48-747 Shape Grammars

HISTORICAL

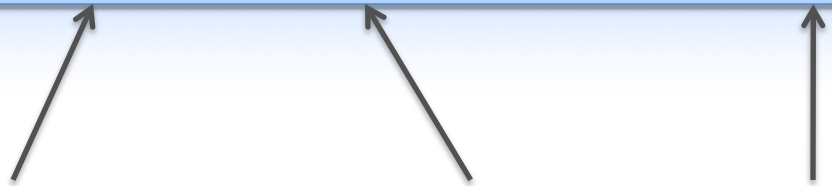
WHAT IS A GRAMMAR? A LANGUAGE?

Oxford English Dictionary

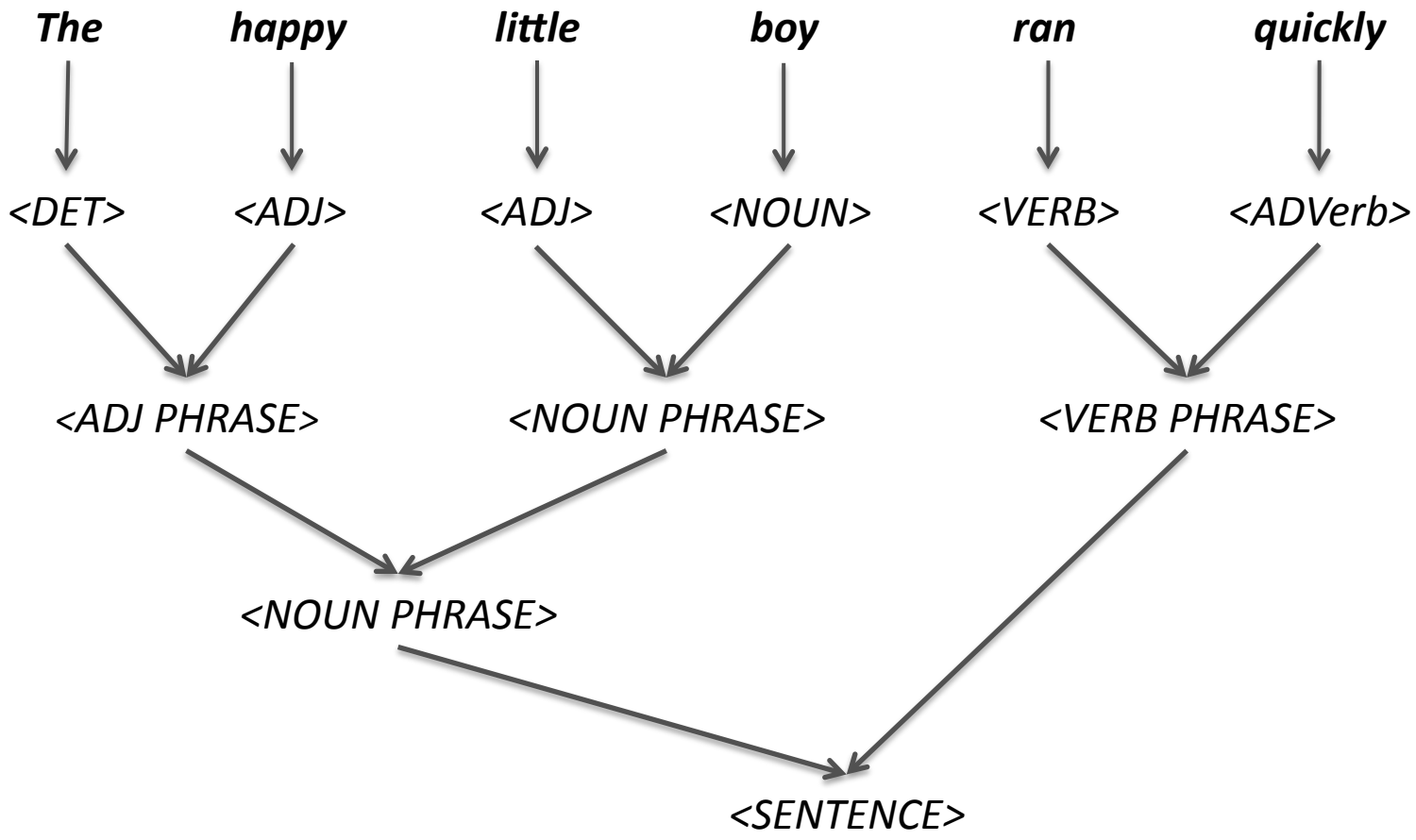
Language is the method of ‘human’ **communication**, either spoken or written, consisting of the **use** of ‘words’ in an **agreed** way.

Language is the **style** or faculty of expression

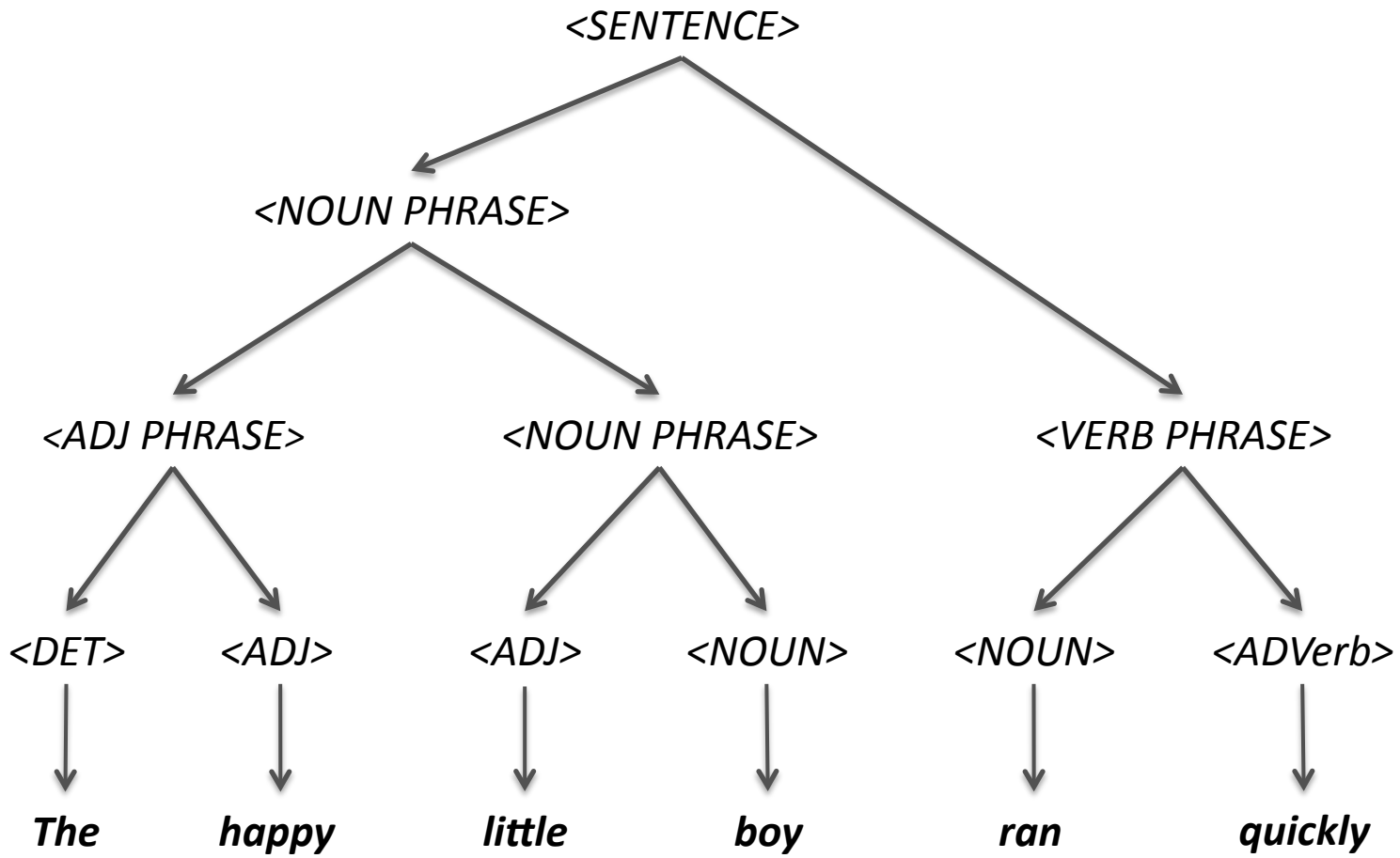
Language is the system of **symbols** and **rules** for writing **algorithms**



language



underlying language is the notion of a **sentence**


<SENTENCE>
<NOUN PHRASE>
<ADJ PHRASE>
<NOUN PHRASE>
<VERB PHRASE>
<DET>
<ADJ>
<ADJ>
<NOUN>
<NOUN>
<ADVerb>
The
happy
little
boy
ran
quickly

parse tree

entities:
things like <SENTENCE> <NOUN PHRASE>
things like ***the***, ***little***, ***boy***, ***quickly***

relationships between *parts* (of the sentence):
a <SENTENCE> is a <NOUN PHRASE> followed by a <VERB
a <NOUN PHRASE> is a <ADJ PHRASE> followed by a <NOUN PHRASE>
a <ADJ PHRASE> is a <DET> followed by a <ADJective>
a <NOUN PHRASE> is an <ADJective> followed by a <NOUN>
a <VERB PHRASE> is a <VERB> followed by a <ADVerb>
boy is a <NOUN> and ***ran*** is a <VERB>

we **recognize** parts of a sentence

<SENTENCE> ➔ *<NOUN PHRASE> <VERB>*

<NOUN PHRASE> ➔ *<ADJ PHRASE> <NOUN PHRASE>*

<ADJ PHRASE> ➔ *<DET> <ADJective>*

<NOUN PHRASE> ➔ *<ADJective> <NOUN>*

<VERB PHRASE> ➔ *<VERB> <ADVerb>*

<NOUN> ➔ *boy*

<VERB> ➔ *ran*

relationships specify **rules**

vocabulary or alphabet symbols

- *TERMINAL*
- *NON TERMINAL*

entities

- *strings of symbols juxtapositioned together*

sentence

- *string made up of* terminal *vocabulary symbols*

production

- *RULES made up of strings expressing a relationship between two such entities and expressed as* $a \rightarrow b$

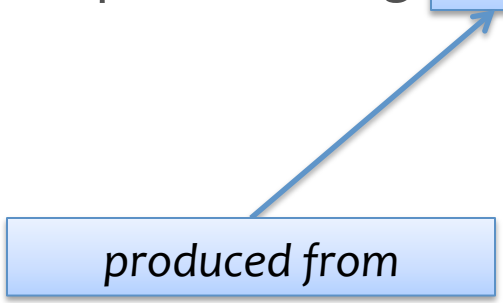
initial string or seed

made up of vocabulary elements

grammar

The ***language***, *L*, of a grammar, *G*, is the set of all sentences (i.e., without non-terminals) that are produced from the initial string

L(*G*) = { sentence | initial-string ⇒* sentence }



language

A rule is ***applicable***

to the **current** string which is either the initial string or a string produced from the initial string

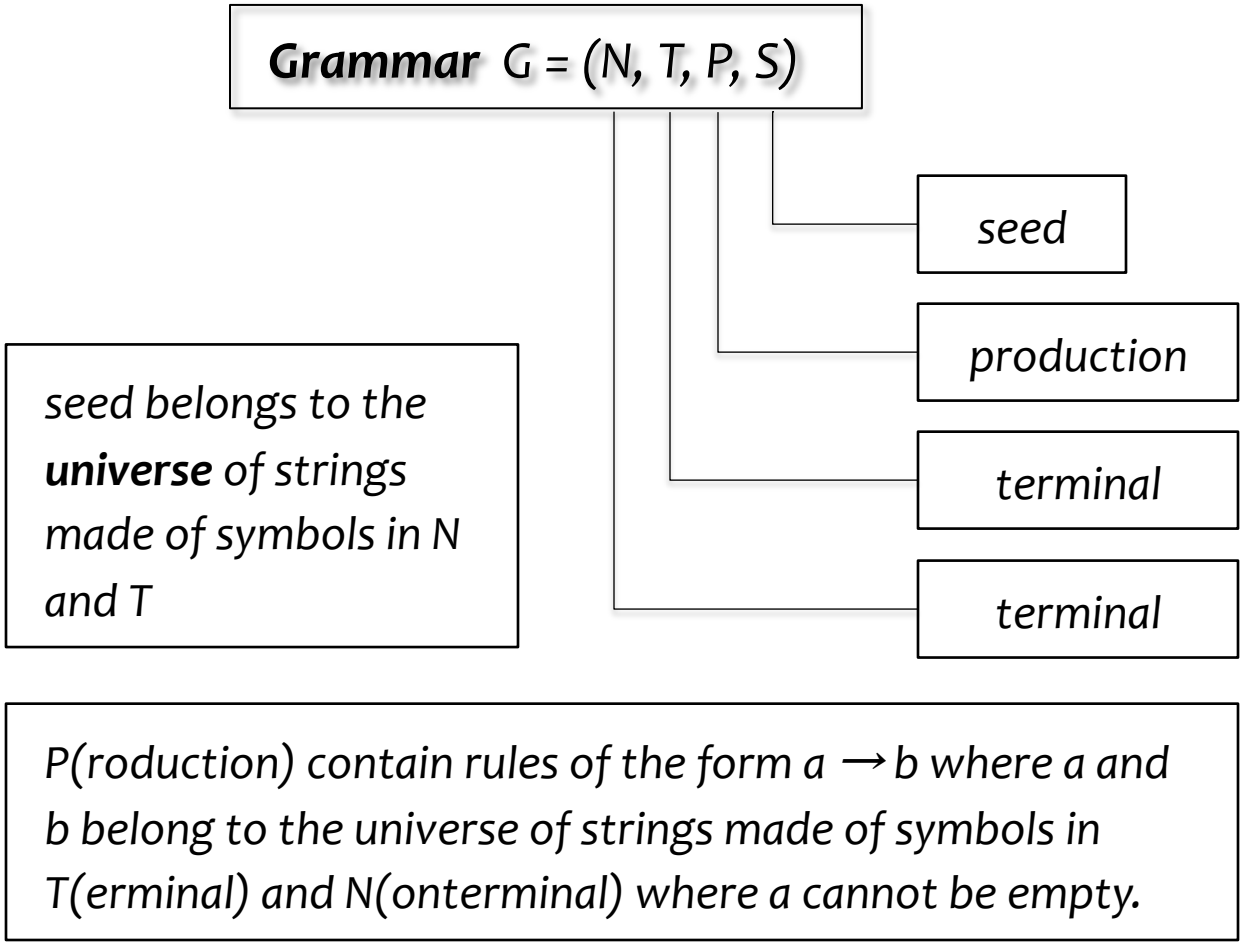
whenever

the *left hand side* of the rule '***occurs***' in the object

in which case

it can be replaced by the *right hand side* of the rule under rule application

rule application



formal definition

VOCABULARY *is a limited set of symbols no two of which are similar or identical*

If we are given a set of symbols V, then we can create a set V called the* ***UNIVERSE*** *(or* ***LEAST SET****) of V in the following manner:*

The empty string e is in V.*

Every symbol in V is in V.*

If a and b are strings in V, so too is their juxtaposition (concatenation) ab in V**

V is closed under concatenation*

We denote the set $V^* - \{e\}$ *by* V^+.

For any production $a \rightarrow b$, *we have* $a \in V^+$ *and* $b \in V^*$

vocabulary and **universe**

$V = \{ 0, 1\}$

$V^* = \{ e, 0, 1, 00, 01, 10, 11, 000, 001, \ldots \}$

$V^+ = \{ 0, 1, 00, 01, 10, 11, 000, 001, \ldots \}$

example - binary sequences

From S by applying rule S → 0S1 we get 0S1
That is, substitute the occurrence of the left hand side of the current rule in the current string by the right hand side. That is, S ⇒ 0S1

By applying rule S → 0S1 again we get 00S11.
That is, S ⇒ 0S1 ⇒ 00S11. Or, S ⇒ 00S11*

If we apply rule S → 01 we get S ⇒ 0S1 ⇒ 00S11 ⇒ 000111

G = (N= {S}, T = {0, 1}, P = { S→ 0S1, S→ 01}, S)

What can we do with this grammar?

example of a grammar

PHRASE STRUCTURE GRAMMARS

productions of the form $\alpha \rightarrow \beta$

CONTEXT SENSITIVE GRAMMARS

productions of the form $\alpha_1 A \alpha_2 \rightarrow \beta_1 B \beta_2$ *equivalently,* $|\alpha| \leq |\beta|$

CONTEXT FREE GRAMMARS

productions of the form $A \rightarrow \beta$ *where* A *is a single variable*

REGULAR GRAMMARS

productions of the form $A \rightarrow aB$ *or* $A \rightarrow a$ *where* A *is a single variable*

types of grammars

$G = ($

$N = \{ S, A, B \},$ *non-terminals*

$T = \{ a, b \},$ *terminals*

All strings with equal numbers of a's and b's.

$A \rightarrow a,\ A \rightarrow aS,\ A \rightarrow bAA,$

$B \rightarrow b,\ B \rightarrow bS,\ B \rightarrow aBB \},$

$S)$ *seed*

What is the language L(G)?

a **cfg** example

Note that if we can recognize whether a sentence is in a language then we can always generate the language simply by generating each element in the universe and checking whether the element is in the language

example:

given an integer i > 1, is it prime?

1. set j to 2
2. if j ≥ i, then halt. i is prime.
3. if i /j is an integer, then halt. i is not prime.
4. set j to j+1 and go to 2.

grammars, **procedures** and **algorithms**

another example:

given an integer i > 1, is there a perfect integer greater than i?

1. set k = i
2. set k = k+1, sum= 0, j = 1
3. if j < k, go to 6
4. if sum | k, go to 2
5. Halt. There is a perfect integer.
6. if k/j is not an integer, go to 8.
7. set sum=sum+j
8. set j to j+1 and go to 3.

A set is ***recursively enumerable*** (r.e) if it can be generated by a *procedure*

Another way of looking at this is that there is a 1-1 correspondence between elements of the set and the natural numbers.
In other words, we can "***index***" the elements of the set

A set is ***recursive*** if it can be generated by an *algorithm*

Languages of grammars are *recursively enumerable*

psl are recursively enumerable
csl, cfl, rl are recursive

Equivalently, *a set that is recursively enumerable is the language of some grammar*

recursively enumerable and **recursive**

"the search for rigorous formulation in **linguistics** has a much more serious motivation than mere concern for logical niceties or the desire to purify well-established methods for **language** analysis

precisely constructed models for **linguistic** structure can play an important role, both negative and positive, in the process of discovery itself

by pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the **linguistic** data"

noam chomsky – ***syntactic structures***