FORMEL DİLLER VE SOYUT MAKİNALAR

Hafta 1

DOĞAL DİL NEDİR?

- Sözcük ve cümle birimleri aracılığıyla, düşünceyi konuşmayla ilişkilendiren çok seviyeli bir sistemdir *(Noam Chomsky)*.

- Dil, düşünce aktarma ve iletişim için çeşitli modüllerin kullanıldığı geleneksel sembollerden oluşan karmaşık ve dinamik bir sistemdir *(Amerikan Konuşma-Dil-İşitme Derneği - ASHA)*.

- Dil, insanların **karmaşık iletişim sistemlerini edinme** ve kullanma becerisidir.

FORMEL DİL NEDİR?

- Bir dilin formel olabilmesi için bazı niteliklerinin matematiksel kesinlikte tanımlı olması gereklidir. Bu özellikler;
  - Sembollerden oluşan **bir alfabe** ve
  - Bu alfabedeki sembollerden oluşan bir ifadenin dile uygun (well-formed) olup olmadığını belirleyen **oluşum (formation) kurallarıdır**.

- Bu şekildeki ilk formel dil Gottlob Frege tarafından 1879 yılında tanımlanmış olup **birinci dereceden mantık (first-order logic)** olarak adlandırılmaktadır.

- Formel diller özellikle **dilbilim** ve **bilgisayar bilimleri** açısından önem taşımaktadır.

DİLBİLİM VE FORMEL DİLLER

- **Dilbilimde formel diller**, insan dilinin (yani doğal dilin) bilimsel bir şekilde incelenmesi amacıyla kullanılmaktadır.

- Dilbilimciler, üretimsel (generative) bir yaklaşıma önem verirler. Bunun nedeni, bir dile göre kabul edilebilir herhangi bir cümlenin oluşturulmasında kullanılabilecek (sonlu) kurallar kümesi olan dil düzeneğini / grameri (grammar) tanımlama / ortaya koyma işiyle ilgilenmeleridir.

- Bir gramer cümlelerin yalnızca biçimleriyle (form) ilgilenmekte olup, anlamlarını tanımlamamakta, farklı bağlamlarda kullanımlarıyla da ilgilenmemektedir.

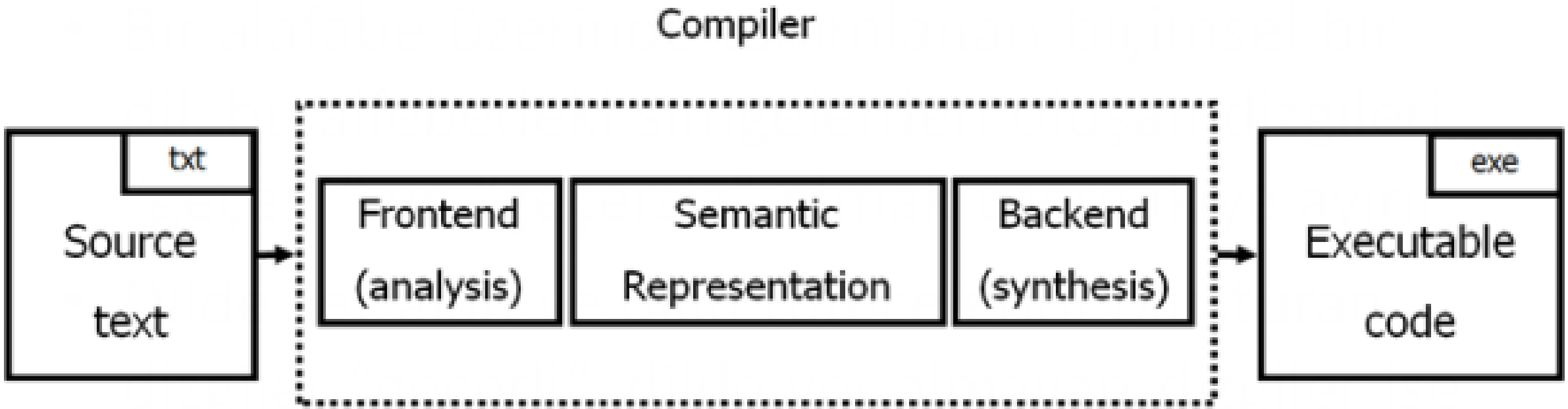
BİLGİSAYAR BİLİMLERİ VE FORMEL DİLLER

- **Bilgisayar bilimlerinde formel diller** programlama dillerinin kesin ve katı kurallarını tanımlamada kullanılmaktadır. Bu yönüyle derleyicilerin de temellerini oluşturmaktadırlar.
- **Bir derleyici**, bir programlama dilinde (kaynak dil) yazılmış olan bir kaynak kodun başka bir bilgisayar diline (hedef dil) dönüştürülmesi işini yerine getiren program ya da programlar kümesidir.
- Derleyicilerin en genel amacı, kaynak kodun çalıştırılabilir bir program haline getirilmesi için dönüştürülmesidir.
- Bilgisayar bilimlerinde önem verilen, soyut makinelere (otomatlar) dayanan tanıma (recognition) yaklaşımıdır. Bu yaklaşımda soyut makine bir girdi cümlesi almakta ve bu cümlenin referans alınan dile ait olup olmadığını belirlemektedir.

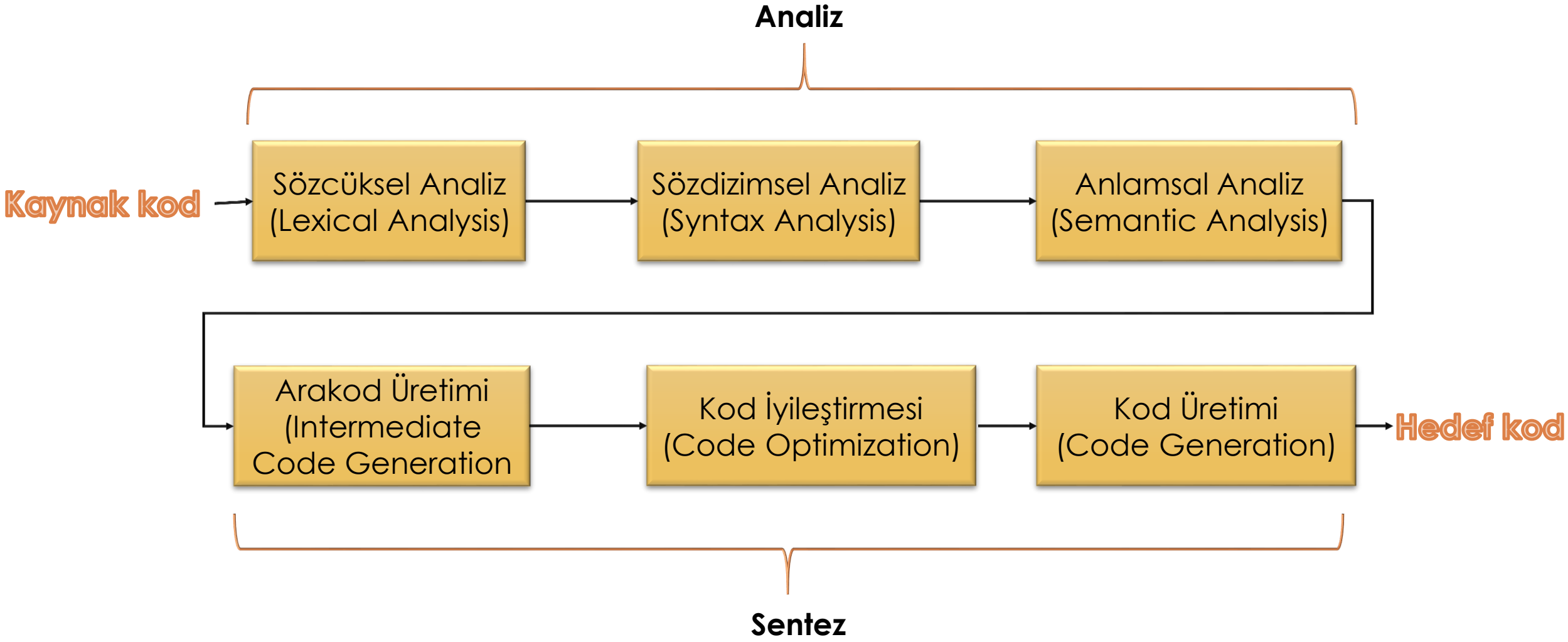
DERLEYİCİ

- Herhangi bir programlama dilinde yazılmış olan kaynak kodunu başka bir dile (genellikle makine koduna) çeviren yazılımdır.

DERLEYİCİ


Compiler
txt
Source text
Frontend (analysis)
Semantic Representation
Backend (synthesis)
exe
Executable code

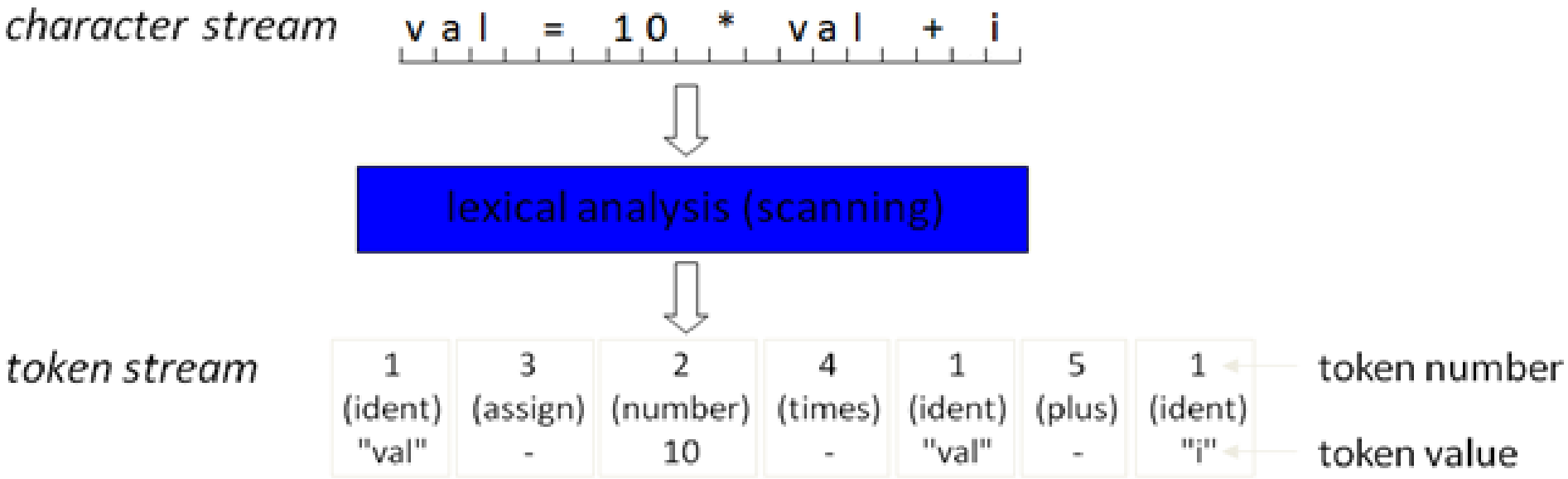
DERLEME AŞAMALARI



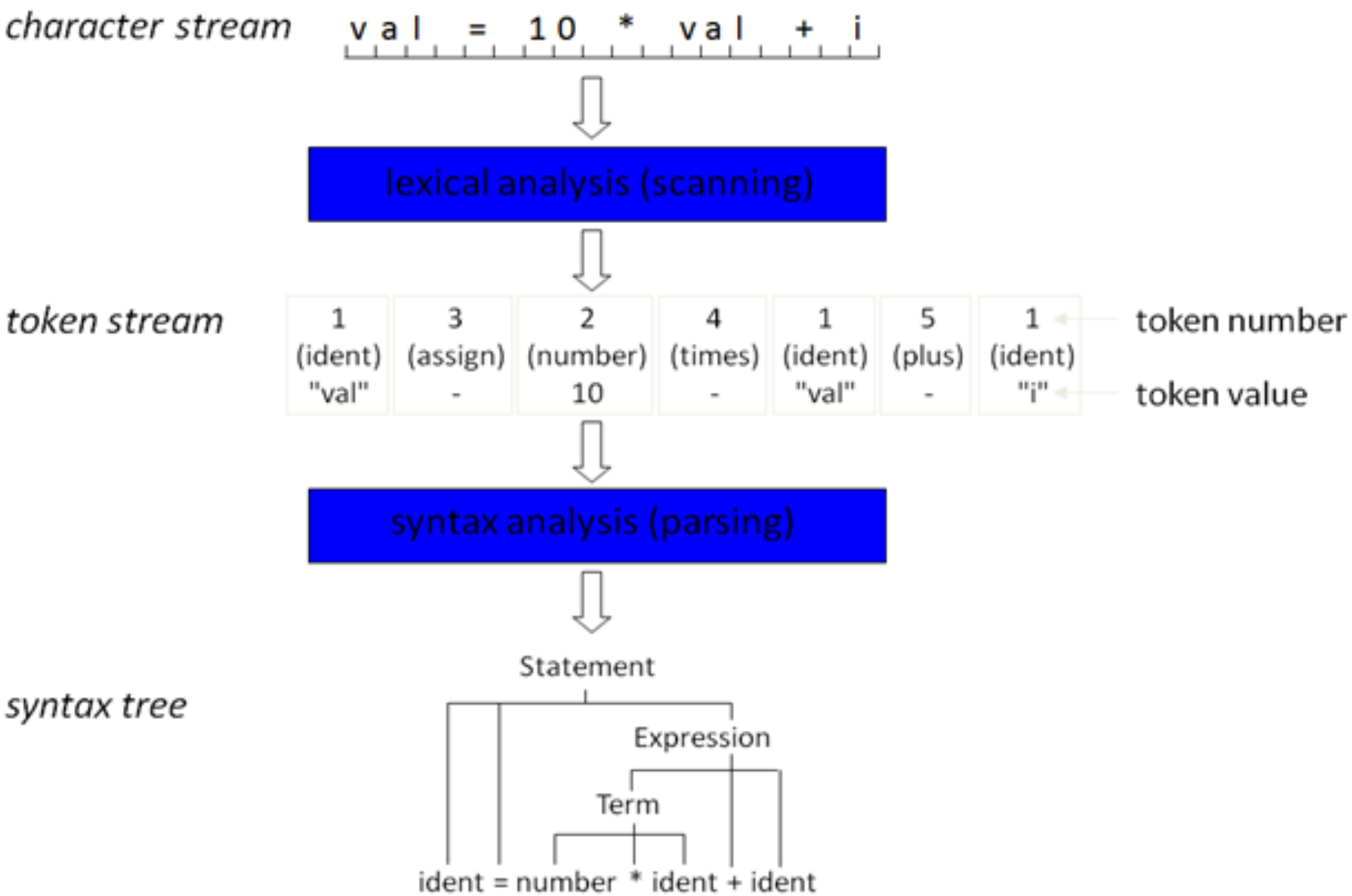
DERLEYİCİ AŞAMALARI
ÖRNEK

character stream 

DERLEYİCİ AŞAMALARI
ÖRNEK

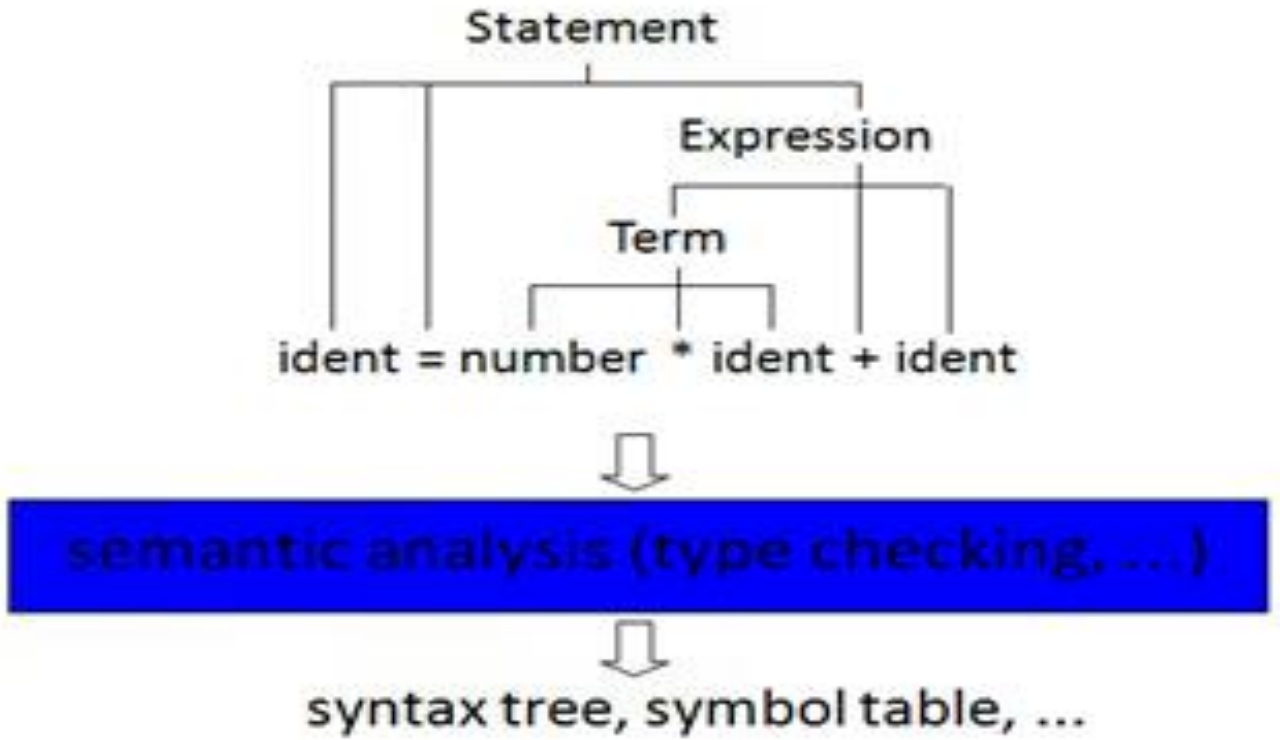


DERLEYİCİ AŞAMALARI
ÖRNEK



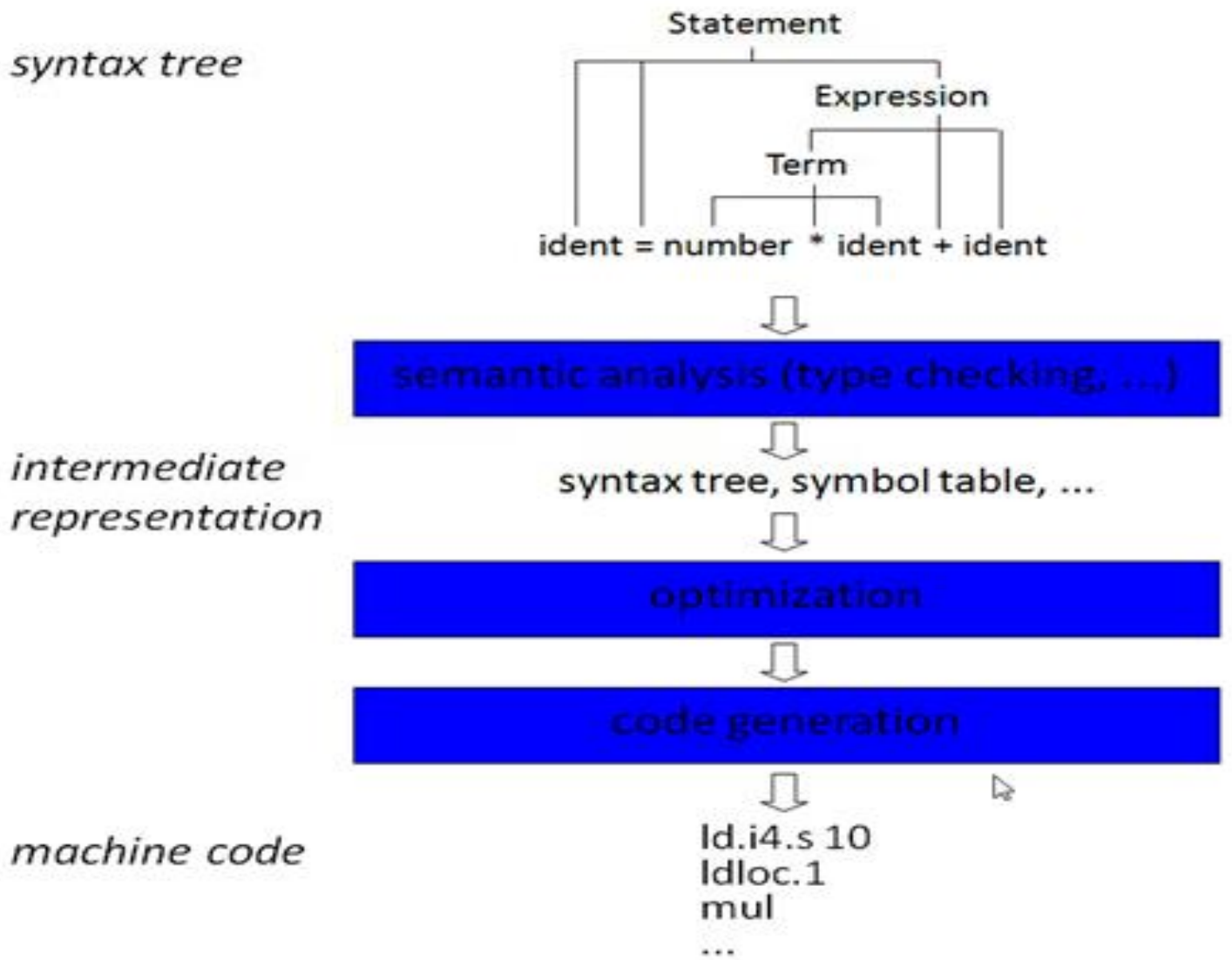
DERLEYİCİ AŞAMALARI
ÖRNEK

syntax tree



intermediate representation

DERLEYİCİ AŞAMALARI
ÖRNEK

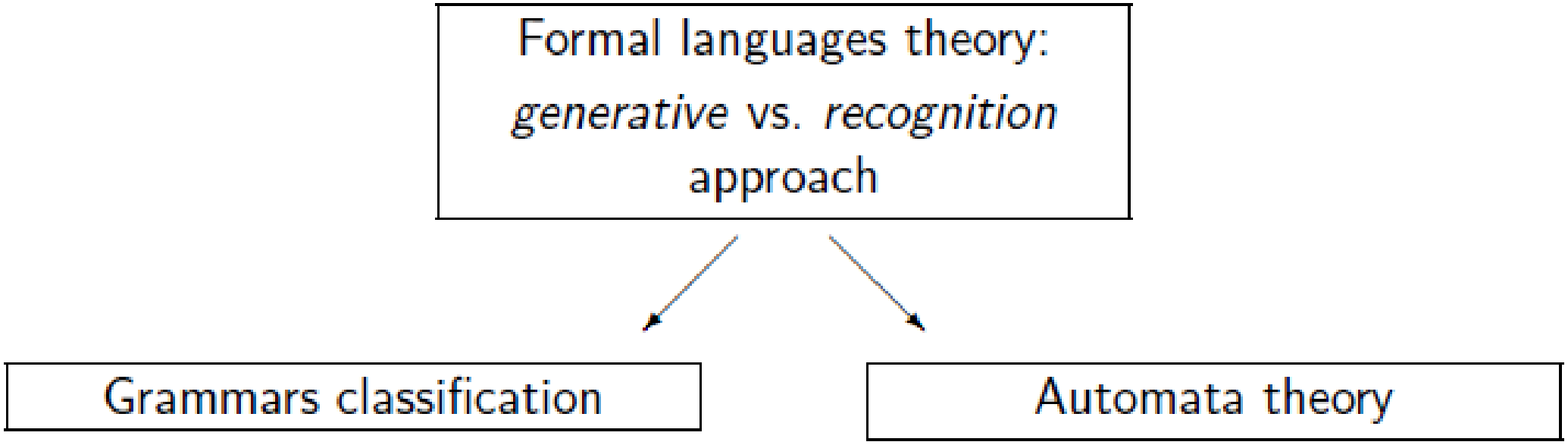


OTOMATLAR VE FORMEL DİLLER

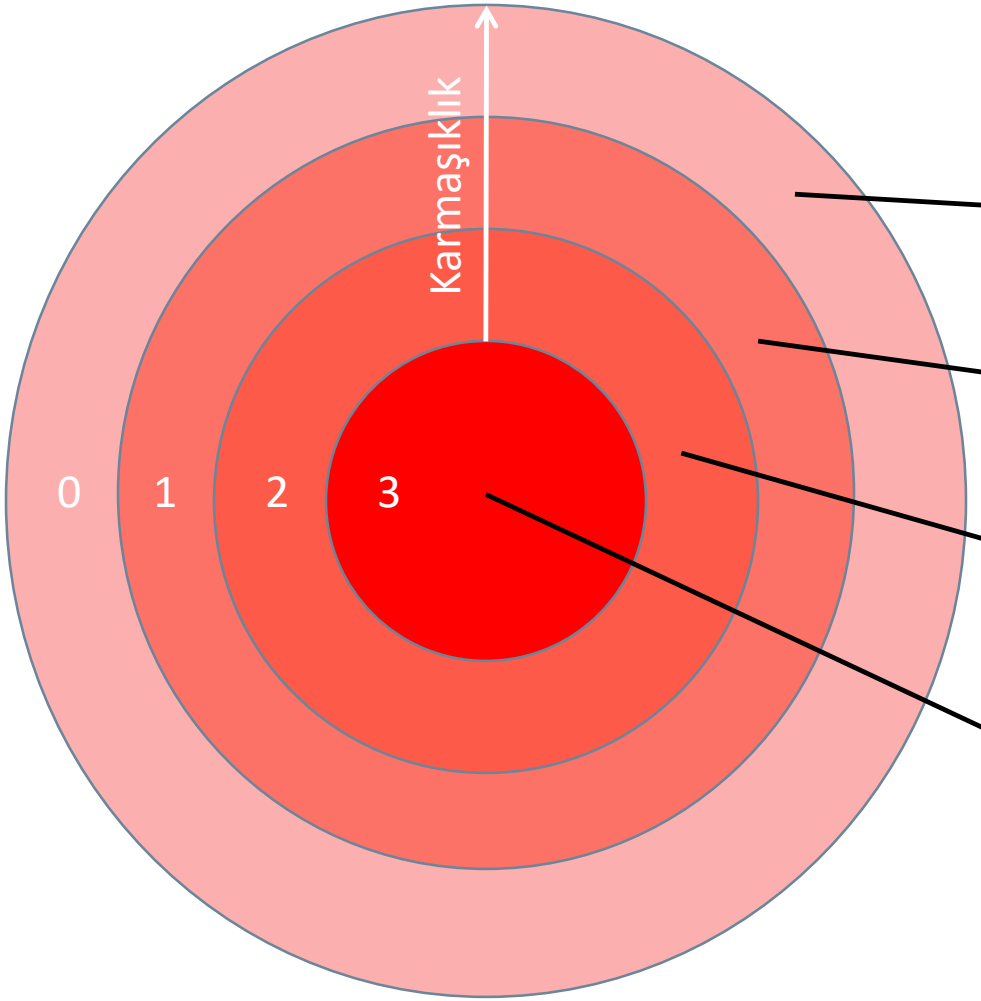
- **« Automata »** terimi Yunancadan gelmekte ve **« kendi kendine eylemde bulunabilen »** anlamına gelmektedir.
- Bir otomat, önceden belirlenmiş bir işlemler dizisini takip ederek kendiliğinden çalışabilen soyut bir bilgisayım cihazıdır.
- 1950'lerde Stephen Kleene, sonlu bir bellekle donanmış soyut durum makineleri olan **sonlu otomatları** ortaya atmıştır.
- Kleene, bu modelle temel mantıksal öncülleri kullanarak böyle bir modelin sembol dizileri ile eşleniklik gösterdiğini belirtmiştir.
- Alan Turing (ve ondan bağımsız olarak Emil Post ile John Backus), **bas-bırak otomatlarının (push-down automata)** ardında yatan düşünceleri ortaya koymuşlardır.
- Alan Turing, 1936'da **Turing Makinesi** kavramını ortaya atmıştır. Bu makine, bir soyut durum makinesi olup, şerit şeklinde sonsuz bir belleğe sahiptir. Turing makineleri, herhangi bir algoritmanın işleyişinin benzetimini yapabileceği gibi daha yüksek seviyelerdeki biçimsel dilleri de tanıyabilmektedir.

OTOMATLAR VE GRAMERLER

- Hangi formel dil sınıfı hangi otomat türüyle tanınabilmektedir?
  - Chomsky hiyerarşisi ile farklı tipteki otomatlar arasında bir eşleniklik vardır. Bu nedenle formel dillere ilişkin kuramlar hem gramerler hem de otomatlar olarak ele alınabilmektedir.



CHOMSKY HİYERARŞİSİ



Özyinelemeli - Sayılabilir Diller : Turing Makinesi
(Recursively – Enumerable Languages : Turing Machine)

Bağlama - Duyarlı Diller : Doğrusal – Sınırlandırılmış otomatalar
(Context – Sensitive Languages : Linear – Bounded Automata)

Bağlam - Bağımsız Diller : Bas – Bırak otomataları
(Context – Free Languages : Push – Down Automata)

Düzenli Diller : Sonlu - Durum otomataları
(Regular Language : Finite – State Automata)

***(Dil : otomata)**

FORMEL DİLLERİN BETİMLENMESİ: ÜRETİMSEL (GENERATIVE)YAKLAŞIM

- Üretimsel yaklaşımda bir dil, bir gramer tarafından üretilen karakter katarlarının (string) kümesidir.

- Bu yaklaşımdaki üretim süreci;

  - bir başlangıç sembolü ile **başlama**,

  - yeniden yazma (rewrite) kuralları ile **genişletme**,

  - dile ait bir ifade üretildiğinde de **durma** şeklinde gerçekleştirilmektedir.

FORMEL DİLLERİN BETİMLENMESİ: TANIMA (RECOGNITION)YAKLAŞIMI

- Bu yaklaşıma göre bir dil bir otomat tarafından kabul edilen ifadeler / katarlar kümesidir.

- Bu yaklaşımdaki tanıma süreci;

  - Bir başlangıç durumu ile **başlama**,

  - Katardaki semboller yardımıyla diğer durumlara **geçişler**,

  - Bütün katar tükendiğinde kabul durumuna **ulaşma** ya da katarın belirli bir konumunda **reddetme** şeklinde olmaktadır.

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- Bir **formel dil**, o dilin üzerinden tanımlanmış olduğu alfabedeki sembollerin kullanımıyla oluşturulan sonlu karakter dizilerinin /karakter katarlarının / ifadelerin kümesidir.
- **Alfabe**, sonlu ve boş olmayan bir semboller kümesidir.
  - $\sum_1 = \{0, 1\}$
  - $\sum_2 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  - $\sum_3 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
  - $\sum_4 = \{a, b, c, ..., z\}$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- Bir ∑ alfabesi üzerinde tanımlı olan bir **katar** (ya da **kelime**), ∑ alfabesindeki sembollerden oluşan sonlu bir dizidir.
  - **1010 ϵ $\sum_1$**
  - **123 ϵ $\sum_2$**
  - **merhaba ϵ $\sum_4$**
- Bir alfabedeki her sembol kendi başına bir katardır.

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- Alfabe olarak $\Sigma = \{a, b\}$ verilmiş olsun.
- Bu alfabe ile oluşturulabilecek karakter katarları (strings) şunlar olabilir:

a

ab

$abba$

$baba$

$aaabbbaabab$

$u = ab$

$v = bbbaaa$

$w = abba$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Karakter katarları üzerinde yapılabilecek işlemler:

- **Birleştirme (Concatenation) :** İki karakter katarının ardarda eklenmesidir.

$$w = a_1 a_2 \cdots a_n$$

$$v = b_1 b_2 \cdots b_m$$

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Birleştirme örnekleri:

$x = abbba$

$y = aaababbbab$

$xy = abbbaaaababbbab$

$x = trakya$

$y = \text{ü}niversitesi$

$xy = trakya\text{ü}niversitesi$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Reverse işlemi:** Bir karakter katarının tersi işlemi, katarın sondan başlanarak tekrar yazılmasıdır.

$$w = a_1 a_2 \cdots a_n$$

$$w^R = a_n \cdots a_2 a_1$$

$$x = trakya \quad \Rightarrow \quad x^R = aykart$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Karakter katarı uzunluğunu bulma:** **|w|**, w katarının **uzunluğudur**.

$$w = a_1 a_2 \cdots a_n \longrightarrow |w| = n$$

|a| = 1

|125| = 3

| ε | = 0

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Birleştirme sonucu oluşan katarın uzunluğunu bulma**

$$|uv| = |u| + |v|$$

$$u = aab, \quad |u| = 3$$

$$v = abaab, \quad |v| = 5$$

$$|uv| = |aababaaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Boş karakter katarı

- **ε** veya **λ** boş katardır ve sembol içermez.

$$|\lambda| = 0$$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Alt karakter katarı (Substring): Bir karakter katarı başka bir katarın içinde ise onun alt karakter katarıdır.

Karakter katarı (String)	Alt katar (substring)
$ababbba$	$abbb$
1000011	0011
$123abc4$	$3ab$
aba	ab

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Prefix ve suffix**

$$w = uv$$

Prefix	$w = abbab$	*Suffix*
λ		$abbab$
a		$bbab$
ab		bab
abb		ab
$abba$		b
$abbab$		λ

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Üs işlemi**

$$w^n = \underbrace{ww \ldots w}_{n}$$

$$w^0 = \lambda$$

$$(abba)^2 = abbaabba$$

$$(abba)^0 = \lambda$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Kleene Yıldızı (Kleene Star):** ∑ üzerinde tanımlı olası bütün katarlar kümesini gösterir.

$$\sum^* = \sum_0 \cup \sum_1 \cup \sum_2 \cup \ldots\ldots$$

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Kleene Artısı (Kleene Plus):** ∑ alfabesi için $\sum^+$ bu alfabeden oluşturulan boş katar hariç tüm katarların kümesini göstermektedir.

$$\sum^+ = \sum_1 \cup \sum_2 \cup \ldots\ldots$$

$$\sum^+ = \sum^* - \{ \lambda \}$$

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- Formel dilin matematiksel tanımı: belirli bir ∑ alfabesi için ∑*'nin herhangi bir alt kümesi bir **dildir**.

  - İngilizce, Türkçe, Çince,...
  - C, Pascal, Java, HTML,...
  - İkili tabandaki asal sayılar:
    - {10, 11, 101, 111, 1011, ... }
  - **{ε}**

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$$

- Tanımlanabilecek diller:

$$\{\lambda\}$$

$$\{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Sets $\varnothing = \{\} \neq \{\lambda\}$

Set size $|\{\}| = |\varnothing| = 0$

Set size $|\{\lambda\}| = 1$

String length $|\lambda| = 0$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Örnek: $L = \{a^n b^n : n \geq 0\}$ verilmiş olsun.

Bu dil ile tanımlanabilecek karakter katarları:

λ

ab

$aabb$

$aaaaabbbbb$

$\in L$

abb

a

$baaa$

$ababab$

$\notin L$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Diller üzerinde yapılabilecek işlemler:**

- **Birleşme, Kesişme:** Diller kümeler ile gösterilebildiği için bu işlemler yapılabilir. L_1 ve L_2 dillerinin sırasıyla $\sum_1$ ve $\sum_2$ alfabeleri üzerinden tanımlanmış diller olduğunu kabul edersek;

$$L_1 \cup L_2 = \{w | w \in L_1 \vee w \epsilon L_2\}$$

$$L_1 \cap L_2 = \{w | w \in L_1 \wedge w \epsilon L_2\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- $L_1 = \{a, ab, aaaa\}$

$L_2 = \{bb, ab\}$

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$
$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$
$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Tümleyen işlemi**

$$\overline{L} = \Sigma^* - L$$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Reverse işlemi:** $L^R = \{w^R : w \in L\}$

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

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

$$L^R = ?$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Birleştirme**

$$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

$$L_1 = \{a, ab, ba\}$$

$$L_2 = \{b, aa\}$$

$$\{a, ab, ba\}\{b, aa\} = \{ab, aaa, abb, abaa, bab, baaa\}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Üs işlemi:**

$$L^n = \underbrace{LL \ldots L}_{n}$$

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

$$L^0 = \{\lambda\}$$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

Özel durum

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

Örnek: $L = \{a^n b^n : n \geq 0\}$

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

$$aabbaaabbb \in L^2$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Kleene Yıldızı (Kleene Star):** L^* ile gösterilir. Bir dilde 0 veya daha fazla karakter katarının birleştirme işlemi sonucu oluşur.

$$L^* = L^0 \cup L^1 \cup L^2 \ldots$$

$$\{a, bb\}^* = \begin{Bmatrix} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{Bmatrix}$$

FORMEL DİLLER: TANIMLAR VE TEMEL KAVRAMLAR

- **Kleene Artısı (Kleene Plus):** L^+ ile gösterilir. L dilini ve L dilindeki karakter katarlarının eklenmesi ile elde edilen tüm karakter katarlarını içeren bir dildir.

$$L^+ = L^1 \cup L^2 \cup \cdots$$

$$= L^* - \{\lambda\}$$

$$\{a, bb\}^+ = \begin{Bmatrix} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{Bmatrix}$$

KAYNAKLAR

- Hopcroft, J.E. and Ullman J.D. (1979). Introduction to Automata Theory, Languages, and Computation (1st ed.). Addison-Wesley.
- Yarımağan, Ünal, "Özdevinirler Kuramı ve Biçimsel Diller", Bıçaklar Kitabevi, 2003, ISBN# 975-8695-05-3
- Namık Kemal Üniversitesi, Yrd. Doç. Dr. E. Serdar Güner Ders Sunumları
- Sakarya Üniversitesi, Biçimsel Diller ve Soyut Makineler Ders Sunumları
- Gazi Üniversitesi, Biçimsel Diller ve Otomatlar Ders Sunumları