context free languages and pushdown automata

on the abstract side but…wait!

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \to \varepsilon \ , \ A \in N$$

allowed

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \rightarrow \varepsilon \ , \ A \in N$$

allowed

def: cfg in Chomsky normal form Productions have the form

- $A \rightarrow BC$ with $A, B, C \in N$
- $A \rightarrow b$ with $A \in N$ and $b \in T$
- $S \rightarrow \varepsilon$

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \to \varepsilon \ , \ A \in N$$

allowed

def: cfg in Chomsky normal form Productions have the form

- $A \to BC$ with $A, B, C \in N$
- $A \to b$ with $A \in N$ and $b \in T$
- $S \to \varepsilon$

you have almost seen this

where?

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \to \varepsilon\ ,\ A \in N$$

allowed

def: cfg in Chomsky normal form Productions have the form

- $A \to BC$ with $A, B, C \in N$
- $A \to b$ with $A \in N$ and $b \in T$
- $S \to \varepsilon$

Lemma 1. *every cfg G (with new definition) can be transformed into a cfg G' in Chomsky normal form such that $L(G) = L(G')$.*

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \to \varepsilon\ ,\ A \in N$$

allowed

def: cfg in Chomsky normal form Productions have the form

- $A \to BC$ with $A, B, C \in N$
- $A \to b$ with $A \in N$ and $b \in T$
- $S \to \varepsilon$

Lemma 1. *every cfg G (with new definition) can be transformed into a cfg G' in Chomsky normal form such that $L(G) = L(G')$.*

- start symbol does not appear on right: add new start symbol S' and production $S' \to S$.

1 The empty word revisited

def: cfg, more comfortable

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

$$P \subset N \times (N \cup T)^*$$

i.e.

$$A \to \varepsilon \ , \ A \in N$$

allowed

def: cfg in Chomsky normal form Productions have the form

- $A \to BC$ with $A, B, C \in N$
- $A \to b$ with $A \in N$ and $b \in T$
- $S \to \varepsilon$

Lemma 1. *every cfg G (with new definition) can be transformed into a cfg G' in Chomsky normal form such that $L(G) = L(G')$.*

- start symbol does not appear on right: add new start symbol S' and production $S' \to S$.
- eliminate right hand sides with more than 2 symbols: for each production

  $$P: \quad n \to a_1 \dots a_s \; s \geq 2$$

  introduce a new non terminal x and and replace p by

  $$p \to a_1 \dots a_{s-2}x \ , \ x \to a_{s-1}a_s$$

  Repeat until all right hand sides have length 2 or 1.

1 The empty word revisited

- eliminate right hand sides with more than 2 symbols: for each production

$$P: \quad n \to a_1 \ldots a_s \; s \geq 2$$

introduce a new non terminal x and and replace p by

$$p \to a_1 \ldots a_{s-2}x \; , \; x \to a_{s-1}a_s$$

Repeat until all right hand sides have length 2 or 1.

- eliminate ε-rules $A \to \varepsilon$. Order set of nonterminals different from S'

$$N \setminus \{S'\} = \{n_1, \ldots, n_k\}$$

for $i = 1$ to k:
{
if $n_i \to \varepsilon$: i) drop this rule; ii) for each rule with n_i on right side add a rule where each occurrence of n_i is dropped.

$$n \to xn_i \text{ or } n \to n_i x : \quad \text{add} \quad n \to x$$

$$n_k \to n_i n_i \wedge k > i : \quad \text{add} \quad n \to \varepsilon$$

}

1 The empty word revisited

- eliminate right hand sides with more than 2 symbols: for each production

$$P: \quad n \to a_1 \dots a_s \; s \geq 2$$

introduce a new non terminal x and and replace p by

$$p \to a_1 \dots a_{s-2}x \; , \; x \to a_{s-1}a_s$$

Repeat until all right hand sides have length 2 or 1.

- eliminate ε-rules $A \to \varepsilon$. Order set of nonterminals different from S'

$$N \setminus \{S'\} = \{n_1, \dots, n_k\}$$

for $i = 1$ to k:
{
if $n_i \to \varepsilon$: i) drop this rule; ii) for each rule with n_i on right side add a rule where each occurrence of n_i is dropped.

$$n \to xn_i \text{ or } n \to n_i x : \quad \text{add} \quad n \to x$$

$$\underline{n_k \to n_i n_i \wedge k > i} : \quad \text{add} \quad n \to \varepsilon$$

}

Lemma 2. *Let G_i be the grammar after pass i of the loop. Then $L(G) = L(G_i)$.*

induction on i. $i = 0$: $G = G_0$, trivial.

$i - 1 \to i$. not completely trivial!

- Let $w \in L(G)$. By induction hypothesis there is a derivation tree for w in G_{i-1}. *It uses no productions $n_k \to \varepsilon$* for $k < i$. Uses of productions $n_i \to \varepsilon$ are replaced by G_i, thus $w \in L(G_i)$.
- Let $w \in L(G_i)$. Replace in a derivation tree for w in G_i each use of a new production by its original in G_{i-1} and $n_i \to \varepsilon$. Thus $w \in L(G_{i-1}) = L(G)$ by induction hypothesis.

1 The empty word revisited

- eliminate right hand sides with more than 2 symbols: for each production

$$P: \quad n \to a_1 \dots a_s \; s \geq 2$$

introduce a new non terminal x and and replace p by

$$p \to a_1 \dots a_{s-2} x \; , \; x \to a_{s-1} a_s$$

Repeat until all right hand sides have length 2 or 1.

- eliminate ε-rules $A \to \varepsilon$. Order set of nonterminals different from S'

$$N \setminus \{S'\} = \{n_1, \dots, n_k\}$$

for $i = 1$ to k:
{
if $n_i \to \varepsilon$: i) drop this rule; ii) for each rule with n_i on right side add a rule where each occurrence of n_i is dropped.

$$n \to x n_i \text{ or } n \to n_i x : \quad \text{add} \quad n \to x$$

$$n_k \to n_i n_i \wedge k > i : \quad \text{add} \quad n \to \varepsilon$$

}

- for all chain rules between nonterminals $A \to B$ with $A, B \in N$ drop $A \to B$ and for all productions

$$B \to u : \quad \text{add} \quad A \to u$$

repeat until no chain rules between nonterminals are left.

1 The empty word revisited

- eliminate right hand sides with more than 2 symbols: for each production

$$P: \quad n \to a_1 \dots a_s \; s \geq 2$$

  introduce a new non terminal x and and replace p by

$$p \to a_1 \dots a_{s-2}x \; , \; x \to a_{s-1}a_s$$

  Repeat until all right hand sides have length 2 or 1.

- eliminate ε-rules $A \to \varepsilon$. Order set of nonterminals different from S'

$$N \setminus \{S'\} = \{n_1, \dots, n_k\}$$

  for $i = 1$ to k:
  {
  if $n_i \to \varepsilon$: i) drop this rule; ii) for each rule with n_i on right side add a rule where each occurrence of n_i is dropped.

$$n \to xn_i \text{ or } n \to n_i x : \quad \text{add} \quad n \to x$$

$$n_k \to n_i n_i \; \wedge \; k > i : \quad \text{add} \quad n \to \varepsilon$$

  }

- for all chain rules between nonterminals $A \to B$ with $A, B \in N$ drop $A \to B$ and for all productions

$$B \to u : \quad \text{add} \quad A \to u$$

  repeat until no chain rules between nonterminals are left.

- replacing for $A, B, C \in N$ and $b, c \in T$

$$A \to bC \; , \; a \to Bc \; , \; A \to bc$$

  exercise

1 The empty word revisited

- eliminate right hand sides with more than 2 symbols: for each production

$$P: \quad n \to a_1 \dots a_s \; s \geq 2$$

introduce a new non terminal x and and replace p by

$$p \to a_1 \dots a_{s-2}x \; , \; x \to a_{s-1}a_s$$

Repeat until all right hand sides have length 2 or 1.

- eliminate ε-rules $A \to \varepsilon$. Order set of nonterminals different from S'

$$N \setminus \{S'\} = \{n_1, \dots, n_k\}$$

for $i = 1$ to k:
{
if $n_i \to \varepsilon$: i) drop this rule; ii) for each rule with n_i on right side add a rule where each occurrence of n_i is dropped.

$$n \to xn_i \text{ or } n \to n_i x : \quad \text{add} \quad n \to x$$

$$n_k \to n_i n_i \wedge k > i : \quad \text{add} \quad n \to \varepsilon$$

}

- for all chain rules between nonterminals $A \to B$ with $A, B \in N$ drop $A \to B$ and for all productions

$$B \to u : \quad \text{add} \quad A \to u$$

repeat until no chain rules between nonterminals are left.

- replacing for $A, B, C \in N$ and $b, c \in T$

$$A \to bC \; , \; a \to Bc \; , \; A \to bc$$

exercise

2 Deciding the 'word problem' $w \in L$ for context free languages:

Lemma 3. *For each cf language L there is an algorithm which decides $w \in L$ in time $O(n^3)$ for $n = |w|$.*

Proof. dynamic programing; Younger's algorithm from I2EA □

def: nondeterministic pushdown automaton (npda)

$$M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$$

- Z finite set of states
- Σ finite input alphabet.

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$

  ε ignores current input symbol.

- Γ finite pushdown/stack alphabet

$$\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$$

  Pushdown actions:

$$PA = \{pop\} \cup \{push\,\gamma \;:\; \gamma \in \Gamma_\varepsilon\}$$

  $push(\epsilon)$: no pop on stack

  ε ignores current top symbol of stack.

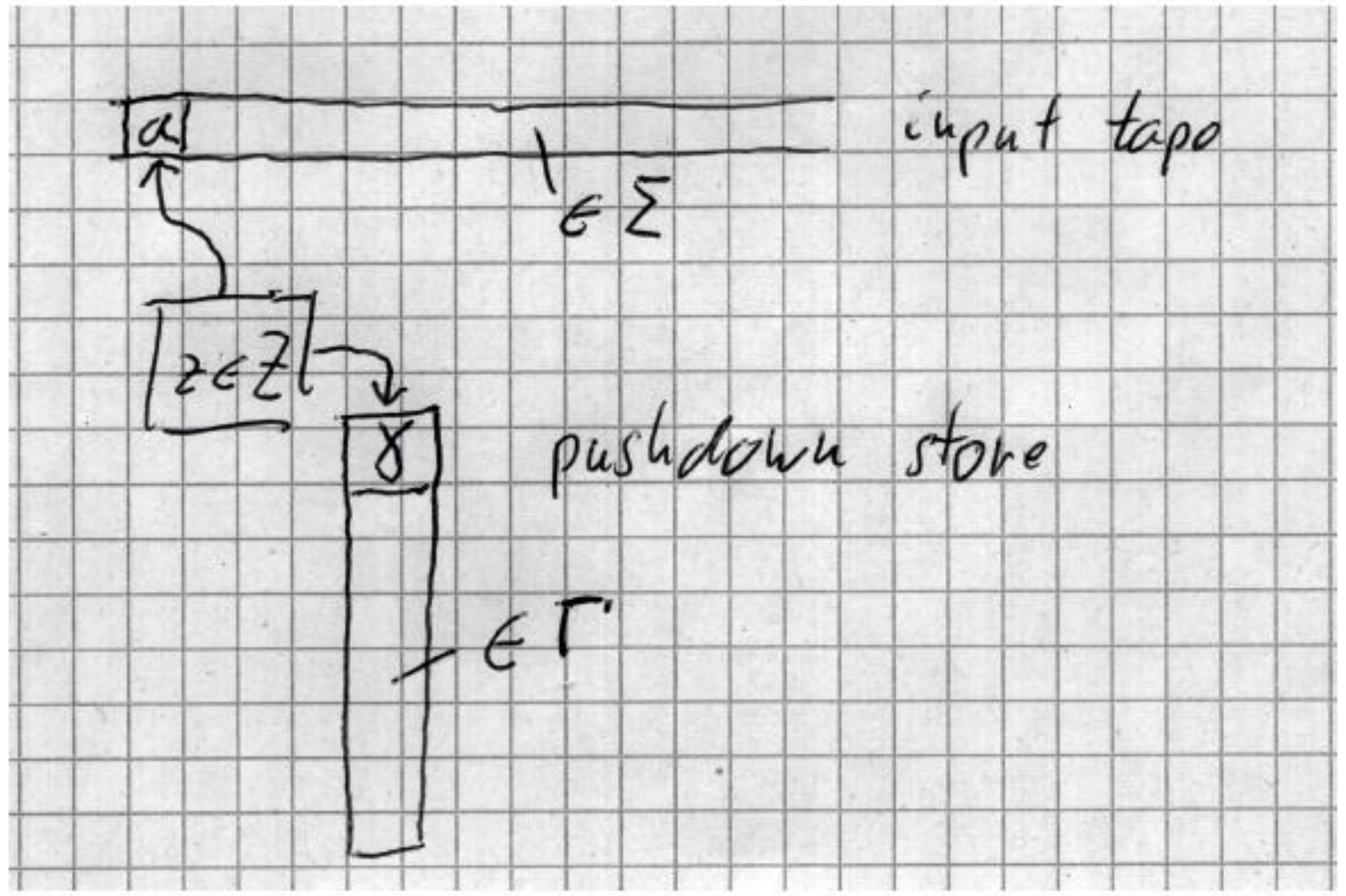
- nondeterministic transition function

$$\delta : Z \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Z \times PA}$$

  1 step: $(z', pa) \in \delta(z, a, \gamma)$ means: if M reads input a and top of stack γ, then it can go to state z' and perform pushdown action pa. It moves the input head to the right of a (if $a = \varepsilon$ the head does not move).

- no pop ignoring (or from empty) stack: $(z', pop) \notin \delta(z, a, \epsilon)$

- $z_0 \in Z$ start state
- $Z_A \subseteq Z$ set of accepting states



def: nondeterministic pushdown automaton (npda)

$$M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$$

- Z finite set of states
- Σ finite input alphabet.

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$

  ε ignores current input symbol.

- Γ finite pushdown/stack alphabet

$$\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$$

  Pushdown actions:

$$PA = \{pop\} \cup \{push\, \gamma \;:\; \gamma \in \Gamma_\varepsilon\}$$

  $push(\epsilon)$: no pop on stack

  ε ignores current top symbol of stack.

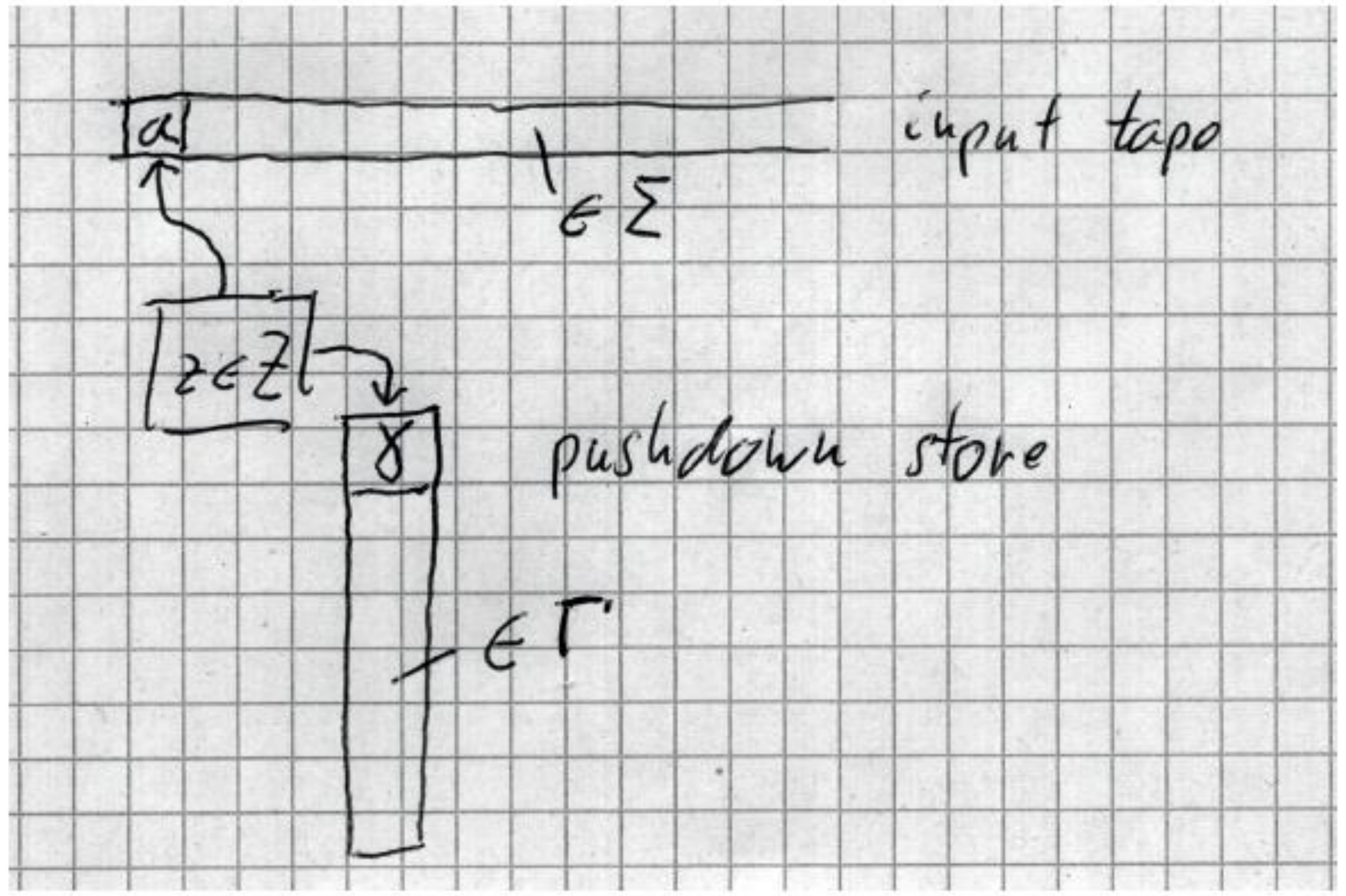
- nondeterministic transition function

$$\delta : Z \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Z \times PA}$$

  1 step: $(z', pa) \in \delta(z, a, \gamma)$ means: if M reads input a and top of stack γ, then it can go to state z' and perform pushdown action pa. It moves the input head to the right of a (if $a = \varepsilon$ the head does not move).

- no pop ignoring (or from empty) stack: $(z', pop) \notin \delta(z, a, \epsilon)$

- $z_0 \in Z$ start state
- $Z_A \subseteq Z$ set of accepting states



def: configurations

$$k = (z, w, p)$$

- $z \in Z$ current state
- $w \in \Sigma^*$ remaining input
- $p \in \Gamma^*$ pushdown tape/stack. From top to bottom $p_1 \dots p_k$

def: initial configuration with input w:

$$k_0 = (z_0, w, \varepsilon)$$

def: possible next/successor configuration

$$k = (z, w, p) \vdash k' = (z', w', p')$$

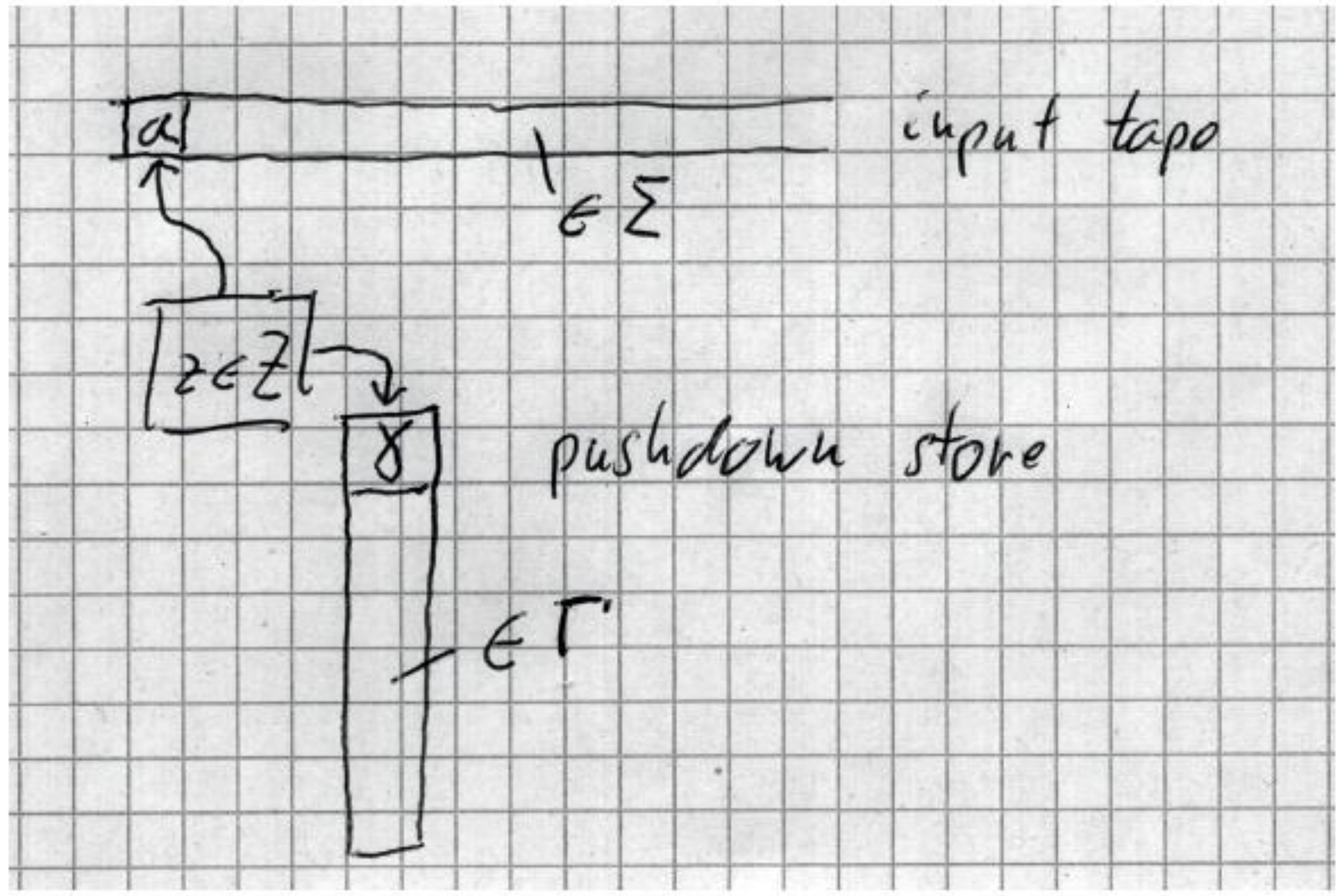
holds if one of the following holds

- input tape not ignored and pop

$$(z', pop) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon) \ , \ w' = tail(w) \ , \ p' = tail(p)$$

- input not ignored and push

$$(z', push\ \gamma) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon) \ , \ w' = tail(w) \ , \ p' = \gamma \circ p$$



def: configurations

$$k = (z, w, p)$$

- $z \in Z$ current state
- $w \in \Sigma^*$ remaining input
- $p \in \Gamma^*$ pushdown tape/stack. From top to bottom $p_1 \dots p_k$

def: initial configuration with input w:

$$k_0 = (z_0, w, \varepsilon)$$

def: possible next/successor configuration

$$k = (z, w, p) \vdash k' = (z', w', p')$$

holds if one of the following holds

- input tape not ignored and pop

  $$(z', pop) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon) \ , \ w' = tail(w) \ , \ p' = tail(p)$$

- input not ignored and push

  $$(z', push\ \gamma) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon) \ , \ w' = tail(w) \ , \ p' = \gamma \circ p$$

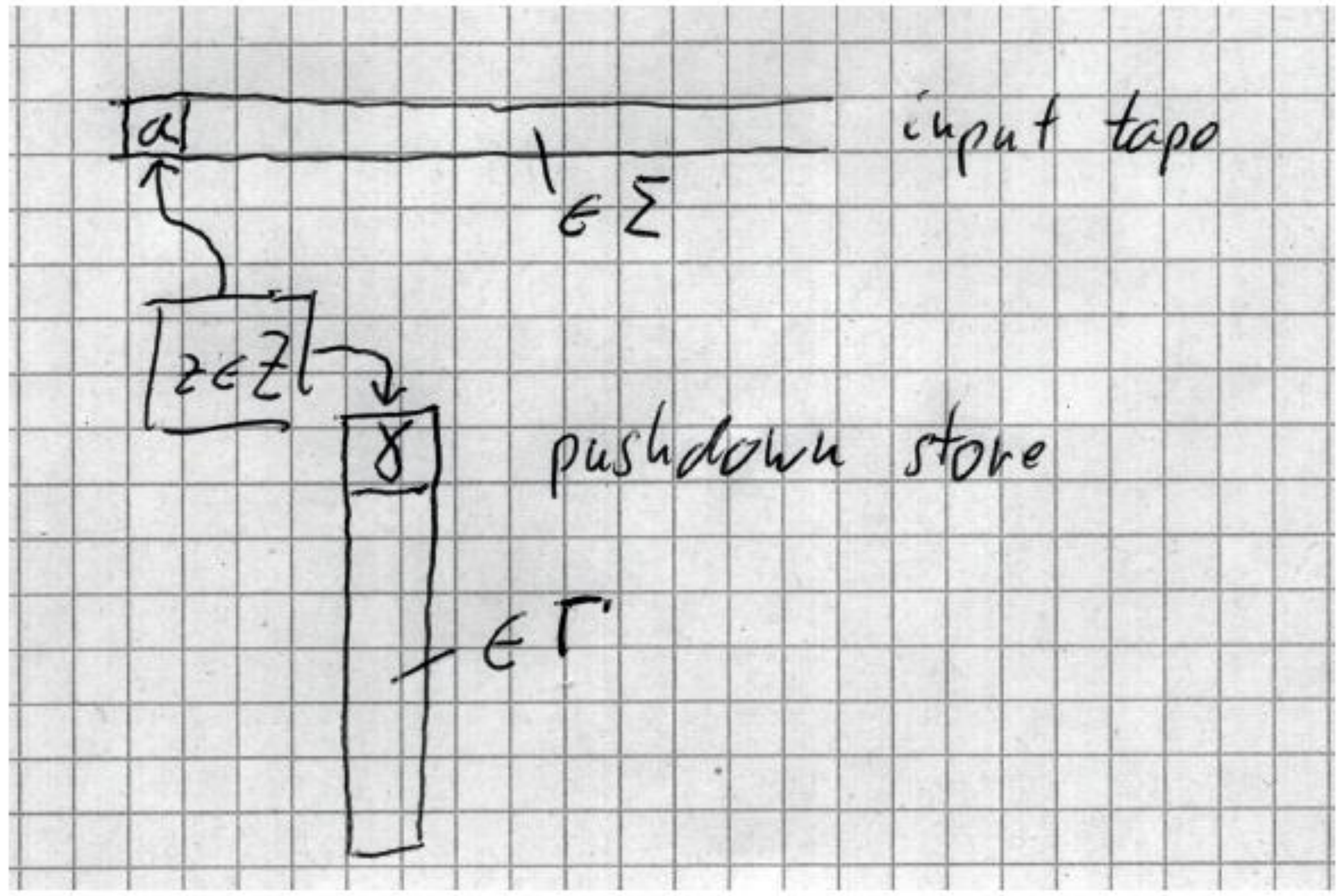
- input ignored and pop

  $$(z', pop) \in \delta(z, \varepsilon, p_1) \cup \delta(z, \varepsilon, \varepsilon) \ , \ w' = w \ , \ p' = tail(p)$$

- input ignored and push

  $$(z', push\ \gamma) \in \delta(z, \varepsilon, p_1) \cup \delta(z, \varepsilon, \varepsilon) \ , \ w' = w \ , \ p' = \gamma \circ p$$

  Pop only possible with $p \neq \varepsilon$.



def: configurations

$$k = (z, w, p)$$

- $z \in Z$ current state
- $w \in \Sigma^*$ remaining input
- $p \in \Gamma^*$ pushdown tape/stack. From top to bottom $p_1 \dots p_k$

def: initial configuration with input w:

$$k_0 = (z_0, w, \varepsilon)$$

def: possible next/successor configuration

$$k = (z, w, p) \vdash k' = (z', w', p')$$

holds if one of the following holds

- input tape not ignored and pop

$$(z', pop) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon)\ ,\ w' = tail(w)\ ,\ p' = tail(p)$$

- input not ignored and push

$$(z', push\ \gamma) \in \delta(z, w_1, p_1) \cup \delta(z, w_1, \varepsilon)\ ,\ w' = tail(w)\ ,\ p' = \gamma \circ p$$

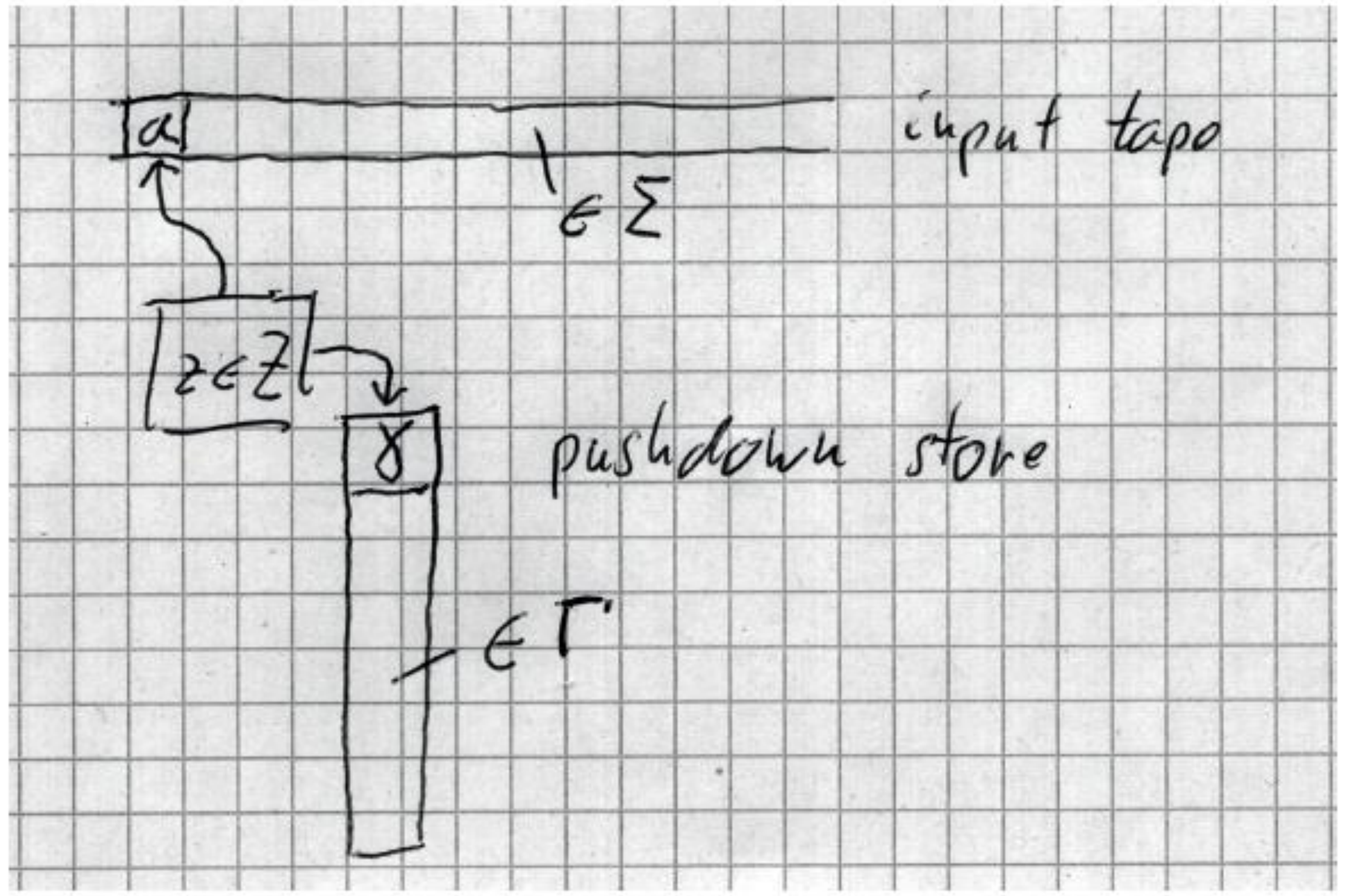
- input ignored and pop

$$(z', pop) \in \delta(z, \varepsilon, p_1) \cup \delta(z, \varepsilon, \varepsilon)\ ,\ w' = w\ ,\ p' = tail(p)$$

- input ignored and push

$$(z', push\ \gamma) \in \delta(z, \varepsilon, p_1) \cup \delta(z, \varepsilon, \varepsilon)\ ,\ w' = w\ ,\ p' = \gamma \circ p$$

  Pop only possible with $p \neq \varepsilon$.



def: computation sequence $(k_0, \ldots k_m)$ with $k_i \vdash k_{i+1}$ for $i < m$. Accepting if state of k_m in Z_A and stack of k_m empty

$$k_m = (q, \varepsilon, \varepsilon)\ ,\ q \in Z_A$$

Variations: i) accept by empty stack ii) by final state in Z_A.

def: $L(M)$ M accepts w if there exists and accepting computation of M started with w.

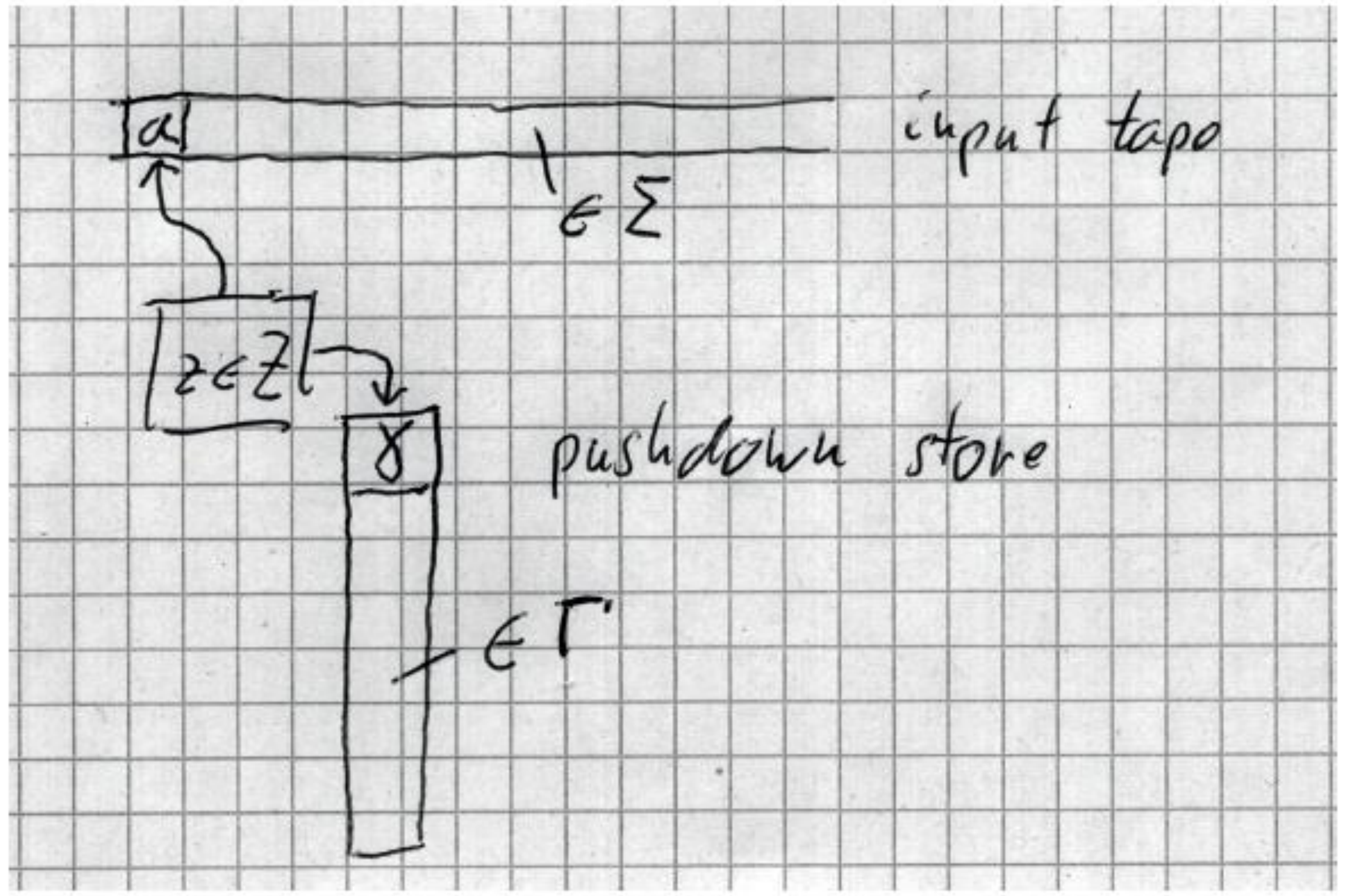
$$L(M) = \{w \in \Sigma^*\ :\ M \text{ accepts } w\}$$

example

$$\begin{aligned} w &\in A^* \\ w &= w_1 \dots w_n \\ w^R &= w_n \dots w_1 \\ L &= \{ww^R \,:\, w \in \mathbb{B}\} \end{aligned}$$

L is context free:

$$S \to 0S0 \mid 1S1 \mid \varepsilon$$

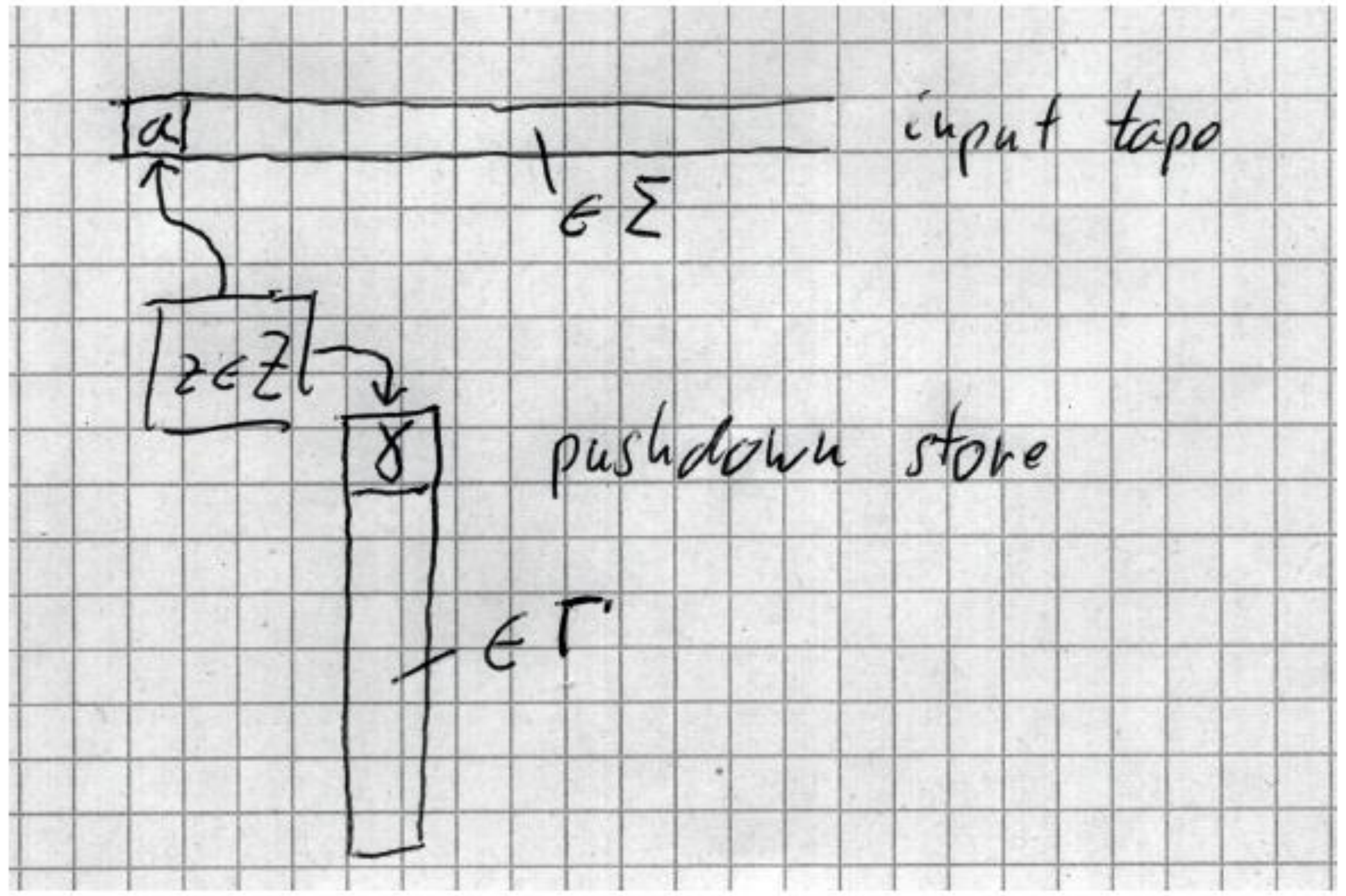


example

$$\begin{aligned} w &\in A^* \\ w &= w_1 \dots w_n \\ w^R &= w_n \dots w_1 \\ L &= \{ww^R : w \in \mathbb{B}\} \end{aligned}$$

L is context free:

$$S \to 0S0 \mid 1S1 \mid \varepsilon$$



pushdown automaton for L:

$$Z = \{z, z'\} \quad , \quad z_0 = z \quad , \quad Z_A = \{z'\}$$

push w

$$\begin{aligned} (z, push\ 0) &\in \delta(z, 0, \varepsilon) \\ (z, push\ 1) &\in \delta(z, 1, \varepsilon) \end{aligned}$$

guess middle

$$(z', push\ \varepsilon) \in \delta(z, \varepsilon, \varepsilon)$$

check matching symbols

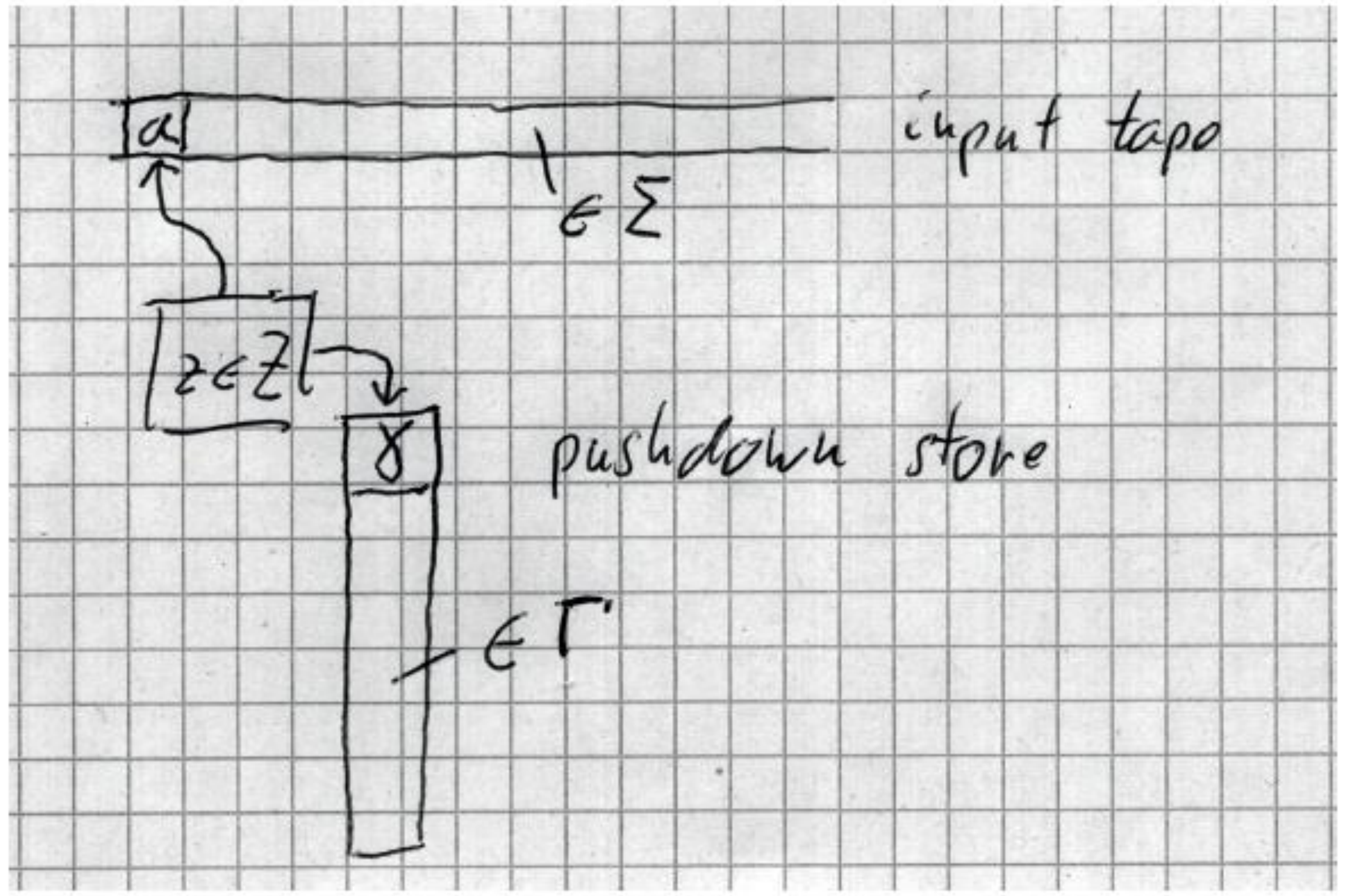
$$\begin{aligned} (z', pop) &\in (z', 0, 0) \\ (z', pop) &\in (z', 1, 1) \end{aligned}$$

example

$$
\begin{aligned}
w &\in A^* \\
w &= w_1 \dots w_n \\
w^R &= w_n \dots w_1 \\
L &= \{ww^R \,:\, w \in \mathbb{B}\}
\end{aligned}
$$

L is context free:

$$S \to 0S0 \mid 1S1 \mid \varepsilon$$



pushdown automaton for L:

$$Z = \{z, z'\} \quad , \quad z_0 = z \quad , \quad Z_A = \{z'\}$$

push w

$$(z, push\ 0) \in \delta(z, 0, \varepsilon)$$

$$(z, push\ 1) \in \delta(z, 1, \varepsilon)$$

guess middle

$$(z', push\ \varepsilon) \in \delta(z, \varepsilon, \varepsilon)$$

check matching symbols

$$(z', pop) \in (z', 0, 0)$$

$$(z', pop) \in (z', 1, 1)$$

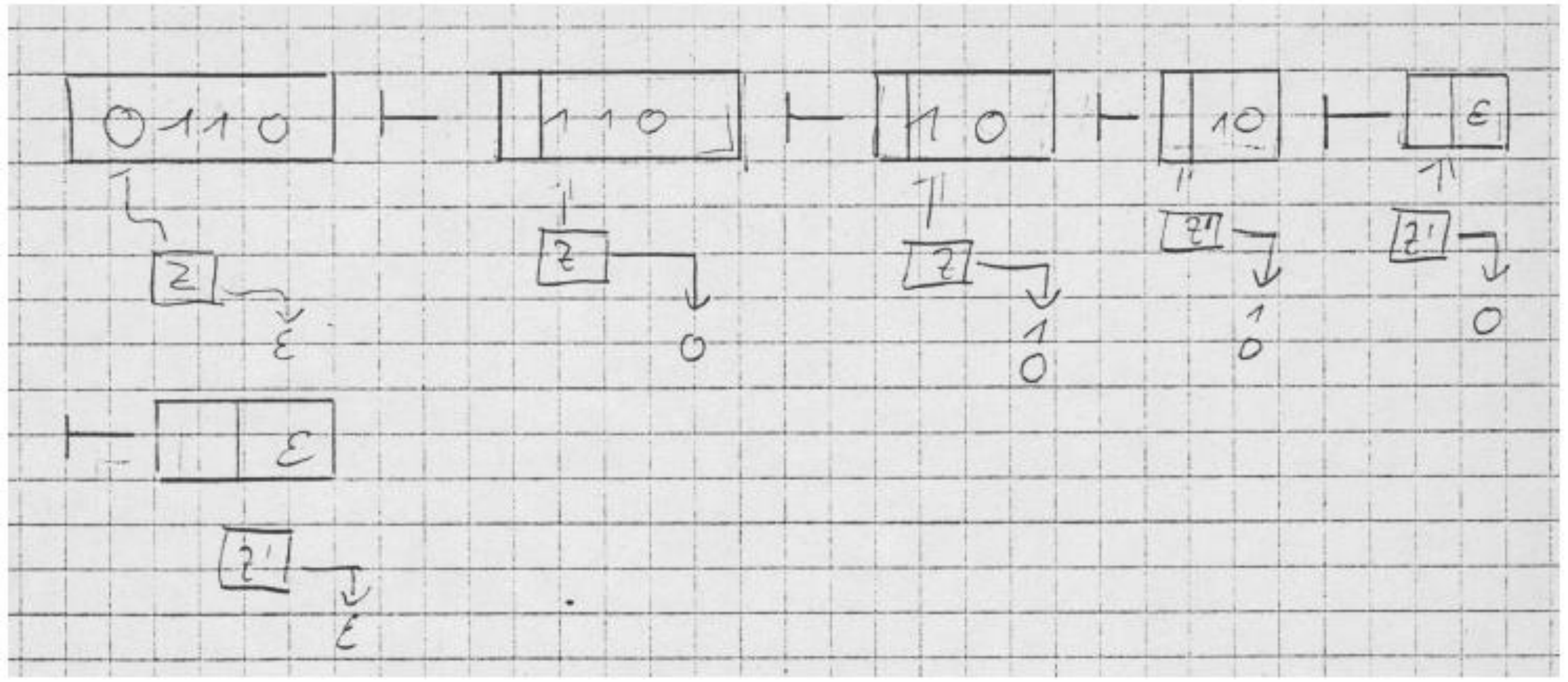


Figure 2: computation of the npda with input $w = 0110$

4 Context free languages are accepted by npda's

Lemma 5. *Let* $G = (N, T, P, S)$ *be a cf grammar. Then there is an npda* $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ *with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of M from G
2. $w \in L(G) \rightarrow w \in L(M)$
3. $w \in L(M) \rightarrow w \in L(G)$

4 Context free languages are accepted by npda's

Lemma 5. *Let $G = (N,T,P,S)$ be a cf grammar. Then there is an npda $M = (Z,\Sigma,\Gamma,\delta,z_0,Z_A)$ with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of M from G
2. $w \in L(G) \to w \in L(M)$
3. $w \in L(M) \to w \in L(G)$

4.1 Construction of M

$$\Sigma = T \quad , \quad \Gamma = N \cup T$$

Number productions

$$P = \{P_1,\dots,P_n\}$$

production i

$$P_i : \quad n_i = X_{i_1} \dots X_{i_{g(i)}} \quad \text{with } g_i \in \mathbb{N}_0$$

use state $r \in Z$ to guess what production should be applied next

4 Context free languages are accepted by npda's

Lemma 5. *Let $G = (N,T,P,S)$ be a cf grammar. Then there is an npda $M = (Z,\Sigma,\Gamma,\delta,z_0,Z_A)$ with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of M from G
2. $w \in L(G) \rightarrow w \in L(M)$
3. $w \in L(M) \rightarrow w \in L(G)$

4.1 Construction of M

$$\Sigma = T \quad , \quad \Gamma = N \cup T$$

Number productions

$$P = \{P_1,\ldots,P_n\}$$

production i

$$P_i : \quad n_i = X_{i_1} \ldots X_{i_{g(i)}} \quad \text{with } g_i \in \mathbb{N}_0$$

use state $r \in Z$ to guess what production should be applied next

1. initially push start symbol and prepare to guess

$$\delta(z_0,\varepsilon,\varepsilon) = \{(r, push\ S)\}$$

4 Context free languages are accepted by npda's

Lemma 5. *Let $G = (N, T, P, S)$ be a cf grammar. Then there is an npda $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of M from G
2. $w \in L(G) \rightarrow w \in L(M)$
3. $w \in L(M) \rightarrow w \in L(G)$

4.1 Construction of M

$$\Sigma = T \quad , \quad \Gamma = N \cup T$$

Number productions

$$P = \{P_1, \dots, P_n\}$$

production i

$$P_i : \quad n_i = X_{i_1} \dots X_{i_{g(i)}} \quad \text{with } g_i \in \mathbb{N}_0$$

use state $r \in Z$ to guess what production should be applied next

1. initially push start symbol and prepare to guess

$$\delta(z_0, \varepsilon, \varepsilon) = \{(r, push\ S)\}$$

2. choose a production P_i matching the top symbol $n \in N$ on stack

$$\delta(r, \varepsilon, n) = \{(r_i, push\ \varepsilon) \ : \ n_i = n\}$$

now replace $n = n_i$ on stack by right hand side $X_{i_1} \dots X_{i_{g(i)}}$ of production P_i

$$\delta(r_i, \varepsilon, n_i) = \begin{cases} \{(r_{i,g(i)}, pop)\} & g(i) \geq 1 \\ \{(r, pop)\} & g(i) = 0 \quad \text{i.e. } n_i \rightarrow \varepsilon \end{cases}$$

$$\delta(r_{i,j}, \varepsilon, \varepsilon) = \begin{cases} \{(r_{i,j-1}, push\ X_{i_j})\} & j \geq 2 \\ \{(r, push\ X_{i_1})\} & j = 1 \end{cases}$$

Effect of these transitions:

$$(r, w, n_i\alpha) \vdash^* (r, w, X_{i_1} \dots X_{i_{g(i)}}\alpha)$$

4 Context free languages are accepted by npda's

Lemma 5. *Let $G = (N,T,P,S)$ be a cf grammar. Then there is an npda $M = (Z,\Sigma,\Gamma,\delta,z_0,Z_A)$ with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of M from G
2. $w \in L(G) \to w \in L(M)$
3. $w \in L(M) \to w \in L(G)$

4.1 Construction of M

$$\Sigma = T \quad , \quad \Gamma = N \cup T$$

Number productions

$$P = \{P_1,\dots,P_n\}$$

production i

$$P_i : \quad n_i = X_{i_1} \dots X_{i_{g(i)}} \quad \text{with } g_i \in \mathbb{N}_0$$

use state $r \in Z$ to guess what production should be applied next

1. initially push start symbol and prepare to guess

$$\delta(z_0,\varepsilon,\varepsilon) = \{(r, push\ S)\}$$

2. choose a production P_i matching the top symbol $n \in N$ on stack

$$\delta(r,\varepsilon,n) = \{(r_i, push\ \varepsilon) \ : \ n_i = n\}$$

now replace $n = n_i$ on stack by right hand side $X_{i_1} \dots X_{i_{g(i)}}$ of production P_i

$$\delta(r_i,\varepsilon,n_i) = \begin{cases} \{(r_{i,g(i)}, pop)\} & g(i) \geq 1 \\ \{(r, pop)\} & g(i) = 0 \quad \text{i.e. } n_i \to \varepsilon \end{cases}$$

$$\delta(r_{i,j},\varepsilon,\varepsilon) = \begin{cases} \{(r_{i,j-1}, push\ X_{i_j})\} & j \geq 2 \\ \{(r, push\ X_{i_1})\} & j = 1 \end{cases}$$

Effect of these transitions:

$$(r,w,n_i\alpha) \vdash^* (r,w,X_{i_1} \dots X_{i_{g(i)}}\alpha)$$

3. cancel matching terminals $\in T$

$$\delta(r,t,t) = \{(R, pop)\}$$

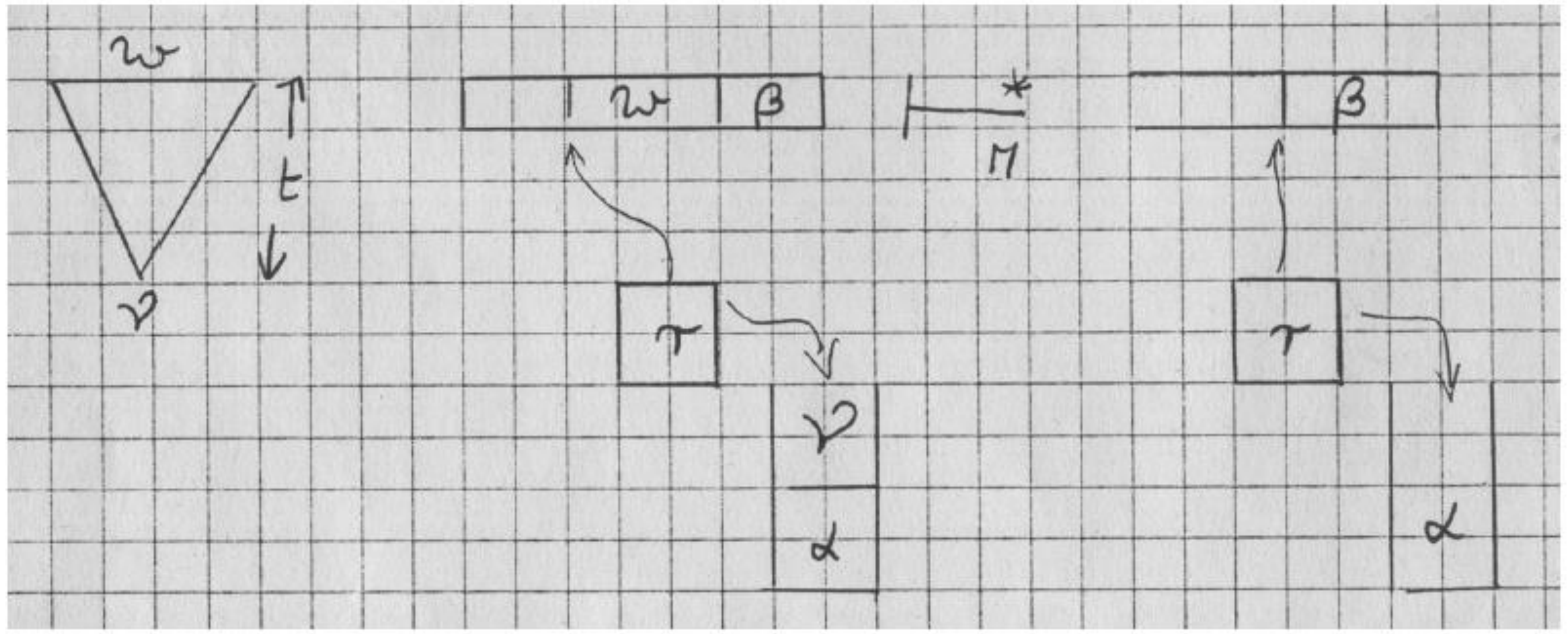
Effect

$$(r,t\beta,t\alpha) \vdash^* (r,\beta,\alpha)$$

4.2 $w \in L(G) \to w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \to^* w$ by a derivation tree of depth $\leq t$. Then*

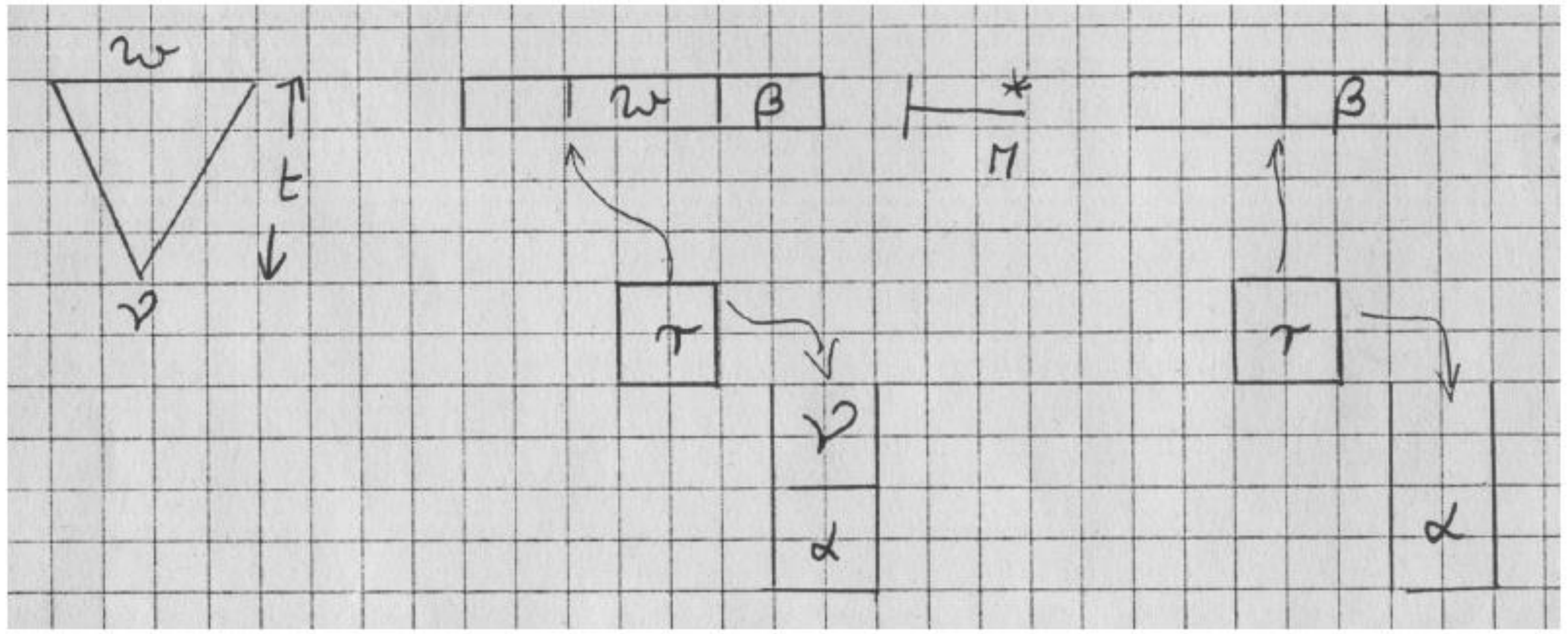
$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



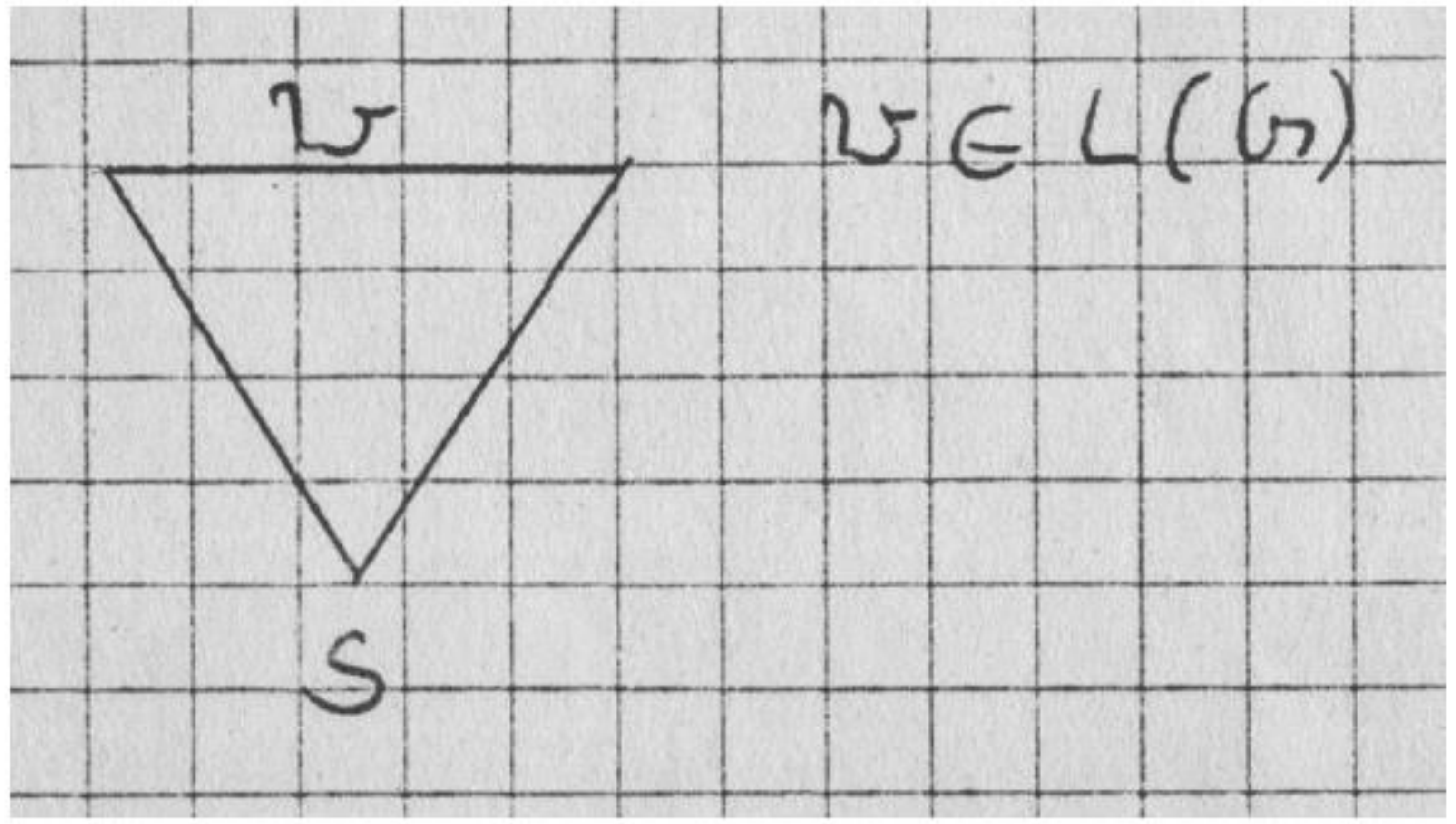
4.2 $w \in L(G) \rightarrow w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \rightarrow^* w$ by a derivation tree of depth $\leq t$. Then*

$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



Lemma implies claim: $w \in L(G)$ implies derivation tree as in figure 4

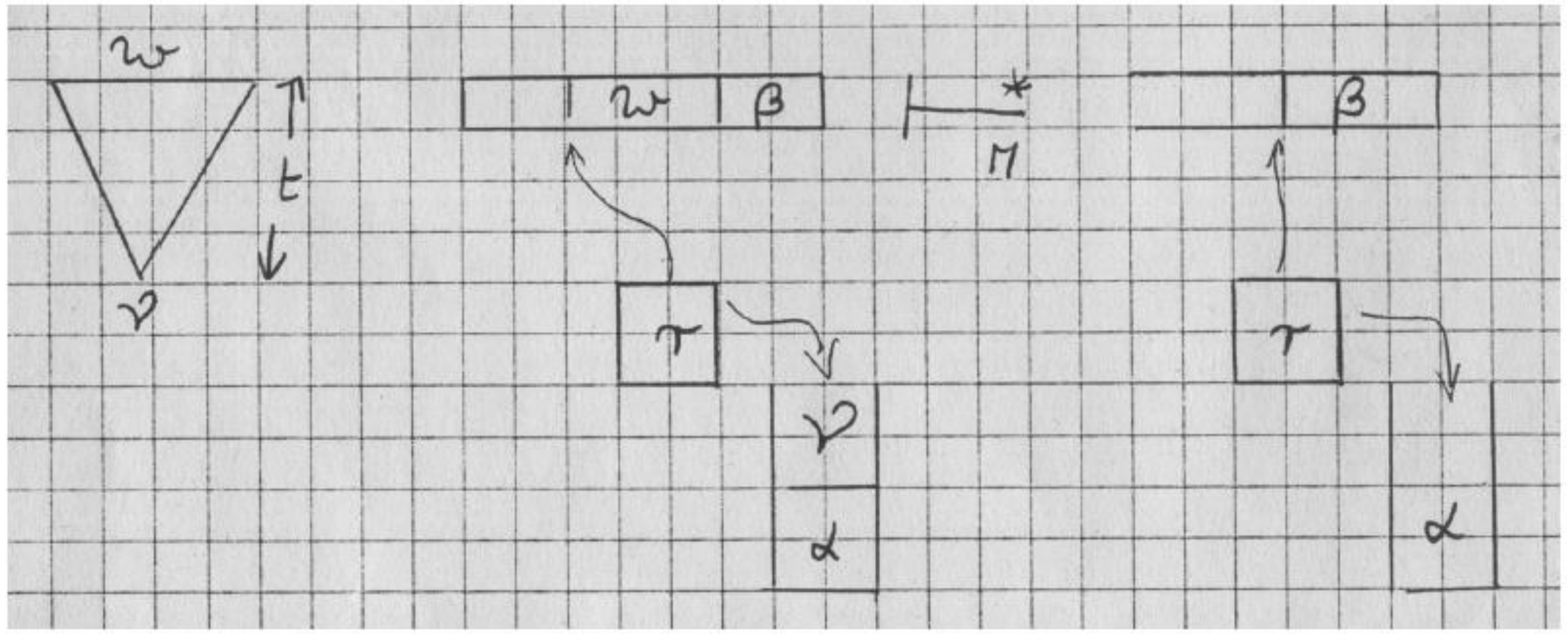


$$\begin{aligned} (z_0, w, \varepsilon) &\vdash (r, w, S) && \text{(construction 1)} \quad \text{pushing S} \\ &\vdash^* (r, \varepsilon, \varepsilon) && \text{(lemma 6)} \end{aligned}$$

4.2 $w \in L(G) \rightarrow w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \rightarrow^* w$ by a derivation tree of depth $\leq t$. Then*

$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



proof of lemma 6 by induction on t:

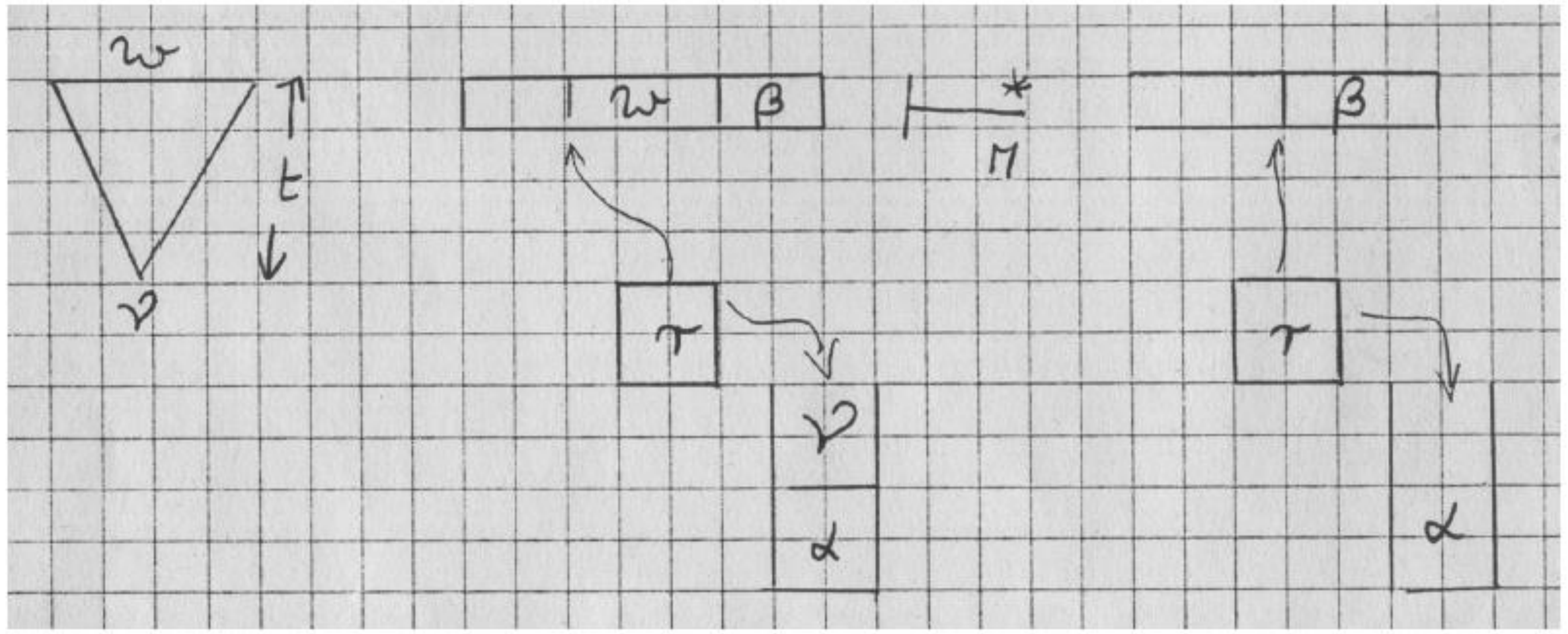
- $t = 0$: $v = w \in T$. Construction 3 $\rightarrow$ claim.

cancelling
matching terminals

4.2 $w \in L(G) \to w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \to^* w$ by a derivation tree of depth $\leq t$. Then*

$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



proof of lemma 6 by induction on t:

- $t = 0$: $v = w \in T$. Construction 3 $\to$ claim.

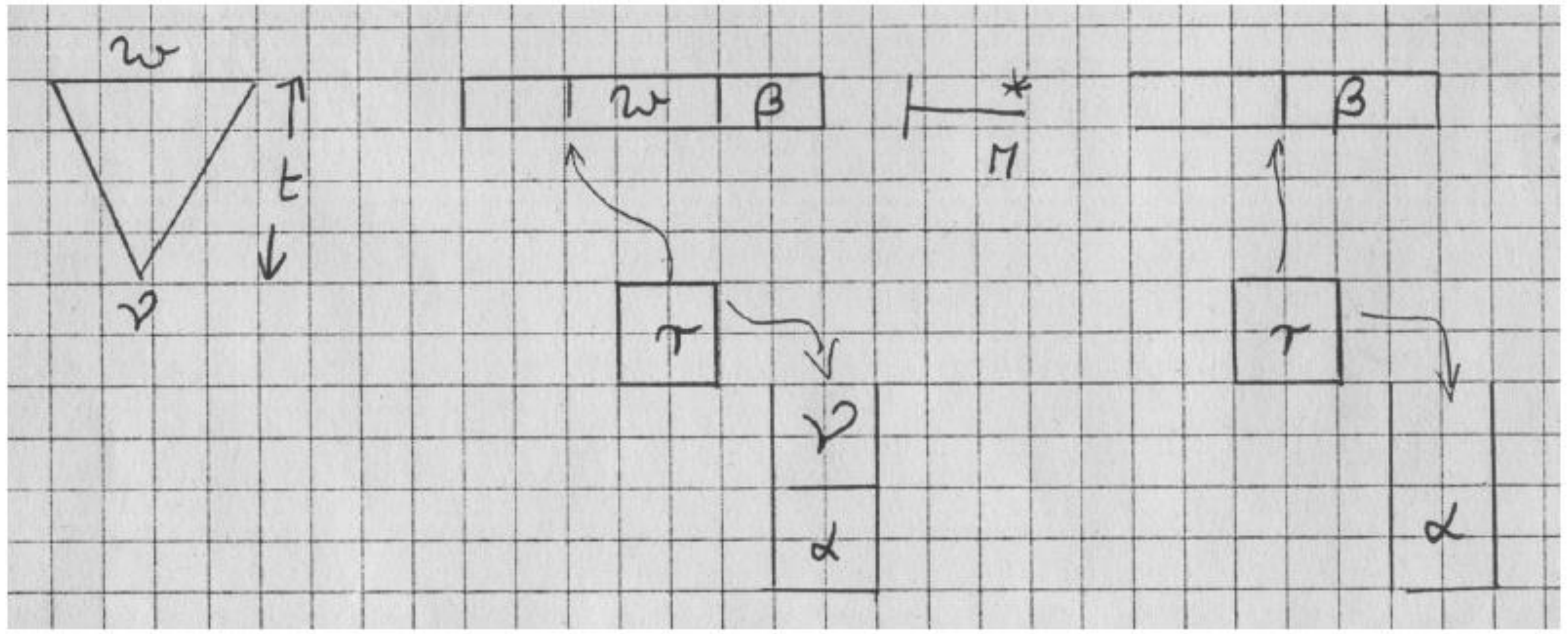
- $t - 1 \to t$:
  special case $v \to \varepsilon$. Construction 2 for $g(i) = 0$ $\to$ claim.

$$\delta(r_i, \varepsilon, n_i) = \begin{cases} \{(r_{i,g(i)}, pop)\} & g(i) \geq 1 \\ \{(r, pop)\} & g(i) = 0 \quad \text{i.e. } n_i \to \varepsilon \end{cases}$$

4.2 $w \in L(G) \rightarrow w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \rightarrow^* w$ by a derivation tree of depth $\leq t$. Then*

$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



proof of lemma 6 by induction on t:

- $t = 0$: $v = w \in T$. Construction 3 $\rightarrow$ claim.

- $t - 1 \rightarrow t$:
  special case $v \rightarrow \varepsilon$. Construction 2 for $g(i) = 0$ $\rightarrow$ claim.

otherwise: $n = n_i$ derivation tree as in figure 5

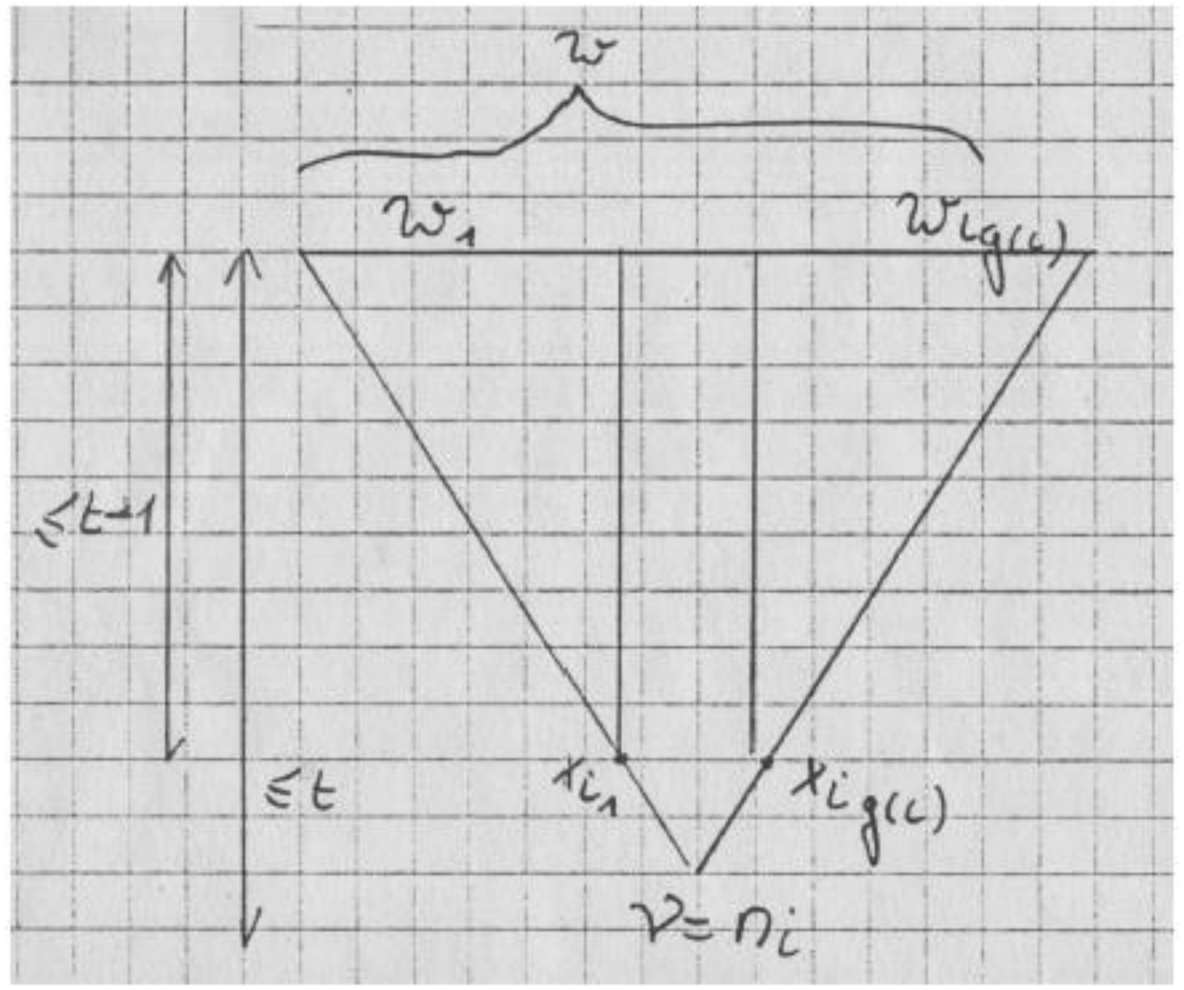


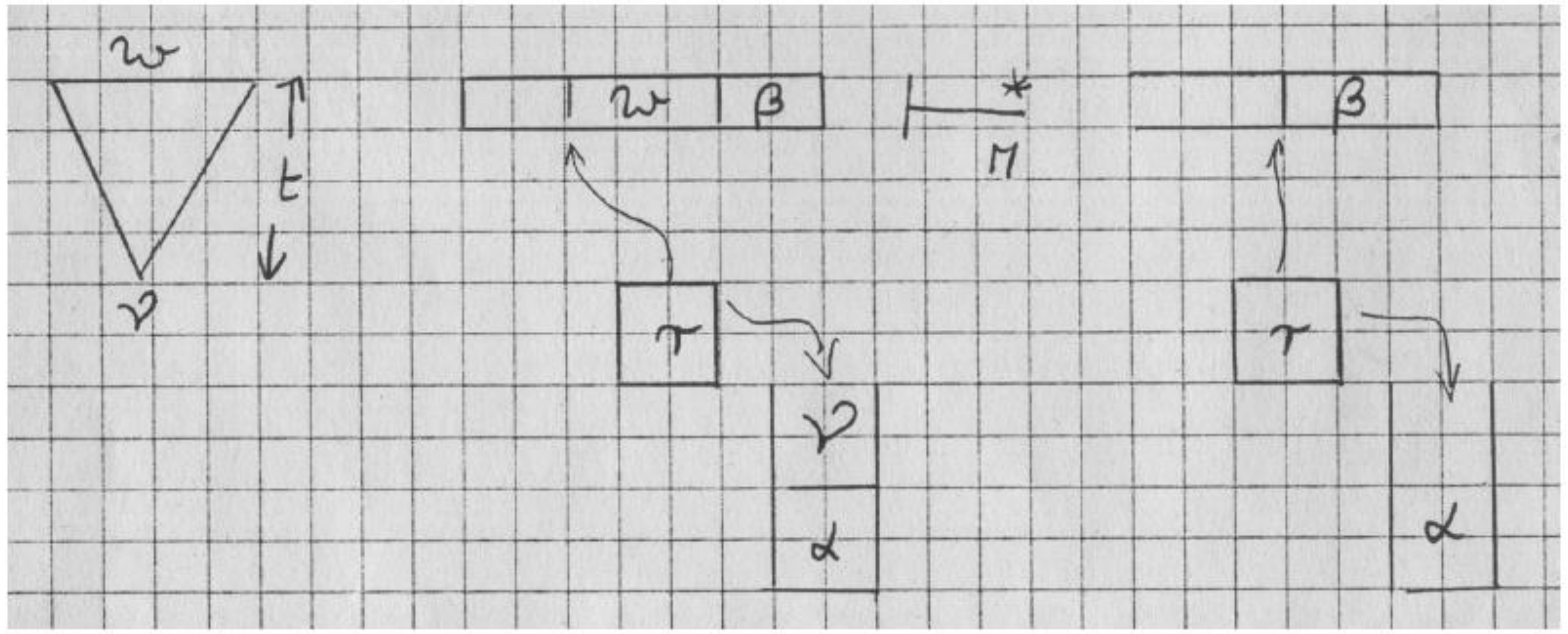
Figure 5: derivation tree of w from $v = n_i$

$$\begin{aligned}
(r, w\beta, v\alpha) &\vdash^*_{constr.2} (r, w_1 \ldots w_{g(i)}\beta, X_{i_1} \ldots X_{i_{g(i)}}\alpha) \\
&\vdash^*_{IH} (r, w_2 \ldots\ldots w_{g(i)}\beta, X_{i_2} \ldots X_{i_{g(i)}}\alpha) \\
&\ldots \\
&\vdash^*_{IH} (r, \beta, \alpha)
\end{aligned}$$

4.2 $w \in L(G) \rightarrow w \in L(M)$:

Lemma 6. *Let $w, \beta \in T^*$ and $v \in N \cup T$ and $v \rightarrow^* w$ by a derivation tree of depth $\leq t$. Then*

$$(r, w\beta, v\alpha) \vdash^* (r, \beta, \alpha)$$



proof of lemma 6 by induction on t:

- $t = 0$: $v = w \in T$. Construction 3 $\rightarrow$ claim.
- $t-1 \rightarrow t$:
  special case $v \rightarrow \varepsilon$. Construction 2 for $g(i) = 0$ $\rightarrow$ claim.

otherwise: $n = n_i$ derivation tree as in figure 5

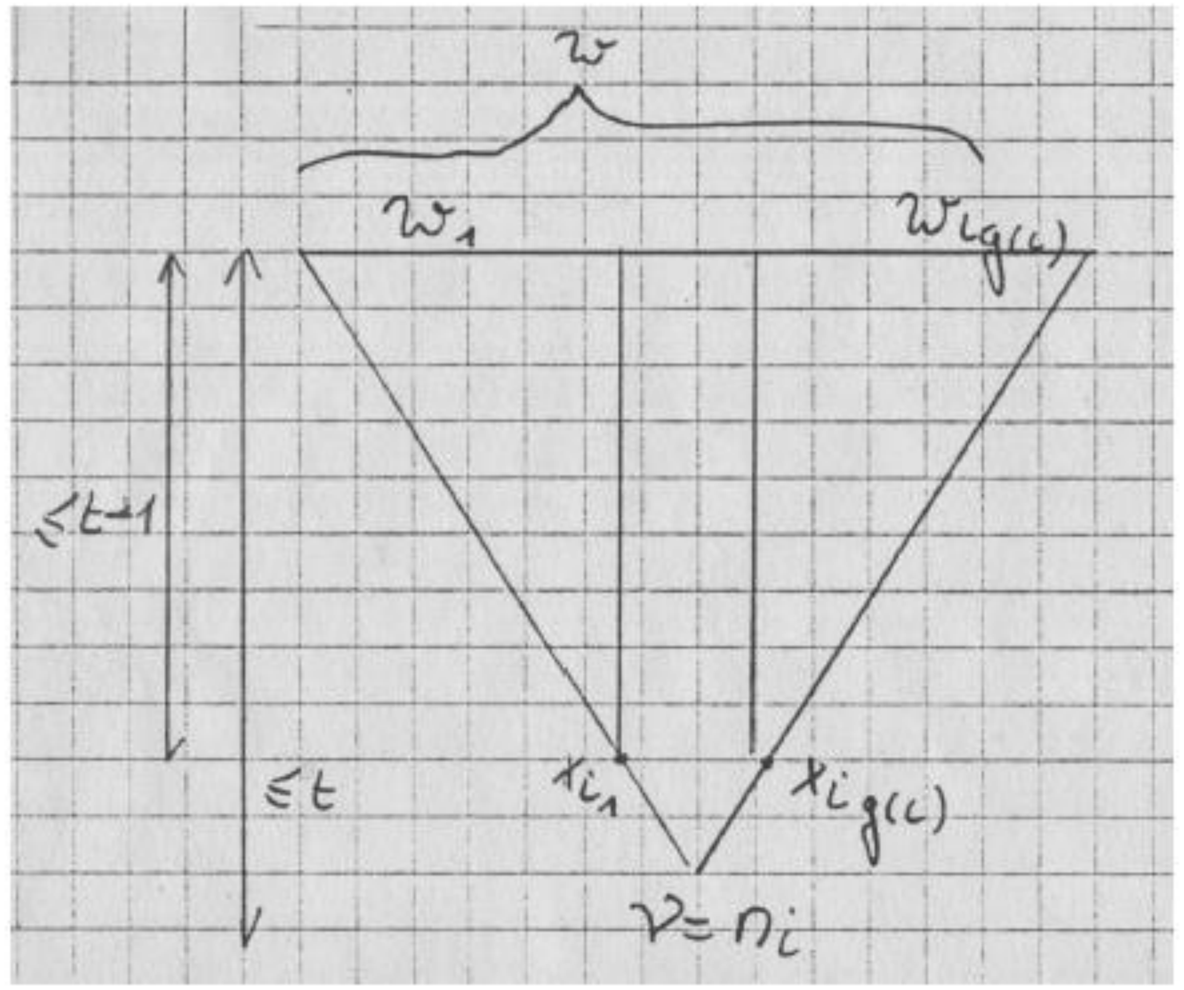


Figure 5: derivation tree of w from $v = n_i$

$$\begin{aligned}
(r, w\beta, v\alpha) &\vdash^*_{constr.2} (r, w_1 \dots w_{g(i)}\beta, X_{i_1} \dots X_{i_{g(i)}}\alpha) \\
&\vdash^*_{IH} (r, w_2 \dots\dots w_{g(i)}\beta, X_{i_2} \dots X_{i_{g(i)}}\alpha) \\
&\dots \\
&\vdash^*_{IH} (r, \beta, \alpha)
\end{aligned}$$

-

$Z_A = \{z_s, r\}$

4.3 $w \in L(M) \rightarrow w \in L(G)$:

Let $w \in L(M)$. Consider accepting computation

$$R = (k_0, \dots, k_s)$$

of M started with w

$$k_0 = (z_0, w, \varepsilon) \quad , \; k_s = (r, \varepsilon, \varepsilon)$$

For

$$k \in \{k_0, \dots, k_s\} \; , \; k = (r, \mu, \boxed{p\alpha}) \; , \; p \in N \cup T$$

define $N(k)$ as first configuration after k with state r and stack α

$$N(k) = (r, \mu', \boxed{\alpha})$$

Define $\tau(k)$ as number of steps between k and $N(k)$.

4.3 $w \in L(M) \rightarrow w \in L(G)$:

Let $w \in L(M)$. Consider accepting computation

$$R = (k_0, \ldots, k_s)$$

of M started with w

$$k_0 = (z_0, w, \varepsilon) \quad , \; k_s = (r, \varepsilon, \varepsilon)$$

For

$$k \in \{k_0, \ldots, k_s\} \; , \; k = (r, \mu, \boxed{p\alpha}) \; , \; p \in N \cup T$$

define $N(k)$ as first configuration after k with state r and stack α

$$N(k) = (r, \mu', \boxed{\alpha})$$

Define $\tau(k)$ as number of steps between k and $N(k)$.

Lemma 7. *If*

$$k = (r, \mu\beta, v\alpha) \; , \; N(k) = (r, \beta, \alpha)$$

then

$$v \rightarrow^* \mu$$

Claim follows with $\alpha = \beta = \varepsilon$

4.3 $w \in L(M) \to w \in L(G)$:

Let $w \in L(M)$. Consider accepting computation

$$R = (k_0, \ldots, k_s)$$

of M started with w

$$k_0 = (z_0, w, \varepsilon) \quad , \; k_s = (r, \varepsilon, \varepsilon)$$

For

$$k \in \{k_0, \ldots, k_s\} \; , \; k = (r, \mu, \boxed{p\alpha)} \; , \; p \in N \cup T$$

define $N(k)$ as first configuration after k with state r and stack α

$$N(k) = (r, \mu', \boxed{\alpha)}$$

Define $\tau(k)$ as number of steps between k and $N(k)$.

Lemma 7. *If*

$$k = (r, \mu\beta, v\alpha) \; , \; N(k) = (r, \beta, \alpha)$$

then

$$v \to^* \mu$$

Claim follows with $\alpha = \beta = \varepsilon$

proof of lemma : induction on $\tau(k)$

- $\tau(k) = 1$: construction 3 $\to$

$$\mu = v \in T \quad , \quad v \to^0 \mu$$

cancelling
matching terminals

4.3 $w \in L(M) \to w \in L(G)$:

Let $w \in L(M)$. Consider accepting computation

$$R = (k_0, \dots, k_s)$$

of M started with w

$$k_0 = (z_0, w, \varepsilon) \quad , k_s = (r, \varepsilon, \varepsilon)$$

For

$$k \in \{k_0, \dots, k_s\} \ , \ k = (r, \mu, \boxed{p\alpha}) \ , \ p \in N \cup T$$

define $N(k)$ as first configuration after k with state r and stack α

$$N(k) = (r, \mu', \boxed{\alpha})$$

Define $\tau(k)$ as number of steps between k and $N(k)$.

Lemma 7. *If*

$$k = (r, \mu\beta, v\alpha) \ , \ N(k) = (r, \beta, \alpha)$$

then

$$v \to^* \mu$$

Claim follows with $\alpha = \beta = \varepsilon$

proof of lemma : induction on $\tau(k)$

- $\tau(k) = 1$: construction 3 $\to$

$$\mu = v \in T \quad , \quad v \to^0 \mu$$

- $\tau(k) > 1$
special case of construction 2: $g(i) = 0$. Then

$$v \to \varepsilon \in P \ , \ \mu = \varepsilon \ , v \to^1 \mu$$

$$\delta(r_i, \varepsilon, n_i) \quad = \quad \begin{cases} \{(r_{i,g(i)}, pop)\} & g(i) \geq 1 \\ \underline{\{(r, pop)\} \qquad g(i) = 0} & \text{i.e. } n_i \to \varepsilon \end{cases}$$

Lemma 7. *If*

$$k = (r, \mu\beta, v\alpha)\ ,\ N(k) = (r, \beta, \alpha)$$

then

$$v \rightarrow^{*} \mu$$

Claim follows with $\alpha = \beta = \varepsilon$

proof of lemma : induction on $\tau(k)$

- $\tau(k) = 1$: construction 3 $\rightarrow$

$$\mu = v \in T \quad , \quad v \rightarrow^{0} \mu$$

- $\tau(k) > 1$
  special case of construction 2: $g(i) = 0$. Then

$$v \rightarrow \varepsilon \in P\ ,\ \mu = \varepsilon\ ,v \rightarrow^{1} \mu$$

otherwise

$$g(i) \geq 1, v = n_i$$

decompose computation

$$\begin{array}{rlll}
(r, \mu\beta, v\alpha) & \vdash^{*} & k_1 = (r, \mu\beta, X_{i_1} \dots X_{i_{g(i)}}\alpha) & \text{(construction 2)} \\
& \vdash^{\tau(k_1)} & k_2 = (r, \overline{\mu_1}\beta, X_{i_2} \dots X_{i_{g(i)}}\alpha) & \text{with } \mu = \mu_1 \circ \overline{\mu_1} \\
& \dots & & \\
& \vdash^{\tau(k_{g(i)-1})} & k_{g(i)} = (r, \overline{\mu_{g(i)-1}}\beta, X_{i_{g(i)}}\alpha) & \\
& \vdash^{\tau(k_{g(i)})} & N(k) = (r, \beta, \alpha) &
\end{array}$$

with $\tau(k_j) < \tau(k)$ for all j.

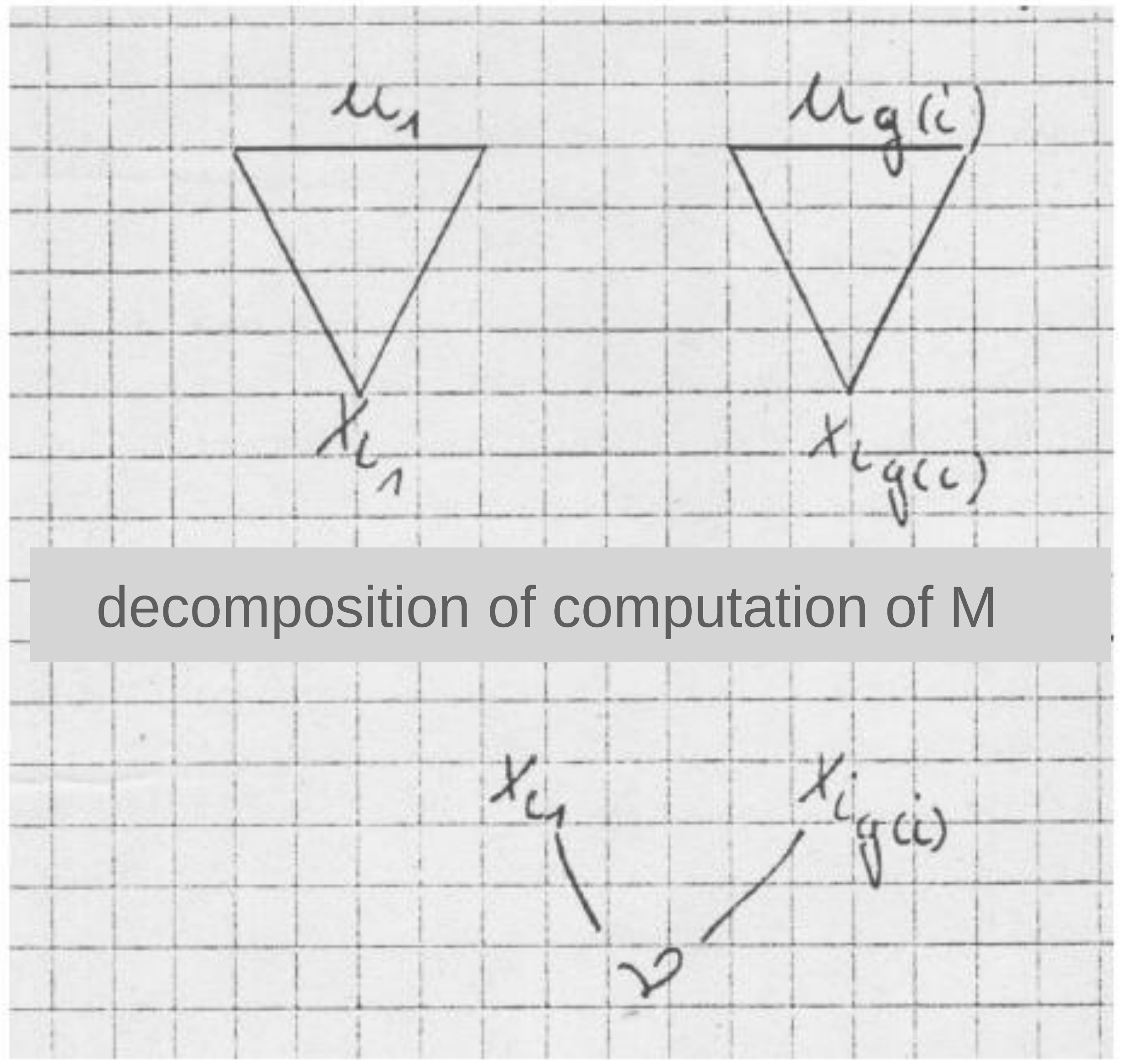


Figure 6: derivation tree of w from $v = n_i$. Upper subtrees exist by induction hypothesis, production at the root exists by construction of M.

Lemma 7. *If*

$$k = (r, \mu\beta, \nu\alpha)\ ,\ N(k) = (r, \beta, \alpha)$$

then

$$\nu \to^* \mu$$

Claim follows with $\alpha = \beta = \varepsilon$

proof of lemma : induction on $\tau(k)$

- $\tau(k) = 1$: construction 3 $\to$

$$\mu = \nu \in T \quad , \quad \nu \to^0 \mu$$

- $\tau(k) > 1$
  special case of construction 2: $g(i) = 0$. Then

$$\nu \to \varepsilon \in P\ ,\ \mu = \varepsilon\ , \nu \to^1 \mu$$

otherwise

$$g(i) \geq 1, \nu = n_i$$

part 1: reasonably straight forward construction
parts 2 and 3: clever induction hypothesis,
Then: bookkeeping.

decompose computation

$$\begin{array}{llll}
(r, \mu\beta, \nu\alpha) & \vdash^* & k_1 = (r, \mu\beta, X_{i_1} \dots X_{i_{g(i)}} \alpha) & \text{(construction 2)} \\
 & \vdash^{\tau(k_1)} & k_2 = (r, \overline{\mu_1}\beta, X_{i_2} \dots X_{i_{g(i)}} \alpha) & \text{with } \mu = \mu_1 \circ \overline{\mu_1} \\
 & \dots & & \\
 & \vdash^{\tau(k_{g(i)-1})} & k_{g(i)} = (r, \overline{\mu_{g(i)-1}}\beta, X_{i_{g(i)}} \alpha) & \\
 & \vdash^{\tau(k_{g(i)})} & N(k) = (r, \beta, \alpha) &
\end{array}$$

with $\tau(k_j) < \tau(k)$ for all j.

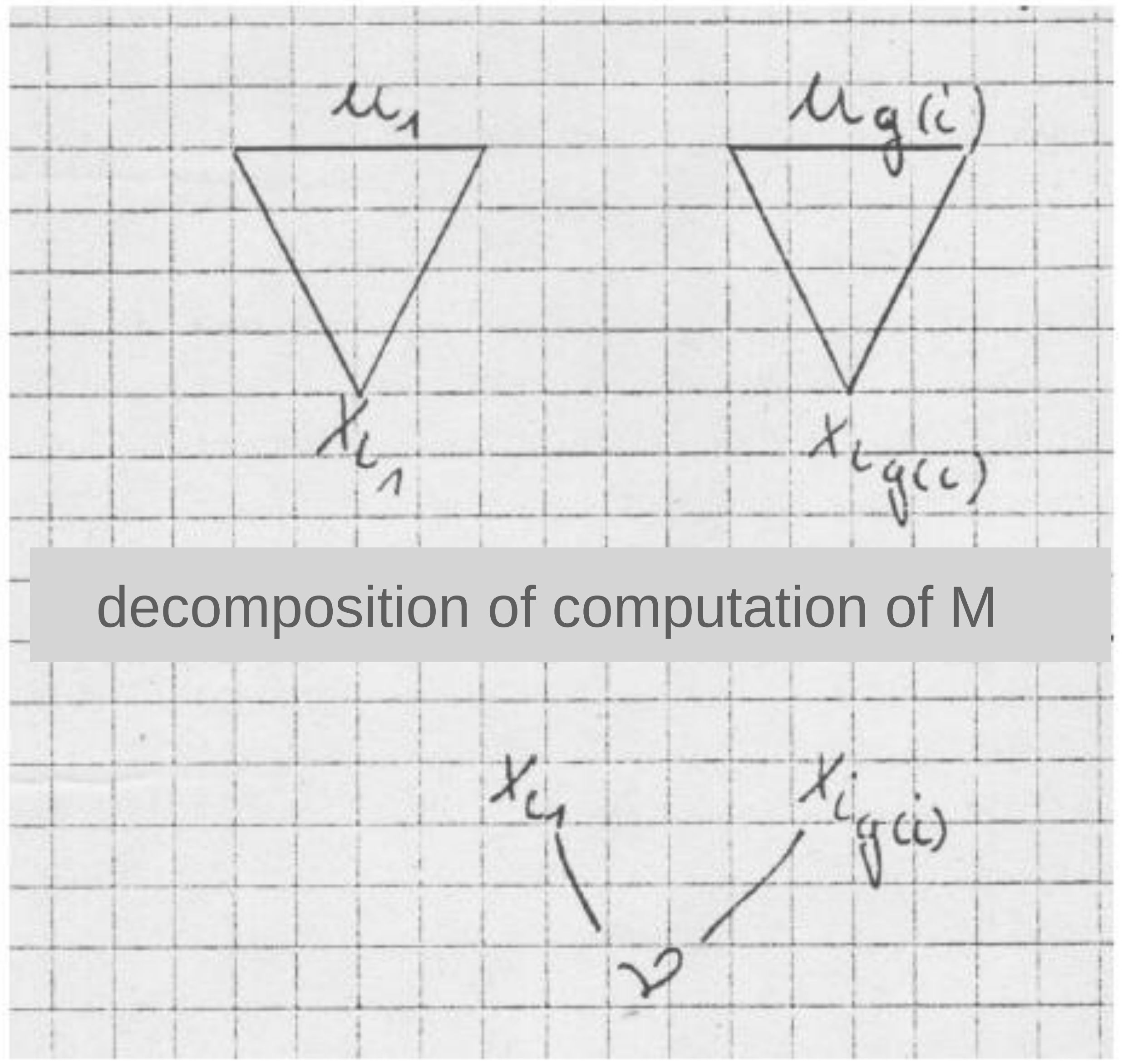


Figure 6: derivation tree of w from $\nu = n_i$. Upper subtrees exist by induction hypothesis, production at the root exists by construction of M.

5 Languages accepted by npda's are context free

‚computing with grammars'

Lemma 8. *Let* $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ *be an npda. Then there is a context free grammar* $G = (N, T, P, S)$ *with*

$$L(M) = L(G)$$

5 Languages accepted by npda's are context free

‚computing with grammars'

Lemma 8. *Let* $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ *be an npda. Then there is a context free grammar* $G = (N, T, P, S)$ *with*

$$L(M) = L(G)$$

Proof in 3 steps:

1. construction of G
2. $x \in L(G) \rightarrow x \in L(M)$
3. $x \in L(M) \rightarrow x \in L(G)$

Notation For $n \in N$ we define the language generated by n as

$$L(n) = \{w \in T^* \ : \ n \rightarrow^* w\}$$

Lemma 8. *Let $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ be an npda. Then there is a context free grammar $G = (N, T, P, S)$ with*

$$L(M) = L(G)$$

Notation For $n \in N$ we define the language generated by n as

$$L(n) = \{w \in T^* \ : \ n \to^* w\}$$

5.1 Construction of G

$$\begin{aligned} T &= \Sigma \\ N &= \{\langle q, A, p\rangle \ : \ p, q \in Z \ , \ A \in \Gamma_\varepsilon\} \cup \{S\} \\ S &= S \end{aligned}$$

intention:

$$L(\langle q, A, p\rangle) = \{w \ : \ (q, w, A) \vdash^* (p, \varepsilon, \varepsilon)\}$$

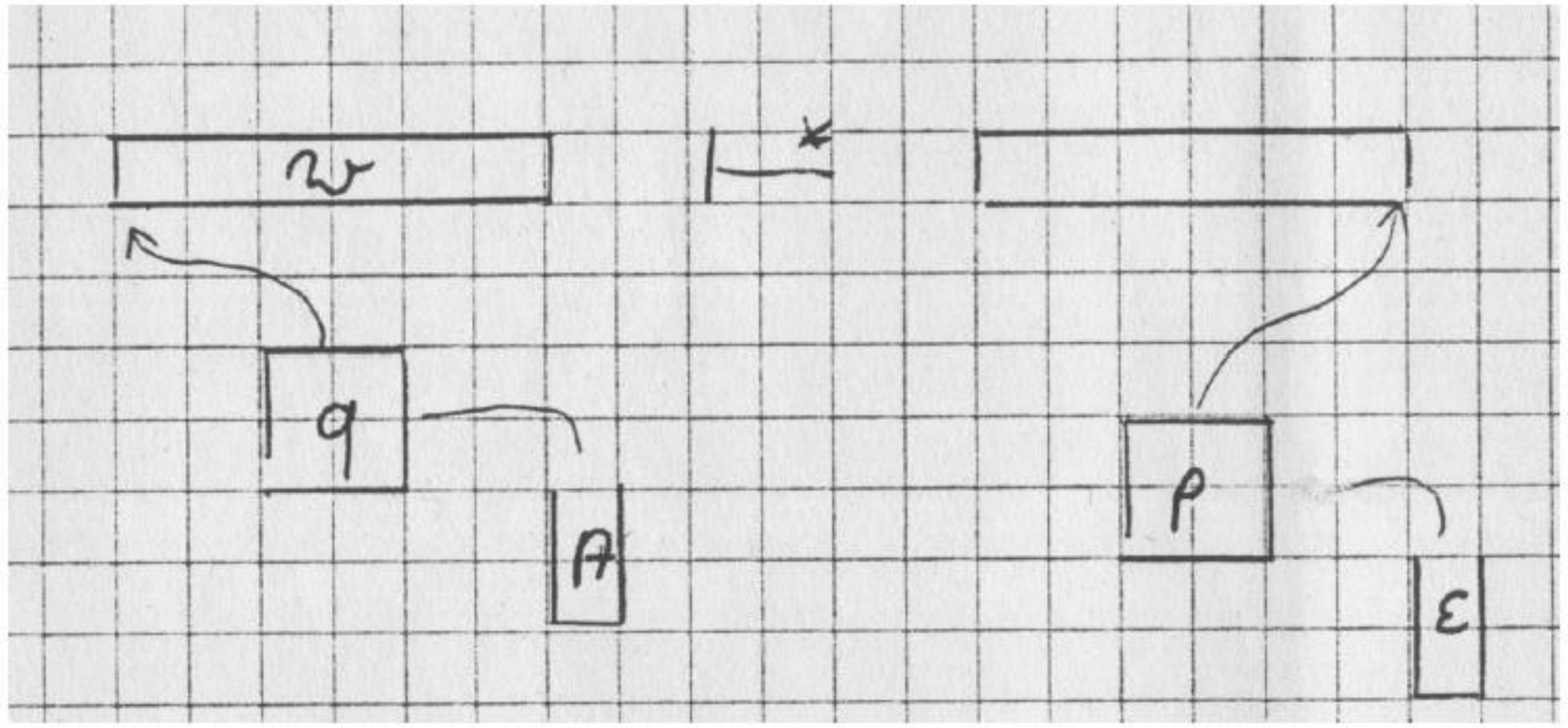


Figure 7: Symbol $\langle q, A, p\rangle$ should generate all words w such that M started in state q with input w and stack A consumes the input, empties the stack and ends in state p

5.1 Construction of G

$$\begin{aligned} T &= \Sigma \\ N &= \{\langle q,A,p\rangle \ : \ p,q \in Z \ , \ A \in \Gamma_\varepsilon\} \cup \{S\} \\ S &= S \end{aligned}$$

intention:

$$L(\langle q,A,p\rangle) = \{w \ : \ (q,w,A) \vdash^* (p,\varepsilon,\varepsilon)\}$$

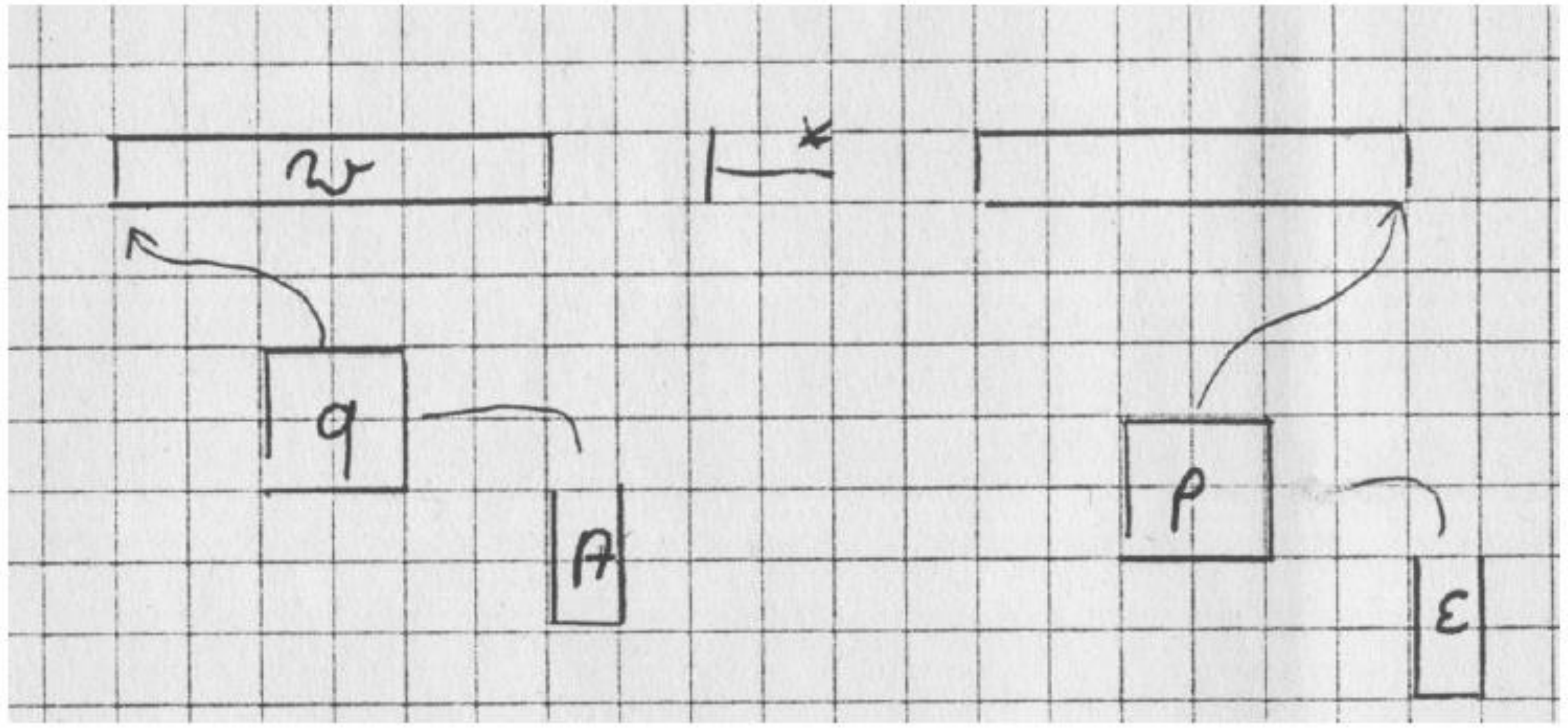


Figure 7: Symbol $\langle q,A,p\rangle$ should generate all words w such that M started in state q with input w and stack A consumes the input, empties the stack and ends in state p

1. for all accepting states $f \in Z_A$:

$$S \rightarrow \langle z_0, \varepsilon, f\rangle$$

2. if

$$(r, push\ a) \in \delta(q,u,A) \quad \text{with} \quad u \in \Sigma_\varepsilon \quad a \in \Gamma_\varepsilon, \quad A \in \Gamma$$

then for all $q_1, p \in Z$ intermediate and end states

$$\langle q,A,p\rangle \rightarrow u\langle r,a,q_1\rangle\langle q_1,A,p\rangle$$

3. similar; if

$$(r, push\ a) \in \delta(q,u,\varepsilon) \quad \text{with} \quad u \in \Sigma_\varepsilon \quad a \in \Gamma_\varepsilon$$

then for all $q_1, p \in Z$ and $b \in \Gamma_\varepsilon$: also possible top(stack)

$$\langle q,b,p\rangle \rightarrow u\langle r,a,q_1\rangle\langle q_1,b,p\rangle$$

5.1 Construction of G

$$
\begin{aligned}
T &= \Sigma \\
N &= \{\langle q,A,p\rangle \;:\; p,q\in Z\,,\, A\in\Gamma_\varepsilon\}\cup\{S\} \\
S &= S
\end{aligned}
$$

intention:

$$L(\langle q,A,p\rangle) = \{w \;:\; (q,w,A) \vdash^* (p,\varepsilon,\varepsilon)\}$$

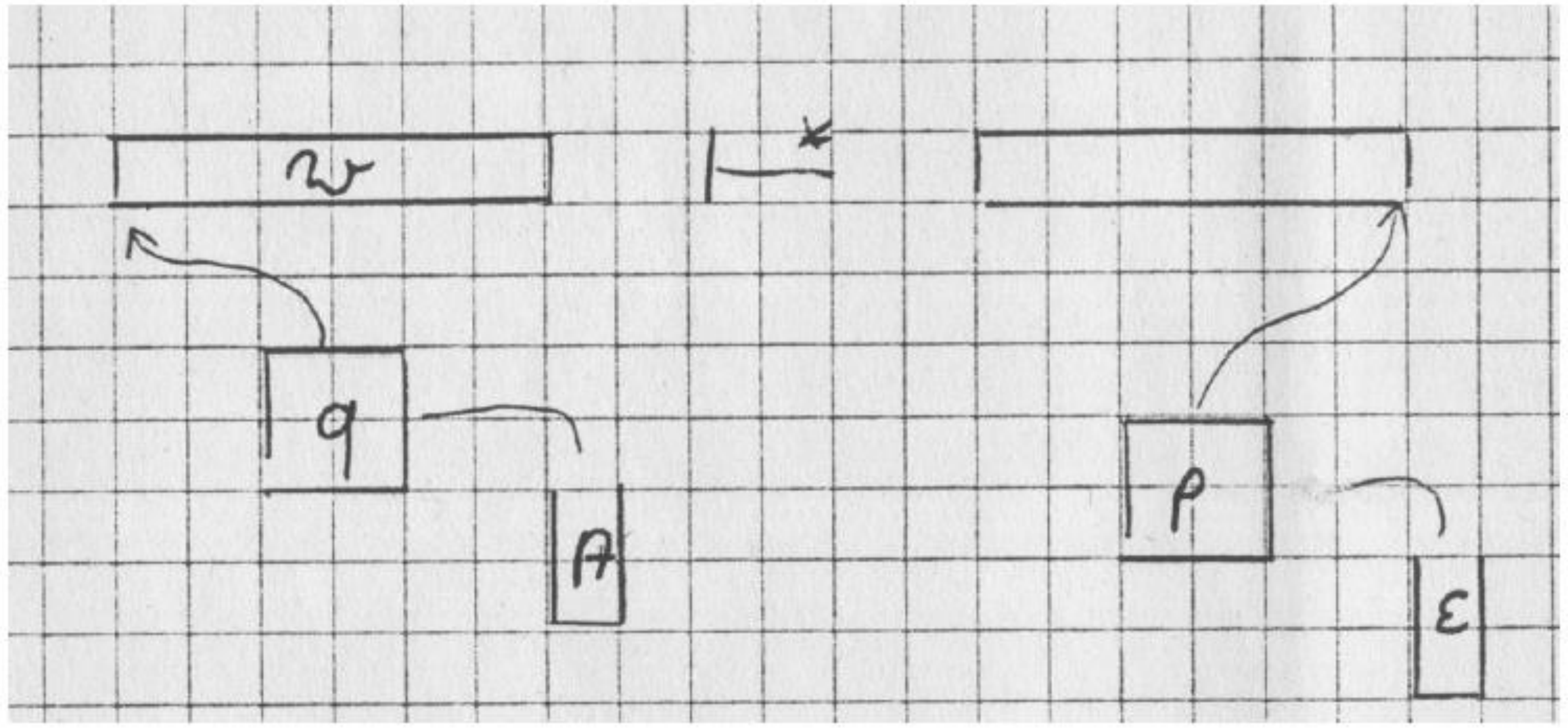


Figure 7: Symbol $\langle q,A,p\rangle$ should generate all words w such that M started in state q with input w and stack A consumes the input, empties the stack and ends in state p

1. for all accepting states $f\in Z_A$:

   $$S\to\langle z_0,\varepsilon,f\rangle$$

2. if

   $$(r, push\ a)\in\delta(q,u,A) \quad \text{with} \quad u\in\Sigma_\varepsilon \quad a\in\Gamma_\varepsilon, \quad A\in\Gamma$$

   then for all $q_1,p\in Z$ intermediate and end states

   $$\langle q,A,p\rangle\to u\langle r,a,q_1\rangle\langle q_1,A,p\rangle$$

3. similar; if

   $$(r, push\ a)\in\delta(q,u,\varepsilon) \quad \text{with} \quad u\in\Sigma_\varepsilon \quad a\in\Gamma_\varepsilon$$

   then for all $q_1,p\in Z$ and $b\in\Gamma_\varepsilon$: also possible top(stack)

   $$\langle q,b,p\rangle\to u\langle r,a,q_1\rangle\langle q_1,b,p\rangle$$

4. if

   $$(r, pop)\in\delta(q,u,A) \quad \text{with} \quad u\in\Sigma_\varepsilon \quad A\in\Gamma_\varepsilon$$

   then for all $p\in Z$ end states

   $$\langle q,A,p\rangle\to u\langle r,\varepsilon,p\rangle$$

5. similar; if

   $$(r, pop)\in\delta(q,u,\varepsilon) \quad \text{with} \quad u\in\Sigma_\varepsilon$$

   then for all $b\in\Gamma$ and $p\in Z$ also possible top(stack)

   $$\langle q,b,p\rangle\to u\langle r,\varepsilon,p\rangle$$

5.1 Construction of G

$$
\begin{aligned}
T &= \Sigma \\
N &= \{\langle q,A,p\rangle \;:\; p,q \in Z\,,\, A \in \Gamma_\varepsilon\} \cup \{S\} \\
S &= S
\end{aligned}
$$

intention:

$$L(\langle q,A,p\rangle) = \{w \;:\; (q,w,A) \vdash^* (p,\varepsilon,\varepsilon)\}$$

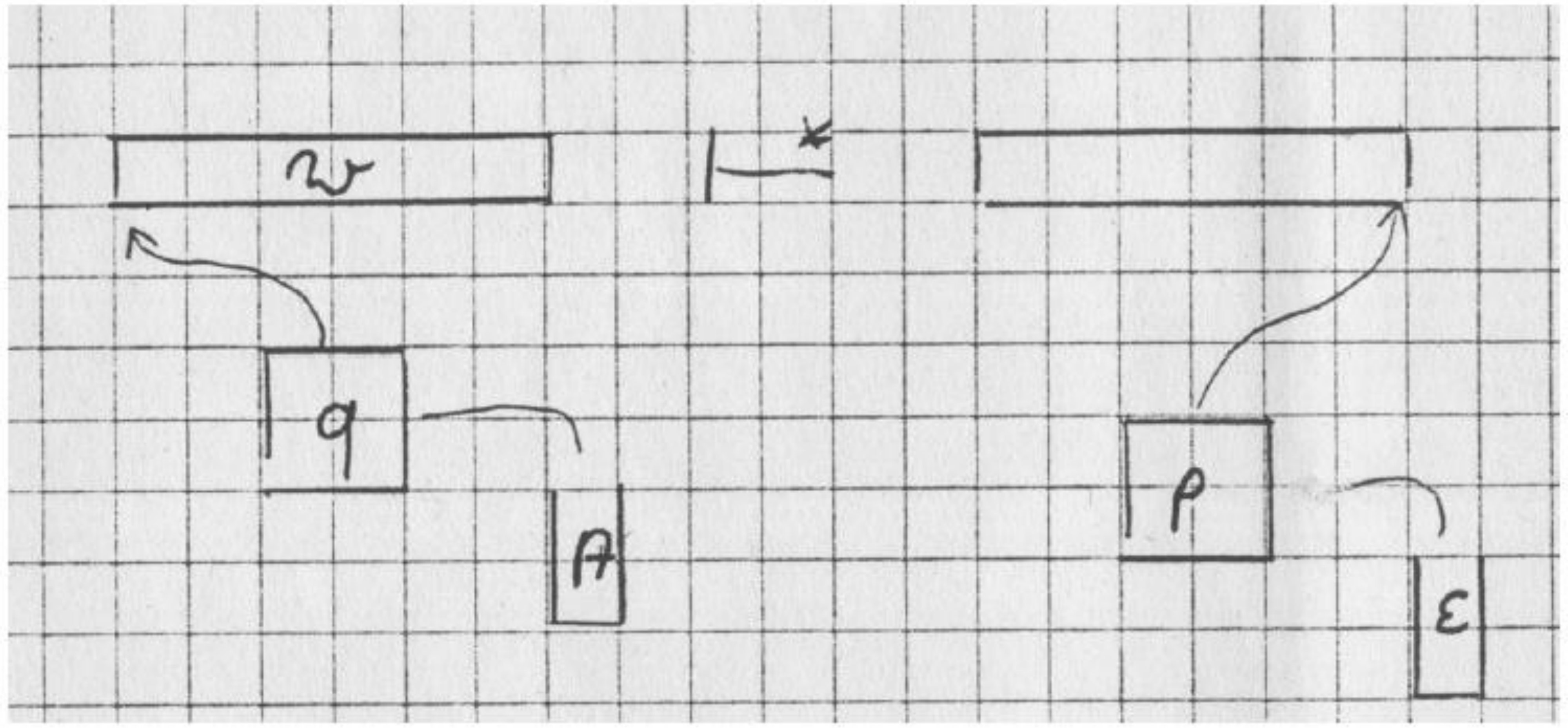


Figure 7: Symbol $\langle q,A,p\rangle$ should generate all words w such that M started in state q with input w and stack A consumes the input, empties the stack and ends in state p

1. for all accepting states $f \in Z_A$:

$$S \to \langle z_0, \varepsilon, f\rangle$$

2. if

$$(r, push\ a) \in \delta(q,u,A) \quad \text{with} \quad u \in \Sigma_\varepsilon \quad a \in \Gamma_\varepsilon, \quad A \in \Gamma$$

then for all $q_1, p \in Z$ intermediate and end states

$$\langle q,A,p\rangle \to u\langle r,a,q_1\rangle\langle q_1,A,p\rangle$$

3. similar; if

$$(r, push\ a) \in \delta(q,u,\varepsilon) \quad \text{with} \quad u \in \Sigma_\varepsilon \quad a \in \Gamma_\varepsilon$$

then for all $q_1, p \in Z$ and $b \in \Gamma_\varepsilon$: also possible top(stack)

$$\langle q,b,p\rangle \to u\langle r,a,q_1\rangle\langle q_1,b,p\rangle$$

4. if

$$(r, pop) \in \delta(q,u,A) \quad \text{with} \quad u \in \Sigma_\varepsilon \quad A \in \Gamma_\varepsilon$$

then for all $p \in Z$ end states

$$\langle q,A,p\rangle \to u\langle r,\varepsilon,p\rangle$$

5. similar; if

$$(r, pop) \in \delta(q,u,\varepsilon) \quad \text{with} \quad u \in \Sigma_\varepsilon$$

then for all $b \in \Gamma$ and $p \in Z$ also possible top(stack)

$$\langle q,b,p\rangle \to u\langle r,\varepsilon,p\rangle$$

6. for all $q \in Z$

$$\langle q,\varepsilon,q\rangle \to \varepsilon$$

5.2 $x \in L(G) \to x \in L(M)$

Lemma 9.

$$x \in L(\langle q, A, p \rangle) \to (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

5.2 $x \in L(G) \to x \in L(M)$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \to (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

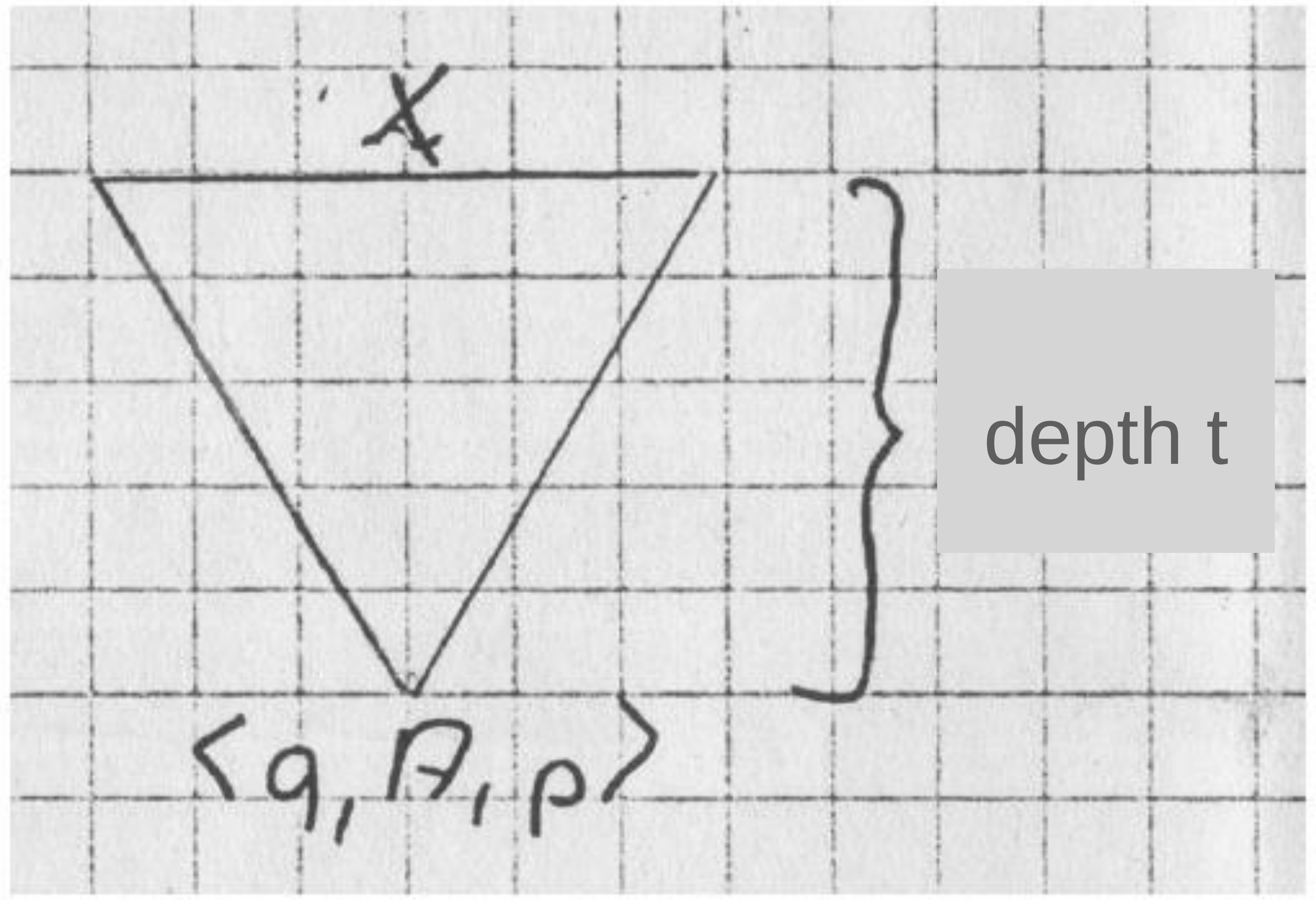


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

- $t = 1$ only for construction 6.

$$\langle q, \varepsilon, q\rangle \to \varepsilon \quad , \quad \varepsilon \in T^*$$

$$p = q \ , x = A = \varepsilon$$

$$(q, x, A) = (q, \varepsilon, \varepsilon) \vdash^0 (q, \varepsilon, \varepsilon) = (p, \varepsilon, \varepsilon)$$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \to (q,x,A) \vdash^* (p,\varepsilon,\varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

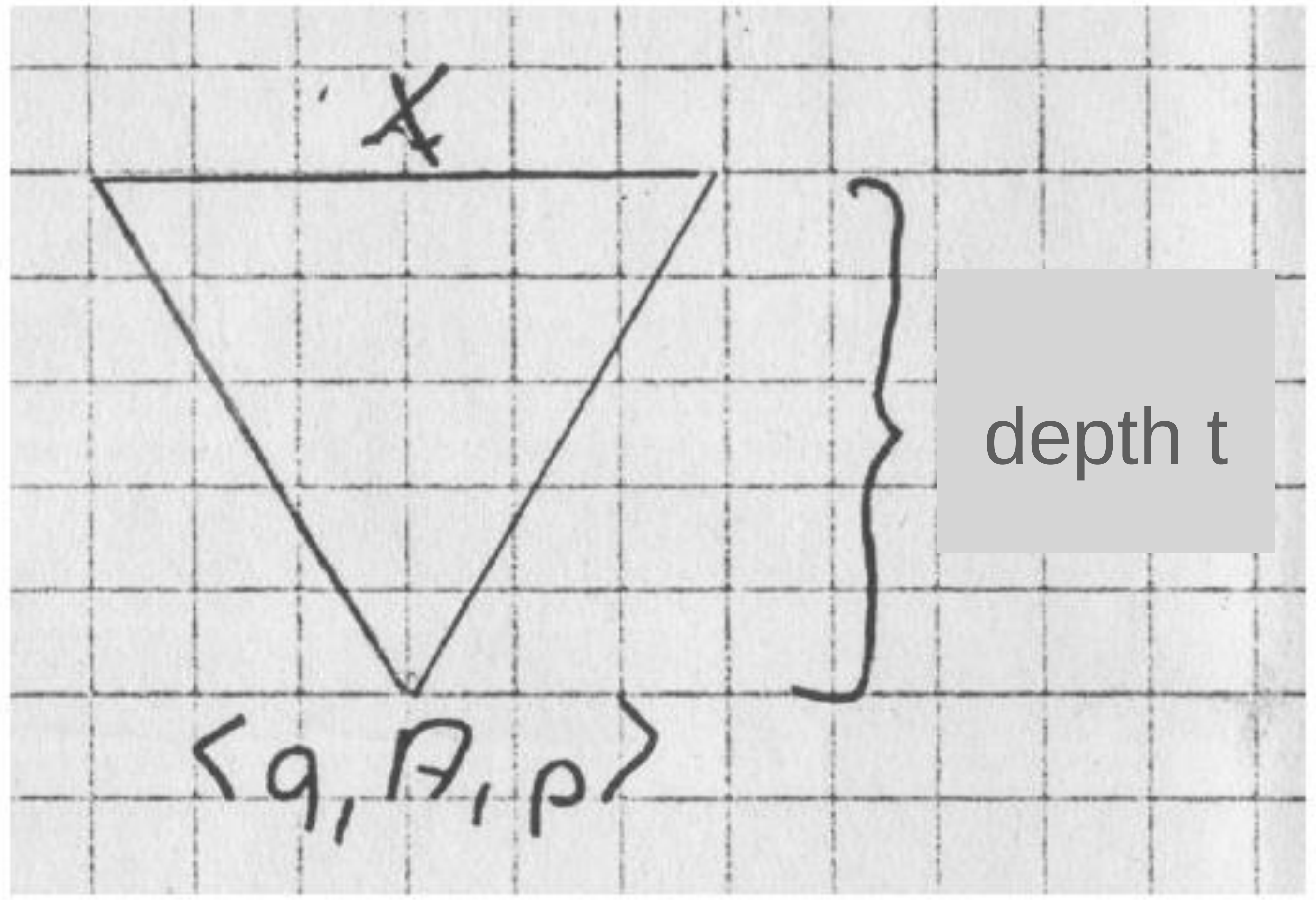


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q,x,A) \vdash^* (p,\varepsilon,\varepsilon)$.

- $t-1 \to t$

construction 2: see figure 9 push a with $A \neq \epsilon$

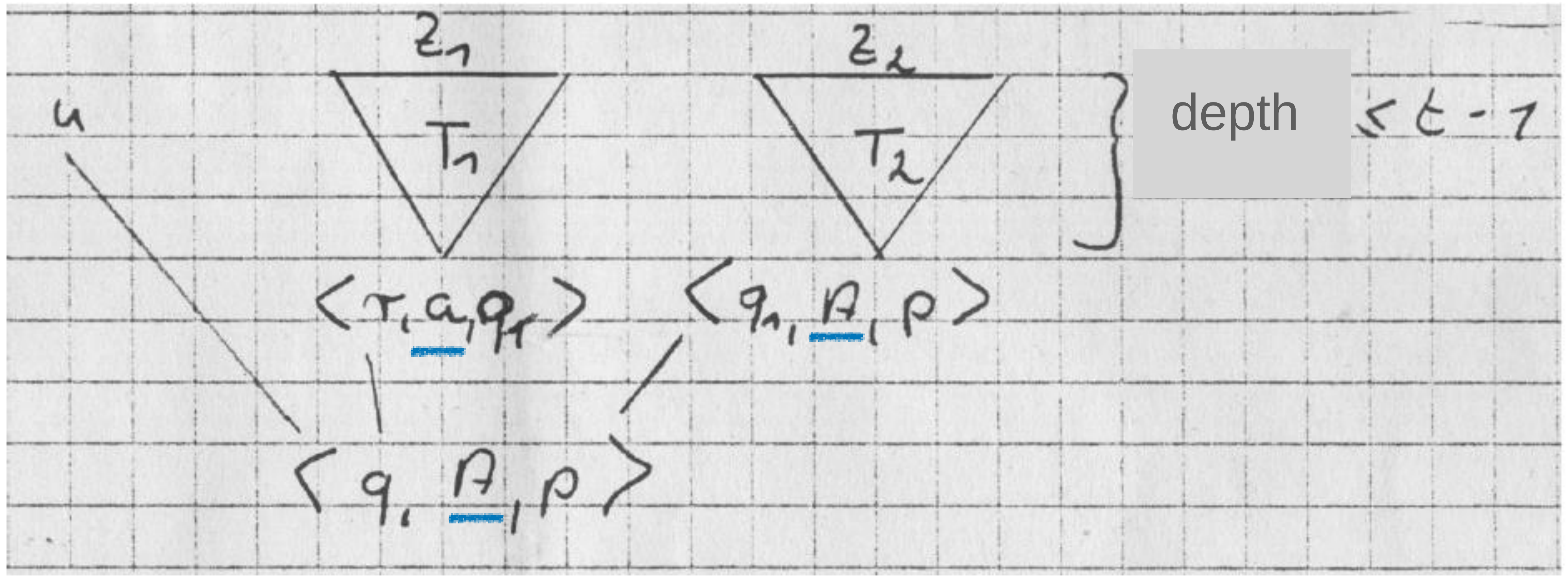


Figure 9: derivation tree for case 2 of the construction. The input word is $x = uz_1z_2$

Lemma 9.

$$x \in L(\langle q, A, p \rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p \rangle$ to x

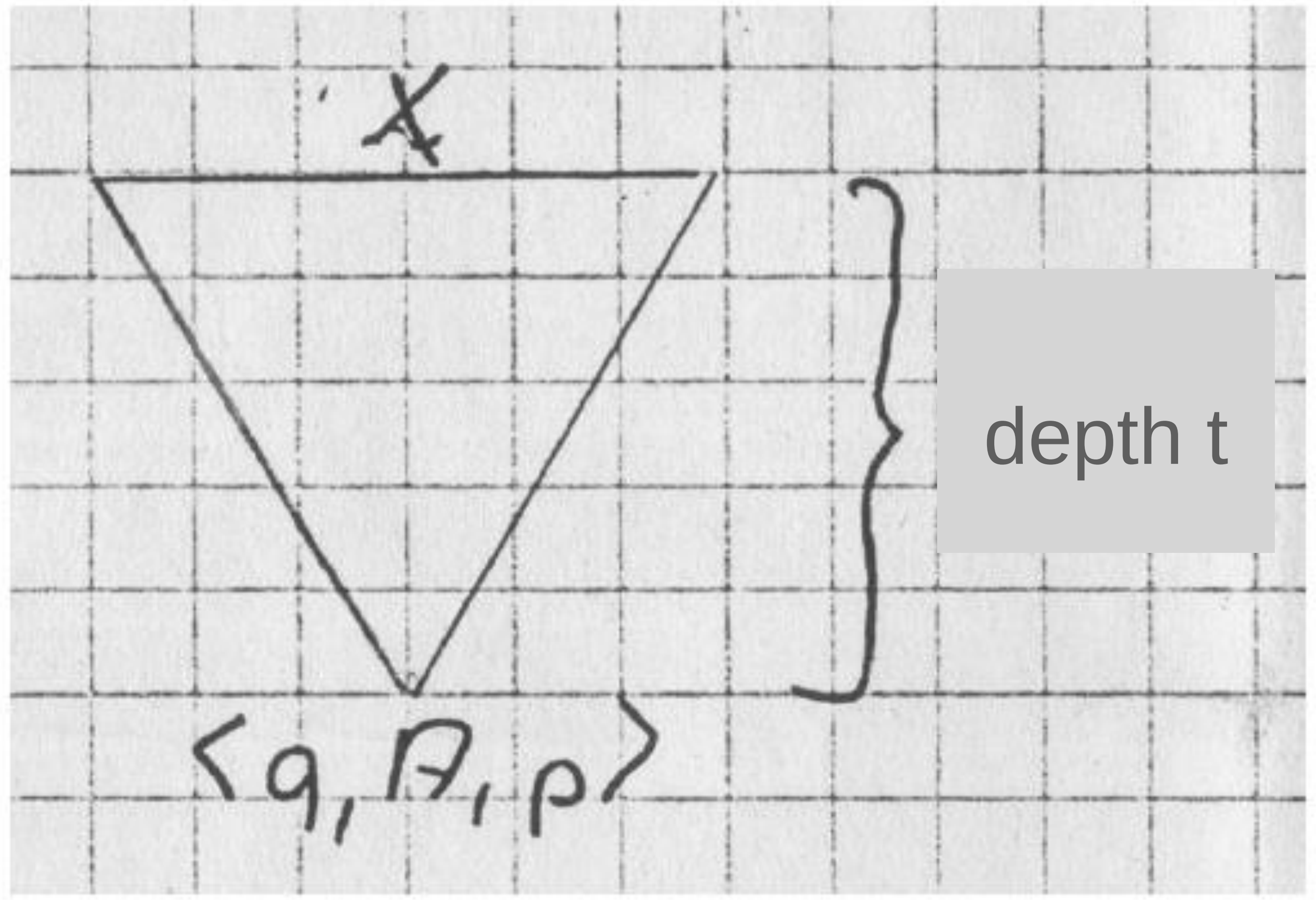


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

- $t - 1 \rightarrow t$

construction 2: see figure 9 push a with $A \neq \epsilon$

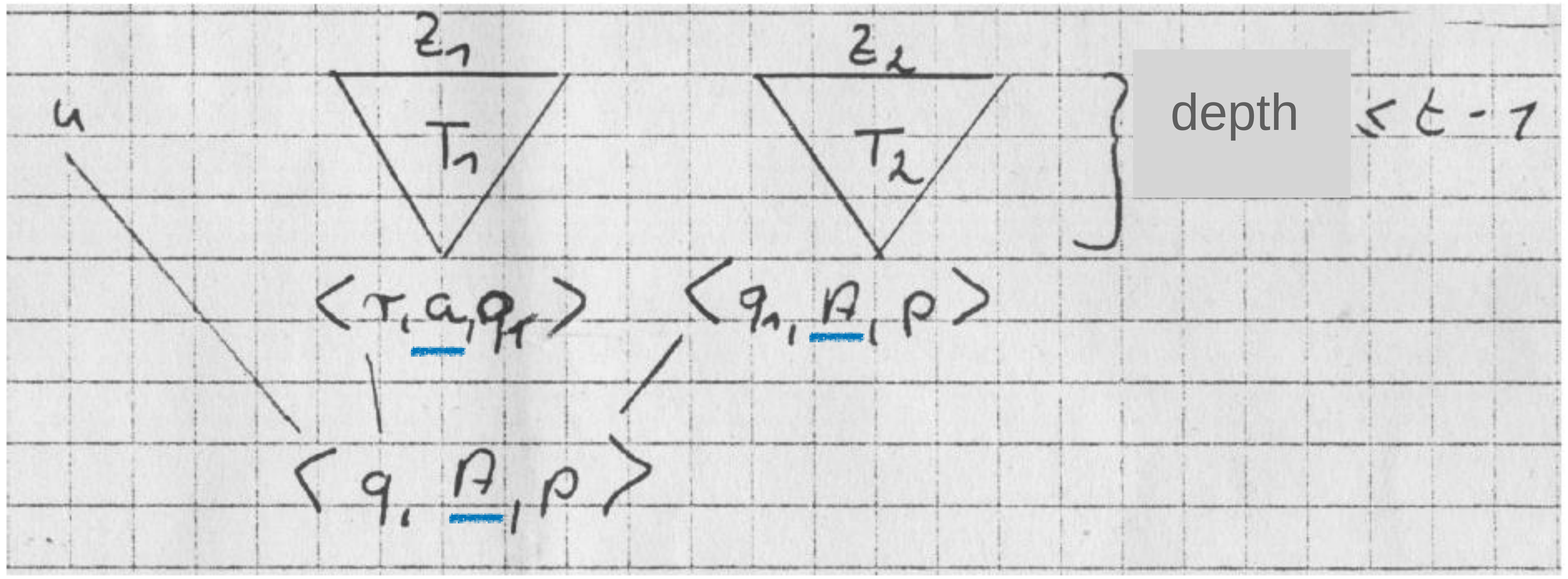


Figure 9: derivation tree for case 2 of the construction. The input word is $x = uz_1z_2$

$$\begin{array}{llll} (q, x, A) & \vdash & (r, z_1 z_2, aA) & \text{(construction of } G\text{)} \\ (r, z_1, a) & \vdash^* & (q_1, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_1\text{)} \end{array}$$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

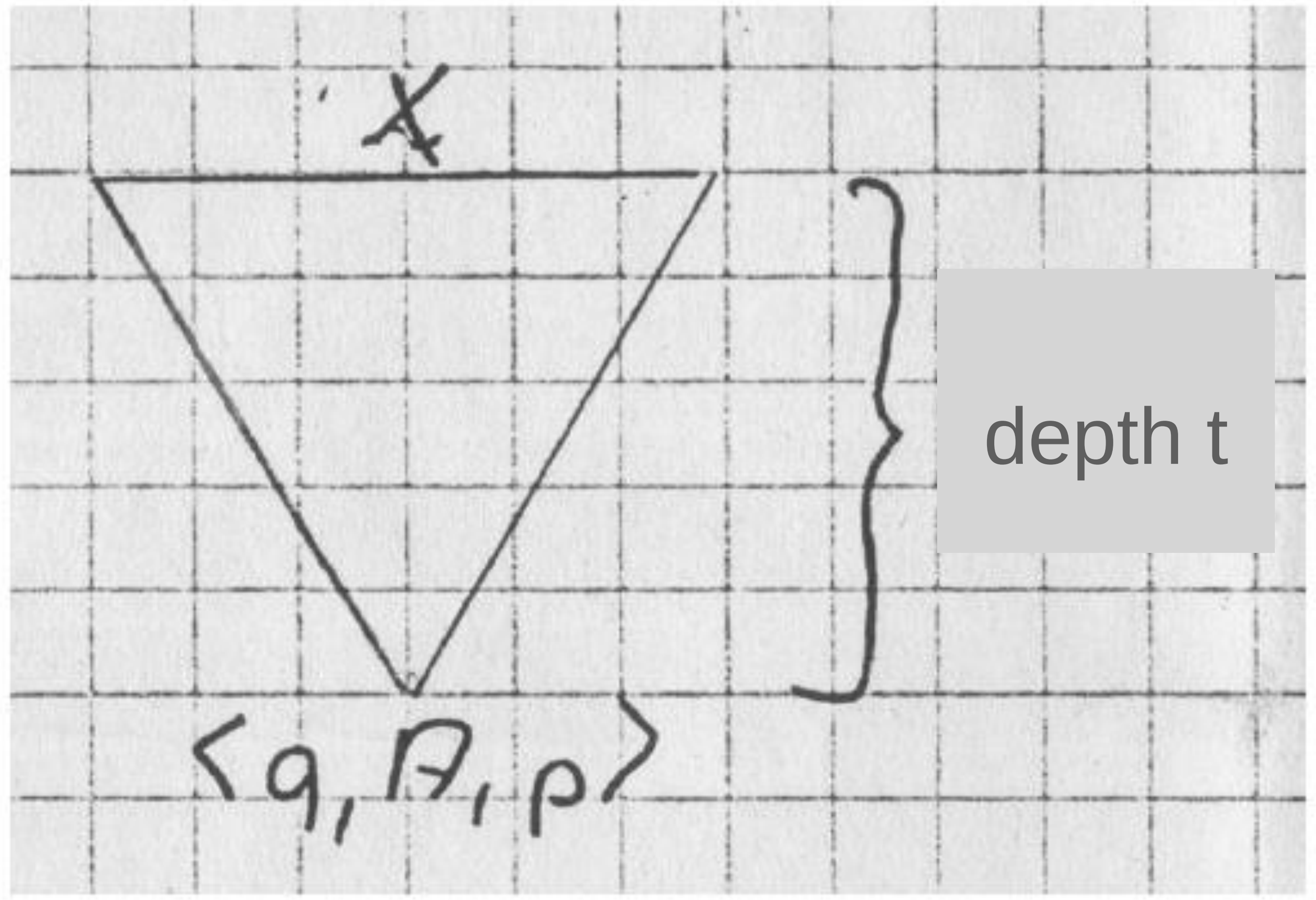


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

- $t - 1 \rightarrow t$

construction 2: see figure 9

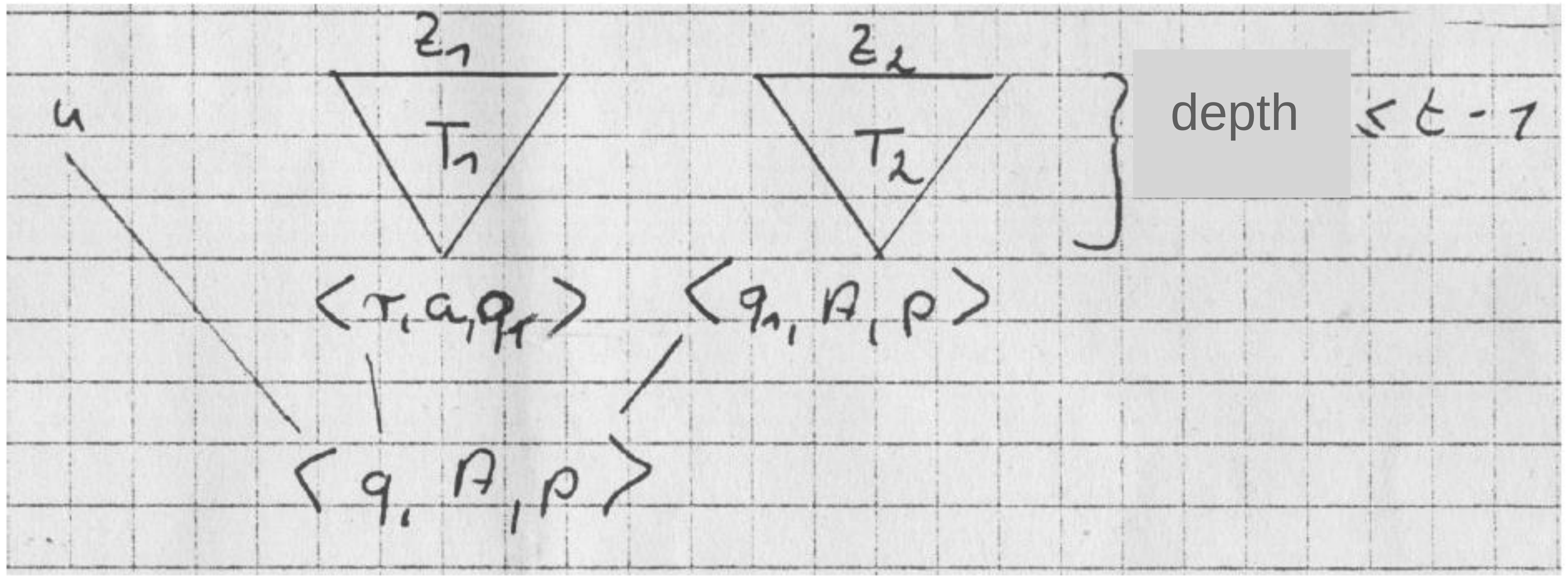


Figure 9: derivation tree for case 2 of the construction. The input word is $x = uz_1z_2$

$$\begin{array}{llll} (q,x,A) & \vdash & (r, z_1z_2, aA) & \text{(construction of } G\text{)} \\ (r, z_1, a) & \vdash^* & (q_1, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_1\text{)} \end{array}$$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

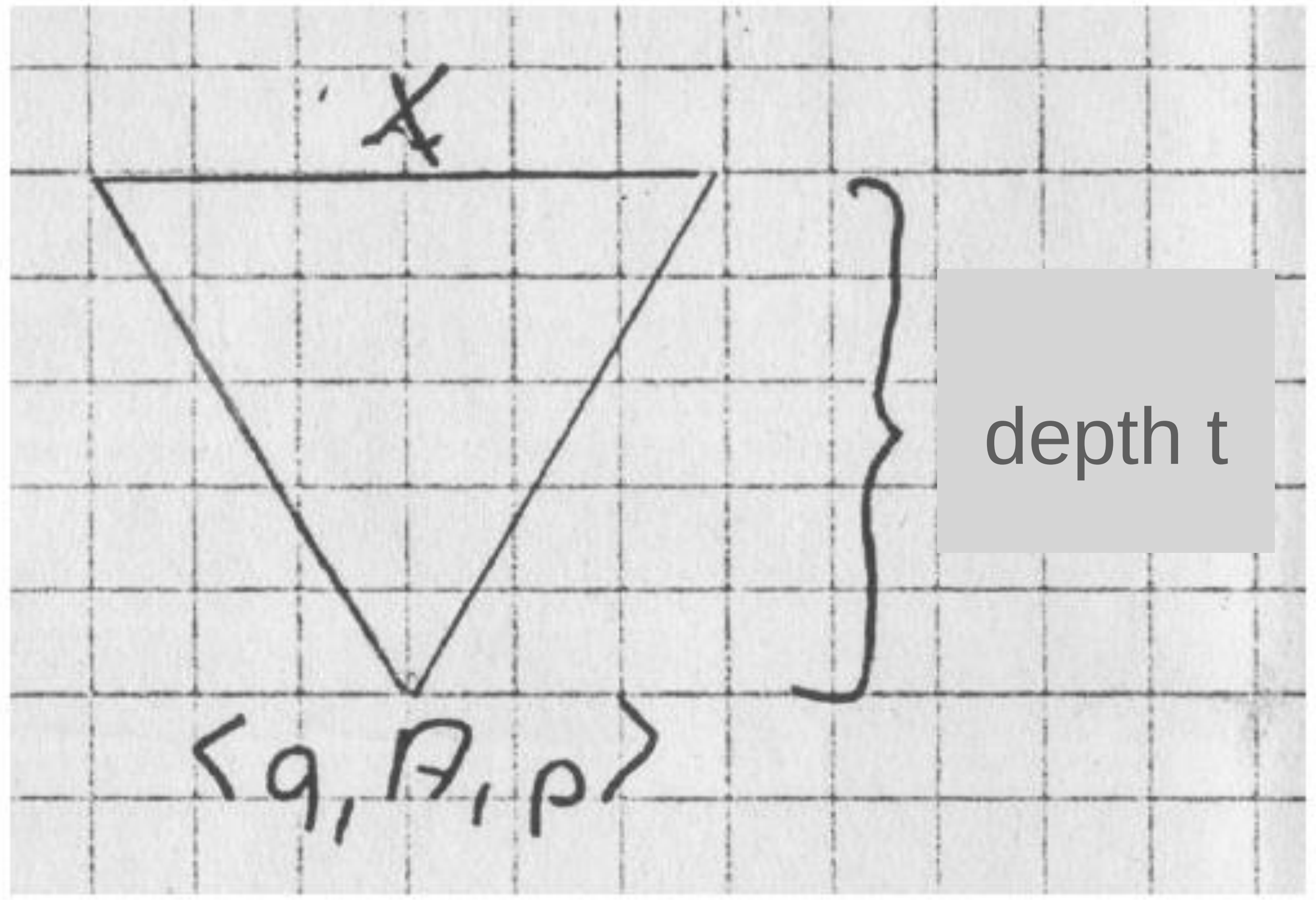


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

- $t - 1 \rightarrow t$

construction 2: see figure 9

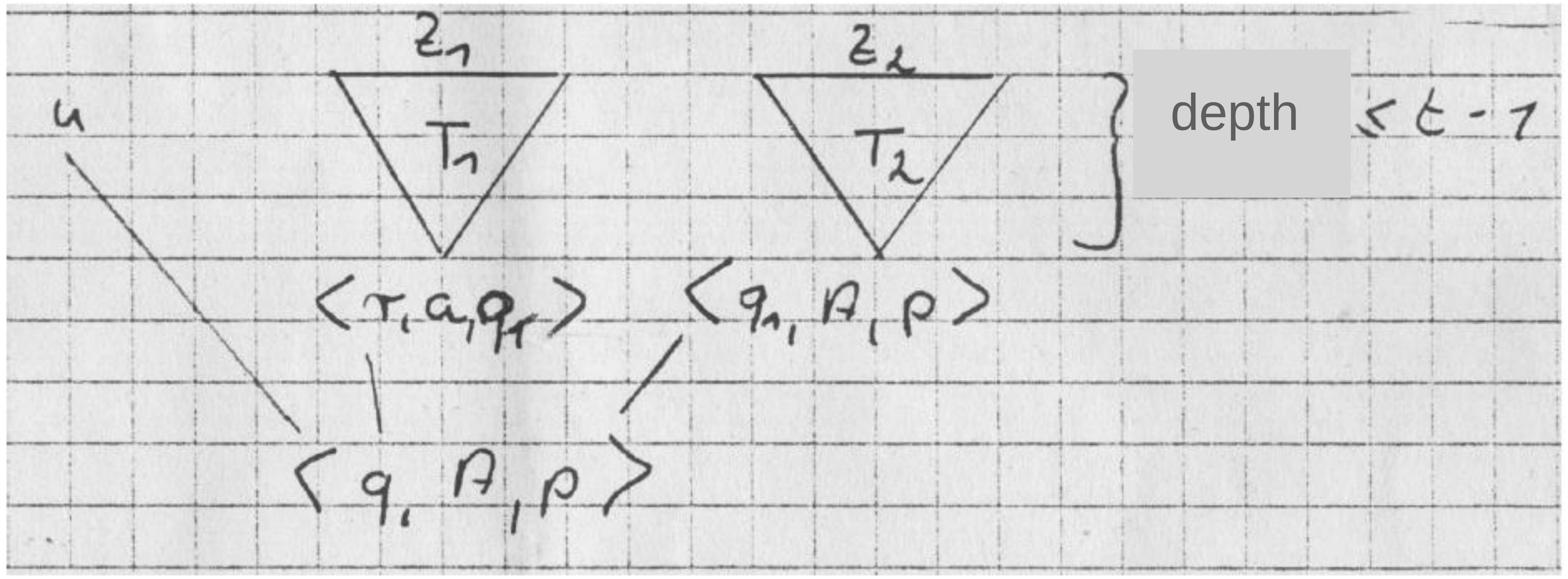


Figure 9: derivation tree for case 2 of the construction. The input word is $x = uz_1z_2$

$$\begin{array}{llll} (q, x, A) & \vdash & (r, z_1z_2, aA) & \text{(construction of } G) \\ (r, z_1, a) & \vdash^* & (q_1, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_1) \end{array}$$

hence

$$\begin{array}{llll} (r, z_1z_2, aA) & \vdash^* & (q_1, z_2, A) & \text{(pushdown automata work this way, def of } \vdash) \\ & \vdash^* & (p, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_2) \end{array}$$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

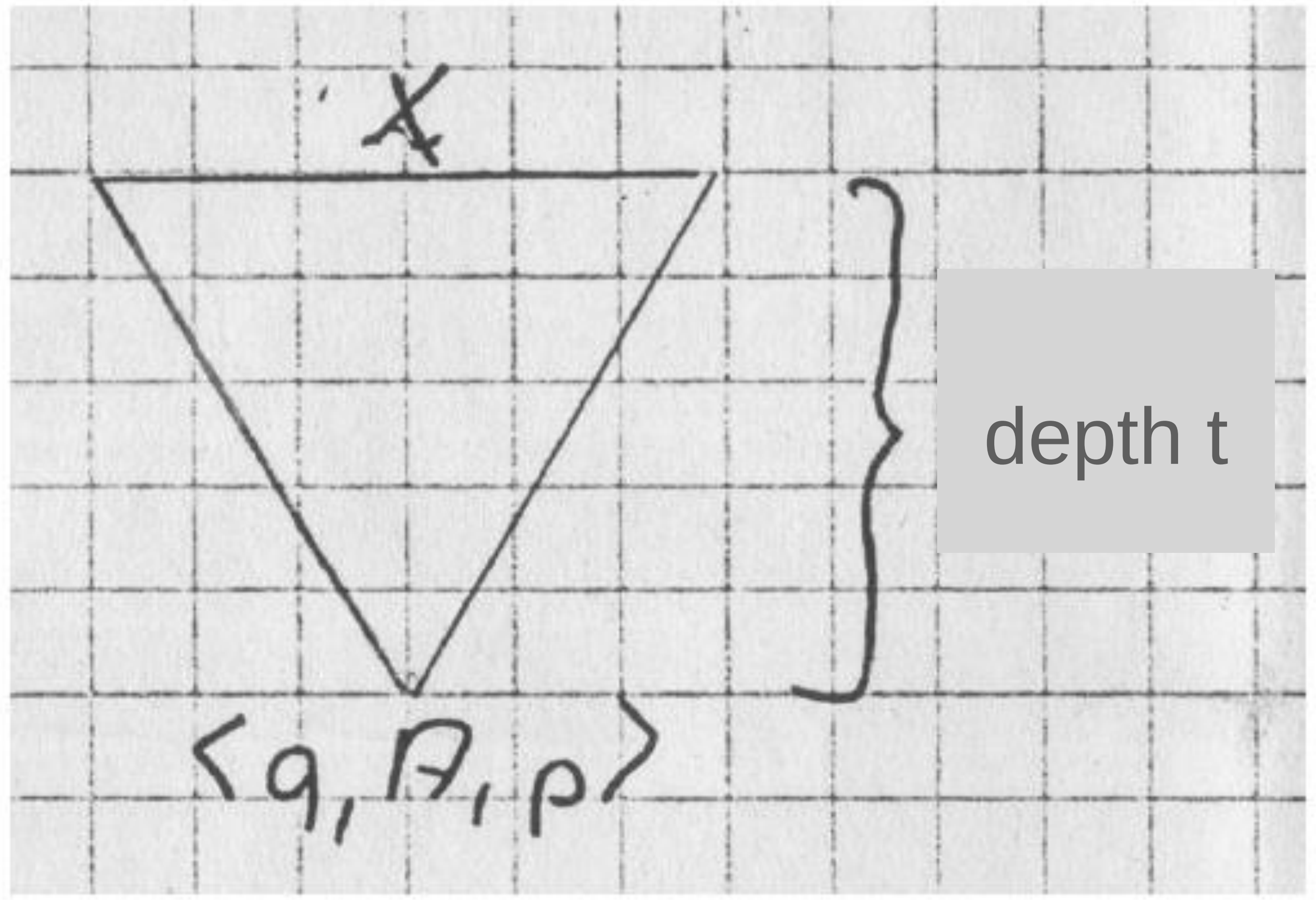


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

- $t-1 \rightarrow t$

construction 2: see figure 9

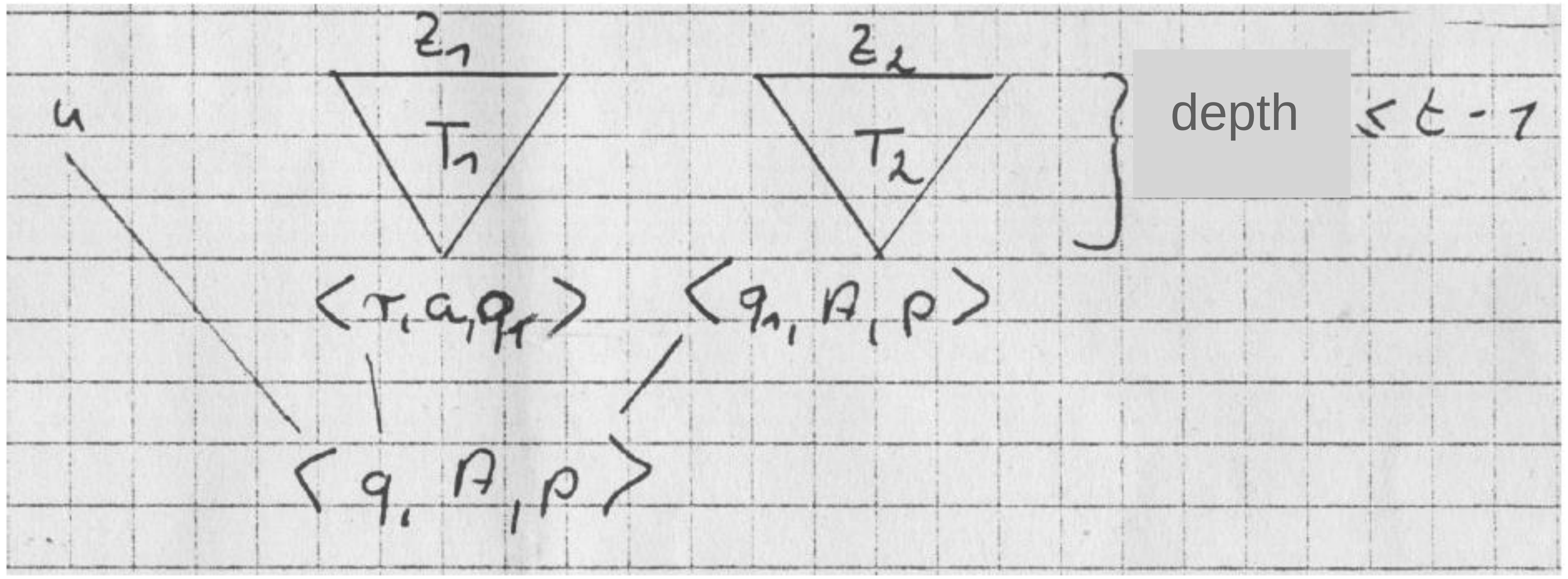


Figure 9: derivation tree for case 2 of the construction. The input word is $x = uz_1z_2$

$$\begin{array}{llll} (q,x,A) & \vdash & (r, z_1z_2, aA) & \text{(construction of } G) \\ (r, z_1, a) & \vdash^* & (q_1, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_1) \end{array}$$

hence

$$\begin{array}{llll} (r, z_1z_2, aA) & \vdash^* & (q_1, z_2, A) & \text{(pushdown automata work this way, def of } \vdash) \\ & \vdash^* & (p, \varepsilon, \varepsilon) & \text{(induction hypothesis for } T_2) \end{array}$$

construction 3: exercise

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

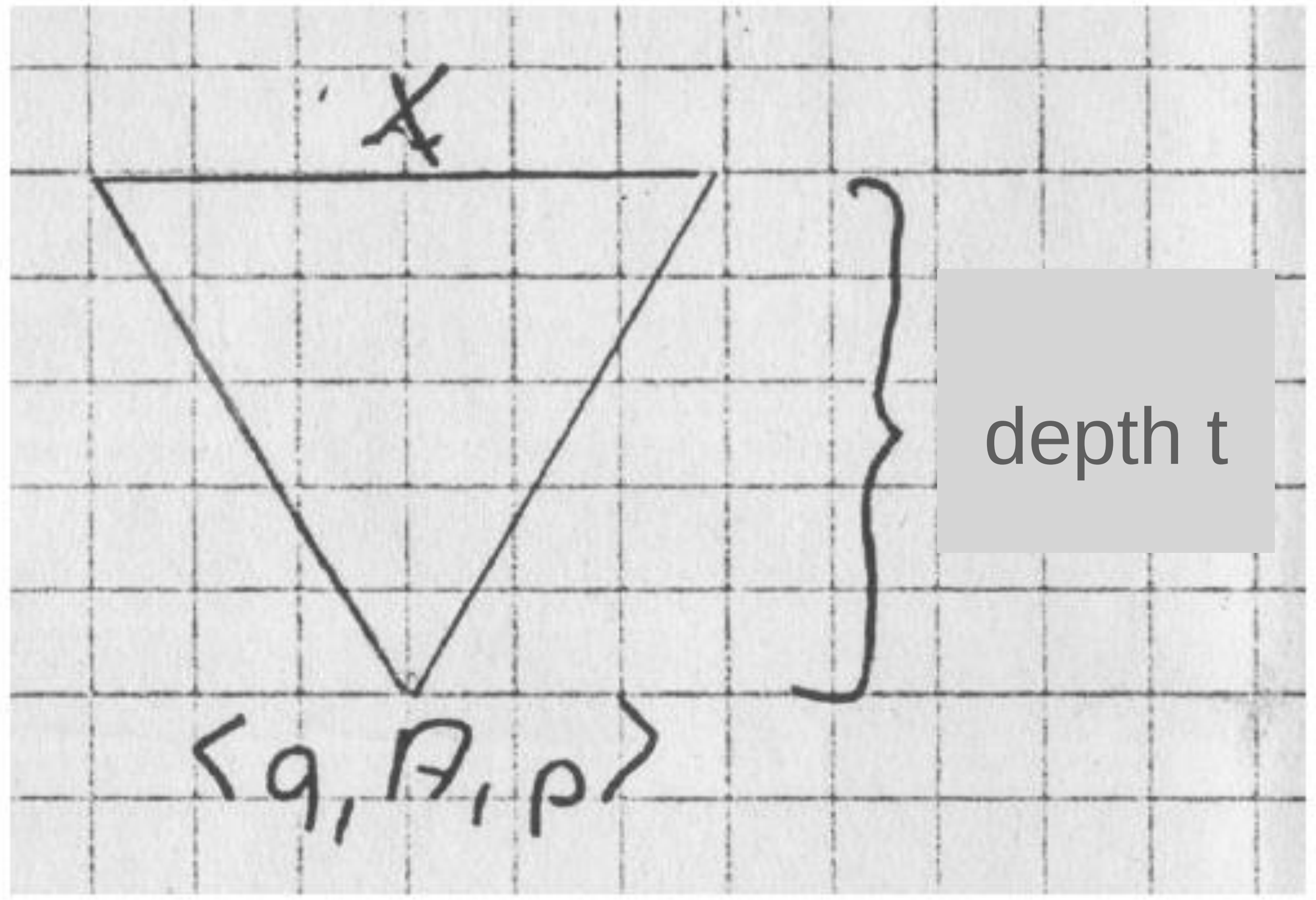


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

construction 4: pop with $A \neq \epsilon$

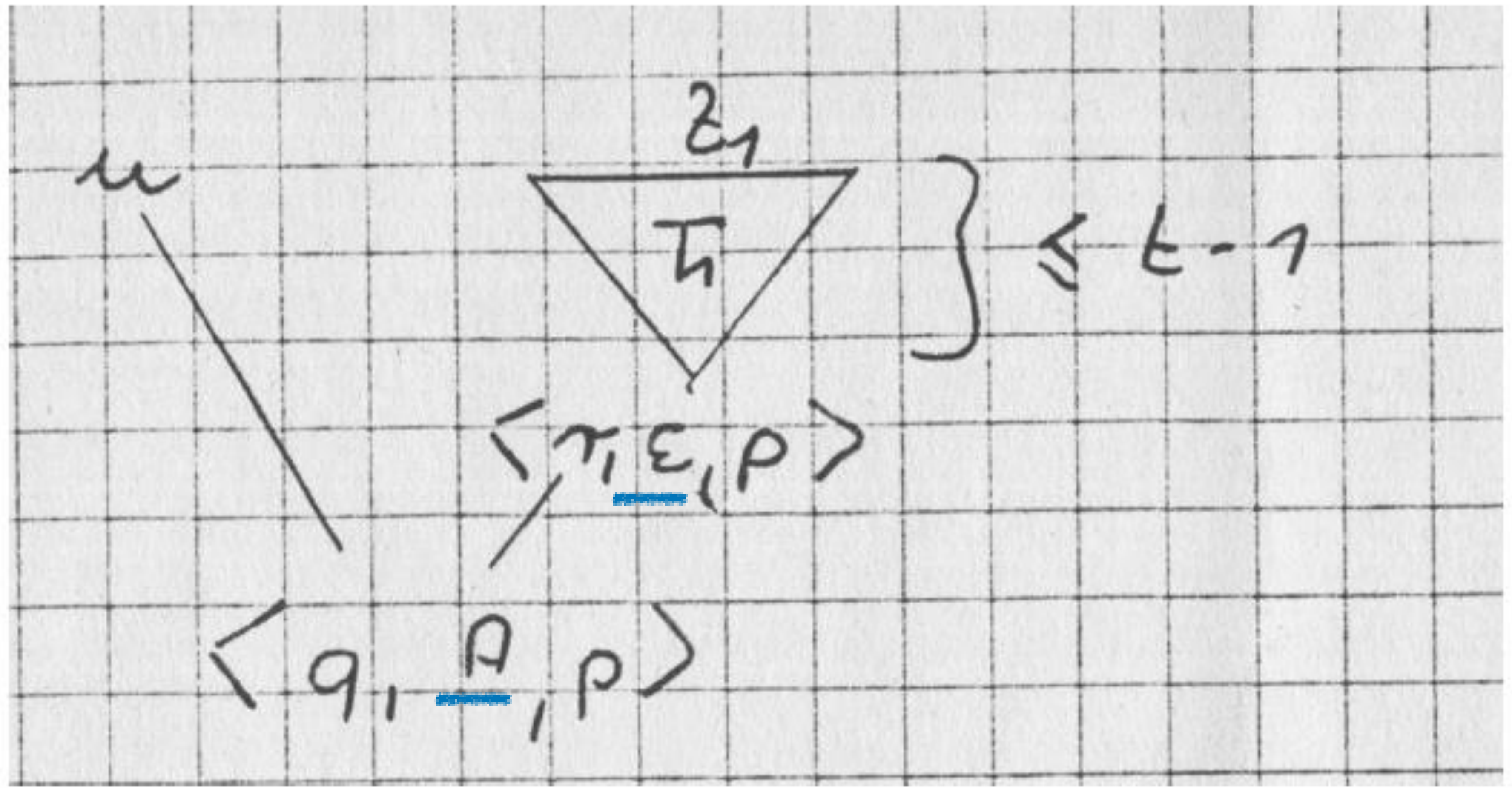


Figure 10: derivation tree for case 4 of the construction. The input word is $x = uz_1$

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \rightarrow (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

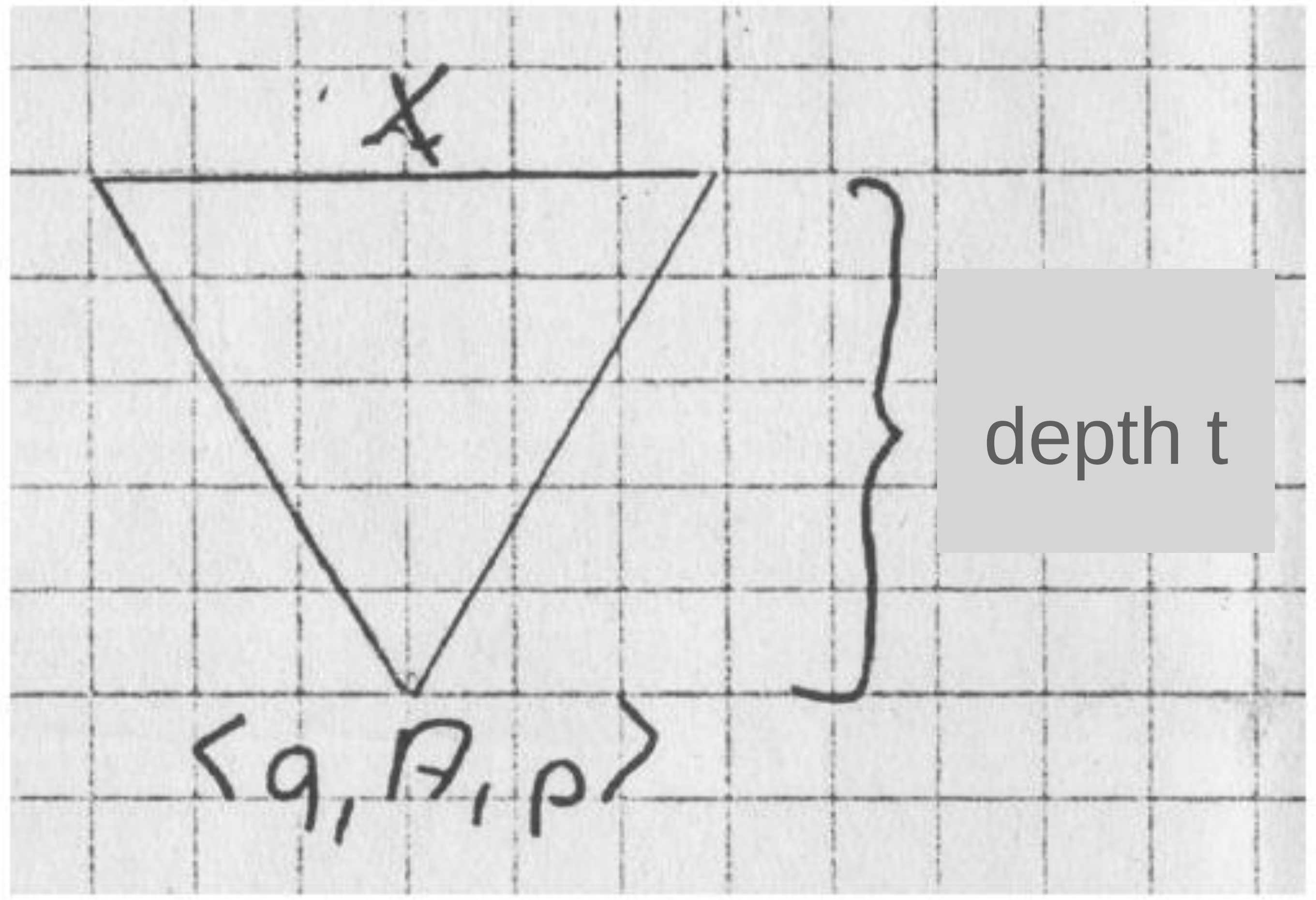


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q,x,A) \vdash^* (p,\varepsilon,\varepsilon)$.

construction 4: pop with $A \neq \epsilon$

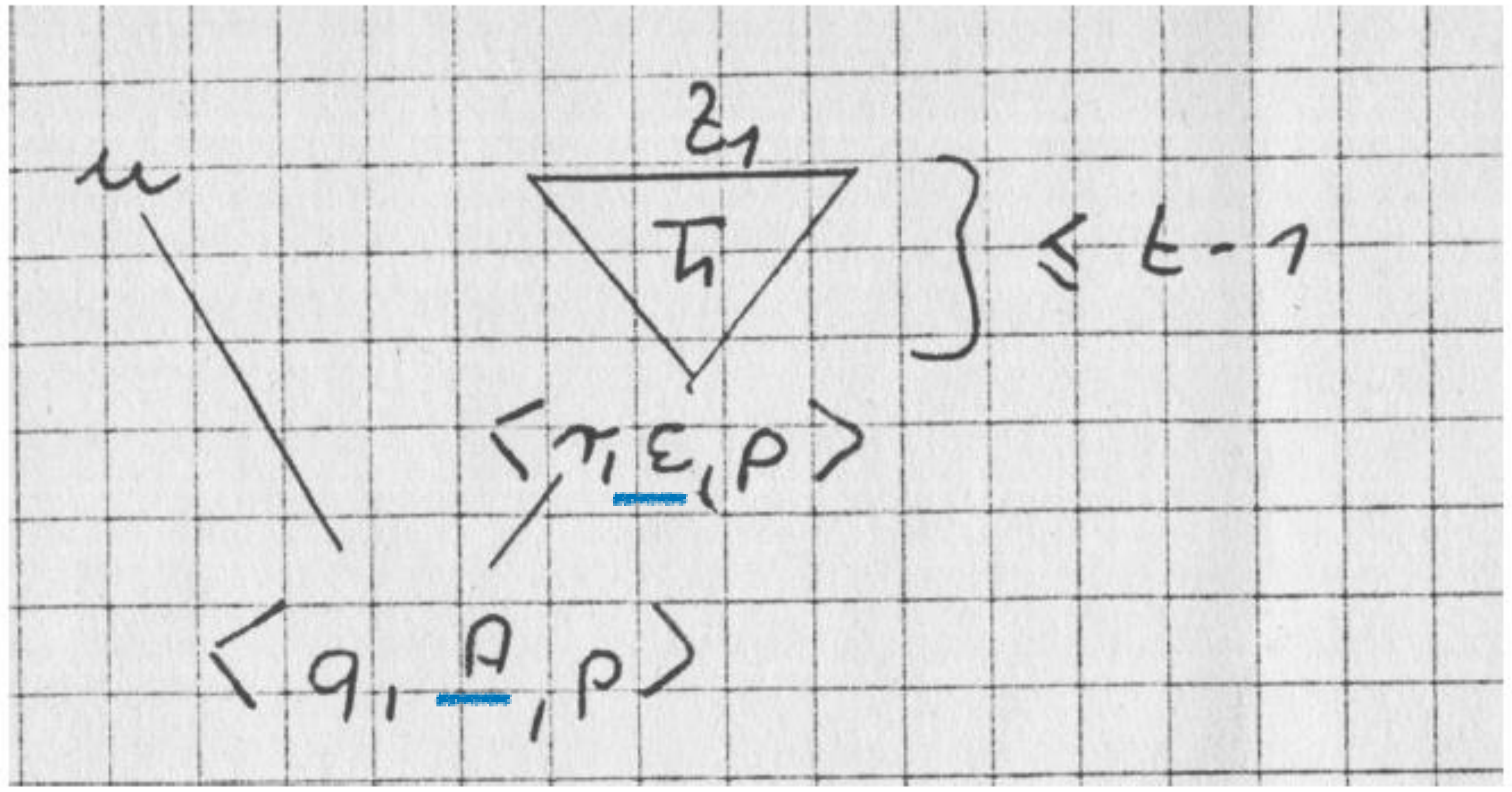


Figure 10: derivation tree for case 4 of the construction. The input word is $x = uz_1$

$$\begin{array}{rcll} (q,x,A) & \vdash & (r,z_1,\varepsilon) & \text{(construction of } G) \\ & \vdash^* & (p,\varepsilon,\varepsilon) & \text{(induction hypothesis for } T_1) \end{array}$$

Lemma 9.

$$x \in L(\langle q,A,p\rangle) \rightarrow (q,x,A) \vdash^* (p,\varepsilon,\varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q,A,p\rangle$ to x

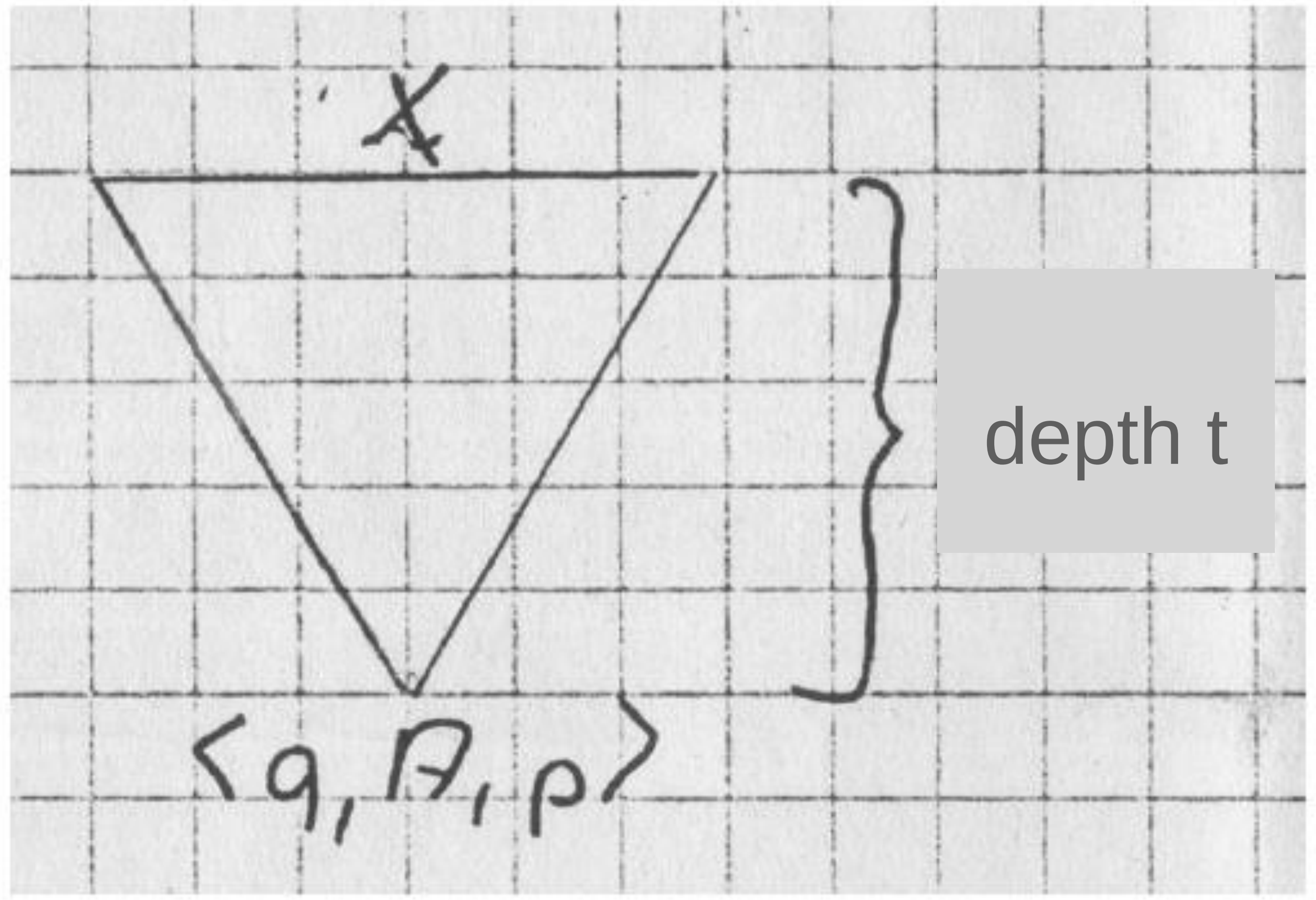


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q,x,A) \vdash^* (p,\varepsilon,\varepsilon)$.

construction 4: pop with $A \neq \epsilon$

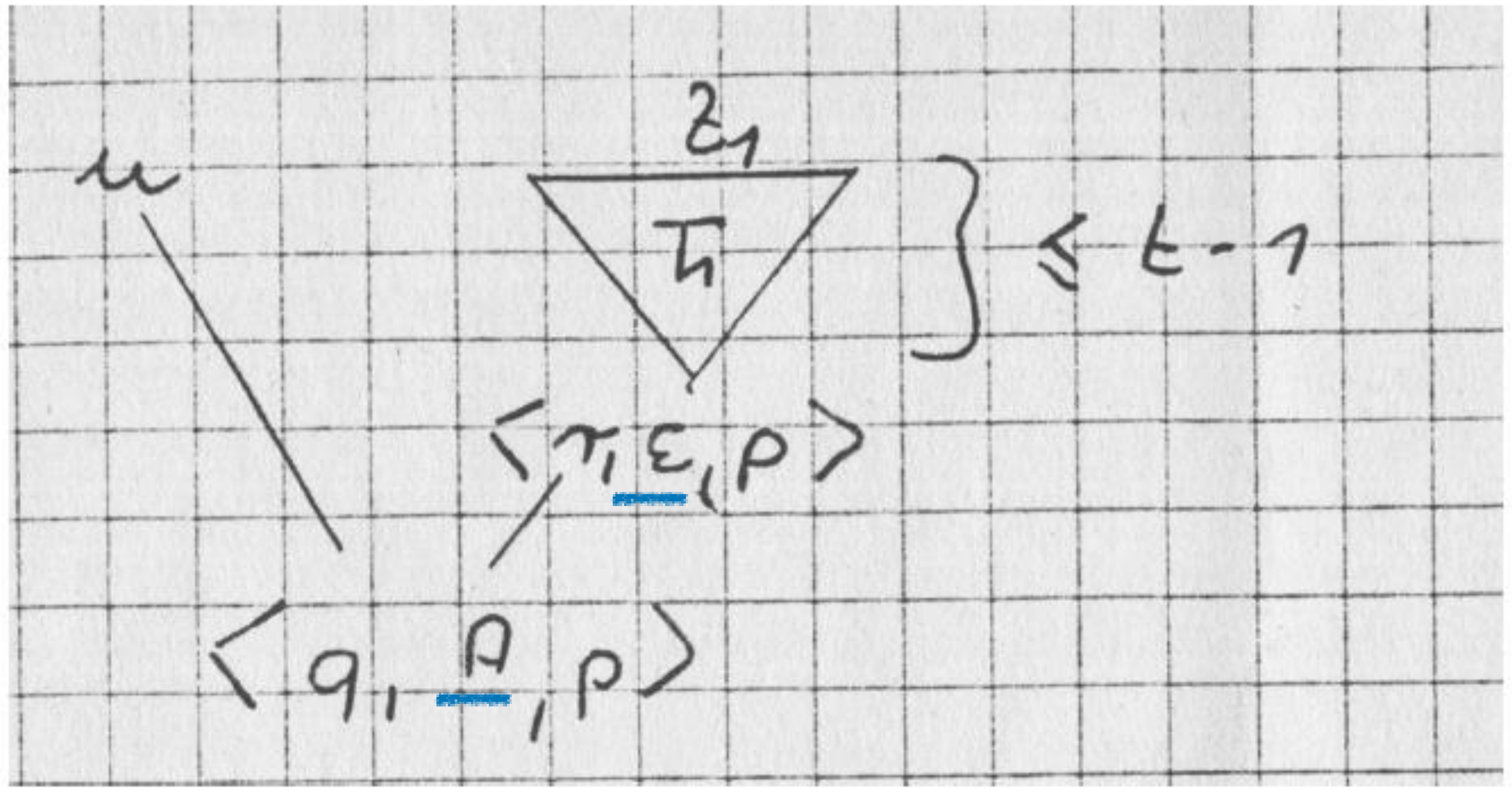


Figure 10: derivation tree for case 4 of the construction. The input word is $x = uz_1$

$$\begin{array}{lcll} (q,x,A) & \vdash & (r,z_1,\varepsilon) & \text{(construction of } G) \\ & \vdash^* & (p,\varepsilon,\varepsilon) & \text{(induction hypothesis for } T_1) \end{array}$$

~~construction 5: exercise~~

classical place for errors (although I did not find it yet)

Lemma 9.

$$x \in L(\langle q, A, p\rangle) \to (q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$$

Proof by induction on depth t of derivation tree from $\langle q, A, p\rangle$ to x

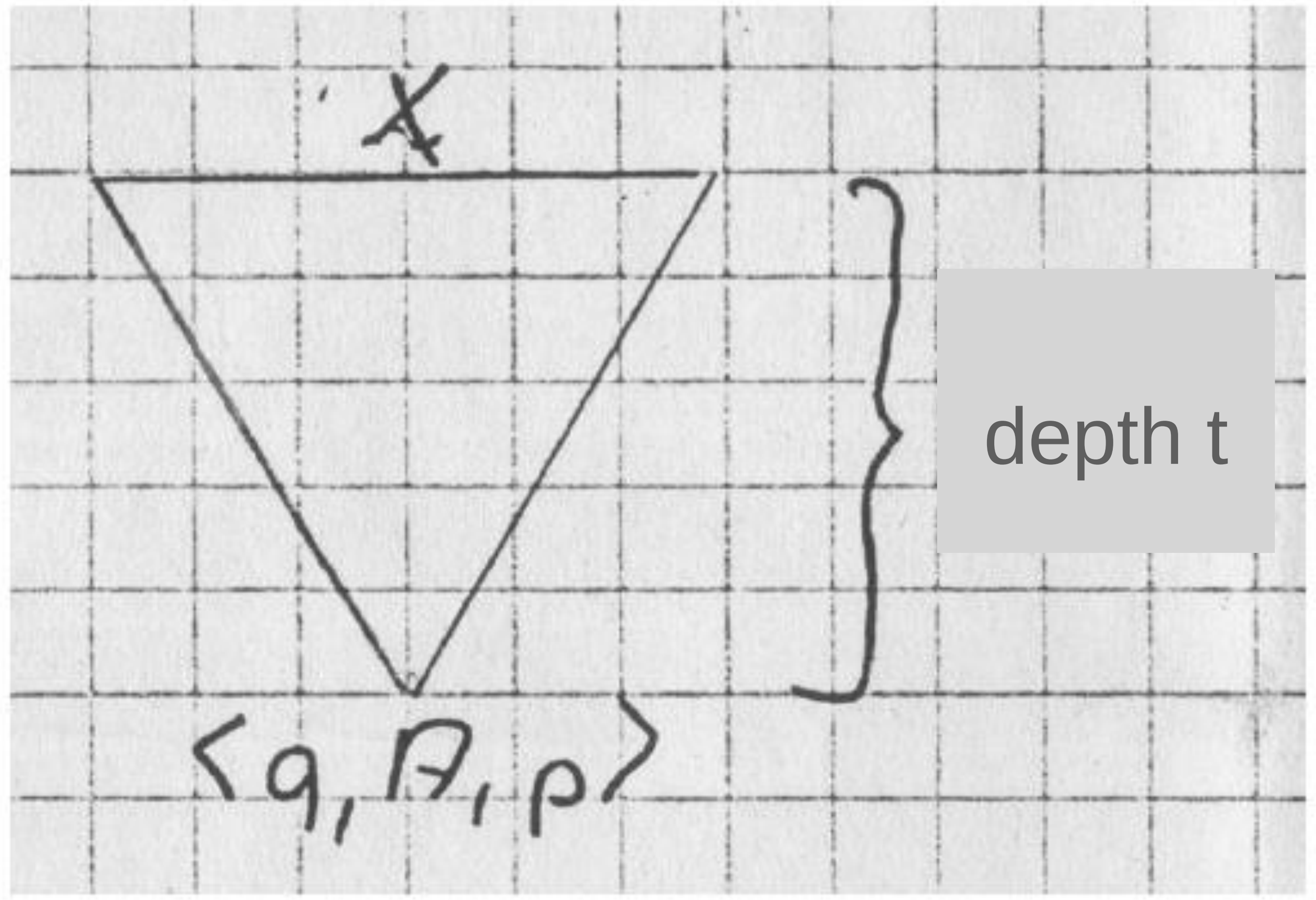


Figure 8: by induction on the depth t of the derivation tree we want to conclude $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

lemma implies claim $w \in L(G) = L(S)$ has derivation tree as in figure 11 .

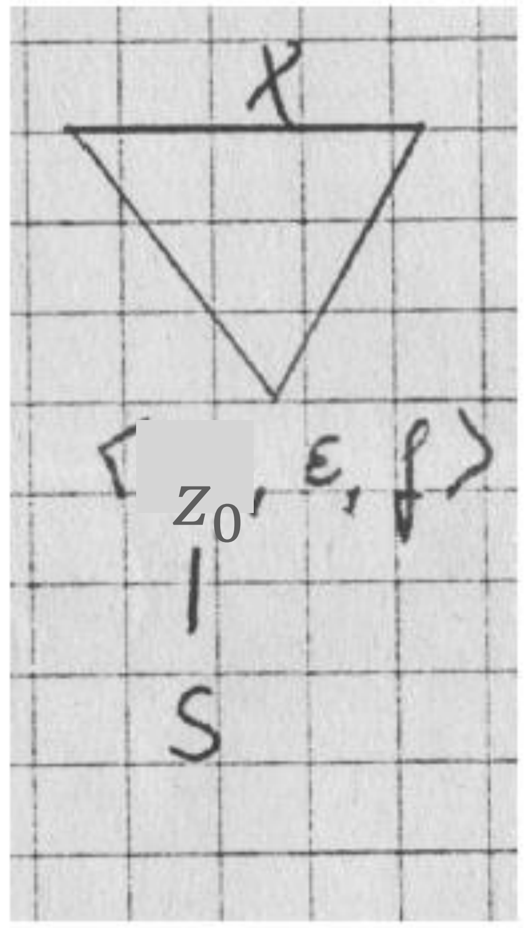


Figure 11: derivation tree for x from start symbol S

The lemma implies

$$(z_0, x, \varepsilon) \vdash^* (f, \varepsilon, \varepsilon)$$

$$f \in Z_A \to x \in L(M)$$

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q, x, A) \vdash^t (p, \varepsilon, \varepsilon)$$

then

$$\langle q, A, p \rangle \rightarrow^* x$$

5.3 $x \in L(M) \to x \in L(G)$

Lemma 10. *For all* $x \in \Sigma^*$ *and* $A \in \Gamma_\varepsilon$ *and* $p, q \in Z$ *holds: if*

$$(q,x,A) \vdash^t (p,\varepsilon,\varepsilon)$$

then

$$\langle q,A,p \rangle \to^* x$$

lemma implies claim: choose

$$q = z_0 \,,\ A = \varepsilon,\ p = f \in Z_A$$

Then

$$S \to \langle z_0, \varepsilon, f \rangle \to^* x$$

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q, x, A) \vdash^t (p, \varepsilon, \varepsilon)$$

then

$$\langle q, A, p \rangle \rightarrow^* x$$

proof of lemma by induction on t

- $t = 0$

$$q = p \ , \ x = \varepsilon \ , \ A = \varepsilon$$

$$\langle q, \varepsilon, q \rangle \rightarrow \varepsilon \quad \text{(construction 6)}$$

5.3 $x \in L(M) \to x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q, x, A) \vdash^t (p, \varepsilon, \varepsilon)$$

then

$$\langle q, A, p \rangle \to^* x$$

- $t - 1 \to t$:

  case 1: computation starts with *push a* for some $a \in \Gamma_\varepsilon$.

  $$(q, x, A) \vdash (r, z, aA) \vdash^{t-1} (p, \varepsilon, \varepsilon) \quad , \quad x = uz \, , \, u \in \Sigma_\varepsilon$$

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q, x, A) \vdash^t (p, \varepsilon, \varepsilon)$$

then

$$\langle q, A, p \rangle \rightarrow^* x$$

- $t - 1 \rightarrow t$:

  case 1: computation starts with *push a* for some $a \in \Gamma_\varepsilon$.

  $$(q, x, A) \vdash (r, z, aA) \vdash^{t-1} (p, \varepsilon, \varepsilon) \quad , \quad x = uz \ , \ u \in \Sigma_\varepsilon$$

  then

  $$(r, push\ a) \in \delta(q, u, A) \quad \text{or} \quad (r, push\ a) \in \delta(q, u, \varepsilon)$$

  in both cases by construction 2 and 3 there is production

  $$\langle q, A, p \rangle \rightarrow u \langle r, a, q_1 \rangle \langle q_1, A, p \rangle \quad \text{for all} \quad q_1 \in Z$$

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q,x,A) \vdash^t (p,\varepsilon,\varepsilon)$$

then

$$\langle q,A,p\rangle \rightarrow^* x$$

- $t-1 \rightarrow t$:

case 1: computation starts with *push* a for some $a \in \Gamma_\varepsilon$.

$$(q,x,A) \vdash (r,z,aA) \vdash^{t-1} (p,\varepsilon,\varepsilon) \quad , \quad x = uz \; , \; u \in \Sigma_\varepsilon$$

then

$$(r, push\ a) \in \delta(q,u,A) \quad \text{or} \quad (r, push\ a) \in \delta(q,u,\varepsilon)$$

in both cases by construction 2 and 3 there is production

$$\langle q,A,p\rangle \rightarrow u\langle r,a,q_1\rangle\langle q_1,A,p\rangle \quad \text{for all} \quad q_1 \in Z$$

Determining q_1: choose t_1 as first t when a is popped from stack.

$$(r,z,aA) \vdash^{t_1} (q_1,z_2,A) \vdash^{t_2} (p,\varepsilon,\varepsilon) \quad \text{with} \quad z = z_1 z_2 \; , \; t_1 + t_2 = t$$

M is pushdown automaton/definition of $\vdash$:

$$(r,z_1,a) \vdash^{t_1} (q_1,\varepsilon,\varepsilon)$$

induction hypothesis gives derivation trees T_1, T_2 of figure 12

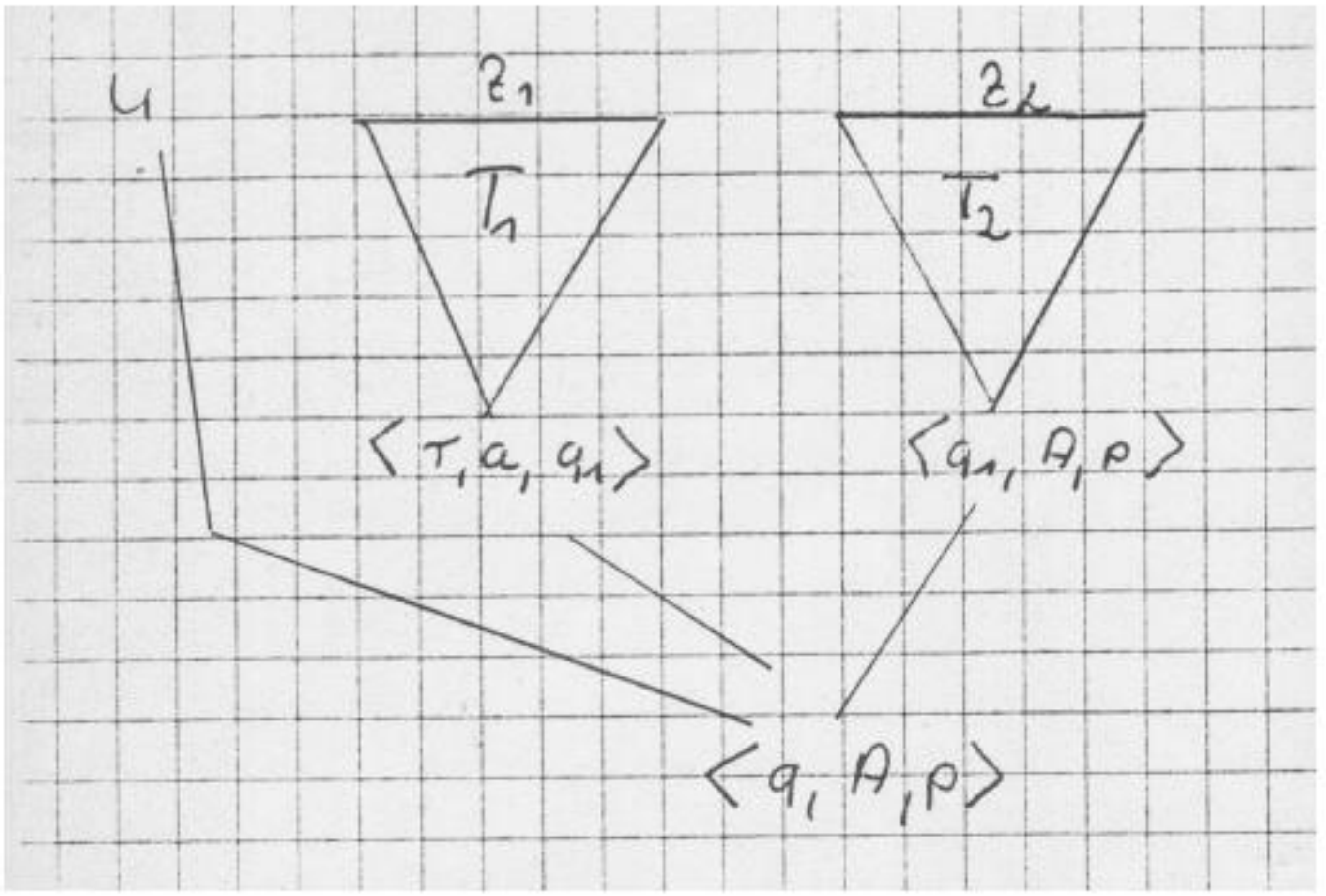


Figure 12: derivation tree for x if the computation starts with *push* a

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p, q \in Z$ holds: if*

$$(q,x,A) \vdash^t (p,\varepsilon,\varepsilon)$$

then

$$\langle q,A,p\rangle \rightarrow^* x$$

case 2: computation starts with pop

$$(q,x,A) \vdash (r,z,\varepsilon) \vdash^{t-1} (p,\varepsilon,\varepsilon) \quad , \quad x = uz \ , \ u \in \Sigma_\varepsilon$$

then

$$(r,pop) \in \delta(q,u,A) \quad \text{or} \quad (r,pop) \in \delta(q,u,\varepsilon)$$

in both cases by construction 4 and 5 there is production

$$\langle q,A,p\rangle \rightarrow u\langle r,\varepsilon,p\rangle$$

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p,q \in Z$ holds: if*

$$(q,x,A) \vdash^t (p,\varepsilon,\varepsilon)$$

then

$$\langle q,A,p \rangle \rightarrow^* x$$

case 2: computation starts with *pop*

$$(q,x,A) \vdash (r,z,\varepsilon) \vdash^{t-1} (p,\varepsilon,\varepsilon) \quad , \quad x = uz \text{ , } u \in \Sigma_\varepsilon$$

then

$$(r,pop) \in \delta(q,u,A) \quad \text{or} \quad (r,pop) \in \delta(q,u,\varepsilon)$$

in both cases by construction 4 ~~and 5~~ there is production

$$\langle q,A,p \rangle \rightarrow u\langle r,\varepsilon,p \rangle$$

induction hypothesis gives derivation tree T_1 of figure 13

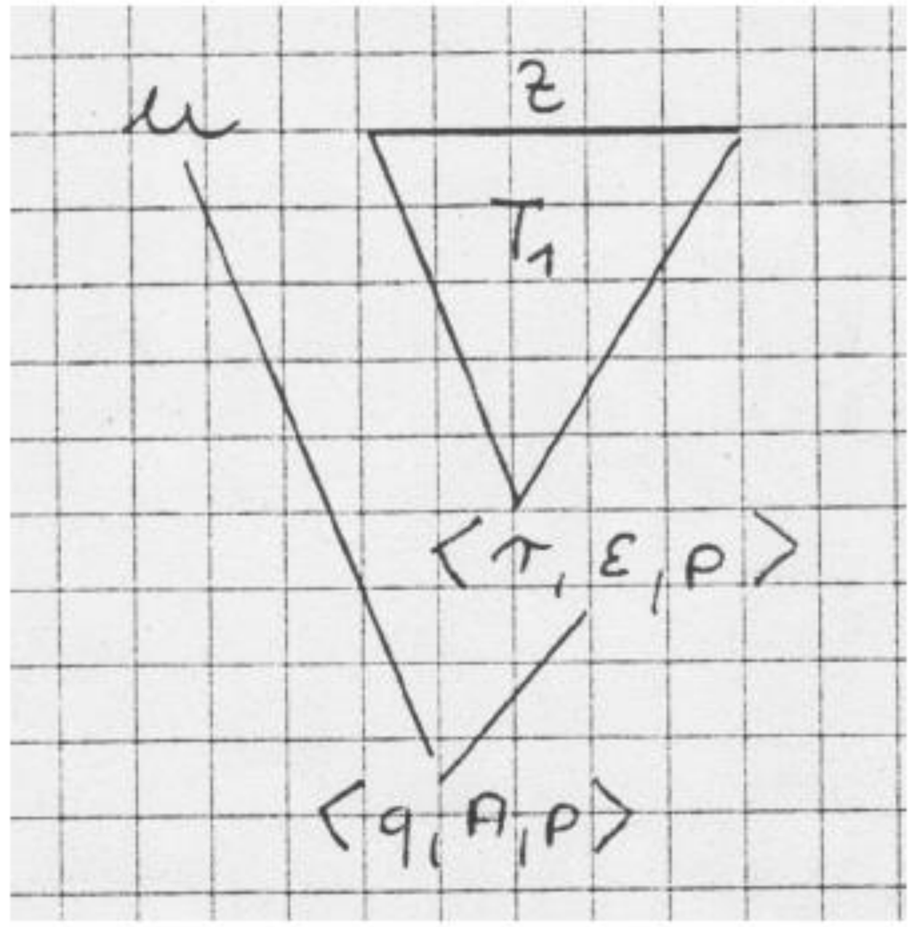


Figure 13: derivation tree for x if the computation starts with *pop*

5.3 $x \in L(M) \rightarrow x \in L(G)$

Lemma 10. *For all $x \in \Sigma^*$ and $A \in \Gamma_\varepsilon$ and $p,q \in Z$ holds: if*

$$(q,x,A) \vdash^t (p,\varepsilon,\varepsilon)$$

then

$$\langle q,A,p\rangle \rightarrow^* x$$

case 2: computation starts with *pop*

$$(q,x,A) \vdash (r,z,\varepsilon) \vdash^{t-1} (p,\varepsilon,\varepsilon) \quad , \quad x = uz \, , \, u \in \Sigma_\varepsilon$$

then

$$(r,pop) \in \delta(q,u,A) \quad \text{or} \quad (r,pop) \in \delta(q,u,\varepsilon)$$

in both cases by construction 4 ~~and 5~~ there is production

$$\langle q,A,p\rangle \rightarrow u\langle r,\varepsilon,p\rangle$$

done

remember: this is
hardcore CS
from 1962

I was 11 years old

induction hypothesis gives derivation tree T_1 of figure 13

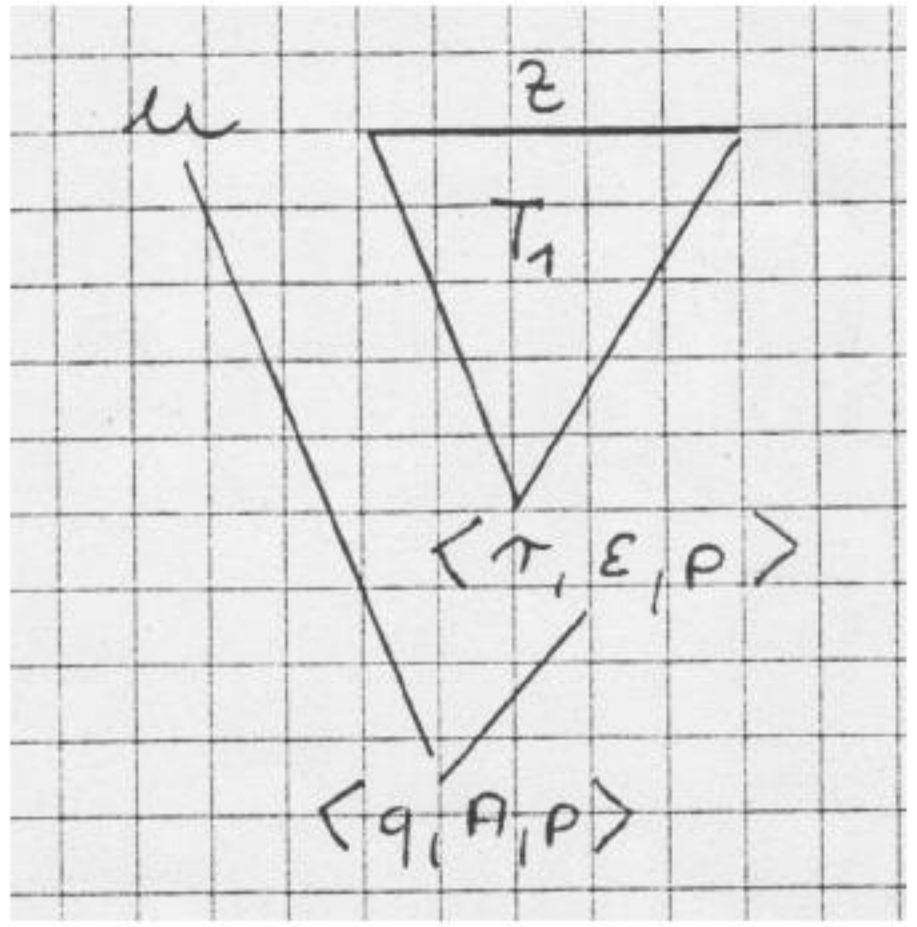


Figure 13: derivation tree for x if the computation starts with *pop*