

10- ÖZDEVİNİRLER KURAMI

AFYON KOCATEPE ÜNİVERSİTESİ - MÜHENDİSLİK FAKÜLTESİ – BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BM103 – BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DR. ÖĞR. ÜYESİ ÖZKAN ASLAN

ÖZDEVİNİRLER KURAMI

- Özdevinirler (otomata) kuramı, soyut makineler ve otomatlar kullanarak çözülebilecek hesaplama problemlerinin incelenmesidir.
- Kuramsal bilgisayar bilimlerinde bir çalışma alanıdır.
- Otomata (otomat kelimesinin çoğulu) kelimesi, "kendi kendine hareket eden, kendi iradesiyle" anlamındaki Yunanca αὐτόματος kelimesinden gelir.

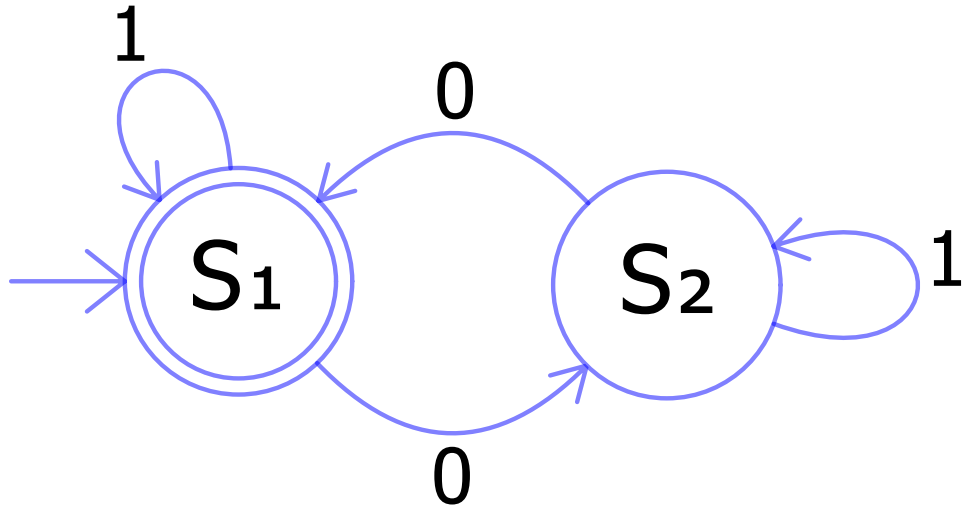
ÖZDEVİNİRLER KURAMI

- Bir otomat; önceden belirlenmiş bir işlem dizisini otomatik olarak izleyen, soyut, kendinden hareketli bir bilgi işlem cihazıdır.
- Sonlu sayıda durumu olan bir otomat, Sonlu Otomat (Finite Automata) veya Sonlu Durum Makinesi (Finite State Machine) olarak adlandırılır.

SONLU DURUM MAKİNESİ

- Bu otomat, durumlardan (şekilde dairelerle temsil edilir) ve geçişlerden (oklarla temsil edilir) oluşur.
- Bir girdi sembolü gördüğünde, önceki durumu ve mevcut girdi sembolünü argümanları olarak alan geçiş fonksiyonuna göre başka bir duruma geçiş yapar (veya atlar).

SONLU DURUM MAKİNESİ



	S1	S2
S1	1	0
S2	0	1

TURING MAKİNESİ

- Turing makinesi, bir girdi-çıkı şeridindeki sembolleri bir kurallar tablosuna göre işleyen soyut bir makineyi tanımlayan matematiksel bir hesaplama modelidir.
- Modelin basitliğine rağmen, herhangi bir bilgisayar algoritması verildiğinde, o algoritmanın mantığını simüle edebilen bir Turing makinesi oluşturulabilir.

TURING MAKİNESİ

- Makine, ayrık "hücrelere" bölünmüş sonsuz bir bellek bandı üzerinde çalışır.
- Okuma-yazma kafası bir hücrenin üzerine konumlandırılır ve oradaki sembolü “okur” veya “tarar”.
- Daha sonra, sembole ve kullanıcı tanımlı talimatların “sonlu tablosunda” makinenin kendi mevcut durumuna bağlı olarak, makine önce hücreye bir sembol yazar.
- Ardından ya kafayı bir hücre sola ya da sağa hareket ettirir.
- Sonra, gözlemlenen sembole ve makinenin tablodaki kendi durumuna göre, ya başka bir komuta geçer ya da hesaplamayı durdurur.

BİÇİMSEL DİLLER

- Mantık, matematik, bilgisayar bilimi ve dilbilimde bir biçimsel dil (formal language), harfleri bir alfabeden alınan ve belirli bir dizi kurala göre iyi biçimlendirilmiş kelimelerden oluşur.
- Bir biçimsel dilin alfabesi, birleşen semboller, harfler veya simgelerden (token) oluşur.
- Bu alfabenin sembolleriyle oluşturulan her bir diziye kelime denir.
- Bir biçimsel dil, genellikle, düzenli bir dilbilgisi aracılığıyla tanımlanır.

BİÇİMSEL DİLLER

- Biçimsel dil kuramı alanı, öncelikle bu tür dillerin saf sözdizimsel yönlerini, yani iç yapısal kalıplarını inceler.
- Biçimsel dil kuramı, doğal dillerin sözdizimsel düzenliliklerini anlamamanın bir yolu olarak dilbilimden doğmuştur.
- Bilgisayar biliminde, biçimsel diller, programlama dillerinin dilbilgisini ve dilin sözcüklerinin belirli anlamlar veya anlambilim ile ilişkili kavramları temsil ettiği doğal dillerin alt kümelerinin biçimselleştirilmiş versiyonlarını tanımlamak için temel olarak kullanılır.
- Hesaplamalı karmaşıklık teorisinde, karar problemleri tipik olarak biçimsel diller olarak tanımlanır ve karmaşıklık sınıfları, sınırlı hesaplama gücüne sahip makineler tarafından çözümlenebilen biçimsel dil kümeleri olarak tanımlanır.
- Mantıkta ve matematiğin temellerinde, biçimsel diller aksiyomatik sistemlerin sözdizimini temsil etmek için kullanılır.

BAĞLAM DAN BAĞIMSIZ DİLLER

- Biçimsel dil kuramında, bağlamdan bağımsız bir dil (context-free language), bağlamdan bağımsız bir dilbilgisi (context-free grammar) (CFG) tarafından oluşturulan bir dildir.
- Bağlamdan bağımsız dillerin programlama dillerinde birçok uygulaması vardır, özellikle çoğu aritmetik ifade bağlamdan bağımsız dilbilgisi tarafından oluşturulur.

BAĞLAM DAN BAĞIMSIZ DİLBİLGİSİ

- Bilgisayar bilimlerinde, dil tasarımı sırasında kullanılan bir dilbilgisi tipidir. Basitçe bir dilin kurallarını (gramer) tanımlamak için kullanılır.
- Örnek:
$$S \rightarrow aSb \mid ab$$

BACKUS-NAUR FORM

- Bilgisayar biliminde, Backus–Naur form veya Backus normal form (BNF), genellikle bilgisayar programlama dilleri, belge gibi bilgi işlemde kullanılan dillerin sözdizimini tanımlayan bağlamdan bağımsız gramerler için bir gösterimdir.
- Dillerin tam tanımlarının gerekli olduğu her yerde uygulanırlar: Örneğin, biçimsel dil belirtimlerinde, kılavuzlarda ve programlama dili kuramına ilişkin ders kitaplarında.

BACKUS-NAUR FORM

$\langle \text{tümce} \rangle ::= \langle \text{ad öbeği} \rangle \langle \text{eylem öbeği} \rangle$

$\langle \text{ad öbeği} \rangle ::= \langle \text{ad} \rangle \mid \langle \text{sıfat} \rangle \langle \text{ad} \rangle \mid \langle \text{ad} \rangle \langle \text{ad} \rangle$

$\langle \text{eylem öbeği} \rangle ::= \langle \text{eylem} \rangle \mid \langle \text{belirteç} \rangle \langle \text{eylem öbeği} \rangle \mid \langle \text{nesne} \rangle \langle \text{eylem öbeği} \rangle$

$\langle \text{ad} \rangle ::= \text{“çocuk”} \mid \text{“kedi”} \mid \text{“anne”} \mid \text{“baba”} \mid \text{“öğretmen”} \mid \text{“okul”}$

$\langle \text{sıfat} \rangle ::= \text{“küçük”} \mid \text{“sevimli”} \mid \text{“eski”} \mid \text{“yaşlı”} \mid \text{“bir”}$

$\langle \text{eylem} \rangle ::= \text{“gördü”} \mid \text{“okudu”} \mid \text{“sevdi”} \mid \text{“çağırdı”} \mid \text{“tanıdı”}$

$\langle \text{belirteç} \rangle ::= \text{“yavaşça”} \mid \text{“aniden”} \mid \text{“biraz”} \mid \text{“nasıl”} \mid \text{“nazikçe”}$

$\langle \text{nesne} \rangle ::= \text{“çocuğu”} \mid \text{“kediği”} \mid \text{“anneyi”} \mid \text{“babayı”} \mid \text{“öğretmeni”} \mid \text{“okulu”} \mid \text{“kitabı”}$

KAYNAKLAR

1. https://en.wikipedia.org/wiki/Formal_language