Technological University,
Eindhoven, the Netherlands.

Example of a text written in AUTOMATH.

by N.G. de Bruijn.

1. BOOLEANS

1.1	0	bool	:=	PN	<u>sort</u>
1.2	0	x	:=	- - - - - -	bool
1.3	x	TRUE	:=	PN	<u>sort</u>
1.4	0	CONTR	:=	[v bool] TRUE(v)	<u>sort</u>
1.5	0	a	:=	- - - - -	CONTR
1.6	a	b	:=	- - - - -	bool
1.7	b	then 1	:=	{b} a	TRUE(b)
1.8	0	ksi	:=	- - - - -	<u>sort</u>
1.9	ksi	nonempty	:=	PN	bool
1.10	ksi	a	:=	- - - - -	ksi
1.11	a	then 2	:=	PN	TRUE(nonempty)
1.12	ksi	a	:=	- - - - -	TRUE(nonempty)
1.13	a	then 3	:=	PN	ksi
1.14	ksi	EMPTY	:=	[u ksi] CONTR	<u>sort</u>
1.15	ksi	x	:=	- - - - -	ksi
1.16	x	u	:=	- - - - -	EMPTY(ksi)
1.17	u	then 4	:=	{x} u	CONTR
1.18	x	then 5	:=	[t EMPTY(ksi)] then 4 (t)	EMPTY(EMPTY(ksi)
1.19	ksi	PARADISE II	:=	[t EMPTY(EMPTY(ksi))] ksi	<u>sort</u>

2. EQUALITY

		ksi	:=	- - - - -	sort
		x	:=	- - - - -	ksi
2.1	x	y	:=	- - - - -	ksi
2.2	y	IS	:=	PN	sort
2.3	y	equal	:=	nonempty(IS)	bool
2.4	x	reflexive	:=	PN	IS(x,x)
2.5	y	ass 1	:=	- - - - -	IS(x,y)
2.6	ass 1	symm	:=	PN	IS(y,x)
2.7	ass 1	z	:=	- - - - -	ksi
2.8	z	ass 2	:=	- - - - -	IS(y,z)
2.9	ass 2	transitive	:=	PN	IS(x,z)
2.10	ksi	theta	:=	- - - - -	sort
2.11	theta	x 1	:=	- - - - -	ksi
2.12	x 1	P 1	:=	- - - - -	[t ksi]theta
2.13	P 1	x 2	:=	- - - - -	ksi
2.14	x 2	ass 3	:=	- - - - -	IS(ksi, x1, x2)
2.15	ass 3	then 6	:=	PN	IS(theta,{x1} P1, {x2} P1)
2.16	P 1	P2	:=	- - - - -	[x ksi] theta
2.17	P 1	ass 4	:=	- - - - -	IS([x ksi] theta, P1, P2)
2.18	ass 4	then 7	:=	PN	[x ksi]IS(theta,{x1}P1,{x} P2)
2.19	P 1	ass 4a	:=	- - - - -	[x ksi] IS(theta,{x} P1,{x} P2
2.20	ass 4a	then 7a	:=	PN	IS([x ksi] theta, P1, P2)

2ª. IFELSE

2ª.1		(ksi	:=	-----	sort)
2ª.2		x	:=	-----	ksi)
2ª.3	x	a	:=	-----	bool
2ª.4	a	ifelse	:=	PN	ksi
2ª.5	a	ass 4b	:=	-----	TRUE(a)
2ª.6	ass 4b	then 7b	:=	PN	IS(ksi,ifelse,x)
2ª.7	a	ass 4c	:=	-----	IS(ksi,ifelse, x)
2ª.8	ass 4c	then 7c	:=	PN	TRUE(a)

2ª. Equality for two sorts.

2ᵇ.1		(ksi	:=	-----	sort)
2ᵇ.2	ksi	eta	:=	-----	sort
2ᵇ.3	eta	a	:=	-----	ksi
2ᵇ.4	a	b	:=	-----	eta
2ᵇ.5	b	ISS	:=	PN	sort
2ᵇ.6	b	equall	:=	nonempty(ISS)	bool
2ᵇ.7	b	ass 4d	:=	-----	ISS(ksi,eta,a,b)
2ᵇ.8	ass 4d	symmm	:=	PN	ISS(eta,ksi,b,a)
2ᵇ.9	eta	zeta	:=	-----	sort
2ᵇ.10	zeta	a	:=	-----	ksi
2ᵇ.11	a	b	:=	-----	eta
2ᵇ.12	b	c	:=	-----	zeta
2ᵇ.13	c	ass 4e	:=	-----	ISS(ksi,eta,a,b)
2ᵇ.14	ass 4e	ass 4f	:=	-----	ISS(eta,zeta,b,c)
2ᵇ.15	ass 4f	transitivv	:=	PN	ISS(ksi,zeta,a,c)

(comment: The PN's in 2.2, 2.6, 2.9 can now be replaced respectively by ISS(ksi,ksi,x,y), symmm(ksi,ksi,x,y); transitivv(ksi,ksi,ksi,x,y,z).

2*. EMBEDDING.

```
2c.1             ((ksi        :=  -----                               sort )
2c 2               |eta       :=  -----                               sort )

2c.3   eta          |p        :=  -----                               [x,eta]ksi
2c.4   p            |EMBED    :=  [t,eta] ISS(ksi,eta,{t}p,t)         sort
2c.5   p             |w       :=  -----                               EMBED
2c.6   w              |image  :=  [x,ksi] nonempty(EXISTS(eta,[b,eta]equal({b}p,x))
                                                                      [x,ksi] bool
```

3. P A I R S O R T.

		ksi	:=	- - - - -	sort
		theta	:=	- - - - -	sort
3.1	theta	pairsort	:=	PN	sort
3.2	theta	x	:=	- - - - -	ksi
3.3	x	y	:=	- - - - -	theta
3.4	y	pair	:=	PN	pairsort
3.5	theta	u	:=	- - - - -	pairsort
3.6	u	first	:=	PN	ksi
3.7	u	second	:=	PN	theta
3.8	u	then 8	:=	PN	IS(pairsort,u,pair(first,second
3.9	y	then 9	:=	PN	IS(ksi,x,first(pair))
3.10	y	then 10	:=	then 9 (theta,ksi,y,x)	IS(theta,y,second(pair))

4. B O O L E Q U A L, I M P L I C A T I O N.

4.1	0	a	:=	- - - - -	bool
4.2	a	b	:=	- - - - -	bool
4.3	b	c	:=	- - - - -	pairsort([t TRUE(a)] TRUE(b), [s TRUE(b)] TRUE(a))
4.4	c	then 11	:=	PN	IS(bool,a,b)
4.5	b	d	:=	- - - - -	IS(bool,a,b)
4.6	d	then 12	:=	PN	pairsort([t TRUE(a)] TRUE(b), [s TRUE(b)] TRUE(a))
4.7	b	IMPL	:=	[u TRUE(a)] TRUE(b)	sort
4.8	b	ass 1	:=	- - - - -	TRUE(a)
4.9	ass 1	ass 2	:=	- - - - -	IMPL
4.10	ass 2	modpon	:=	{ass 1} ass 2	TRUE(b)

5. SOME LOGICAL CONSTANTS.

5.1	0	contradiction	:=	nonempty(CONTR)	bool
5.2	0	OBVIOUSLY	:=	IMPL(contradiction,contradiction)	<u>sort</u>
5.3	0	trivial	:=	nonempty(OBVIOUSLY)	bool
5.4	0	now 1	:=	[u TRUE(contradiction)] u	OBVIOUSLY
5.5	0	now 2	:=	then 2 (OBVIOUSLY, now 1)	TRUE(trivial)
5.6	0	now 3	:=	[u CONTR] u	EMPTY(CONTR)

6. N O N, A N D.

6.1	0	b	:=	- - - - -	bool
6.2	b	NON	:=	EMPTY(TRUE(b))	<u>sort</u>
6.3	b	non	:=	nonempty (NON)	bool
6.4	b	c	:=	- - - - -	bool
6.5	c	AND	:=	pairsort(TRUE(b),TRUE(c))	<u>sort</u>
6.6	c	and	:=	nonempty(AND)	bool
6.7	c	if	:=	- - - - -	AND
6.8	if	then 12 u	:=	first(TRUE(b),TRUE(c),if)	TRUE(b)
6.9	if	then 12 b	:=	second(TRUE(b),TRUE(c),if)	TRUE(c)
6.10	b	if	:=	- - - - -	TRUE(non(b))
6.11	if	then 12 c	:=	then 3(NON(b),if)	NON(b)
6.12	b	if	:=	- - - - -	NON(b)
6.13	if	then 12 d	:=	then 2(NON(b),if)	TRUE(non(b))
6.14	b	if	:=	- - - - -	TRUE(b)
6.15	if	then 12 c	:=	then 5(TRUE(b),if)	EMPTY(NON(b))

7. E X I S T S, A L L.

		(ksi	:=	- - - - -	sort)
7.1	ksi	P	:=	- - - - -	[u ksi] bool
7.2	P	EXISTS	:=	PN	sort
7.3	P	v	:=	- - - - -	ksi
7.4	v	ass 1	:=	- - - - -	TRUE({v} P)
7.5	ass 1	then 13	:=	PN	EXISTS
7.6	P	ass 2	:=	- - - - -	EXISTS
7.7	ass 2	then 13 a	:=	PN	ksi
7.8	ass 2	then 13 b	:=	PN	TRUE({then 13 a
7.9	P	ALL	:=	[u ksi] TRUE({u} P)	sort
7.10	P	(v	:=	- - - - -	ksi)
7.11	v	ass 3	:=	- - - - -	ALL
7.12	ass 3	specialize	:=	{v} ass 3	TRUE({v}P)
7.13	P	NONEXIST	:=	[u ksi] EMPTY({u} P)	sort
7.14	P	WEAKEXIST	:=	EMPTY(NONEXIST)	sort
7.15	ksi	PARADISE I	:=	[Q,[u ksi] bool][t WEAKEXIST(Q)] EXISTS(Q)	sort
7.16	P	a	:=	- - - - -	PARADISE I
7.17	a	b	:=	- - - - -	WEAKEXIST(P)
7.18	b	then 14	:=	{b}{P} a	EXIST(P)

8. CONSTANT PREDICATE

		ksi	:=	- - - - -	sort
		theta	:=	- - - - -	sort
8.1	theta	pi	:=	- - - - -	[t ksi] theta
8.2	pi	CONSTANT	:=	[s ksi][t ksi] IS(theta, {t}pi, {s} pi)	sort
8.3	g	a	:=	- - - - -	ksi
8.4	a	b	:=	- - - - -	ksi
8.5	b	c	:=	- - - - -	CONSTANT
8.6	c	then 15	:=	{a} {b} c	(IS, theta, {a} pi, {b} pi)

9. CONDITIONAL BRACING.

		(ksi	:=	- - - - -	sort)
9.1	ksi	P	:=	- - - - -	[t ksi] bool
9.2	P	h	:=	- - - - -	[t ksi][s TRUE({t} P)] bool
9.3	h	x	:=	- - - - -	ksi
9.4	x	sigma	:=	TRUE({x}P)	sort
9.5	x	then 16	:=	EXISTS(sigma, {x}h)	sort
9.6	h	Q	:=	[t ksi] nonempty(then 16(t))	[t ksi] bool
9.7	x	a	:=	- - - - -	TRUE({x} Q)
9.8	a	then 17	:=	then 3(then 16, a)	then 16
.9	a	then 18	:=	then 13a (sigma, {x}h, then 17)	TRUE({x} P)
.10	a	then 19	:=	then 13b (sigma, {x}h, then 17)	TRUE({then 18}{x} h)
.11	x	a	:=	- - - - -	TRUE({x}P)
.12	a	b	:=	- - - - -	TRUE({a}{x} h)
.13	b	then 20	:=	then 13(TRUE({x}P), {x}h, a, b)	then 16
.14	b	then 21	:=	then 2(then 16, then 20)	TRUE({x} Q)
.15	b	then 22	:=	then 2(then 16, then 13 (sigma, {x}h, a, b))	TRUE({x} Q)

10. D I R E C T   B R A C I N G.

		(ksi	:=	- - - - -	sort)
		(P	:=	- - - - -	[t ksi] bool)
10.1	P	Q	:=	- - - - -	[t ksi] bool
10.2	Q	R	:=	[t ksi] and ({t}P,{t}Q)	[t ksi] bool

11. N A M E C H A N G I N G .

11.1	0	NAME	:=	PN	sort
11.2	0	dash	:=	PN	NAME
11.3	0	ksi	:=	- - - - -	sort
11.4	ksi	classin	:=	pairsort([t ksi]bool, NAME)	sort
11.5	ksi	c	:=	- - - - -	classin
11.6	c	predicof	:=	first([t ksi] bool,NAME,c)	[t ksi] bool
11.7	ksi	d	:=	- - - - -	[t ksi] bool
11.8	d	classof	:=	pair([t ksi]bool,NAME,d,dash)	classin(ksi)

11.9	0	NAME 2	:=	PN	sort
11.10		dot	:=	PN	NAME 2
11.11	0	(ksi	:=	- - - - -	sort)
11.12	ksi	PREDICATE	:=	pairsort([t ksi]bool,NAME 2)	sort
11.13	ksi	c	:=	- - - - -	PREDICATE
11.14	c	predicup	:=	first([t ksi] bool,NAME,c)	[t ksi] bool
11.15	ksi	d	:=	- - - - -	[t ksi] bool
11.16	d	predicdown	:=	pair([t ksi]bool,NAME 2,d,dot)	PREDIACTE

11.17		(ksi	:=	- - - - -	sort)
11.18		(theta	:=	- - - - -	sort)
11.19	theta	P	:=	- - - - -	[t ksi] theta
11.20	P	a	:=	- - - - -	EMPTY(theta)
11.21	a	then 25	:=	[s ksi] {{s }P}a	EMPTY(ksi)
11.22	ksi	b	:=	- - - - -	EMPTY(ksi)
11.23	b	then 26	:=	then 25(TRUE(nonempty(ksi)), ksi, [s TRUE(nonempty(ksi))], then 3(s),b)	EMPTY(TRUE(nonempty(ks
11.24	ksi	c	:=	- - - - -	EMPTY(TRUE(nonempty(ks
11.25	c	then 27	:=	then 25(ksi,TRUE(nonempty(ksi)), [s ksi] then 2(s),c)	EMPTY(ksi)
11.26	0	b	:=	- - - - -	bool
11.27	b	x	:=	- - - - -	TRUE(non(non(b))
11.28	x	then 28	:=	then 3(EMPTY(TRUE(non(b)),x)	EMPTY(TRUE(non(b))
11.29	x	then 29	:=	then 27(NON(b),x)	EMPTY(NON(b))
11.30	x	then 30	:=	then 29	EMPTY(EMPTY(TRUE(b)))
11.31	b	y	:=	- - - - -	EMPTY(EMPTY(TRUE(b))
11.32	y	then 31	:=	y	EMPTY(NON(b))
11.33	y	then 32	:=	then 26(NON(b), then 31)	NON(non(b))
11.34	y	then 33	:=	then 2(NON(non(b))	TRUE(non(non(b))
11.35	b	z	:=	- - - - -	TRUE(b)
11.36	z	then 34	:=	then 5(TRUE(b),z)	EMPTY(EMPTY(TRUE(b))
11.37	z	then 35	:=	then 33(TRUE(b),then 34)	TRUE(non(non(b))

12. EXCLUDED THIRD.

12.1	0	EXCLTHIRD	:=	[t bool] PARADISE II(TRUE(t))	<u>sort</u>
12.2	0	excl	:=	- - - - -	EXCLTHIRD
12.3	excl	a	:=	- - - - -	bool
12.4	b	if	:=	- - - - -	EMPTY(NON(a))
12.5	if	then 36	:=	{a} {if} excl	TRUE(a)
12.6	0	(b	:=	- - - - -	bool)
12.7	b	VALID	:=	[s EXCLTHIRD]TRUE(b)	<u>sort</u>
12.8	b	if	:=	- - - - -	TRUE(b)
12.9	if	then 37	:=	[s EXCLTHIRD] if	VALID(b)

(comment: VALID is the notion of truth in non-intuitionistic logic)

12.10	b	p	:=	- - - - -	VALID(b)
12.11	p	q	:=	- - - - -	EXCLTHIRD
12.13	q	then 38	:=	{q}p	TRUE(b)
12.14	q	then 39	:=	then 35(then 38)	TRUE(non(non(b))
12.15	p	then 40	:=	[s EXCLTHIRD] then 39(s)	VALID(non(non(b))
12.16	b	p	:=	- - - - -	VALID(non(non(b))
12.17	p	q	:=	- - - - -	EXCLTHIRD
12.18	q	then 40	:=	{q}p	TRUE(non(non(b))
12.19	q	then 41	:=	then 30(then 40)	EMPTY(NON(b))
12.20	q	then 42	:=	then 36(q,b,then 41)	TRUE(b)
12.21	p	then 42 a	:=	[s EXCLTHIRD] then 42(s)	VALID(b)

13. S E T S.

13.1		(ksi	:=	- - - - -	sort)
13.2	ksi	set	:=	[x ksi] bool	sort
13.3	ksi	x	:=	- - - - -	ksi
13.4	x	s	:=	- - - - -	set
13.5	s	ESTI	:=	TRUE({x}s)	sort
13.6	s	esti	:=	{x}s	bool
13.7	ksi	s	:=	- - - - -	set
13.8	s	t	:=	- - - - -	set
13.9	t	INCL	:=	ALL(ksi, [u ksi] nonempty (IMPL ({u}s, {u}t)))	sort
13.10	t	incl	:=	nonempty(INCL)	bool
13.11	ksi	(s	:=	- - - - -	set)
13.12	s	powerset	:=	[x set] incl(x,s)	set(set(ksi))
13.13	ksi	universe	:=	[x ksi]trivial	set
13.14	ksi	x	:=	- - - - -	ksi
13.15	x	then 43	:=	now 2	ESTI(x,universe)
13.16	ksi	emptyset	:=	[x ksi] contradiction	set
13.17	ksi	x	:=	- - - - -	ksi
13.18	x	ass	:=	- - - - -	ESTI(x,emptyset)
13.19	ass	then 44	:=	assume	TRUE(contradiction)
13.20	ass	then 45	:=	then 3(CONTR,then 44)	CONTR
13.21	x	then 46	:=	[t ESTI(x,emptyset)] then 45	EMPTY(ESTI(x,emptyset
13.22	x	then 47	:=	then 46	NON(esti(x,emptyset))

14. TRANSITIVITY OF SET-INCLUSION.

14.1	ksi	(s	:=	- - - - -	set
14.2	s	t	:=	- - - - -	set
14.3	t	r	:=	- - - - -	set
14.4	r	ass 1	:=	- - - - -	INCL(s,t)
14.5	ass 1	ass 2	:=	- - - - -	INCL(t,r)
14.6	ass 2	x	:=	- - - - -	ksi
14.7	x	ass 3	:=	- - - - -	TRUE({x}s)
14.8	ass 3	then 48	:=	then 3(IMPL({x}s, {x}t), {x} ass 1)	IMPL({x}s, {x}t)
14.9	ass 3	then49	:=	{ass 3} then 48	TRUE({x} t)
14.10	ass 3	then 50	:=	then 49(t,r,r,ass 2, ass 2, x, then 49)	TRUE({x}r)
14.11	x	then 51	:=	[p TRUE({x}s)]then 50(p)	IMPL({x}s, {x}r)
14.12	x	then 52	:=	then 2(IMPL({x}r, {x}t), then 51)	TRUE(nonempty(IMPL({ {x}r)))
14.13	ass 2	then 53	:=	[t ksi] then 52(t)	INCL(s,r)

15. INCLUSION INDUCED IN POWERSET.

15.1	ksi	(s	:=	- - - - -	set)
15.2	s	, (t	:=	- - - - -	set)
15.3	t	ass 4	:=	- - - - -	INCL(s,t)
15.4	ass 4	u	:=	- - - - -	set
15.5	u	a	:=	- - - - -	TRUE({u} powerset(s))
15.6	a	then 54	:=	a	TRUE(incl(u,s))
15.7	a	then 55	:=	then 3(INCL(u,s),then 54)	INCL(u,s)
15.8	a	then 56	:=	then 53(u,s,t,then 55,ass 4)	INCL(u,t)
15.9	a	then 57	:=	then 2(INCL(u,t),then 56)	TRUE(incl(u,t))
15.10	a	then 58	:=	then 57	TRUE({u} powerset (t)
15.11	u	then 59	:=	[a TRUE({u} powerset(s))] then 58	IMPL({u} powerset(s), {u} powerset (t))
15.12	u	then 60	:=	IMPL({u} powerset(s),{u}powerset(t)	sort
15.13	u	then 61	:=	then 2(then 60, then 59)	TRUE(nonempty(then 60
15.14	ass 4	then 62	:=	[t set] then 61(t)	INCL(set,powerset(s), powerset(t))