PUSHDOWN OTOMAT

PDA → FA + yığın (stack)

Δ → banttaki boş hücreleri gösterir.

Giriş Bandı → FA'nın girdi stringinin çalıştırılırken bulunduğu kısım

→ Giriş bandı olası herhangi bir girdi için yeterince uzundur.

→ Bantta ilk konum ilk harf içindir, ikinci konum ikinci harf.... Bant sadece bir yönde sonsuzdur.

Hücre → Girdi konumlarını kaydığımız konumlar

Δ → Banttaki boş hücreler

Ⓜ Bu bandı makinede ~~oku~~ işlerken her seferinde bir harf okuruz ve kullandıkça her harfi ortadan kaldırırız.

Ⓜ İlk boş hücreye ulaştığımızda dururuz. (İlk boşluğa rastlandığında bandın geri kalanının da boş old. varsayarız.)

Ⓜ Soldan sağa okuruz ve daha önce dokunmuş bir hücreye asla geri dönmeyiz.

Çizim kuralları — Statelere giren veya çıkan oklar herhangi bir harfle çizilebilir

START state → bir TG'deki başka bir state'le λ kenarıyla bağlanır → 'a benzer.

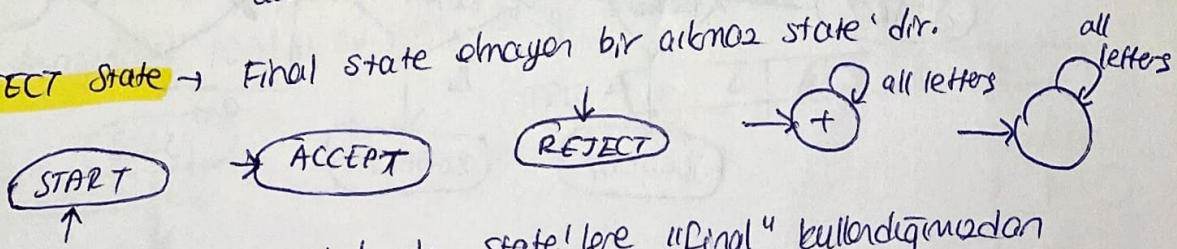
→ İşleme bundan başlarız hiçbir girdi harfi okunmaz

→ Sadece hemen sonraki state'e geçeriz.

→ Bir start state'in kendisine gelen hiçbir ok yoktur.

ACCEPT State → Bir çıkmaz final state'dir Bir kez girildikten sonra çıkılmaz.

REJECT State → Final state olmayan bir çıkmaz state'dir.



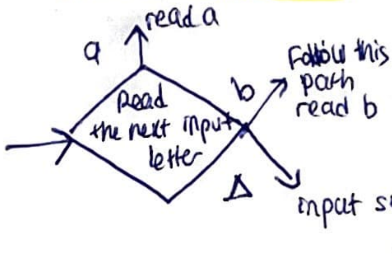
⇒ FA'larda yalnızca kabul eden state'lere "final" kullandığımızdan Accept, Reject state'lere "halt state" leri (duma statelei) adını veriyoruz

⇒ Önceden girdi stringini ~~okuyup~~ okumayı bitirmediysek bir final state'den geçebiliyorduk. Halt state'ler geçilemez.

⇒ Bir state'in gerçekleştirdiği ~~işlem~~ her işlem görseldeki ayrı bir kutu tıkan yapılan

⇒ Bir FA'daki state tıkan yapılan en önemli iş, bir girdi harfini okumak ve okunan harfe bağlı olarak diğer state'lere dallandırmaktır.

Okuma stateti Elmas şeklinde kutular



Kenarlar birçok olasılıktan yalnızca birini girir.

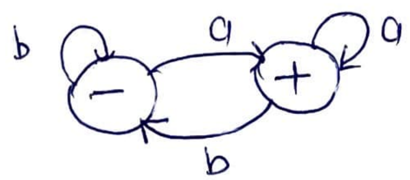
Banttan Δ okunduğunda, girdi harflerinin tükendiği anlamına gelir.

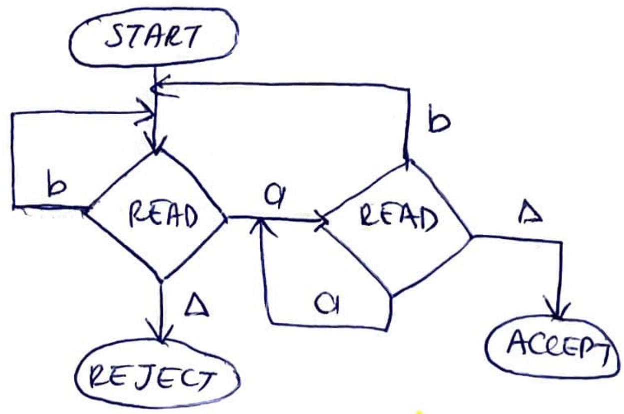
Δ kenarı → Durduğumuz state bir final state ise accept state'e gider, bir final state değilse reject state'e gider.

→ Bu öneriler makinenin gücünü değiştirmemiştir.

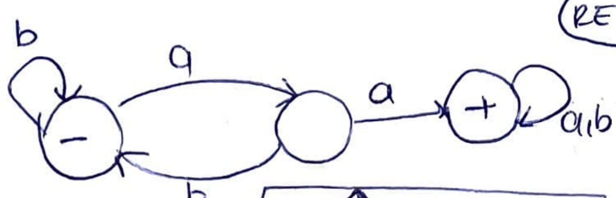
ÖRN a harfi ile biten tüm kelimeleri kabul eden FA bu yeni makineye dönüşür.

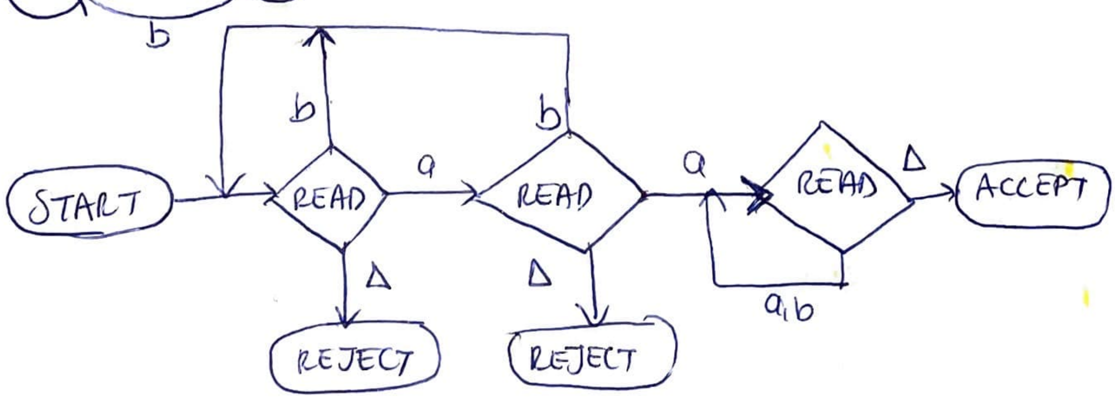
! Start'tan çıkan kenarın etikete ihtiyacı olmadığını unutmayın, çünkü start hiç harf okumaz.





ÖRN





Adding Pushdown Stack

Push işlemi → Stack'in en üstüne yeni bir harf ekler ve diğer tüm harfler geri (veya aşağı) itilir.

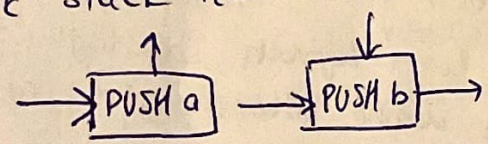
→ Makine girdi stringini işlemeye başlamadan önce stack'in boş old. varsayılır.

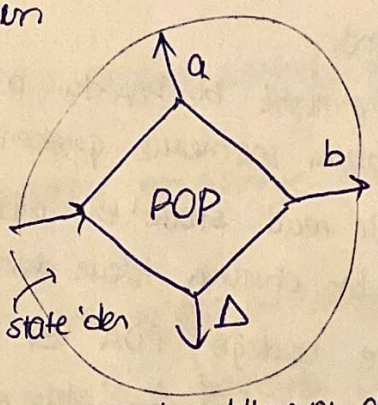
Pop → Stackten bir harf alınma talimatı
→ Stacktekı diğer harfler bir pozisyon yukarı taşınır.

Bir Push Down Stack → LIFO (Last in First Out) dosyası

⇒ Bir Push Down stack yalnızca en üstteki harfi okuma olanağı verir.
⇒ Yani üçüncü harfi okumak için pop, pop, pop demeliyiz.
→ Ayrıca en alttaki harfi belirlemek, stack'te kaç tane b harfi old. söylemek vb. için talimatımız da yoktur

Tek stack işlemleri → push ve pop





KURALLAR

Bir pop stateinden çıkan kenarlar, bir read state'den gelen kenarlarla aynı şekilde etiketlenir

Pop statelerden dallanmalar olabilir ancak push statelerden dallanma olmaz!

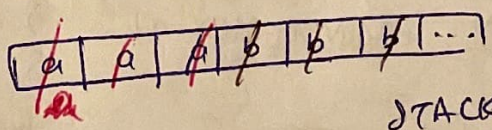
Push statelere yalnızca belirtilen nota ile terk edebiliriz.

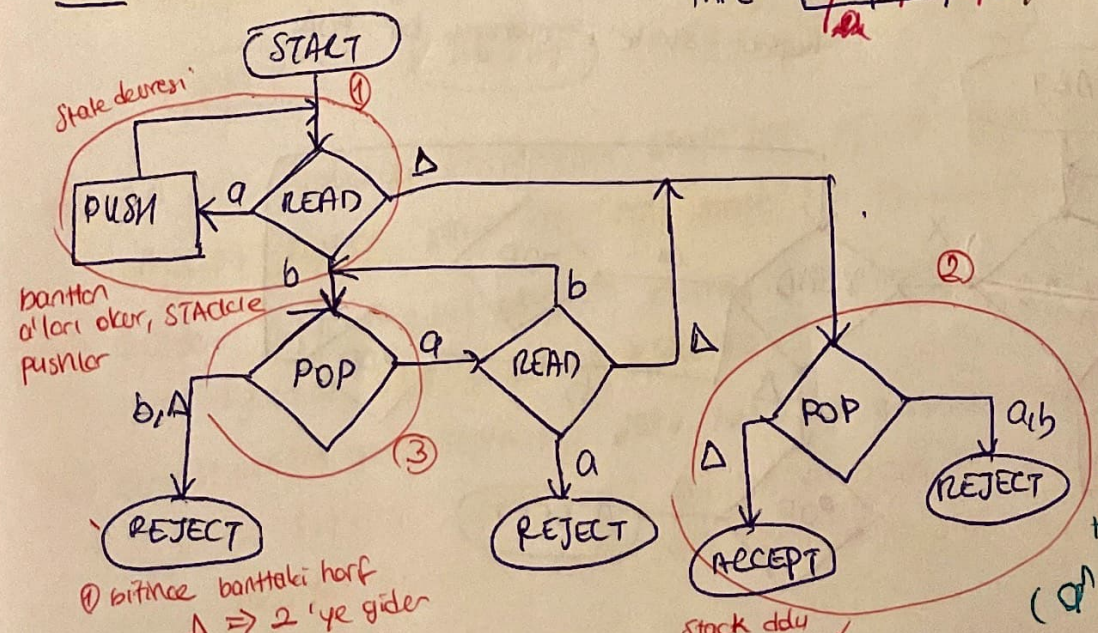
Bir push state'e her hangi bir yerden gidebiliriz.

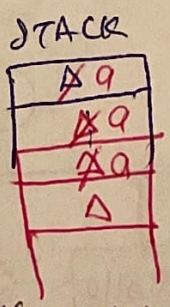
FA'lara bir stack, pop ve push statelerı ekleyince buna PDA (pushdown otomata) denir.

ÖRN Aşağıdaki PDA'yı şu girdi stringi üzerinde düşünün: aaabbb

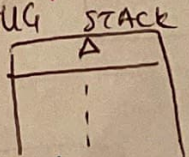
TAPE







SONUÇ STACK



aaabbb bu makine tarafından kabul edilir

$(a^n b^n \; n=0,1,2\ldots)$

PDA

Bu makineleri FA'dan daha güçlü yapan nedir?

FA $a^n b^n$ dilini kabul edemez çünkü a^n parçası bir devre üzerinde gezmiştir ve FA devre etrafında kaç kez döndüğünü takip edemez.

Ancak PDA'nın ilkel bir bellek birimi vardır, başlangıçtan kaç tane a'nın okunduğunu takip edebilir.

~~[illegible]~~

PDA'lar eğer CFG'lere karşılık gelecekse aynı zamanda non deterministik olmak zorunda kalacaklar.

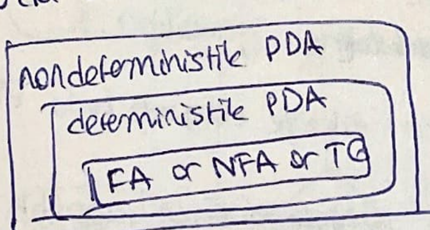
Deterministik bir PDA'da her girdi string'inin makine üzerinde eşsiz bir yolu vardır.

Nondeterministik bir PDA'da bazı noktalarda birkaç farklı seçenek arasından birini bizim seçmemiz gereken yollar içerir.

=> Belirli bir read state'ten çıkan bir b kenarı koymama özgürlüğümüz olacak. Bu kenardan bir okunursa işleme devam edilemez makine çöker ve girdi reddedilir.

CFG'lere eşdeğer PDA => Nondeterministik PDA

FA'lar için nondeterminizmin makinenin gücünü değiştirmediğini bulmuştuk, PDA'larda bu durum farklı:



ÖRN PALINDROMEX dili s X reverse(s)
s, (a+b)* içindeki herhangi bir string
Reject state'i olmayan bir PDA