

Deterministic PDA's

Deepak D'Souza

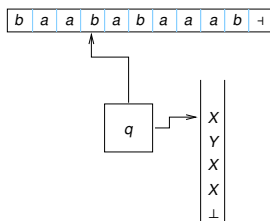
Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

25 October 2013

Outline

- 1 Deterministic PDA's
- 2 Closure properties of DCFL's
- 3 Complementing DPDA's

Deterministic PDA's



A PDA with restrictions that:

- **At most** one move possible in any configuration.
 - For any state p , $a \in A$, and $X \in \Gamma$: at most one move of the form $(p, a, X) \rightarrow (q, \gamma)$ or $(p, \epsilon, X) \rightarrow (q, \gamma)$.
 - Effectively, a DPDA must see the current state, and top of stack, and decide whether to make an ϵ -move or read input and move.
- Accepts by final state.
- We need a right-end marker " ϵ " for the input.

Example DPDA

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

$$(s, a, \perp) \rightarrow (p, A\perp)$$
$$(p, a, A) \rightarrow (p, AA)$$
$$(p, b, A) \rightarrow (q, \epsilon)$$
$$(q, b, A) \rightarrow (q, \epsilon)$$
$$(q, \uparrow, \perp) \rightarrow (t, \perp)$$
$$(s, \uparrow, \perp) \rightarrow (t, \perp).$$

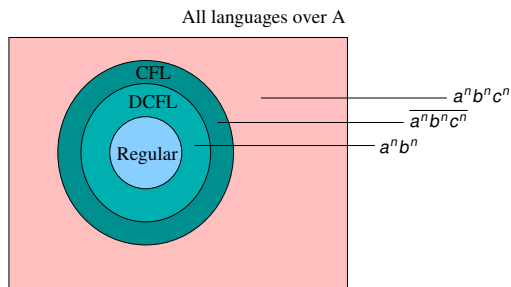
Example DPDA

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

$$(s, a, \perp) \rightarrow (p, A \perp)$$
$$(p, a, A) \rightarrow (p, AA)$$
$$(p, b, A) \rightarrow (q, \epsilon)$$
$$(q, b, A) \rightarrow (q, \epsilon)$$
$$(q, \uparrow, \perp) \rightarrow (t, \perp)$$
$$(s, \uparrow, \perp) \rightarrow (t, \perp).$$

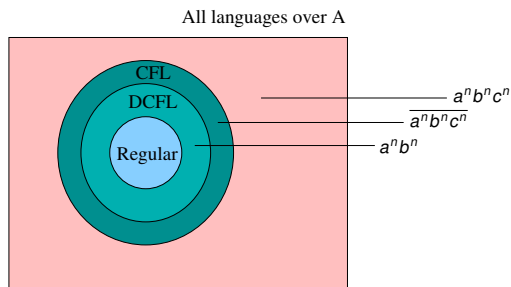
Class of languages accepted by DPDA's are called **DCFL's**.

Closure Properties of DCFL's



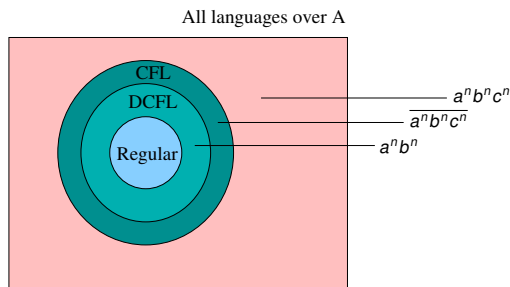
	Closed?
Complementation	

Closure Properties of DCFL's



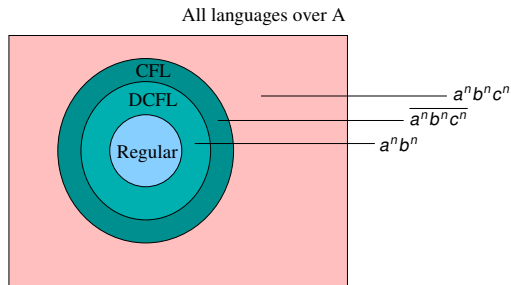
	Closed?
Complementation	✓
Union	

Closure Properties of DCFL's



	Closed?
Complementation	✓
Union	X
Intersection	

Closure Properties of DCFL's



	Closed?
Complementation	✓
Union	X
Intersection	X

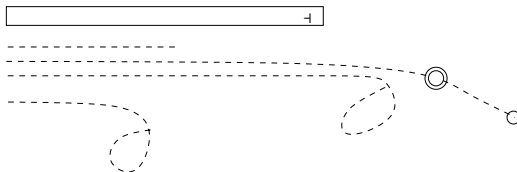
DCFL's are closed under complementation

Theorem (Closure under complementation)

The class of languages definable by Deterministic Pushdown Automata (i.e. DCFL's) is closed under complementation.

Problem with complementing a DPDA

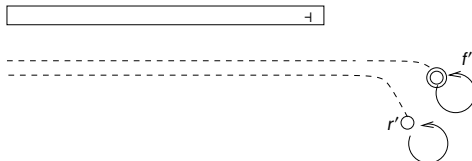
Try flipping final and non-final states.
Problems?



Loops denote an infinite sequence of ϵ -moves.

Desirable form of DPDA

Goal is to convert the DPDA into the form:



That is, always reads its input and reaches a final/reject sink state.

Then we can make r' the unique accepting state, to accept the complement of M .

Construction - Step 1

Let $M = (Q, A, \Gamma, s, \delta, \perp, F)$ be given DPDA. First construct DPDA M' which

- Does not get stuck due to no transition or stack empty.
- Has only “sink” final states.

Construction - Step 1

Define $M' = (Q \cup Q' \cup \{s_1, r, r'\}, A, \Gamma \cup \{\perp\}, s_1, \delta', \perp, F')$ where

- $Q' = \{q' \mid q \in Q\}$ and $F' = \{f' \mid f \in F\}$.
- δ' is obtained from δ as follows:
 - Assume M is “complete” (does not get stuck due to no transition). (If not, add a dead state and add transitions to it.)
 - Make sure M' never empties its stack, keep track of whether we have seen end of input (primed states) or not (unprimed states):

$$(s_1, \epsilon, \perp) \rightarrow (s, \perp \perp)$$

$$(p, \epsilon, \perp) \rightarrow (r, \perp) \quad (p \in Q)$$

$$(p', \epsilon, \perp) \rightarrow (r', \perp) \quad (p' \notin F')$$

$$(p, \uparrow, X) \rightarrow (q', \gamma) \quad \text{if } (p, \uparrow, X) \rightarrow (q, \gamma) \in \delta.$$

$$(p', \epsilon, X) \rightarrow (q', \gamma) \quad \text{if } (p, \epsilon, X) \rightarrow (q, \gamma) \in \delta.$$

$$(r, a, X) \rightarrow (r, X)$$

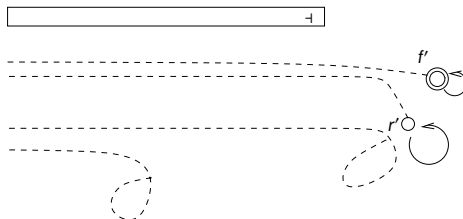
$$(r, \uparrow, X) \rightarrow (r', X)$$

$$(r', \epsilon, X) \rightarrow (r', X)$$

$$(f', \epsilon, X) \rightarrow (f', X) \quad (f \in F) \text{ Also drop trans. going from } f'.$$

After Step 1

DPDA M' only has the following kinds of behaviours now:



Loops denote an infinite sequence of ϵ -moves.

Construction - Step 2

A **spurious transition** in M' is a transition of the form $(p, \epsilon, X) \rightarrow (q, \gamma)$ such that

$$(p, \epsilon, X) \xRightarrow{*} (p, \epsilon, X\alpha)$$

for some stack contents α .



Identify **spurious transitions** in M' and remove them:

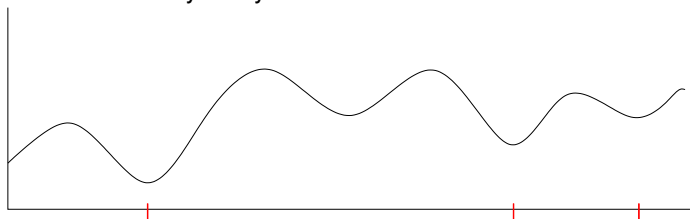
If $(p, \epsilon, X) \rightarrow (q, \gamma)$ is a spurious transition, replace it with

$$\begin{array}{ll} (p, \epsilon, X) \rightarrow (r, X) & \text{If } p \in Q \\ (p, \epsilon, X) \rightarrow (r', X) & \text{If } p \in Q' - F'. \end{array}$$

Correctness

Argue that:

- Deleting a spurious transition (starting from a non- F' -final state) does not change the language of M' .
- All infinite loops use a spurious transition.
 - Look at graph of stack height along infinite loop, and argue that there are infinitely many **future minimas**.



- Further look at transitions applied at these points and observe that one must repeat.
- Thus replacing spurious transitions as described earlier will remove the remaining undesirable loops from M' 's behaviours.

Complementing

- Resulting M'' has the desired behaviour (every run either reaches a final sink state or the reject sink state r').



- Now make r' unique final state to complement the language of M .

Detecting spurious transitions

Question: How can we effectively detect spurious transitions?

Detecting spurious transitions

Question: How can we effectively detect spurious transitions?

Use algorithm for pushdown reachability.