Visibly Pushdown Automata

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

09 November 2021

Outline

1. Visibly Pushdown Automata
2. Closure properties of VPL
3. Determinization
4. Logical Characterization

Visibly Pushdown Automata

- A sub-class of Pushdown Automata (PDA's) in which pushing/popping from the stack is dictated by input letters.
- Useful properties for verification
  - Closed under operations like union, intersection, complementation, concatentation, Kleene-*.
  - Decidable language inclusion and universality problems.

Proposed by Rajeev Alur and P. Madhusudan in STOC 2004.

Example VPA

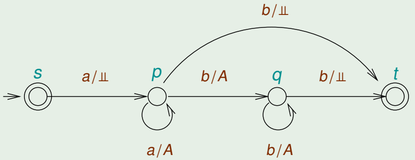
Example VPA for $\{a^n b^n \mid n \geq 0\}$

$$\Sigma_c = \{a\}$$
$$\Sigma_r = \{b\}$$
$$\Sigma_{int} = \emptyset$$

$$(s, a, p, \perp\!\!\!\perp)$$
$$(p, a, p, A)$$
$$(p, b, A, q)$$
$$(q, b, A, q)$$
$$(q, b, \perp\!\!\!\perp, t)$$

$$F = \{s, t\}.$$

State Diagram of VPA



Definitions

A VPA over a partitioned alphabet $\widetilde{\Sigma} = (\Sigma_c, \Sigma_r, \Sigma_{int})$ is a structure $M = (Q, Q_0, \Gamma, \perp, \delta, F)$ where Q is a finite set of states, Q_0 is a set of initial states, Γ is a stack alphabet with $\perp \in \Gamma$, F is a set of final states, and δ is the transition relation of the form:

- (p, a, q, A) if $a \in \Sigma_c$ (push transition)
- (p, a, A, q) if $a \in \Sigma_r$ (pop transition)
- (p, a, q) if $a \in \Sigma_{int}$ (internal transition).

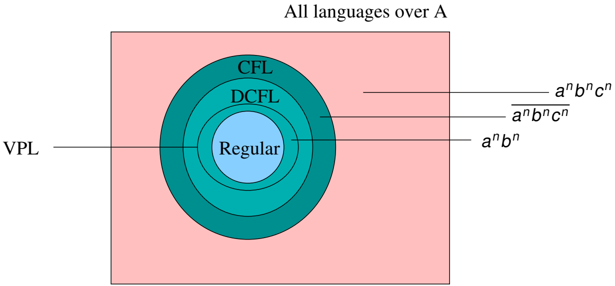
Restrictions:

- $\perp$ is never pushed on the stack
- Pop transitions can read $\perp$ from the stack but must leave it in place.
- No epsilon transitions

Run of M on a word $w = a_1 a_2 \ldots a_n$.

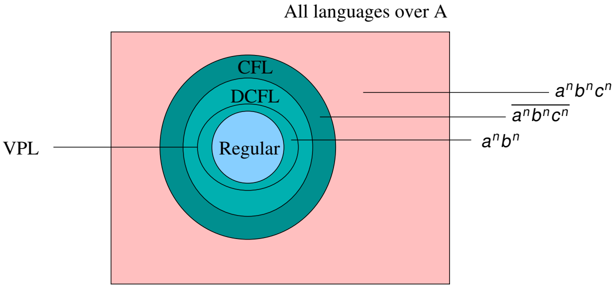
Class of languages accepted by VPA's are called Visibly Pushdown Languages (VPL).

Closure Properties of VPL



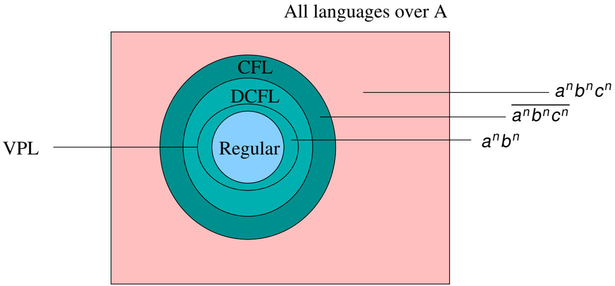
	Closed?
Union	

Closure Properties of VPL



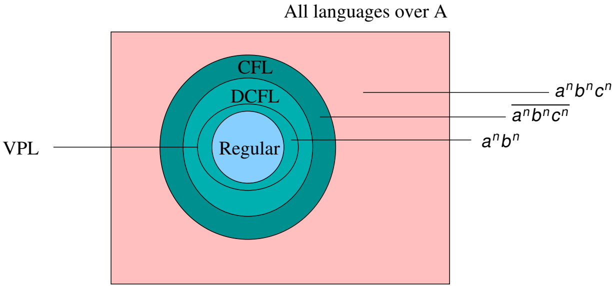
	Closed?
Union	√
Intersection	

Closure Properties of VPL



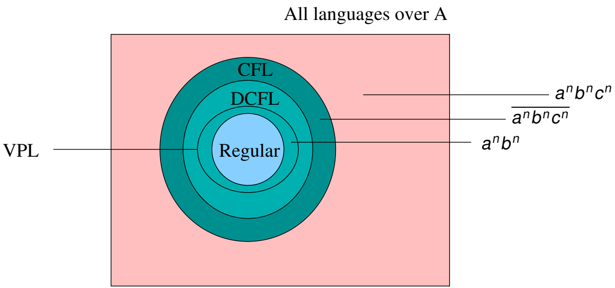
	Closed?
Union	√
Intersection	√
Concatentation	

Closure Properties of VPL



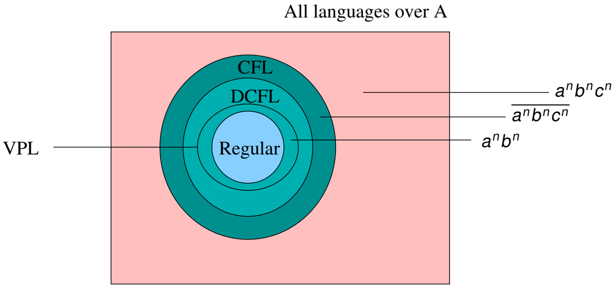
	Closed?
Union	√
Intersection	√
Concatentation	√
Kleene-*	

Closure Properties of VPL



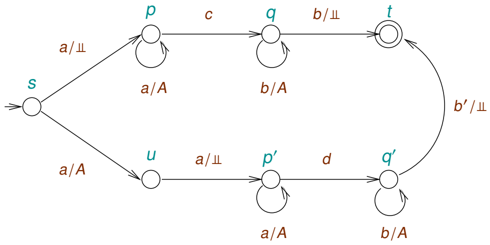
	Closed?
Union	√
Intersection	√
Concatenation	√
Kleene-*	√
Complementation	

Closure Properties of VPL



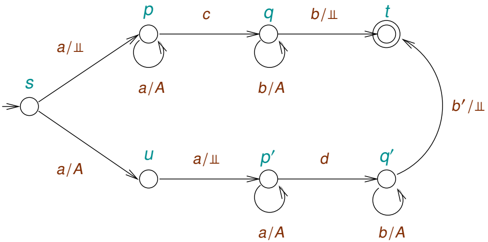
	Closed?
Union	√
Intersection	√
Concatenation	√
Kleene-*	√
Complementation	√

Example non-deterministic VPA



Partitioned alphabet is $(\{a\}, \{b, b'\}, \{c, d\})$.

Example non-deterministic VPA

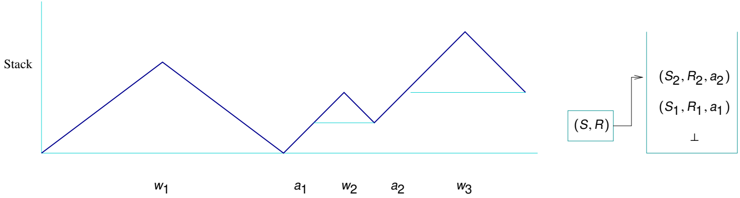


Partitioned alphabet is $(\{a\}, \{b, b'\}, \{c, d\})$.
Accepts language $\{a^n c b^n \mid n \geq 1\} \cup \{a^n d b^{n-2} b' \mid n \geq 1\}$.

Determinizing VPA's

Let $M = (Q, Q_0, \Gamma, \delta, F)$ be a VPA over $\widetilde{\Sigma}$. We define a new VPA M' as follows:

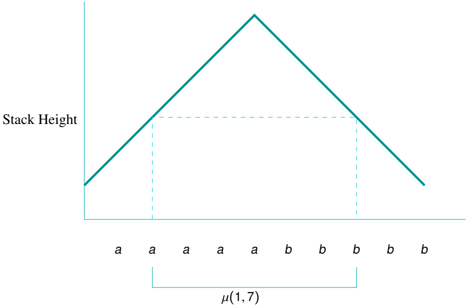
- Control state is of the form (S, R) where $S \subseteq Q \times Q$ and $R \subseteq Q$.
- Stack symbols will be of the form (S, R, a) where S and R are as above, and $a \in \Sigma_c$ is a call alphabet.
- Construction maintains the following invariant:
  - S_1 is the summary of w_1, S_2 of w_2, and S of w_3.
  - R_1 is the reach set after w_1, R_2 after $w_1 a_1 w_2$, and R after w.



Decision Procedures for VPLs

- Emptiness
- Language inclusion / equivalance
- Universality

MSO over $\widetilde{\Sigma}$ with matching predicate



- Interpreted over finite words w over Σ.
- Syntax:

$$Q_a(x) \mid x < y \mid \mu(x,y) \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x \varphi \mid \exists X \varphi.$$

- $\mu(x,y)$ is true if $w(x)$ is a call and matching return is at $w(y)$.
- Example over $(\{a\},\{b\},\{d\})$:
  $\forall x(Q_a(x) \implies \exists y(\mu(x,y) \wedge Q_b(y)))$.

Logical characterization of VPLs

Theorem

L is a VPL over $\widetilde{\Sigma}$ iff L is definable in MSO($\widetilde{\Sigma}$).

Proof is similar to that of regular langauages.