

Earley Parsing and Examples



Sometimes the second
one is even better!



Outline

- Earley's Algorithm
 - Chart States
 - Operations
 - Example
- MyEarley.py
- PA3.jison
- Grammar “Conflicts”
 - Shift/Reduce



Administrivia

- Midterm 1 will be next week
 - Likely: it will appear as a “Quiz” on Canvas
 - Pick any 90-minute slot on Thursday or Friday
 - Start any time. Once you start you have 90 minutes to finish.
 - Open notes, book, laptop, internet, “cool.exe”, etc.
 - Usual Bowdoin Academic Honor Code
- Forbidden:
 - ChatGPT or similar services
 - I may “spot check” suspicious answers
 - Assistance from another live human
 - No posting new questions on StackOverflow

In One Slide

- **Earley parsers** are top-down and use dynamic programming. An Earley **state** records incremental information: when we started, what has been seen so far, and what we expect to see. The Earley **chart** holds a set of states for each input position. **Shift**, **reduce** and **closure** operations fill in the chart.
- **You** enjoy parsing. Parsing is **approachable** and **fun**.

Review: Earley States

- Let X be a non-terminal
- Let a and b be (possibly-empty) sequences of terminals and non-terminals
- Let $X \rightarrow ab$ be a production in your grammar
- Let j be a position in the input
- Each **Earley State** is a tuple $\langle X \rightarrow a \bullet b, j \rangle$
 - We are currently parsing an X
 - We have seen a , we expect to see b
 - We started parsing this X after seeing the first j tokens from the input.

Review: Earley Parse Table

- An **Earley parsing table** (or **chart**) is a one-dimensional array. Each array element is a **set** of Earley states.
 - **chart[i]** holds the set of valid parsing states we could be in after seeing the first i input tokens
- Then the string $\text{tok}_1 \dots \text{tok}_n$ is in the language of a grammar with start symbol S *iff*
 - $\text{chart}[n]$ contains $\langle S \rightarrow ab \bullet, 0 \rangle$ for some production rule $S \rightarrow ab$ in the grammar.
 - We then say the parser **accepts** the string.

Review: Filling In The Chart

- Three operations build up $\text{chart}[n]$
- The first is called **shift** or **scan**.
 - It corresponds to “seeing the next expected token” or “helping to confirm the current hypothesis” or “we're winning”.
- Example:
 - $\text{chart}[1]$ contains $\langle E \rightarrow E \bullet + E, 0 \rangle$
 - 2nd token is “+”
 - Then put $\langle E \rightarrow E + \bullet E, 0 \rangle$ in $\text{chart}[2]$

Review: Filling In The Chart (2)

- The second operation is the **closure** or **predictor**.
 - It corresponds to “expanding rewrite rules” or “substituting in the definitions of non-terminals”
- Suppose the grammar is:
$$S \rightarrow E \qquad E \rightarrow E + E \mid E - E \mid \text{int}$$
- If $\text{chart}[0]$ has $\langle S \rightarrow \bullet E, 0 \rangle$ then add
 - $\langle E \rightarrow \bullet E + E, 0 \rangle$
 - $\langle E \rightarrow \bullet E - E, 0 \rangle$
 - $\langle E \rightarrow \bullet \text{int}, 0 \rangle$

Review: Filling In The Chart (3)

- The third operation is **reduction** or **completion**.
 - It corresponds to “finishing a grammar rewrite rule” or “being done parsing a non-terminal” or “doing a rewrite rule in reverse and then shifting over the non-terminal”.
- Suppose:
 - $E \rightarrow \text{int} \mid E + E \mid E - E \mid (E)$, input is “(int”
 - chart[2] contains $\langle E \rightarrow \text{int} \bullet, 1 \rangle$
 - chart[1] contains $\langle E \rightarrow (\bullet E) , 0 \rangle$
 - Then chart[2] += $\langle E \rightarrow (E \bullet) , 0 \rangle$

Shift Practice

- `chart[3]` contains

$\langle S \rightarrow E \bullet , 0 \rangle$

$\langle E \rightarrow E \bullet + E , 0 \rangle$

$\langle E \rightarrow E \bullet - E , 2 \rangle$

$\langle E \rightarrow \text{int} \bullet , 2 \rangle$

$\langle E \rightarrow E \bullet - E , 0 \rangle$

$\langle E \rightarrow E - E \bullet , 0 \rangle$

$\langle E \rightarrow E \bullet + E , 2 \rangle$

- The 4th token is “+”. What does **shift** bring in?

Shift Practice

- chart[3] contains

$\langle S \rightarrow E \bullet, 0 \rangle$

$\langle E \rightarrow E \bullet + E, 0 \rangle$

$\langle E \rightarrow E \bullet - E, 2 \rangle$

$\langle E \rightarrow \text{int} \bullet, 2 \rangle$

$\langle E \rightarrow E \bullet - E, 0 \rangle$

$\langle E \rightarrow E - E \bullet, 0 \rangle$

$\langle E \rightarrow E \bullet + E, 2 \rangle$

- The 4th token is “+”. What does **shift** bring in?

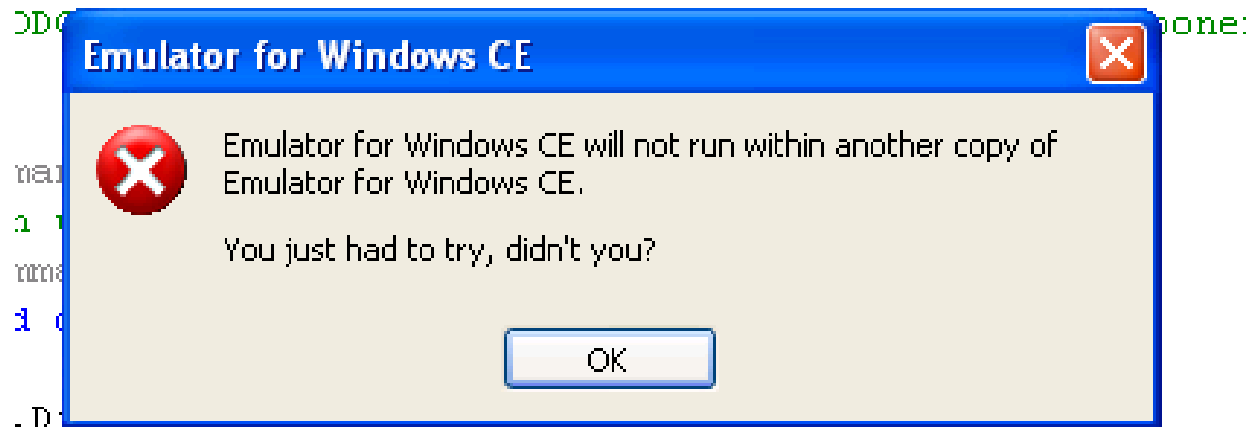
$\langle E \rightarrow E + \bullet E, 0 \rangle$

$\langle E \rightarrow E + \bullet E, 2 \rangle$

... are both added to chart[4].

Closure Practice

- Grammar is
 - $S \rightarrow E$ $E \rightarrow E + E \mid E - E \mid (E) \mid \text{int}$
- `chart[4]` contains:
 - $\langle E \rightarrow E + \bullet E, 0 \rangle$ $\langle E \rightarrow E + \bullet E, 2 \rangle$
- What does the **closure** operation bring in?



Closure Practice

- Grammar is
 - $S \rightarrow E$ $E \rightarrow E + E \mid E - E \mid (E) \mid \text{int}$
- $\text{chart}[4]$ contains:
 - $\langle E \rightarrow E + \bullet E, 0 \rangle$ $\langle E \rightarrow E + \bullet E, 2 \rangle$
- What does the **closure** operation bring in?
 - $\langle E \rightarrow \bullet E + E, 4 \rangle$ $\langle E \rightarrow \bullet E - E, 4 \rangle$
 - $\langle E \rightarrow \bullet (E), 4 \rangle$ $\langle E \rightarrow \bullet \text{int}, 4 \rangle$
 - ... are all added to $\text{chart}[4]$.

Reduction Practice

- chart[4] contains:

$\langle E \rightarrow E + \bullet E, 0 \rangle$

$\langle E \rightarrow \bullet E + E, 4 \rangle$

$\langle E \rightarrow \bullet (E), 4 \rangle$

$\langle E \rightarrow E + \bullet E, 2 \rangle$

$\langle E \rightarrow \bullet E - E, 4 \rangle$

$\langle E \rightarrow \bullet \text{int}, 4 \rangle$

- chart[5] contains:

- $\langle E \rightarrow \text{int} \bullet, 4 \rangle$

- What does the **reduce** operator bring in?



Reduction Practice

- chart[4] contains:

$\langle E \rightarrow E + \bullet E, 0 \rangle$

$\langle E \rightarrow \bullet E + E, 4 \rangle$

$\langle E \rightarrow \bullet (E), 4 \rangle$

$\langle E \rightarrow E + \bullet E, 2 \rangle$

$\langle E \rightarrow \bullet E - E, 4 \rangle$

$\langle E \rightarrow \bullet \text{int}, 4 \rangle$

- chart[5] contains:

- $\langle E \rightarrow \text{int} \bullet, 4 \rangle$

- What does the **reduce** operator bring in?

$\langle E \rightarrow E + E \bullet, 0 \rangle$

$\langle E \rightarrow E \bullet + E, 4 \rangle$

$\langle E \rightarrow E + E \bullet, 2 \rangle$

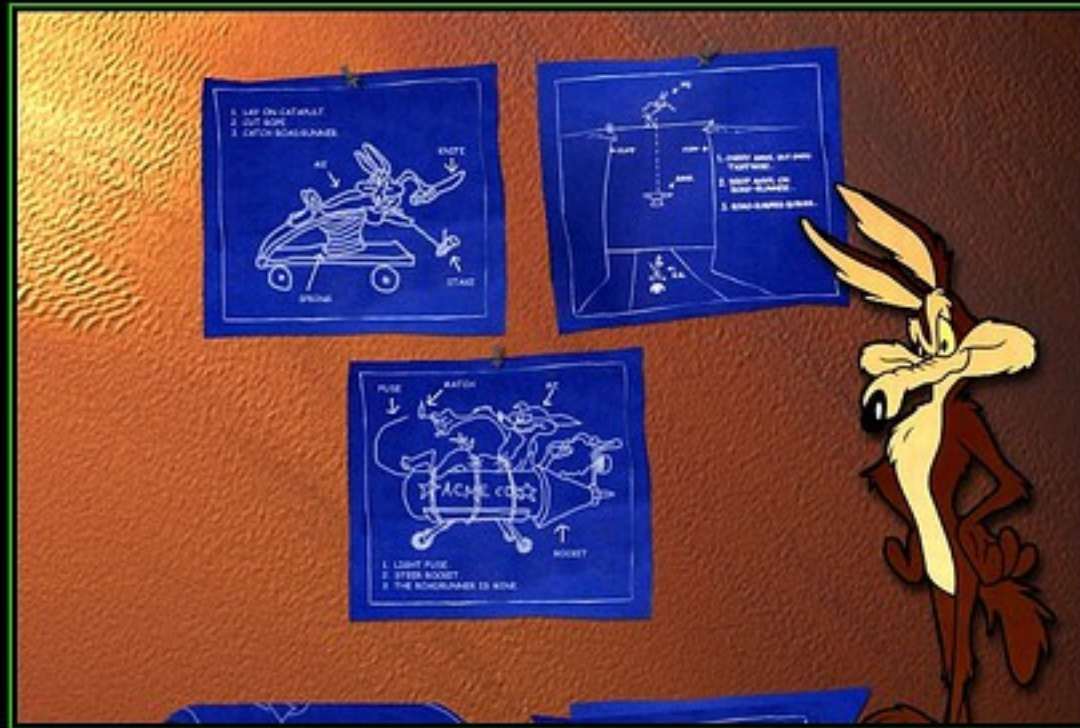
$\langle E \rightarrow E \bullet - E, 4 \rangle$

- ... are all added to chart[5]. (Plus more in a bit!)

Earley Parsing Algorithm

- Input: CFG G , Tokens $\text{tok}_1 \dots \text{tok}_n$
- Work:
 - $\text{chart}[0] = \{ \langle S \rightarrow \bullet ab, 0 \rangle \}$
 - for $i = 0$ to n
 - repeat
 - use shift, reduce and closure on $\text{chart}[i]$
 - until no new states are added
- Output:
 - true iff $\langle S \rightarrow ab\bullet, 0 \rangle$ in $\text{chart}[n]$

Massive Earley Example



chart[0]

$S \rightarrow \bullet F, 0$

chart[6]

PLANNING

No Plan Survives First Contact Intact

Massive Earley Example

Grammar

$S \rightarrow F$

$F \rightarrow \text{id} (A)$

$A \rightarrow N$

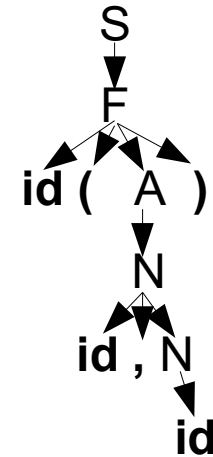
$A \rightarrow \epsilon$

$N \rightarrow \text{id}$

$N \rightarrow \text{id} , N$

Input

$\text{id} (\text{id} , \text{id})$



id

(

id

,

id

)

chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

$S \rightarrow \bullet F , 0$

Massive Earley Example

Grammar

$S \rightarrow F$

$F \rightarrow \text{id} (A)$

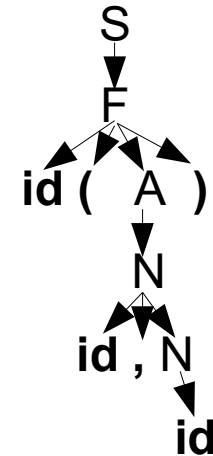
$A \rightarrow N$

$A \rightarrow \epsilon$

$N \rightarrow \text{id}$

$N \rightarrow \text{id} , N$

Input
 $\text{id} (\text{id} , \text{id})$



chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

$S \rightarrow \bullet F , 0$

$F \rightarrow \bullet \text{id} (A) , 0$

Closure on F

Massive Earley Example

Grammar

$S \rightarrow F$

$F \rightarrow \text{id} (A)$

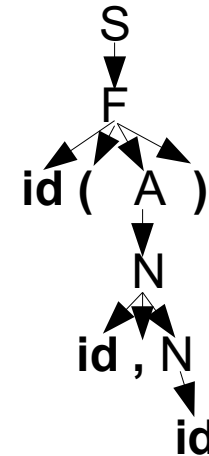
$A \rightarrow N$

$A \rightarrow \epsilon$

$N \rightarrow \text{id}$

$N \rightarrow \text{id} , N$

Input
 $\text{id} (\text{id} , \text{id})$



chart[0] chart[1] chart[2] chart[3] chart[4] chart[5] chart[6]

$S \rightarrow \bullet F , 0$

$F \rightarrow \text{id} \bullet (A) , 0$

$F \rightarrow \bullet \text{id} (A) , 0$

Shift on "id"

Massive Earley Example

Grammar

$S \rightarrow F$

$F \rightarrow \text{id} (A)$

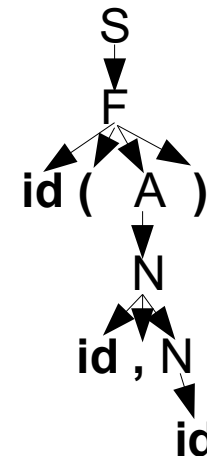
$A \rightarrow N$

$A \rightarrow \epsilon$

$N \rightarrow \text{id}$

$N \rightarrow \text{id} , N$

Input
 $\text{id} (\text{id} , \text{id})$



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$ $F \rightarrow \text{id} \bullet (A) , 0$ $F \rightarrow \text{id} (\bullet A) , 0$

$F \rightarrow \bullet \text{id} (A) , 0$

Shift on "("

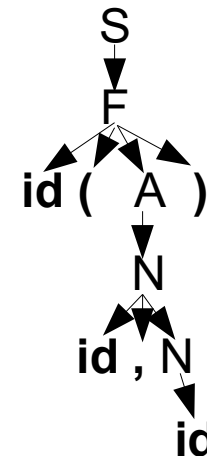
Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input

id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$ $F \rightarrow id \bullet (A) , 0$ $F \rightarrow id (\bullet A) , 0$

$F \rightarrow \bullet id (A) , 0$

$A \rightarrow \bullet N , 2$

$A \rightarrow \bullet , 2$

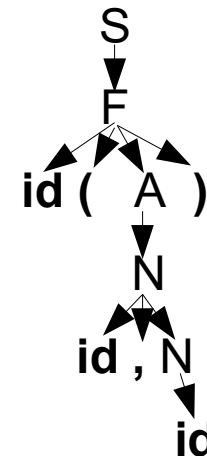
Closure on A

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$ $F \rightarrow id \bullet (A) , 0$ $F \rightarrow id (\bullet A) , 0$

$F \rightarrow \bullet id (A) , 0$

$A \rightarrow \bullet N , 2$

$A \rightarrow \bullet , 2$

$N \rightarrow \bullet id , 2$

$N \rightarrow \bullet id , N , 2$

Closure on N

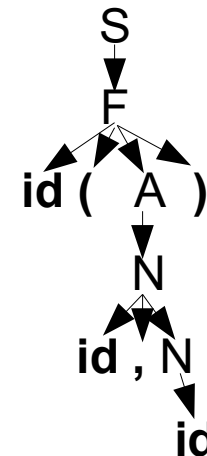
Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input

id (id , id)



chart[0] chart[1] chart[2] chart[3] chart[4] chart[5] chart[6]

$S \rightarrow \bullet F , 0$ $F \rightarrow id \bullet (A) , 0$ $F \rightarrow id (\bullet A) , 0$

$F \rightarrow \bullet id (A) , 0$

$A \rightarrow \bullet N , 2$

$A \rightarrow \bullet , 2$

$N \rightarrow \bullet id , 2$

$N \rightarrow \bullet id , N , 2$

$F \rightarrow id (A \bullet) , 0$

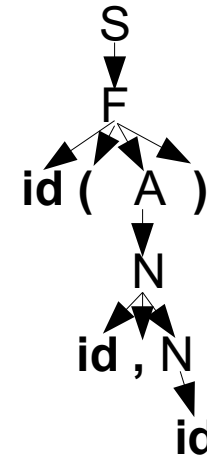
Reduce on A

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$
		$A \rightarrow \bullet , 2$	
		$N \rightarrow \bullet id , 2$	
		$N \rightarrow \bullet id , N , 2$	
		$F \rightarrow id (A \bullet) , 0$	

Shift on "id"

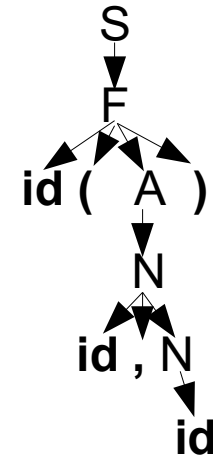
Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input

id (id , id)



chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$			
-------------------------------	--------------------------------------	--------------------------------------	--------------------------------	--	--	--

$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$			
--------------------------------------	--	-------------------------------	------------------------------------	--	--	--

		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$			
--	--	-----------------------------	-------------------------------	--	--	--

		$N \rightarrow \bullet id , 2$				
--	--	--------------------------------	--	--	--	--

		$N \rightarrow \bullet id , N , 2$				
--	--	------------------------------------	--	--	--	--

		$F \rightarrow id (A \bullet) , 0$				
--	--	--------------------------------------	--	--	--	--

Reduce
on N

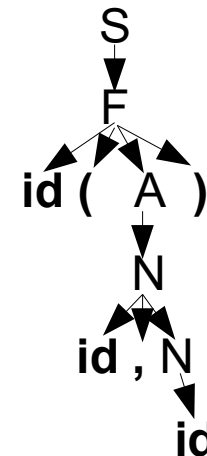
Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input

id (id , id)



chart[0] chart[1] chart[2] chart[3] chart[4] chart[5] chart[6]

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$
		$N \rightarrow \bullet id , N , 2$	
		$F \rightarrow id (A \bullet) , 0$	

Reduce
on A

Massive Earley Example

Grammar

$S \rightarrow F$

$F \rightarrow id (A)$

$A \rightarrow N$

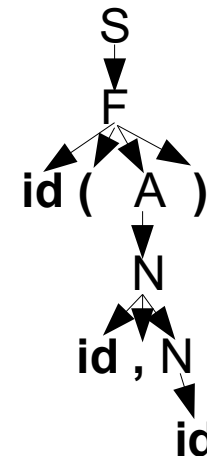
$A \rightarrow \epsilon$

$N \rightarrow id$

$N \rightarrow id , N$

Input

id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$		
-------------------------------	------------------------------------	--------------------------------------	--------------------------------	------------------------------------	--	--

$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$			
------------------------------------	--	-------------------------------	------------------------------------	--	--	--

$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$
-----------------------------	-------------------------------

$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$
--------------------------------	------------------------------------

$N \rightarrow \bullet id , N , 2$

$F \rightarrow id (A \bullet) , 0$

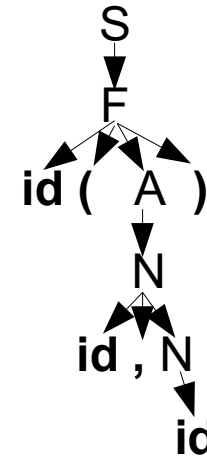
Shift on “,”

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$		
-------------------------------	--------------------------------------	--------------------------------------	--------------------------------	------------------------------------	--	--

$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$		
--------------------------------------	--	-------------------------------	------------------------------------	--------------------------------	--	--

		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$		
--	--	-----------------------------	-------------------------------	------------------------------------	--	--

		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$			
--	--	--------------------------------	--------------------------------------	--	--	--

		$N \rightarrow \bullet id , N , 2$				
--	--	------------------------------------	--	--	--	--

		$F \rightarrow id (A \bullet) , 0$				
--	--	--------------------------------------	--	--	--	--

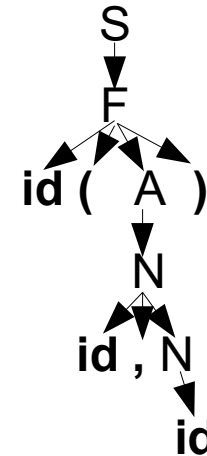
Closure
on N

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
----------	----------	----------	----------	----------	----------	----------

$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$		
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$			
		$N \rightarrow \bullet id , N , 2$				
		$F \rightarrow id (A \bullet) , 0$				

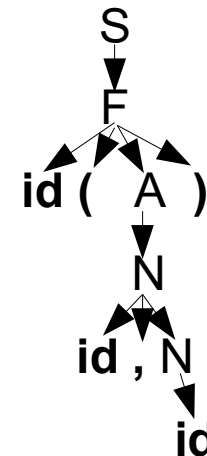
Shift
on "id"

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
 $id (id , id)$



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$			
		$N \rightarrow \bullet id , N , 2$				
		$F \rightarrow id (A \bullet) , 0$				

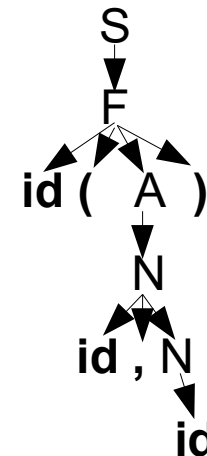
Reduce on N

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
 $id (id , id)$



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$		$A \rightarrow N \bullet , 2$	
		$N \rightarrow \bullet id , N , 2$				
		$F \rightarrow id (A \bullet) , 0$				

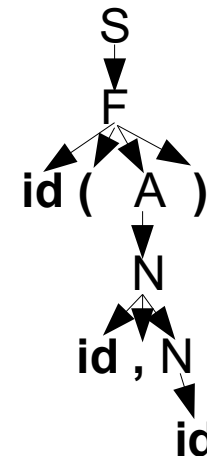
Reduce on N

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
 $id (id , id)$



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$		$A \rightarrow N \bullet , 2$	
		$N \rightarrow \bullet id , N , 2$			$F \rightarrow id (A \bullet) , 0$	
		$F \rightarrow id (A \bullet) , 0$				

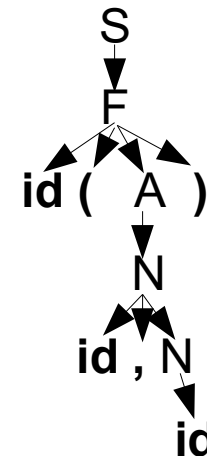
Reduce on A

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
 $id (id , id)$



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	$F \rightarrow id (A) \bullet , 0$
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$		$A \rightarrow N \bullet , 2$	
		$N \rightarrow \bullet id , N , 2$			$F \rightarrow id (A \bullet) , 0$	
		$F \rightarrow id (A \bullet) , 0$				

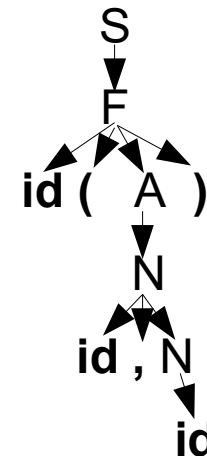
Shift
on ")"

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
 $id (id , id)$



chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	$F \rightarrow id (A) \bullet , 0$
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	$S \rightarrow F \bullet , 0$
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$		$A \rightarrow N \bullet , 2$	
		$N \rightarrow \bullet id , N , 2$			$F \rightarrow id (A \bullet) , 0$	
		$F \rightarrow id (A \bullet) , 0$				

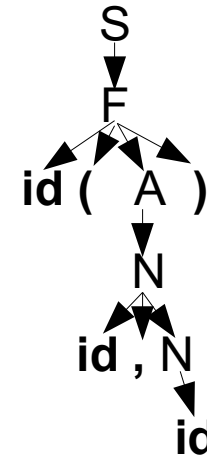
Reduce on F

Massive Earley Example

Grammar

$S \rightarrow F$
 $F \rightarrow id (A)$
 $A \rightarrow N$
 $A \rightarrow \epsilon$
 $N \rightarrow id$
 $N \rightarrow id , N$

Input
id (id , id)



id	(id	,	id)
----	---	----	---	----	---

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]
$S \rightarrow \bullet F , 0$	$F \rightarrow id \bullet (A) , 0$	$F \rightarrow id (\bullet A) , 0$	$N \rightarrow id \bullet , 2$	$N \rightarrow id , \bullet N , 2$	$N \rightarrow id \bullet , 4$	$F \rightarrow id (A) \bullet , 0$
$F \rightarrow \bullet id (A) , 0$		$A \rightarrow \bullet N , 2$	$N \rightarrow id \bullet , N , 2$	$N \rightarrow \bullet id , 4$	$N \rightarrow id \bullet , N , 4$	$S \rightarrow F \bullet , 0$
		$A \rightarrow \bullet , 2$	$A \rightarrow N \bullet , 2$	$N \rightarrow \bullet id , N , 4$	$N \rightarrow id , N \bullet , 2$	
		$N \rightarrow \bullet id , 2$	$F \rightarrow id (A \bullet) , 0$		$A \rightarrow N \bullet , 2$	
		$N \rightarrow \bullet id , N , 2$			$F \rightarrow id (A \bullet) , 0$	
		$F \rightarrow id (A \bullet) , 0$				

Accept!

Let's Implement It

- We'll use Python and Functional Programming
- Introducing **List Comprehensions**

```
>>> range(10)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> [ x*x for x in range(10) ]
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
>>> [ x for x in range(10) if x > 5 ]
```

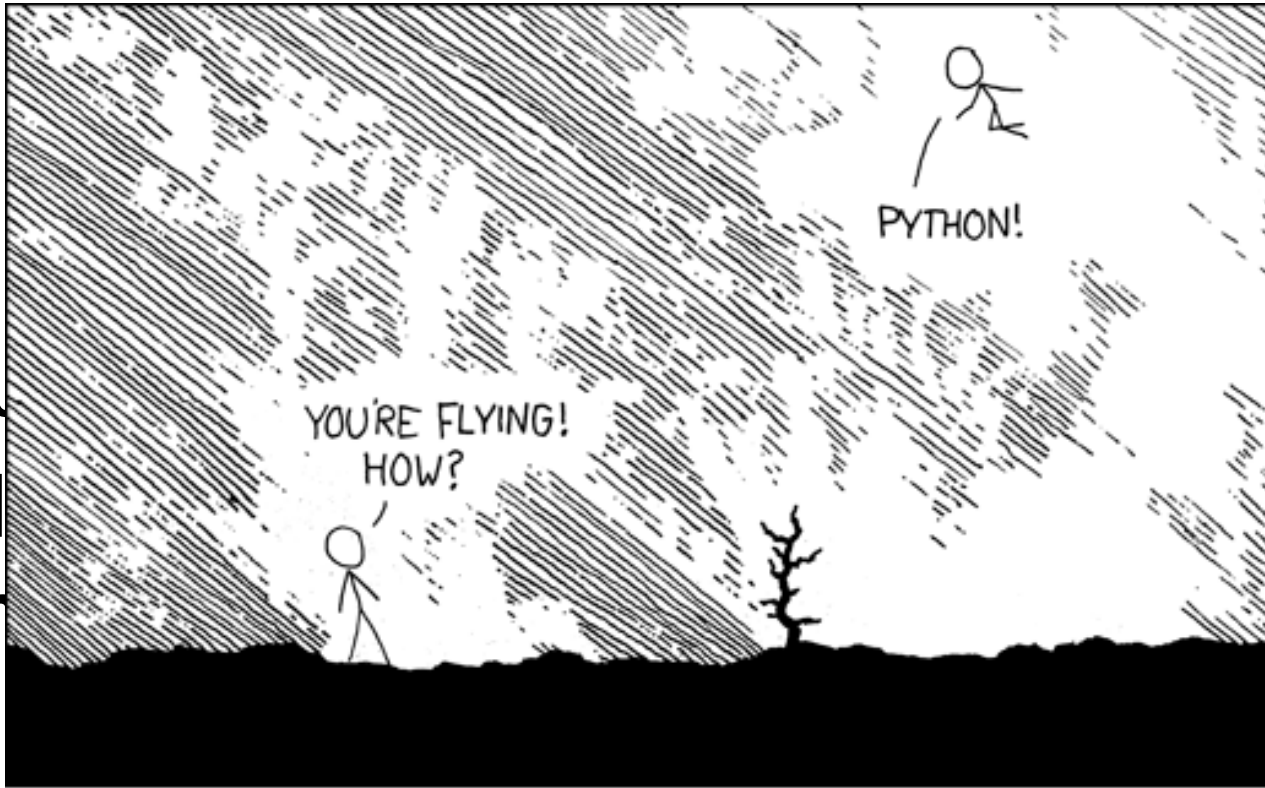
```
[6, 7, 8, 9]
```

```
>>> [ x*x for x in range(10) if x > 5 ]
```

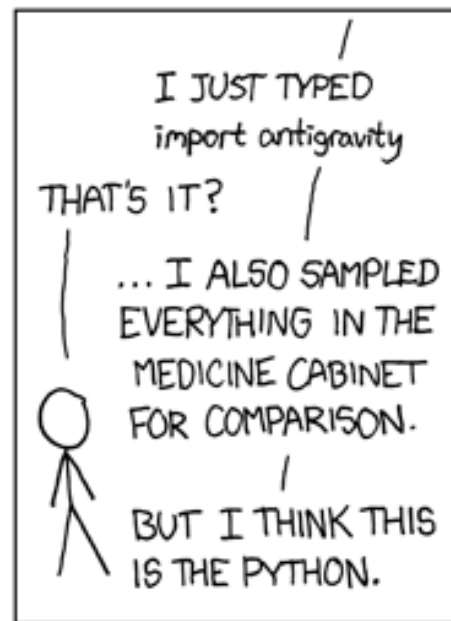
```
[36, 49, 64, 81]
```

- We'll u
- Introdu

```
>>> ran
[0, 1,
>>> [ x
[0, 1,
>>> [ x
[6, 7,
>>> [ x
[36, 49
```



mming



]] 5]

Data Structure Decisions

- For brevity, we'll use Lists and Tuples.
 - Not used: Named Tuples, Classes, etc.

```
grammar = [  
    ("S", ["F"]),  
    ("F", ["id", "(", "A", ")"]),  
    ("A", [ ] ),  
    ("A", ["N"] ),  
    ("N", ["id", ]),  
    ("N", ["id", ",", "N" ]),  
]  
tokens = [ "id" , "(", "id", ",", "id", ")" ]  
# X→ab.cd, i == ("X",["a","b"],["c","d"],i)
```


Initialization

By convention, the starting rule is
the first rule in the grammar.

```
start_rule = grammar[0]
```

The starting parse state is "S -> . abcd , from 0"
start_state = (start_rule[0], [], start_rule[1], 0)

The parsing chart is a one-dimensional array,
initially empty.

```
chart = {}
```

```
for i in range(len(tokens)+1): chart[i] = [ ]
```

Start by placing the starting state in chart[0].

```
chart[0] = [ start_state ]
```

Shift

```
# If chart[i] contains "X -> ab.cd , from j"
# and c is token[i] then add:
# "X -> abc.d , from j" to chart[i+1]
def shift(tokens, i, x, ab, cd, j):
    if cd <> [] and tokens[i] == cd[0]:
        c = cd[0]
        d = cd[1:]
        abc = ab + [c]
        new_chart_state = (x, abc, d, j)
        new_chart_index = i + 1
        return [(new_chart_index, new_chart_state)]
    else:
        return []
```

Closure

```
# If chart[i] contains "X -> ab.cd , from j":
#     and cd is not empty
#     and c is a non-terminal
#     and there is a grammar rule "c -> pqr"
# Then add:
#     "c -> . pqr , from i"
#     to chart[i]
def closure(grammar,i,x,ab,cd,j):
    return [ (i , (rule[0],[],rule[1],i)) \
              for rule in grammar \
              if cd <> [] and cd[0] == rule[0] ]
```

Reduction

```
# If chart[i] contains "X -> ab. , from j"
# (that is: cd is empty)
# and chart[j] contains "Y -> pq.Xr , from k"
# Then add
# "Y -> pqX.r , from k" to chart[i]
def reduction(chart,i,x,ab,cd,j):
    return [ (i, (jstate[0], jstate[1] + [x],
                  (jstate[2])[1:], jstate[3] ))
             for jstate in chart[j]
             if cd == [] and jstate[2] <> []
                and (jstate[2])[0] == x ]
```

Main Loop

```
# Step 2: Dynamic Programming
```

```
for i in range(len(tokens)):
```

```
    # Apply shift, closure and reduction until
```

```
    # no new parsing states are added to the chart.
```

```
    def apply_shift_closure_reduction():
```

```
        if any([add_to_chart(chart,
```

```
                    shift(tokens,i,x,ab,cd,j) +
```

```
                    closure(grammar,i,x,ab,cd,j) +
```

```
                    reduction(chart,i,x,ab,cd,j))
```

```
            for x, ab, cd, j in chart[i] ]):
```

```
                apply_shift_closure_reduction()
```

```
            # do it again if any changes
```

```
apply_shift_closure_reduction()
```

Example

```
grammar3 = [  
    ("S", ["E"]),  
    ("E", ["E", "-", "E" ]),  
    ("E", ["E", "+", "E" ]),  
    ("E", ["(", "E", ")"] ),  
    ("E", ["int"]),  
]
```

```
tokens3 = [ "int", "-", "int" ]
```

```
chart[0]
```

```
S -> . E , from 0
```

```
E -> . E - E , from 0
```

```
E -> . E + E , from 0
```

```
E -> . ( E ) , from 0
```

```
E -> . int , from 0
```

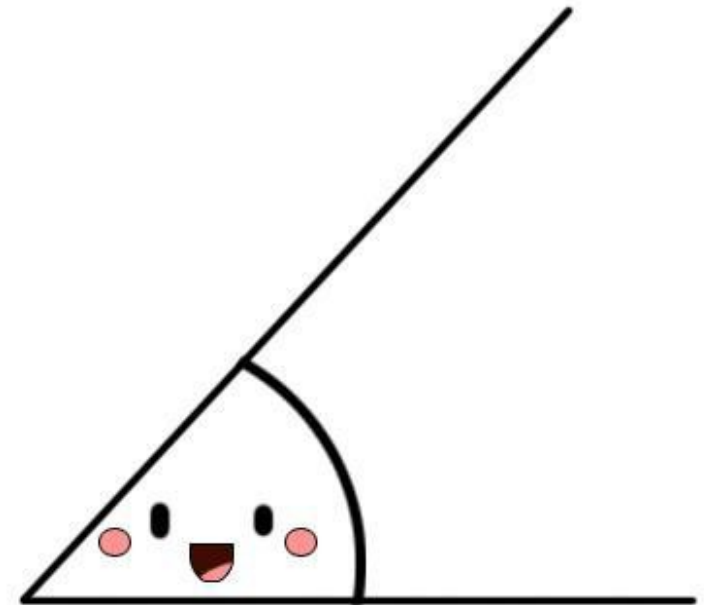
```
chart[1]
```

```
E -> int . , from 0
```

```
S -> E . , from 0
```

```
...
```

```
String Accepted: True
```



Acute angle

PA3 in JavaScript: parser.jison

%token PLUS MINUS INT

%left PLUS MINUS

%start program

%%

```
program: exp EOF { return $1; }  
      ;
```

```
exp: exp PLUS exp { $$ = ["plus_node", $1, $3]; }  
   | exp MINUS exp { $$ = ["minus_node", $1,$3]; }  
   | INT           { $$ = ["int_node",  
                           Number(yytext) ]; }  
   ;
```

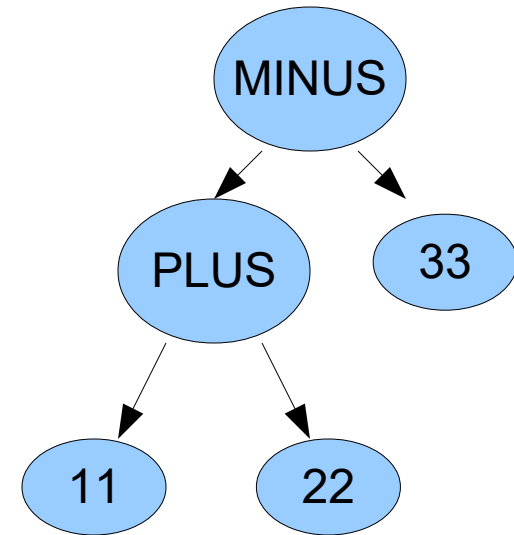
PA3 in JavaScript: main.js

```
var cl_lex = [  
  ['INT', "11"] ,  
  ['PLUS' ] ,  
  ['INT', "22"] ,  
  ['MINUS' ] ,  
  ['INT', "33"] ,  
  ['EOF' ] ,  
]  
var token_count = 0  
  
var parser =  
  require("./parser").parser;
```

```
parser.lexer = {  
  lex : function() {  
    var cl_lex_entry =  
      cl_lex[token_count++] ;  
    var token = cl_lex_entry[0] ;  
    var lexeme = cl_lex_entry[1] ;  
    parser.lexer.yytext = lexeme ;  
    return token;  
  },  
  setInput : function(str) { }  
}  
  
var final_ast = parser.parse("");  
  
console.log(final_ast);
```


PA3 in JavaScript Output:

```
$ node main.js  
[ 'minus_node',  
  [ 'plus_node',  
    [ 'int_node', 11 ],  
    [ 'int_node', 22 ]  
  ],  
  [ 'int_node', 33 ]  
]
```



PA3 Not Shown Here

- Reading in the .cl-lex file
- Handling line number information
- Printing out the AST in the desired format
- Adding parsing rules for whole classes and not just simple expressions
- Massive testing effort
 - diff vs. “cool --parse” requires “almost done”
- Dealing with ambiguity (“conflicts”)
 - Let's do this one now.

Conflicts

- Add “%token NEG” and “exp: NEG exp”.
- Oh noes:

Conflict in grammar: multiple actions possible when lookahead token is PLUS in state 8

- **reduce** by rule: `exp -> NEG exp`
- **shift** token (then go to state 6)

Conflict in grammar: multiple actions possible when lookahead token is MINUS in state 8

- **reduce** by rule: `exp -> NEG exp`
- **shift** token (then go to state 7)

States with conflicts:

State 8

<code>exp -> NEG exp .</code>	<code>#lookaheads= EOF PLUS MINUS</code>
<code>exp -> exp . PLUS exp</code>	
<code>exp -> exp . MINUS exp</code>	

Con

- Add “%token NEG” and
- Oh noes:

Conflict in grammar: multiple actions
PLUS in state 8

- **reduce** by rule: $\text{exp} \rightarrow \text{NEG exp}$
- **shift** token (then go to state 6)

Conflict in grammar: multiple actions
MINUS in state 8

- **reduce** by rule: $\text{exp} \rightarrow \text{NEG exp}$
- **shift** token (then go to state 7)

States with conflicts:

State 8

$\text{exp} \rightarrow \text{NEG exp} \cdot$ #lookahead

$\text{exp} \rightarrow \text{exp} \cdot \text{PLUS exp}$

$\text{exp} \rightarrow \text{exp} \cdot \text{MINUS exp}$



Conflict Interpretation

- So some table entry has all three:
 - $\text{exp} \rightarrow \text{NEG exp}$.
 - $\text{exp} \rightarrow \text{exp} . \text{PLUS exp}$
 - $\text{exp} \rightarrow \text{exp} . \text{MINUS exp}$
- What would the input have to look like to get to that table entry?



Internet Explorer

Question of the day: Which technological invention do you think has impacted our lives more - the telephone or the internet?

about a minute ago · Like · Comment



Billy

You know you can post Polls on facebook now, right?
IE, Always a little behind the times.

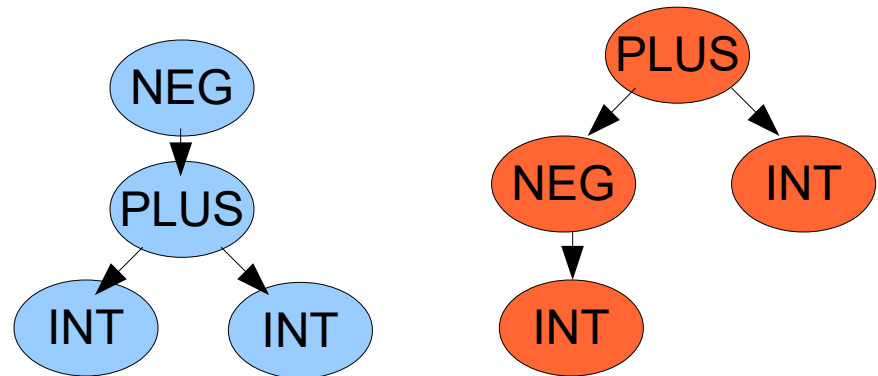
2 seconds ago · Like

Conflict Interpretation

- So some table entry has all three:
 - $\text{exp} \rightarrow \text{NEG exp} .$
 - $\text{exp} \rightarrow \text{exp} . \text{PLUS exp}$
 - $\text{exp} \rightarrow \text{exp} . \text{MINUS exp}$
- What would the input have to look like to get to that table entry?
 - $\text{NEG INT} . \text{PLUS INT}$

Conflict Interpretation

- So some table entry has all three:
 - $\text{exp} \rightarrow \text{NEG exp} .$
 - $\text{exp} \rightarrow \text{exp} . \text{PLUS exp}$
 - $\text{exp} \rightarrow \text{exp} . \text{MINUS exp}$
- What would the input have to look like to get to that table entry?
 - $\text{NEG INT} . \text{PLUS INT}$



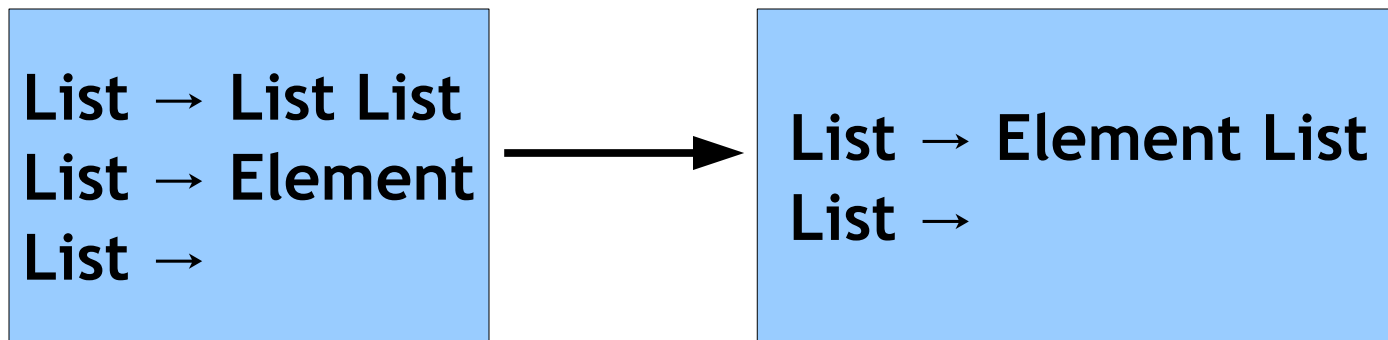
Conflict Solution

- Shift/Reduce

- Carefully specify precedence and associativity of operators (and sometimes of random tokens).
 - In last example, NEG has higher precedence than PLUS or MINUS.

- Reduce/Reduce

- Rewrite grammar to avoid gross ambiguity:



Homework

- Midterm 1 Next Week
- PS3 recommended for next Tuesday
- PA3 due next Thursday