Chomsky hierarchy

of grammars

1 Noam Chomsky

- born 1928 -
- a linguist
- studied grammars
- among others: context free grammars
- studied languages as purely syntactical objects: sets of strings derived by a grammar.
- at the time this was a sensational abstraction

2 Grammars

like context free grammars but:

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- more general $\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$ to put it mildly, as we shall see..

  finite set of productions. We write

  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$

  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write

  $$u \rightarrow v_1 | \ldots | v_s$$

- $S \in N$ start symbol

2 Grammars

like context free grammars but:

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$
  finite set of productions. We write
  $$u \to_G v \text{ instead of } (u, v) \in P$$
  For $u \to v_1, \ldots, u \to v_s$ we write
  $$u \to v_1 | \ldots | v_s$$
- $S \in N$ start symbol

def: extend productions to strings For $a, b \in (N \cup T)^*$ we define a directly derives b in grammar G

$$a \to_G b$$

if we can decompose

$$a = xuy \quad , b = xvy$$

and

$$u \to_G v$$

like context free grammars but:

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$

  finite set of productions. We write

  $$u \to_G v \text{ instead of } (u, v) \in P$$

  For $u \to v_1, \ldots, u \to v_s$ we write

  $$u \to v_1 | \ldots | v_s$$

- $S \in N$ start symbol

def: extend productions to strings For $a, b \in (N \cup T)^*$ we define a directly derives b in grammar G

$$a \to_G b$$

if we can decompose

$$a = xuy \quad , b = xvy$$

and

$$u \to_G v$$

def: derivation A sequence of words $(w_1, \ldots, w_n)$ with $w_i \in (N \cup T)^*$ is a derivation from w_1 to w_n in G if for all $i < n$:

$$w_i \to_G w_{i+1}$$

We write

$$w_1 \to_G^* w_n$$

def: language generated by G:

$$L(G) = \{w \in T^* \ : S \to_G^* w\}$$

as for cfg's

example (a known context free grammar G)

$$
\begin{aligned}
N &= \{S\} \\
T &= \{a,b\} \\
S &= S \\
S &\to ab|aSb
\end{aligned}
$$

$$S \to aSb \to aaSbb \to aaaSbbb$$

$$L(G) = \{a^n b^n \ : \ n \in \mathbb{N}\}$$

def: extend productions to strings For $a,b \in (N \cup T)^*$ we define a directly derives b in grammar G

$$a \to_G b$$

if we can decompose

$$a = xuy \quad ,b = xvy$$

and

$$u \to_G v$$

def: derivation A sequence of words $(w_1, \ldots, w_n)$ with $w_i \in (N \cup T)^*$ is a derivation from w_1 to w_n in G if for all $i < n$:

$$w_i \to_G w_{i+1}$$

We write

$$w_1 \to_G^* w_n$$

def: language generated by G:

$$L(G) = \{w \in T^* \ : S \to_G^* w\}$$

as for cfg's

def: extend productions to strings For $a,b \in (N \cup T)^*$ we define a directly derives b in grammar G

$$a \to_G b$$

if we can decompose

$$a = xuy \quad ,b = xvy$$

and

$$u \to_G v$$

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^{+} \times (N \cup T)^{*}}$$

  finite set of productions. We write

  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$

  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write

  $$u \rightarrow v_1 | \ldots | v_s$$

- $S \in N$ start symbol

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^{+} \times (N \cup T)^{*}}$$
  finite set of productions. We write
  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$
  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write
  $$u \rightarrow v_1 | \ldots | v_s$$
- $S \in N$ start symbol

Restriction on productions $u \rightarrow v$:

- type-0: no restriction
- type-1 (context sensitive grammar, csg)
  $$|u| \leq |v|$$
  right hand sides is not get shorter than left hand side
- type-2 (context free grammar, cfg)
  $$u \in N \ \wedge |u| \leq |v|$$
  left hand side is single non terminal, right hand side is not empty
- type-3 (regular grammar, rg)
  $$u \in N \wedge (u \rightarrow aB \vee u \rightarrow a) \quad \text{with } a \in T \ , \ B \in N$$
  Very special case of cfg

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$

  finite set of productions. We write

  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$

  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write

  $$u \rightarrow v_1 | \ldots | v_s$$

- $S \in N$ start symbol

Restriction on productions $u \rightarrow v$:

- type-0: no restriction
- type-1 (context sensitive grammar, csg)

  $$|u| \leq |v|$$

  right hand sides is not get shorter than left hand side
- type-2 (context free grammar, cfg)

  $$u \in N \ \wedge |u| \leq |v|$$

  left hand side is single non terminal, right hand side is not empty
- type-3 (regular grammar, rg)

  $$u \in N \wedge (u \rightarrow aB \vee u \rightarrow a) \quad \text{with } a \in T \text{ , } B \in N$$

  Very special case of cfg

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$

  finite set of productions. We write

  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$

  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write

  $$u \rightarrow v_1 | \ldots | v_s$$

- $S \in N$ start symbol

Restriction on productions $u \rightarrow v$:

- type-0: no restriction
- type-1 (context sensitive grammar, csg)

  $$|u| \leq |v|$$

  right hand sides is not get shorter than left hand side
- type-2 (context free grammar, cfg)

  $$u \in N \ \wedge |u| \leq |v|$$

  left hand side is single non terminal, right hand side is not empty
- type-3 (regular grammar, rg) for the time being

  $$u \in N \wedge (u \rightarrow aB \vee u \rightarrow a) \quad \text{with } a \in T \ , \ B \in N$$

  Very special case of cfg

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$

  finite set of productions. We write

  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$

  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write

  $$u \rightarrow v_1 | \ldots | v_s$$

- $S \in N$ start symbol

Restriction on productions $u \rightarrow v$:

- type-0: no restriction
- type-1 (context sensitive grammar, csg)

  $$|u| \leq |v|$$

  right hand sides is not get shorter than left hand side
- type-2 (context free grammar, cfg)

  $$u \in N \ \wedge |u| \leq |v|$$

  left hand side is single non terminal, right hand side is not empty
- type-3 (regular grammar, rg)

  $$u \in N \wedge (u \rightarrow aB \vee u \rightarrow a) \quad \text{with } a \in T \ , \ B \in N$$

  Very special case of cfg

3 The Chomsky hierarchy of grammars

def: grammar

$$G = (N, T, P, S)$$

- N finite set, alphabet of non terminals
- T finite set with $N \cap T = \emptyset$. Alphabet of terminals.
- $$\boxed{P \subset (N \cup T)^+ \times (N \cup T)^*}$$
  finite set of productions. We write
  $$u \rightarrow_G v \text{ instead of } (u, v) \in P$$
  For $u \rightarrow v_1, \ldots, u \rightarrow v_s$ we write
  $$u \rightarrow v_1 | \ldots | v_s$$
- $S \in N$ start symbol

Restriction on productions $u \rightarrow v$:

- type-0: no restriction
- type-1 (context sensitive grammar, csg)
  $$|u| \leq |v|$$
  right hand sides is not get shorter than left hand side
- type-2 (context free grammar, cfg)
  $$u \in N \ \wedge |u| \leq |v|$$
  left hand side is single non terminal, right hand side is not empty
- type-3 (regular grammar, rg)
  $$u \in N \wedge (u \rightarrow aB \ \vee u \rightarrow a) \quad \text{with } a \in T \ , \ B \in N$$
  Very special case of cfg

Lemma 1. *Let G be a cfg, $B \in N$ and $w \in (N \cup T)^+$. Then there is a derivation tree for G with root B and border word w if and only if (iff)*

$$B \rightarrow_G^* w$$

Proof. exercise □

Thus definition of $L(G)$ with derivation trees and with derivations give the same result.

4 Deriving the empty word

- so far only possible for type-0 grammars
- we wish to characterize classes of languages L by machine models, which accept L, i.e. with $L(M) = L$.
- if start state z_0 of machine M is accepting ($z_0 \in Z_A$), then M accepts the empty word:

$$\varepsilon \in L(M)$$

4 Deriving the empty word

- so far only possible for type-0 grammars

- we wish to characterize classes of languages L by machine models, which accept L, i.e. with $L(M) = L$.

- if start state z_0 of machine M is accepting ($z_0 \in Z_A$), then M accepts the empty word:

$$\varepsilon \in L(M)$$

extending csg, cfg,rg for this purpose

Lemma 2. *Let $G = (N,T,P,S)$ be a csg, cfg or rg. Then G can be transformed to grammar G' such that the start symbol never appears on the right hand side of a production and such that $L(G) = L(G')$.*

Proof.
- new start symbol S'
- add production $S' \to S$.

4 Deriving the empty word

- so far only possible for type-0 grammars
- we wish to characterize classes of languages L by machine models, which accept L, i.e. with $L(M) = L$.
- if start state z_0 of machine M is accepting ($z_0 \in Z_A$), then M accepts the empty word:

$$\varepsilon \in L(M)$$

extending csg, cfg,rg for this purpose

Lemma 2. *Let $G = (N, T, P, S)$ be a csg, cfg or rg. Then G can be transformed to grammar G' such that the start symbol never appears on the right hand side of a production and such that $L(G) = L(G')$.*

Proof.

- new start symbol S'
- add production $S' \to S$.

new production allowed: If start symbol S' does not appear on right hand side of any production allow

$$S' \to \varepsilon$$

exciting ... *however:*

5 Finite automata accept exactly the languages generated by regular grammars

Lemma 3. *If L is generated by a regular grammar, then it is accepted by an nfa M*

Let

$$G = (N, T, P, S)$$

be a regular grammar generating L, i.e. $L = L(G)$. Derivation trees of words $a_1 \dots a_n$ have the form shown in figure 1.

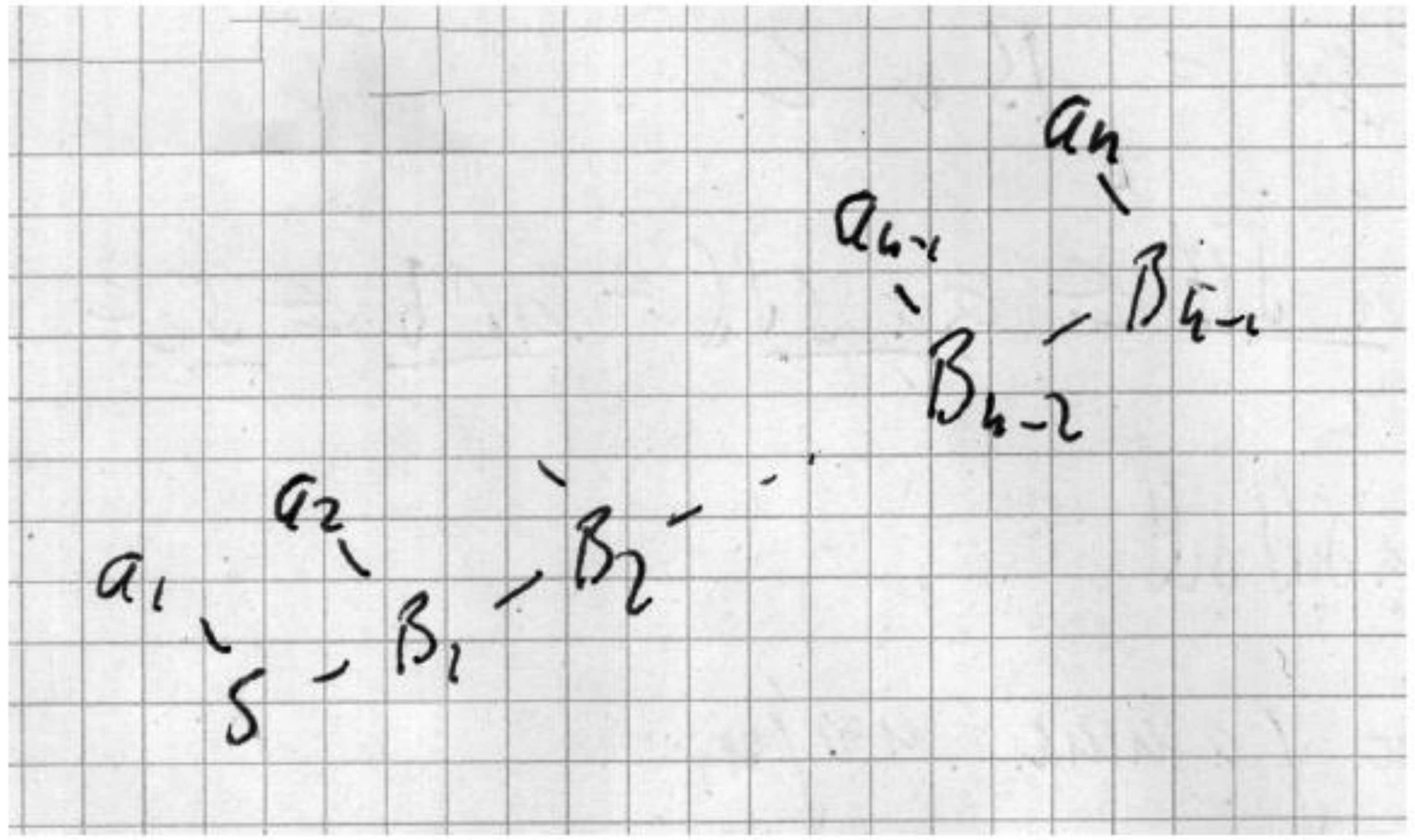


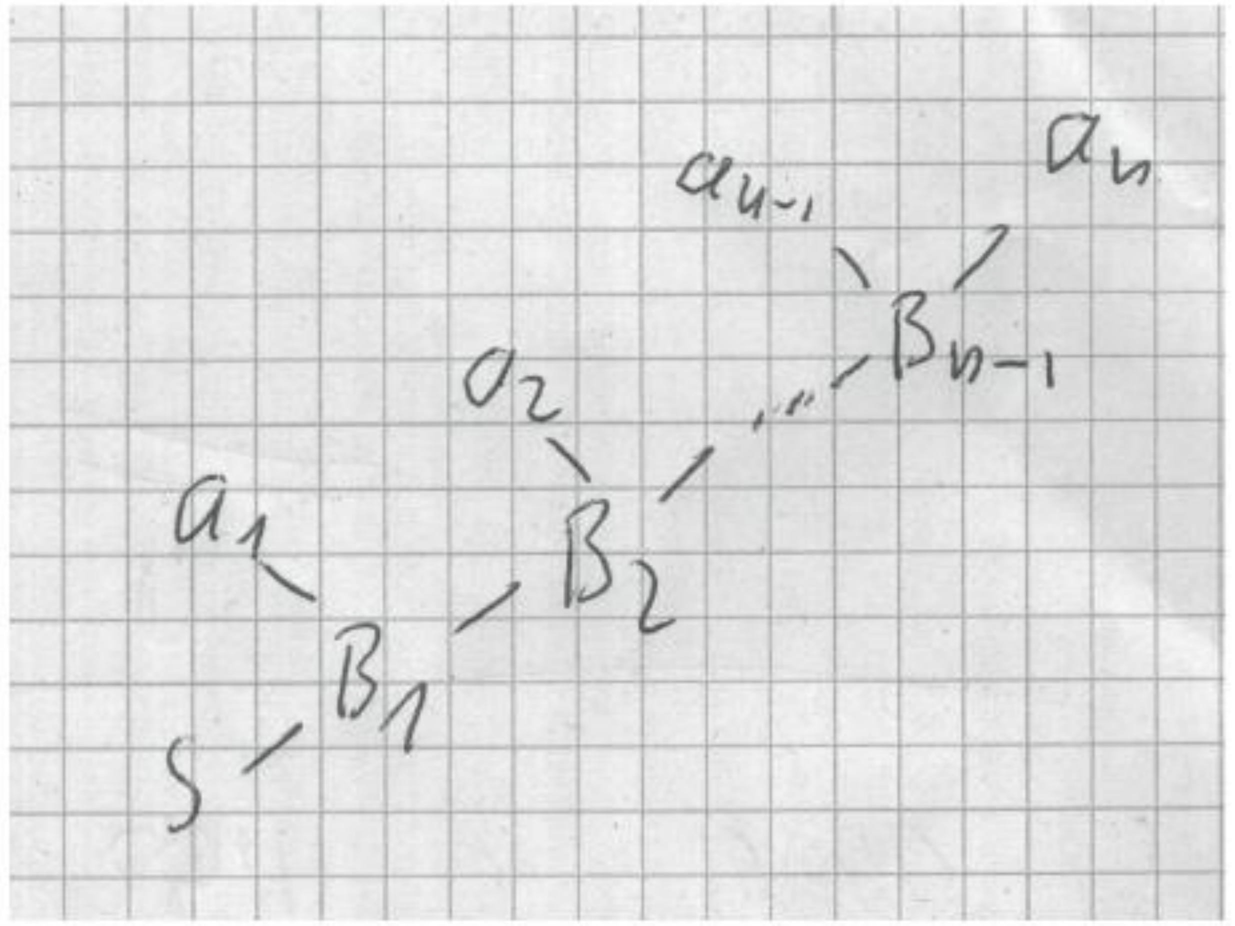
Figure 1: Derivation tree for $w = a_1 \dots w_n$ in a regular grammar G

Lemma 3. *If L is generated by a regular grammar, then it is accepted by an nfa M*

Let

$$G = (N, T, P, S)$$

be a regular grammar generating L, i.e. $L = L(G)$. Derivation trees of words $a_1 \dots a_n$ have the form shown in figure 2.



Construct $M = (Z, T, \delta, z_0, Z_A)$ by

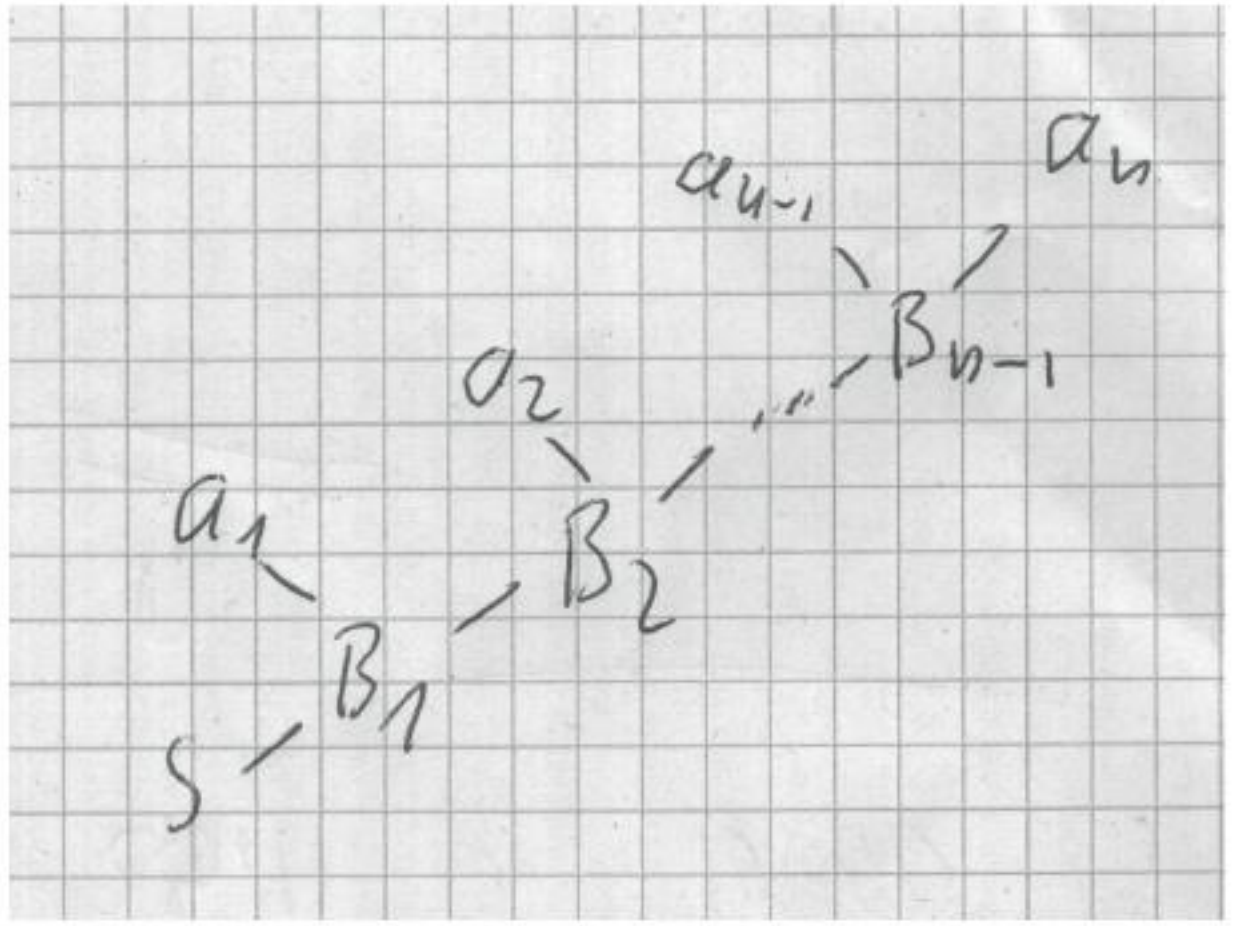
$$\begin{aligned}
Z &= N \cup \{A\} \quad \text{(new accepting state), } A \notin N \\
z_0 &= S \\
Z_A &= \begin{cases} \{A\} & \varepsilon \notin L \\ \{S, A\} & \varepsilon \in L \end{cases} \\
B \in \delta(N, a) &\leftrightarrow N \rightarrow aB \\
& \vee N \rightarrow a \wedge B = A
\end{aligned}$$

Lemma 3. *If L is generated by a regular grammar, then it is accepted by an nfa M*

Let

$$G = (N, T, P, S)$$

be a regular grammar generating L, i.e. $L = L(G)$. Derivation trees of words $a_1 \dots a_n$ have the form shown in figure 2.



Construct $M = (Z, T, \delta, z_0, Z_A)$ by

$$\begin{aligned}
Z &= N \cup \{A\} \quad \text{(new accepting state), } A \notin N \\
z_0 &= S \\
Z_A &= \begin{cases} \{A\} & \varepsilon \notin L \\ \{S, A\} & \varepsilon \in L \end{cases} \\
B \in \delta(N, a) &\leftrightarrow N \rightarrow aB \\
&\quad \vee N \rightarrow a \wedge B = A
\end{aligned}$$

Let $w = a_1 \dots a_n$

- if $w \in L(G)$, then there is derivation tree as in figure 2 and an accepting computation of M with input w and states

$$(S, B_1, \dots, B_{n-1}, A)$$

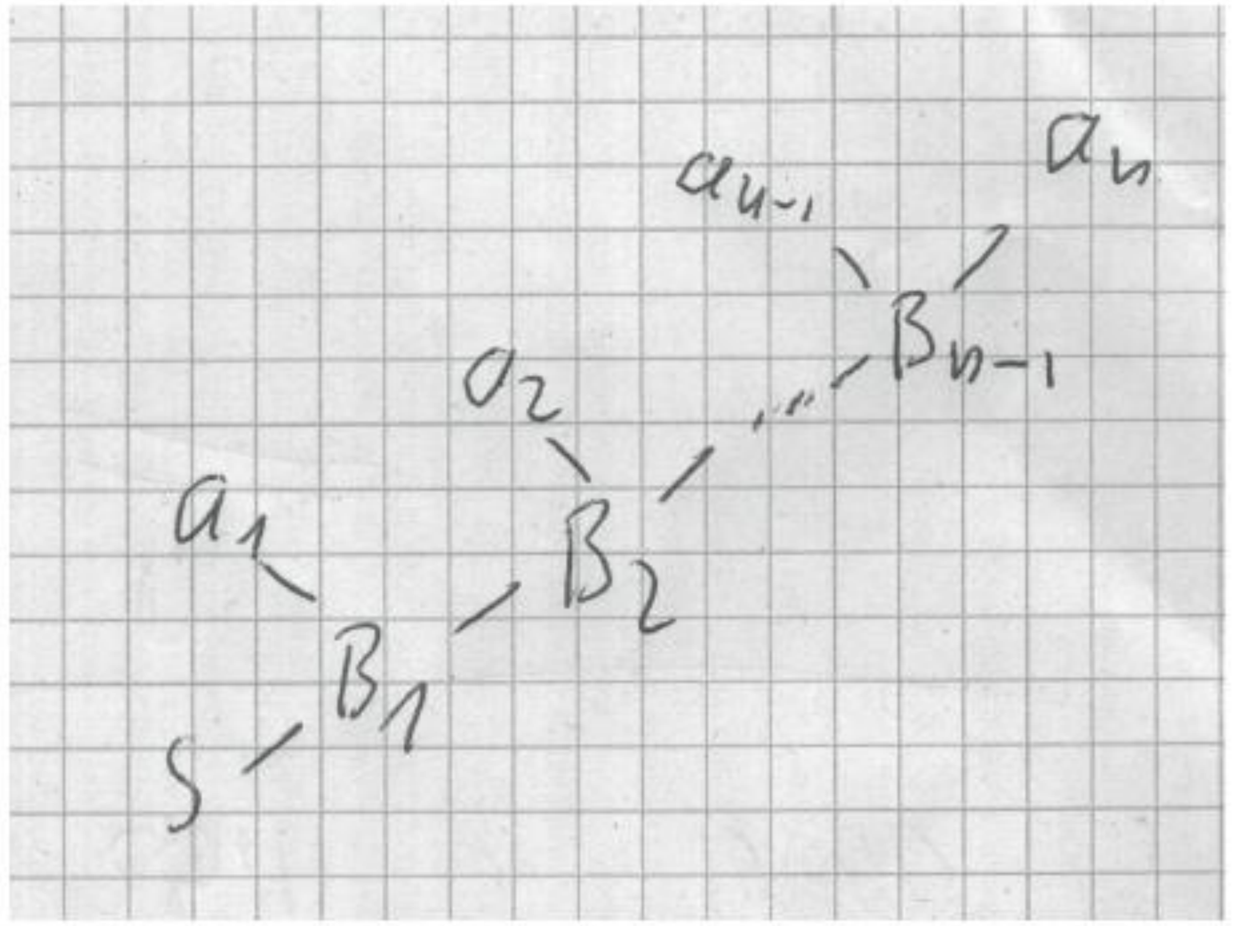
thus $w \in L(M)$.

Lemma 3. *If L is generated by a regular grammar, then it is accepted by an nfa M*

Let

$$G = (N, T, P, S)$$

be a regular grammar generating L, i.e. $L = L(G)$. Derivation trees of words $a_1 \dots a_n$ have the form shown in figure 2.



Construct $M = (Z, T, \delta, z_0, Z_A)$ by

$$\begin{aligned}
Z &= N \cup \{A\} \quad \text{(new accepting state), } A \notin N \\
z_0 &= S \\
Z_A &= \begin{cases} \{A\} & \varepsilon \notin L \\ \{S, A\} & \varepsilon \in L \end{cases} \\
B \in \delta(N, a) &\leftrightarrow N \to aB \\
& \quad \vee N \to a \wedge B = A
\end{aligned}$$

Let $w = a_1 \dots a_n$

- if $w \in L(G)$, then there is derivation tree as in figure 2 and an accepting computation of M with input w and states

  $$(S, B_1, \dots, B_{n-1}, A)$$

  thus $w \in L(M)$.

- If $w \in L(M)$, then there is an accepting computation of M with input w and states $S, B_1, \dots, B_{n-1}, A$. Then figure 2 is a derivation tree for w in G. Thus $w \in L(G)$.

Lemma 4. *If* $M = (Z, T, \delta, z_0, Z_A)$ *be a dfa accepting* L*, i.e.* $L = L(M)$*. Then there is a regular grammar* G *with* $L(M) = L(G)$*.*

Lemma 4. *If* $M = (Z, T, \delta, z_0, Z_A)$ *be a dfa accepting* L*, i.e.* $L = L(M)$*. Then there is a regular grammar* G *with* $L(M) = L(G)$*.*

Construct regular grammar $G = (Z, T, P, z_0)$ by

$$\begin{aligned} B \to aC &\quad\leftrightarrow\quad \delta(B,a) = C \\ B \to a &\quad\leftrightarrow\quad \delta(B,a) = C \,\wedge\, C \in Z_A \end{aligned}$$

- claim $L(M) = L(G) \setminus \{\varepsilon\}$. Proof: exercise.
- if $z_0 \in Z_A$ we have $\varepsilon \in L$. Transform above grammar as in lemma 2 and then add production $S' \to \varepsilon$.

Lemma 4. *If $M = (Z, T, \delta, z_0, Z_A)$ be a dfa accepting L, i.e. $L = L(M)$. Then there is a regular grammar G with $L(M) = L(G)$.*

Construct regular grammar $G = (Z, T, P, z_0)$ by

$$\begin{aligned} B \to aC \quad &\leftrightarrow \quad \delta(B, a) = C \\ B \to a \quad &\leftrightarrow \quad \delta(B, a) = C \,\wedge\, C \in Z_A \end{aligned}$$

- claim $L(M) = L(G) \setminus \{\varepsilon\}$. Proof: exercise.
- if $z_0 \in Z_A$ we have $\varepsilon \in L$. Transform above grammar as in lemma 2 and then add production $S' \to \varepsilon$.

6 Bottom stage of hierarchy

Lemma 5. *There is a context free language L, which is not regular*

- $$L = \{a^n b^n \;:\; n \in \mathbb{N}\}$$
- context free: above.
- not regular by pumping lemma for regular languages.

7 pumping lemma for context free languages

Lemma 6. *For every context free grammar G there is a constant n_0 such that every word in $W \in L(G)$ with $|w| > n_0$ can be decomposed as*

$$W = uvwxy$$

such that

$$vx \neq \varepsilon$$

and

$$\forall i \in \mathbb{N}_0. \quad uv^i wx^i y \in L(G)$$

7 pumping lemma for context free languages

Lemma 6. *For every context free grammar G there is a constant n_0 such that every word in $W \in L(G)$ with $|w| > n_0$ can be decomposed as*

$$W = uvwxy$$

such that

$$vx \neq \varepsilon$$

and

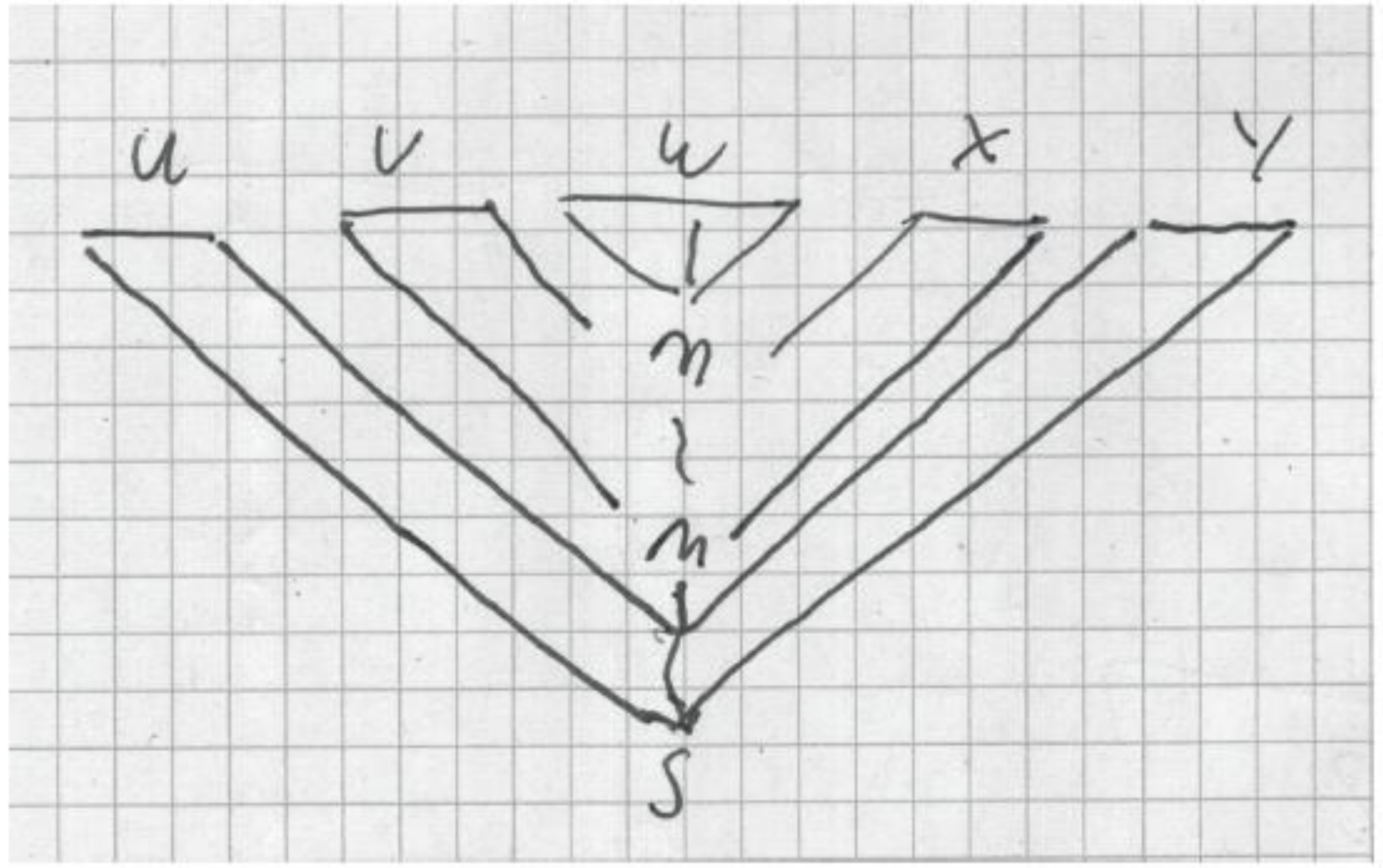
$$\forall i \in \mathbb{N}_0. \quad uv^iwx^iy \in L(G)$$

Let $G = (N,T,P,S)$ and let s be the length of the longest right hand side of a production in P

- set

$$n_0 = s^{\#N}$$

- a derivation tree for G with $|W| > n_0$ leaves has depth $> \#N$
- pidgeon hole argument: there is a path from S to a leaf where a non terminal n is repeated as shown in figure 2. Define vwx as the border word generated by the lower occurrence and w the border word generated by the higher occurrence of n



7 pumping lemma for context free languages

Lemma 6. *For every context free grammar G there is a constant n_0 such that every word in $W \in L(G)$ with $|w| > n_0$ can be decomposed as*

$$W = uvwxy$$

such that

$$vx \neq \varepsilon$$

and

$$\forall i \in \mathbb{N}_0. \quad uv^i wx^i y \in L(G)$$

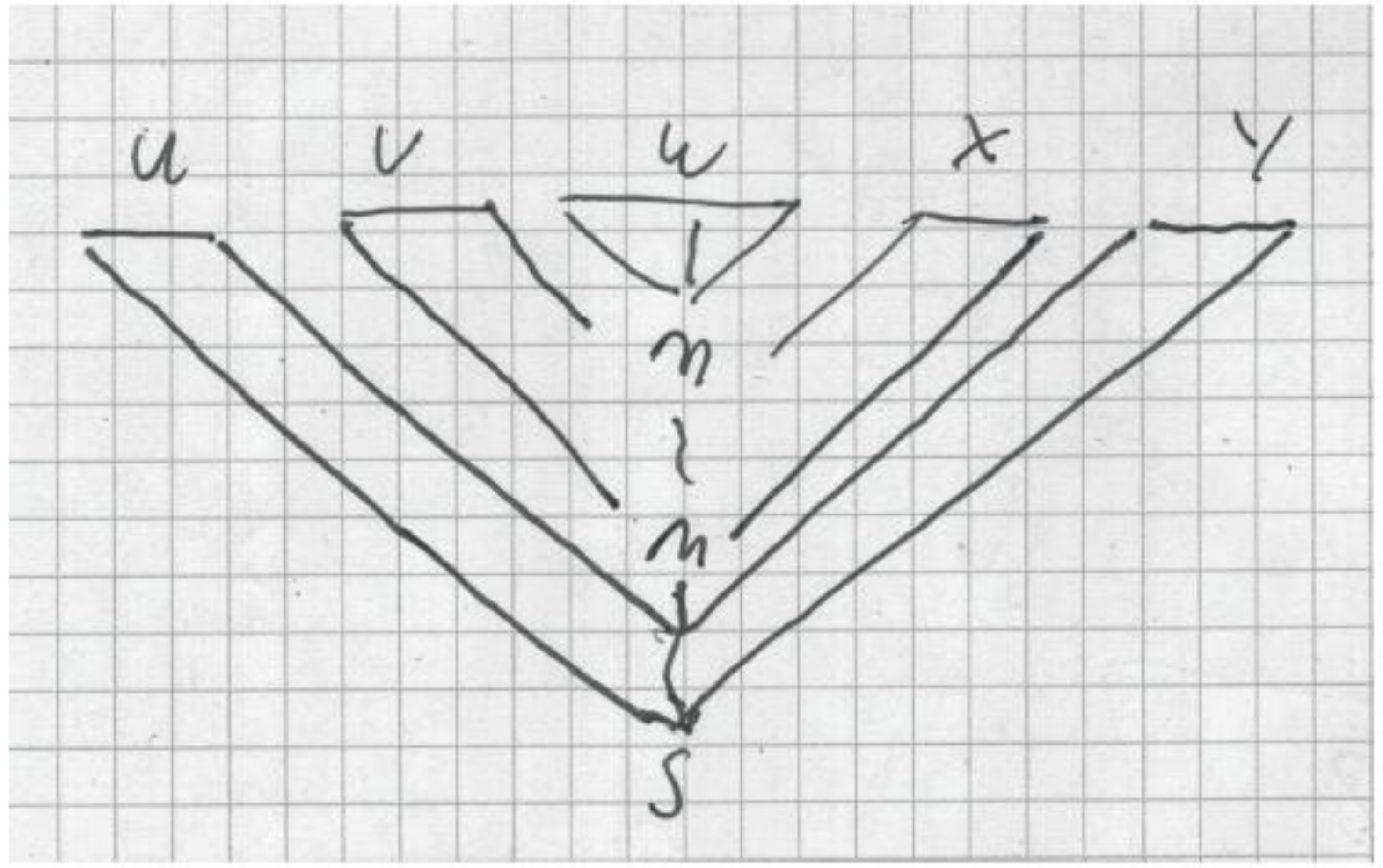
Let $G = (N,T,P,S)$ and let s be the length of the longest right hand side of a production in P

- set

$$n_0 = s^{\#N}$$

- a derivation tree for G with $|W| > n_0$ leaves has depth $> \#N$

- pidgeon hole argument: there is a path from S to a leaf where a non terminal n is repeated as shown in figure 2. Define vwx as the border word generated by the lower occurrence and w the border word generated by the higher occurrence of n



- repeat the portion of the tree generated by the path from n to n exactly i times. For $i = 0$ delete that portion.

7 pumping lemma for context free languages

Lemma 6. *For every context free grammar G there is a constant n_0 such that every word in $W \in L(G)$ with $|w| > n_0$ can be decomposed as*

$$W = uvwxy$$

such that

$$vx \neq \varepsilon$$

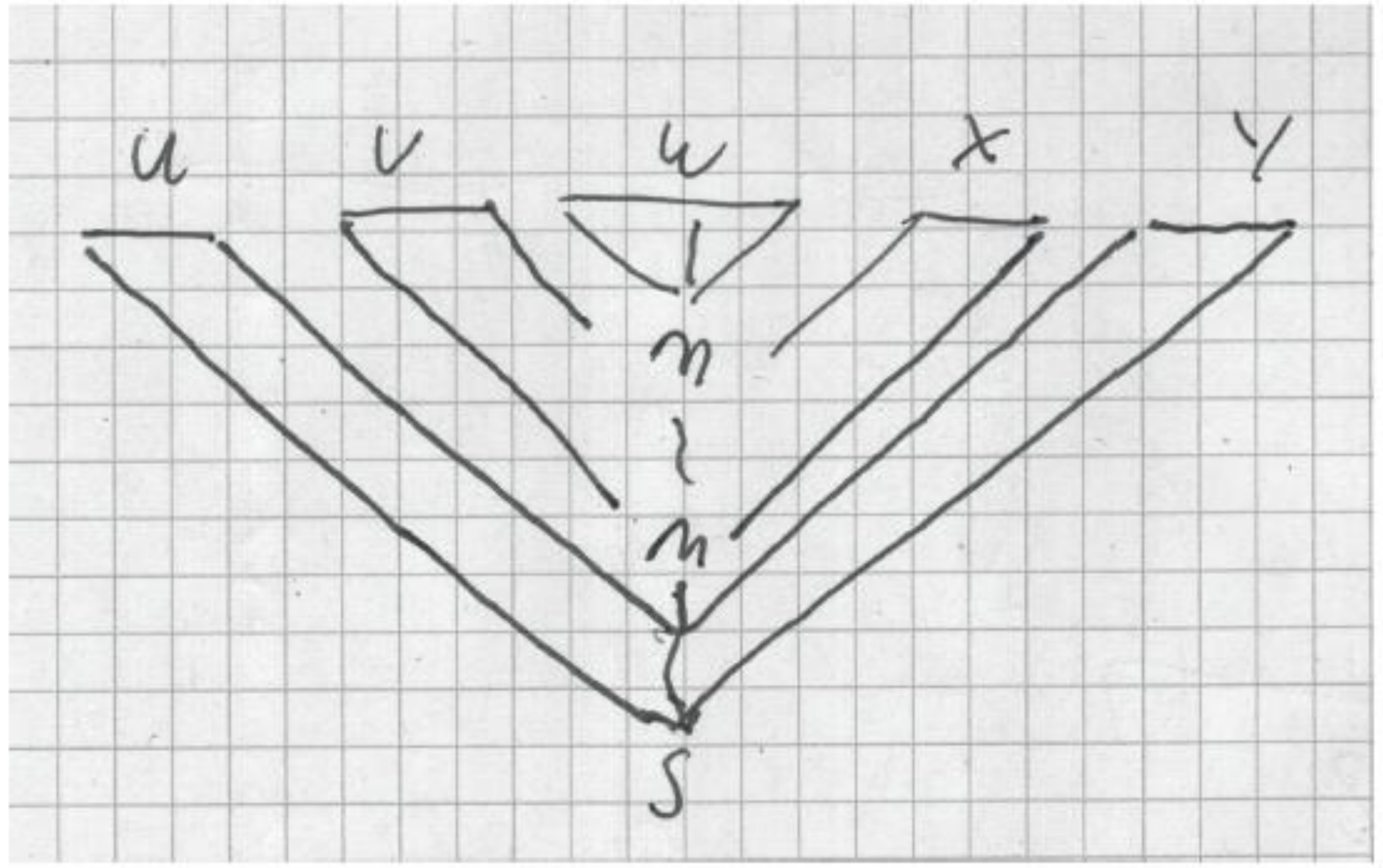
and

$$\forall i \in \mathbb{N}_0. \quad uv^iwx^iy \in L(G)$$

Let $G = (N,T,P,S)$ and let s be the length of the longest right hand side of a production in P

- set
$$n_0 = s^{\#N}$$
- a derivation tree for G with $|W| > n_0$ leaves has depth $> \#N$

- pidgeon hole argument: there is a path from S to a leaf where a non terminal n is repeated as shown in figure 2. Define vwx as the border word generated by the lower occurrence and w the border word generated by the higher occurrence of n



- repeat the portion of the tree generated by the path from n to n exactly i times. For $i = 0$ delete that portion.

Exercise: fix the argument such that

$$vx \neq \epsilon$$

follows

7 pumping lemma for context free languages

Lemma 6. *For every context free grammar G there is a constant n_0 such that every word in $W \in L(G)$ with $|w| > n_0$ can be decomposed as*

$$W = uvwxy$$

such that

$$vx \neq \varepsilon$$

and

$$\forall i \in \mathbb{N}_0. \quad uv^iwx^iy \in L(G)$$

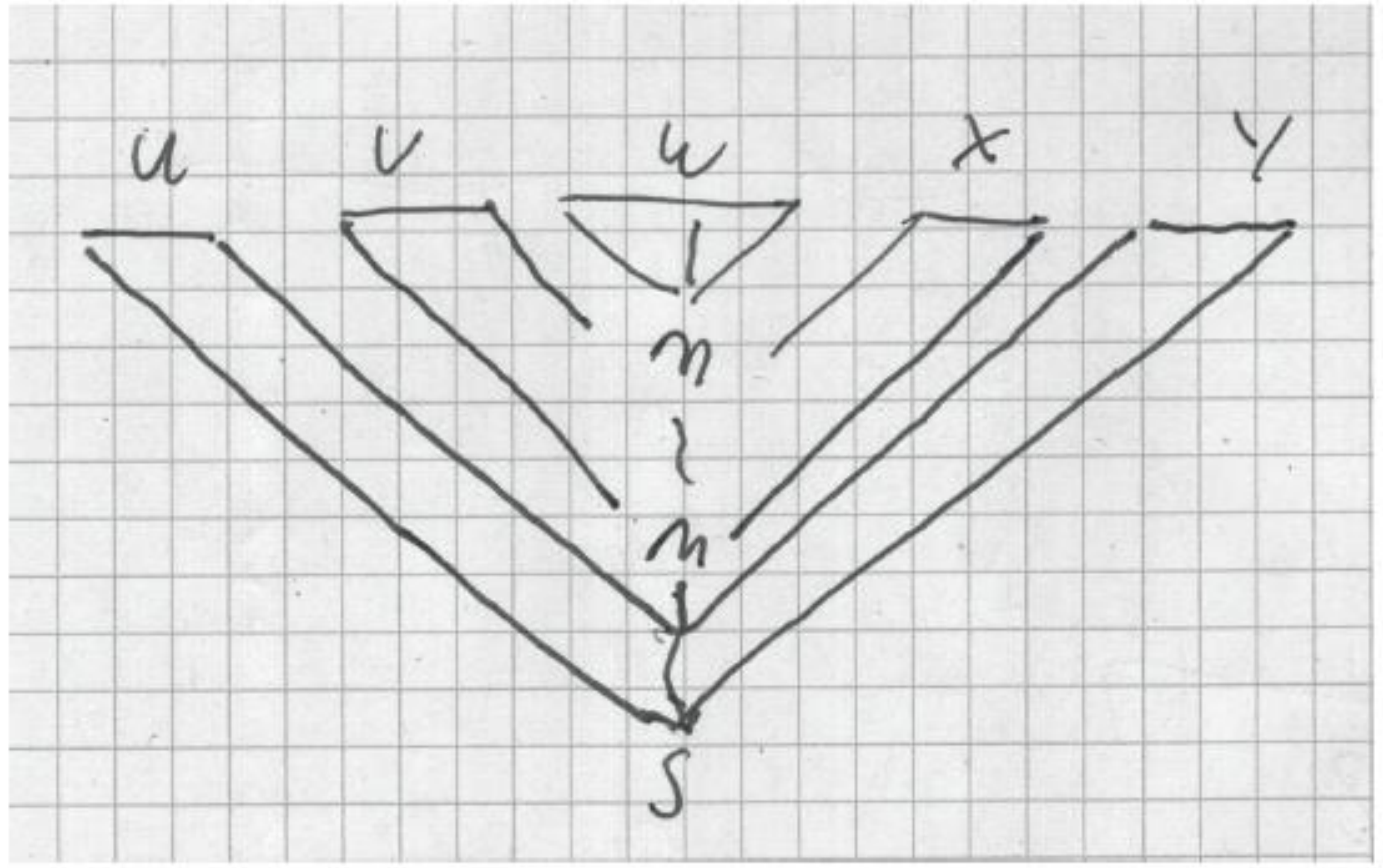
Let $G = (N, T, P, S)$ and let s be the length of the longest right hand side of a production in P

- set

$$n_0 = s^{\#N}$$

- a derivation tree for G with $|W| > n_0$ leaves has depth $> \#N$

- pidgeon hole argument: there is a path from S to a leaf where a non terminal n is repeated as shown in figure 2. Define vwx as the border word generated by the lower occurrence and w the border word generated by the higher occurrence of n



- repeat the portion of the tree generated by the path from n to n exactly i times. For $i = 0$ delete that portion.

Lemma 7. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is not context free

Proof. exercise ☐

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is context sensitive

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \ , \ T = \{a, b, c\}$$

Productions:

- special case abc:

$$S \to abc|ABCT$$

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \ , \ T = \{a, b, c\}$$

Productions:

- special case abc:

$$S \to abc | ABCT$$

- creating $i - 1 \geq 0$ groups of ABC followed by ABc

$$T \to ABCT \mid ABc$$

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \ , \ T = \{a, b, c\}$$

Productions:

- special case abc:

$$S \to abc|ABCT$$

- creating $i-1 \geq 0$ groups of ABC followed by ABc

$$T \to ABCT \mid ABc$$

- shifting C's right of A's and B's:

$$CA \to AC \ , \ CB \to BC$$

- shifting B's right of A's:

$$BA \to AB$$

- converting non terminals to terminals from right to left

$$Cc \to cc \ , \ Bc \to bc \ , \ Bb \to bb \ , \ Ab \to ab \ , \ Aa \to aa$$

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \; : \; n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \; , \; T = \{a, b, c\}$$

this is kind of computing
with the grammar

Productions:

- special case abc:

$$S \to abc|ABCT$$

- creating $i - 1 \geq 0$ groups of ABC followed by ABc

$$T \to ABCT \mid ABc$$

- shifting C's right of A's and B's:

$$CA \to AC \; , \; CB \to BC$$

- shifting B's right of A's:

$$BA \to AB$$

- converting non terminals to terminals from right to left

$$Cc \to cc \; , \; Bc \to bc \; , \; Bb \to bb \; , \; Ab \to ab \; , \; Aa \to aa$$

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \ : \ n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \ , \ T = \{a, b, c\}$$

this is kind of computing
with the grammar

Productions:

- special case abc:

$$S \to abc | ABCT$$

$L(G) = L$: exercise. interesting

- creating $i - 1 \geq 0$ groups of ABC followed by ABc

$$T \to ABCT \mid ABc$$

- shifting C's right of A's and B's:

$$CA \to AC \ , \ CB \to BC$$

- shifting B's right of A's:

$$BA \to AB$$

- converting non terminals to terminals from right to left

$$Cc \to cc \ , \ Bc \to bc \ , \ Bb \to bb \ , \ Ab \to ab \ , \ Aa \to aa$$

8 Next stage of hierarchy

Lemma 8. *The language*

$$L = \{a^n b^n c^n \; : \; n \in N\}$$

is context sensitive

$$N = \{S, T, A, B, C\} \; , \; T = \{a, b, c\}$$

Productions:

- special case *abc*:

$$S \to abc|ABCT$$

- creating $i - 1 \geq 0$ groups of ABC followed by ABc

$$T \to ABCT \mid ABc$$

- shifting C's right of A's and B's:

$$CA \to AC \; , \; CB \to BC$$

- shifting B's right of A's:

$$BA \to AB$$

- converting non terminals to terminals from right to left

$$Cc \to cc \; , \; Bc \to bc \; , \; Bb \to bb \; , \; Ab \to ab \; , \; Aa \to aa$$

this is kind of computing with the grammar

$L(G) = L$: exercise.

interesting

finding a type-0 language which is not context sensitive:

a different world!