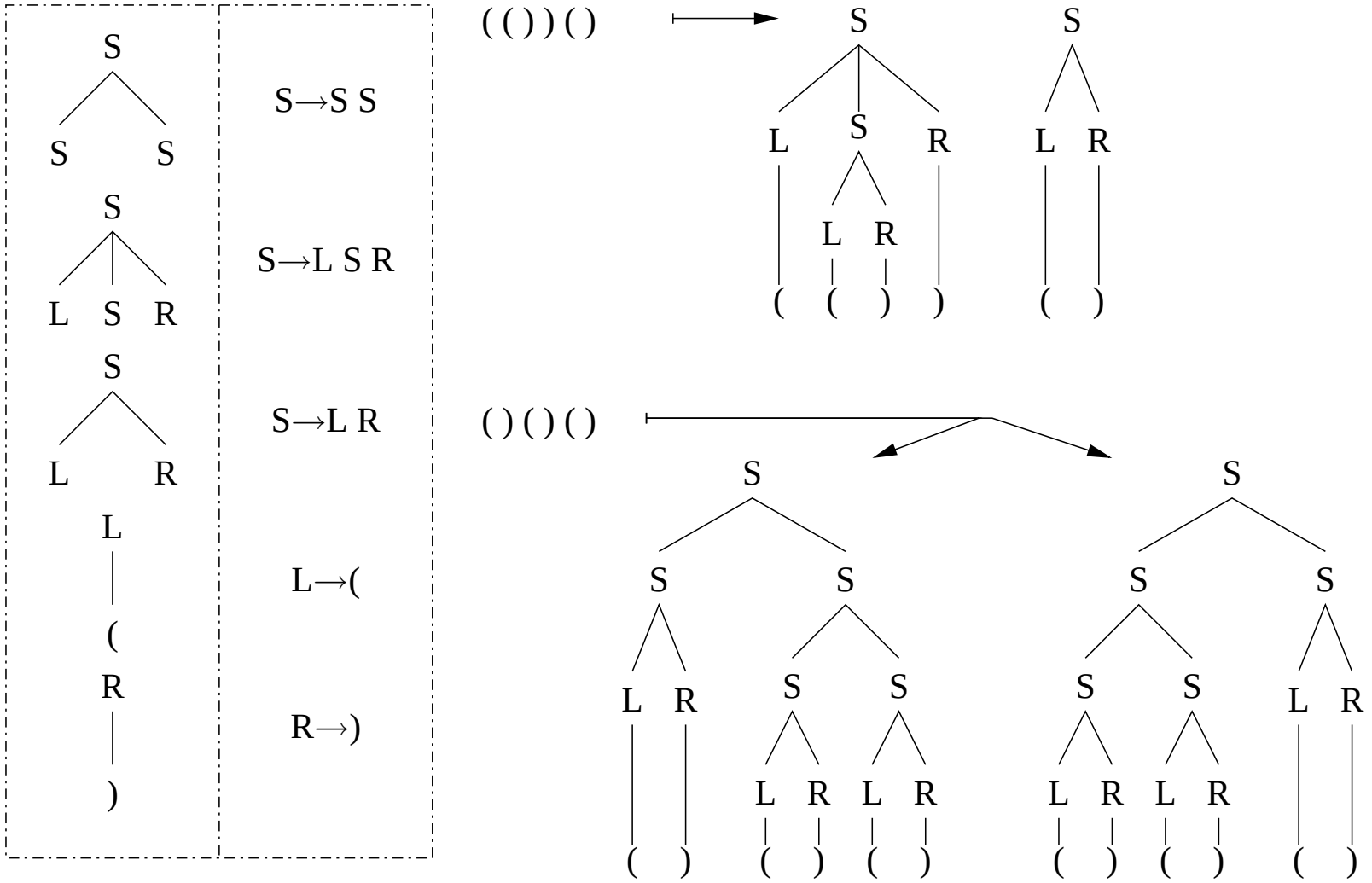
A Parallel Extension of Earley's Parsing Algorithm

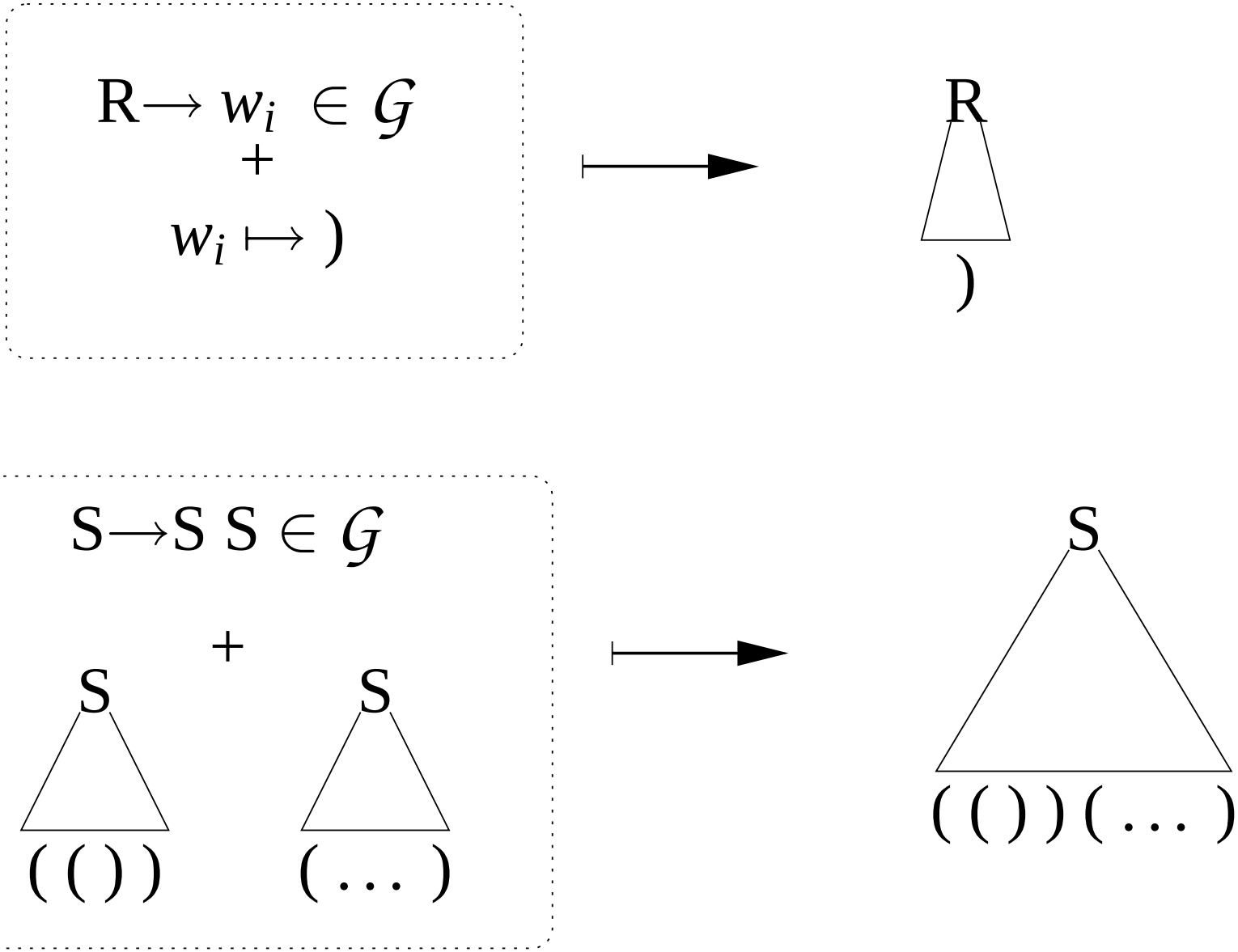
Greg Sandstrom

CS Senior Seminar

What is parsing?



CKY Algorithm as Dynamic Programming



Inference Rules for CKY Algorithm

Item: $[A, i, j]$ $\quad A \Rightarrow w_i \ldots w_{j-1}$

Rules:

Scan $\quad [A, i, i+1] \quad A \rightarrow w_i$

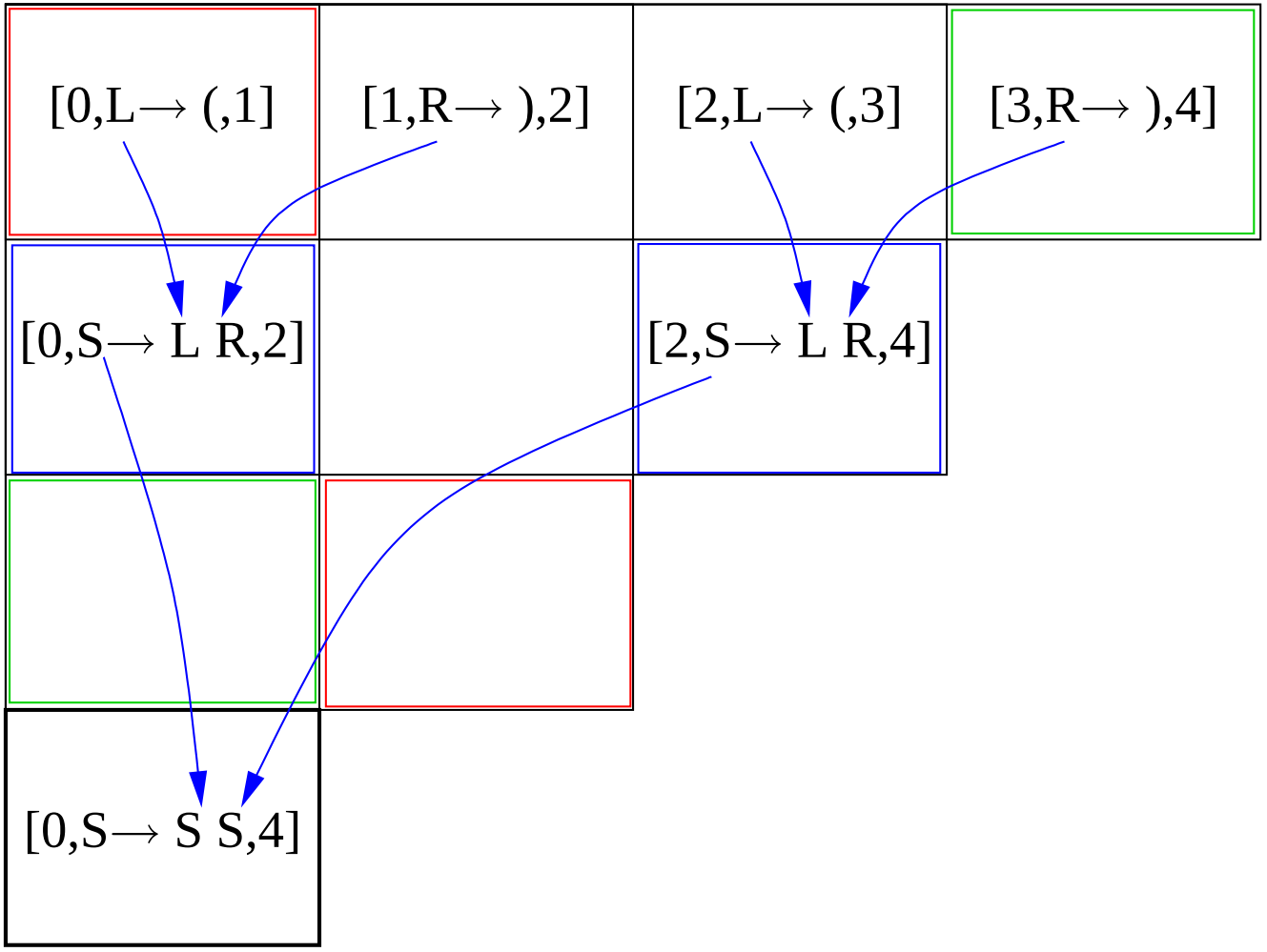
Complete $\quad \dfrac{[B, i, k] \; [C, k, j]}{[A, i, j]} \quad A \rightarrow BC \in \mathcal{G}$

$\mathcal{G}$:
Sentence $\rightarrow$ *NP Verb*
NP $\rightarrow$ *Det Noun*
Det $\rightarrow$ The
Noun $\rightarrow$ Man
Verb $\rightarrow$ Runs

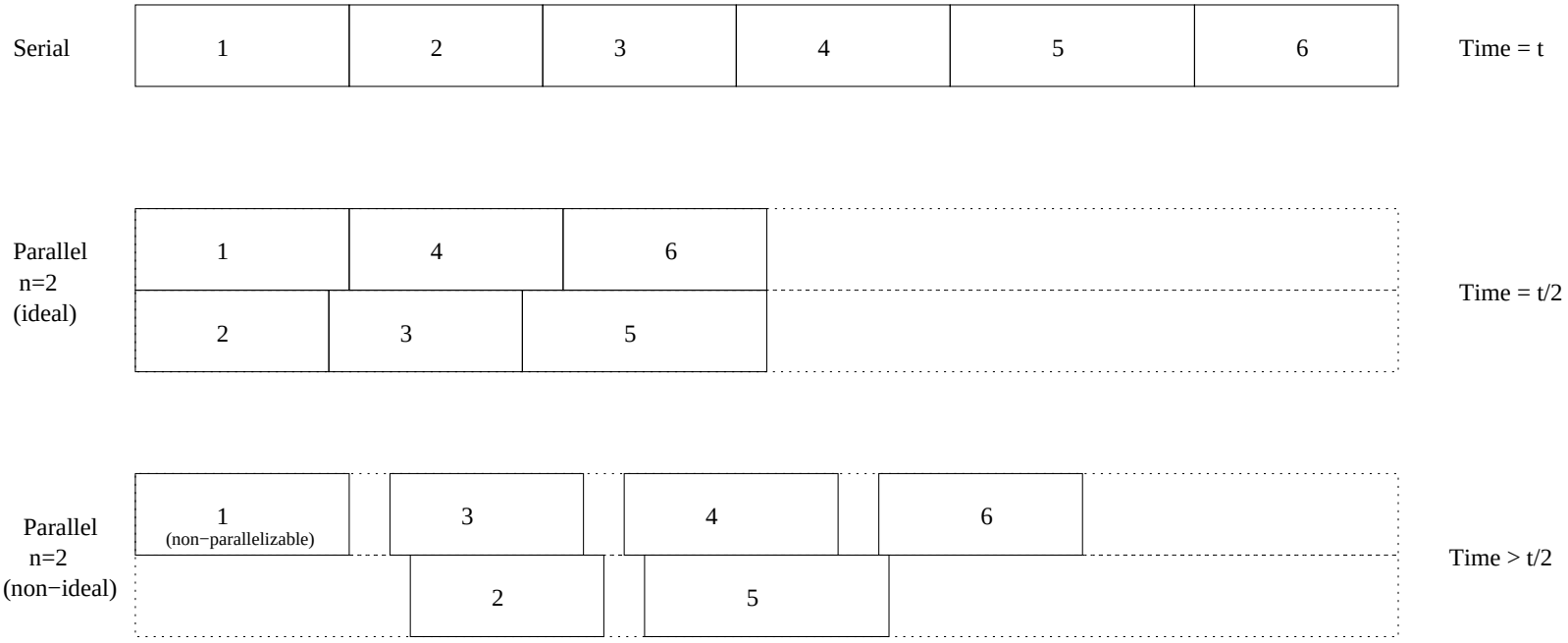
$$\begin{array}{rc} & \underline{\text{The}} \qquad \underline{\text{Man}} \qquad \underline{\text{Runs}} \\ \text{Scan} & [\mathit{Det}, 0, 1] \quad [\mathit{Noun}, 1, 2] \quad [\mathit{Verb}, 2, 3] \\ \text{Complete} & \underline{\qquad\qquad [\mathit{NP}, 0, 2] \qquad\qquad} \\ \text{Complete} & \underline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad} \\ & [\mathit{Sentence}, 0, 3] \end{array}$$

CKY Table

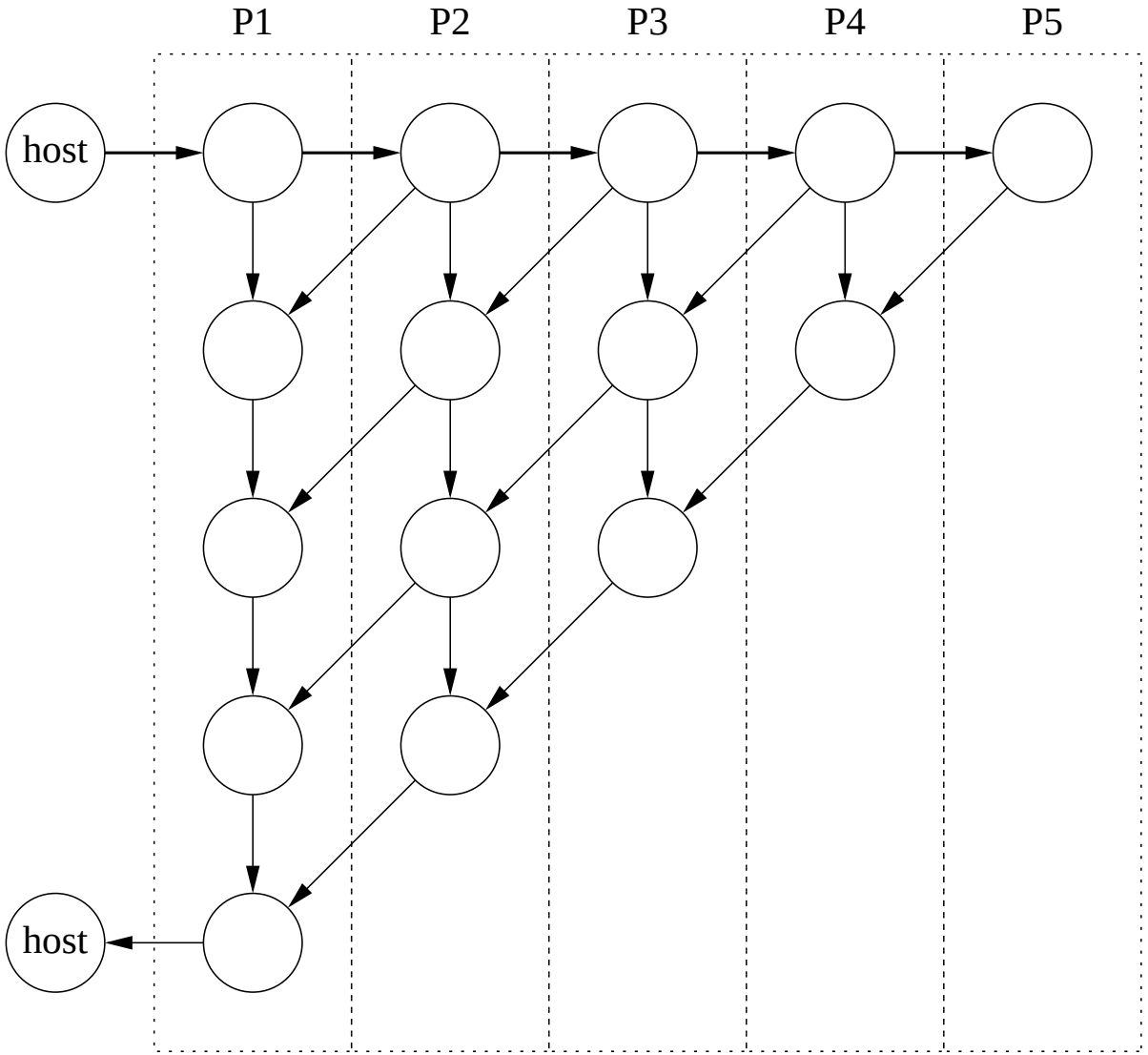
( ) ( )



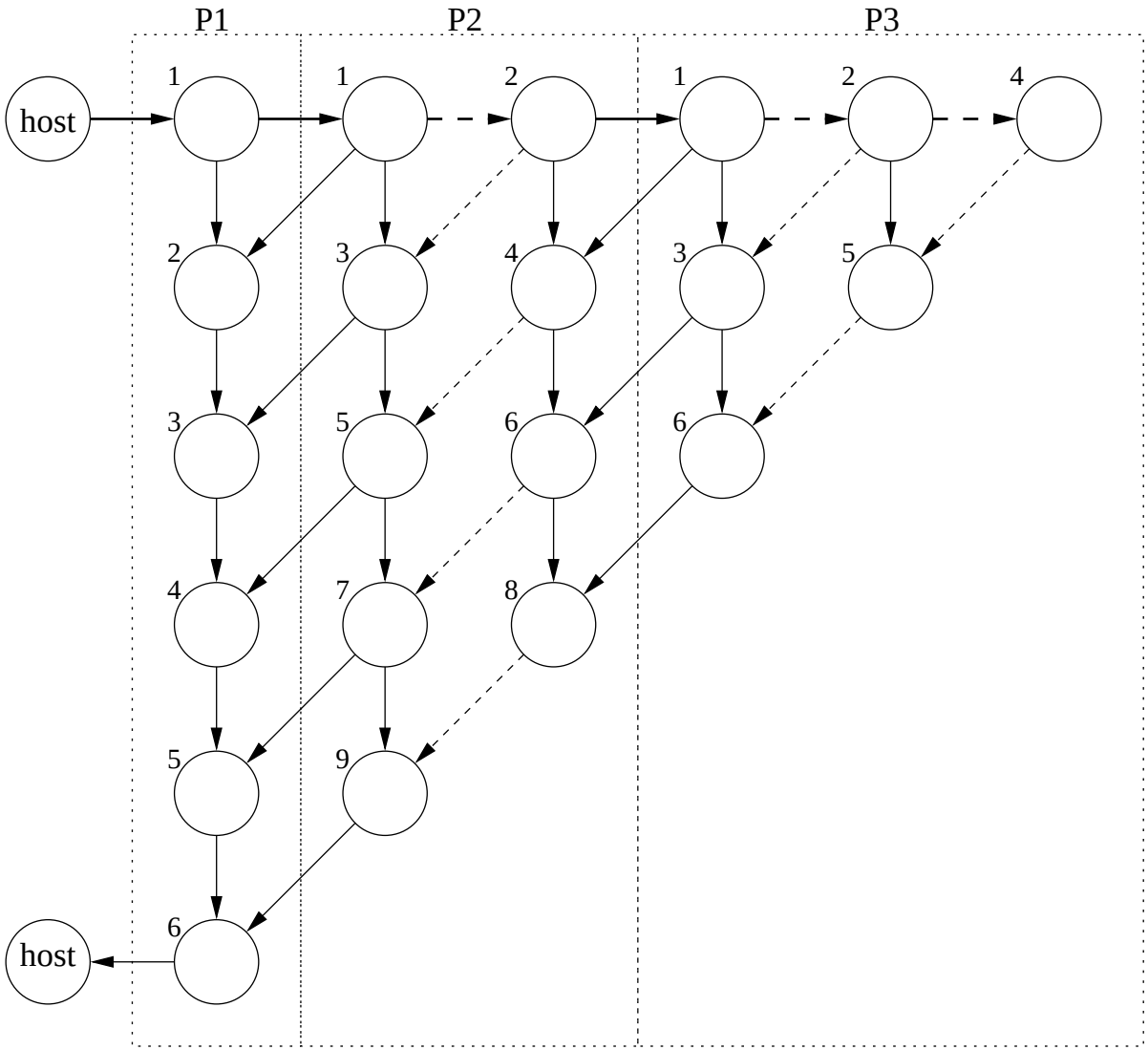
Parallelization



Hill & Wayne's Original Parallel CKY Algorithm



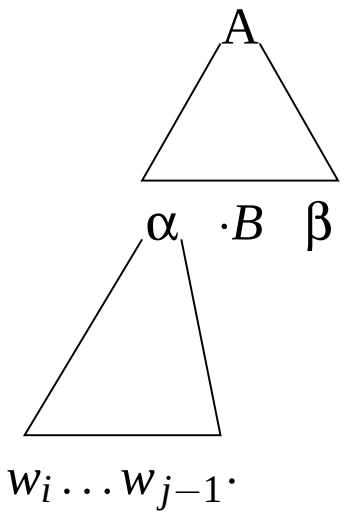
Hill & Wayne's Improved Parallel CKY Algorithm



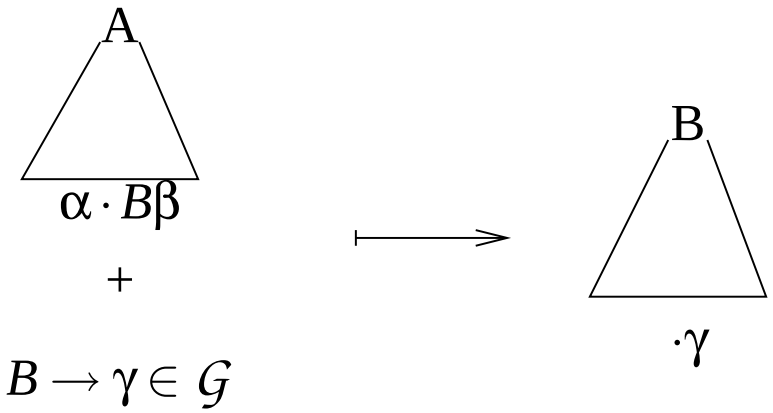
Earley's Algorithm

The dotted rule:

$[i, A \rightarrow \alpha \cdot B\beta, j]$



Prediction:



Inference Rules For Earley's Algorithm

Item: $[i, A \rightarrow \alpha \cdot B\beta, j]$ $\quad A \rightarrow \alpha B\beta \in \mathcal{G}$

$\alpha \Rightarrow w_i \ldots w_{j-1}$

Rules:

Scan $\quad \dfrac{[i, A \rightarrow \alpha \cdot a\beta, j]}{[i, A \rightarrow \alpha a \cdot \beta, j+1]} \quad a = w_j$

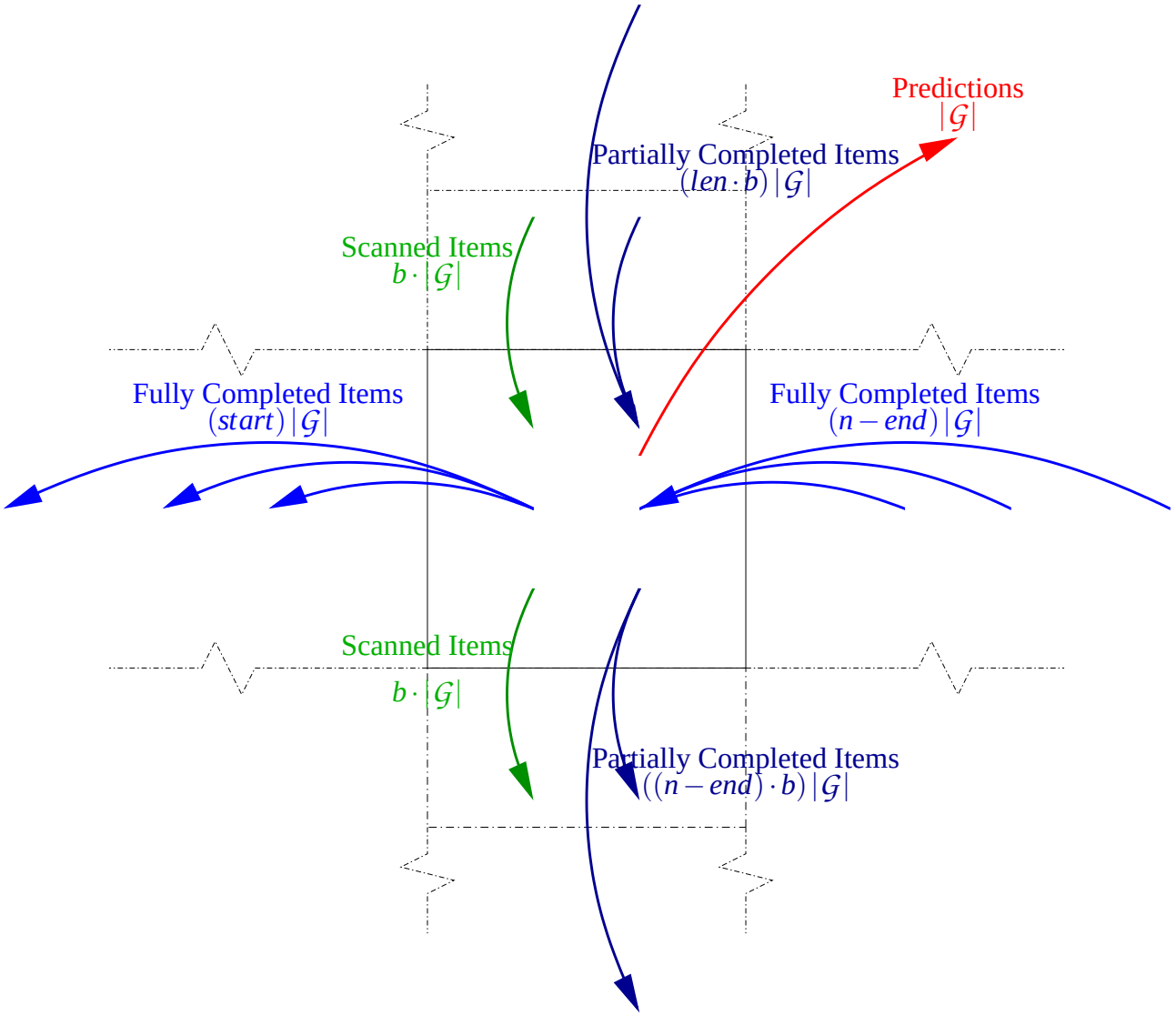
Predict $\quad \dfrac{[i, A \rightarrow \alpha \cdot C\beta, j]}{[j, C \rightarrow \cdot\gamma, j]} \quad C \rightarrow \gamma \in \mathcal{G}$

Complete $\quad \dfrac{[i, A \rightarrow \alpha \cdot C\beta, k]\ [k, C \rightarrow \gamma\cdot, j]}{[i, A \rightarrow \alpha C \cdot \beta, j]} \quad i \leq k \leq j \leq n$

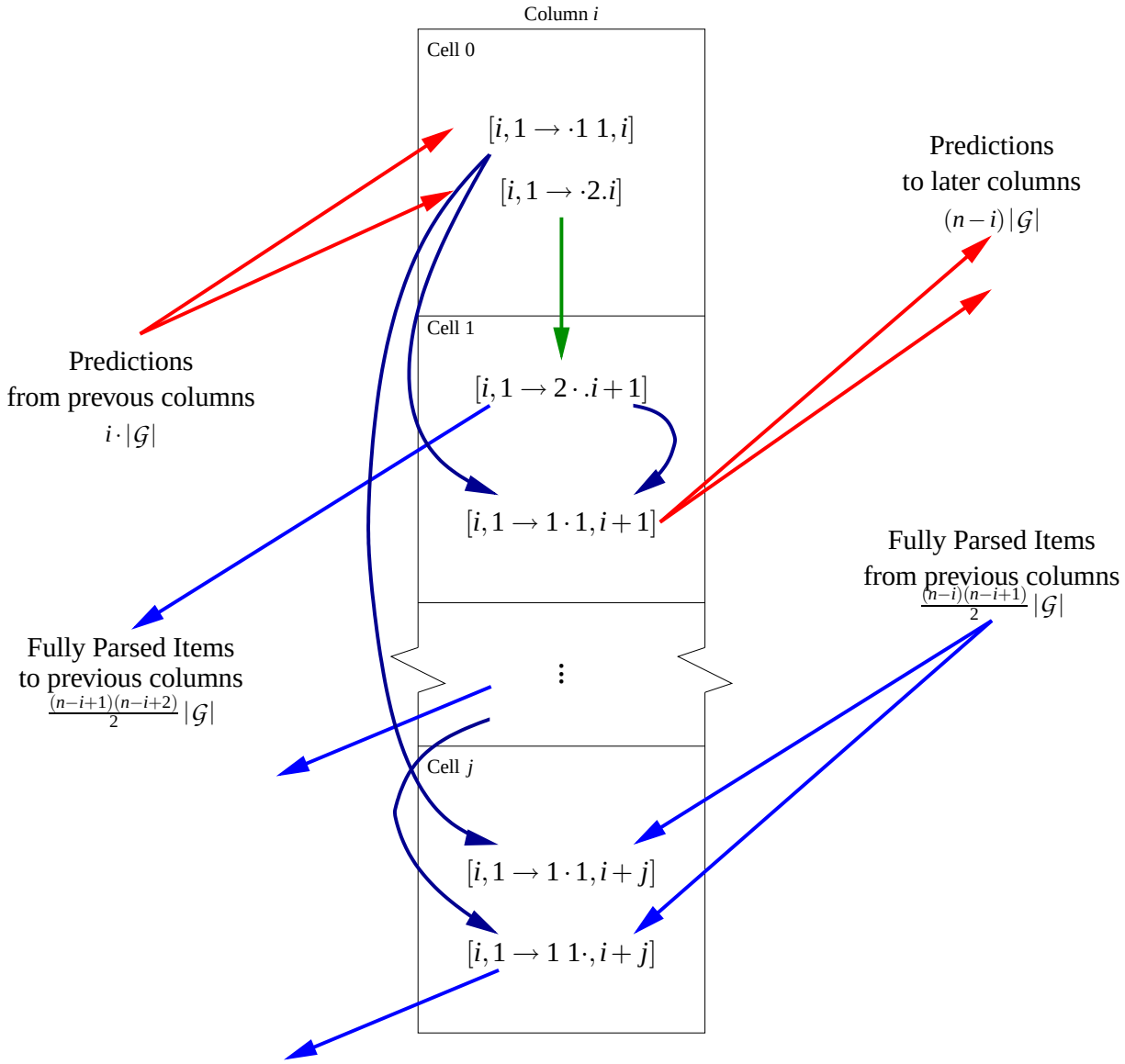
Earley Table

(	)	(	)
[0,S→ · S S,0] [0,S→ · L S R,0] [0,S→ · L R,0] [0,L→ · (,0]	[1,S→ · S S,1] [1,S→ · L S R,1] [1,S→ · L R,1] [1,R→ ·),1]	[2,S→ · S S,2] [2,S→ · L S R,2] [2,S→ · L R,2] [2,L→ · (,2]	[3,S→ · S S,3] [3,S→ · L S R,3] [3,S→ · L R,3] [3,R→ ·),3]
[0,L→(·, 1] [0,S→L· S R,1] [0,S→L· R,1]	[1,R→)·, 2]	[2,L→(·, 3] [2,S→L· S R,3] [2,S→L· R,3]	[3,R→)·, 4]
[0,S→L R·,2] [0,S→ · S S,2]		[2,S→L R·,4] [2,S→ · S S,4]	
[0,S→ S S,4]			

Worst Case Message Analysis



Per-Column Parallel Earley Communications Map



Performance Analysis for Fully Ambiguous Gramar

Length=16

Processors	Runtime	Speedup	Efficiency
1	1.73 s	1	100%
2	1.49 s	1.16	57.95%
4	1.21 s	1.43	35.76%
8	0.8 s	2.17	27.13%
16	0.79 s	2.19	13.68%

Length=32

Processors	Runtime	Speedup	Efficiency
1	7.99 s	1	100%
2	4.98 s	1.61	80.25%
4	2.78 s	2.88	71.97%
8	1.56 s	5.11	63.91%
16	1.03 s	7.77	48.55%

Length=48

2	11.31 s	1	100%[1]
4	5.92 s	1.91	95.52%
8	3.22 s	3.51	87.84%
16	2.1 s	5.38	67.21%

Length=64

2	31.74 s	1	100%[1]
4	15.58 s	2.04	101.84%
8	8.45 s	3.76	93.91%
16	4.66 s	6.81	85.18%

Length=128

4	916.75 s	1	100%[1]
8	464.23 s	1.97	98.84%
16	232.6 s	3.94	98.53%

footnote[1]: speedup and effi ciency based on least processors tested.

Possiblilities for Future Work

- Reduce communication overhead to communication payload ratio.
- Group collumns together as per Hill & Wayne.