Biçimsel Diller ve Otomata Teorisi

Dr. Öğr. Üyesi Hayri Volkan Agun

Bilgisayar Mühendisliği Bölümü

Bursa Teknik Üniversitesi

Kaynaklar

Ders Kitabı

- An Introduction to Formal Languages and Autotamata, Peter Linz, 6th Edition, 2017.
- An Introduction to Computer Theory, Daniel Isaac Aryeh Cohen, 2nd Edition, 1996.

İçerik

- %100 Teorik
- Klasik sınav
- Vize %40, Final %60

Pushdown Otomat

Yığıtlı (Pushdown) Otomat

- Düzenli dillerin tartışmasında, düzenli dilleri keşfetmenin birkaç yolu olduğunu gördük: sonlu otomat, düzenli gramerler ve düzenli ifadeler. Bağlam bağımsız dilleri bağlam bağımsız gramerler aracılığıyla tanımladıktan sonra, şimdi başka seçeneklerin olup olmadığını soruyoruz. Ortaya çıkıyor ki, düzenli ifadelerin bir karşılığı yok, ancak yığınsal otomatlar bağlam bağımsız dillerle ilişkilendirilen otomat türüdür.
- Yığınsal otomatlar, onlara kararsızlık izni verdiğimiz sürece, bağlam bağımsız gramerlere eşdeğerdir. Kararlı yığınsal otomatları da tanımlayabiliriz, ancak bunlarla ilişkilendirilen dil ailesi, bağlam bağımsız dillerin uygun bir alt kümesidir.

Yığın (Pushdown) Otomatı

Yığınlı (Pushdown) Otomat

- Örneğin, L = {a^nb^n : n ≥ 0} dilinden bir dizeyi tararken, sadece tüm a'ların ilk b'den önce geldiğini kontrol etmekle kalmamalı, aynı zamanda a'ların sayısını da saymalıyız. Çünkü n sınırsızdır, bu sayım sınırlı bir bellekle yapılamaz.
- Sınırsız sayma yeteneğine sahip bir makine istiyoruz. Ancak diğer örneklerden, örneğin {ww^R}, gördüğümüz gibi, sadece sınırsız sayma yeteneğinden daha fazlasına ihtiyacımız var: Sıralı sembollerin ters sırasında bir dizi saklama ve eşleştirme yeteneğine ihtiyaç duyuyoruz. Bu, depolama mekanizması olarak bir yığını deneyebileceğimizi düşündürmektedir, sınırsız depolamaya izin veren ancak bir yığın gibi çalışması kısıtlanmış bir mekanizma. Bu bize yığınsal otomat (pda) adı verilen bir makine sınıfını verir.

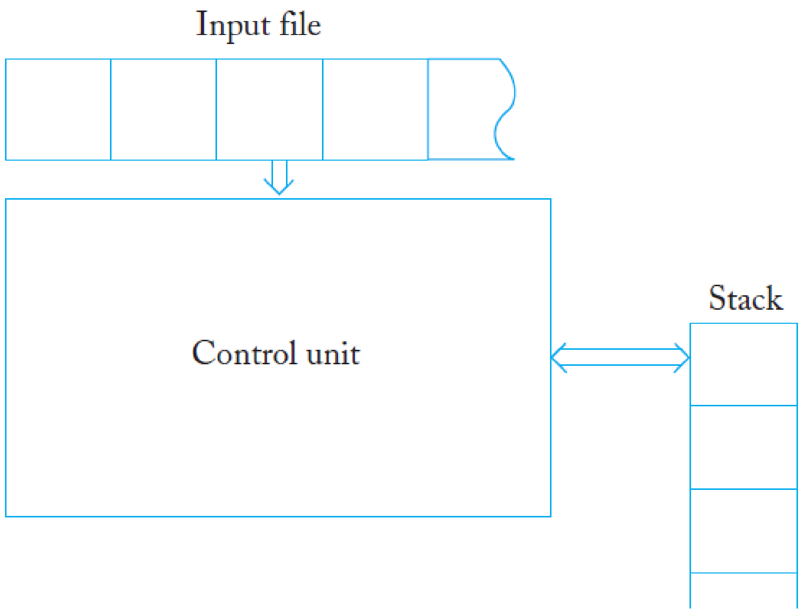
Yığın (Pushdown) Otomatı

- Kararsız sonlu yığıt otomatını tanımlarken aşağıdaki gibi ekstra yığtıtın durumuna da kattığımız bir tanıma ihtiyaç duyarız.

Yığıt alfabesi Yığıt başlangıcı

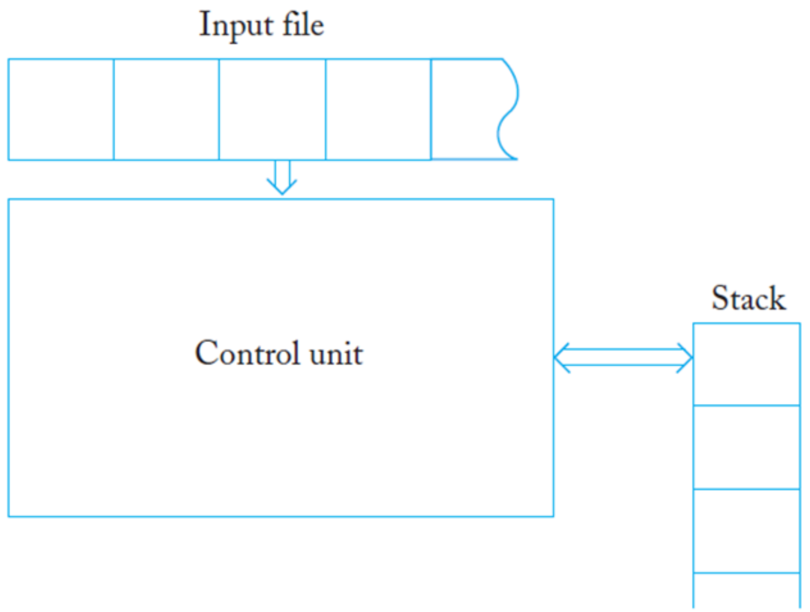
$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F),$$

Durumlar Alfabe Başlangıç durumu Son durumlar



Yığın Otomatı Örnek

- Varsayalım ki bir npda'nın geçiş kuralları kümesi, $\delta(q1, a, b) = \{(q2, cd), (q3, \lambda)\}$ içeriyor.
- Herhangi bir zamanda kontrol birimi durum q1'de ise, okunan giriş sembolü a ve yığının üstündeki sembol b ise, iki şeyden biri olabilir: (1) kontrol birimi durumu q2'ye geçer ve cd dizisi yığının üstündeki b'nin yerine geçer, ya da (2) kontrol birimi durumu q3'e geçer ve yığının üstünden sembol b çıkarılır. Notasyonumuzda, bir dizinin yığına eklenmesinin, sembole sağ ucundan başlanarak yapıldığını varsayarız.



Kararsız Yığın (Pushdown) Otomatı

- Yığıt (Pushdown) otomatı için yandaki tanımlamaya göre:
  a) Durumlar: q0, q1, q2, ve q3
  b) Girdi alfabesi: a ve b
  c) Yığıt alfabesi: 0 yada 1
  d) Yığıt başlangıçta 0 barındırıyor.
  e) Son durum q3.
  f) Başlangıç belirtilmemiş.

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3\}, \\
\Sigma &= \{a, b\}, \\
\Gamma &= \{0, 1\}, \\
z &= 0, \\
F &= \{q_3\},
\end{aligned}
$$

Kararsız Yığın (Pushdown) Otomatı

- 1. Kurala göre q0 durumunda iken girdi alfabesi a ve yığıtın en üsteki elemanı 0 ise
  - q1 durumuna geçilir ve stack içine 1 (eklenen solda) eklenir.
  - q3 durumuna geçilir ve yığıttan 0 çıkartılır. Burada λ **gördüğümüzde yığıttan çıkarma işlemi** yapılmıştır.
- 3. Kurala göre q1 durumunda iken girdi alfabesi a ve yığıtın en üstündeki elemanı 1 ise
  - q1 durumuna geçilir ve yığıta 1 eklenir.

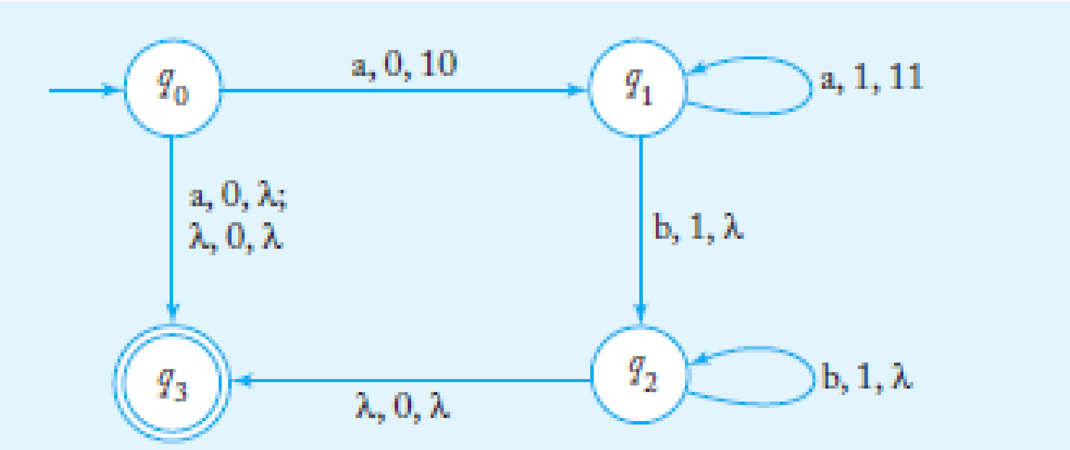
$$
\begin{aligned}
\delta(q_0, a, 0) &= \{(q_1, 10), (q_3, \lambda)\}, \\
\delta(q_0, \lambda, 0) &= \{(q_3, \lambda)\}, \\
\delta(q_1, a, 1) &= \{(q_1, 11)\}, \\
\delta(q_1, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, \lambda, 0) &= \{(q_3, \lambda)\}.
\end{aligned}
$$

Kararsız Yığın (Pushdown) Otomatı

- Geçiş kuralları yanda belirtilen kararsız yığıt otomatını çiziniz.
- Kuralları verilen bu kararsız yığıt otomatı hangi dili yakalayabilir. Gösteriniz?

$$
\begin{aligned}
\delta(q_0, a, 0) &= \{(q_1, 10), (q_3, \lambda)\}, \\
\delta(q_0, \lambda, 0) &= \{(q_3, \lambda)\}, \\
\delta(q_1, a, 1) &= \{(q_1, 11)\}, \\
\delta(q_1, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, b, 1) &= \{(q_2, \lambda)\}, \\
\delta(q_2, \lambda, 0) &= \{(q_3, \lambda)\}.
\end{aligned}
$$

Kararsız Yığın (Pushdown) Otomatı



$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\},$$
$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\},$$
$$\delta(q_1, a, 1) = \{(q_1, 11)\},$$
$$\delta(q_1, b, 1) = \{(q_2, \lambda)\},$$
$$\delta(q_2, b, 1) = \{(q_2, \lambda)\},$$
$$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}.$$

Dil Tanımı

- Bu dil içinde geçen

$$\delta\left(q_1, a, 1\right)=\left\{\left(q_1, 11\right)\right\},$$

Kuralı ile a gördükçe yığıt içine 1 eklenmektedir. Aşağıdaki kuralda ise b gördükçe yığıtın üstündeki 1 elemanı çıkarılmaktadır.

$$\delta\left(q_2, b, 1\right)=\left\{\left(q_2, \lambda\right)\right\}$$

Ayrıca a gördüğümüzde sonlu duruma ulaşılmaktadır.

$$\delta\left(q_0, a, 0\right)=\left\{\left(q_1, 10\right),\left(q_3, \lambda\right)\right\},$$

$$L=\left\{a^n b^n: n \geq 0\right\} \cup\{a\}.$$

Örnek

$$L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}.$$

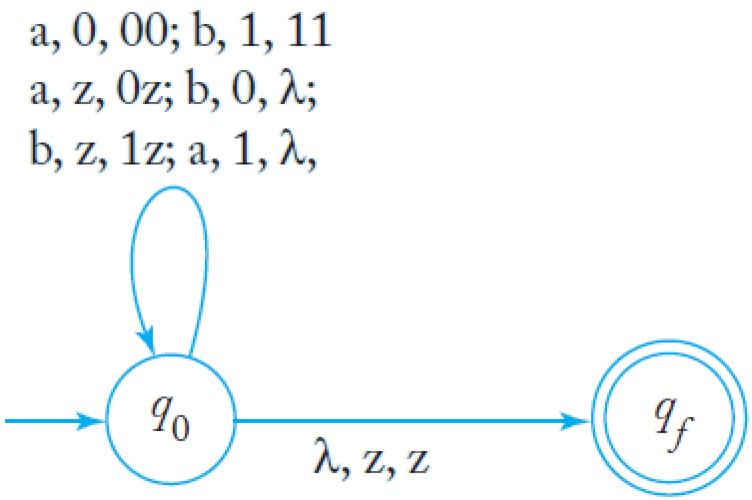
- Yanda tanımı verilen dil aabbabab şeklinde a sayısının b sayısına eşit olduğu tüm karakter katarlarını yakalamaktadır.
- Burada a ve b karışık olarak geçmektedir. Bu dile ait karakter katarlarını yakalayan otomatı çiziniz?

Örnek

$$L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}.$$

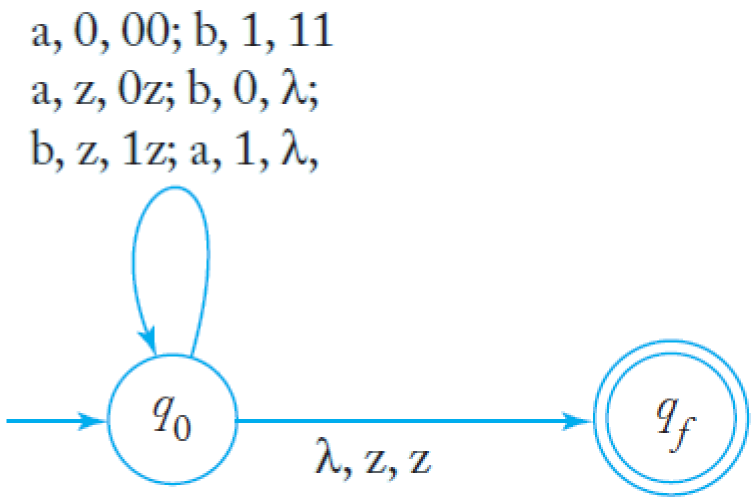
- Bu otomat ile a gördükçe 1 çıkartıp 0 ekleriz b gördükçe 0 çıkartıp 1 ekleriz.
- Ancak burada ekleme ve döngü için geçiş durumuz a için yığıt tepesinde 1 olma yada hiçbirşey olmama durumu, b için yığıt tepesinde 0 olma yada hiçbirşey olmama durumudur.

Örnek



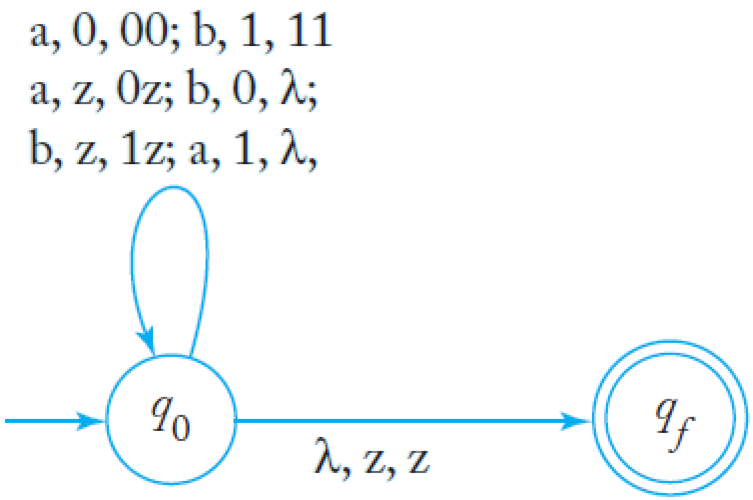
- Bu otomat ile a gördükçe 1 çıkartıp 0 ekleriz b gördükçe 0 çıkartıp 1 ekleriz.
- Ancak burada ekleme ve döngü için geçiş durumuz a için yığıt tepesinde 1 olma yada hiçbirşey olmama durumu, b için yığıt tepesinde 0 olma yada hiçbirşey olmama durumudur.
- Herhangş bir zamanda yığıt tepesinde 0 yada 1 olduğu durumda boş geçiş ile son duruma gidebiliriz.

Örnek



- Bu otomat ile baab karakter katarının hangi durumlar ile yakalandığını adım adım gösteriniz?
- Burada baab tüketildiğinde kaç kere q0 ve son olarak q1 durumuna hangi geçişler ile gidilmiştir, gösteriniz?

Örnek



girdi	yığıt	geçiş	yeni yığıt	yeni durum
baab	z	q0 -> (b,0,1z)	1z	q0
aab	1z	q0 -> (a,1,λ)	z	q0
ab	z	q0 -> (a,z,0z)	0z	q0
b	0z	q0 -> (b,0,λ)	z	q0
λ	z	q0 -> (λ,z,z)	z	qf

Örnek 2

- Aşağıda tanımı yapılan dili kabul eden otomatı oluşturunuz.

$$L = \left\{ ww^R : w \in \{a, b\}^+ \right\},$$

- Burada her hangi bir karakter katarının belirli bir orta noktadan simetriğini kabul eden bir dil tanımı yapılmıştır.
- Örneğin aaabbb bbbaaa şeklinde bir karakter katarı bu dile ait olacaktır.
- Bu dile ait olan yığıt otomatını tanımlayalım. Öncelikle semboller a, b ve yığıt boş durumu z ile ifade edilsin.

Örnek 2

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2\}\,, \\
\Sigma &= \{a, b\}\,, \\
\Gamma &= \{a, b, z\}\,, \\
F &= \{q_2\}\,.
\end{aligned}
$$

- Tanımlamada q0 başlangıç ve q2 son durumu ifade etmektedir. Burada yapmamız gereken q1 içinde herhangi bir girdi için ya yığıtı besleyerek karakterleri buraya kaydedeceğiz ya da orta noktaya ulaşaıldığını farz edip yığıtını boşaltma işlemine geçeceğiz.
- Yığıtı doldruma için tüm olası durumları belirleyelim.

Örnek 2

$$\begin{aligned}
\delta(q_0, a, a) &= \{(q_0, aa)\}, \\
\delta(q_0, b, a) &= \{(q_0, ba)\}, \\
\delta(q_0, a, b) &= \{(q_0, ab)\}, \\
\delta(q_0, b, b) &= \{(q_0, bb)\}, \\
\delta(q_0, a, z) &= \{(q_0, az)\}, \\
\delta(q_0, b, z) &= \{(q_0, bz)\},
\end{aligned}$$

- Yığıtı doldruma için tüm olası durumları belirleyelim. Burada a gördüğümüzde yığıtın tepesi boş, a yada b olabilir. Benzer şekilde b gördüğümüzde yığıtın tepesi boş, a yada b olabilir.
- Toplam q0 durumunda q 0 a toplam 6 kural oluşturduk.
- Herhangi bir zamanda karakter katarının orta noktası farz edip boşaltma işlemine geçmeliyiz. Bunu kararsız özellikte bir geçiş kullanarak çözebiliriz.

Örnek 2

$$\delta(q_0, \lambda, a) = \{(q_1, a)\},$$
$$\delta(q_0, \lambda, b) = \{(q_1, b)\},$$

$$\delta(q_1, a, a) = \{(q_1, \lambda)\},$$
$$\delta(q_1, b, b) = \{(q_1, \lambda)\},$$

$$\delta(q_1, \lambda, z) = \{(q_2, z)\},$$

- Herhangi bir zamanda karakter katarının orta noktası farz edip boşaltma işlemine geçmeliyiz. Bunu kararsız özellikte bir geçiş kullanarak çözebiliriz.
- Burada q1 durumunda boşaltma için yığıtın tepesi ile mevcut sembol eş mi (a yada b) diye kontrol ederek işlemi gerçekleştirmiş oluruz.
- Son olarak yığtın tepesi boş ise son duruma gideriz.

Örnek 2

- *abba* karakter katarını aşağıdaki kural açılımları ile yakalayabiliriz.

girdi	yığın	geçiş	yeni yığın	yeni durum
abba	z	q0 -> (a, z, az)	a	q0
bba	a	q0 -> (b, a, ba)	ba	q0
ba	ba	q0 -> (λ,b, b)	ba	q1
ba	ba	q1 -> (b, b, λ)	a	q1
a	a	q1 -> (a, a, λ)	z	q1
λ	z	q1 -> (λ, z, λ)	z	q2

Soru

- Aşağıdaki dil tanımları için kararsız yığın otamatını (npda) tanımlayınız.

$$L = \{a^n b^{2n} : n \geq 0\}.$$

$$L = \{a^n b^m : n \leq m \leq 3n\}.$$

Soru

$$L = \{a^n b^{2n} : n \geq 0\}.$$

- Bu dil tanımı için tanımlanan otomatta önce a sembolleri için yığın içerisine 1 koyarız.
- Sonra her bir b sembolü ve yığın üst değeri 1 için yığın içindeki her 1 değerini çıkartırız ve yerine 0 değeri koyarız.
- Her bir b sembolü ve yığın üst değeri 0 için 0 değerini çıkartırız.
- Yığın boş olduğundan boş geçişle son duruma gideriz.