BACK TO THE MENU

1. Introduction
2. Belt-selectors

→ 3. Tetrasystems

- letter-refiner
- components of a tetrasystem
- belt-selector stagnancy
- tetrasystem models of other devices

A LETTER-REFINER

- unboundedly context-sensitive
- use example:

ABCDEF

↓

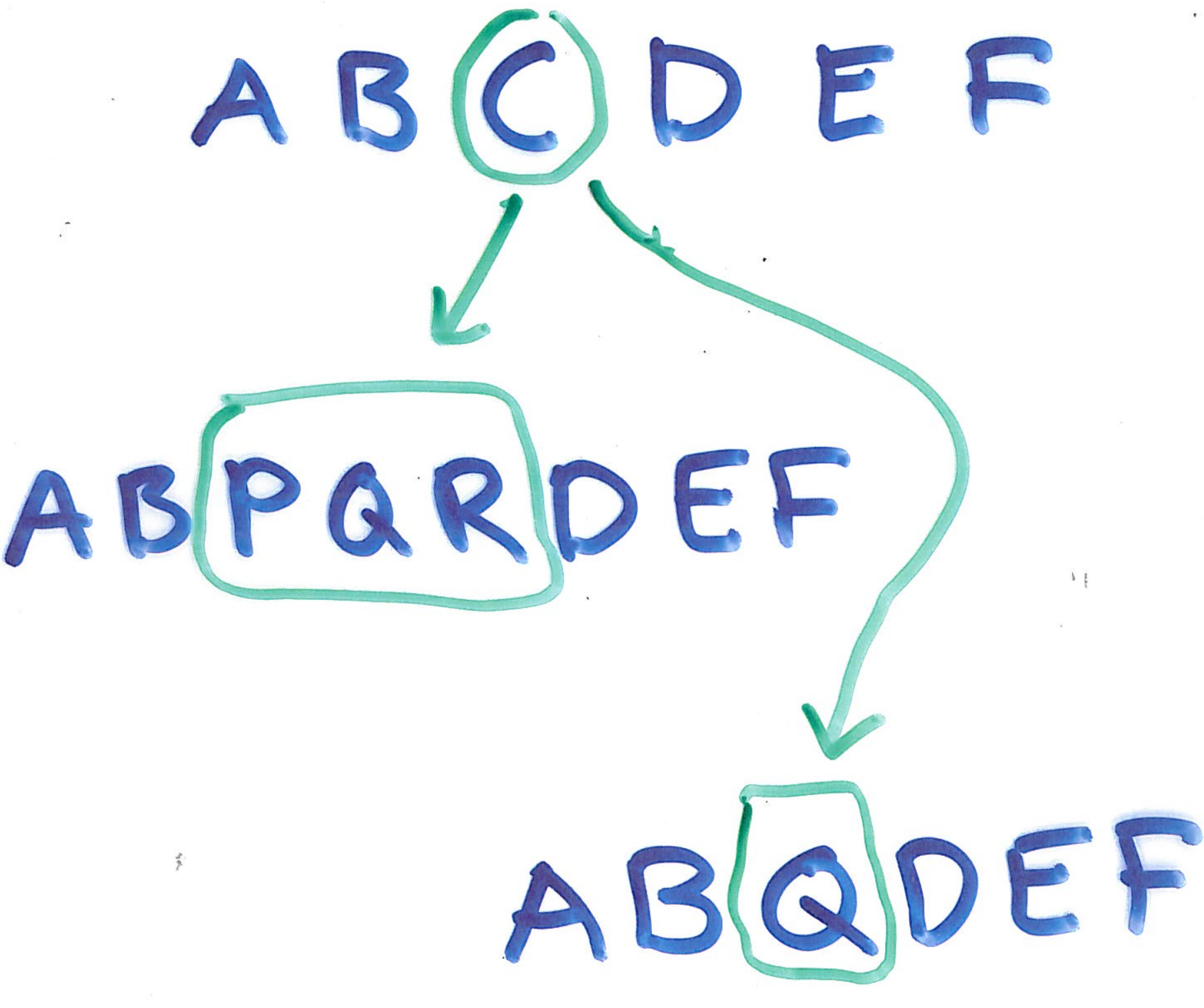
ABPQRDEF

supposing that

$PQR \in r(AB, C, DEF)$

for our letter-refiner r

The letter-refiner may be nondeterministic!



ON THE ALPHABET

- may be infinite (in order to cope with structured symbols)
- divides as follows:
  - nonterminals, including the seed letter
  - (optional) terminals, which cannot be refined

COMPONENTS OF A TETRASYSTEM

$$\langle V_N, V_T, c_s, r, \langle s_1, s_2, s_3, s_4 \rangle \rangle$$

V_N nonterminals

V_T terminals

c_s seed letter

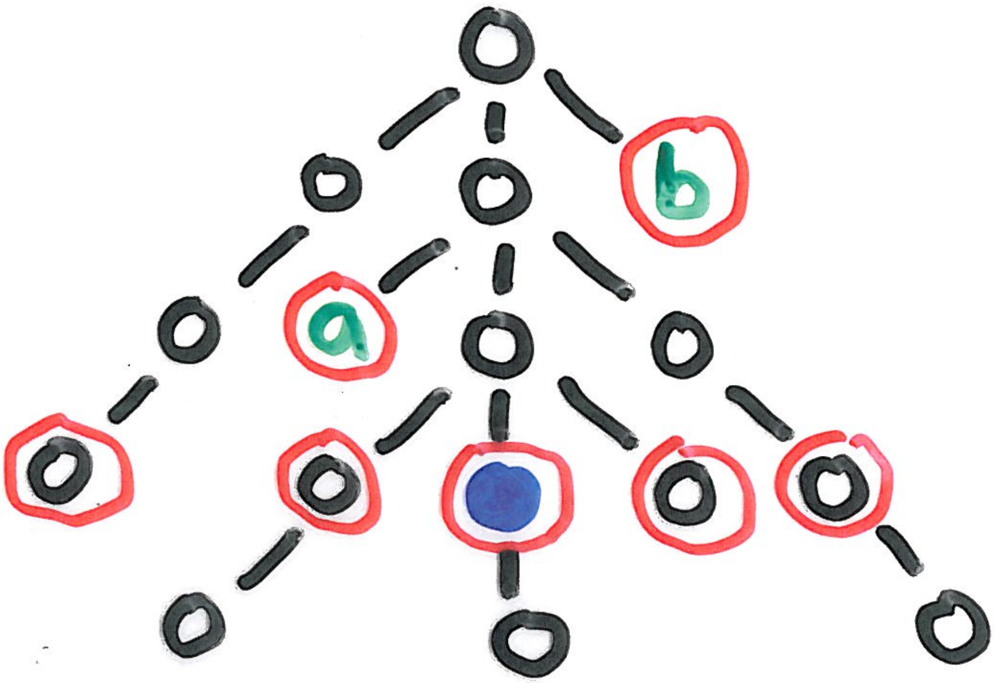
r letter-refiner

$\langle s_1, s_2, s_3, s_4 \rangle$ is the control frame consisting of four belt-selectors

A PREVIEW: THE FRAMES OF THE TETRASYSTEM MODELS OF...

	S_1	S_2	S_3	S_4
macro processors	$\sigma_E \parallel \sigma_I$	σ_E	σ_E	σ_E
(context-sensitive) Chomsky grammars	σ_I	σ_E	σ_E	σ_E
pure grammars	σ_I	σ_E	σ_I	σ_E
L systems	σ_C	σ_C	$\Delta\sigma_C$	$\Delta\sigma_C$

BELT-SELECTOR STAGNANCY



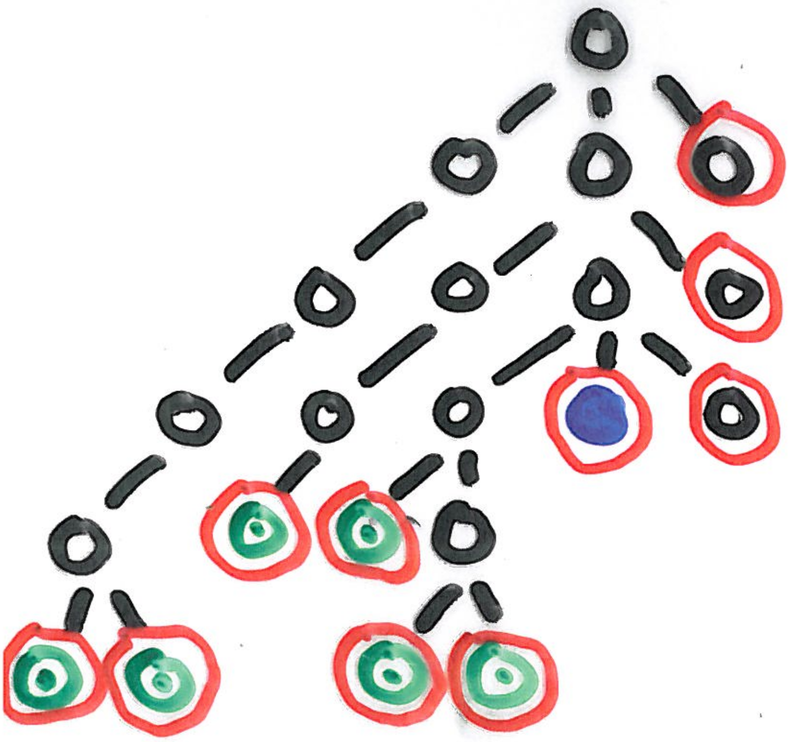
Here σ_c (with $\varphi_{\sigma_c}(i) = |i|$) is stagnant iff both a and b are terminal letters

ON MACRO PROCESSING

- leaf expansion proceeds left-to-right and depth-first
- global variables are read from the left context, and the right context is ignored
- when all leaves hold terminals, they constitute the output sequence

(M)

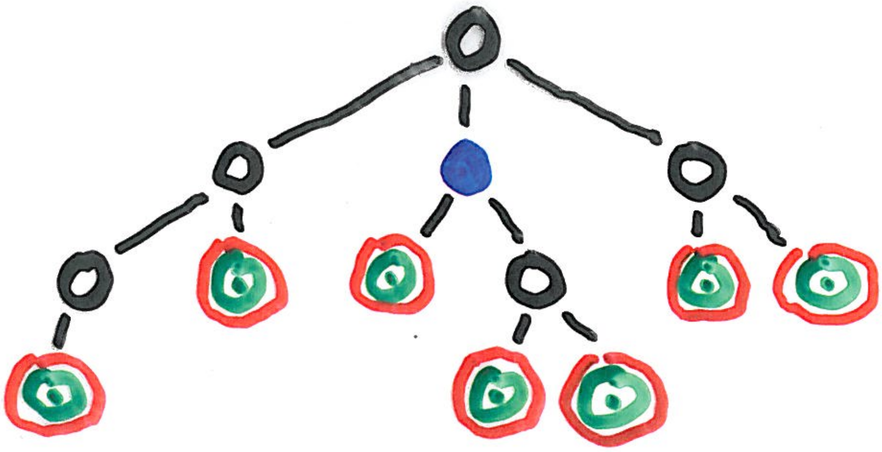
LEAF EXPANSION



$G_E || G_I$

- the leaf expansion can take place when $S_1 = G_E || G_I$ is stagnant at a nonterminal-lettered leaf
- the refinement context is picked by $S_2 = G_E$

(M) OUTPUT EXTRACTION



- output extraction can take place when $S_3 = \sigma_E$ is stagnant at a non-leaf node
- the actual output sequence is picked up by $S_4 = \sigma_E$

	when ready?	from where the data?
leaf expansion	fertile-ness S_1	S_2 context
output extraction	S_3 mature-ness	S_4 output

33F

A CONTEXT-SENSITIVE CHOMSKY GRAMMAR

Start symbol: S

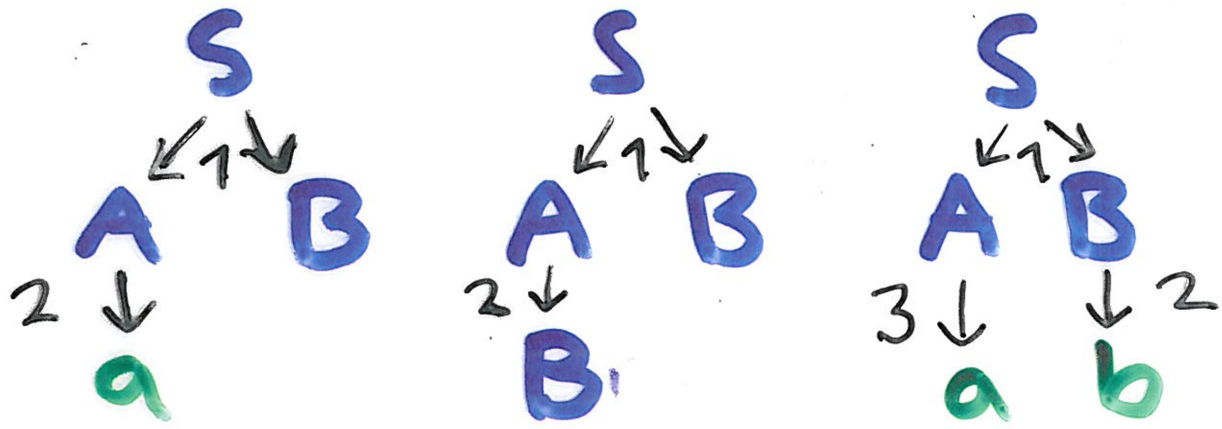
Productions:

$S \rightarrow AB$

$A \rightarrow a$

$AB \rightarrow BB$

$AB \rightarrow Ab$



So ab is the only output word produced

Ⓒ LETTER-REFINER CONSTRUCTION

$$r(w_1, S, w_2) = \{AB\}$$

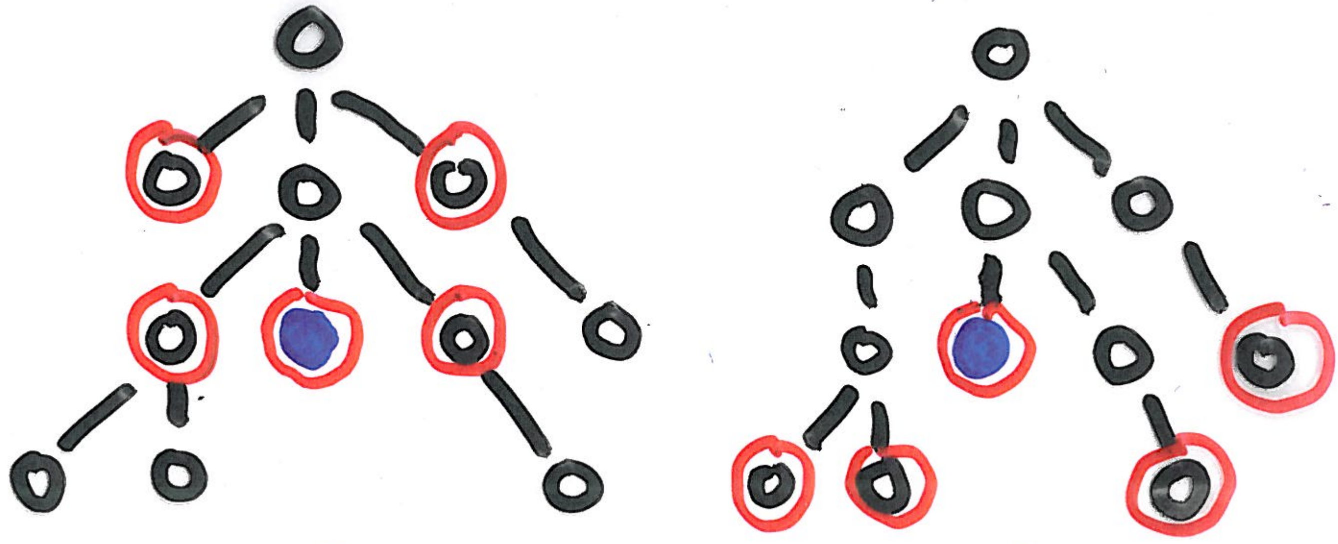
$$r(w_1, A, w_2) = \begin{cases} \{a, B\} & \text{when } w_2 = Bw \\ \{a\} & \text{otherwise} \end{cases}$$

$$r(w_1, B, w_2) = \begin{cases} \{b, B\} & \text{when } w_1 = wA \\ \{B\} & \text{otherwise} \end{cases}$$

AN ASIDE: PROBLEMS IN LETTER-REFINER CONSTRUCTION IN GENERAL

1. The original rule set may effectively have to be augmented with dummy productions such as $B \to B$
2. Length-decreasing productions such as $S \to \Lambda$ cannot be modeled
3. If the rewriting starts from a word (and not a letter), an extra seed letter is needed

(C) LEAF EXPANSION



σ_I σ_E

$s_1 = \sigma_I$ means that every nonterminal-lettered leaf can always be expanded

$s_2 = \sigma_E$ picks up the refinement context

(c) OUTPUT EXTRACTION

$S_3 = \sigma_E$ means that output cannot be extracted before all leaves are terminal-lettered

$S_4 = \sigma_E$ means that output is then extracted from the leaf sequence

(In all, the same as with macro processors!)

A PURE GRAMMAR

- By definition:
  - there are no terminals
  - every word produced is an output word

Start word: BAB
Productions (only one): $AB \to AAB$

- So the output words are:

$BAB \to BAAB \to BAAAB \to BAAAAB \to \ldots$

Ⓟ LEAF EXPANSION

$s_1 = G_I$ $s_2 = G_E$

Thus, the same as with context-sensitive Chomsky grammars!

Ⓟ OUTPUT EXTRACTION

$s_3 = G_I$ $s_4 = G_E$

The difference from context-sensitive Chomsky grammars is that output can be extracted right away after the (auxiliary) root has been expanded

ON L SYSTEMS

- rewriting is synchronously parallel
- like with pure grammars:
  - no terminals
  - start word
  - every word produced is an output word

Our following example is deterministic and has context window (0,1):

AN L SYSTEM

- start word: ABA
- productions:

$(\Lambda, A, A) \to B$

$(\Lambda, A, B) \to AB$

$(\Lambda, A, \Lambda) \to A$

$(\Lambda, B, A) \to A$

$(\Lambda, B, B) \to A$

$(\Lambda, B, \Lambda) \to A$

ABA

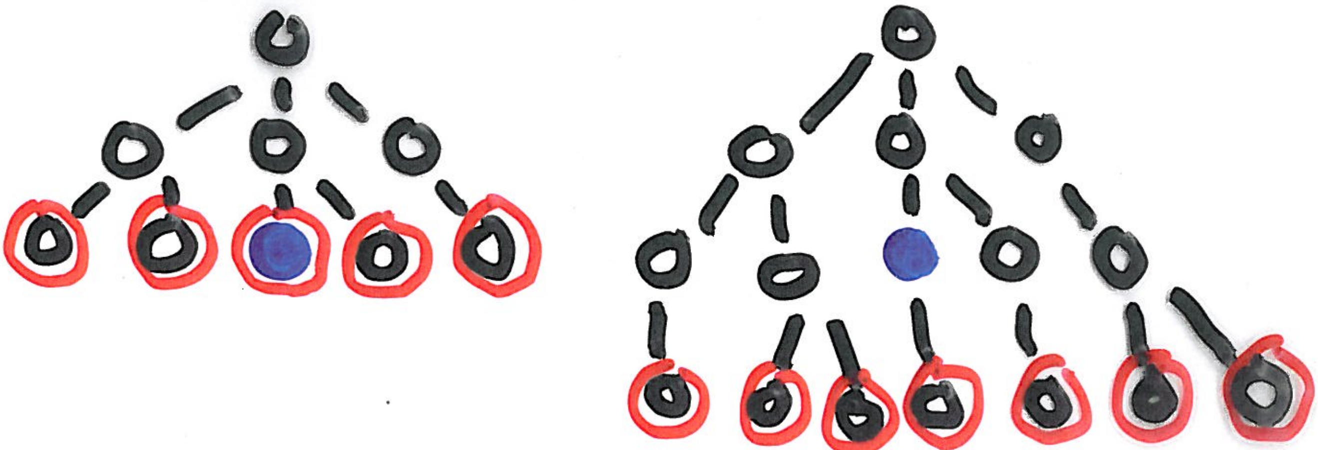
ABAA

ABAAA

ABAAAA

← The first four output words!

(L) FRAME FOR L SYSTEMS



σ_c $\Delta\sigma_c$

- $s_1 = s_2 = \sigma_c$
- $s_3 = s_4 = \Delta\sigma_c$
  (if we used σ_c, then the belt consisting of the (auxiliary) root would not be eliminated)