CS 164 Discussion Notes

Earley's Algorithm

GSIs: Marten Lohstroh, Jonathon Cai

Spring '18

Berkeley
UNIVERSITY OF CALIFORNIA

Algorithm

The state set at input position k is called $S(k)$. The parser is seeded with $S(0)$ consisting of only the top-level rule. The parser then repeatedly executes three operations:

- ***Prediction***: For every state in $S(k)$ of the form $(X \rightarrow \alpha \bullet Y\ \beta, j)$ (where j is the origin position as above), add $(Y \rightarrow \bullet\ \gamma, k)$ to $S(k)$ for every production in the grammar with Y on the left-hand side $(Y \rightarrow \gamma)$.
- ***Scanning***: If a is the next symbol in the input stream, for every state in $S(k)$ of the form $(X \rightarrow \alpha \bullet a\ \beta, j)$, add $(X \rightarrow \alpha\ a \bullet \beta, j)$ to $S(k+1)$.
- ***Completion***: For every state in $S(k)$ of the form $(X \rightarrow \gamma \bullet, j)$, find states in $S(j)$ of the form $(Y \rightarrow \alpha \bullet X\ \beta, i)$ and add $(Y \rightarrow \alpha\ X \bullet \beta, i)$ to $S(k)$.

Note: duplicate states are not added to the state set, only new ones. These three operations are repeated until no new states can be added to the set.

https://en.wikipedia.org/wiki/Earley_parser

Example

Consider the following grammar:

P → E ⊣

E → E + E

E → E * E

E → ID

Example (con't)

Consider the following input string:

```
ID + ID * ID
```

And show whether or not:

- it is part of the language defined by our grammar;
- our grammar is ambiguous.

Representation

We represent our state sets in a table, where each column denotes a state set, and we use the row indexes (*a*-*k*) to trace the origin of newly produced completions. Our first item in S(0) is based on the start symbol of our grammar:

$$\mathbf{P} \rightarrow \mathbf{E} \dashv$$

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P → ⚛E ⊣, 0					
b						
c						
d						
e						
f						
g						
h						
i						
j						
k						

Note: For enhanced readability, we use '⚛' to denote the usual dot.

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

		ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→⚛E ⊣, 0					
b	E→⚛E+E, 0					
c	E→⚛E*E, 0					
d	E→⚛ID, 0					
e						
f						
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣

E → E + E

E → E * E

E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❀E ⊣, 0	E→ID❀, 0				
b	E→❀E+E, 0					
c	E→❀E*E, 0					
d	E→❀ID, 0					
e						
f						
g						
h						
i						
j						
k						

...scanning the first token.

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❀E ⊣, 0	E→ID❀, 0				
b	E→❀E+E, 0	P→E_a❀ ⊣, 0				
c	E→❀E*E, 0	E→E_a❀+E, 0				
d	E→❀ID, 0					
e						
f						
g						
h						
i						
j						
k						

...we have a finished item! Now we must complete items from the previous set...

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→⚛E ⊣, 0	E→ID⚛, 0	E→E+⚛E, 0			
b	E→⚛E+E, 0	P→E_a⚛ ⊣, 0				
c	E→⚛E*E, 0	E→E_a⚛+E, 0				
d	E→⚛ID, 0	E→E_a⚛*E, 0				
e						
f						
g						
h						
i						
j						
k						

Note: we can safely remove items that are incompatible with the next token on the input queue.

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0			
b	E→❁E+E, 0		E→❁E+E, 2			
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2			
d	E→❁ID, 0		E→❁ID, 2			
e						
f						
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣

E → E + E

E → E * E

E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2		
b	E→❁E+E, 0		E→❁E+E, 2			
c	E→❁E*E, 0	$E \rightarrow E_a$❁+E, 0	E→❁E*E, 2			
d	E→❁ID, 0		E→❁ID, 2			
e						
f						
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣

E → E + E

E → E * E

E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2		
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0		
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2	E→E_a❁+E, 2		
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2		
e						
f						
g						
h						
i						
j						
k						

*Note: we **only** advance items from the set that the completed item (E→ID❁, 2) started in; S(2) in this particular case.*

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

	ID	+	ID	*	ID	
	0	1	2	3	4	5
a	P→❀E ⊣, 0	E→ID❀, 0	E→E+❀E, 0	E→ID❀, 2		
b	E→❀E+E, 0		E→❀E+E, 2	E→E+E_a❀, 0		
c	E→❀E*E, 0	E→E_a❀+E, 0	E→❀E*E, 2	E→E_a❀+E, 2		
d	E→❀ID, 0		E→❀ID, 2	E→E_a❀*E, 2		
e				P→E_b❀ ⊣, 0		
f				E→E_b❀*E, 0		
g				E→E_b❀+E, 0		
h						
i						
j						
k						

...yet another finished item!
Do completion again...

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2			
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2		
e						
f				E→E_b❁*E, 0		
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0		
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0		
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	E→E_a❁*E, 4
e					E→❁E+E, 4	E→E_a❁+E, 4
f				E→E_b❁*E, 0		
g						
h						
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	E→E_a❁*E, 4
e					E→❁E+E, 4	E→E_a❁+E, 4
f				E→E_b❁*E, 0		E→E+E_b❁, 0
g						E→E_b❁+E, 2
h						E→E_b❁*E, 2
i						
j						
k						

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	E→E_a❁*E, 4
e					E→❁E+E, 4	E→E_a❁+E, 4
f				E→E_b❁*E, 0		E→E+E_b❁, 0
g						E→E_b❁+E, 2
h						E→E_b❁*E, 2
i						P→E_f❁ ⊣, 0
j						E→E_f❁+E, 0
k						E→E_f❁*E, 0

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0		E→E+E_b❁, 0
g						
h						
i					ID+(ID*ID)	P→E_f❁ ⊣, 0
j						
k						

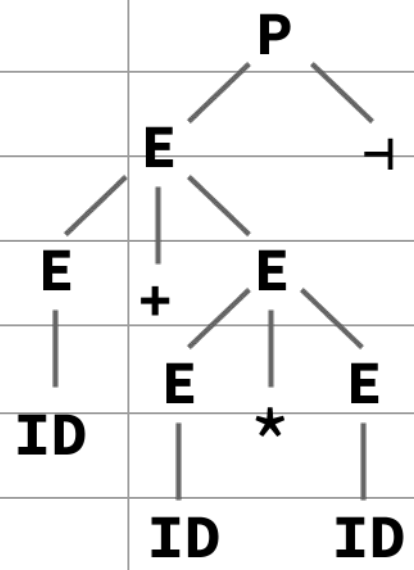
*...we **recognized** the input string!*

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0		E→E+E_b❁, 0
g						
h						
i					ID+(ID*ID)	P→E_f❁ ⊣, 0
j						
k						



Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	E→E_a❁*E, 4
e					E→❁E+E, 4	E→E_a❁+E, 4
f				E→E_b❁*E, 0		P→E_c❁ ⊣, 0
g						E→E_c❁+E, 0
h						E→E_c❁*E, 0
i						
j						
k						

*Note: ...**picking up from Slide 18**...*

Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0	(ID+ID)*ID	P→E_c❁ ⊣, 0
g						
h						
i						
j						
k						

*...we found **another** valid parse tree!*

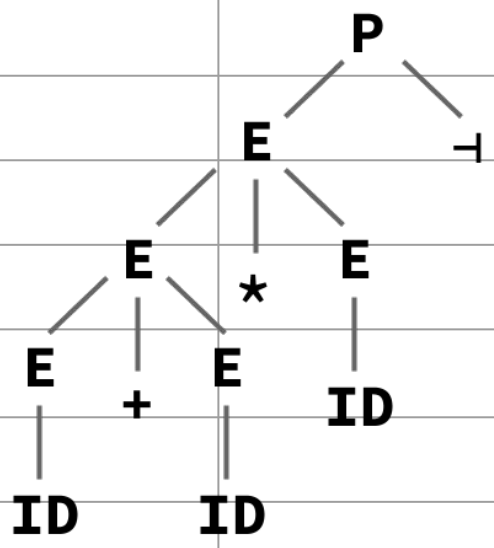
Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

	ID	+	ID	*	ID
	0	1	2	3	4	5
a	P→❁E ⊣, 0	E→ID❁, 0	E→E+❁E, 0	E→ID❁, 2	E→E*❁E, 2	E→ID❁, 4
b	E→❁E+E, 0		E→❁E+E, 2	E→E+E_a❁, 0	E→E*❁E, 0	E→E*E_a❁, 2
c	E→❁E*E, 0	E→E_a❁+E, 0	E→❁E*E, 2		E→❁ID, 4	E→E*E_a❁, 0
d	E→❁ID, 0		E→❁ID, 2	E→E_a❁*E, 2	E→❁E*E, 4	
e					E→❁E+E, 4	
f				E→E_b❁*E, 0	(ID+ID)*ID	P→E_c❁ ⊣, 0
g						
h						
i						
j						
k						



Prediction Scan Completion

Productions:

P → E ⊣
E → E + E
E → E * E
E → ID

Berkeley
UNIVERSITY OF CALIFORNIA

Acceptance

We found two *complete* Earley items that end with ‘⊣’ in our last state set. Because these items span the entire input string (they starts at *zero*), the input string is accepted and hence is part of the language.

Parse Tree Construction

We constructed a parse tree for each accepting state by tracing back the items that led to that accepting state. Since we found more than one tree, our grammar is *ambiguous*.