Choosing an operator

Only one operator applies to a given state, since the conditions that select an operator are mutually exclusive:

> If there is no symbol after the dot, use the **completer**.
> If there is a symbol after the dot and it is terminal, use the **scanner**.
> If there is a symbol after the dot and it is non-terminal, use the **predictor**.

Predictor:
The predictor is the top-down aspect of the algorithm. It draws the consequences from a previous top-down prediction, by looking up all the rules whose left-hand side is equal to the (non-terminal) symbol after the dot, and building a new state from each such rule. It operates like this[1]:

> Applying to a state of the form $Z \rightarrow \alpha\ .X\ \beta\ \ j, k$, where X is non-terminal:
>
> For **each** rule in the grammar of the form $X \rightarrow \ RHS$, add a new state of the form $X \rightarrow ^{o} .RHS\ \ k, k$

Scanner:
The scanner is the operator that compares a top-down predicted terminal with the actual next word of input. If there is a match, it constructs a new state by moving the dot and incrementing k to represent the fact that a word of input has been successfully processed.

> Applying to a state of the form $Z \rightarrow \alpha\ \ .X\ \beta\ \ j, k$, where X is terminal:
>
> Compare X to the next symbol in the input string. The next symbol is the k_{th} symbol of the input. If X and the k_{th} input symbol match, add a state of the form $Z \rightarrow \alpha\ \ X.\ \beta\ \ j, k+1$

Completer:
The role of the completer is to complete previous predictions that have been at least partially supported by the subsequent parse represented by the current state. The completer looks back in the state list to find hypotheses that are waiting for a constituent of the kind that has just been completed, at the position where it was just found.

> Applying to a state of the form $Z \rightarrow \gamma.\ \ j, k$:
>
> For each (necessarily previous) state of the form $X \rightarrow \ \alpha\ .Z\ \beta\ \ i, j$, add a new state of the form $X \rightarrow \ \alpha\ \ Z.\ \beta\ \ i, k$

[1] Here and throughout, Greek letters are interpreted as variables that stand for any (possibly empty!) sequence of syntactic symbols.