

# Earley Parsing

## Informatics 2A: Lecture 19

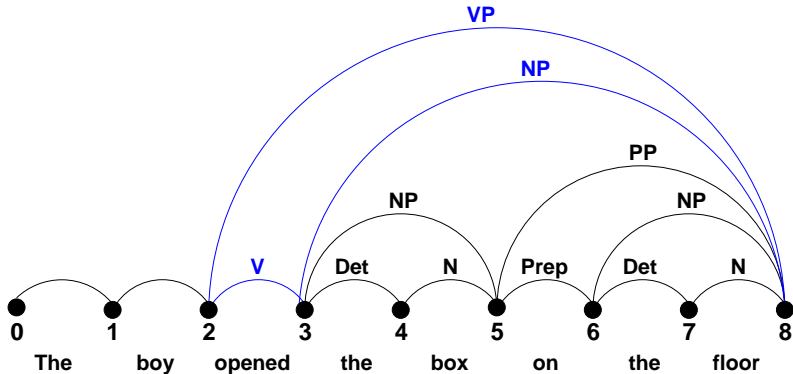
John Longley

All Hallows' Eve,  
31 October 2013

- 1 The CYK chart as a graph
  - What's wrong with CYK
  - Adding Prediction to the Chart
  
- 2 The Earley Parsing Algorithm
  - The PREDICTOR Operator
  - The SCANNER Operator
  - The COMPLETER Operator
  - Earley parsing: example
  - Comparing Earley and CYK

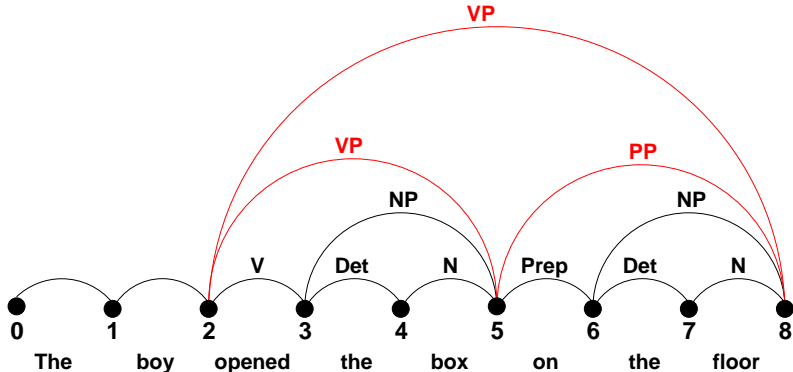
## Graph representation

The CYK chart can also be represented as a **graph**. E.g. for a certain grammar containing rules  $VP \rightarrow V NP$  and  $VP \rightarrow VP PP$ :



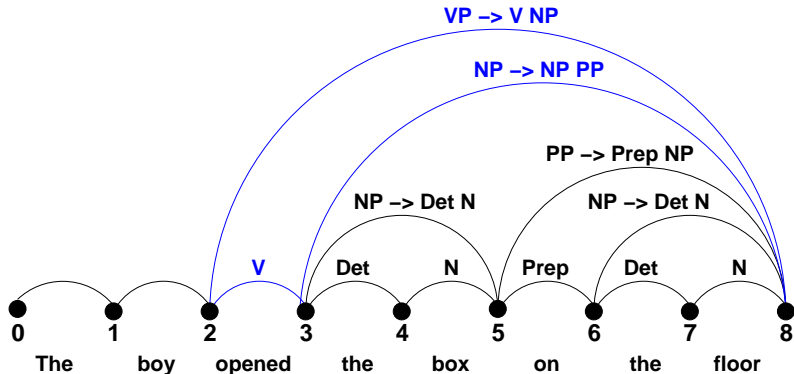
## Graph representation

An alternative analysis. Note we don't know which production the VP arc [2, 8] represents:  $VP \rightarrow V NP$  or  $VP \rightarrow VP PP$ .



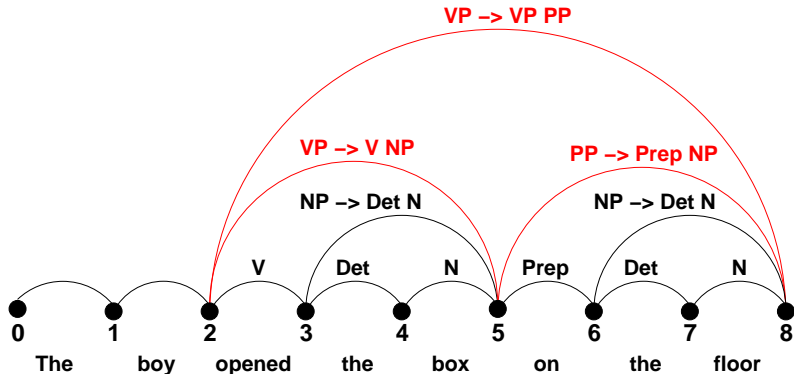
## CYK Chart entries

If the entire **production** were recorded, rather than just its LHS (ie, the constituent that it analyses), then we'd (usually) know.

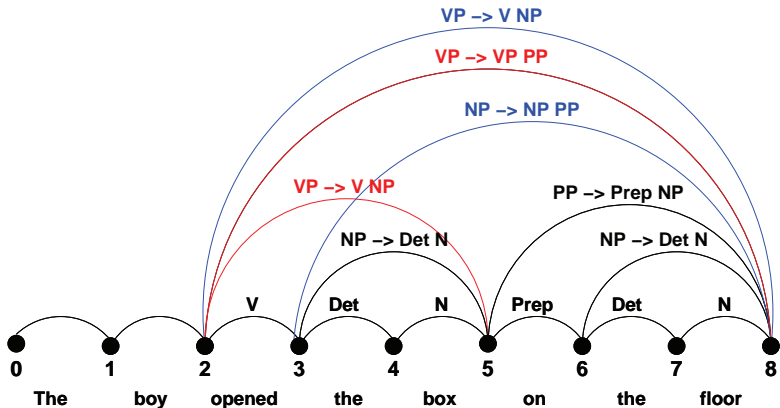


## CYK Chart entries

If the entire **production** were recorded, rather than just its LHS (ie, the constituent that it analyses), then we'd (usually) know.



## Chart entries: Both analyses



## CYK Chart entries

The CYK algorithm avoids redundant work by storing in a chart all the constituents it finds.

But it populates the table with **phantom constituents**, that don't form part of any complete parse. Can be a significant problem in long sentences.

The idea of the *Earley algorithm* is to avoid this, by only building constituents that are compatible with the input read so far.



## Earley Parsing

**Key idea:** as well as **completed productions** (ones whose entire RHS have been recognized), we also record **incomplete productions** (ones for which there may so far be only partial evidence).

- **Incomplete productions** (aka **incomplete constituents**) are effectively **predictions** about what might come next and what will be learned from finding it.
- **Incomplete constituents** can be represented using an extended form of production rule called a **dotted rule**, e.g.  
 $VP \rightarrow V \bullet NP.$
- The **dot** indicates how much of the RHS has already been found. The rest is a prediction of what is to come.

# Earley Parsing

- Allows arbitrary CFGs
- Top-down control
- Fills a table in a single sweep over the input
- Table entries represent:
  - **Completed** constituents and their locations
  - **In-progress** constituents
  - **Predicted** constituents

## States

The table entries are called states and are represented with **dotted-rules**.

$S \rightarrow \bullet VP$  [0,0]

A *VP* is **predicted** at the start of the sentence

$NP \rightarrow Det \bullet Nominal$  [1,2]

An NP is **in progress**; seen *Det*, *Nominal* is expected

$VP \rightarrow V NP \bullet$  [0,3]

A VP **has been found** starting at 0 and ending at 3

Once chart is populated there should be an *S* the final column that spans from 0 to *N* and is complete:  $S \rightarrow \alpha \bullet [0, N]$ . If that's the case you're done.

## Sketch of Earley Algorithm

- 1 **Predict** all the states you can upfront, working top-down from  $S$
- 2 For each word in the input:
  - 1 **Scan in** the word.
  - 2 **Complete** or extend existing states based on matches.
  - 3 Add new **predictions**.
- 3 When out of words, look at the chart to see if you have a winner.

The algorithm uses three basic operations to process states in the chart: PREDICTOR and COMPLETER add states to the chart entry being processed; SCANNER adds a state to the next chart entry.

## PREDICTOR

- Creates new states representing top-down expectations
- Applied to any state that has a non-terminal (other than a part-of-speech category) immediately to right of dot
- Application results in creation of one new state for each alternative expansion of that non-terminal
- **New states placed into same chart entry as generating state**

$S \rightarrow \bullet VP, [0,0]$		
$VP$	$\rightarrow \bullet$	$Verb, [0,0]$
$VP$	$\rightarrow \bullet$	$Verb NP, [0,0]$
$VP$	$\rightarrow \bullet$	$Verb NP PP, [0,0]$
$VP$	$\rightarrow \bullet$	$Verb PP, [0,0]$
$VP$	$\rightarrow \bullet$	$VP PP, [0,0]$

## SCANNER

- Applies to states with a part-of-speech category to right of dot
- Incorporates into chart a state corresponding to prediction of a word with particular part-of-speech
- **Creates new state from input state with dot advanced over predicted input category**
- Unlike CYK, only parts-of-speech of a word that are predicted by some existing state will enter the chart (top-down input)

$VP \rightarrow \bullet \textit{Verb NP}, [0,0]$

$VP \rightarrow \textit{book} \bullet, [0,1]$

## COMPLETER

- Applied to state when its dot has reached right end of the rule
- This means that parser has successfully discovered a particular grammatical category over some span of the input
- COMPLETER finds and advances all previously created states that were looking for this category at this position in input
- Creates states copying the older state, advancing dot over expected category, and installing new state in chart

$NP \rightarrow Det \text{ Nominal} \bullet, [1,3]$

finds state

$VP \rightarrow Verb \bullet NP, [0,1]$

finds state

$VP \rightarrow Verb \bullet NP PP, [0,1]$

## COMPLETER

- Applied to state when its dot has reached right end of the rule
- This means that parser has successfully discovered a particular grammatical category over some span of the input
- COMPLETER finds and advances all previously created states that were looking for this category at this position in input
- Creates states copying the older state, advancing dot over expected category, and installing new state in chart

$NP \rightarrow Det\ Nominal \bullet, [1,3]$

finds state

$VP \rightarrow Verb \bullet NP, [0,1]$

finds state

$VP \rightarrow Verb \bullet NP\ PP, [0,1]$

adds complete state

$VP \rightarrow Verb\ NP \bullet, [0,3]$

adds incomplete state

$VP \rightarrow Verb\ NP \bullet\ PP, [0,3]$



## Earley parsing: example

We will use the grammar to parse the sentence “*Book that flight*”.

### Grammar Rules

$S \rightarrow NP VP$

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal Noun$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb NP$

$VP \rightarrow Verb NP PP$

$VP \rightarrow Verb PP$

$VP \rightarrow VP PP$

$PP \rightarrow Preposition NP$

$Verb \rightarrow book|include|prefer$

$Noun \rightarrow book|flight|meal$

$Det \rightarrow that|this|these$

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet NP VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux NP VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP PP$	[0,0]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor



## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
<b>S18</b>	<b><i>VP</i> → <i>VP</i> • <i>PP</i></b>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
<b>S18</b>	<b><i>VP</i> → <i>VP</i> • <i>PP</i></b>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
S21	<i>NP</i> → • <i>Det Nominal</i>	[1,1]	Predictor
<b>S22</b>	<b><i>PP</i> → • <i>Prep NP</i></b>	[1,1]	Predictor

## Earley parsing: example[1]

state	rule	start/end	reason
S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
S13	<i>VP</i> → <i>Verb</i> •	[0,1]	Completer
S14	<i>VP</i> → <i>Verb</i> • <i>NP</i>	[0,1]	Completer
S15	<i>VP</i> → <i>Verb</i> • <i>NP PP</i>	[0,1]	Completer
S16	<i>VP</i> → <i>Verb</i> • <i>PP</i>	[0,1]	Completer
S17	<i>S</i> → <i>VP</i> •	[0,1]	Completer
S18	<i>VP</i> → <i>VP</i> • <i>PP</i>	[1,1]	Completer
S19	<i>NP</i> → • <i>Pronoun</i>	[1,1]	Predictor
S20	<i>NP</i> → • <i>Proper-Noun</i>	[1,1]	Predictor
<b>S21</b>	<b><i>NP</i> → • <i>Det Nominal</i></b>	[1,1]	Predictor
S22	<i>PP</i> → • <i>Prep NP</i>	[1,1]	Predictor

## Earley parsing: example[2]

state	rule	start/end	reason
S23	<i>Det</i> → <i>that</i> •	[1,2]	Scanner
S24	<i>NP</i> → <i>Det</i> • <i>Nominal</i>	[1,2]	Completer
S25	<i>Nominal</i> → • <i>Noun</i>	[2,2]	Predictor
S26	<i>Nominal</i> → • <i>Nominal Noun</i>	[2,2]	Predictor
S27	<i>Nominal</i> → • <i>Nominal PP</i>	[2,2]	Predictor



## Earley parsing: example[2]

state	rule	start/end	reason
S23	<i>Det</i> → <i>that</i> •	[1,2]	Scanner
S24	<i>NP</i> → <i>Det</i> • <i>Nominal</i>	[1,2]	Completer
S25	<i>Nominal</i> → • <i>Noun</i>	[2,2]	Predictor
S26	<i>Nominal</i> → • <i>Nominal Noun</i>	[2,2]	Predictor
S27	<i>Nominal</i> → • <i>Nominal PP</i>	[2,2]	Predictor

## Earley parsing: example[2]

state	rule	start/end	reason
S23	<i>Det</i> → <i>that</i> •	[1,2]	Scanner
S24	<i>NP</i> → <i>Det</i> • <i>Nominal</i>	[1,2]	Completer
S25	<i>Nominal</i> → • <i>Noun</i>	[2,2]	Predictor
S26	<i>Nominal</i> → • <i>Nominal Noun</i>	[2,2]	Predictor
S27	<i>Nominal</i> → • <i>Nominal PP</i>	[2,2]	Predictor

## Earley parsing: example[2]

state	rule	start/end	reason
S23	<i>Det</i> → <i>that</i> •	[1,2]	Scanner
S24	<i>NP</i> → <i>Det</i> • <i>Nominal</i>	[1,2]	Completer
S25	<i>Nominal</i> → • <i>Noun</i>	[2,2]	Predictor
S26	<i>Nominal</i> → • <i>Nominal Noun</i>	[2,2]	Predictor
S27	<i>Nominal</i> → • <i>Nominal PP</i>	[2,2]	Predictor

## Earley parsing: example[3]

state	rule	start/end	reason
S28	<i>Noun</i> → • <i>flight</i>	[2,3]	Scanner
S29	<i>Nominal</i> → <i>Noun</i> •	[2,3]	Completer
S30	<i>NP</i> → <i>Det Nominal</i> •	[1,3]	Completer
S31	<i>Nominal</i> → <i>Nominal</i> • <i>Noun</i>	[2,3]	Completer
S32	<i>Nominal</i> → <i>Nominal</i> • <i>PP</i>	[2,3]	Completer
S33	<i>VP</i> → <i>Verb NP</i> •	[0,3]	Completer
S34	<i>VP</i> → <i>Verb NP</i> • <i>PP</i>	[0,3]	Completer
S35	<i>PP</i> → <i>Prep</i> • <i>NP</i>	[3,3]	Predictor
S36	<i>S</i> → <i>VP</i> •	[0,3]	Completer
S37	<i>VP</i> → <i>VP</i> • <i>PP</i>	[0,3]	Completer

## Earley parsing: example[3]

state	rule	start/end	reason
S28	<i>Noun</i> → • <i>flight</i>	[2,3]	Scanner
S29	<i>Nominal</i> → <i>Noun</i> •	[2,3]	Completer
S30	<i>NP</i> → <i>Det Nominal</i> •	[1,3]	Completer
S31	<i>Nominal</i> → <i>Nominal</i> • <i>Noun</i>	[2,3]	Completer
S32	<i>Nominal</i> → <i>Nominal</i> • <i>PP</i>	[2,3]	Completer
S33	<i>VP</i> → <i>Verb NP</i> •	[0,3]	Completer
S34	<i>VP</i> → <i>Verb NP</i> • <i>PP</i>	[0,3]	Completer
S35	<i>PP</i> → <i>Prep</i> • <i>NP</i>	[3,3]	Predictor
S36	<i>S</i> → <i>VP</i> •	[0,3]	Completer
S37	<i>VP</i> → <i>VP</i> • <i>PP</i>	[0,3]	Completer

## Earley parsing: example[3]

state	rule	start/end	reason
S28	<i>Noun</i> → • <i>flight</i>	[2,3]	Scanner
S29	<i>Nominal</i> → <i>Noun</i> •	[2,3]	Completer
S30	<i>NP</i> → <i>Det Nominal</i> •	[1,3]	Completer
S31	<i>Nominal</i> → <i>Nominal</i> • <i>Noun</i>	[2,3]	Completer
S32	<i>Nominal</i> → <i>Nominal</i> • <i>PP</i>	[2,3]	Completer
S33	<i>VP</i> → <i>Verb NP</i> •	[0,3]	Completer
S34	<i>VP</i> → <i>Verb NP</i> • <i>PP</i>	[0,3]	Completer
S35	<i>PP</i> → <i>Prep</i> • <i>NP</i>	[3,3]	Predictor
S36	<i>S</i> → <i>VP</i> •	[0,3]	Completer
S37	<i>VP</i> → <i>VP</i> • <i>PP</i>	[0,3]	Completer

## Earley parsing: example[3]

state	rule	start/end	reason
S28	<i>Noun</i> → • <i>flight</i>	[2,3]	Scanner
S29	<i>Nominal</i> → <i>Noun</i> •	[2,3]	Completer
S30	<i>NP</i> → <i>Det Nominal</i> •	[1,3]	Completer
S31	<i>Nominal</i> → <i>Nominal</i> • <i>Noun</i>	[2,3]	Completer
S32	<i>Nominal</i> → <i>Nominal</i> • <i>PP</i>	[2,3]	Completer
S33	<i>VP</i> → <i>Verb NP</i> •	[0,3]	Completer
S34	<i>VP</i> → <i>Verb NP</i> • <i>PP</i>	[0,3]	Completer
S35	<i>PP</i> → <i>Prep</i> • <i>NP</i>	[3,3]	Predictor
S36	<i>S</i> → <i>VP</i> •	[0,3]	Completer
S37	<i>VP</i> → <i>VP</i> • <i>PP</i>	[0,3]	Completer

## The Earley Algorithm

---

**function** EARLEY-PARSE(*words*, *grammar*) **returns** *chart*

ENQUEUE( $(\gamma \rightarrow \bullet S, [0, 0])$ , *chart*[0])

**for**  $i \leftarrow$  **from** 0 **to** LENGTH(*words*) **do**

**for each** *state* **in** *chart*[*i*] **do**

**if** INCOMPLETE?(*state*) **and**

      NEXT-CAT(*state*) is not a part of speech **then**

      PREDICTOR(*state*)

**elseif** INCOMPLETE?(*state*) **and**

      NEXT-CAT(*state*) is a part of speech **then**

      SCANNER(*state*)

**else**

      COMPLETER(*state*)

**end**

**end**

**return**(*chart*)

---



## The Earley Algorithm

```
procedure PREDICTOR( $(A \rightarrow \alpha \bullet B \beta, [i, j])$ )  
  for each  $(B \rightarrow \gamma)$  in GRAMMAR-RULES-FOR( $B, grammar$ ) do  
    ENQUEUE( $(B \rightarrow \bullet \gamma, [j, j]), chart[j]$ )  
end  
  
procedure SCANNER( $(A \rightarrow \alpha \bullet B \beta, [i, j])$ )  
  if  $B \subset$  PARTS-OF-SPEECH( $word[j]$ ) then  
    ENQUEUE( $(B \rightarrow word[j], [j, j + 1]), chart[j + 1]$ )  
  
procedure COMPLETER( $(B \rightarrow \gamma \bullet, [j, k])$ )  
  for each  $(A \rightarrow \alpha \bullet B \beta, [i, j])$  in  $chart[j]$  do  
    ENQUEUE( $(A \rightarrow \alpha B \bullet \beta, [i, k]), chart[k]$ )  
end
```

## Parsing the Input

As with CYK we have formulated a **recognizer**. We can change it to a **parser** by adding backpointers so that each state knows where it came from.

Chart[1]	S12	<i>Verb</i> → <i>book</i> •	[0,1]	Scanner
Chart[2]	S23	<i>Det</i> → <i>that</i> •	[1,2]	Scanner
Chart[3]	S28	<i>Noun</i> → <i>flight</i> •	[2,3]	Scanner
	S29	<i>Nominal</i> → <i>Noun</i> •	[2,3]	(S28)
	S30	<i>NP</i> → <i>Det Nominal</i> •	[1,3]	(S23, S29)
	S33	<i>VP</i> → <i>Verb NP</i> •	[0,3]	(S12, S30)
	S36	<i>S</i> → <i>VP</i> •	[0,3]	(S33)

## Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

## Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

**Did we solve ambiguity?**

## Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

**Did we solve ambiguity?** Both CKY and Earley may result in multiple  $S$  structures for the  $[0, N]$  table entry. Of course, neither can tell us which one is 'right'.

## Summary

- The Earley algorithm uses dynamic programming to implement a **top-down** search strategy.
- Single left to right pass that fills chart with entries.
- Dotted rule represents progress in recognizing RHS of rule.
- Algorithm always moves forward, never backtracks to previous chart entry, once it has moved on.
- States are processed using PREDICTOR, COMPLETER, SCANNER operations.

**Reading:** Same as for Lecture 17

**Next lecture:** Resolving ambiguity using statistical parsing.