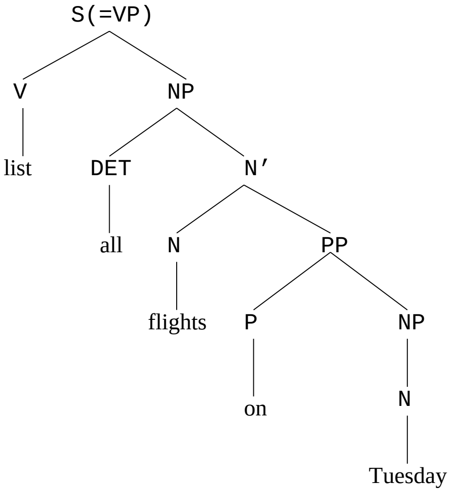
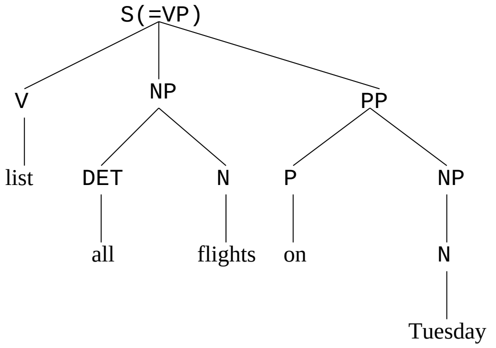
CS/ENGRI 172, Fall 2003: Computation, Information, and Intelligence
11/5/03: Context Free Grammars

Office Hours Change: William Lin's office hours have been moved from 2:30-3:30 on Mondays to **4:00-5:00 on Mondays in Upson 328A** for the remainder of the semester.

Alternate Example Sentence Structures





Context Free Grammars

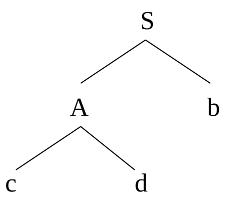
A *context free grammar* (CFG) is made up of four parts:

- *terminals*: a finite set of at least one symbol;
- *non-terminals* (or *variables*): a finite set of at least one symbol (distinct from the non-terminals);
- a single designated *start non-terminal*;
- *rewrite rules*: a finite set of at least one rule describing how a single non-terminal can be rewritten as a sequence of terminals and/or non-terminals (possibly intermixed).

A CFG specifies a *language* made up of *sentences.* A sentence is a sequence of terminal symbols; sentences are generated by applying rewrite rules to the non-terminals an a sequence (starting with the sequence of just the start non-terminal) until only terminals remain.

Parse Trees

We can represent the rewriting process by *parse trees.* In a parse tree, the interior nodes are labeled by non-terminals, with the root labeled with the start non-terminal. The leaves are labeled by terminals. The children of an internal node represent, in order, the result of rewriting the non-terminal labeling the node according to one of the rewrite rules in the grammar. That is, if we have the following parse tree:



then the CFG generating the parse tree must contain the rewrite rules $A \to cd$ and $S \to Ab$.

Example CFG

Recall our example sentence "List all flights on Tuesday." with its two structural analyses given above, each corresponding to a different interpretation. We give below a CFG that will generate these two structures as parse trees (as well as many other sentences). Observe that we can use a "linear" notation with brackets (rather than the larger tree notation) to indicate the primary difference between the structures of the two interpretations:

list [all [flights on Tuesday]$_{N'}$]$_{NP}$
list [all flights]$_{NP}$ [on Tuesday]$_{PP}$

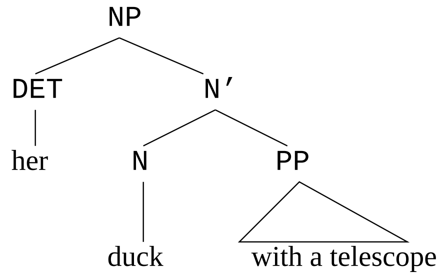
The following CFG will generate parse trees representing both interpretations:

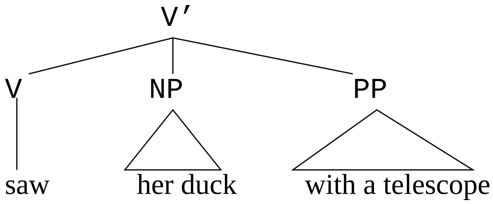
- Terminals: list, all, flights, on, Tuesday
- Non-terminals: S, NP, N′, PP, V, DET, N, P
- Start non-terminal: S
- Rewrite rules:

(1)	S	→	V NP	(7)	N	→	flights
(2)	S	→	V NP PP	(8)	N′	→	N PP
(3)	V	→	list	(9)	PP	→	P NP
(4)	NP	→	DET N′	(10)	P	→	on
(5)	NP	→	DET N	(11)	NP	→	N
(6)	DET	→	all	(12)	N	→	Tuesday

X-bar Theory example

Recall that XP is equivalent to X″, and that X-bar theory says that X is the head of X′ and X″. We can use X-bar theory to explain why we can rule out some interpretations of "I saw her duck with a telescope.", where we assume "duck" is a noun:


NP
DET
N′
her
N
PP
duck
with a telescope


V′
V
NP
PP
saw
her duck
with a telescope