HAFTA 4

2.2 – Nondeterministic Finite Automata

Bu bölümde sonlu otomatlara güçlü ve ilgi çekici bir özellik ekliyoruz. Bu özellik nondeterminizm olarak adlandırılır ve esasen mevcut durum ve girdi sembolü tarafından sadece kısmen belirlenen bir şekilde durumları değiştirme yeteneğidir. Yani, mevcut durum ve girdi sembolünün belirli bir kombinasyonu için birkaç olası “sonraki duruma” izin vereceğiz. Otomat, girdi dizesini okurken, her adımda bu yasal sonraki durumlardan herhangi birine gitmeyi seçebilir; seçim modelimizdeki herhangi bir şey tarafından belirlenmez ve bu nedenle belirsiz olduğu söylenir. Öte yandan, seçim tamamen sınırsız da değildir; sadece belirli bir girdi sembolü ile belirli bir durumdan yasal olan sonraki durumlar seçilebilir.

Bu tür nondeterministik cihazlar bilgisayarların gerçekçi modelleri olarak düşünülmemelidir. Bunlar sadece sonlu otomatların yararlı bir notasyonel genellemesidir, çünkü bu otomatların tanımını büyük ölçüde basitleştirebilirler. Dahası, aşağıda göreceğiz ki belirlenimsizlik sonlu otomatların vazgeçilmez bir özelliğidir: her belirlenimsiz sonlu otomat, belirlenimli bir sonlu otomata eşdeğerdir. Böylece deterministik olmayan sonlu otomatların güçlü notasyonundan faydalanacağız ve eğer gerekirse her zaman geri dönüp her şeyi sıradan, ayakları yere basan deterministik otomatların alt düzey diliyle yeniden yapabileceğimizi bileceğiz.

taslak, çizim, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Deterministik olmayan bir sonlu otomatın deterministik bir sonlu otomattan çok daha kullanışlı bir tasarım aracı olabileceğini görmek için Şekil 2-4'te gösterilen deterministik sonlu otomat tarafından kabul edilen L = (ab U aba)\* dilini ele alalım. Diyagramla bile, deterministik bir sonlu otomatın gösterildiğini anlamak birkaç dakika alır; her düğümden biri a diğeri b olarak etiketlenmiş tam olarak iki ok çıktığını kontrol etmek gerekir. Ve bu oldukça karmaşık cihaz tarafından kabul edilen dilin basit dil (ab U aba)\* olduğuna kendini ikna etmek için biraz düşünmek gerekir. L dilini kabul eden daha basit bir deterministik sonlu otomat bulmayı umabiliriz; ne yazık ki beşten az durumu olan hiçbir deterministik sonlu otomatın bu dili kabul edemeyeceği gösterilebilir (bu bölümün ilerleyen kısımlarında deterministik sonlu otomatların durum sayısını minimize etmek için yöntemler geliştireceğiz).

Ancak, L Şekil 2-5'te gösterilen basit deterministik olmayan aygıt tarafından kabul edilir. Bu cihaz q1 durumundayken ve giriş sembolü b iken, q0 ve q2 olmak üzere iki olası sonraki durum vardır. Dolayısıyla Şekil 2-5 deterministik bir sonlu otomatı temsil etmemektedir. Bununla birlikte, diyagramı L'yi kabul eden bir cihaz olarak yorumlamanın doğal bir yolu vardır. Bir dizi, dizinin sembolleriyle etiketlenmiş okları takip ederken başlangıç durumundan (q0) son duruma (bu durumda q0) ulaşmanın bir yolu varsa kabul edilir. Örneğin ab, q0'dan q1'e, q0'a gidilerek kabul edilir; aba, q0'dan q1'e, q2'ye, q0'a gidilerek kabul edilir. Elbette, cihaz yanlış tahmin edebilir ve aba girdisinde q0'dan q1'e, q0'dan q1'e giderek nihai olmayan bir duruma geçebilir; ancak bu önemli değildir, çünkü bu girdi ile başlangıç durumundan nihai duruma geçmenin bir yolu vardır. Öte yandan, abb girdisi kabul edilmez, çünkü bu dizeyi okurken q0'dan q0'a geri dönmenin bir yolu yoktur.

taslak, çizim, beyaz, tasarım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Aslında, q0'dan itibaren girdi b olduğunda girilecek bir durum olmadığını fark edeceksiniz. Bu, nondeterministik sonlu otomatların bir başka özelliğidir: bazı girdilerle bazı durumlardan birkaç olası sonraki durum olabileceği gibi, diğer durum ve girdi sembolleri kombinasyonlarıyla da olası hamleler olmayabilir.

Ayrıca, belirsiz bir otomatın durum diyagramında boş dizgi e ile etiketlenmiş oklara da izin veriyoruz. Örneğin, Şekil 2-6'daki cihaz aynı L dilini kabul eder. q2'den bu makine ya bir a okuyarak ya da herhangi bir girdi tüketmeden hemen q0'a dönebilir.

taslak, çizim, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 2-5 ve 2-6'da gösterilen cihazlar aşağıdaki genel tipin örnekleridir:

metin, ekran görüntüsü, yazı tipi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, beyaz, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Her (q,u, p) ∈ △ üçlüsüne M'nin bir geçişi denir - M'nin durum diyagramında q'dan p'ye a etiketli bir okun biçimsel karşılığı. M q durumundaysa ve bir sonraki giriş sembolü a ise, M daha sonra (q,a, p) veya (q,e, p) biçimindeki herhangi bir geçişi takip edebilir; (q,e, p) geçişi takip edilirse, o zaman hiçbir giriş sembolü okunmaz.

Deterministik olmayan sonlu otomatların hesaplamalarının resmi tanımları deterministik sonlu otomatlar için olanlara çok benzerdir. M'nin bir konfigürasyonu, bir kez daha, K x ∑\*'in bir elemanıdır. Konfigürasyonlar arasındaki I--M ilişkisi (bir adımda elde edilir) aşağıdaki gibi tanımlanır: (q, w) I--M (q', w') ancak ve ancak bir u ∈ ∑ U {e} varsa, öyle ki w= uw' ve (q,u, q') ∈ △ olsun. I--M'nin bir fonksiyon olması gerekmediğine dikkat edin; bazı (q,w) konfigürasyonları için, (q,w) I--M (q', w') olacak şekilde birkaç (q', w') çifti olabilir veya hiç olmayabilir. Daha önce olduğu gibi, 'I--\*M' I--M'nin dönüşlü, geçişli kapanışıdır ve bir w ∈ ∑\* dizesi ancak ve ancak (s,w) I--\*M (q,e) olacak şekilde bir q ∈ F durumu varsa M tarafından kabul edilir. Son olarak L(M), M tarafından kabul edilen dil, M tarafından kabul edilen tüm dizgilerin kümesidir.

Örnek 2.2.1: Şekil 2-7, bb veya bab örüntüsünün bir oluşumunu içeren tüm dizgilerin kümesini kabul eden birkaç olası deterministik olmayan sonlu otomattan birini göstermektedir (giriş dizgisindeki örüntüleri tespit etmeye yönelik otomatların sistematik bir çalışması için Bölüm 2.5'e bakınız). Biçimsel olarak bu makine (K,∑, △,s, F) şeklindedir, burada:

metin, yazı tipi, ekran görüntüsü, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

M'ye girdi olarak bababab dizesi verildiğinde, birkaç farklı hamle dizisi ortaya çıkabilir. Örneğin, M aşağıdaki durumlarda nihai olmayan q0 durumuna gelebilir:

diyagram, taslak, çizim, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Kullanılan tek geçişler (q0,a,q0) ve (q0,b,q0)'dır.

metin, yazı tipi, el yazısı, hat sanatı, kaligrafi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Aynı girdi dizisi M'yi q0 durumundan q4 son durumuna götürebilir ve bunu üç farklı şekilde yapabilir. Bu yollardan biri aşağıdaki gibidir.

metin, yazı tipi, ekran görüntüsü, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

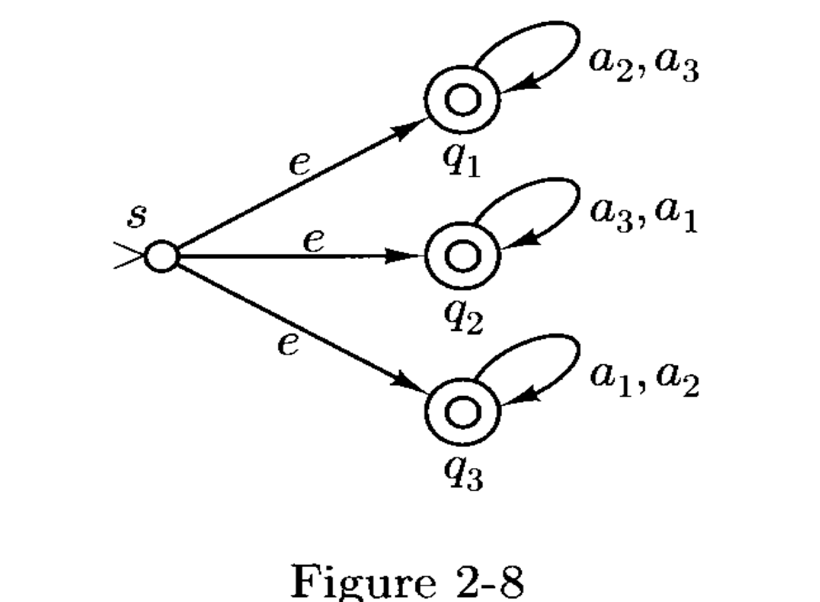
Bir dizge deterministik olmayan bir sonlu otomat tarafından ancak ve ancak son duruma götüren en az bir hamle dizisi varsa kabul edildiğinden, bababab ∈ L(M) olduğu sonucu çıkar.

Örnek 2.2.2: ∑, n ≥ 2 olmak üzere n sembol içeren ∑ = {a1,..., an} alfabesi olsun ve aşağıdaki dili göz önünde bulundurun:

L = {w: w'de görünmeyen bir ai ∈ ∑ sembolü vardır}.

Yani, L, ∑\* içinde ∑'deki tüm sembollerin oluşumlarını içermeyen tüm dizeleri içerir. Örneğin, n = 3 ise, e, a1,a2,a1a1a3a1∈ L, ancak a3a1a3a1a2 ∉ L'dir.

Bu oldukça karmaşık dili kabul eden nondeterministik bir sonlu otomat M = (K,∑, △, s, F) tasarlamak nispeten kolaydır. Burada K, hepsi (F = K) kabul eden n + 1 durum K = {s, q1,q2.., qn} içerir. △ iki tür geçişe sahiptir (n = 3 durumunun bir örneği için Şekil 2-8'e bakınız). İlk geçişler, tüm i, 1 ≤ i ≤ n için (s,e, qi) biçiminde olanlardır ve ana geçişler, i ≠ j olan (qi, aj, qi) biçimindeki tüm üçlülerdir. Bu, △'deki geçişlerin listesini tamamlar.



Sezgisel olarak, M bir girdi üzerindeki işlemine, girdide eksik olan sembolü tahmin ederek ve ilgili duruma geçerek başlar. Eğer seçilen sembol ai ise, qi durumu ziyaret edilir. Bu durumda otomat, tahmin edilen sembolün gerçekten de dizede bulunup bulunmadığını kontrol eder. Eğer öyleyse, kabul ile sonuçlanır. Bu otomat, deterministik olmayan cihazların olağanüstü gücünü canlı bir şekilde göstermektedir: tahmin edebilirler ve her zaman doğru olabilirler, çünkü kabul için gereken tek şey başarılı bir hesaplamadır. Daha sonra Bölüm 2.5'te göreceğimiz gibi, aynı dili kabul eden herhangi bir deterministik sonlu otomat çok daha karmaşık olmalıdır.

Deterministik bir sonlu otomat, deterministik olmayan bir sonlu otomatın sadece özel bir türüdür: Deterministik bir sonlu otomatta, △ geçiş ilişkisinin aslında K x ∑ 'den K'ya bir fonksiyon olduğu görülür. Yani, terministik olmayan bir sonlu otomat (K, ∑, δ, s, F) ancak ve ancak △ içinde (q,e,p) biçiminde geçişler yoksa ve her q ∈ K ve a ∈ ∑ için (q,a,p) ∈ △ olacak şekilde tam olarak bir p ∈ K varsa deterministiktir. Bu nedenle, deterministik otomatlar tarafından kabul edilen diller sınıfının, deterministik olmayan otomatlar tarafından kabul edilen diller sınıfının bir alt kümesi olduğu açıktır. Oldukça şaşırtıcı bir şekilde, bu sınıflar aslında eşittir. Belirlenimci olmayan otomatların sahip olduğu görünürdeki güç ve genelliğe rağmen, kabul ettikleri diller açısından belirlenimci otomatlardan daha güçlü değildirler: Deterministik olmayan bir sonlu otomat her zaman eşdeğer bir deterministik otomata dönüştürülebilir.

Biçimsel olarak, iki sonlu otomat M1 ve M2'nin (deterministik veya nondeterministik) ancak ve ancak L(M1) = L(M2) ise eşdeğer olduğunu söyleriz. Bu nedenle, iki otomat, bunu yapmak için “farklı yöntemler kullansalar” bile, aynı dili kabul ediyorlarsa eşdeğer olarak kabul edilirler. Örneğin, Şekil 2-4--2-6'daki üç otomatın hepsi eşdeğerdir.

çizgi, diyagram, daire içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Örnek 2.2.3: Şekil 2-9'daki otomatta E(q0) = {q0, q1,q2, q3}, E(q1) = {q1,q2, q3} ve E(q2) = {q2}'dir.

Şimdi M'ye denk olan M' =(K', ∑, δ', s', F') deterministik otomatını resmi olarak tanımlamaya hazırız,

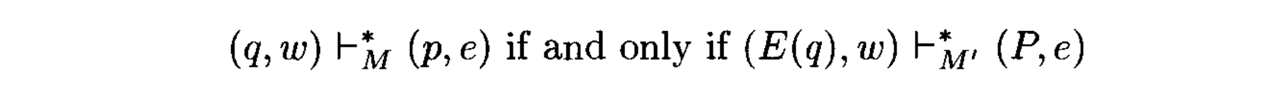
metin, makbuz, yazı tipi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Yani, δ'(Q, a), M'nin a girdisini okuyarak (ve muhtemelen birkaç e geçişini takip ederek) gidebileceği tüm M durumlarının kümesi olarak alınır. Örneğin, M Şekil 2-9'daki otomat ise, o zaman s' = {q0, q1, q2, q3). q1'den a girişine tek geçişler (q1,a, q0) ve (q1,a,q4) olduğundan, δ'({q1}, a) = E(q0) U E(q4) = {q0,q1,q2, q3,q4} olur.

Geriye M' nin deterministik ve M'ye denk olduğunu göstermek kalıyor. M' nin deterministik olduğunu göstermek basittir: δ' nin tek değerli ve tüm Q ∈ K' ve a ∈ ∑ üzerinde iyi tanımlı olduğunu fark etmemiz yeterlidir. (Bazı Q ∈ K' ve a ∈ ∑ için δ'(Q, a) = 0 olması δ''nın iyi tanımlı olmadığı anlamına gelmez; Ø, K' 'nın bir üyesidir).

Şimdi herhangi bir w ∈ ∑\* dizisi ve herhangi bir p, q ∈ K durumu için şunu iddia ediyoruz,



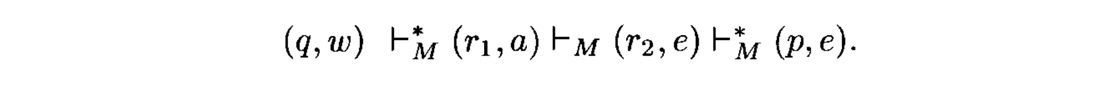
M ve M' nin eşdeğer olduğunu göstermek için, herhangi bir w ∈ ∑\* dizisini düşünün. O zaman w ∈ L(M) ancak ve ancak bazı f ∈ F için (s,w) I--\*M (f,e) ise (tanım gereği) ve ancak ve ancak f içeren bazı Q için (E(s), w)I--\*M, (Q,e) ise (yukarıdaki iddia gereği); başka bir deyişle, ancak ve ancak bazı Q ∈ F' için (s',w) I--\*M (Q,e) ise. Son koşul w ∈ L(M')'nin tanımıdır. |w| üzerinde tümevarım yoluyla iddiayı kanıtlıyoruz.

Temel Adım. |w| = 0 için - yani w = e için - p içeren bazı P kümeleri için (q,e) I--\*M (p,e) ancak ve ancak (E(q), e) I--\*M (P, e) ise göstermeliyiz. İlk ifade p ∈ E(q) demekle eşdeğerdir. M' deterministik olduğundan, ikinci ifade P = E(q) ve P p içerir demekle eşdeğerdir; yani p ∈ E(q). Bu, temel adımının kanıtını tamamlar.

Tümevarım Hipotezi. Bazı k >= 0 için uzunluğu k veya daha az olan tüm w dizgileri için iddianın doğru olduğunu varsayalım.

Tümevarım Adımı. İddiayı k + 1 uzunluğundaki herhangi bir w dizisi için kanıtlıyoruz. w = va olsun, burada a ∈ ∑ ve v ∈ ∑\*.

Sadece eğer yönü için, (q,w) I--\*M (p,e) olduğunu varsayalım. O zaman r1 ve r2 durumları vardır, öyle ki:



Yani, M q durumundan p durumuna v girdisinin okunduğu belli sayıda hamle, ardından a girdisinin okunduğu bir hamle ve ardından hiçbir girdinin okunmadığı belli sayıda hamle ile ulaşır. Şimdi (q, va) I--\*M (r1, a), (q,v) I--\*M (r1, e) ile eşdeğerdir ve |v| = k olduğundan, tümevarım hipotezi ile r1 içeren bazı R1 kümeleri için (E(q), v) I--\*M (R1,e). (r1, a) I--M (r2,e) olduğundan, bir (r1, a, r2) ∈ △ üçlüsü vardır ve dolayısıyla M' yapısı gereği E(r2) ⊆ δ'(R1,a)'dır. Ancak (r2, e) I--\*M (p,e) olduğundan, p ∈ E(r2) ve dolayısıyla p ∈ δ'(R1,a) olur. Bu nedenle p içeren bazı P'ler için (R1,a) I--M' (P,e) ve (E(q),va) I--\*M' (R1,a) I--M' (P,e).

Diğer yönü kanıtlamak için, p içeren bazı P ve δ'(R1,a) = P olacak şekilde bazı R1 için (E(q), va) I--\*M' (R1, a) I--M' (P, e) olduğunu varsayalım. Şimdi δ' tanımına göre, δ'(R1,a), bazı r1 ∈ R1 durumu için (r1,a, r2)'nin M'nin bir geçişi olduğu tüm E(r2) kümelerinin birleşimidir. p ∈ P = δ'(R1,a) olduğundan, öyle bir r2 vardır ki p ∈ E(r2) olsun ve bazı r1 ∈ R1 için (r1,a,r2) M'nin bir geçişi olsun. O halde (r2, e)I--\*M (p,e) E(r2)'nin tanımı gereğidir. Ayrıca tümevarım hipotezi ile (g, v) I--\*M (r1, e) ve dolayısıyla (q, va) I--\*M (r1,a) I--M (r2,e) I--\*M (p,e).

ÖDEV =  2.2.1, 2.2.2, 2.2.3, 2.2.6(a)