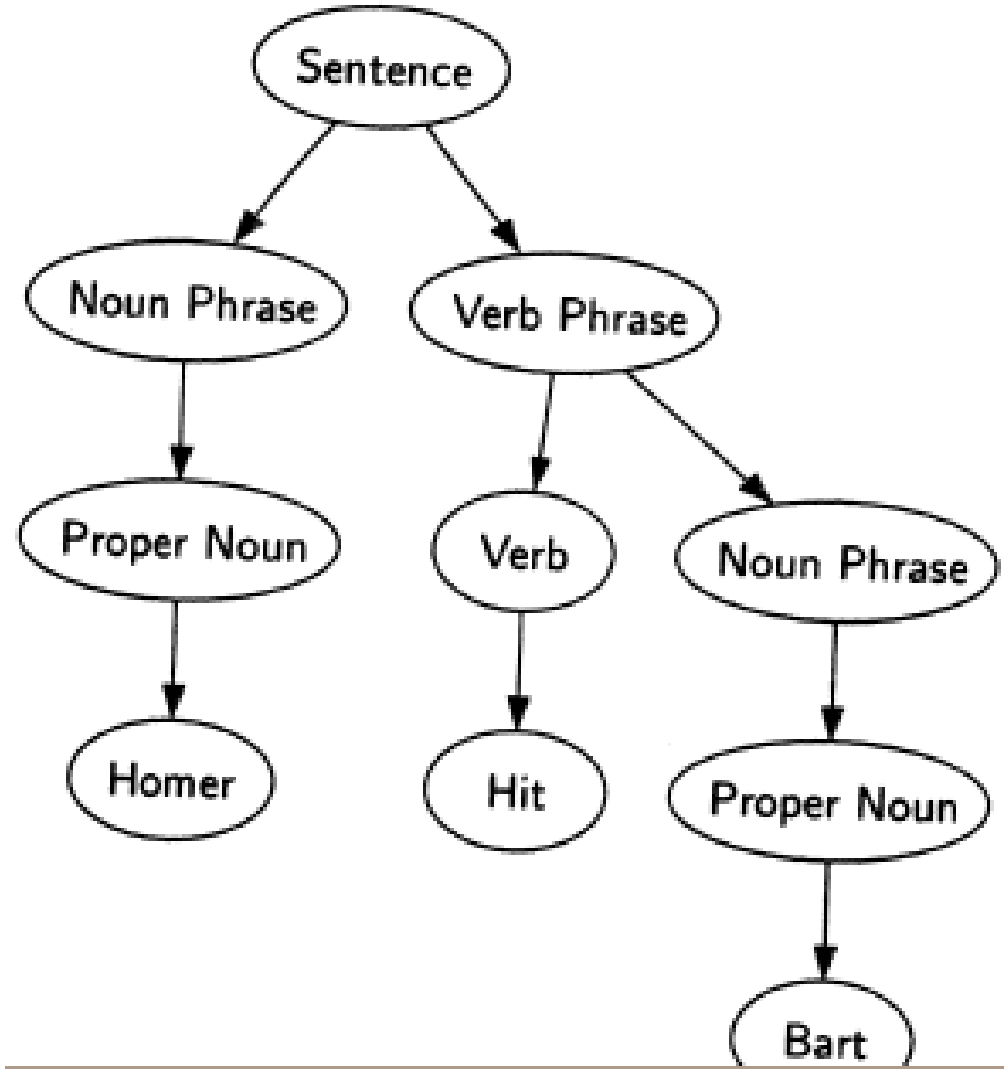
5.5 ikil ağaç uygulamaları

todo

5.5.1 parse ağacı

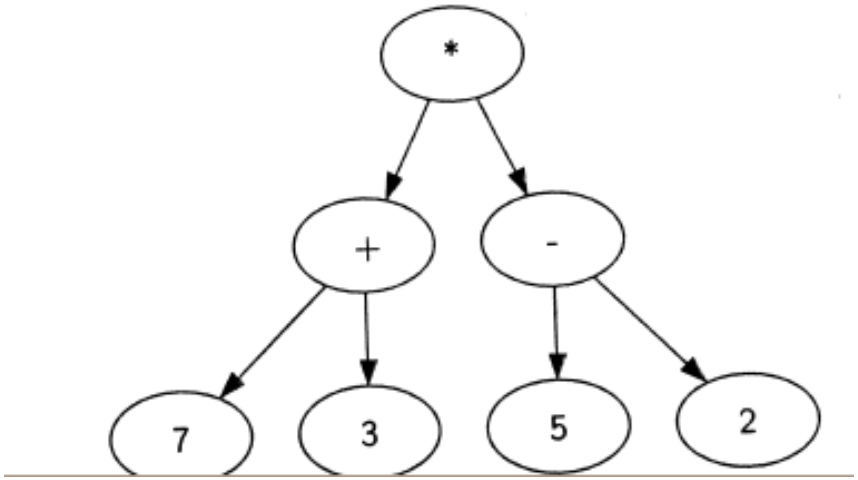
parse ağacı



→ cümle veya matematiksel ifadeleri temsil etmede kullanılır

matematiksel parse ağacı

parse ağacı



bir aşama ilerletilip,
sadeleştirilmişi

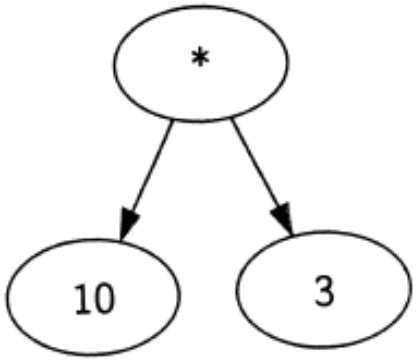


Figure 5.11: A simplified parse tree for $((7+3)*(5-2))$

$\longrightarrow$ önce en diptekileri hesapla

ilgimiz

1. tam parantezli matematiksel ifadeden parse ağacını inşa etme
2. parse tree deki ifadeyi hesaplama
3. parse tree den orjinal (infix) ifadeye dönme

token list: sol|sağ parantez, işleç, işlenen

1. **sol** parantez ifade başlatır; yeni ağacı oluştur
2. **sağ** parantez ifade sonlandırır
3. **işlenenler** yaprak düğümlerinde ve işleçlerin çocuklarıdır
4. her **işlecin** sağ ve sol çocuğu vardır

aşamalar

matematiksel ifade: **(** (7 + 3) * (5 - 2))

1. token = '(',
   a. şimdiki düğümün sol çocuğu olarak yeni düğüm ekle ve
   b. sol düğüme geç
2. token = '(', yinele (1)

1. gerçekleme

```
1  currentTree.insertLeft('')
2
3  pStack.push(currentTree)
4  currentTree = currentTree.\
5      getLeftChild()
```

2. yinele (1)

aşamalar

matematiksel ifade: ((**7** + 3) * (5 - 2))

3. token=7, işlenen
   a. şimdiki düğümün köküne işleneni koy ve
   b. ebeveyne çık

3. gerçekleme

```
1  currentTree.setRootVal(eval(i))
2  parent = pStack.pop()
3  currentTree = parent
```

aşamalar

matematiksel ifade: ((7 + 3) * (5 - 2))

4. token=+, işleç
   a. şimdiki düğümün köküne işleci koy,
   b. sağa yeni düğüm ekle ve
   c. sağ düğüme geç

4. gerçekleme

```
1  currentTree.setRootVal(i)
2
3  currentTree.insertRight('')
4
5  pStack.push(currentTree)
6  currentTree = currentTree.\
7      getRightChild()
```

aşamalar

matematiksel ifade: ((7 + 3) * (5 - 2))

5. token=3, işelenen: yinele (3)
6. token=), kapama parantezi
    a. şimdiki düğümün ebeveynine dön

5. gerçekleme: yinele (3)
6. gerçekleme

```
1   currentTree = pStack.pop()
```

aşamalar

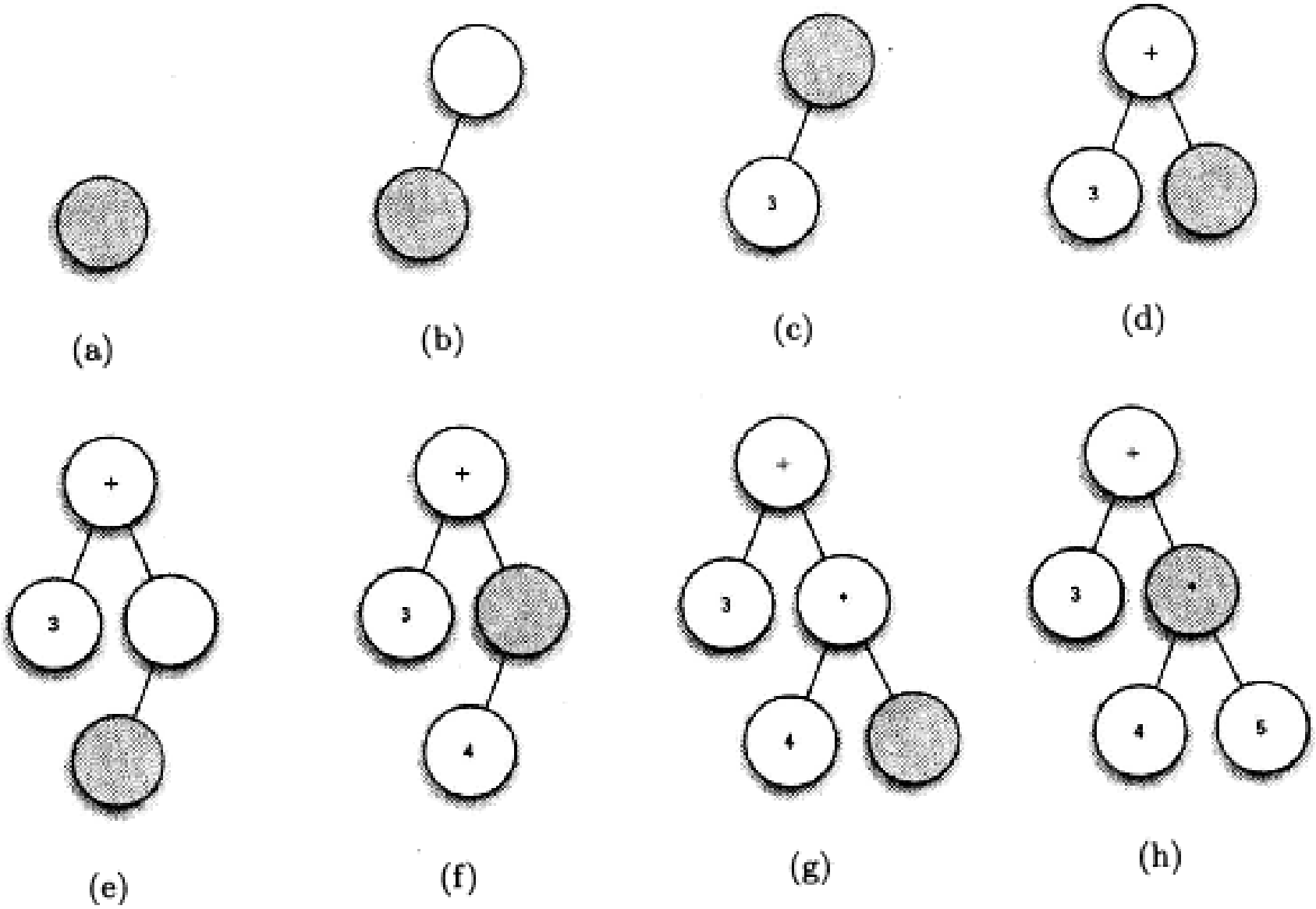
matematiksel ifade: ((7 + 3) * (5 - 2))

7. token=*, yinele (3)
8. token=(, yinele (1)
9. token=5, yinele (3)
10. token=-, yinele (4)
11. vs

→ dipnot: agacsunu.pdf

gösterimler

matematiksel ifade: (3 + (4 * 5)) için ağacın büyümesi



gerçekleme

gerçekleme

```
1       def buildParseTree(fpexp):
2           fplist = fpexp.split()
3           pStack = Stack()
4           eTree = BinaryTree('')
5           pStack.push(eTree)
6           currentTree = eTree
7           for i in fplist:
8               if i == '(':
9                   currentTree.insertLeft('')
10                  pStack.push(currentTree)
11                  currentTree = currentTree.getLeftChild()
12              elif i not in '+-*/)':
13                  currentTree.setRootVal(eval(i))
14                  parent = pStack.pop()
15                  currentTree = parent
16              elif i in '+-*/':
17                  currentTree.setRootVal(i)
18                  currentTree.insertRight('')
19                  pStack.push(currentTree)
20                  currentTree = currentTree.getRightChild()
21              elif i == ')':
22                  currentTree = pStack.pop()
23              else:           print "error:  I don't recognize " + i
24          return eTree
```

açıklama

- → pStack: ebeveyni tutmak için kullanılan yığıt
- → alt düğüme/çocuğa atlarken, şimdiki düğüm (çocuğun ebeveyni) yığıta itiliyor: pStack.push(currentTree)
- → ebeveyne dönmek gerektiğinde çekiliyor: parent=pStack.pop()