CS224N/Ling280 – Generalized CKY Parsing

Initial grammar

S	→	NP VP	N	→	*cats*
VP	→	V NP	N	→	*claws*
VP	→	V NP PP	N	→	*people*
NP	→	NP PP	N	→	*scratch*
NP	→	N	V	→	*scratch*
NP	→	e	P	→	*with*
NP	→	N N			
PP	→	P NP			

Epsilon removal

S	→	NP VP	N	→	*cats*
VP	→	V NP	N	→	*claws*
VP	→	V NP PP	N	→	*people*
NP	→	NP PP	N	→	*scratch*
NP	→	N	V	→	*scratch*
NP	→	N N	P	→	*with*
PP	→	P NP			
PP	→	P			
NP	→	PP			
VP	→	V			
VP	→	V PP			
S	→	VP			

Binarization

S	→	NP VP	N	→	*cats*
VP	→	V NP	N	→	*claws*
VP	→	V $\boxed{\text{VP}\to\text{V.}}$	N	→	*people*
NP	→	NP PP	N	→	*scratch*
NP	→	N	V	→	*scratch*
NP	→	N N	P	→	*with*
PP	→	P NP			
PP	→	P			
NP	→	PP			
VP	→	V			
VP	→	V PP			
S	→	VP			
$\boxed{\text{VP}\to\text{V.}}$	→	NP PP			

There are many ways to do binarization. The assignment code does a simple left to right binarization that turns the grammar into a trie data structure for each category. This is a sensible thing to do because it compacts the grammar search space, and makes it easy to find appropriate rules based on the left corner.

Unaries

The standard Chomsky Normal Form transformation folds unaries. That can be done in two directions: to keep the lowest or highest element in the chain. This is okay if you never want to see them again. But it might be better to keep them. For two reasons: you want to preserve the nodes in unary chains, and you want to keep your grammar smaller. This leads you into the space of "Generalized CKY parsing". It is then necessary to correctly deal with the special properties of unaries: you either need to in advance do a closure over unary chains (for instance, the grammar also licenses S → V), and then to reconstruct the chains afterwards, or else you need to loop in each cell to make sure all unaries are explored. Below we fold unaries, keeping the lowest item.

S	→	NP VP	N	→	*cats*
VP	→	V NP	N	→	*claws*
VP	→	V $\boxed{\text{VP}\rightarrow\text{V.}}$	N	→	*people*
NP	→	NP PP	N	→	*scratch*
NP	→	N	V	→	*scratch*
NP	→	N N	P	→	*with*
PP	→	P NP			
PP	→	P			
NP	→	PP			
VP	→	V			
VP	→	V PP			
S	→	VP			
$\boxed{\text{VP}\rightarrow\text{V.}}$	→	NP PP			
S	→	V			
NP	→	P			

Parse Triangle

	0	1 *cats*	2 *scratch*	3 *people*	4 *with*	5 *claws*
0						
1						
2						
3						
4						
5						

	0	1	2	3	4	5
		cats	*scratch*	*people*	*with*	*claws*
0		N NP		S		S
1			N V NP VP S	VP S		VP S
2				N NP		NP $\boxed{\text{VP}\rightarrow\text{V.}}$
3					P PP NP	PP NP
4						N NP
5						