Visibly Rational Expressions

Ping Lu

luping@ios.ac.cn

May 20, 2013

Motivation

Regular Language:

- Right-linear grammar (left-linear grammar)
- NFA
- Regular expressions

Motivation

Regular Language:

- Right-linear grammar (left-linear grammar)
- NFA
- Regular expressions

Visibly Pushdown Languages:

- Visibly pushdown grammar
- VPA
- ?

Pushdown Alphabet

A pushdown alphabet $\widetilde{\Sigma} = \{\Sigma_{call}, \Sigma_{ret}, \Sigma_{int}\}$:

- Σ_{call} : a finite set of calls, using symbols like $c, c_1, c_2, \ldots$.
- Σ_{ret} : a finite set of returns, using symbols like $r, r_1, r_2, \ldots$.
- Σ_{int} : a finite set of internal actions, using symbols like $\Box, \Box_1, \Box_2, \ldots$.

Pushdown Alphabet

A pushdown alphabet $\widetilde{\Sigma} = \{\Sigma_{call}, \Sigma_{ret}, \Sigma_{int}\}$:

- Σ_{call} : a finite set of calls, using symbols like $c, c_1, c_2, \ldots$.
- Σ_{ret} : a finite set of returns, using symbols like $r, r_1, r_2, \ldots$.
- Σ_{int} : a finite set of internal actions, using symbols like $\square, \square_1, \square_2, \ldots$.

$\Sigma = \Sigma_{call} \cup \Sigma_{ret} \cup \Sigma_{int}$ is the support of $\widetilde{\Sigma}$.
We use $\sigma, \sigma_1, \ldots$ for arbitrary elements of Σ.

VPA and VPL

A Nondeterministic Visibly Pushdown Automaton on finite word (NVPA) over $\widetilde{\Sigma} = \{\Sigma_{call}, \Sigma_{ret}, \Sigma_{int}\}$ is a tuple $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$, where:

- Q: a finite set of (control) states;
- $q_{in} \in Q$: the initial state;
- $F \subseteq Q$: a set of accepting states;
- $\Delta \subseteq (Q \times \Sigma_{call} \times Q \times \Gamma) \cup (Q \times \Sigma_{ret} \times (\Gamma \cup \{\perp\}) \times Q) \cup (Q \times \Sigma_{int} \times Q)$

VPA and VPL

Configuration: (q, β) s.t. $q \in Q$ and $\beta \in \Gamma^* \cdot \{\bot\}$.

Run π of $\mathcal{P}$ over $\sigma_1 \ldots \sigma_{n-1} \in \Sigma^*$: $(q_1, \beta_1) \xrightarrow{\sigma_1} \ldots \xrightarrow{\sigma_{n-1}} (q_n, \beta_n)$

- (q_i, β_i): Configuration for all $1 \leq i \leq n$;
- The following conditions hold for all $1 \leq i \leq n$:
  - **Push** If σ_i is a call, then for some $\gamma \in \Gamma$, $(q_i, \sigma_i, q_{i+1}, \gamma) \in \Delta$ and $\beta_{i+1} = \gamma \cdot \beta_i$.
  - **Pop** If σ_i is a return, then for some $\gamma \in \Gamma \cup \{\bot\}$, $(q_i, \sigma_i, \gamma, q_{i+1}) \in \Delta$, and either $\gamma \neq \bot$ and $\beta_i = \gamma \cdot \beta_{i+1}$, or $\gamma = \bot$ and $\beta_i = \beta_{i+1} = \bot$.
  - **Internal** If σ_i is an internal action, then $(q_i, \sigma_i, q_{i+1}) \in \Delta$ and $\beta_{i+1} = \beta_i$.

VPA and VPL

For $1 \leq i \leq j \leq n$, $\pi_{ij} = (q_i, \beta_i) \xrightarrow{\sigma_i} \ldots \xrightarrow{\sigma_{j-1}} (q_j, \beta_j)$ is a subrun of π.

The run π is initialized if $q_1 = q_{in}$ and $\beta_1 = \perp$.

The run is accepting if $q_n \in F$.

VPA and VPL

For $1 \leq i \leq j \leq n$, $\pi_{ij} = (q_i, \beta_i) \xrightarrow{\sigma_i} \dots \xrightarrow{\sigma_{j-1}} (q_j, \beta_j)$ is a subrun of π.

The run π is initialized if $q_1 = q_{in}$ and $\beta_1 = \perp$.

The run is accepting if $q_n \in F$.

$\mathcal{L}(\mathcal{P})$: $\{w \in \Sigma^* |$ there is an initialized accepting run of $\mathcal{P}$ on $w\}$.

$\mathcal{L} \subseteq \Sigma^*$ is a visibly pushdown language (VPL) with respect to $\widetilde{\Sigma}$:
$\exists \mathcal{P}$ over $\widetilde{\Sigma}$ s.t. $\mathcal{L} = \mathcal{L}(\mathcal{P})$.

VPA and VPL

The visibly pushdown automata on infinite words (ω-NVPA):

- Büchi ω-NVPA over $\widetilde{\Sigma}$: $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$.

VPA and VPL

The visibly pushdown automata on infinite words (ω-NVPA):

- Büchi ω-NVPA over $\widetilde{\Sigma}$: $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$.
- Run π over an infinite word $\sigma_1 \sigma_2 \ldots \in \Sigma^\omega$: $(q_1, \beta_1) \xrightarrow{\sigma_1} (q_2, \beta_2) \ldots$.
- The run is accepting: for infinitely many $i \geq 1$, $q_i \in F$.

VPA and VPL

The visibly pushdown automata on infinite words (ω-NVPA):

- Büchi ω-NVPA over $\widetilde{\Sigma}$: $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$.
- Run π over an infinite word $\sigma_1 \sigma_2 \ldots \in \Sigma^\omega$: $(q_1, \beta_1) \xrightarrow{\sigma_1} (q_2, \beta_2) \ldots$.
- The run is accepting: for infinitely many $i \geq 1$, $q_i \in F$.
- ω-language of $\mathcal{P}$: infinite words $w \in \Sigma^\omega$ s.t. there is an initialized accepting run of $\mathcal{P}$ on w.

- ω-language $\mathcal{L}$ is an ω-visibly pushdown language (ω-VPL) with respect to $\widetilde{\Sigma}$:

there ia a Büchi ω-NVPA $\mathcal{P}$ over $\widetilde{\Sigma}$ such that $\mathcal{L} = \mathcal{L}(\mathcal{P})$.

Matched calls and returns

Fix a pushdown alphabet $\widetilde{\Sigma} = \{\Sigma_{call}, \Sigma_{ret}, \Sigma_{int}\}$.
The well-matched words $WM(\widetilde{\Sigma})$ is defined as:

- $\varepsilon \in WM(\widetilde{\Sigma})$;
- $\Box \cdot w \in WM(\widetilde{\Sigma})$, if $\Box \in \Sigma_{int}$ and $w \in WM(\widetilde{\Sigma})$;
- $c \cdot w \cdot r \cdot w' \in WM(\widetilde{\Sigma})$, if $c \in \Sigma_{call}$, $r \in \Sigma_{ret}$, and $w, w' \in WM(\widetilde{\Sigma})$.

Matched calls and returns

Fix a pushdown alphabet $\widetilde{\Sigma} = \{\Sigma_{call}, \Sigma_{ret}, \Sigma_{int}\}$.
The well-matched words $WM(\widetilde{\Sigma})$ is defined as:

- $\varepsilon \in WM(\widetilde{\Sigma})$;
- $\Box \cdot w \in WM(\widetilde{\Sigma})$, if $\Box \in \Sigma_{int}$ and $w \in WM(\widetilde{\Sigma})$;
- $c \cdot w \cdot r \cdot w' \in WM(\widetilde{\Sigma})$, if $c \in \Sigma_{call}$, $r \in \Sigma_{ret}$, and $w, w' \in WM(\widetilde{\Sigma})$.

The minimally well-matched words $MWM(\widetilde{\Sigma})$ is defined as:
$c \cdot w \cdot r$, if $c \in \Sigma_{call}$, $r \in \Sigma_{ret}$, and $w \in WM(\widetilde{\Sigma})$.

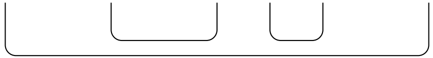
For a language $\mathcal{L} \subseteq \Sigma^*$, $MWM(\mathcal{L}) \stackrel{def}{=\!=} \mathcal{L} \cap MWM(\widetilde{\Sigma})$.

An example

Example

Let $\Sigma_{call} = \{c\}$, $\Sigma_{ret} = \{r\}$, and $\Sigma_{int} = \{\Box\}$. Consider

		1	2	3	4	5	6	7	8	9	10
w	$=$	c	c	$\Box$	c	$\Box$	r	c	r	$\Box$	r

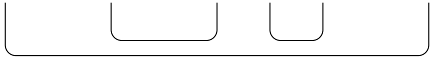


An example

Example

Let $\Sigma_{call} = \{c\}$, $\Sigma_{ret} = \{r\}$, and $\Sigma_{int} = \{\Box\}$. Consider

$$\begin{array}{ccccccccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ w & = & c & c & \Box & c & \Box & r & c & r & \Box & r \end{array}$$



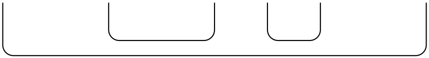
Note that w is not well-matched.

An example

Example

Let $\Sigma_{call} = \{c\}$, $\Sigma_{ret} = \{r\}$, and $\Sigma_{int} = \{\square\}$. Consider

		1	2	3	4	5	6	7	8	9	10
w	$=$	c	c	$\square$	c	$\square$	r	c	r	$\square$	r



Note that w is not well-matched.

The subword $w[2] \ldots w[10]$ is minimally well-matched.

M-substitution

Definition

(M-substitution) Let $w \in \Sigma^*, \Box \in \Sigma_{int}$, and $\mathcal{L} \subseteq \Sigma^*$. The M-substitution of $\Box$ by $\mathcal{L}$ in w, denoted by $w \curvearrowleft_{\Box} \mathcal{L}$, is defined as follows:

- $\varepsilon \curvearrowleft_{\Box} \mathcal{L} \overset{def}{=\!=} \{\varepsilon\}$
- $(\Box \cdot w') \curvearrowleft_{\Box} \mathcal{L} \overset{def}{=\!=} (MWM(\mathcal{L}) \cdot (w' \curvearrowleft_{\Box} \mathcal{L})) \cup ((\{\Box\} \cap \mathcal{L}) \cdot (w' \curvearrowleft_{\Box} L))$
- $(\sigma \cdot w') \curvearrowleft_{\Box} \mathcal{L} \overset{def}{=\!=} \{\sigma\} \cdot (w' \curvearrowleft_{\Box} \mathcal{L}))$ for each $\sigma \in \Sigma \setminus \{\Box\}$.

M-substitution

For two languages $\mathcal{L}, \mathcal{L}' \subseteq \Sigma^*$ and $\Box \in \Sigma_{int}$

M-substitution of $\Box$ *by* $\mathcal{L}'$ *in* $\mathcal{L}$:

$$\mathcal{L} \curvearrowleft_{\Box} \mathcal{L}' \stackrel{def}{=\!=} \bigcup_{w \in \mathcal{L}} w \curvearrowleft_{\Box} \mathcal{L}'.$$

If $\{\Box\} \cap \mathcal{L} = \emptyset$, then $\{\Box\} \curvearrowleft_{\Box} \mathcal{L} = MWM(\mathcal{L})$.

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{c_1^n \Box\Box r^n | n \geq 1\}$ and $\mathcal{L}' = \{c_2\}^* \cdot \{r\}^*$.
Then $\mathcal{L} \curvearrowleft_{\Box} \mathcal{L}' =?$.

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{c_1^n \Box\Box r^n | n \geq 1\}$ and $\mathcal{L}' = \{c_2\}^* \cdot \{r\}^*$.
Then $\mathcal{L} \curvearrowleft_{\Box} \mathcal{L}' = \{c_1^n c_2^m r^m c_2^k r^k r^n | n, m, r \geq 1\}$.

Associative

Theorem

$\curvearrowleft_{\Box}$ *is associative.*

Theorem

If $\Box \notin L(L')$, $MWM(L') \curvearrowleft_{\Box} L'' = MWM(L' \curvearrowleft_{\Box} L'')$

Associative

Theorem

$\curvearrowleft_{\Box}$ *is associative.*

Theorem

If $\Box \notin L(L')$, $MWM(L') \curvearrowleft_{\Box} L'' = MWM(L' \curvearrowleft_{\Box} L'')$

Proof.

$(\subseteq)$ Let $w \in MWM(L') \curvearrowleft_{\Box} L''$.

$\exists r w_1 c \in MWM(L')$ s.t. w_1 is well-matched and $w \in c(w_1 \curvearrowleft_{\Box} L'')r$.

w_1 is well-matched

$\Rightarrow w \curvearrowleft_{\Box} L''$ are also well-matched

$\Rightarrow c(w_1 \curvearrowleft_{\Box} L'')r \subseteq MWM(L' \curvearrowleft_{\Box} L'')$

$\Rightarrow w \in MWM(L' \curvearrowleft_{\Box} L'')$. □

Associative

Theorem

$\curvearrowleft_{\square}$ is associative.

Theorem

If $\square \notin L(L')$, $MWM(L') \curvearrowleft_{\square} L'' = MWM(L' \curvearrowleft_{\square} L'')$

Proof.

($\supseteq$) Let $w \in MWM(L' \curvearrowleft_{\square} L'')$.

$\exists r w_1 c \in L'$ s.t. $w \in c(w_1 \curvearrowleft_{\square} L'')r$, and $w_1 \curvearrowleft_{\square} L''$ are well-matched.
$\Rightarrow w_1$ is well-matched
$\Rightarrow c w_1 r \in MWM(L')$
$\Rightarrow w \in MWM(L') \curvearrowleft_{\square} L''$.

$\square$

M-closure and S-closure

Definition

(M-closure and S-closure) Given $\mathcal{L} \subseteq \Sigma^*$ and $\Box \in \Sigma_{int}$, the *M*-closure of $\mathcal{L}$ through $\Box$, written by $\mathcal{L}^{\curvearrowleft\Box}$, is defined as:

$$\mathcal{L}^{\curvearrowleft\Box} \overset{def}{=\!=} \bigcup_{n \geq 0} \mathcal{L} \underbrace{\curvearrowleft_{\Box} (\mathcal{L} \cup \{\Box\}) \curvearrowleft_{\Box} \ldots \curvearrowleft_{\Box} (\mathcal{L} \cup \{\Box\})}_{n\ occurrences\ of\ \curvearrowleft_{\Box}}.$$

The *S*-closure of $\mathcal{L}$ through $\Box$, written by $\mathcal{L}^{\circlearrowleft\Box}$, is defined as:
$\mathcal{L}^{\circlearrowleft\Box} \overset{def}{=\!=} MWM(\mathcal{L})^{\curvearrowleft\Box}$.

M-closure and S-closure

Definition

(M-closure and S-closure) Given $\mathcal{L} \subseteq \Sigma^*$ and $\Box \in \Sigma_{int}$, the *M*-closure of $\mathcal{L}$ through $\Box$, written by $\mathcal{L}^{\curvearrowleft\Box}$, is defined as:

$$\mathcal{L}^{\curvearrowleft\Box} \stackrel{def}{=\!=} \bigcup_{n \geq 0} \mathcal{L} \underbrace{\curvearrowleft_\Box (\mathcal{L} \cup \{\Box\}) \curvearrowleft_\Box \ldots \curvearrowleft_\Box (\mathcal{L} \cup \{\Box\})}_{n \ occurrences \ of \ \curvearrowleft_\Box}.$$

The *S*-closure of $\mathcal{L}$ through $\Box$, written by $\mathcal{L}^{\circlearrowleft\Box}$, is defined as:

$$\mathcal{L}^{\circlearrowleft\Box} \stackrel{def}{=\!=} MWM(\mathcal{L})^{\curvearrowleft\Box}.$$

Relations of the operators:

$$\mathcal{L}^{\curvearrowleft\Box} = \mathcal{L} \curvearrowleft_\Box (\mathcal{L}^{\circlearrowleft\Box} \cup \{\Box\}).$$

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r_1, r_2\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{\Box, c_1\Box r_1, c_2\Box r_2\}$ and $\mathcal{L}' = \{c_1 r_1, c_2 r_2\}$.
Then $\mathcal{L}^{\curvearrowleft_\Box} \curvearrowleft_\Box \mathcal{L}' =?$.

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r_1, r_2\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{\Box, c_1\Box r_1, c_2\Box r_2\}$ and $\mathcal{L}' = \{c_1 r_1, c_2 r_2\}$.
Then $\mathcal{L}^{\curvearrowleft_\Box} \curvearrowleft_\Box \mathcal{L}' = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | n \geq 1, i_1, \ldots, i_n \in \{1, 2\}\}$.

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r_1, r_2\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{\Box, c_1\Box r_1, c_2\Box r_2\}$ and $\mathcal{L}' = \{c_1 r_1, c_2 r_2\}$.
Then $\mathcal{L}^{\curvearrowleft_\Box} \curvearrowleft_\Box \mathcal{L}' = \{c_{i_1} c_{i_2} \dots c_{i_n} r_{i_n} \dots r_{i_2} r_{i_1} | n \geq 1, i_1, \dots, i_n \in \{1, 2\}\}$.
There is no regular expression E s.t. $\mathcal{L}(E) = \mathcal{L}^{\curvearrowleft_\Box} \curvearrowleft_\Box \mathcal{L}'$

Pumping Lemma

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r_1, r_2\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{\Box, c_1\Box r_1, c_2\Box r_2\}$ and $\mathcal{L}' = \{c_1 r_1, c_2 r_2\}$.
Then $\mathcal{L}^{\curvearrowleft\Box} \curvearrowleft_{\Box} \mathcal{L}' = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | n \geq 1, i_1, \ldots, i_n \in \{1,2\}\}$.
There is no regular expression E s.t $MWM(\mathcal{L}(E)) = \mathcal{L}^{\curvearrowleft\Box} \curvearrowleft_{\Box} \mathcal{L}'$?

An example

Example

Let $\Sigma_{call} = \{c_1, c_2\}$, $\Sigma_{ret} = \{r_1, r_2\}$, and $\Sigma_{int} = \{\Box\}$.
$\mathcal{L} = \{\Box, c_1\Box r_1, c_2\Box r_2\}$ and $\mathcal{L}' = \{c_1 r_1, c_2 r_2\}$.
Then $\mathcal{L}^{\curvearrowright\Box} \curvearrowleft_{\Box} \mathcal{L}' = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | n \geq 1, i_1, \ldots, i_n \in \{1, 2\}\}$.
There is no regular expression E s.t $MWM(\mathcal{L}(E)) = \mathcal{L}^{\curvearrowright\Box} \curvearrowleft_{\Box} \mathcal{L}'$?

Suppose $\mathcal{N} = (Q, \Sigma, \delta, q_0, F)$ *s.t.* $MWM(\mathcal{L}(\mathcal{N})) = \mathcal{L}^{\curvearrowright\Box} \curvearrowleft_{\Box} \mathcal{L}'$.
Let $n = |Q|$, $|\{c_{i_1} c_{i_2} \ldots c_{i_n} | c_{i_j} \in \{1, 2\}\}| = 2^n$.
$\{q' | q' \in Q \text{ and } \delta(q_0, c_{i_1} c_{i_2} \ldots c_{i_n}) = q'\} \subseteq Q$.
$\exists c_{i_1} c_{i_2} \ldots c_{i_n}, c_{i'_1} c_{i'_2} \ldots c_{i'_n}$ *s.t.* $\delta(q_0, c_{i_1} c_{i_2} \ldots c_{i_n}) = \delta(q_0, c_{i'_1} c_{i'_2} \ldots c_{i'_n})$.

Since $c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} r_{i_{n-1}} \ldots r_{i_1} \in \mathcal{L}(\mathcal{N})$ *and* $c_{i'_1} c_{i'_2} \ldots c_{i'_n} r_{i'_n} r_{i'_{n-1}} \ldots r_{i'_1} \in \mathcal{L}(\mathcal{N})$,
$c_{i_1} c_{i_2} \ldots c_{i_n} r_{i'_n} r_{i'_{n-1}} \ldots r_{i'_1} \in \mathcal{L}(\mathcal{N})$.
Hence, $c_{i_1} c_{i_2} \ldots c_{i_n} r_{i'_n} r_{i'_{n-1}} \ldots r_{i'_1} \in MWM(\mathcal{L}(\mathcal{N}))$, *but*
$c_{i_1} c_{i_2} \ldots c_{i_n} r_{i'_n} r_{i'_{n-1}} \ldots r_{i'_1} \notin \mathcal{L}^{\curvearrowright\Box} \curvearrowleft_{\Box} \mathcal{L}'$.

Closure property

Theorem

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ and $\mathcal{P}' = \langle Q', q'_{in}, \Gamma', \Delta', F' \rangle$ be two NVPA over $\widetilde{\Sigma}$, and $\square \in \Sigma_{int}$. Then, one can construct in polynomial time:

- *1. an NVPA accepting $(L(\mathcal{P}))^{\circlearrowleft_\square}$ with $|Q| + 2$ states and $|\Gamma| \cdot (|Q| + 2)$ stack symbols.*
- *2. an NVPA accepting $L(\mathcal{P}) \curvearrowleft_\square L(P')$ with $|Q| + |Q'|$ states and $|\Gamma| + |\Gamma'| \cdot (|Q| + 1)$ stack symbols.*
- *3. an NVPA accepting $(L(\mathcal{P}))^{\curvearrowleft_\square}$ with $2|Q| + 2$ states and $2|\Gamma| \cdot (|Q| + 1)$ stack symbols.*

Closure property: Construction

Proof.

At first, we show how to construct an NVPA $\mathcal{P}' = \langle Q', q'_{in}, \Gamma \cup \widehat{\Gamma}, \Delta', F' \rangle$ accepting $MWM(L(\mathcal{P}))$. $\mathcal{P}'$ is defined as follows:

- $Q' = \{q'_{in}, q_f\} \cup Q$.
- $F' = \{q_f\}$.
- $\Delta' = (\Delta \cup (\{(q'_{in}, \sigma, q', \widehat{\gamma}) | (q_{in}, \sigma, q, \gamma) \in \Delta, \text{and } \sigma \in \Sigma_{call}\}$ $\bigcup \{(q, \sigma, \widehat{\gamma}, q_f) | (q, \sigma, \gamma, q_1) \in \Delta,\ q_1 \in F,\ \text{and } \sigma \in \Sigma_{ret}\})$

□

Closure property: Construction

1, The NVPA $\mathcal{P}'' = \langle Q', q'_{in}, \Gamma', \Delta'', F' \rangle$, accepting $(L(\mathcal{P}))^{\circlearrowleft\Box}$, can be constructed as follows (Suppose $L(P') = \mathcal{L}(\mathcal{P}) \cap MWM(\widetilde{\Sigma})$):

- $\Gamma' = \Gamma \cup \widehat{\Gamma} \cup Q \times \widehat{\Gamma}$.
- $\Delta'' = \Delta'$

$$\bigcup\{(q_1, \sigma, q_3, (q_2, \widehat{\gamma})) | (q_1, \Box, q_2) \in \Delta'(\Box \in \Sigma_{int}), (q'_{in}, \sigma, q_3, \widehat{\gamma}) \in \Delta', \text{and } \sigma \in \Sigma'_{call}\}$$
$$\bigcup\{(q, \sigma, (q_2, \widehat{\gamma}), q_2) | (q, \sigma, \widehat{\gamma}, q_f) \in \Delta', \text{ and } \sigma \in \Sigma'_{ret}\}$$

Closure property: Construction

2, An NVPA $\mathcal{P}'' = \langle Q_2, q_{in}, \Gamma_2, \Delta_2, F\rangle$, accepting $L(\mathcal{P}) \curvearrowleft_{\Box} L(P')$, can be constructed as follows (Suppose $L(P') \subseteq MWM(\widetilde{\Sigma})$, $Q \cap Q' = \emptyset$, and $\Gamma \cap \Gamma' = \emptyset$):

- $Q_2 = Q \cup Q'$
- $\Gamma_2 = \Gamma \cup \Gamma' \cup Q \times \Gamma'$
- $\Delta_2 = (\Delta \setminus \{(q_1, \Box, q_2) | q_1, q_2 \in Q\}) \cup \Delta'$
  $\bigcup \{(q_1, \sigma, q_3, (q_2, \widehat{\gamma})) | (q_1, \Box, q_2) \in \Delta,\ (q'_{in}, \sigma, q_3, \widehat{\gamma}) \in \Delta', \text{and } \sigma \in \Sigma_{call}\}$
  $\bigcup \{(q, \sigma, (q_2, \widehat{\gamma}), q_2) | (q, \sigma, \widehat{\gamma}, q_f) \in \Delta',\ q_2 \in Q,\ \text{and } \sigma \in \Sigma_{ret}\}$
  $\bigcup \{(q_1, \Box, q_2) | (q_1, \Box, q_2) \in \Delta, (q'_{in}, \Box, q) \in \Delta', q \in F'\}$

Closure property: Construction

3, An NVPA $\mathcal{P}'' = \langle Q_2, q_{in}, \Gamma_2, \Delta_2, F \rangle$, accepting $(L(\mathcal{P}))^{\curvearrowleft \Box}$, can be constructed as follows (Suppose $L(P') = \mathcal{L}(\mathcal{P}) \cap MWM(\widetilde{\Sigma})$, $Q \cap Q' = \emptyset$, and $\Gamma \cap \Gamma' = \emptyset$):

- $Q_2 = Q \cup Q'$
- $\Gamma_2 = \Gamma \cup \widehat{\Gamma} \cup Q \times \Gamma \cup Q \times \widehat{\Gamma}$
- $\Delta_2 = \Delta \cup \Delta'$
  $\bigcup \{(q_1, \sigma, q_3, (q_2, \widehat{\gamma})) | (q_1, \Box, q_2) \in \Delta,\ (q'_{in}, \sigma, q_3, \widehat{\gamma}) \in \Delta', \text{and } \sigma \in \Sigma'_{call}\}$
  $\bigcup \{(q_1, \sigma, q_3, (q_2, \gamma)) | (q_1, \Box, q_2) \in \Delta',\ (q'_{in}, \sigma, q_3, \widehat{\gamma}) \in \Delta', \text{and } \sigma \in \Sigma'_{call}\}$
  $\bigcup \{(q, \sigma, (q_2, \widehat{\gamma}), q_2) | (q, \sigma, \widehat{\gamma}, q_f) \in \Delta' \text{ and } \sigma \in \Sigma'_{ret}\}$

VRE

Definition

(VRE). The syntax of VRE E over the pushdown alphabet $\widetilde{\Sigma}$ is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\Box} E) \mid E^{\circlearrowleft_{\Box}} \mid E^{\curvearrowleft_{\Box}}$$

where $\sigma \in \Sigma$ and $\Box \in \Sigma_{int}$.

VRE

Definition

(VRE). The syntax of VRE E over the pushdown alphabet $\widetilde{\Sigma}$ is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}} \mid E^{\curvearrowleft_{\square}}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

A pure VRE is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

VRE

Definition

(VRE). The syntax of VRE E over the pushdown alphabet $\widetilde{\Sigma}$ is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}} \mid E^{\curvearrowleft \square}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

A pure VRE is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

The language $\mathcal{L}$ of a VRE E is defined as:

(1) $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$, and $\mathcal{L}(\sigma) = \{\sigma\}$ for each $\sigma \in \Sigma$;

(2) $\mathcal{L}(E_1 \cup E_2) = \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$, $\mathcal{L}(E_1 \cdot E_2) = \mathcal{L}(E_1) \cdot \mathcal{L}(E_2)$, and $\mathcal{L}(E^*) = \mathcal{L}(E_1)^*$;

(3) $\mathcal{L}(E \curvearrowleft_{\square} E) = \mathcal{L}(E_1) \curvearrowleft_{\square} \mathcal{L}(E_2)$, $\mathcal{L}(E^{\circlearrowleft \square}) = [\mathcal{L}(E_1)]^{\circlearrowleft \square}$, and $\mathcal{L}(E^{\curvearrowleft \square}) = [\mathcal{L}(E)]^{\curvearrowleft \square}$

VRE

Definition

(VRE). The syntax of VRE E over the pushdown alphabet $\widetilde{\Sigma}$ is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}} \mid E^{\curvearrowleft \square}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

A pure VRE is defined as:

$$E := \emptyset \mid \varepsilon \mid \sigma \mid (E \cup E) \mid (E \cdot E) \mid E^* \mid (E \curvearrowleft_{\square} E) \mid E^{\circlearrowleft_{\square}}$$

where $\sigma \in \Sigma$ and $\square \in \Sigma_{int}$.

The language $\mathcal{L}$ of a VRE E is defined as:

(1) $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$, and $\mathcal{L}(\sigma) = \{\sigma\}$ for each $\sigma \in \Sigma$;

(2) $\mathcal{L}(E_1 \cup E_2) = \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$, $\mathcal{L}(E_1 \cdot E_2) = \mathcal{L}(E_1) \cdot \mathcal{L}(E_2)$, and $\mathcal{L}(E^*) = \mathcal{L}(E_1)^*$;

(3) $\mathcal{L}(E \curvearrowleft_{\square} E) = \mathcal{L}(E_1) \curvearrowleft_{\square} \mathcal{L}(E_2)$, $\mathcal{L}(E^{\circlearrowleft_{\square}}) = [\mathcal{L}(E_1)]^{\circlearrowleft_{\square}}$, and $\mathcal{L}(E^{\curvearrowleft \square}) = [\mathcal{L}(E)]^{\curvearrowleft \square}$

Since $\mathcal{L}^{\curvearrowleft \square} = \mathcal{L} \curvearrowleft_{\square} (\mathcal{L}^{\circlearrowleft_{\square}} \cup \{\square\})$, pure VRE and VRE capture the same class of languages.

Succinctness of VRE

Theorem

There are a pushdown alphabet $\widetilde{\Sigma}$ *and a family* $\{\mathcal{L}_n\}_{n\geq 1}$ *of regular languages over* $\widetilde{\Sigma}$ *such that for each* $n \geq 1$, $\mathcal{L}_n$ *can be denoted by a VRE of size* $O(n)$ *and every regular expression denoting* $\mathcal{L}_n$ *has size at least* $2^{\Omega(n)}$.

Succinctness of VRE

Theorem

There are a pushdown alphabet $\widetilde{\Sigma}$ and a family $\{\mathcal{L}_n\}_{n\geq 1}$ of regular languages over $\widetilde{\Sigma}$ such that for each $n \geq 1$, $\mathcal{L}_n$ can be denoted by a VRE of size $O(n)$ and every regular expression denoting $\mathcal{L}_n$ has size at least $2^{\Omega(n)}$.

Theorem

Let $\widetilde{\Sigma} = \langle \Sigma_{call}, \Sigma_{ret}, \{\Box\}\rangle$ with $\Sigma_{call} = \{c_1, c_2\}$ and $\Sigma_{ret} = \{r_1, r_2\}$. For $n \geq 1$, any NFA accepting $\mathcal{L}_n = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | i_1 \ldots i_n \in \{1, 2\}\}$ requires at least 2^n states.

Succinctness of VRE

Theorem

Let $\widetilde{\Sigma} = \langle \Sigma_{call}, \Sigma_{ret}, \{\Box\} \rangle$ with $\Sigma_{call} = \{c_1, c_2\}$ and $\Sigma_{ret} = \{r_1, r_2\}$. For $n \geq 1$, any NFA accepting $\mathcal{L}_n = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | i_1 \ldots i_n \in \{1, 2\}\}$ requires at least 2^n states.

Proof.

Let $\mathcal{L}(\mathcal{N}) = \mathcal{L}_n$ with $|Q| < 2^n$.

q_0 : the initial state.

$\exists c_{i_1} c_{i_2} \ldots c_{i_n}, c_{i'_1} c_{i'_2} \ldots c_{i'_n}$ s.t. $c_{i_1} c_{i_2} \ldots c_{i_n} \neq c_{i'_1} c_{i'_2} \ldots c_{i'_n}$, and

$$\delta(q_0, c_{i_1} c_{i_2} \ldots c_{i_n}) = \delta(q_0, c_{i'_1} c_{i'_2} \ldots c_{i'_n}) = q_1.$$

If $c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} \in \mathcal{L}_n$, then $\delta(q_1, r_{i_n} \ldots r_{i_2} r_{i_1}) = q_2$, where $q_2 \in F$.

Hence $c_{i'_1} c_{i'_2} \ldots c_{i'_n} r_{i_n} \ldots r_{i_2} r_{i_1} \in \mathcal{L}_n$ (Contradiction). □

Succinctness of VRE

Theorem

Let $\widetilde{\Sigma} = \langle \Sigma_{call}, \Sigma_{ret}, \{\Box\} \rangle$ with $\Sigma_{call} = \{c_1, c_2\}$ and $\Sigma_{ret} = \{r_1, r_2\}$. For $n \geq 1$, any NFA accepting $\mathcal{L}_n = \{c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} | i_1 \ldots i_n \in \{1, 2\}\}$ requires at least 2^n states.

Proof.

Let $\mathcal{L}(\mathcal{N}) = \mathcal{L}_n$ with $|Q| < 2^n$.

q_0 : the initial state.

$\exists c_{i_1} c_{i_2} \ldots c_{i_n}, c_{i'_1} c_{i'_2} \ldots c_{i'_n}$ s.t. $c_{i_1} c_{i_2} \ldots c_{i_n} \neq c_{i'_1} c_{i'_2} \ldots c_{i'_n}$, and
$\delta(q_0, c_{i_1} c_{i_2} \ldots c_{i_n}) = \delta(q_0, c_{i'_1} c_{i'_2} \ldots c_{i'_n}) = q_1$.

If $c_{i_1} c_{i_2} \ldots c_{i_n} r_{i_n} \ldots r_{i_2} r_{i_1} \in \mathcal{L}_n$, then $\delta(q_1, r_{i_n} \ldots r_{i_2} r_{i_1}) = q_2$, where $q_2 \in F$.

Hence $c_{i'_1} c_{i'_2} \ldots c_{i'_n} r_{i_n} \ldots r_{i_2} r_{i_1} \in \mathcal{L}_n$ (Contradiction). $\Box$

1, $\mathcal{L}_n$ can be expressed by the VRE of size $O(n)$ given by

$$\underbrace{E_{\curvearrowleft\Box} E_{\curvearrowleft\Box} \ldots_{\curvearrowleft\Box} E}_{n-1 \ \textit{times}} {}_{\curvearrowleft\Box} (c_1 \cdot r_1 \cup c_2 \cdot r_2), \text{ where } E = (c_1 \cdot \Box \cdot r_1 \cup c_2 \cdot \Box \cdot r_2).$$

2, Regular expressions can be converted in linear time into equivalent NFA.

Properties of NVPA

Theorem (R. Alur and P. Madhusudan. Visibly Pushdown Languages. STOC 2004)

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F\rangle$ and $\mathcal{P}' = \langle Q', q'_{in}, \Gamma', \Delta', F'\rangle$ be two NVPA over $\widetilde{\Sigma}$. Then, one can construct in linear time:

- *1. an NVPA accepting $\mathcal{L}(\mathcal{P}) \cup \mathcal{L}(\mathcal{P}')$ (resp. $\mathcal{L}(\mathcal{P}) \cdot \mathcal{L}(\mathcal{P}')$) with $|Q| + |Q'|$ states and $|\Gamma| + |\Gamma'|$ stack symbols.*
- *2. an NVPA accepting $\mathcal{L}(\mathcal{P})^*$ with $2|Q|$ states and $2|\Gamma|$ stack symbols.*

VRE to NVPA

Corollary

Given a VRE E, one can construct in single exponential time an NVPA accepting $\mathcal{L}(E)$.

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.

A run π is a summary of $\mathcal{P}$ from p to p': $\exists w \in WMW(\widetilde{\Sigma})$ s.t. $(p, \beta) \xrightarrow{w} (p', \beta)$.

A run uses only sub-summaries from $\mathcal{S}$: $\forall q, q' \in Q$, if $\exists w \in WMW(\widetilde{\Sigma})$ s.t. $(p, \beta) \xrightarrow{w} (p', \beta)$, then $(p, p') \in \mathcal{S}$. □

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$.
$\Lambda = \{\Box_{pp'} | p, p' \in Q\}$.
$\mathcal{P}' = \langle Q, q_{in}, \Gamma, \Delta \cup \{(p, \Box_{pp'}, p') | \Box_{pp'} \in \Lambda\}, F \rangle$ over $\widetilde{\Sigma}_{\Lambda}$.

□

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$.

$\Lambda = \{\Box_{pp'} | p, p' \in Q\}$.

$\mathcal{P}' = \langle Q, q_{in}, \Gamma, \Delta \cup \{(p, \Box_{pp'}, p') | \Box_{pp'} \in \Lambda\}, F \rangle$ over $\widetilde{\Sigma}_\Lambda$.

Given $q, q' \in Q$, $\mathcal{S} \subseteq Q \times Q$, $\Lambda' \subseteq \{\Box_{pp'} | p, p' \in Q\}$, we define:

$R(p, p', \mathcal{S}, \Lambda') : \{w | (p, \bot) \xrightarrow{w} (p', \beta)$ use only sub-summaries from $\mathcal{S}\}$.

$$\mathcal{L}(\mathcal{P}) = \bigcup_{q = q_{in}, q' \in F} R(q, q', Q \times Q, \emptyset).$$

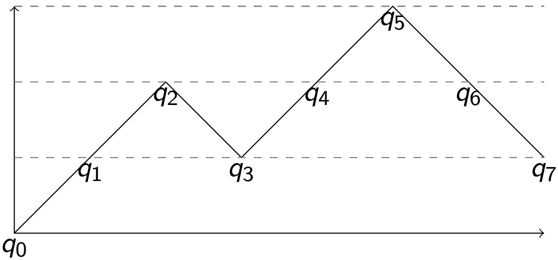
$WM(R(q, q', S, \Lambda)) = R(q, q', S, \Lambda) \cap WM(\widetilde{\Sigma}^*)$. □

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.



$w \in R(q_0, q_7, \{(q_1, q_3), (q_3, q_7), (q_1, q_7), (q_4, q_6)\}, \emptyset)$

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.

Basic case: $\mathcal{S} = \emptyset$.

- $R(q, q', \mathcal{S}, \Lambda') = (\Sigma_{call} \cup \Sigma_{int} \cup \Lambda')^* \cup (\Sigma_{ret} \cup \Sigma_{int} \cup \Lambda')^*$
- $WM(R(q, q', \mathcal{S}, \Lambda')) = (\Sigma_{int} \cup \Lambda')^*$

□

NVPA to VRE

Theorem

Given an NVPA $\mathcal{P}$, one can construct in single exponential time a VRE E such that $\mathcal{L}(E) = \mathcal{L}(\mathcal{P})$.

Proof.

Induction step: $\mathcal{S} = \mathcal{S}' \cup \{(p, p')\}$ with $(p, p') \notin \mathcal{S}'$.

$P_{p \to p'} = \{(s, c, r, s') \in Q \times \Sigma_{call} \times \Sigma_{ret} \times Q | \exists \gamma \in \Gamma.(p, c, s, \gamma), (s', r, \gamma, p') \in \Delta\}$.

$\mathcal{S}(p, p', \mathcal{S}' \cup \{(p, p')\}, \Lambda') :=$

$$([\bigcup_{(s,c,r,s') \in P_{p \to p'}} \{c\} \cdot WM(R(s, s', \mathcal{S}', \Lambda' \cup \{\Box_{pp'}\})) \cdot \{r\}]^{\curvearrowleft \Box_{pp'}}) \curvearrowleft \Box_{pp'}$$

$$[\bigcup_{(s,c,r,s') \in P_{p \to p'}} \{c\} \cdot WM(R(s, s', \mathcal{S}', \Lambda'\})) \cdot \{r\}].$$

$WM(R(p, p', \mathcal{S}' \cup \{(p, p')\}, \Lambda')) := WM(R(p, p', \mathcal{S}', \Lambda')) \cup$
$\quad WM(R(s, s', \mathcal{S}', \Lambda' \cup \{\Box_{pp'}\})) \curvearrowleft \Box_{pp'} \mathcal{S}(p, p', \mathcal{S}' \cup \{(p, p')\}, \Lambda')$.

$R(p, p', \mathcal{S}' \cup \{(p, p')\}, \Lambda')) := R(p, p', \mathcal{S}', \Lambda') \cup$
$\quad R(s, s', \mathcal{S}', \Lambda' \cup \{\Box_{pp'}\}) \curvearrowleft \Box_{pp'} \mathcal{S}(p, p', \mathcal{S}' \cup \{(p, p')\}, \Lambda')$.

□

VRE and VPL

Corollary

(Pure) Visibly Rational Expressions capture the class of VPL.

Strong NVPA

Definition

A strong NVPA over $\widetilde{\Sigma}$ is an NVPA $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ over $\widetilde{\Sigma}$ such that $\widehat{\perp} \in \Gamma$ and the following holds:

- Initial State Requirement: $q_{in} \notin F$ and there are no transitions leading to q_{in}.
- Final State requirement: there are no transitions from accepting states.

Strong NVPA

Definition

A strong NVPA over $\widetilde{\Sigma}$ is an NVPA $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ over $\widetilde{\Sigma}$ such that $\widehat{\bot} \in \Gamma$ and the following holds:

- Initial State Requirement: $q_{in} \notin F$ and there are no transitions leading to q_{in}.
- Final State requirement: there are no transitions from accepting states.
- Push Requirement: every push transition from the initial state q_{in} pushes onto the stack the special symbol $\widehat{\bot}$.
- Pop Requirement: for all $q, p \in Q$ and $r \in \Sigma_{ret}$, $(q, r, \bot, p) \in \Delta$ iff $(q, r, \widehat{\bot}, p) \in \Delta$.

Strong NVPA

Definition

A strong NVPA over $\widetilde{\Sigma}$ is an NVPA $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ over $\widetilde{\Sigma}$ such that $\widehat{\perp} \in \Gamma$ and the following holds:

- Initial State Requirement: $q_{in} \notin F$ and there are no transitions leading to q_{in}.
- Final State requirement: there are no transitions from accepting states.
- Push Requirement: every push transition from the initial state q_{in} pushes onto the stack the special symbol $\widehat{\perp}$.
- Pop Requirement: for all $q, p \in Q$ and $r \in \Sigma_{ret}$, $(q, r, \perp, p) \in \Delta$ iff $(q, r, \widehat{\perp}, p) \in \Delta$.
- Will-formed (semantic) Requirement: for all $w \in \mathcal{L}(\mathcal{P})$, every initialized accepting run of $\mathcal{P}$ over w leads to a configuration whose stack content is in $\{\widehat{\perp}\}^* \perp$.

Strong NVPA

Definition

A strong NVPA over $\widetilde{\Sigma}$ is an NVPA $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ over $\widetilde{\Sigma}$ such that $\widehat{\perp} \in \Gamma$ and the following holds:

- Initial State Requirement: $q_{in} \notin F$ and there are no transitions leading to q_{in}.
- Final State requirement: there are no transitions from accepting states.
- Push Requirement: every push transition from the initial state q_{in} pushes onto the stack the special symbol $\widehat{\perp}$.
- Pop Requirement: for all $q, p \in Q$ and $r \in \Sigma_{ret}$, $(q, r, \perp, p) \in \Delta$ iff $(q, r, \widehat{\perp}, p) \in \Delta$.
- Will-formed (semantic) Requirement: for all $w \in \mathcal{L}(\mathcal{P})$, every initialized accepting run of $\mathcal{P}$ over w leads to a configuration whose stack content is in $\{\widehat{\perp}\}^* \perp$.

Note that the initial state requirement implies that $\varepsilon \notin \mathcal{L}(\mathcal{P})$.

Closure property

Theorem

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ and $\mathcal{P}' = \langle Q', q'_{in}, \Gamma', \Delta', F' \rangle$ be two strong NVPA over $\widetilde{\Sigma}$. Then, one can construct in linear time:

- *1. a strong NVPA accepting $(L(\mathcal{P})) \cup L(\mathcal{P}'))$ with $|Q| + |Q'| + 1$ states and $|\Gamma| + |\Gamma'| - 1$ stack symbols, and*
- *2. a strong NVPA accepting $[L(\mathcal{P})]^* \setminus \{\varepsilon\}$ with $|Q| + 1$ states and $|\Gamma|$ stack symbols.*

Closure property

Proof.

1. The NVPA accepting $(L(\mathcal{P})) \cup L(\mathcal{P}'))$
$\mathcal{P}'' = \langle Q \cup Q' \cup \{q''_{in}\}, q''_{in}, \Gamma \cup \Gamma' \cup \{\widehat{\perp}\}, \Delta'', F \cup F' \rangle$ can be constructed as follows:

$$\begin{aligned}
\Delta'' = {} & \Delta \cup \Delta' \cup \\
& \bigcup(\{(q''_{in}, \sigma, q, \widehat{\perp}) | (q_{in}, \sigma, q, \widehat{\perp}) \in \Delta, \text{and } \sigma \in \Sigma_{call}\} \\
& \qquad \cup \{(q''_{in}, \sigma, \gamma, q) | (q_{in}, \sigma, \gamma, q) \in \Delta, \text{and } \sigma \in \Sigma_{ret}\} \\
& \qquad \cup \{(q''_{in}, \sigma, q) | (q_{in}, \sigma, q) \in \Delta, \text{and } \sigma \in \Sigma_{int}\}) \\
& \bigcup(\{(q''_{in}, \sigma, q, \widehat{\perp}) | (q'_{in}, \sigma, q, \widehat{\perp}) \in \Delta', \text{and } \sigma \in \Sigma'_{call}\} \\
& \qquad \cup \{(q''_{in}, \sigma, \gamma, q) | (q'_{in}, \sigma, \gamma, q) \in \Delta', \text{and } \sigma \in \Sigma'_{ret}\} \\
& \qquad \cup \{(q''_{in}, \sigma, q) | (q'_{in}, \sigma, q) \in \Delta', \text{and } \sigma \in \Sigma'_{int}\})
\end{aligned}$$

□

Closure property

Proof.

2. The NVPA accepting $[L(\mathcal{P})]^* \setminus \{\varepsilon\}$ $\mathcal{P}'' = \langle Q \cup \{q'_{in}\}, q'_{in}, \Gamma, \Delta'', F\rangle$ can be constructed in two step:

$\Delta \to \Delta_0 : \Delta_0 = \Delta \cup$

$$\bigcup(\{(q_1, \sigma, q_{in}, \gamma) | (q_1, \sigma, q_2, \gamma) \in \Delta, q_2 \in F, \text{and } \sigma \in \Sigma_{call}\}$$
$$\cup\{(q_1, \sigma, \gamma, q_{in}) | (q_1, \sigma, \gamma, q_2) \in \Delta, q_2 \in F, \text{and } \sigma \in \Sigma_{ret}\}$$
$$\cup\{(q_1, \sigma, q_{in}) | (q_1, \sigma, q_2) \in \Delta, q_2 \in F, \text{and } \sigma \in \Sigma_{int}\})$$

$\Delta_0 \to \Delta' : \Delta' = \Delta \cup$

$$\bigcup(\{(q'_{in}, \sigma, q, \widehat{\perp}) | (q_{in}, \sigma, q, \widehat{\perp}) \in \Delta, \text{and } \sigma \in \Sigma_{call}\}$$
$$\cup\{(q'_{in}, \sigma, \gamma, q) | (q_{in}, \sigma, \gamma, q) \in \Delta, \text{and } \sigma \in \Sigma_{ret}\}$$
$$\cup\{(q'_{in}, \sigma, q) | (q_{in}, \sigma, q) \in \Delta, \text{and } \sigma \in \Sigma_{int}\})$$

□

Closure property

Theorem

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ be a strong NVPA over $\widetilde{\Sigma}$. Then, one can construct in linear time a strong NVPA accepting $MWM(L(\mathcal{P}))$ with $|Q|$ states and $|\Gamma| + 1$ stack symbols.

Closure property

Theorem

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F\rangle$ be a strong NVPA over $\widetilde{\Sigma}$. Then, one can construct in linear time a strong NVPA accepting $MWM(L(\mathcal{P}))$ with $|Q|$ states and $|\Gamma| + 1$ stack symbols.

Proof.

The required NVPA $\mathcal{P}' = \langle Q, q_{in}, \Gamma \cup \{\widehat{\bot_1}\}, \Delta', F'\rangle$ is defined as follows:

$$\begin{aligned}\Delta' = (\Delta \setminus (&\{(q_{in}, \sigma, q, \widehat{\bot}) | (q_{in}, \sigma, q, \widehat{\bot}) \in \Delta, \text{and } \sigma \in \Sigma_{call}\}\\ &\cup\{(q_{in}, \sigma, \gamma, q) | (q_{in}, \sigma, \gamma, q) \in \Delta, \text{and } \sigma \in \Sigma_{ret}\}\\ &\cup\{(q_{in}, \sigma, q) | (q_{in}, \sigma, q) \in \Delta, \text{and } \sigma \in \Sigma_{int}\}\\ &\cup\{(q, \sigma, q_1, \gamma) | (q, \sigma, q_1, \gamma) \in \Delta,\ q_1 \in F, \text{and } \sigma \in \Sigma_{call}\}\\ &\cup\{(q, \sigma, \gamma, q_1) | (q, \sigma, \gamma, q_1) \in \Delta,\ q_1 \in F, \text{and } \sigma \in \Sigma_{ret}\}\\ &\cup\{(q, \sigma, q_1) | (q, \sigma, q_1) \in \Delta,\ q_1 \in F, \text{and } \sigma \in \Sigma_{int}\}))\end{aligned}$$

$$\bigcup(\{(q_{in}, \sigma, q, \widehat{\bot_1}) | (q_{in}, \sigma, q, \widehat{\bot}) \in \Delta, \text{and } \sigma \in \Sigma_{call}\}$$

$$\bigcup(\{(q, \sigma, \widehat{\bot_1}, q_1) | (q, \sigma, \widehat{\bot}, q_1) \in \Delta,\ q_1 \in F, \text{and } \sigma \in \Sigma_{ret}\}$$

□

Closure property

Theorem

Let $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ and $\mathcal{P}' = \langle Q', q'_{in}, \Gamma', \Delta', F' \rangle$ be two strong NVPA over $\widetilde{\Sigma}$, and $\square \in \Sigma_{int}$. Then, one can construct in linear time:

- *(1) a strong NVPA accepting $(L(\mathcal{P}))^{\circlearrowleft_\square}$ with $|Q|$ states and $|Q| + |\Gamma| + 1$ stack symbols, and*
- *(2) a strong NVPA accepting $[L(\mathcal{P})]_{\curvearrowleft_\square} \mathcal{L}(\mathcal{P}')$ with $|Q| + |Q'|$ states and $|\Gamma| + |\Gamma'| + |Q|$ stack symbols.*

Closure property

Proof.

(1) Assume $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ s.t.

1, $\mathcal{L}(\mathcal{P}) \subseteq MWM(\widetilde{\Sigma})$;

2, $Q \cap \Gamma = \emptyset$;

3, All the transitions from the initial state are push transitions.

$\square$

Closure property

Proof.

(1) Assume $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ s.t.

1, $\mathcal{L}(\mathcal{P}) \subseteq MWM(\widetilde{\Sigma})$;

2, $Q \cap \Gamma = \emptyset$;

3, All the transitions from the initial state are push transitions.

The NVPA $\mathcal{P}'$ accepting $(L(\mathcal{P}))^{\circlearrowleft_\Box}$ can be constructed by adding to Δ the following transitions:

1 (q, c, q', p), where $(q, \Box, p) \in \Delta$, $q \neq q_{in}$ and $(q_{in}, c, q', \widehat{\bot}) \in \Delta$.

□

Closure property

Proof.

(1) Assume $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$ s.t.

1, $\mathcal{L}(\mathcal{P}) \subseteq MWM(\widetilde{\Sigma})$;

2, $Q \cap \Gamma = \emptyset$;

3, All the transitions from the initial state are push transitions.

The NVPA $\mathcal{P}'$ accepting $(L(\mathcal{P}))^{\circlearrowleft \square}$ can be constructed by adding to Δ the following transitions:

1 (q, c, q', p), where $(q, \square, p) \in \Delta$, $q \neq q_{in}$ and $(q_{in}, c, q', \widehat{\perp}) \in \Delta$.

2 (q, r, p, p), where $(q, r, \widehat{\perp}, q_1) \in \Delta$, $q_1 \in F$, $p \in Q \setminus \{q_{in}\}$.

□

Closure property

Proof.

(2) Assume $\mathcal{P}' = \langle Q', q'_{in}, \Gamma', \Delta', F' \rangle$ s.t.
1, $\mathcal{L}(\mathcal{P}') \subseteq MWM(\widetilde{\Sigma})$;
2, $Q' \cap \Gamma' = \emptyset$;
3, All the transitions from the initial state are push transitions.
The NVPA $\mathcal{P}_1$ accepting $[L(\mathcal{P})]_{\curvearrowleft \Box} \mathcal{L}(\mathcal{P}')$ can be constructed as follows:
1 (q, c, q', p), where $(q, \Box, p) \in \Delta$, $q \neq q_{in}$ and $(q_{in}, c, q', \widehat{\bot}) \in \Delta'$.
2 (q, r, p, p), where $(q, r, \widehat{\bot}, q_1) \in \Delta'$, $q'_1 \in F$, $p \in Q \setminus \{q_{in}\}$.

□

Pure VRE to NVPA

Theorem

Let E be a pure VRE. Then, one can construct in quadratic time an NVPA $\mathcal{P}$ accepting $\mathcal{L}(E)$ with at most $|E| + 1$ states and $|E|^2$ stack symbols.

Pure VRE to NVPA

Theorem

Let E be a pure VRE. Then, one can construct in quadratic time an NVPA $\mathcal{P}$ accepting $\mathcal{L}(E)$ with at most $|E|+1$ states and $|E|^2$ stack symbols.

Proof.

Induction on E.
Basic case: For example, Let $E = c$ $(c \in \Sigma_{call})$.
Then $\mathcal{P} = \langle \{q_{in}, q_f\}, q_{in}, \{\widehat{\bot}\}, \{(q_{in}, c, q_f, \widehat{\bot})\}, \{q_f\} \rangle$. □

Pure VRE to NVPA

Theorem

Let E be a pure VRE. Then, one can construct in quadratic time an NVPA $\mathcal{P}$ accepting $\mathcal{L}(E)$ with at most $|E|+1$ states and $|E|^2$ stack symbols.

Proof.

Induction step: Comes from above theorems.
Take $E_1 \curvearrowleft_{\Box} E_2$ for an example.
$\mathcal{P}_1 = \langle Q_1, q_{in}^1, \Gamma_1, \Delta_1, F_1 \rangle$ and $\mathcal{P}_2 = \langle Q_2, q_{in}^2, \Gamma_2, \Delta_2, F_2 \rangle$ s.t.
$\mathcal{L}(\mathcal{P}_1) = \mathcal{L}(E_1) \setminus \{\varepsilon\}$ and $\mathcal{L}(\mathcal{P}_2) = \mathcal{L}(E_2) \setminus \{\varepsilon\}$.
From the inductive hypothesis

$$|Q_1| \leq |E_1| + 1,\ |Q_2| \leq |E_2| + 1,\ |\Gamma_1| \leq |E_1|^2,\ \textit{and}\ |\Gamma_2| \leq |E_2|^2.$$

Then we can construct in linear time $\mathcal{P} = \langle Q, q_{in}, \Gamma, \Delta, F \rangle$, accepting $E_1 \curvearrowleft_{\Box} E_2$, s.t.
$|Q| = |Q_1| + |Q_2| \leq |E_1| + |E_2| + 2 = |E| + 1$.
$|\Gamma| \leq |E_1|^2 + |E_2|^2 + |E_1| + 1 \leq (|E_1| + |E_2| + 1)^2 = |E|^2$.

Decision Problems

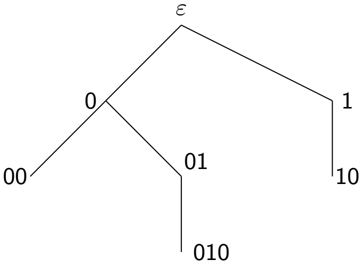
Theorem

The universality, inclusion, and language equivalence problems for pure VRE are EXPTIME-complete.

Universality

Proof.

Upper bounds: Follows from the EXPTIME-completeness of the universality for NVPA.
Lower bounds: Reduction from the word problem for polynomial space bounded alternating Turing Machines (TM) $\mathcal{A}$ with a binary branching degree.



The encoding this running tree is:

$$(fC_\varepsilon)(fC_0)(fC_{00})(b\overline{C_{00}})^r(fC_{01})(fC_{010})(b\overline{C_{010}})^r(b\overline{C_{01}})^r(b\overline{C_0})^r(fC_1)(fC_{10})(b\overline{C_{10}})^r(b\overline{C_1})^r(b\overline{C_\varepsilon})^r$$

Universality

Proof.

Lower bounds: Reduction from the word problem for polynomial space bounded alternating Turing Machines (TM) $\mathcal{A}$ with a binary branching degree.

A word $w \in (\Gamma \cup \{f, b\})^*$ is a unsuccessful computation of M if one of the following conditions holds,

(1) w is not minimal well-matched.

(2) Subword of w like $fC_x f$, $fC_x(C_x)^r b$ such that C_x is not a configuration.

(3) $C_\varepsilon \neq q_0 w B^{(c-1)n}$.

(4) minimal well-matched subword of w like $fC_2(\Gamma \cup \{f, b\})^* \overline{C_1} b$ such that $C_1 \neq C_2$.

(5) w is not accepted by $\mathcal{A}$.

(6) there is a subword $fC_x fC_{x0}$ or $\overline{C_{x0}^r} b \overline{x_x^r} b$ or $\overline{c_{x1}^r} b \overline{x_x^r} b$, such that $C_x \nvdash C_{x0}$, or $C_x \nvdash C_{x1}$: Guess and index $i : 1 < i < cn + 1$, and check the relationship of the $(i-1, i, i+1)$-th symbol of C_x and the i-th symbol of C_{x0},. . . □

Universality

Proof.

Let $\widetilde{\Sigma} = \{\{f\}, \{b\}, \Gamma \cup \overline{\Gamma} \cup \Box\}$

(1) w is not minimal well-matched.

$$\begin{aligned} r_1 = & (\{b\} \cup \Gamma \cup \overline{\Gamma}) \cdot (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \\ & \cup (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot (\{f\} \cup \Gamma \cup \overline{\Gamma}) \\ & \cup f \cdot ((\Gamma \cup \overline{\Gamma} \cup \{\Box, b\})^* b) \curvearrowleft_{\Box} ((\Gamma \cup \overline{\Gamma} \cup \{f, b\})^*) \cdot (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot b \\ & \cup f \cdot ((\Gamma \cup \overline{\Gamma} \cup \{\Box, f\})^* f) \curvearrowleft_{\Box} ((\Gamma \cup \overline{\Gamma} \cup \{f, b\})^*) \\ & \cdot (\Gamma \cup \overline{\Gamma} \cup \{\Box\})^*) \curvearrowleft_{\Box} (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot b \end{aligned}$$

(2) Subword of w like fC_x such that C_x is not a configuration.

$$\begin{aligned} r_2 = & (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot \{(f\Sigma^* f) \cup (f\Sigma^* Q\Sigma^* Q(Q \cup \Gamma)^* f) \cup \\ & (f\Sigma^* b) \cup (f\Sigma^* Q\Sigma^* b) \cup (f\Sigma^* Q\Sigma^* Q\Sigma^* Q(Q \cup \Gamma)^* b)\} \cdot (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^*. \end{aligned}$$

□

Universality

Proof.

Let $\widetilde{\Sigma} = \{\{f\}, \{b\}, \Gamma \cup \overline{\Gamma} \cup \Box\}$

(3) $C_\varepsilon \neq q_0 w B^{(c-1)n}$.

$$r_3 = f(\Gamma \setminus \{q_0\})(\Gamma \cup \overline{\Gamma} \cup \{f,b\})^* \cup f \bigcup_{i=1}^{n} \Gamma^i(\Gamma \setminus \{a_i\})(\Gamma \cup \overline{\Gamma} \cup \{f,b\})^* \cup \bigcup_{i=n+1}^{cn} f\Gamma^i(\Gamma \setminus \{B\})(\Gamma \cup \overline{\Gamma} \cup \{f,b\})^*.$$

(4) minimal well-matched subword of w like $fC_1(\Gamma \cup \{f,b\})^*\overline{C_2}b$ such that $C_1 \neq C_2$.

$$r_4 = (\Gamma \cup \overline{\Gamma} \cup \{f,b\})^* \cdot (f \cdot (\bigcup_{i=0}^{cn-1} \Gamma^i \cdot [\bigcup_{\gamma_1,\gamma_2 \in \Gamma, \gamma_1 \neq \gamma_2} \gamma_1 \cdot (\Gamma \cup \overline{\Gamma} \cup \{\Box\})^* \cdot \overline{\gamma_2}] \cdot \overline{\Gamma}^i) \cdot b) \curvearrowleft_\Box (\Gamma \cup \overline{\Gamma} \cup \{f,b\})^* \cdot (\Gamma \cup \overline{\Gamma} \cup \{f,b\})^*.$$

Universality

Proof.

(5) w is not accepted by $\mathcal{A}$.

$r_5 = (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot f\Gamma^*(Q \setminus F)(\Gamma \cup \overline{\Gamma})^*\overline{(Q \setminus F)}\overline{\Gamma}^* b \cdot (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^*$.

(6) there is a subword $fC_x fC_{x0}$ or $\overline{C^r_{x0}}b\overline{x^r_x}b$ or $\overline{c^r_{x1}}b\overline{x^r_x}b$, such that $C_x \nvdash C_{x0}$, or $C_x \nvdash C_{x1}$: Guess and index $i : 1 < i < cn + 1$, and check the relationship of the $(i-1, i, i+1)$-th symbol of C_x and the i-th symbol of C_{x0},. . .

$$r_6 = (\Gamma \cup \overline{\Gamma} \cup \{f, b\})^* \cdot$$
$$\left(\bigcup_{i=0}^{cn-2} \bigcup_{(\sigma_1,\sigma_2,\sigma_3,\sigma'_1,\sigma'_2,\sigma'_3)\notin f_M} (f\Gamma^i\sigma_1\sigma_2\sigma_3\Gamma^{cn-i+3}f\Gamma^i\sigma'_1\sigma'_2\sigma'_3\Gamma^{cn-i+3})\right.$$
$$\left.\bigcup_{i=0}^{cn-2} \bigcup_{(\sigma_1,\sigma_2,\sigma_3,\sigma'_1,\sigma'_2,\sigma'_3)\notin f_M} (\overline{\Gamma}^i\overline{\sigma'_3\sigma'_2\sigma'_1}\overline{\Gamma}^{cn-i+3}b\overline{\Gamma}^i\overline{\sigma_3\sigma_2\sigma_1}\overline{\Gamma}^{cn-i+3}b)\right)$$
$$\cdot(\Gamma \cup \overline{\Gamma} \cup \{f, b\})^*.$$

Universality

Proof.

$\mathcal{L}(r_1 \cup r_2 \cup r_3 \cup r_4 \cup r_5 \cup r_6) = \widetilde{\Sigma}^*$ iff M does not accept w. □

ω-VRE

Definition

The syntax of ω-VRE I over $\widetilde{\Sigma}$ is inductively defined as follows:

$$I := (E)^{\omega} \mid (I \cup I) \mid (E \cdot I)$$

where E is a VRE over $\widetilde{\Sigma}$.

ω-VRE

Definition

The syntax of ω-VRE I over $\widetilde{\Sigma}$ is inductively defined as follows:

$$I := (E)^{\omega} \mid (I \cup I) \mid (E \cdot I)$$

where E is a VRE over $\widetilde{\Sigma}$.

An ω-VRE I is pure if every VRE subexpression is pure.

The language of ω-VRE

Definition

The language of an ω-VRE I is defined as:
(1) $\mathcal{L}(E^\omega) = [\mathcal{L}(E)]^\omega$;
(2) $\mathcal{L}(I_1 \cup I_2) = \mathcal{L}(I_1) \cup \mathcal{L}(I_2)$;
(3) $\mathcal{L}(E \cdot I) = \mathcal{L}(E) \cdot \mathcal{L}(I)$;

The regular property

Theorem

Let $\mathcal{L}$ be a ω-VPL with respect to $\widetilde{\Sigma}$. Then, there are $n \geq 1$ and VPL $\mathcal{L}_1, \mathcal{L}'_1, \ldots, \mathcal{L}_n, \mathcal{L}_n$ with respect to $\widetilde{\Sigma}$ such that $\mathcal{L} = \cup_{i=1}^{i=n} \mathcal{L}_i \cdot (\mathcal{L}'_i)^\omega$. Moreover, the characterization is constructive.

Proof.

The proof is the same as the one for the ω-regular languages.
Suppose $\mathcal{L}$ can be defined by a ω-VPA $M = (Q, Q_{in}, \Gamma, \delta, \mathcal{F})$. Let $L_{qq'} = \{w \in \widetilde{\Sigma}^* | q \xrightarrow{w} q'\}$. Then $\mathcal{L} = \bigcup_{q_0 \in Q_{in}, q_f \in F} L_{q_0 q_f}(L_{q_f q_f} \setminus \{\varepsilon\})^\omega$. □

ω-VRE and ω-VPL

Theorem

(Pure) ω-VRE capture the class of ω-VPL. Moreover, pure ω-VRE can be converted in quadratic time into equivalent Büchi ω-NVPA.

Proof.

ω-VRE $\rightarrow$ VRE $\rightarrow$ NVPA $\rightarrow$ ω-VPA.

ω-VPA $\rightarrow$ VPA $\rightarrow$ VRE $\rightarrow$ ω-VRE. □

Questions?