

EXERCISES

determinism

Aşağıdaki dillerin deterministik CFL olduğunu gösteriniz.

(a) $\{c^m b^m : m \geq 0\} \cup \{d^m b^{2m} : m \geq 0\}$

(b) $\{a^m c b^m : m \geq 0\} \cup \{a^m d b^{2m} : m \geq 0\}$

SORU 1) Aşağıdaki dilin deterministik CFL olduğunu gösteriniz.

$$\{c^m b^m : m \geq 0\} \cup \{d^m b^{2m} : m \geq 0\}$$

$L = \{c^m b^m : m \geq 0\} \cup \{d^m b^{2m} : m \geq 0\}$. Diğer iki dilin birleşiminden oluşan bir dil için deterministik bir PDA oluşturmak genellikle zordur. Örneğin,

$\{a^m b^m : m \geq 0\} \cup \{a^m b^{2m} : m \geq 0\}$ zor olurdu (aslında imkansız) çünkü b'ler bitene kadar her bir a için iki b mi yoksa sadece bir b mi beklediğimizi bilmemizin hiçbir yolu yoktur.

Fakat $\{c^m b^m : m \geq 0\} \cup \{d^m b^{2m} : m \geq 0\}$ aslında oldukça kolaydır. Her katar bir c veya bir d ile başlar. Eğer bir c ile başlıyorsa, o zaman her a için bir b aramamız gerektiğini biliyoruz; eğer bir d ile başlıyorsa, o zaman iki tane b aramamız gerektiğini biliyoruz.

Yani yaptığımız ilk şey, makinemizi şu şekilde başlatmak:

Durum 1'de başlayan makine, $a^n b^n$ için klasik makinemizdir, tabii ki son durumuna $\$$ üzerinde son bir geçiş yapması gerekir.

Durum 2'de başlayan makine için iki seçeneğimiz var.

Ya gördüğü her a için yığına bir a'ya atabilir ve ardından her b çifti için bir a çekebilir ya da gördüğü her a için iki a yığına atabilir ve ardından her b için bir a çekebilir.

SORU 2) Aşağıdaki dilin deterministik CFL olduğunu gösteriniz.

$$(a^mcb^m : m \geq 0) \cup \{a^mdb^{2m} : m \geq 0\}$$

Ancak bu kez ilk karaktere bakarak dilin hangi bölümüyle uğraştığımıza dair bir ipucu alamıyoruz. Yine de sorun yok, çünkü b'leri işlemeye başlamadan önce her bir a için iki b'ye mi yoksa sadece bir b'ye mi sahip olduğumuzu öğreniyoruz.

Burada ihtiyacımız olan şey, öncelikle gördüğü her a için yığına tek bir a atmak. Ardından, bir c veya d gördüğünde dallara ayrılarak, c durumunda her b için bir a çekeriz veya d durumunda her iki b gördüğümüzde bir a çekeriz.

Bağlamdan bağımsız grameri (CFG) ele alalım:

$G = (V, \Sigma, R, S)$, burada

$V = \{ (,), ., a, S, A \}$,

$\Sigma = \{ (,), . \}$, ve kurallar

$R = \{$ $S \rightarrow ()$,

$S \rightarrow a$,

$S \rightarrow (A)$,

$A \rightarrow S$,

$A \rightarrow A.S \}$ olsun

(a) Yukarıdan aşağıya bir ayrıştırma (top down parsing) yaparak $L(G)$ 'in dilini kabul eden deterministik bir PDA Makinesi oluşturun. $((()))$ 'a dizisindeki M'nin hesaplanmasını inceleyin.

Heuristic Rule 1: Eğer

$$A \rightarrow \alpha \beta_1, A \rightarrow \alpha \beta_2, \dots, A \rightarrow \alpha \beta_n,$$

şeklinde kurallar varsa ve $\alpha \neq \epsilon$ ve $n \geq 2$ ise,

bu kurallar $A \rightarrow \alpha A', A' \rightarrow \beta_i$

kurallarıyla değiştirilir. A' yeni nonterminaldir.

(a) We need to apply left factoring to the two rules $S \rightarrow ()$ and $S \rightarrow (A)$. We also need to eliminate the left recursion from $A \rightarrow A.S$. Applying left factoring, we get the first column shown here. Then getting rid of left recursion gets us the second column:

$S \rightarrow (),$	$S \rightarrow (S'$	$S \rightarrow (S'$
$S \rightarrow a,$	$S' \rightarrow)$	$S' \rightarrow)$
$S \rightarrow (A),$	$S' \rightarrow A)$	$S' \rightarrow A)$
$A \rightarrow S,$	$S \rightarrow a$	$S \rightarrow a$
$A \rightarrow A.S$	$A \rightarrow S$	$A \rightarrow SA'$
	$A \rightarrow A.S$	$A' \rightarrow .SA'$
		$A' \rightarrow \varepsilon$

Heuristic Rule 1: Eğer

$$A \rightarrow \alpha \beta_1, A \rightarrow \alpha \beta_2, \dots, A \rightarrow \alpha \beta_n,$$

şeklinde kurallar varsa ve $\alpha \neq \varepsilon$ ve $n \geq 2$ ise,

bu kurallar $A \rightarrow \alpha A', A' \rightarrow \beta_i$

kurallarıyla değiştirilir. A' yeni nonterminaldir.