

DERLEYİCİ TASARIMI

PROJE

Projede türetim kuralları verilen bir gramer için bir LR(0) ayrıştırıcısı geliştirilecektir. Çalışma iki bölüm halinde gerçekleştirilecek ve farklı tarihlerde teslim edilecektir.

1. BÖLÜM

1. Gramer kurallarını bir dosyadan okuyup, bellekte uygun bir veri yapısı ile temsil edin.
2. Kılıf ve Geçiş fonksiyonlarını kodlayın ve gramere ait DFA'yı oluşturun.
3. DFA'yı kullanarak LR(0) ayrıştırma tablosunu oluşturun.

2. BÖLÜM

4. Ötele-İndirge ayrıştırma algoritmasını gerçekleyen ayrıştırıcıya ait kodu yazın.
5. Örnek tarayıcı çıktıları üzerinde ayrıştırma işlemini yürütün. Ayrıştırma işleminin başarılı ve başarısız olduğu iki farklı örnek için ayrıştırıcıyı deneyin. Ayrıştırıcı her ayrıştırma adımı için açıklayıcı bilgileri içeren bir satırı çıkışa aktarmalıdır.

NOT: 1. Bölümde oluşturulan ayrıştırma tablosunun bazı durumlarında “ötele/indirge” çelişkisi ortaya çıkacaktır. Bu durumu çözmek için iki seçenekten birini iptal edin (hangisi tercih edilmeli, karar verin).

ÖNEMLİ NOT:

- a) 1. bölüme ait çalışma teslim edilmeden ikincisi teslim edilemez.
- b) 1. bölümü zamanında teslim etmemiş olan öğrenciler 2. Bölüme ait çalışmanın teslim tarihinde her ikisini birlikte teslim edebilirler ancak 1. bölüm için puan alamazlar.

TESLİM TARİHLERİ:

BÖLÜM 1: 16 Kasım 2014

BÖLÜM 2: 4 Aralık 2014

ÖDEVİN TESLİM ŞEKLİ:

Ödev C++ dilinde nesneye yönelik programlama kavramları kullanılarak kodlanacaktır. Kodunuz Linux üzerinde g++ derleyicisinde derlenebilir ve çalıştırılabilir olmalıdır. Her bölüm için kod ile birlikte teslim edilecek olan raporda çalışma ayrıntılı bir şekilde anlatılacaktır: kullanılan veri yapıları ile tüm sınıf tanımları verilecektir. Ayrıca, ayrıştırma işleminin uygulandığı örnek tarayıcı çıktıları ile bu örnekler için ayrıştırıcının ürettiği adımlar ve sonuçlar eklenecektir. Kod ve rapor dosyaları Ninova üzerinden teslim edilecektir. Proje bireysel olarak yapılacaktır. Dolayısıyla, bireysel çalışmanız gerekmektedir.

ÖRNEK GRAMER:

Aşağıda verilen örnek gramer kullanılabilir.

1. $S' \rightarrow S \$$
2. $S \rightarrow S ; A$
3. $S \rightarrow A$
4. $A \rightarrow E$
5. $A \rightarrow \text{id} := E$
6. $E \rightarrow E + \text{id}$
7. $E \rightarrow \text{id}$

Ayrıştırıcıya ait DFA şu şekildedir:

```
I0 := { S' → . S $, S → . S ; A,  
        S → . A, A → . E, A → . id := E, E → . E + id, E → . id }  
I1 := { S' → S . $, S → S . ; A }  
  
I2 := { S → A . }  
I5 := { S → S ; . A, A → . E, A → . id := E, E → . E + id, E → . id }  
I8 := { S → S ; A . }  
  
I3 := { A → E . , E → E . + id }  
I4 := { A → id . := E, E → id . }  
I7 := { A → id := . E, E → . E + id, E → . id }  
I10:= { A → id := E . , E → E . + id }  
  
I6 := { E → E + . id}  
  
I9:= { E → E + id . }  
I11:= { E → id . }
```

Durumlar arasındaki geçişler:

$I_0 \rightarrow [S] I_1$
 $I_0 \rightarrow [A] I_2$
 $I_0 \rightarrow [E] I_3$
 $I_0 \rightarrow [id] I_4$
 $I_1 \rightarrow [;] I_5$
 $I_3 \rightarrow [+] I_6$
 $I_4 \rightarrow [:=] I_7$
 $I_5 \rightarrow [A] I_8$
 $I_5 \rightarrow [E] I_3$
 $I_5 \rightarrow [id] I_4$
 $I_6 \rightarrow [id] I_9$
 $I_7 \rightarrow [E] I_{10}$
 $I_7 \rightarrow [id] I_{11}$
 $I_{10} \rightarrow [10] I_6$

LR(0) tablosu aşağıdaki gibi oluşacaktır.

State	Action					Goto			
	Id	;	+	:=	\$	S'	S	A	E
0	S I4						I1	I2	I3
1		S I5			acc				
2	R 3	R 3	R 3	R 3	R 3				
3	R 4	R 4	R 4 S I6	R 4	R 4				
4	R 7	R 7	R 7	R 7 S I7	R 7				
5	S I4							I8	I3
6	S I9								
7	S I11								I10
8	R 2	R 2	R 2	R 2	R 2				
9	R 6	R 6	R 6	R 6	R 6				
10	R 5	R 5	R 5 S I6	R 5	R 5				
11	R 7	R 7	R 7	R 7	R 7				