

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 μ nonempty
 λ als μ 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(line) 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

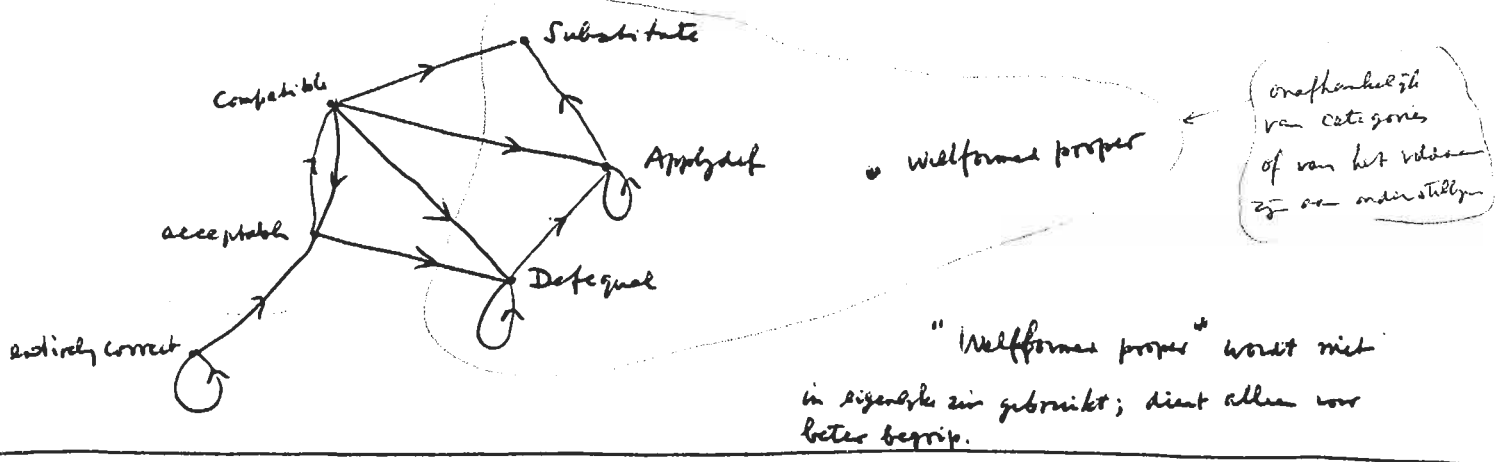
u :=	EB	bool
v :=	EB	bool
ext :=	PN	bool

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"

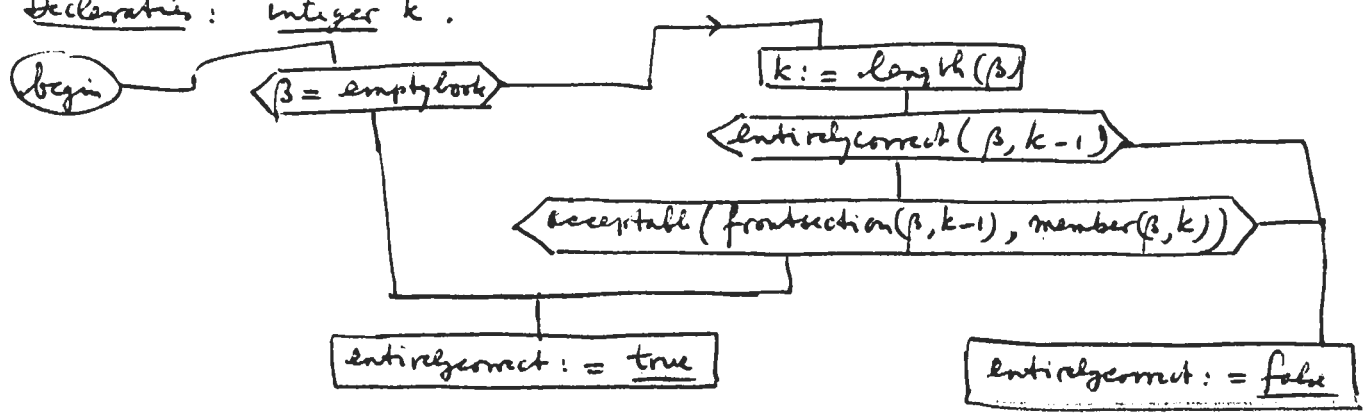
OVERZICHT der processen $A \rightarrow B$ betekent A maakt gebruik van B.



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 (β, k, α)

Alleen aan roepen als

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

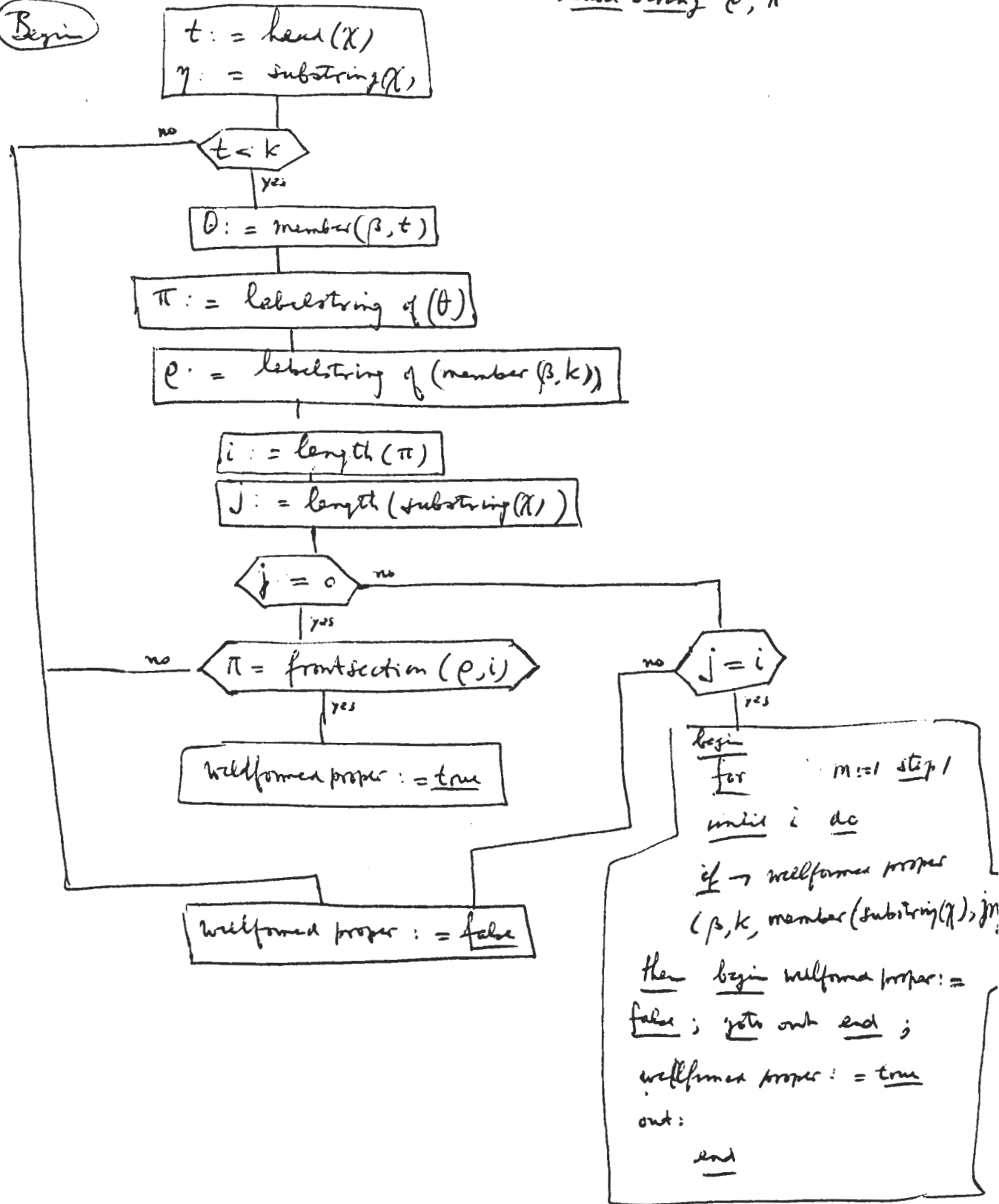
en entirely correct (β)

integer t, i, j, m proper expression string γ

label string ρ, π

Declarative line θ ,

Begin



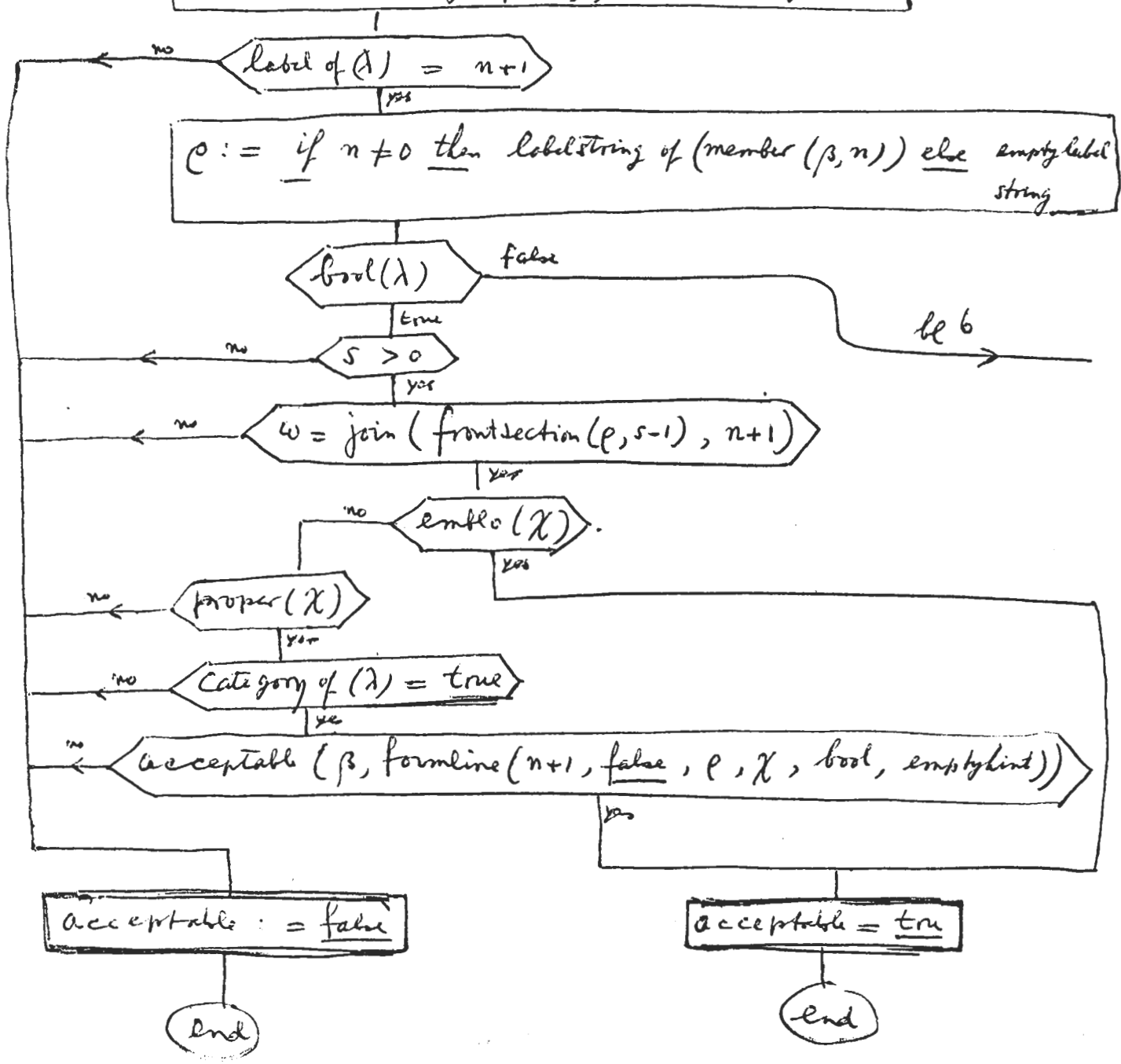
$\beta \in \langle \text{line string} \rangle$
 $\lambda \in \langle \text{line} \rangle$

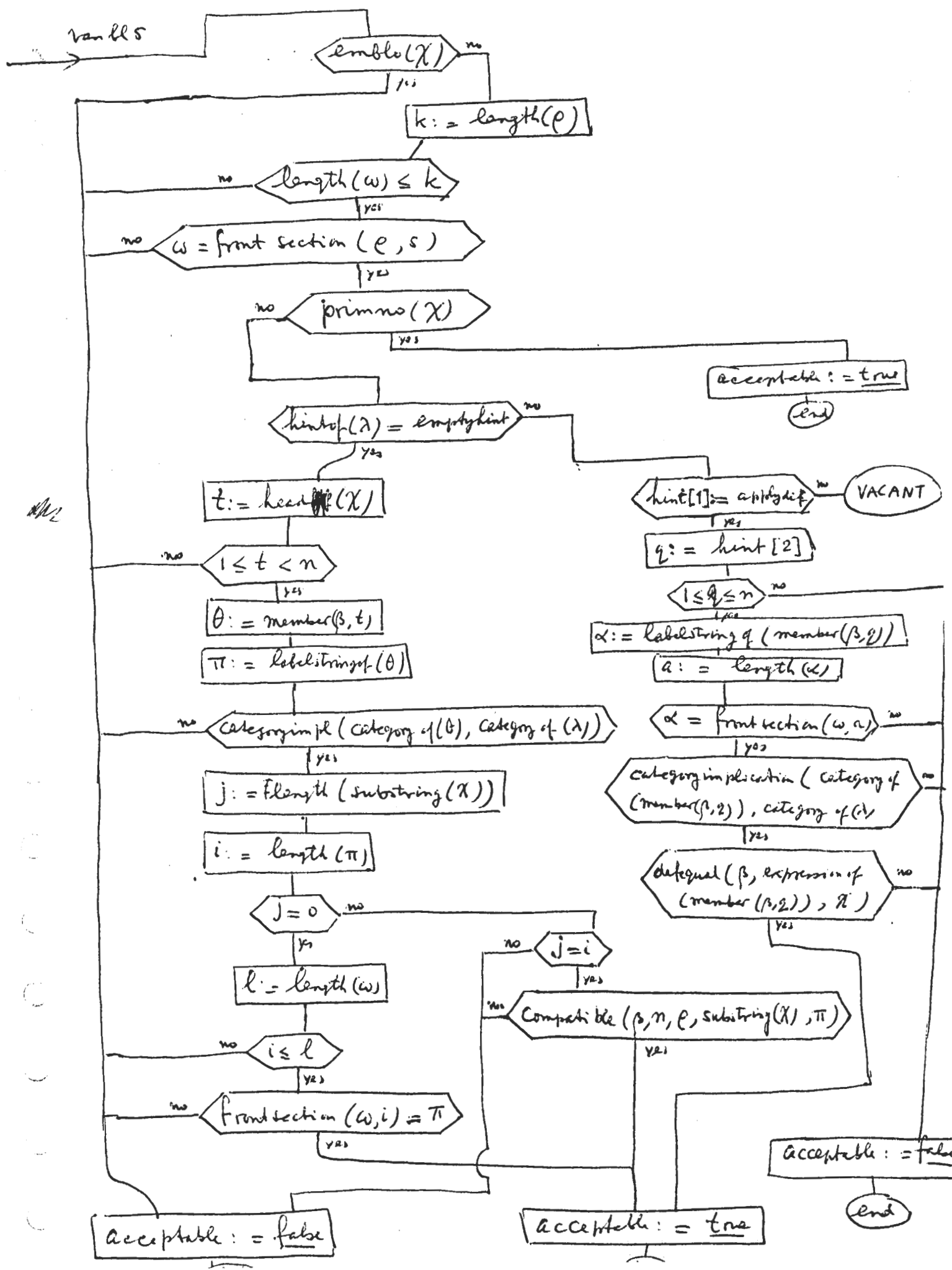
ACCEPTABLE

Definitie van **acceptable** (β, λ). Mag aangeropen worden als *entirely correct* (β) vaststaat.

Declaraties line θ
label string $\omega, \rho, \pi, \alpha$
expression χ
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

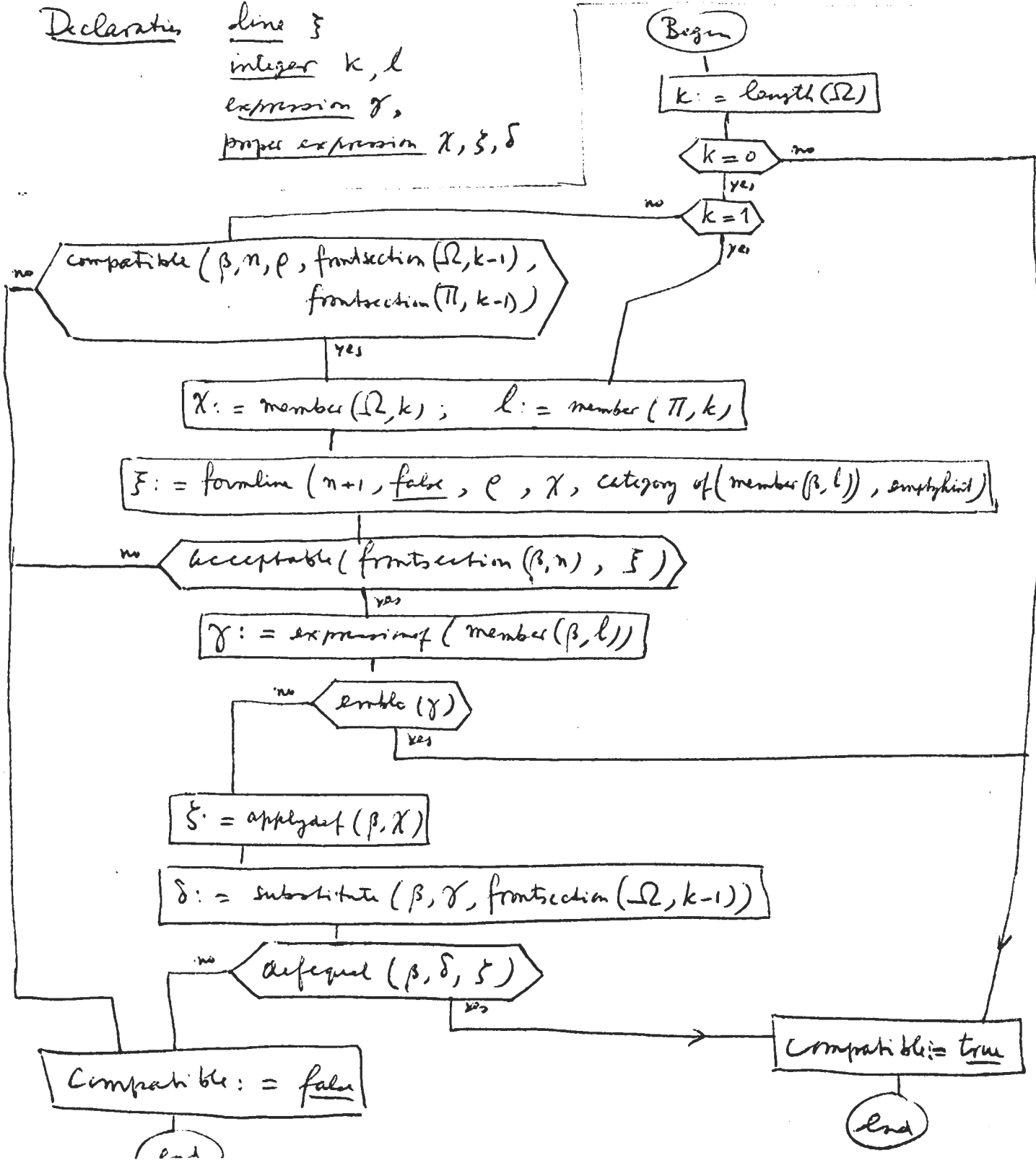
$\beta \in \langle \text{line string} \rangle$
 $n \in \langle \text{integer} \rangle$
 $\rho \in \langle \text{label string} \rangle$
 $\Omega \in \langle \text{proper expression string} \rangle$
 $\Pi \in \langle \text{label string} \rangle$

Mag angeschlossen werden als Teilstrich:

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

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

Declarative line ξ
integer k, l
expression γ ,
proper expression χ, δ, δ



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

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

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

SUBSTITUTE

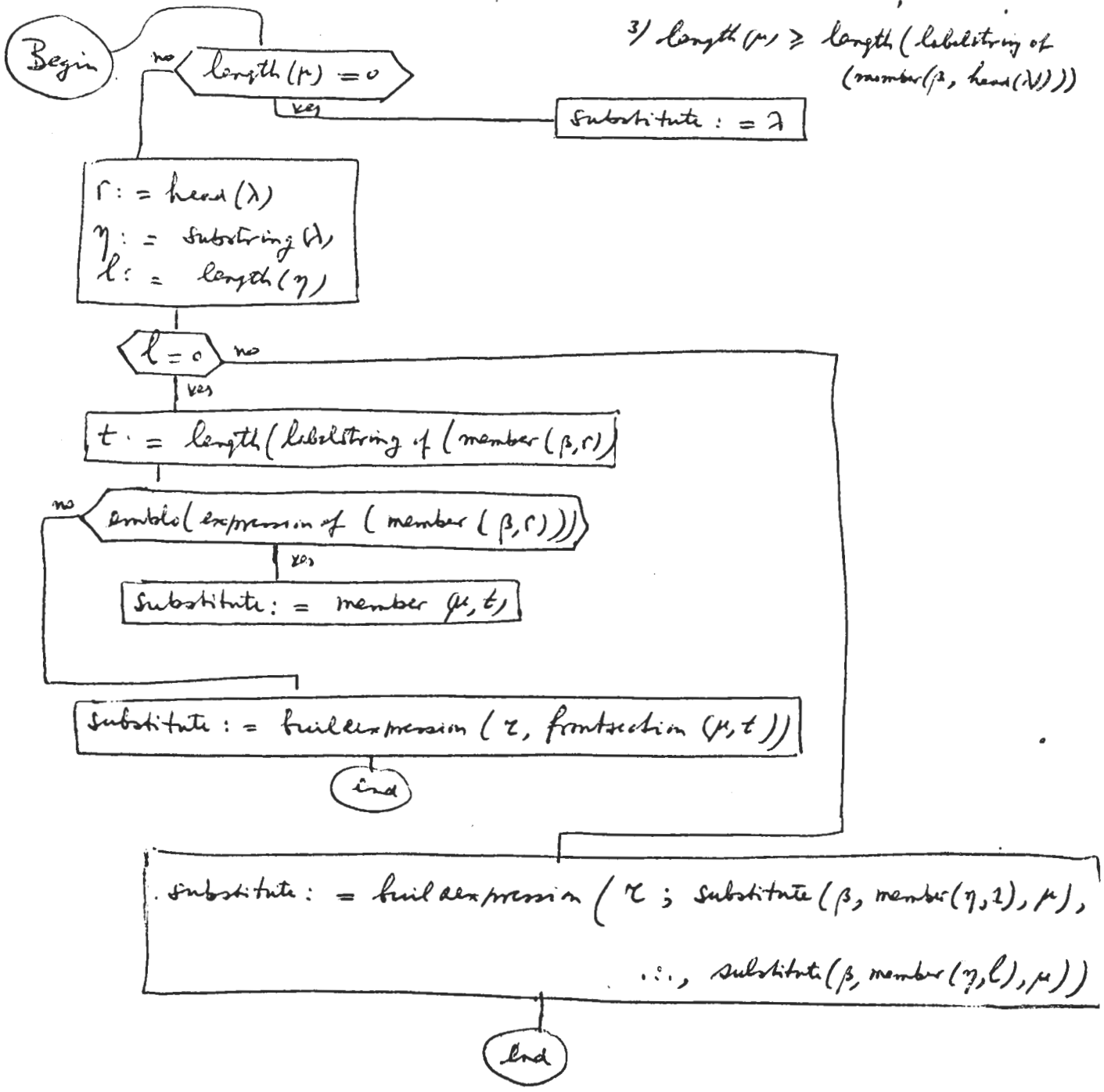
Definition
 substitute (β, λ, μ)

(value is con)
 May assignieren werden ab
 1) entirely correct (β).

Declaratives proper expression string η
 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

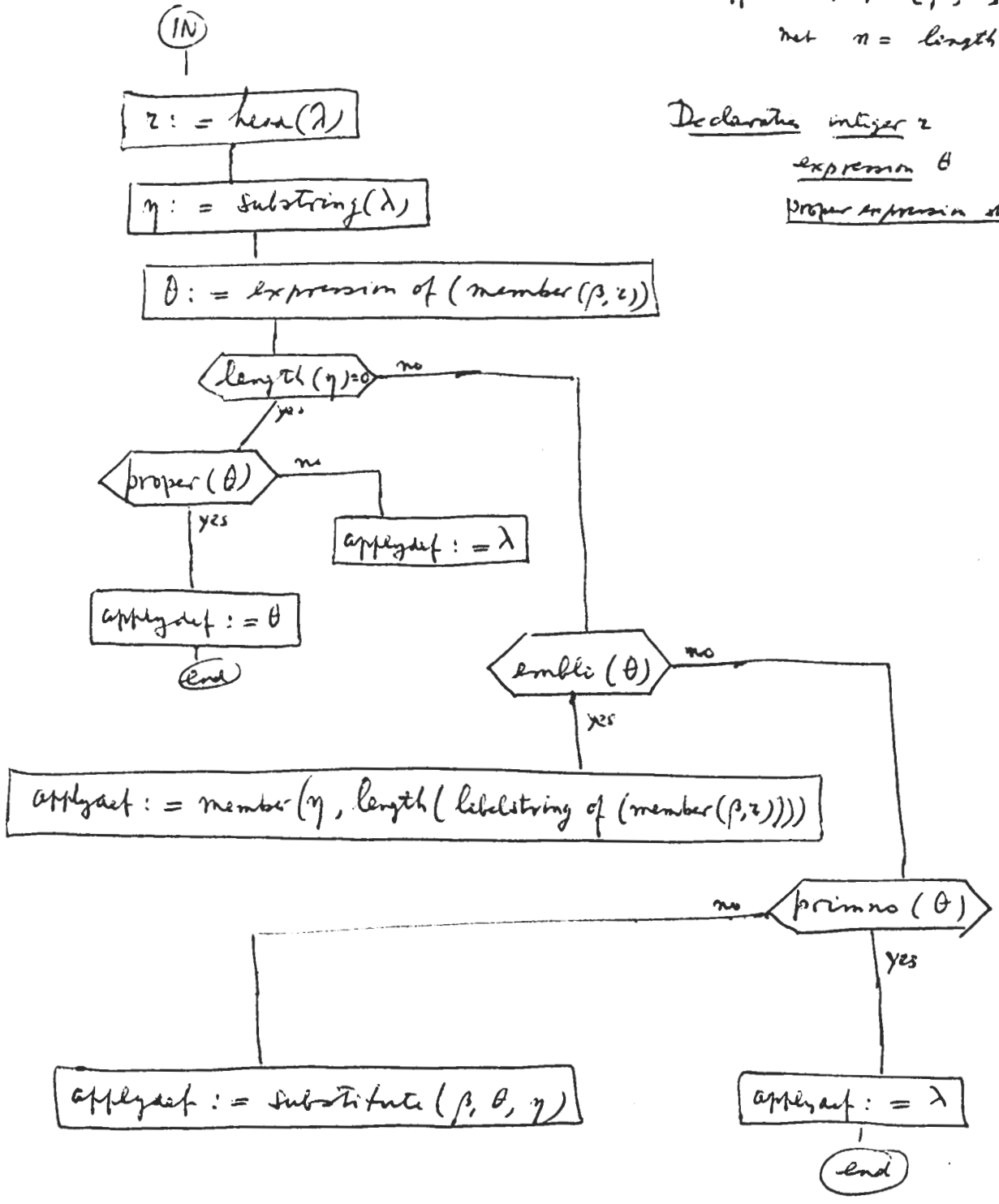
$\beta \in \langle \text{line string} \rangle$
 $\lambda \in \langle \text{proper expression} \rangle$

applydef (β, λ)

May rearrange words as
entirely correct(β)

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

Declarative integer z
expression θ
proper expression string η



DEFEQUAL

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

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

defequal (β, δ, ξ)

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

$k > l$

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

defequal := defequal (θ, ξ)

$k \leq l$

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

$i = j$

$i = 0$

defequal := true

defequal := false

end

Mag kompilierbar werden als ist das kein

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

Declarative

proper expression θ

integer k, l, i, j, m

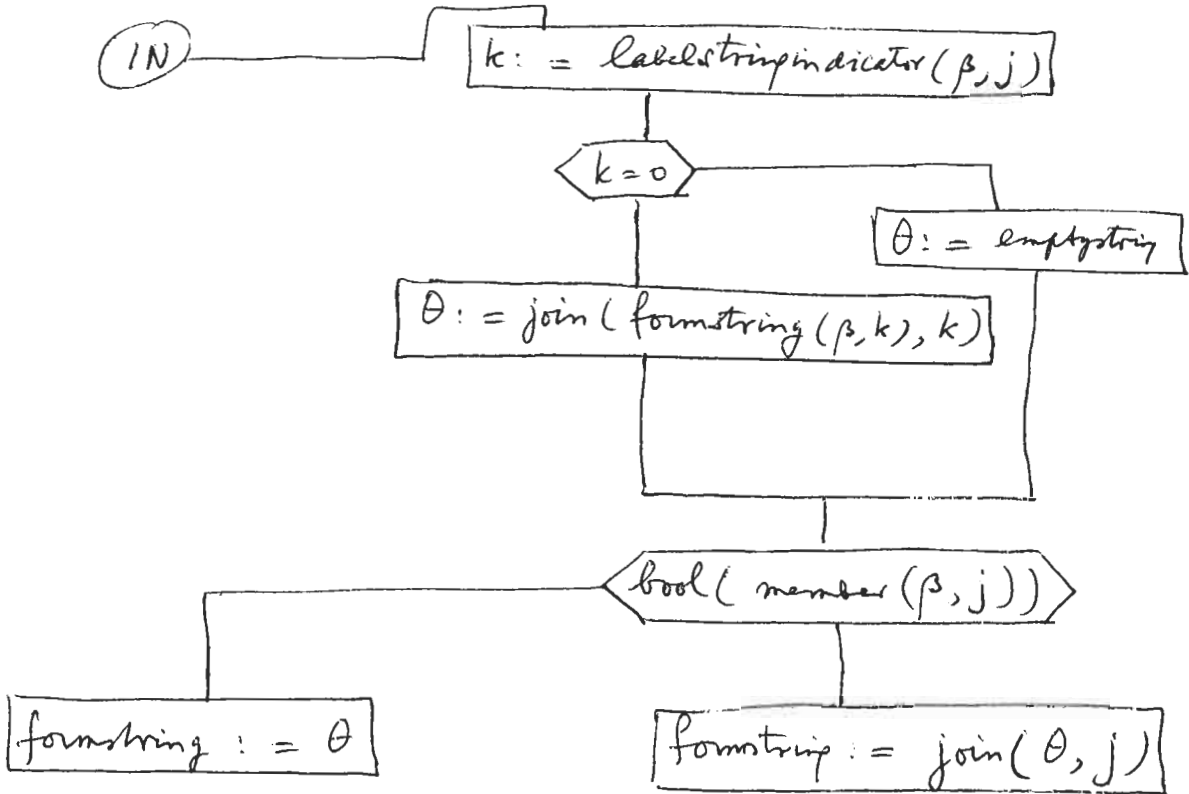
```
begin integer m ;
for m := 1 step 1 until i do
  if  $\rightarrow$  defequal ( member(substring( $\delta$ ), m),
  member(substring( $\xi$ ), m)
  then begin defequal := false ; goto out end ;
defequal := true
out :
end
```

β

formstrip (β, j)

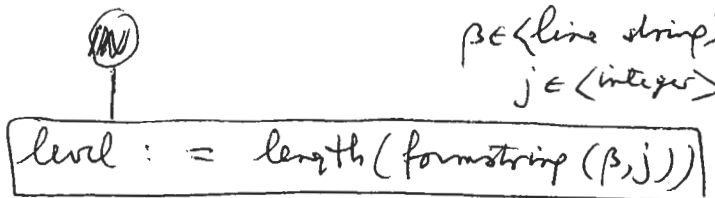
$\beta \in \langle \text{line string} \rangle$
 $j \in \langle \text{integer} \rangle$

Declarative integer k , label string θ



level (β, j)

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



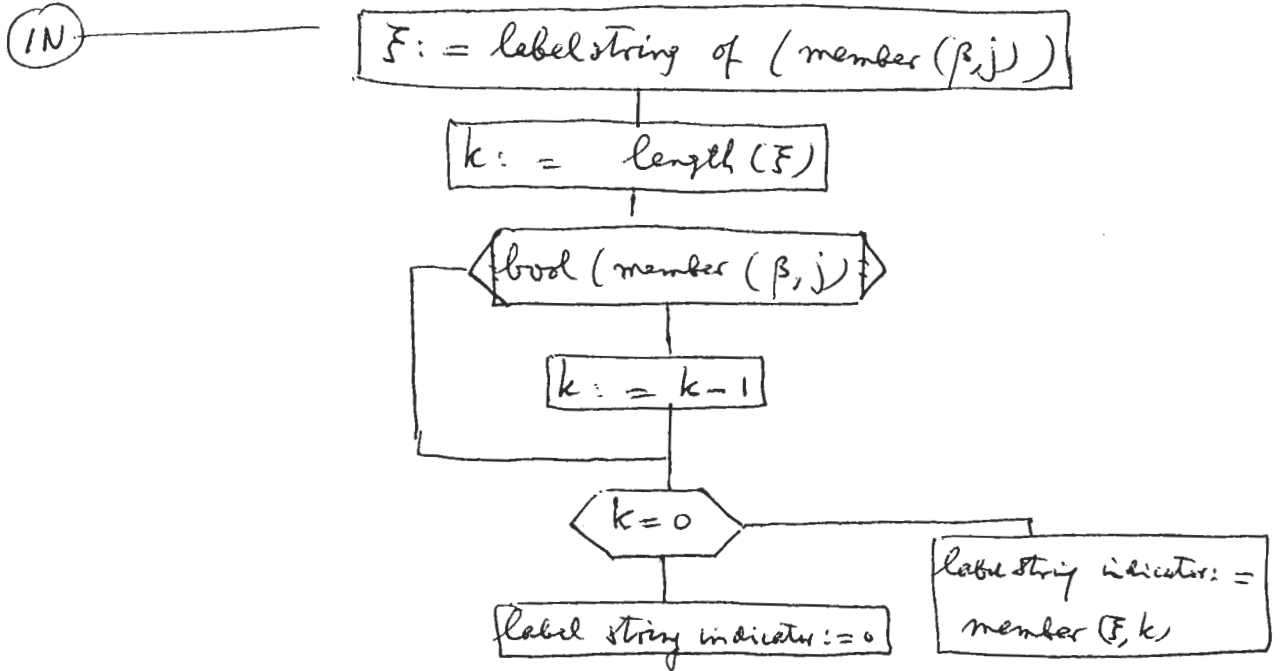
Opn: "formstrip" ~~level~~ (β, j) level or label string of ($\text{member}(\beta, j)$), terminate at β entirely correct is.

labelstring indicator (β, 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 (λ) := labelstring indicator ($\beta, \text{labelof}(\lambda)$)