A basic grammar for English

S	$\rightarrow$	$NP\ VP$	DET	$\rightarrow$	*the* | *a* | *my* ...
NP	$\rightarrow$	$DET\ N$	N	$\rightarrow$	*boy* | *girl* | *book*
NP	$\rightarrow$	$NP\ PP$	V	$\rightarrow$	*cries* | *sleeps*
VP	$\rightarrow$	V	V	$\rightarrow$	*likes* | *reads*
VP	$\rightarrow$	$V\ NP$	V	$\rightarrow$	*dreams* | *complains*
VP	$\rightarrow$	$V\ PP$	V	$\rightarrow$	*shares* | *discusses*
VP	$\rightarrow$	$V\ NP\ NP$	V	$\rightarrow$	*gives* | *tells*
VP	$\rightarrow$	$V\ NP\ PP$	N	$\rightarrow$	*talks* | *speaks*
VP	$\rightarrow$	$V\ PP\ PP$	P	$\rightarrow$	*of* | *about* | *to* | *with*
PP	$\rightarrow$	$P\ NP$		$\rightarrow$	

Context-Free Grammars

- A context-free grammar (CFG) is defined as $G = (\Sigma, V, S, P)$ where :
  - Σ is a finite set of *terminal* symbols
  - V is a finite set of *non-terminal* symbols, $V \cap \Sigma = \emptyset$
  - S is a distinguished symbol in V, termed the *axiom* of the grammar
  - P is a set of rewrite rules such that if $\alpha \Rightarrow \beta$ is in P, then (i) $\alpha \in V$ and (ii) $\beta \in (V \cup \Sigma)^*$
- The language $L(G)$ of a grammar G is defined as :

$$L(G) = \{w \in \Sigma^\star \text{ st. } S \Rightarrow^\star_G w\}$$

- A word in $L(G)$ derives from S and contains only terminal symbols
- A *protoword* is any word α in $(V \cup \Sigma)^*$ such that $S \overset{\star}{\Rightarrow} \alpha$.

A simple context-free Grammar

- Let $G_1 = (\{a, b\}, \{S\}, S, \{S \rightarrow aSb, S \rightarrow ab\})$
- $ab \in L(G_1)$, since $S \Rightarrow ab$
- $aabb \in L(G_1)$: $S \Rightarrow aSb$ (by rule 1) and $aSb \Rightarrow aabb$ (by rule 2), then $S \overset{\star}{\Rightarrow} aabb$.
- $\forall n > 0, a^n b^n \in L(G_1)$. Proof :
  - true for $n = 1, n = 2$.
  - if true for n, then $S \overset{\star}{\Rightarrow} a^n b^n$ and the last rule used must be $S \rightarrow ab$: $S \overset{\star}{\Rightarrow} a^{n-1} S b^{n-1} \Rightarrow a^n b^n$.
    Replace the last rule with $S \Rightarrow aSb \Rightarrow aabb$ to get $a^{n+1} b^{n+1}$
- $L(G_1) = \{a^n b^n, n > 0\}$ (consider the protowords).

Recognition = Search

- $\Rightarrow_G$ defines an oriented graph (=binary relationship) over the set of protowords.
- $w \overset{?}{\in} L(G)$ amounts to finding a path in this graph
- caveat : the graph is infinite (yet locally finite), classical path-finding algorithms don't work
- search strategies :
  - from S to w : top-down recognition
  - from w to S : bottom-up recognition
  - depth-first (sequential exploration of paths)
  - breadth-first (parallel exploration of paths)
  - ...

Brute-force approach : top-down recursive descent

Solution :

```
% top down recognition
tdr(Proto, Word):-  match(Proto, Word, [], []).

tdr([X|Proto], Word) :-
rule(X, RHS),
append(RHS, Proto, NewProto),
match(NewProto, Word, NewProto1, NewWord),
tdr(NewProto1, NewWord).

match([X|L1], [X|L2], R1, R2) :- !,  match(L1,
match(L1, L2, L1, L2).
```

- this procedure only visit left-derivations : is it a problem ?
- speed-up : check that the *terminal prefix* of `NEWP` is consistent with `W`.
- it fails if G contains left-recursive rules $A \rightarrow A\alpha$. Why ?

Brute-force approach : top-down recursive descent

Solution :

```
% top down recognition
bulrp([s]).

bulrp(P):-
    append(Pref,Rest,P),
    append(RHS,Suff,Rest),
    rule(X,RHS),
    append(Pref,[X|Suff], NEWP),
    write(NEWP), nl,
    bulrp(NEWP).
```

- bottom-up parsing is based on *reductions*
- it fails if G contains unproductive rules or chain rules rules $A \overset{\star}{\Rightarrow} A$. Why ?