Overview

- Earley algorithm
  - Predictor
  - Scanner
  - Completer

The Earley Algorithm

- Solve three problems afflicting standard bottom-up or top-down parsers
- Dynamic programming approach:
  - Systematically fill in tables of solutions to sub-problems.
  - When complete, the tables contain the solution to all sub-problems needed to solve the problem as a whole.
  - Reducing an exponential-time problem to a polynomial-time one by eliminating the repetitive solution of sub-problems inherently in backtracking approaches
  - $O(N^3)$, where N is the number of words in the input

The Earley Algorithm

- The core of the Earley is a single left-to-right pass that fills an array called a **chart** that has *N+1* entries
- Each possible subtree is represented only once and thus can be shared by all the parses that need it.
- Each entry in the chart is a list of:
  - a subtree corresponding to a single grammar rule
  - information about the progress made in completing this subtree
  - position of the subtree with respect to the input

The Earley Algorithm

- Use a ***dot*** within the right hand side of a state's grammar rule to *indicate the progress* made in it – **dotted rule**
- A state's position with respect to the input will be represented by two numbers indicating *where the state begins* and *where its dot lies*
- *Book that flight*

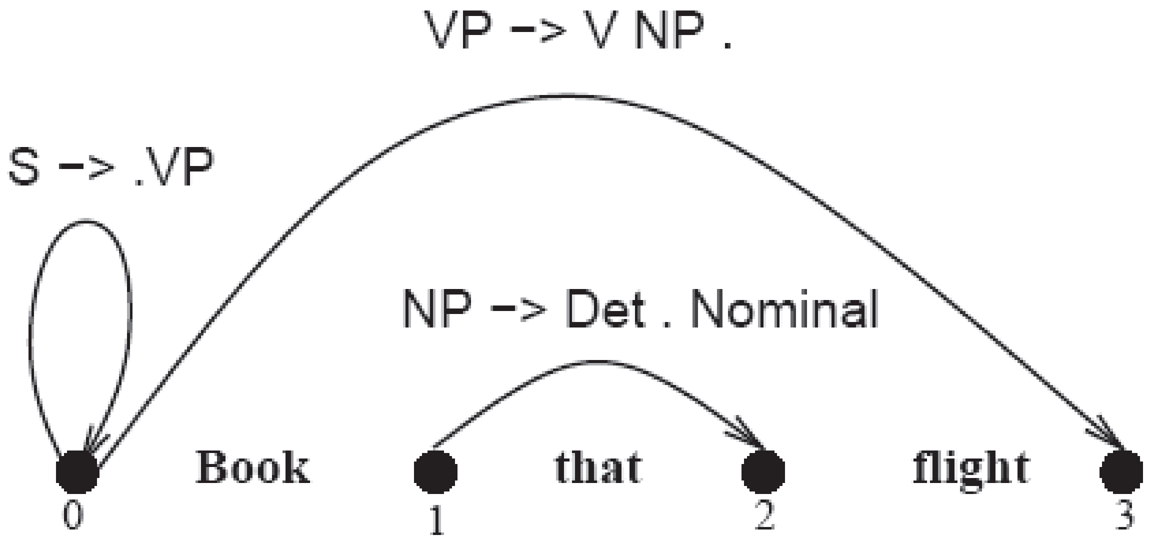
$S \rightarrow \bullet VP,\ [0,0]$	- A VP is *predicted*
$NP \rightarrow Det \bullet Nominal,\ [1,2]$	- An NP is *in progress*
$VP \rightarrow V\ NP \bullet,\ [0,3]$	- A VP is *found*

The Earley Algorithm

Example states in parsing *Book that flight.*

- S → · VP, [0,0]
  - the first 0 indicates that the constituent begins at the start of the input
  - the second 0 indicates that the dot is here as well, and thus indicates a top-down prediction
- NP → Det · Nominal, [1,2]
  - the NP begins at position 1
  - the dot is at position 2
  - Det has thus been successfully parsed
  - Nominal is thus predicted next
- VP → V NP ·, [0,3]
  - a successful VP parse of the entire input

Graphical Representation



The Earley Algorithm

- The fundamental operation is to march through the *N+1* sets of states in the chart in a left-to-right fashion
- At each step, one of three operators is applied to each state → results in the addition of new states to current or next in the chart
- States are *never removed* and algorithm *never backtracks* to a previous chart entry once it has moved on
- The state $S \rightarrow \boldsymbol{\alpha}\ \bullet$, [0,N] indicates a successful parse
- The three operators are:
  - Predictor, Completer – add states to the chart entry being processed
  - Scanner – adds a state to the next chart entry

Predictor

- To create new states representing top-down *expectations*
- Applied to any state that has a non-terminal to the right of the dot that is not a part-of-speech category
- Results in new states for each alternative expansion of that non-terminal
- These new states are placed into the same chart entry

$S \rightarrow \bullet VP, [0,0]$ - *predictor applied to non-terminal*

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

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

Scanner

- When a state has a part-of-speech category to the right of the dot, the scanner is called to examine the input and incorporate a state into the chart
- A new state is created with the dot advanced over the predicted input category
- Earley parser uses input to disambiguate pos ambiguities

$VP \rightarrow \bullet\, Verb\ NP, [0,0]$	- *scanner applied to pos*
$Verb \rightarrow book$	- scanner notes that *book* can be a verb
$VP \rightarrow Verb\, \bullet\, NP, [0,1]$	- a new state added to next chart entry

Completer

- The Completer applied when its *dot has reached the right end* of the rule
- Advance all previously created states
- New states are then created by copying the older state
- Advance the dot over the expected category and install the new state in the current chart entry

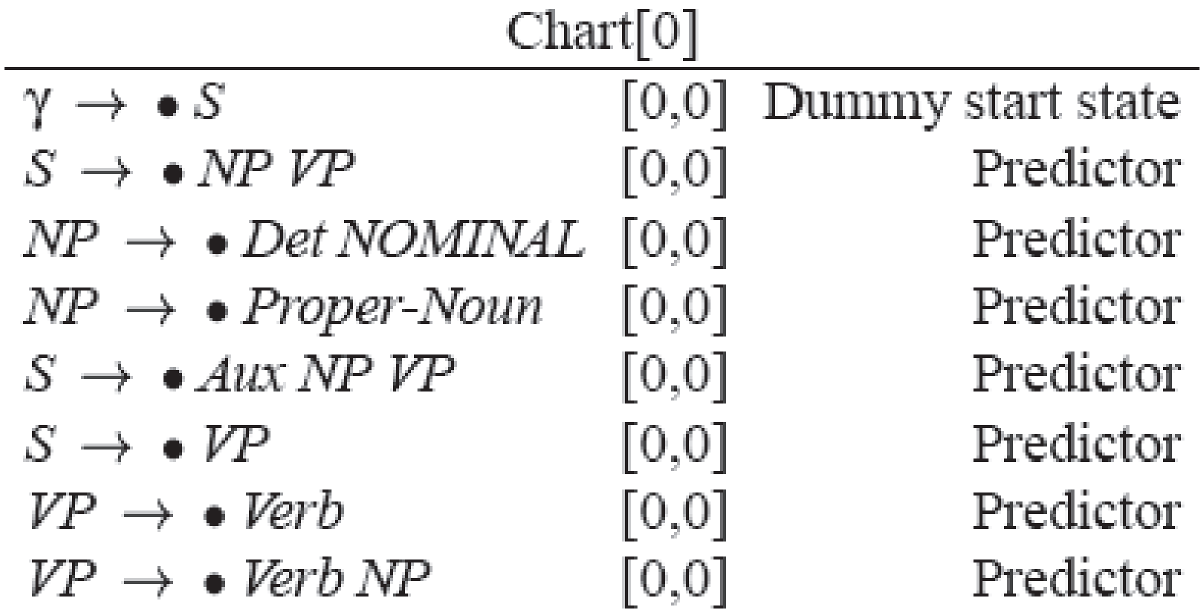
$NP \rightarrow$ *Det Nominal* •, [1,3]

completer looks for states ending at 1 expecting an NP

$VP \rightarrow$ *Verb* • *NP*, [0,1] - a state created by scanner

$VP \rightarrow$ *Verb NP* •, [0,3] - addition of new complete state

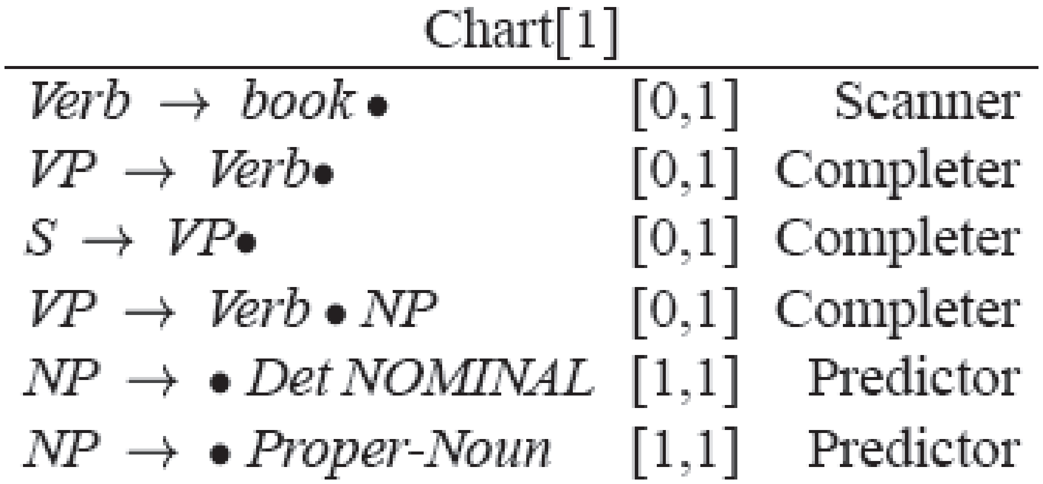
An Example



Chart[0]

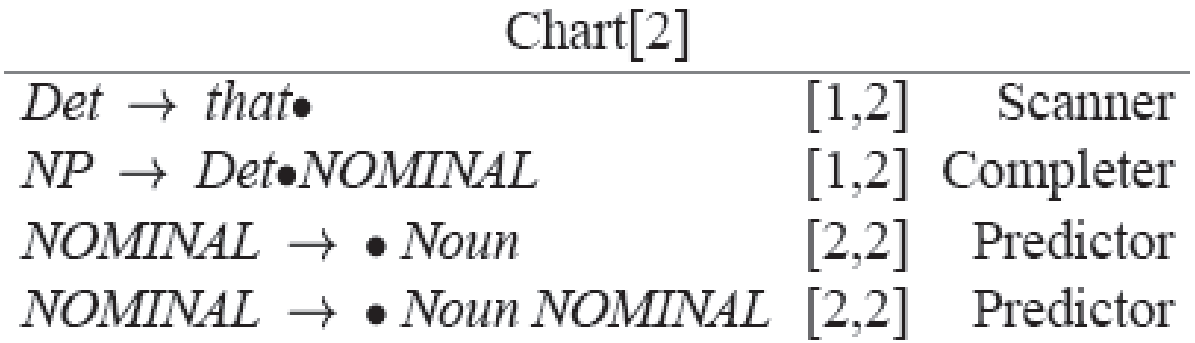
$\gamma \rightarrow \bullet S$	[0,0]	Dummy start state
$S \rightarrow \bullet NP\ VP$	[0,0]	Predictor
$NP \rightarrow \bullet Det\ NOMINAL$	[0,0]	Predictor
$NP \rightarrow \bullet Proper\text{-}Noun$	[0,0]	Predictor
$S \rightarrow \bullet Aux\ NP\ VP$	[0,0]	Predictor
$S \rightarrow \bullet VP$	[0,0]	Predictor
$VP \rightarrow \bullet Verb$	[0,0]	Predictor
$VP \rightarrow \bullet Verb\ NP$	[0,0]	Predictor

An Example



Chart[1]

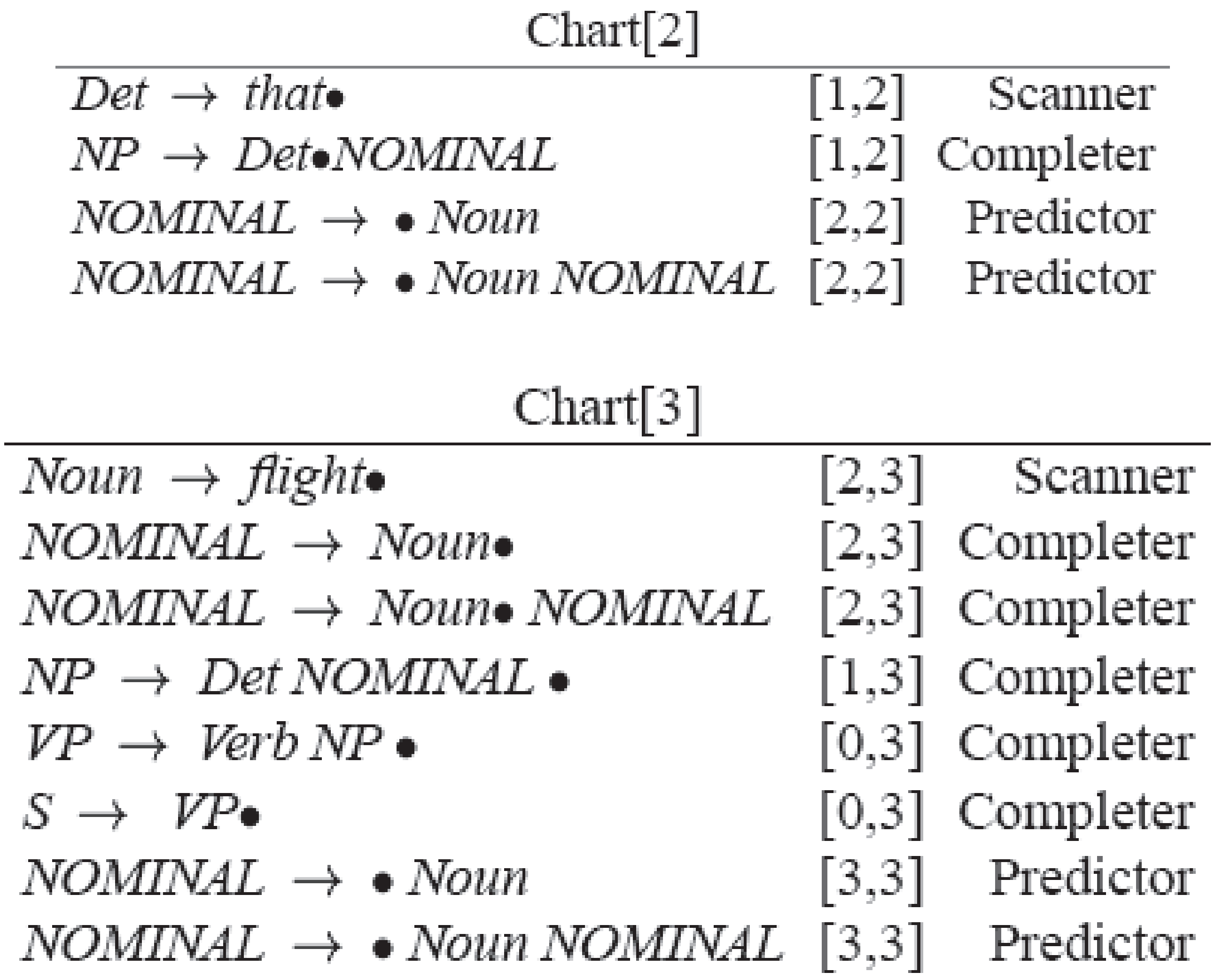
Verb → *book* •	[0,1]	Scanner
VP → *Verb*•	[0,1]	Completer
S → *VP*•	[0,1]	Completer
VP → *Verb* • *NP*	[0,1]	Completer
NP → • *Det NOMINAL*	[1,1]	Predictor
NP → • *Proper-Noun*	[1,1]	Predictor



Chart[2]

Det → *that*•	[1,2]	Scanner
NP → *Det*•*NOMINAL*	[1,2]	Completer
NOMINAL → • *Noun*	[2,2]	Predictor
NOMINAL → • *Noun NOMINAL*	[2,2]	Predictor

An Example



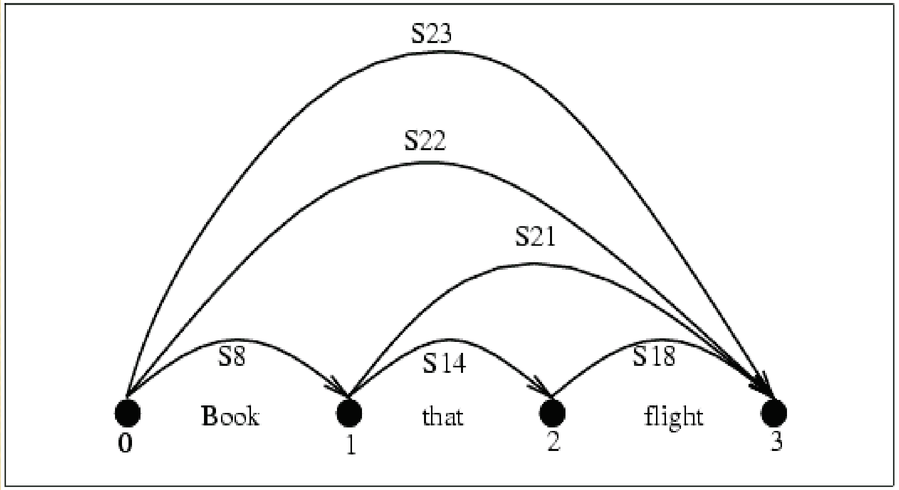
Chart[2]

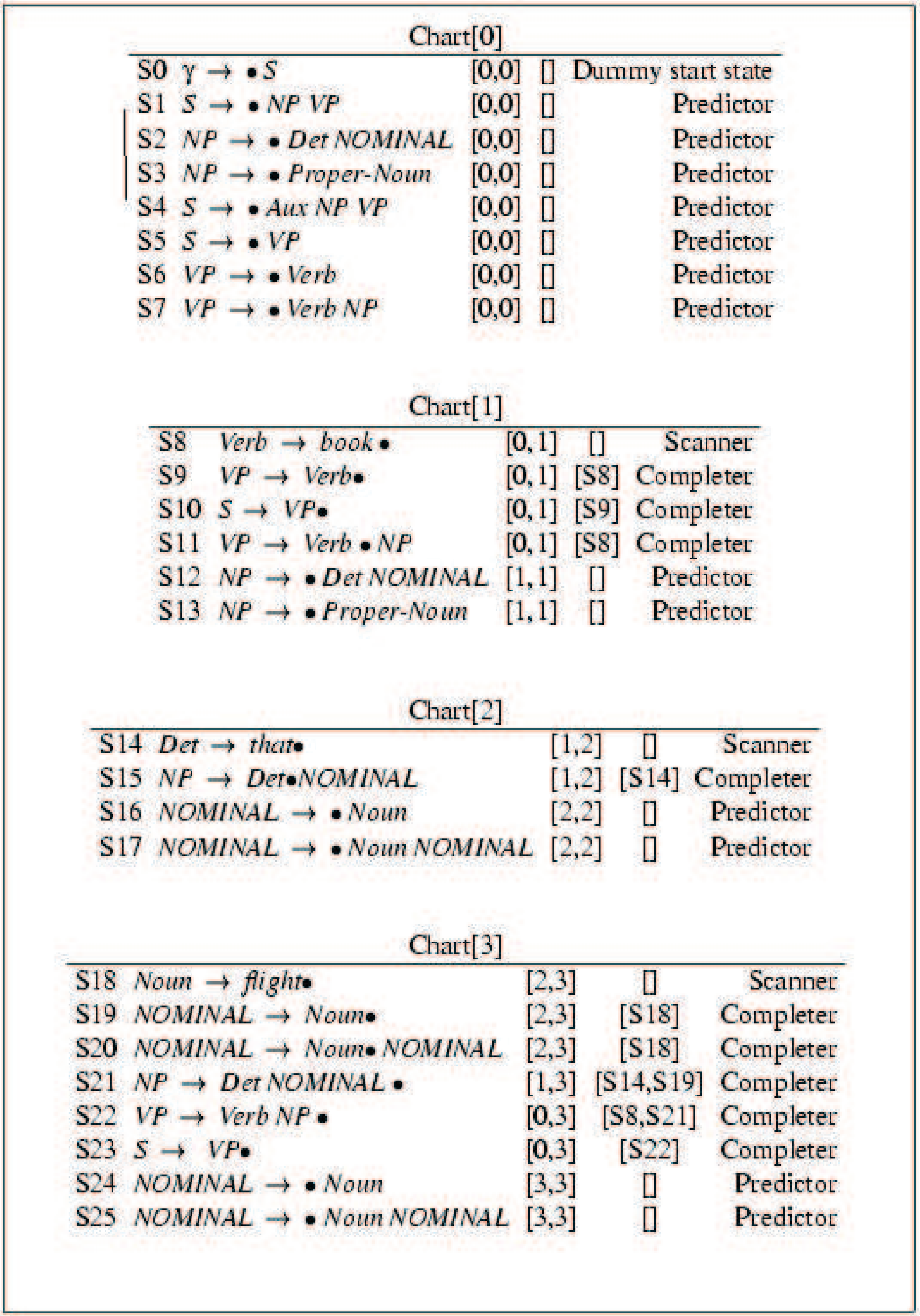
Det → *that*•	[1,2]	Scanner
NP → *Det*•*NOMINAL*	[1,2]	Completer
NOMINAL → • *Noun*	[2,2]	Predictor
NOMINAL → • *Noun NOMINAL*	[2,2]	Predictor

Chart[3]

Noun → *flight*•	[2,3]	Scanner
NOMINAL → *Noun*•	[2,3]	Completer
NOMINAL → *Noun*• *NOMINAL*	[2,3]	Completer
NP → *Det NOMINAL* •	[1,3]	Completer
VP → *Verb NP* •	[0,3]	Completer
S → *VP*•	[0,3]	Completer
NOMINAL → • *Noun*	[3,3]	Predictor
NOMINAL → • *Noun NOMINAL*	[3,3]	Predictor

Earley Algorithm





Chart[0]

S0	$\gamma \rightarrow \bullet S$	[0,0]	[]	Dummy start state
S1	$S \rightarrow \bullet NP\ VP$	[0,0]	[]	Predictor
S2	$NP \rightarrow \bullet Det\ NOMINAL$	[0,0]	[]	Predictor
S3	$NP \rightarrow \bullet Proper\text{-}Noun$	[0,0]	[]	Predictor
S4	$S \rightarrow \bullet Aux\ NP\ VP$	[0,0]	[]	Predictor
S5	$S \rightarrow \bullet VP$	[0,0]	[]	Predictor
S6	$VP \rightarrow \bullet Verb$	[0,0]	[]	Predictor
S7	$VP \rightarrow \bullet Verb\ NP$	[0,0]	[]	Predictor

Chart[1]

S8	$Verb \rightarrow book \bullet$	[0,1]	[]	Scanner
S9	$VP \rightarrow Verb \bullet$	[0,1]	[S8]	Completer
S10	$S \rightarrow VP \bullet$	[0,1]	[S9]	Completer
S11	$VP \rightarrow Verb \bullet NP$	[0,1]	[S8]	Completer
S12	$NP \rightarrow \bullet Det\ NOMINAL$	[1,1]	[]	Predictor
S13	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	[]	Predictor

Chart[2]

S14	$Det \rightarrow that \bullet$	[1,2]	[]	Scanner
S15	$NP \rightarrow Det \bullet NOMINAL$	[1,2]	[S14]	Completer
S16	$NOMINAL \rightarrow \bullet Noun$	[2,2]	[]	Predictor
S17	$NOMINAL \rightarrow \bullet Noun\ NOMINAL$	[2,2]	[]	Predictor

Chart[3]

S18	$Noun \rightarrow flight \bullet$	[2,3]	[]	Scanner
S19	$NOMINAL \rightarrow Noun \bullet$	[2,3]	[S18]	Completer
S20	$NOMINAL \rightarrow Noun \bullet NOMINAL$	[2,3]	[S18]	Completer
S21	$NP \rightarrow Det\ NOMINAL \bullet$	[1,3]	[S14,S19]	Completer
S22	$VP \rightarrow Verb\ NP \bullet$	[0,3]	[S8,S21]	Completer
S23	$S \rightarrow VP \bullet$	[0,3]	[S22]	Completer
S24	$NOMINAL \rightarrow \bullet Noun$	[3,3]	[]	Predictor
S25	$NOMINAL \rightarrow \bullet Noun\ NOMINAL$	[3,3]	[]	Predictor

References

- Speech and Language Processing, *Jurafsky and H.Martin*
  [Chapter 10. Parsing with Context-Free Grammars]

Thank You