

BLG447
Derleyici Tasarımı

2013 – 2014 Güz Dönemi

Ödev 1

Ozan Arkan CAN

040090573

13.11.2013

GİRİŞ

Bu ödevde aşağıda belirtilen gramer için geçerli olmak üzere verilen bir girdinin ayrıştırılabilirliğini kontrol etmek için bir rekürsif iniş ayrıştırıcısı tasarlanıp gerçekleştirilmiştir.

Gramer:

$$\text{Prog} \rightarrow \{ \text{Deyimler} \} \text{ Eof}$$
$$\text{Deyimler} \rightarrow \text{Deyim Deyimler}$$
$$\text{Deyimler} \rightarrow \epsilon$$
$$\text{Deyim} \rightarrow \text{id} = \text{Exp} ;$$
$$\text{Deyim} \rightarrow \text{if} (\text{Exp}) \text{ Deyim}$$
$$\text{Exp} \rightarrow \text{id} \text{ Expt}$$
$$\text{Expt} \rightarrow + \text{Exp}$$
$$\text{Expt} \rightarrow - \text{Exp}$$
$$\text{Expt} \rightarrow \epsilon$$

Gramerde yer alan nonterminal simgeler: { Prog, Deyimler, Deyim, Exp, Expt }

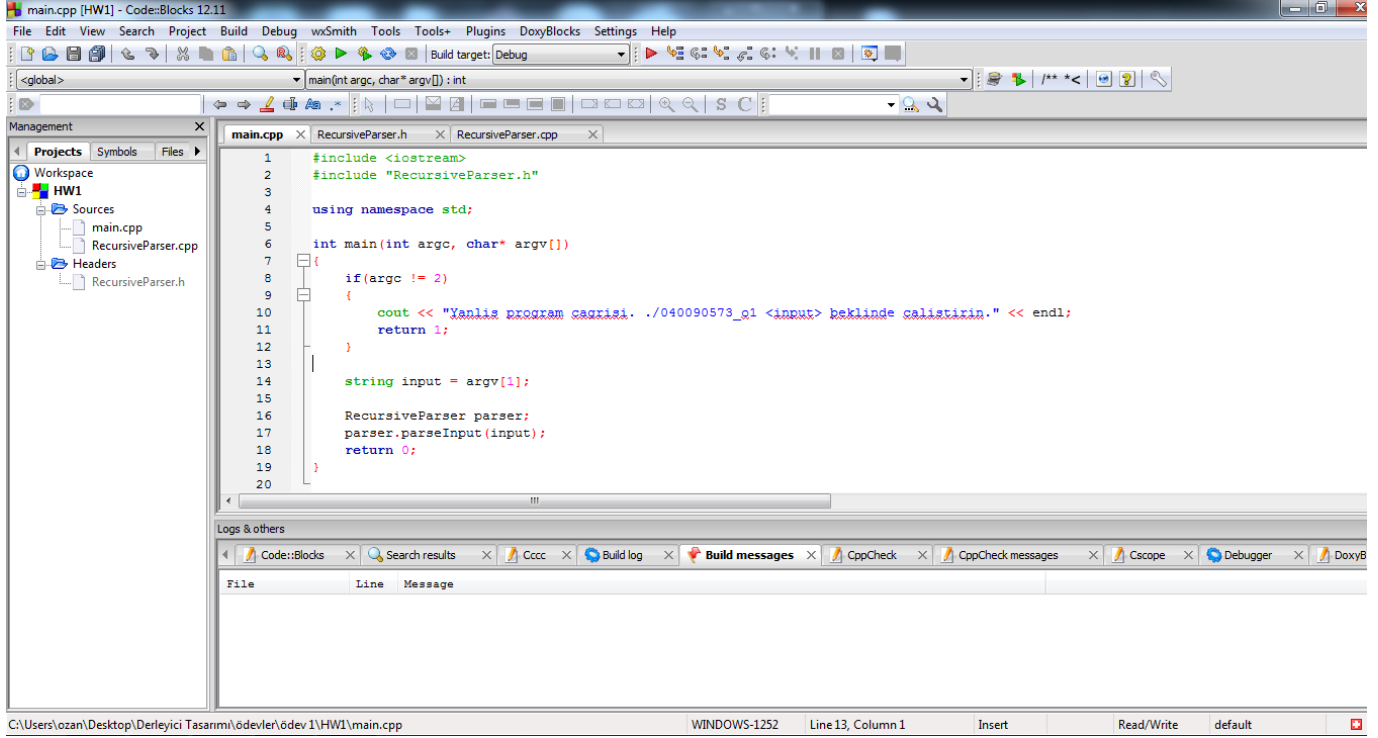
Gramerde yer alan terminal simgeler: { -, +, if, id (değişken adı), (,), =, ;, {}, Eof }

Ayrıştırıcı gerçekleştirirken girdi olarak alınırken sorun olmaması için EOF yerine \$ karakteri kullanılmıştır.

GELİŞTİRME VE ÇALIŞTIRMA ORTAMLARI

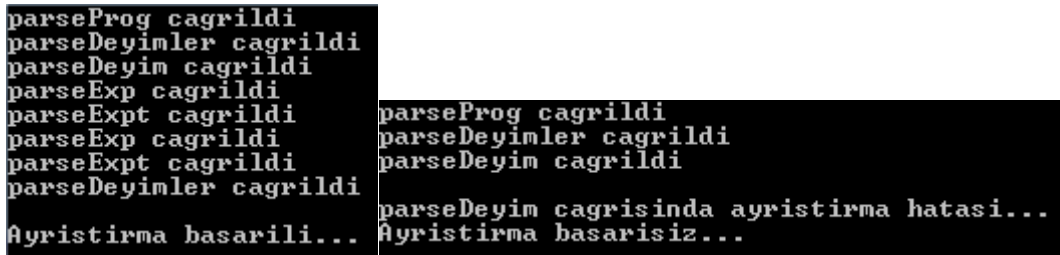
Windows:

Ödev Windows 7 işletim sisteminde g++ destekleyen mingw içeren, Code::Blocks geliştirme ortamında yazılmış ve test edilmiştir.



Şekil 1: Uygulamanın geliştirme ortamına dair ekran görüntüsü

Ödev metninde örnek olarak verilen girdilere ait ekran görüntüleri aşağıdadır. Daha öncede bahsedildiği üzere girdi verilirken Eof yerine \$ karakteri kullanılmıştır.



Şekil 2 ve 3: Sırasıyla “{a=b+c;}\$” ve “{b+c=a;}\$” girdileri için programın çıktısı

Linux:

Derleme işleme için aşağıdaki komut kullanılmıştır:

```
g++ main.cpp RecursiveParser.h RecursiveParser.cpp -o dt_01
```

Programı çalıştırmak için: ./dt_01 <girdi> komutunda istenilen girdinin verilmesi yeterlidir.

```
[cano@ssh 01]$ g++ main.cpp RecursiveParser.h RecursiveParser.cpp -o dt_01
[cano@ssh 01]$ ./dt_01 "{a=b+c;}"
parseProg cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyimler cagrildi

Ayristirma basarili...
[cano@ssh 01]$ ./dt_01 "{b+c=a;}"
parseProg cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi

parseDeyim cagrisinda ayristirma hatasi...
Ayristirma basarisiz...
```

Şekil 4: Programın Linux ortamında derlenmesine ve çalıştırılmasına ilişkin ekran görüntüsü

KAYNAK DOSYA VE SINIF YAPISI

Ödevin gerçekleştirilebilmesi için tek bir sınıfın, RecursiveParser sınıfının olması yeterli görülmüştür.

void parseInput(string input):

parseInput fonksiyonu RecursiveParser sınıfının tek public fonksiyonudur. Bir inputu alır ve gramer kurallarını başlangıç (Prog) simgesinden başlayarak ayrıştırma fonksiyonlarını çağırır.

char getNextToken():

Ödevde değişken isimleri tek karakterle temsil edilmiştir. Dili oluşturan sözcükler içerisinde birden çok karaktere sahip tek sözcük “if” sözcüğüdür. Sıradaki sözcük bulunurken, girdi içerisinde i den sonra f geliyorsa bu dizi if sözcüğü olarak nitelendirilmiştir. switch-case yapısında kontrol edebilmek için if sözcüğü I karakteriyle gösterilmiştir. Yine aynı şekilde tüm değişken isimleri, yani a-z tek tek switch-case yapısında bulunmaması için ortak bir gösterim olarak V karakteri kullanılmıştır. Bu fonksiyon içerisinde, bahsedilen hususlara göre input içerisinden sıradaki sözcük saptanıp döndürülmektedir.

*Diğer fonksiyonlar, nonterminal simgeleri ayrıştıran fonksiyonlardır.

RecursiveParser.h dosyasında RecursiveParser sınıfına ait fonksiyonların imzalarının bulunduğu sınıf yapısı bulunmaktadır. RecursiveParser.cpp dosyasında bu fonksiyonların gövdeleri bulunmaktadır. Bu iki dosyaya ek olarak main.cpp dosyasında, programa verilen parametre sayısının doğru olup olmadığı kontrol edilir, yanlışsa kullanıcı bilgilendirilir, doğru ise girdi RecursiveParser sınıfı yardımıyla ayrıştırılmaya çalışılır.

ÖRNEK ÇALIŞMA

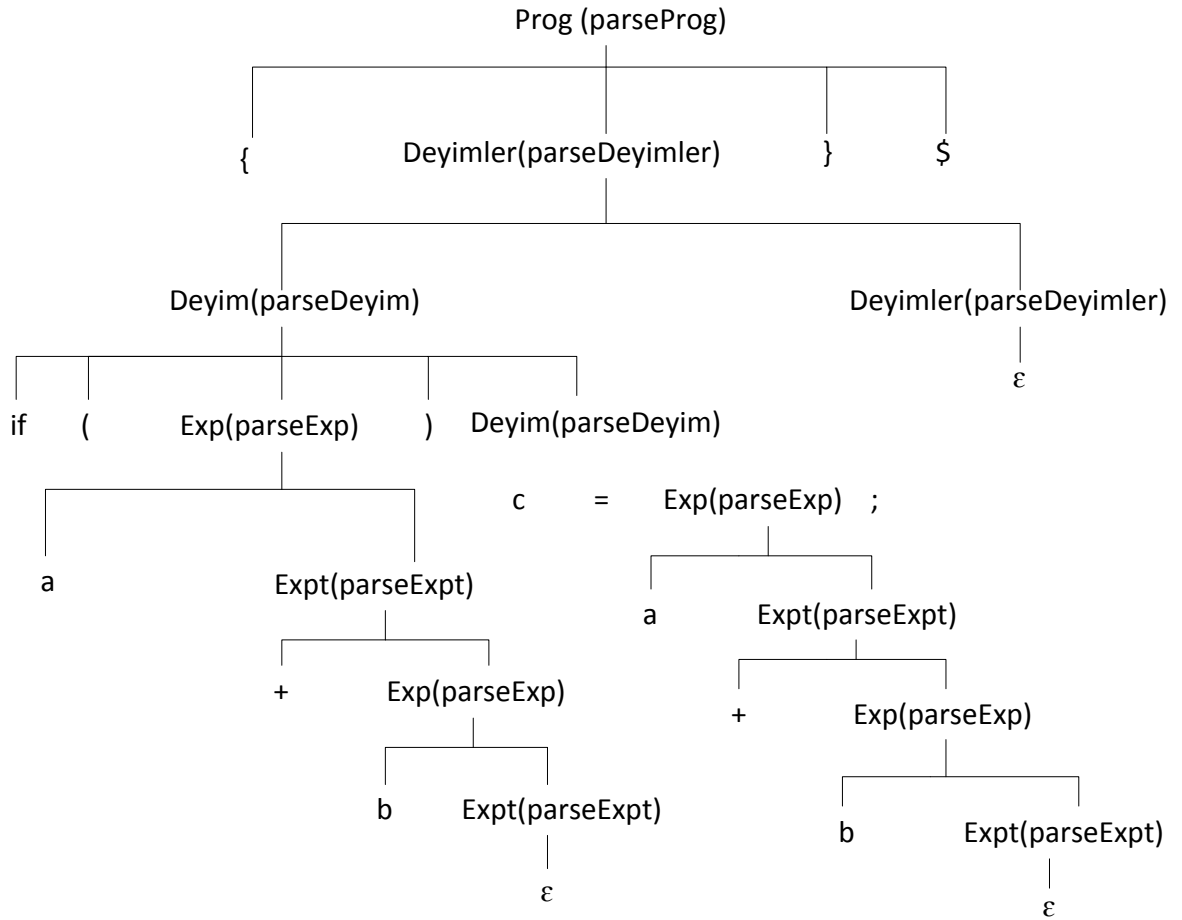
Ayrıştırma işlemi başarılı girdi örnekleri:

Girdi 1 : {if(a+b)c=a+b;}\$

```
parseProg cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyimler cagrildi
Ayrıştırma basarili...
```

Şekil 5: Girdi 1 için ekran görüntüsü

Programın çalışma akışını temsil eden ayrıştırma ağacı aşağıdaki gibidir. Parantez içerisinde ayrıştırma fonksiyonlarının çağrıları görülmektedir.



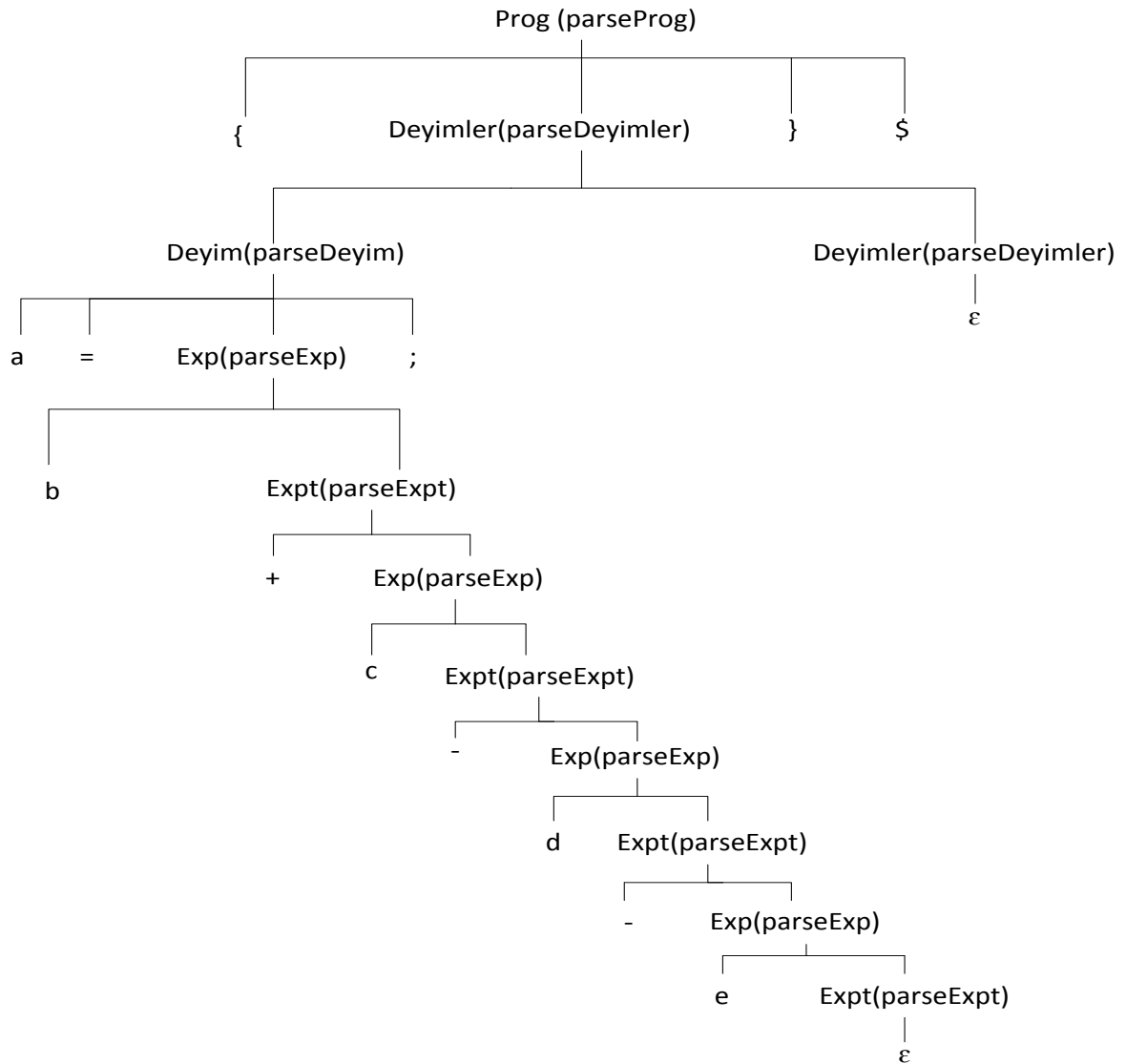
Şekil 6: Girdi 1 için ayrıştırma ağacı

Girdi 2 : {a=b+c-d-e;}\$

```
parseProg cagildi
parseDeyimler cagildi
parseDeyim cagildi
parseExp cagildi
parseExpt cagildi
parseExp cagildi
parseExpt cagildi
parseExp cagildi
parseExpt cagildi
parseExp cagildi
parseExpt cagildi
parseDeyimler cagildi
Ayrıştırma basarili...
```

Şekil 7: Girdi 2 için ekran görüntüsü

Programın çalışma akışını temsil eden ayrıştırma ağacı aşağıdaki gibidir. Parantez içerisinde ayrıştırma fonksiyonlarının çağrıları görülmektedir.



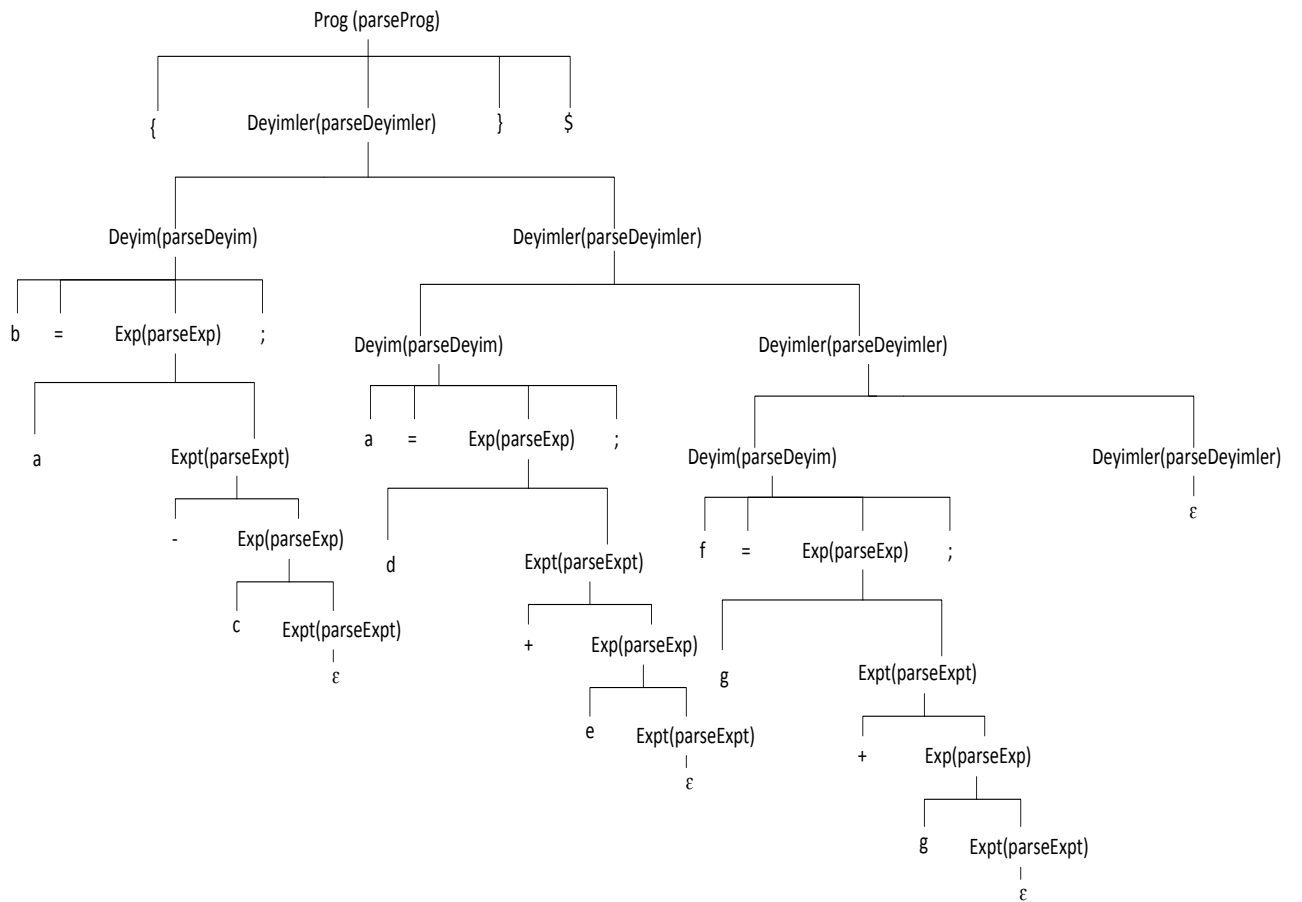
Şekil 8: Girdi 2 için ayrıştırma ağacı

Girdi 3 : {b=a-c;a=d+e;f=g+g;}\$

```
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyimler cagrildi
Ayrıştırma basarili...
```

Şekil 9: Girdi 3 için ekran görüntüsü

Programın çalışma akışını temsil eden ayrıştırma ağacı aşağıdaki gibidir. Parantez içerisinde ayrıştırma fonksiyonlarının çağrıları görülmektedir.



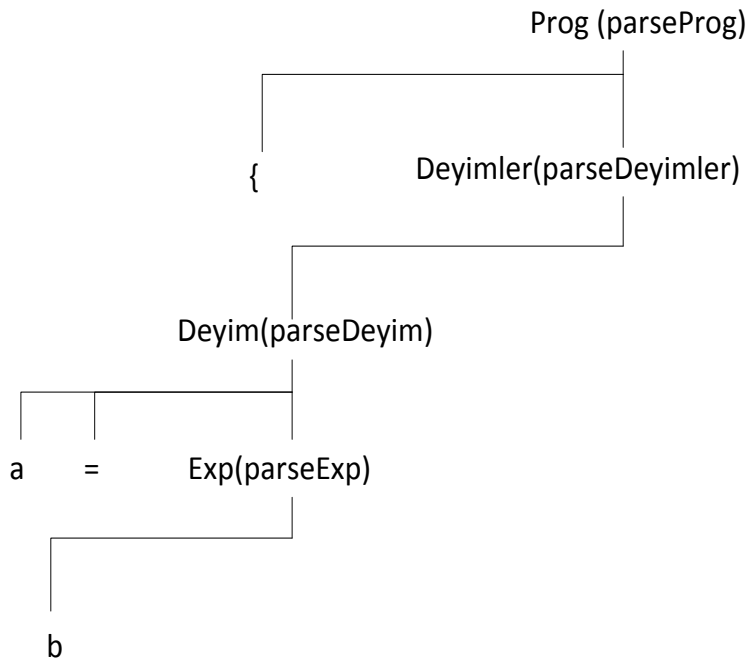
Şekil 10: Girdi 3 için ayrıştırma ağacı

Ayrıştırma işlemi başarısız girdi örnekleri:

Girdi 4: {a=b*c}\$

```
parseProg cagildi  
parseDeyimler cagildi  
parseDeyim cagildi  
parseExp cagildi  
  
Gecersiz girdi bulunmakta...  
Ayrıştırma başarısız...
```

Şekil 11: Girdi 4 için ekran görüntüsü



Şekil 12: Girdi 4 için ayrıştırma ağacının hata alınana kadarki durumu

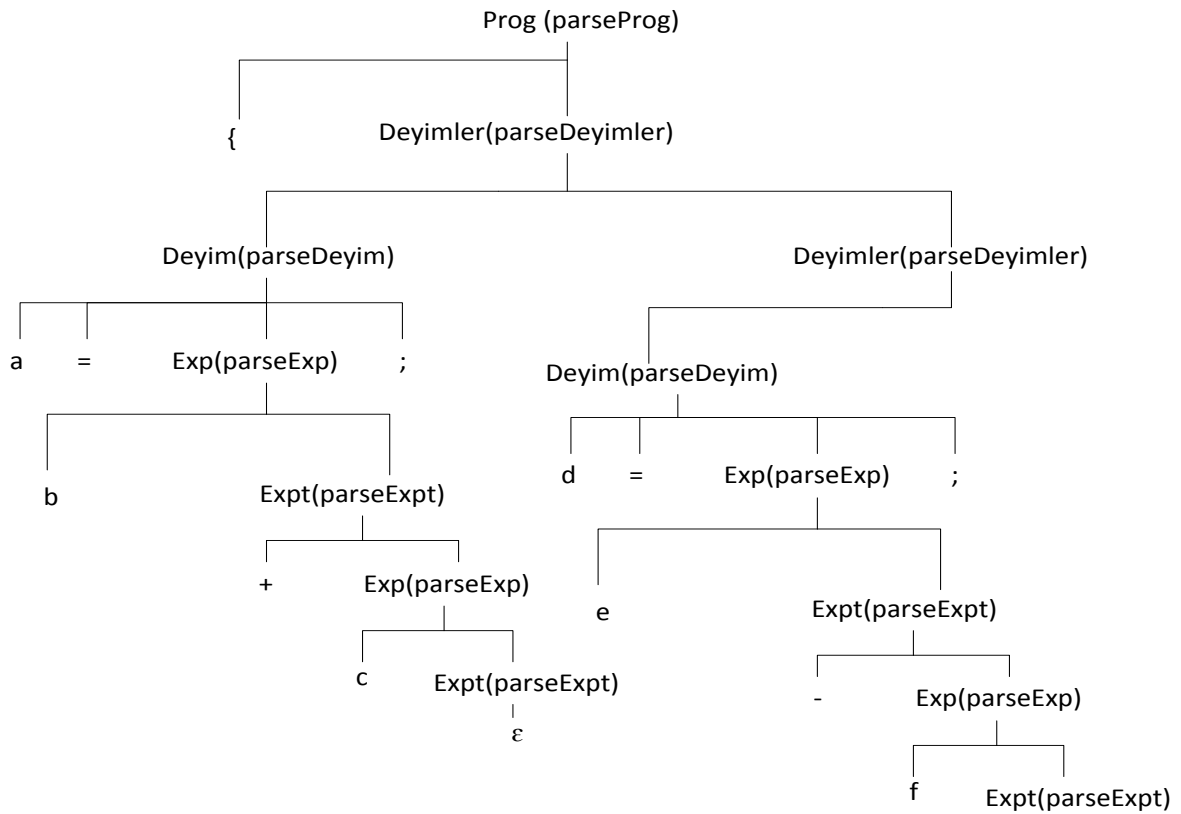
Program sırayla sözcükleri alarak, ayrıştırma tablosuna uygun bir şekilde girdiyi ayrıştırırken *parseExp* fonksiyonu içinde sıradaki sözcük b'yi okuyunca herhangi bir çağrı yapmaz ve sıradaki sözcüğü okur. Sıradaki sözcük '*' olduğu için ve bu sözcük tanımlanmadığı için, *getNextToken* methodu içerisinde geçersiz girdi bulunduğuna dair hata fırlatılır ve ayrıştırmanın başarısız olduğu belirtilir.

Girdi 5: {a=b+c;d=e-f}\$

```
parseProg cagildi
parseDeyimler cagildi
parseDeyim cagildi
parseExp cagildi
parseExpt cagildi
parseExp cagildi
parseExpt cagildi
parseDeyimler cagildi
parseDeyim cagildi
parseExp cagildi
parseExpt cagildi
parseExp cagildi
parseExpt cagildi

parseExpt cagrisinda ayristirma hatasi...
Ayristirma basarisiz...
```

Şekil 13: Girdi 5 için ekran görüntüsü



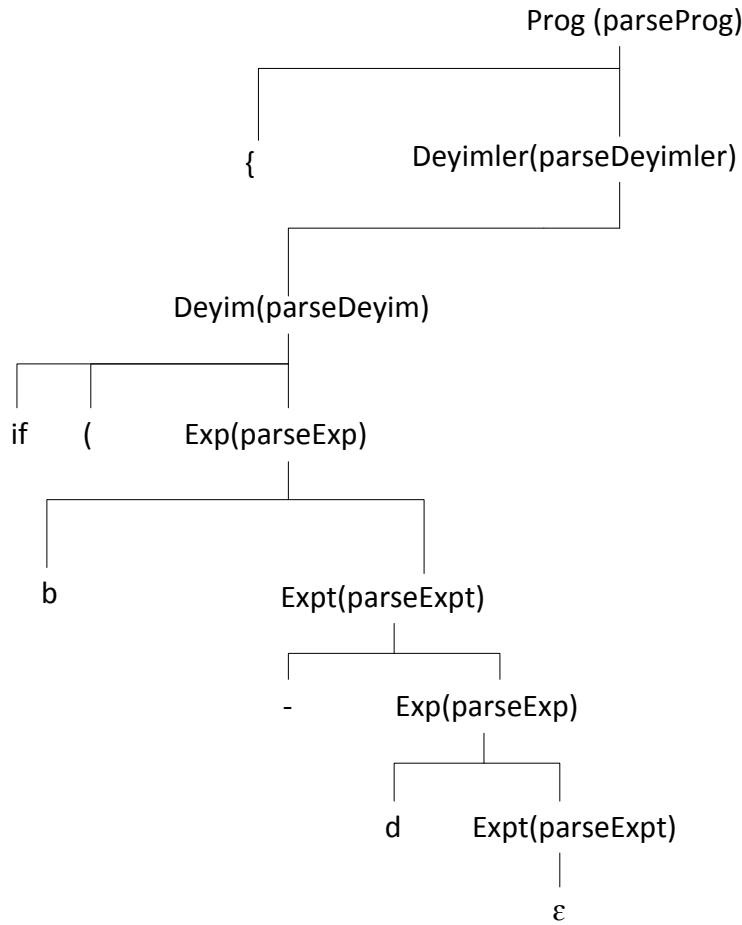
Şekil 14: Girdi 5 için ayrıştırma ağacının hata alınana kadarki durumu

Program sırayla sözcükleri alarak, ayrıştırma tablosuna uygun bir şekilde girdiyi ayrıştırırken yukarıda görülen aşamada sıradaki sözcük '}' karakteridir. *parseExpt* methodu içerisinde ayrıştırma tablosuna göre uygun bir adım bulunmadığı için ayrıştırma hatası verilir.

Girdi 6: {if(b-d;)}\$

```
parseProg cagrildi
parseDeyimler cagrildi
parseDeyim cagrildi
parseExp cagrildi
parseExpt cagrildi
parseExp cagrildi
parseExpt cagrildi
parseDeyim cagrisinda ayristirma hatasi...
Ayristirma basarisiz...
```

Şekil 15: Girdi 6 için ekran görüntüsü



Şekil 16: Girdi 6 için ayrıştırma ağacının hata alınana kadarki durumu

Program akışında yukarıdaki aşamaya gelinildiğinde sıradaki sözcük ‘;’ karakteridir. Son *parseExpt* çağrısı yapıldığında başarılı şekilde fonksiyon çağrılarını *parseDeyim* çağrısına göre geri döner. Bu aşamada *parseDeyim* çağrısı içerisinde sıradaki sözcüğün ‘)’ karakteri olması beklenmekte fakat ‘;’ karakteri olduğu için bu aşamada ayrıştırma hatası dönmektedir.

SONUÇ

Yapılan bu çalışmada ödev metninde verilen gramere ait bir rekürsif iniş ayrıştırıcısı gerçekleştirilmiştir. Gerçeklenen ayrıştırıcı verilen bir girdi için başarılı bir şekilde, o girdinin ayrıştırılıp ayrıştırılamayacağını, ayrıştırılamıyorsa nerede hata olduğunu bildirebilmektedir.

Bu ödevle birlikte;

- Bir dile ait gramer
- Gramere ait ayrıştırma tablosu
- Ayrıştırma ağacı

Kavramları pekiştirilmiş oldu.

Ayrıca, verilen bir gramer için bir girdinin ayrıştırılma adımlarının uygulanmasıyla ilgili pratik edinilmiş olundu.