HAFTA-3

2.1- Deterministic Finite Automata

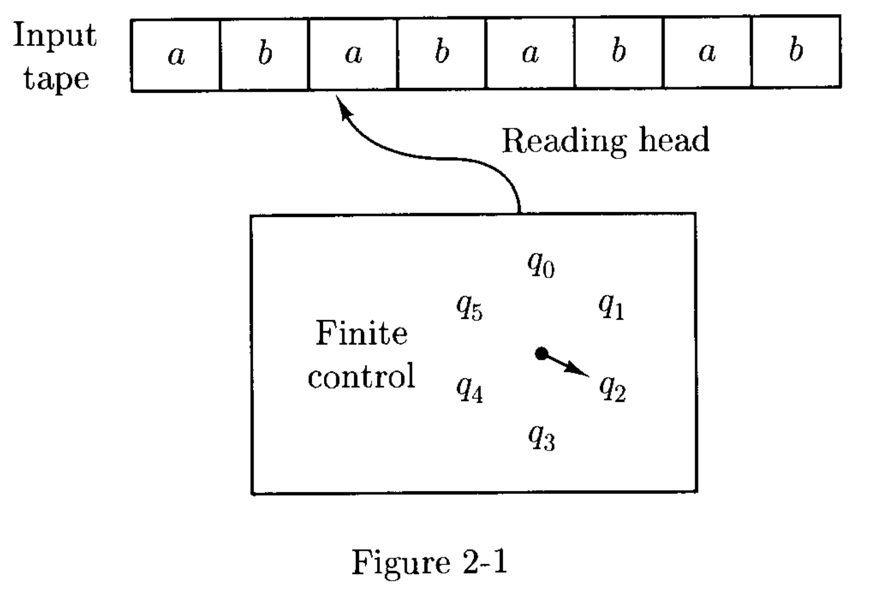
Bu kitap bilgisayarların ve algoritmaların matematiksel modelleri hakkındadır. Bu ve sonraki iki bölümde giderek daha güçlü hesaplama modelleri, dilleri kabul etmek ve üretmek için giderek daha sofistike cihazlar tanımlayacağız. Kitabın tüm gelişimi ışığında bakıldığında, bu bölüm oldukça mütevazı bir başlangıç gibi görünecektir: Burada sonlu otomat ya da sonlu durum makinesi olarak adlandırılan gerçek bir bilgisayarın oldukça kısıtlı bir modelini ele alıyoruz.

Sonlu otomat, sabit ve sonlu kapasiteye sahip bir “merkezi işlem birimine” sahip olması gerçeğini gerçek bir bilgisayarla paylaşır. Girdisini, kendisine bir girdi bandı üzerinde teslim edilen bir dizi olarak alır. Girdinin kabul edilebilir olup olmadığına dair bir gösterge dışında hiçbir çıktı vermez. Diğer bir deyişle, Bölüm 1'in sonunda açıklandığı gibi bir dil tanıma cihazıdır. Sonlu otomatı gerçek bilgisayarların böylesine kısıtlı bir modeli yapan şey, sabit merkezi işlemcisinin dışında belleğin tamamen yokluğudur.

Bu kadar basit bir hesaplama modeli ilk başta ciddi bir çalışmayı hak etmeyecek kadar önemsiz görülebilir: hafızası olmayan bir bilgisayar ne işe yarar? Ancak sonlu bir otomat gerçekte hafızasız değildir; sadece “fabrikada” sabitlenen ve daha sonra genişletilemeyen bir hafıza kapasitesine sahiptir. Herhangi bir bilgisayarın bellek kapasitesinin bütçe kısıtlamaları, fiziksel sınırlar ve nihayetinde evrenin büyüklüğü ile sınırlı olduğu iddia edilebilir. Bir bilgisayar için en iyi matematiksel modelin sonlu belleğe sahip olan mı yoksa sınırsız belleğe sahip olan mı olduğu ilginç bir felsefi sorudur; sonlu modelden başlayarak ve daha sonra sınırsız model üzerinde daha fazla durarak her iki tür modeli de inceleyeceğiz.

Bizim yaptığımız gibi, bilgisayarları ve algoritmaları modellemenin doğru yolunun sınırsız bellek kapasitesi olduğunu düşünsek bile, öncelikle sonlu otomatlar teorisinin iyi anlaşıldığından emin olmalıyız. Teorilerinin zengin ve zarif olduğu ortaya çıkmıştır ve bunu anladığımızda, yardımcı bellek ilavesinin ilave hesaplama gücü açısından tam olarak neyi başardığını takdir etmek için daha iyi bir konumda olacağız.

Sonlu otomatların incelenmesinin bir diğer nedeni de, bunların yaygın olarak kullanılan çeşitli bilgisayar algoritmaları ve programlarının tasarımına uygulanabilirliğidir. Örneğin, bir derleyicinin sözlüksel analiz aşaması ('begin' ve '+' gibi program birimlerinin tanımlandığı) genellikle bir sonlu otomatın simülasyonuna dayanır. Ayrıca, bir dizginin başka bir dizginin içinde geçip geçmediğini bulma problemi de - örneğin, Elements of the Theory of Computationt metninde hava, su, toprak ve ateş dizgilerinden herhangi birinin geçip geçmediği - sonlu otomatlar teorisinden kaynaklanan yöntemlerle verimli bir şekilde çözülebilir.



Şimdi bir sonlu otomatın işleyişini daha ayrıntılı olarak açıklayalım. Dizeler cihaza, karelere bölünmüş ve her kareye bir sembol yazılmış bir giriş bandı aracılığıyla beslenir (bkz. şekil 2-1). Makinenin ana kısmı, herhangi bir anda sonlu sayıda farklı iç durumdan birinde olabilen iç kısımlara sahip bir “kara kutu” dur. Sonlu kontrol olarak adlandırılan bu kara kutu, hareketli bir okuma kafası vasıtasıyla giriş bandı üzerindeki herhangi bir konumda hangi sembolün yazıldığını algılayabilir. Başlangıçta, okuma kafası bandın en sol karesine yerleştirilir ve sonlu kontrol belirlenmiş bir başlangıç durumuna ayarlanır. Otomat düzenli aralıklarla giriş bandından bir sembol okur ve ardından yalnızca mevcut duruma ve yeni okunan sembole bağlı olan yeni bir duruma girer -bu nedenle bu cihazı bir sonraki bölümde tanıtılan deterministik olmayan versiyonla karşılaştırmak için deterministik sonlu otomat olarak adlandıracağız.

Bir giriş sembolünü okuduktan sonra, okuma kafası giriş bandında bir kare sağa hareket eder, böylece bir sonraki hareketinde bir sonraki bant karesindeki sembolü okuyacaktır. Bu süreç tekrar tekrar tekrarlanır; bir sembol okunur, okuma kafası sağa hareket eder ve sonlu kontrolün durumu değişir. Sonunda okuma kafası giriş dizesinin sonuna ulaşır. Otomat daha sonra okuduğu şeyi onayladığını ya da onaylamadığını sonunda bulunduğu durumla belirtir: eğer bir dizi son durumdan birine ulaşırsa girdi dizesi kabul edilmiş sayılır. Makine tarafından kabul edilen dil, kabul ettiği dizgiler kümesidir.

Bu gayri resmî açıklama matematiksel temellerine indirgendiğinde, aşağıdaki resmi tanım ortaya çıkmaktadır.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

M otomatının bir sonraki durumunu seçtiği kurallar geçiş fonksiyonuna kodlanır. Böylece M, q ∈ K durumundaysa ve giriş bandından okunan sembol a ∈ ∑ ise, δ(q, a) ∈ K, K'nın geçtiği benzersiz olarak belirlenmiş durumdur.

Deterministik bir sonlu otomatın temel yapısını biçimlendirdikten sonra, bir otomat tarafından bir girdi dizisi üzerinde yapılan bir hesaplama kavramını matematiksel terimler haline getirmeliyiz. Bu, kabaca, birbirini izleyen anlarda makinenin (sonlu kontrol, okuma kafası ve girdi bandı) durumunu temsil eden bir dizi konfigürasyon olacaktır. Ancak deterministik bir sonlu otomatın okuma kafasını girdi dizisinin daha önce okunmuş olan kısmına geri götürmesine izin verilmediğinden, dizinin okuma kafasının solundaki kısmı makinenin gelecekteki çalışmasını etkileyemez. Böylece bir konfigürasyon mevcut durum ve işlenmekte olan dizenin okunmamış kısmı tarafından belirlenir. Başka bir deyişle, deterministik bir sonlu otomatın (K, ∑, δ, s, F) konfigürasyonu K x ∑\*'nin herhangi bir elemanıdır. Örneğin, Şekil 2-1'de gösterilen konfigürasyon (92, ababab) şeklindedir.

'I--M' ikili bağıntısı M'nin iki konfigürasyonu arasında ancak ve ancak makine tek bir hamle sonucunda birinden diğerine geçebiliyorsa geçerlidir. Böylece, eğer (q,w) ve (q',w') M'nin iki konfigürasyonu ise, o zaman (q,w) I--M (q',w') ancak ve ancak bazı a ∈ ∑ sembolleri için w= aw' ise ve δ(q,a) = q' ise. Bu durumda (q, w)’nin bir adımda (q',w')'yi verdiğini söyleriz. Aslında I--M'nin K x ∑+'dan K x ∑\*'e bir fonksiyon olduğuna, yani (q, e) formu dışındaki her konfigürasyon için benzersiz olarak belirlenmiş bir sonraki konfigürasyon olduğuna dikkat edin. (q, e) biçimindeki bir konfigürasyon, M'nin tüm girdisini tükettiğini ve dolayısıyla işleminin bu noktada sona erdiğini gösterir.

I--M'nin refeksif, geçişli kapanışını I--M\* ile gösteriyoruz; (q,w) I--\*M (q', w') okunduğunda, (q,w) (belli sayıda, muhtemelen sıfır, adımdan sonra) (q', w') verir. Bir w ∈ ∑\* dizgisinin M tarafından kabul edildiği söylenir, ancak ve ancak (s,w) I--\*M (q,e) olacak şekilde bir q ∈ F durumu varsa. Son olarak, M tarafından kabul edilen dil, L(M), M tarafından kabul edilen tüm dizgilerin kümesidir.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

O halde L(M)'nin {a, b}\*'de çift sayıda b'ye sahip tüm dizgilerin kümesi olduğunu görmek kolaydır. Çünkü M, bir b okunduğunda q0 durumundan q1 durumuna veya q1 durumundan q0 durumuna geçer, ancak M esasen a'ları görmezden gelir, bir a okunduğunda her zaman mevcut durumunda kalır. Böylece M b'leri modül 2 sayar ve q0 (başlangıç durumu) aynı zamanda tek son durum olduğundan, M ancak ve ancak b'lerin sayısı çift ise bir dizeyi kabul eder.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

taslak, çizim, beyaz, çizgi sanatı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Bu örnekte kullanılan geçiş fonksiyonunun tablosal gösterimi bir makinenin en açık tanımı değildir. Genellikle durum diyagramı adı verilen daha uygun bir grafiksel gösterim kullanırız (Şekil 2-2). Durum diyagramı, resme bazı ek bilgilerin dahil edildiği yönlendirilmiş bir grafiktir. Durumlar düğümlerle temsil edilir ve δ(q,a) = q' olduğunda q düğümünden q' düğümüne a ile etiketlenmiş bir ok vardır. Son durumlar çift daire ile gösterilir ve başlangıç durumu a > ile gösterilir. Şekil 2-2, Örnek 2.1.1'deki deterministik sonlu otomatın durum diyagramını göstermektedir.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Durum diyagramı Şekil 2-3'te gösterilmektedir. M'nin belirtilen dili gerçekten kabul ettiğini görmek için, ardışık üç b okunmadığı sürece, M'nin giriş dizesini başlatan veya öncesinde bir a bulunan ardışık i b'yi okuduktan hemen sonra qi (burada i 0, 1 veya 2'dir) durumunda olduğuna dikkat edin. Özellikle, bir a okunduğunda ve M, q0, q1 veya q2 durumundayken, M başlangıç durumu olan q0'a döner. q0, q1 ve q2 durumlarının tümü nihai durumlardır, bu nedenle ardışık üç b içermeyen herhangi bir girdi dizesi kabul edilecektir. Bununla birlikte, üç b'den oluşan bir dizi M'yi nihai olmayan q3 durumuna götürecektir ve M daha sonra girdi dizesinin geri kalanındaki sembollerden bağımsız olarak bu durumda kalacaktır. q3 durumunun ölü bir durum olduğu söylenir ve M, q3 durumuna ulaşırsa tuzağa düştüğü söylenir, çünkü başka hiçbir girdi bu durumdan kaçmasını sağlayamaz.

çizgi, taslak, çizim, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

--ÖDEV: 2.1.2 – 2.1.3 – 2.1.7