type-1 Reachability Relation

$$\phi^{T1RC}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{(f_{e,1})_{e\in E}} := \exists (idx_e, sum_e)_{e\in E}$$

$$\phi^{pf}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{f_{e\in E}} \wedge$$

Use PA Formula to Represent Type-1 Reachability Relation

$$\phi^{T1RC}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{(f_{e,1})_{e\in E}} := \exists(idx_e, sum_e)_{e\in E}$$

$$\phi^{pf}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{f_{e\in E}} \wedge \phi^{APC}(f_{e\in E}) \wedge$$

Use PA Formula to Represent Type-1 Reachability Relation

$$\phi^{T1RC}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{(f_{e,1})_{e\in E}} := \exists (idx_e, sum_e)_{e\in E}$$

$$\phi^{pf}_{G_{\mathcal{A}},s,t}(x_s, x_t)_{f_{e\in E}} \wedge \phi^{APC}(f_{e\in E}) \wedge$$

$$\phi^{EDC}((f_e, idx_e, sum_e)_{e\in E}), (f_{e,e'})_{e,e'\in E})$$

OCAReach: Experimental Evaluation

Implemented in Java and utilzing $\mathrm{Z3}$ solver for formula manipulation.

INPUT: file describing the OCA.

OUTPUT: a PA formula ϕ representing reachability relation.

- Experiment on handcrafted cases.

state num.	2	2	2	2	3	3	4	4	4
transtion num.	1	2	2	5	2	3	3	3	6
zero-test num.	0	1	1	0	0	1	1	1	1
time (s)	0.066	0.062	0.078	0.076	0.066	0.072	0.061	0.079	0.093
size (kB)	0.302	0.404	0.697	0.302	0.133	0.929	0.348	0.325	2.592

state num.	5	6	6	6	7	8	10	10	
transtion num.	6	6	7	8	9	7	11	11	
zero-test num.	1	2	2	2	2	2	2	3	
time (s)	0.087	0.078	0.106	0.091	0.106	0.090	0.116	0.117	
size (kB)	2.057	2.469	7.457	3.078	6.427	4.807	8.443	7.515	

- On random cases.

Contributions and Future Work

Contributions:

- Some work to make computation of reachability relation possible.
- We built the gap between the theory and implementation by the tool OCAREACH.

Future work:

- Optimize our tool to improve the efficiency.
- More and larger cases and find benchmarks for experiment.
- Other topics about one-counter automata.

Thanks! & Questions?