

## Processor propositiecalculus voor Automath

**Citation for published version (APA):**

Bruijn, de, N. G. (1968). *Processor propositiecalculus voor Automath*. Technische Hogeschool Eindhoven.

**Document status and date:**

Gepubliceerd: 01/01/1968

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

Type definitities

$\langle \text{label} \rangle ::= \langle \text{positive integer} \rangle$   
 $\langle \text{proper expression} \rangle ::= \langle \text{label} \rangle \mid \langle \text{label} \rangle \langle \langle \text{proper expression string} \rangle \rangle$   
 $\langle \text{label string} \rangle ::= \text{emptystring} \mid \langle \text{label string} \rangle , \langle \text{label} \rangle$   
 $\langle \text{proper expression string} \rangle ::= \text{emptyproperexpressionstring} \mid$   
 $\quad \langle \text{proper expression string} \rangle , \langle \text{proper expression} \rangle$   
 $\langle \text{expression} \rangle ::= \text{EB} \mid \text{PM} \mid \langle \text{proper expression} \rangle$   
 $\langle \text{category} \rangle ::= \text{elt} \mid \text{set} \mid \text{bool} \mid \text{true} \mid \text{ax}$   
 $\langle \text{line} \rangle ::= \langle \text{label} \rangle , \langle \text{boolean} \rangle , \langle \text{label string} \rangle , \langle \text{expression} \rangle ,$   
 $\quad \langle \text{category} \rangle , \langle \text{hint} \rangle$   
 $\langle \text{hint} \rangle ::= \langle \text{integer} \rangle , \langle \text{integer} \rangle \mid \text{emptyhint}$   
 $\langle \text{line string} \rangle ::= \text{emptybook} \mid \langle \text{line string} \rangle , \langle \text{line} \rangle$

Elementaire procedures

Als  $E \in \langle \text{expression} \rangle$  dan  
 $\text{proper}(E) ::= E \in \langle \text{proper expression} \rangle$   
 $\text{emblo}(E) ::= E = \text{EB}$   
 $\text{primno}(E) ::= E = \text{PN}$

Als  $E \in \langle \text{expression} \rangle$  en  $\text{proper}(E) \text{ true}$ , dan

$\text{head}(E) ::= \lambda$   
 $\text{substring}(E) ::= \mu$

als  $E = \lambda(\mu)$  met  $\lambda \in \langle \text{label} \rangle$ ,  $\mu \in \langle \text{proper expression string} \rangle$ .

Als  $\lambda \in \langle \text{label} \rangle$ ,  $\mu \in \langle \text{proper expression string} \rangle$  dan

$\text{buildexpression}(\lambda, \mu) ::= \lambda(\mu)$  als  $\mu$  nonempty  
 $\lambda$  als  $\mu$  empty

Als  $\lambda, \mu \in \langle \text{category} \rangle$  dan

$\text{categoryimplication}(\lambda, \mu) ::= (\lambda, \mu) \in [\text{category implication list}]$

Als  $\lambda \in \langle \text{line} \rangle$  dan worden zijn componenten aangeduid door resp.

$\text{labelof}(\lambda)$ ,  $\text{boolof}(\lambda)$ ,  $\text{labelstringof}(\lambda)$ ,  $\text{expressionof}(\lambda)$ ,  
 $\text{categoryof}(\lambda)$ ,  $\text{hintof}(\lambda)$

Formline  $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$  is de regel( $\langle \text{line} \rangle$ ) waarvan de componenten  $\alpha_1, \dots, \alpha_6$  zijn.

### Definities betreffende strings

THING betekent hier "label" of "proper expression" of "line"

Als  $M \in \langle \text{THINGstring} \rangle$  dan

$\text{length}(M) :=$  aantal THINGS in M.

Als k geheel,  $M \in \langle \text{THINGstring} \rangle$ ,  $0 < k \leq \text{length}(M)$  dan

$\text{member}(M, k) := k^e$  THING in M

Als k geheel,  $M \in \langle \text{THINGstring} \rangle$ ,  $0 \leq k \leq \text{length}(M)$  dan

$\text{frontsection}(M, k) :=$  de THINGstring bestaande uit de eerste k THINGS uit M (als  $k=0$  is het de emptyTHINGstring)

Als  $M \in \langle \text{THINGstring} \rangle$ ,  $t \in \langle \text{THING} \rangle$ , dan is  $\text{join}(M, t)$  de THINGstring die ontstaat door t aan de achterkant aan M toe te voegen

Voorbeeld van aan te bieden text. De volgende linestring, bestaande uit 10 lines, wordt door de procedure "entirelycorrect" in orde bevonden. Terwille van de leesbaarheid zijn de labels uit de eerste kolom, die eigenlijk 1 t.e.m. 10 moeten zijn, hier vervangen door andere identifiers.

u	u	true	EB	bool
v	u,v	true	EB	bool
ext	u,v	false	PN	bool
a	empty	false	PN	bool
ond	ond	true	a	true
x	ond,x	true	EB	bool
Ax	ond,x	false	ext(x,x)	ax
Geg	Geg	true	a	true
Dan	Geg	false	Ax(Geg,a)	true
Gevolg	Geg	false	ext(a,a)	true <i>applydef, Dan</i>

De eerste kolom geeft de namen der nieuw ingevoerde identifiers. De derde kolom geeft met true aan dat er een nieuw blok wordt geopend, anders false. In de tweede kolom staan de labels waar de omvattende blokken achtereenvolgens zijn geopend. De vierde kolom geeft de definitie van het nieuwe label in termen van oude. De PN's zijn primitieve begrippen die niet nader worden gedefinieerd; de EB's zijn emptyblockopeners: de bijbehorende labels zijn nieuw geïntroduceerde variabelen. Bij de regel "ond" wordt het blok geopend met een gemaakte onderstelling, nl. dat  $a = \text{true}$ . De vijfde kolom geeft de category van de nieuwe identifier aan. Een bijzondere rol speelt de category "ax"; de betreffende regel moet met "bool" worden bewezen maar mag met "true" worden gebruikt.

In de laatste kolom is alleen op de onderste regel een hint ingevuld; alle andere zijn emptyhint.

Ingebruikelijke taal luidt het bovenstaande stukje text:

1. Laat voor alle proposities u en v de propositie  $\text{ext}(u,v)$  gegeven zijn.
2. Zij a een gegeven propositie.
3. Axioma. Als a waar is dan geldt voor iedere propositie x dat ook  $\text{ext}(x,x)$  waar is.
4. Neem nu aan dat a waar is. Aangezien aan de onderstelling uit het axioma is voldaan, mag a voor x worden ingevuld, zodat ook  $\text{ext}(a,a)$  waar is.

Andere notatie voor relationele staten tabel

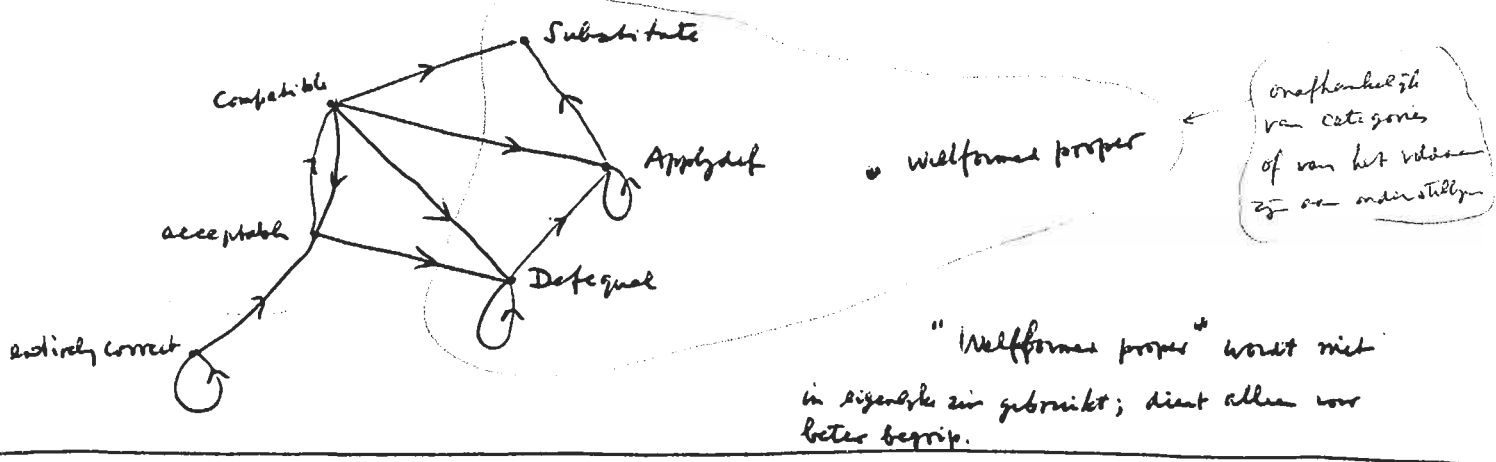
u :=	EB	bool
v :=	EB	bool
ext :=	PN	bool

a := PN bool

ord :=	a	true
x :=	EB	bool
Ax :=	ext(x, x)	ax

Geg :=	a	true	
Dan :=	Ax(Geg, a)	true	
Gevolg :=	ext(a, a)	true	applicatie van "Dan"

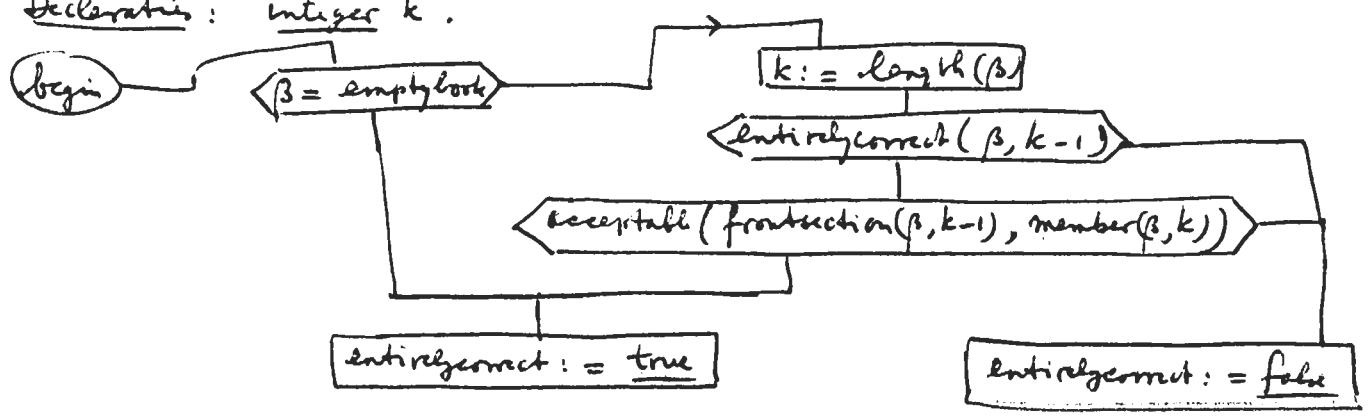
OVERZICHT der processen  $A \rightarrow B$  betekent A maakt gebruik van B.



**ENTIRELY CORRECT**

aanroep:  $entirely\ correct(\beta)$   
 ms  $\beta \in \langle \text{line string} \rangle$

Declaratie: integer k.



# WELLFORMED PROPER

Alleen voor intern gebruik

$\beta \in \langle \text{line string} \rangle$

$k \in \langle \text{integer} \rangle$

$\alpha \in \langle \text{proper expression} \rangle$

wellformed proper ( $\beta, k, \alpha$ )

Alleen aan roepen als

$$1 \leq k \leq \text{length}(\beta)$$

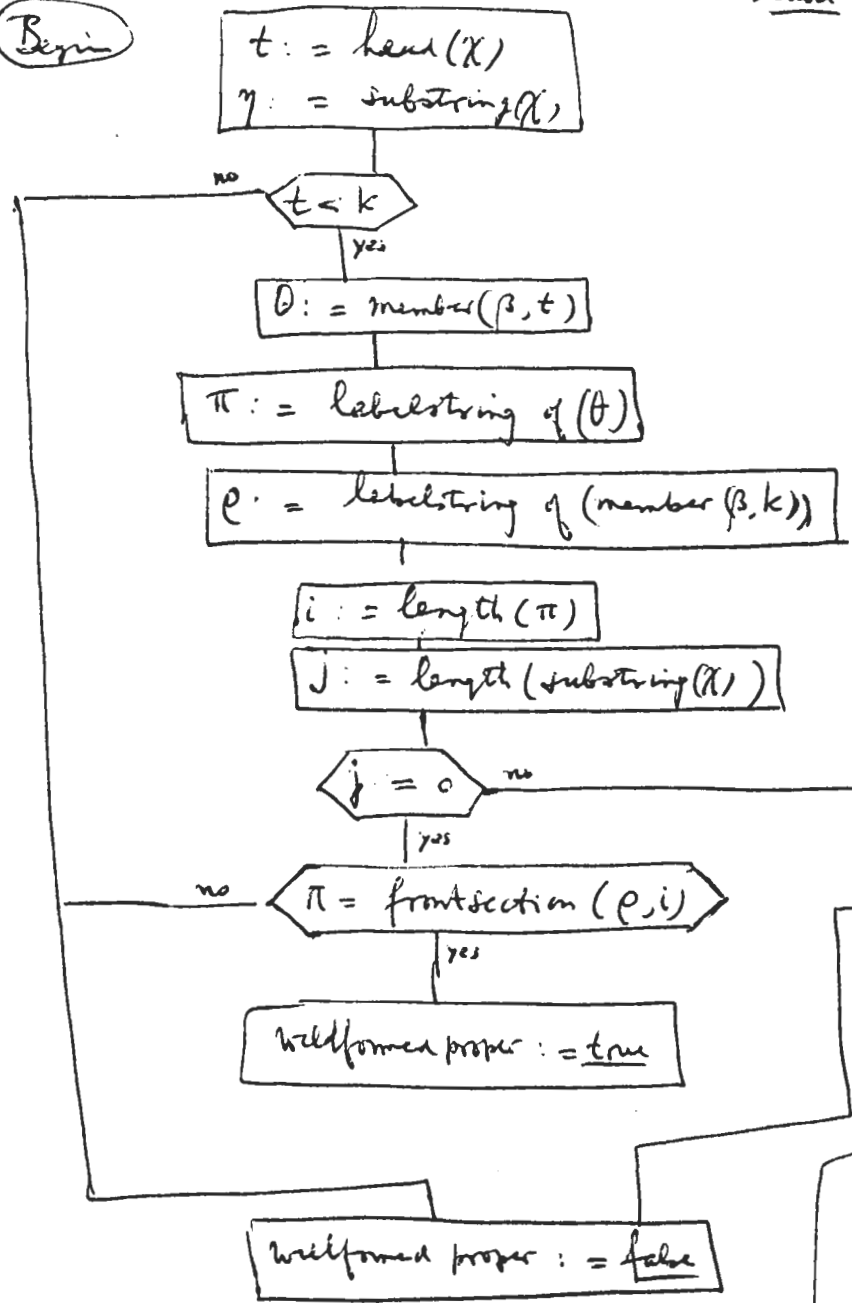
en entirely correct ( $\beta$ )

integer  $t, i, j, m$  proper expression string  $\gamma$

label string  $\rho, \pi$

Declarative line  $\theta$ ,

Begin



no

yes

begin

for  $m := 1$  step 1

until  $i$  do

if  $\rightarrow$  wellformed proper  
 $(\beta, k, \text{member}(\text{substring}(\alpha), \gamma), m)$

then begin wellformed proper := false ; goto out ;

wellformed proper := true

out :

end

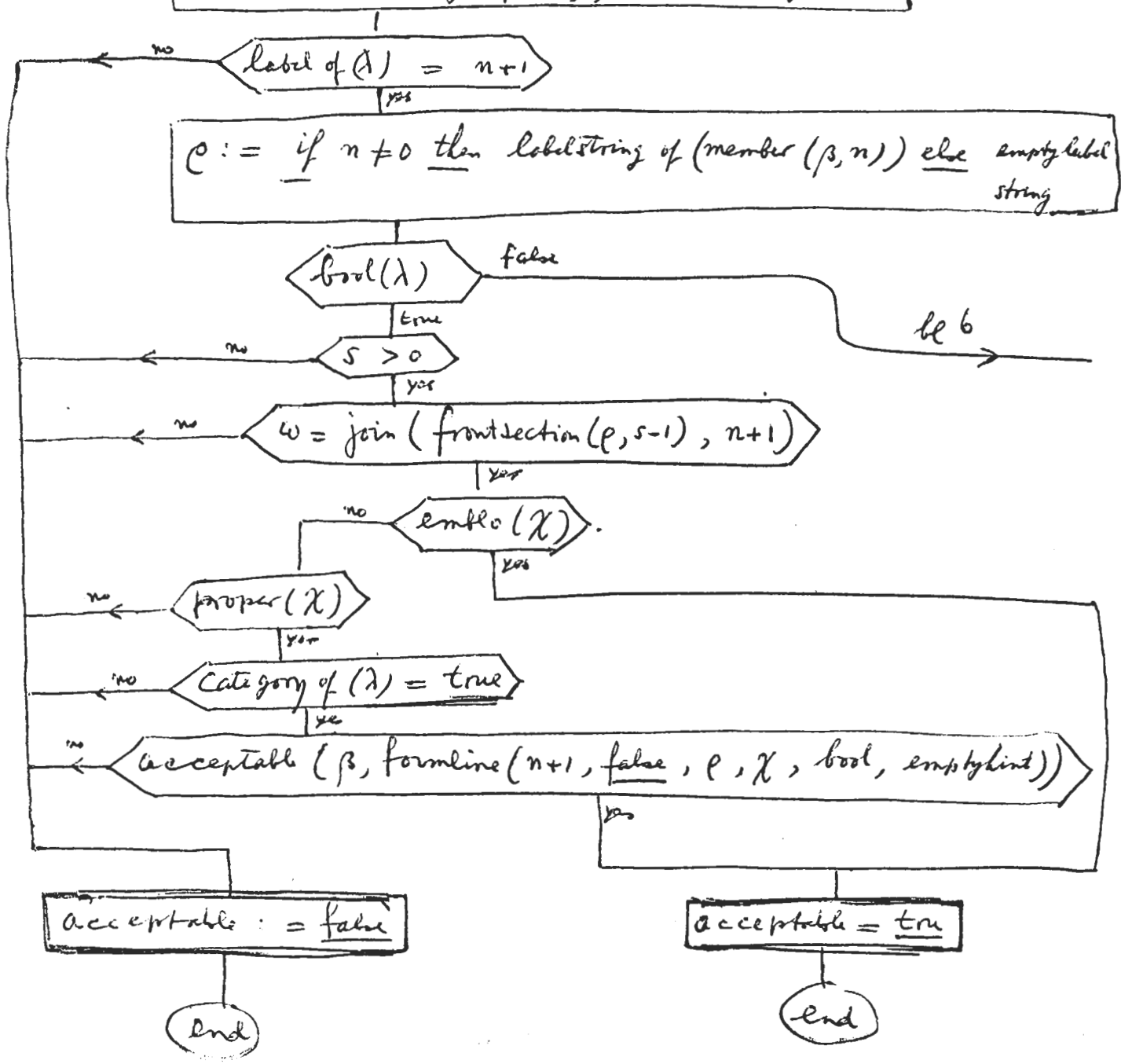
$\beta \in \langle \text{line string} \rangle$   
 $\lambda \in \langle \text{line} \rangle$

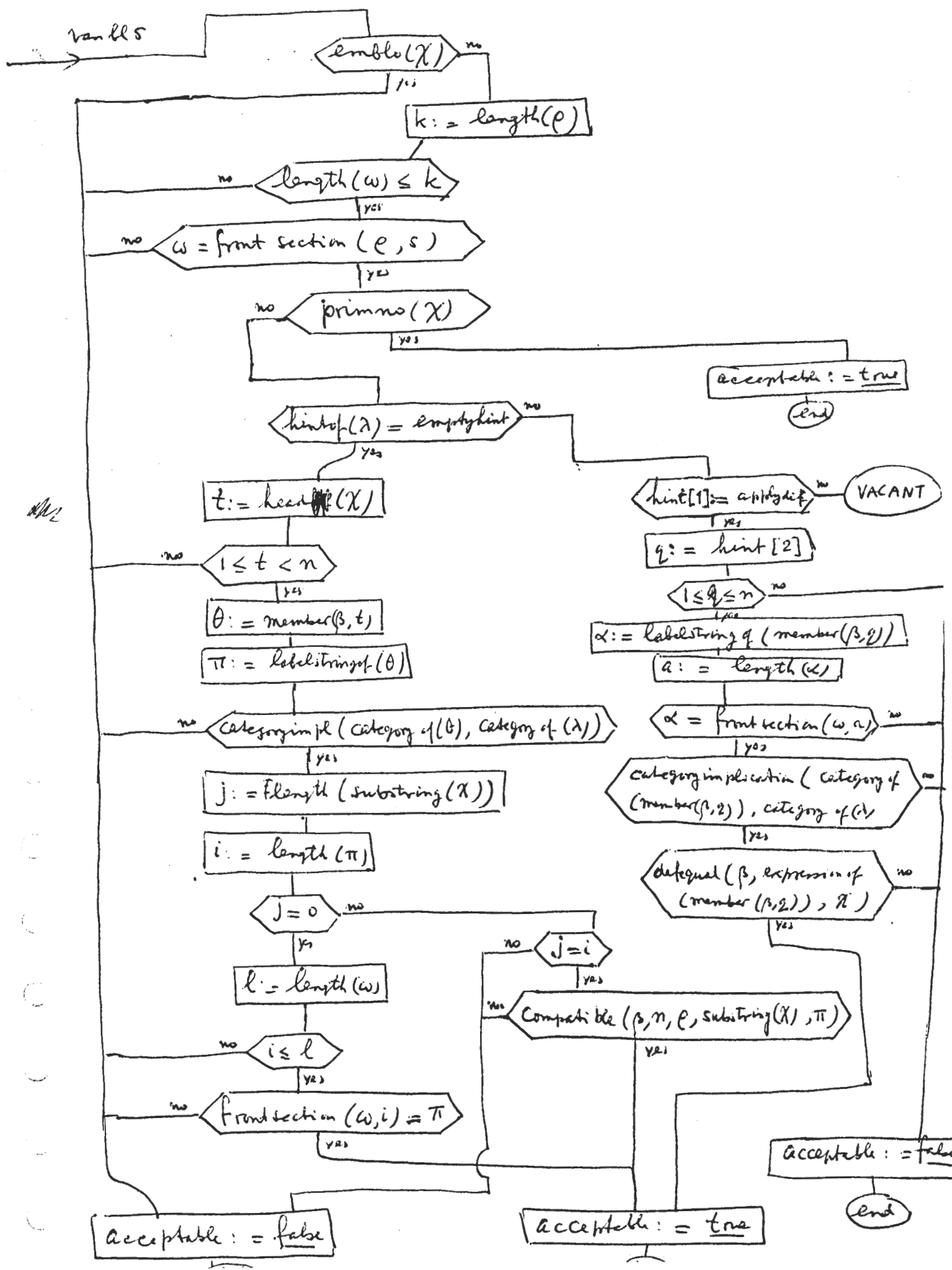
**ACCEPTABLE**

Definitie van **acceptable** ( $\beta, \lambda$ ). Mag aangeropen worden als *entirely correct* ( $\beta$ ) vaststaat.

Declaraties line  $\theta$   
label string  $\omega, \rho, \pi, \alpha$   
expression  $\chi$   
integer  $n, s, t, j, l, i, q, k, a$

**Begin**  $n := \text{length}(\beta) ; \chi := \text{expression of } (\lambda) ;$   
 $\omega := \text{label string of } (\lambda) ; s := \text{length}(\omega)$





**COMPATIBLE**

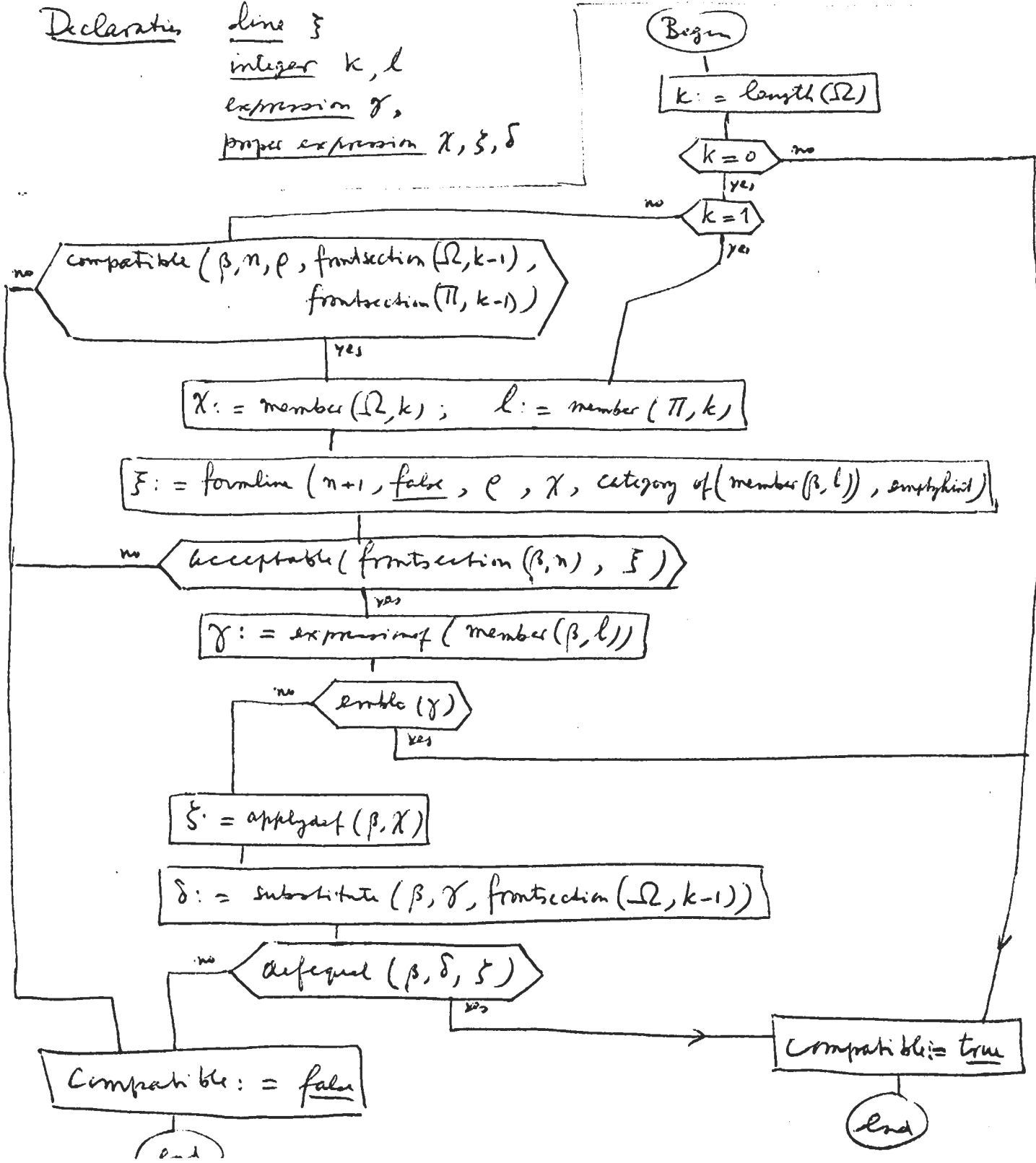
$\beta \in \langle \text{line string} \rangle$   
 $n \in \langle \text{integer} \rangle$   
 $\rho \in \langle \text{label string} \rangle$   
 $\Omega \in \langle \text{proper expression string} \rangle$   
 $\Pi \in \langle \text{label string} \rangle$

Mag angeschlossen werden als Teilstrich:

- 1) entirely correct ( $\beta$ )
- 2)  $1 \leq n \leq \text{length}(\beta)$
- 3)  $\Pi$  is labelstring von einer der regeln mit  $\beta$
- 4)  $\rho$  is labelstring von der  $n^{\text{te}}$  regel  
 $\text{length}(\Omega) = \text{length}(\Pi) \geq 0$

**Compatible** ( $\beta, n, \rho, \Omega, \Pi$ )

Declarative line  $\xi$   
integer  $k, l$   
expression  $\gamma$ ,  
proper expression  $\chi, \delta, \delta$





$\beta \in \langle \text{line string} \rangle$

$\lambda \in \langle \text{proper expression} \rangle$

$\mu \in \langle \text{proper expression string} \rangle$

**SUBSTITUTE**

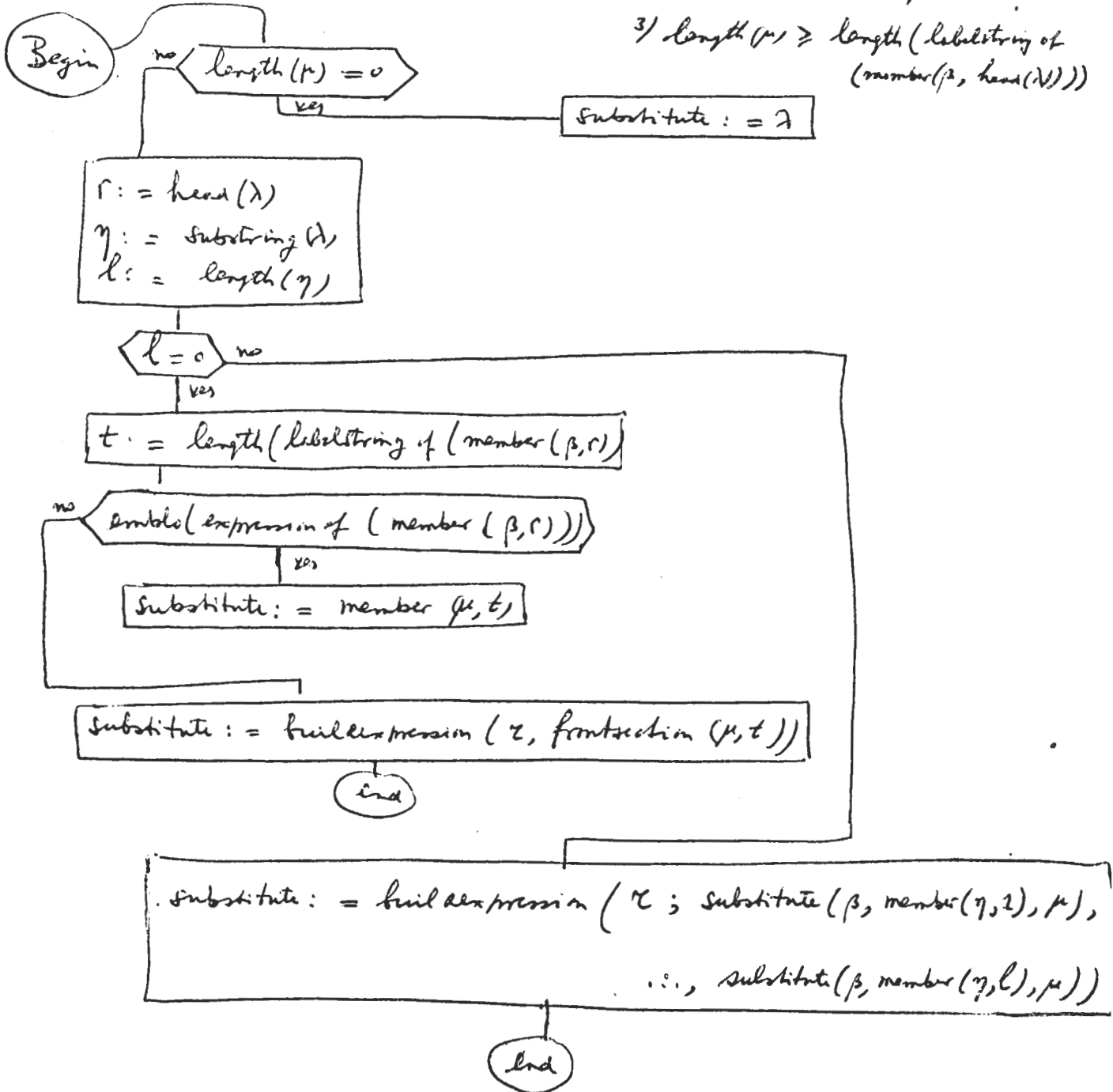
Definition  
Substitute ( $\beta, \lambda, \mu$ )

(value is con)  
May assign word also  
1) entirely correct ( $\beta$ ).

Declaratives  
proper expression string  $\eta$   
integer  $r, t$

2) wellformed proper ( $\beta, \text{length}(\beta), \lambda$ )

3)  $\text{length}(\mu) \geq \text{length}(\text{labelstring of } (\text{member}(\beta, \text{head}(\lambda))))$



# APPLY DEF

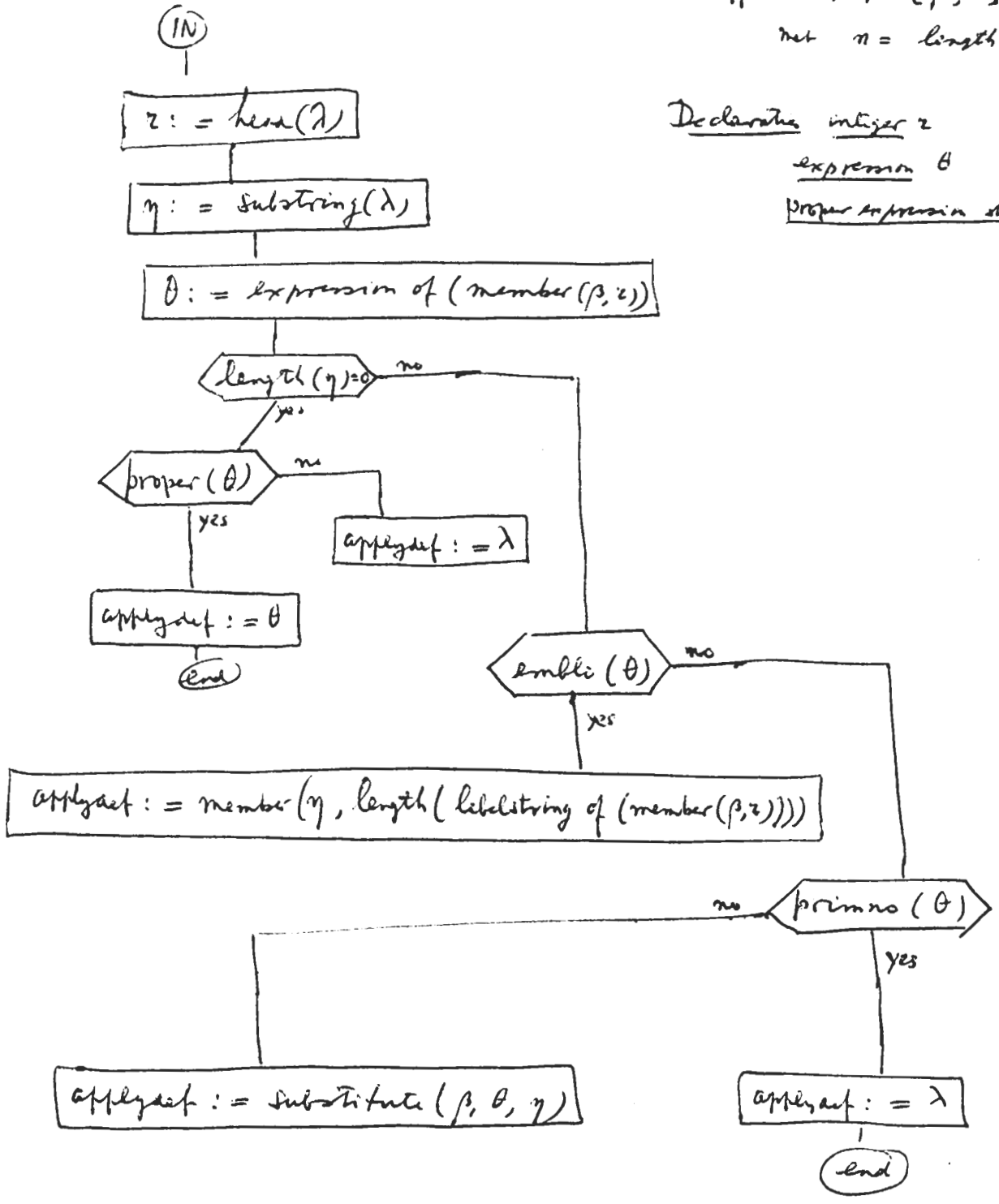
$\beta \in \langle \text{line string} \rangle$   
 $\lambda \in \langle \text{proper expression} \rangle$

applydef ( $\beta, \lambda$ )

May rearrange words as  
entirely correct( $\beta$ )

wellformed proper ( $\beta, n+1, \lambda$ )  
net  $n = \text{length}(\beta)$

Declarative integer  $z$   
expression  $\theta$   
proper expression string  $\eta$



# DEFEQUAL

$\beta \in \langle \text{line string} \rangle$

$\delta, \xi \in \langle \text{proper expression} \rangle$

defequal ( $\beta, \delta, \xi$ )

(IN) k := head( $\delta$ )  
l := head( $\xi$ )

$k > l$

$\theta := \text{apply}(\beta, \delta)$

defequal := defequal( $\theta, \xi$ )

$k \leq l$

$i := \text{length}(\text{substring}(\delta))$   
 $j := \text{length}(\text{substring}(\xi))$

$i = j$

$i = 0$

defequal := true

defequal := false

end

```

begin integer m;
for m := 1 step 1 until i do
  if → defequal ( member(substring( $\delta$ ), m),
                 member(substring( $\xi$ ), m)
  then begin defequal := false; goto out end;
  defequal := true
out:
  end
    
```

Mag kompakter werden als ist das hier

entirely correct ( $\beta$ )  
 well formed proper ( $\beta, n+1, \xi$ )  
 well formed proper ( $\beta, n, \xi$ )  
 max  $n = \text{length}(\beta)$

Declarative

proper expression  $\theta$

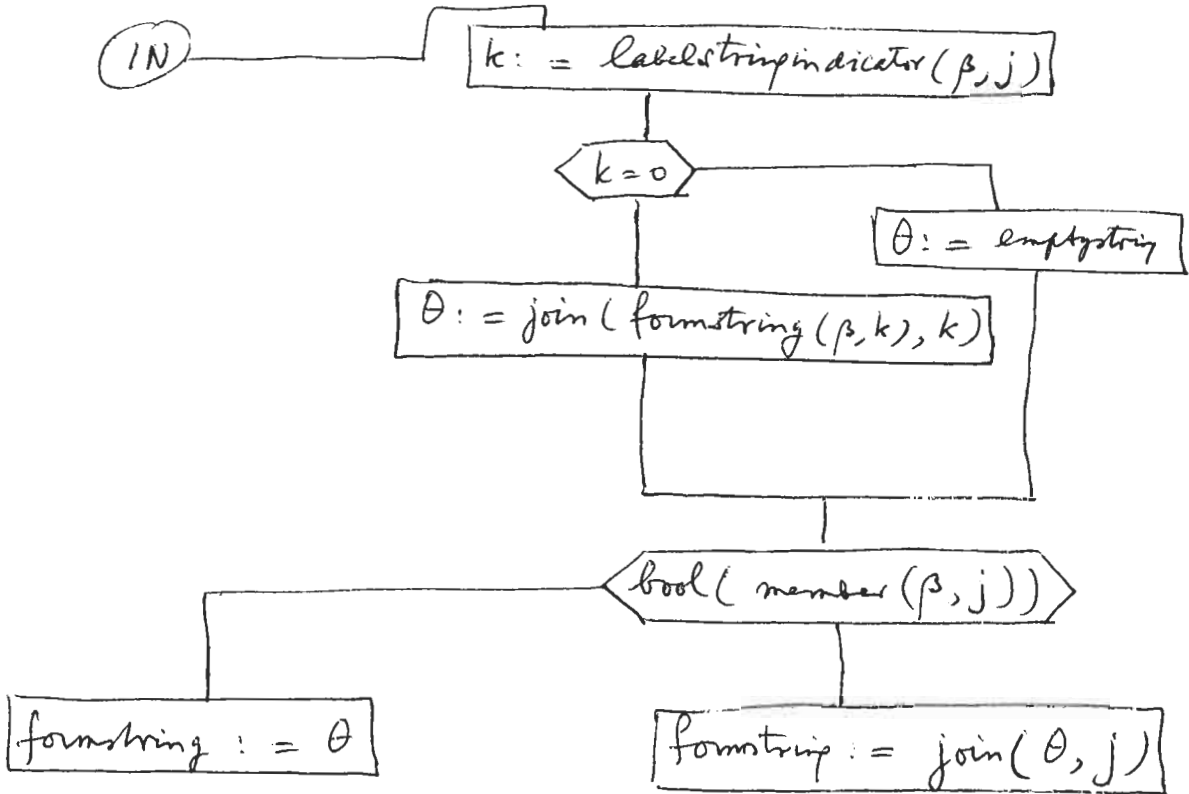
integer  $k, l, i, j, m$

$\beta$

formstrip ( $\beta, j$ )

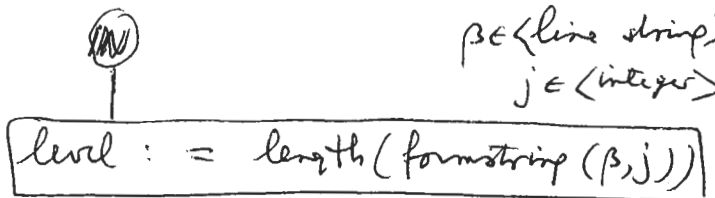
$\beta \in \langle \text{line string} \rangle$   
 $j \in \langle \text{integer} \rangle$

Declarative integer  $k$ , label string  $\theta$



level ( $\beta, j$ )

$\beta \in \langle \text{line string} \rangle$   
 $j \in \langle \text{integer} \rangle$



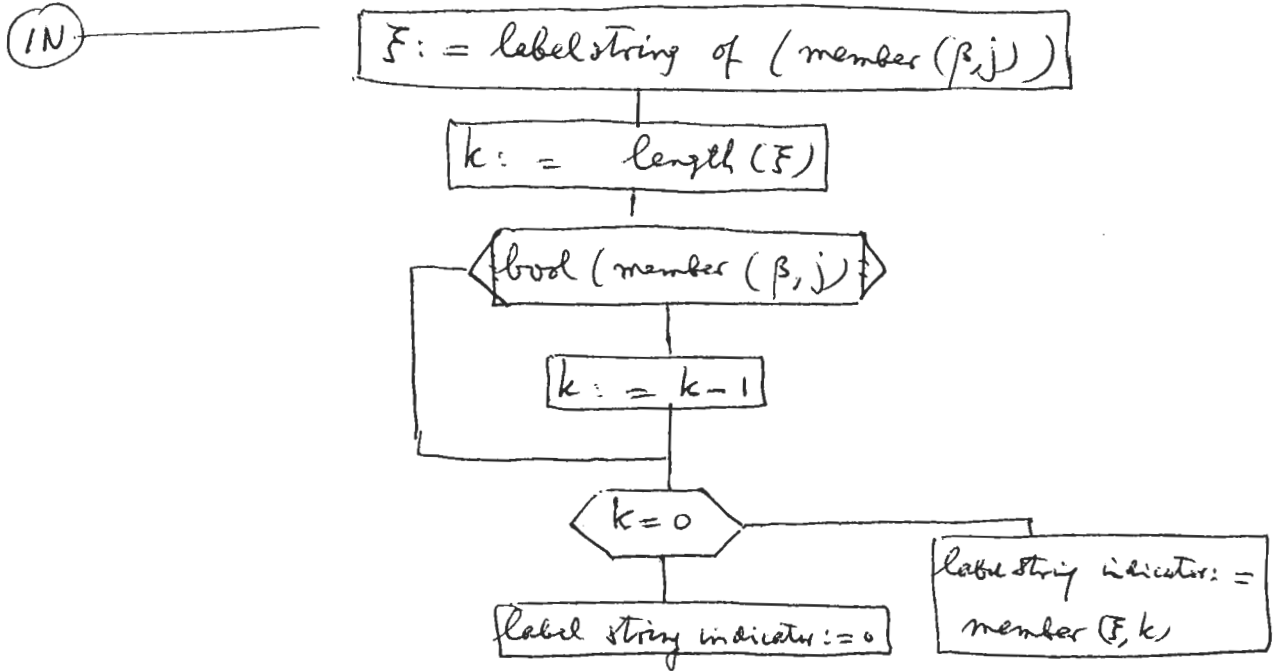
Opm: "formstrip" ~~level~~ ( $\beta, j$ ) level or  
 label string of ( $\text{member}(\beta, j)$ ), terminate at  
 $\beta$  entirely correct is.

labelstring indicator ( $\beta, j$ )

$\beta \in \langle \text{line string} \rangle$   
 $j \in \langle \text{integer} \rangle$

Understood:  $1 \leq j \leq \text{length}(\beta)$

Declarations ~~label~~ integer  $k$ ; labelstring  $F$



labelstring indicator of  $(\alpha)$  := labelstring indicator ( $\beta, \text{labelof}(\lambda)$ )