C 2－3. 数学の証明を行なうプログラム

西村敏男，伊大知紀子　（東京教育大応用数理）

目 的　このプログラムは，記号論理化した数学的概念に対する思考活動過程を機械化することを試みたものである．

述語論理の体系としては，G. GentzenのLKをK. Schütte が修正したものを基礎におき，それに，定義，cutなどの推論規則を加えて，計算機による処理能力を増加させた．

ある種の前提条件（こゝではDefinition, Theorem と称する）をもつた命題（Problemと称する）を入力して，その証明を出力として得るものである．

Theoems の組合せによつて，Lemmasを作成登録する機能も有する．

使用機種　HIPAC103基礎装置及び高速テープセン孔機

memory layout

番　地	語数	内　　容
64~127	64	時間測定用補修プログラムなど
128~511	384	分解
512~767	256	check
768~873	106	drop the rank
874~1087	214	cut
1088~1407	320	output
(注) 1408~2175	768	input
1408~2431	1024	sequent pool
2432~2879	448	prime formula pool（最高223prime formulasまで）
2880~3423	544	formulas key pool（最高543 formulasまで）
3424~3455	32	1 sequent working space（最高27 formulasまで）
3456~3520	64	definition pool（最高31definitionsまで）
3584~8191	4480	theorem pool

(注)　証明実行中，core1408~2431 番地の内容はdrum2048~2815 番地に待避し，点線以降のように各種プールが作られる．

§1. 論理の体系と証明の実行

この体系の証明論的根拠は数学基礎論にゆずるとして，こゝでは，小型計算機と苦闘する我々のalgorithmについて解説する．

1. 使用される記号

入力に使用される文字及び記号は，印字面において次のとおりである．

free variable	a	b	c	d	e	f	g	h
計16種	a_{10}	b_{10}	c_{10}	d_{10}	e_{10}	f_{10}	g_{10}	h_{10}
found variable	x	y	z	u	v	w	s	t (r)
計 8種								
alah variable	i	j	k	l	m	n	o	p	q
計18種	i_{10}	j_{10}	k_{10}	l_{10}	m_{10}	n_{10}	o_{10}	p_{10}	q_{10}
predicate	A	B	C			W	X	Y	Z
計26種								
logical symbol	&	V	→	≡	[ ]	[E]	—		
計 7 種								
auxiliary symbol	( )	,	:	and etc.				

2. formulaの書き方

2－1　prime formula は，predicateのあとにfree variable を並べたものである．

(例) " a＝b "を Iab と表わす．

＊　free variable は8個以内でなければならない．

1－2　$\mathfrak{A}$, $\mathfrak{B}$ が formula であるとき，(　)及び logical symbolを用いて

$(\mathfrak{A}\ \&\ \mathfrak{B})$　$(\mathfrak{A}\ \mathrm{V}\ \mathfrak{B})$　$(\mathfrak{A} \rightarrow \mathfrak{B})$　$(\mathfrak{A} \equiv \mathfrak{B})$　$-\mathfrak{A}$

そして　あるいは　ならば　と同等　でない　と書く．

(例) " a≠b "を－Iab と表わす．

1－3　$\mathfrak{A}(a)$ が formula であるとき，a をすべてbound variableの一つ(例えば x)で置き換えて得られる $\mathfrak{A}(x)$ に logical symbol〔x〕または〔Ex〕を付して，〔x〕$\mathfrak{A}(x)$ または〔Ex〕$\mathfrak{A}(x)$ と書く．

すべての　ある

(例) " すべてのx，yについてx＝yが成立する "ことを〔x〕〔y〕Ixy と表わす．

注　特に－〔x〕$\mathfrak{A}(x)$, 〔Ex〕$\mathfrak{A}(x)$ なる形の formulaを"alah formula "と仮称する．

alah variable は入力の際には使用出来ない．

＊　formula の nesting numberは30以内でなければならない．

注　prime formula 及び，否定のついたprime formulaを総称して，primary formulaと仮称する．

3. problemの規定

Problem. $\mathcal{A}_1, \mathcal{A}_2, \cdots\cdots, \mathcal{A}_n$:

problem は，27個以内の formula $\mathcal{A}_i$ から成るsequentとして表わし，一問題に必ず一個だけ含まれる．例えば，"前提 $\Sigma_1, \Sigma_2 \cdots\cdots, \Sigma_l$ のいずいずれもが成立つときに，$\Gamma_1, \Gamma_2, \cdots\cdots, \Gamma_m$ のうちの少なくとも一つが結論として引出されるという問題は，

$$-\Sigma_1, \cdots\cdots, \Sigma_l, \Gamma_1, \cdots\cdots, \Gamma_m$$

というsequentとして表わす．

sequent中の formulaは原則として左のものから（alah formulaは後廻しとして）推論規則を当てはめて分解し，すべての formulaが primary formula（及びalah formula）となつた所（uppermost sequent）で logical axiomまたは既成の theorem が適用出来るかどうかを調べる．

4. dafinitionの規定

Definition 1. $P_{1x_1\cdots x_{m_1}} \equiv \mathcal{A}_1(x_1, \cdots, x_{m_1})$:

$\vdots$ $\vdots$

Definition n. $P_{nx_1\cdots x_{m_n}} \equiv \mathcal{A}_n(x_1, \cdots, x_{m_n})$:

definitionは上のような形式をもつ．

$[x_1][x_2]\cdots[x_m](P_{x_1x_2\cdots x_m} \equiv \mathcal{A}(x_1, \cdots, x_m))$ という特殊な形の前提を扱うもので，推論規則の一つとして，uppermost sequentにおいて prime formula $P_{b_1b_2\cdots b_m}$ を formula $\mathcal{A}(b_1, \cdots, b_m)$ によつて置換える．definiton は，theorem中の prime formulaに対しては無効である．

＊ definitionは0～31個与えることができ，後方のものから順に使用される．

従つて，$i<j$ のとき predicate P_j は，formula $\mathcal{A}_i$ に含まれてはいけない．

5. theorem の規定

Theorem 1. $[x_1]\cdots[x_{l_1}][E_z][y_1]\cdots[y_{m_1}](P_{11}V\cdots VP_{1k_1})$:

$\vdots$ $\vdots$

Theorem n. $[x_1]\cdots[x_{l_m}][E_z][y_1]\cdots[y_{m_n}](P_{n_1}V\cdots VP_{nk_n})$:

theorem は上のような形式をもつ．

$[x_1][x_2]\cdots[x_l][E_z][y_1]\cdots[y_m](P_1V\cdots VP_k)$ という特殊な形の前提を扱うも

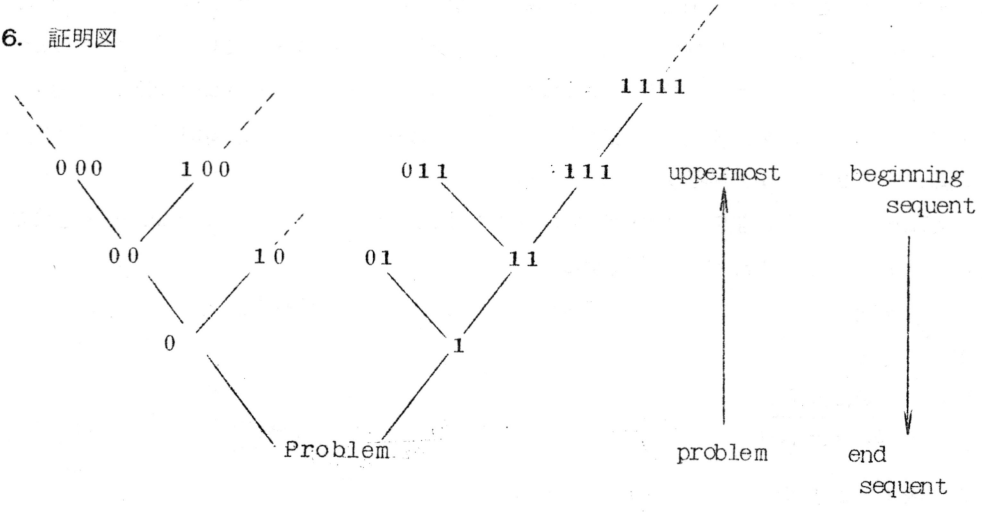
ので，P_i はprime formulaでなければならない．

一般に＆（そして）を含むprenex normal formは数個の theorem に分けて書く必要があるが，〔E_z〕の解は全 theoremsを通じて同一のものでなければならない．

problem が theoremsの集まりに分解されたとき，そのproblemは正しいと結論される．そのまゝの形では適合し得ない場合には，cutという推論規則を用いてlemmaを生産し，theoremsと同様の役割を果たすものとして使用する．theoremsおよびlemmasは初めの方から順に調べられる．

V（あるいは）のない場合は括弧"（"")"は不要．

6. 証明図



$\mathcal{G}_0$，$\mathcal{G}_1$，$\mathcal{G}_2$ がsequentであるとき，$\frac{\mathcal{G}_1}{\mathcal{G}_0}$ または $\frac{\mathcal{G}_1\ \ \mathcal{G}_2}{\mathcal{G}_0}$ という図は" 推論規則を表わし，sequent $\mathcal{G}_0$ がsequent $\mathcal{G}_1$，$\mathcal{G}_2$ から導き出されることを示す．後者のような2 sequents に分れる場合を特に" 分枝" と仮称する．

いくつかのsequentを推論規則によつて上下につなぎ合せ，立木（tree）の形にしたものを" 証明図" という．一番下にあるsequentを" end sequent"，一番上にあるsequentを" beginning sequent" という．

前提となるtheorem（lemma）及びlogical axiomをbeginning sequentとして使用し，提出したproblemがend sequentとなるような証明図を計算機に描かせようとするのが，この" 証明のプログラム" である．

こゝで，problemなるend sequentから始めて，証明図を下から上へとuppermost sequent（primary formulaまたはalah formulaのみで構成されている）まで作り上げるのであるが，分枝がある場合にはbranchに8桁以内の二進数を付して，中間状況を示している．

また途中でlogical axiomとなつて消えて行くsequentや，uppermost sequentには，各々" is logical axiom "　" to uppermost "と出力する．

7. 推論規則

$\mathcal{A}$，$\mathcal{B}$はformula，Φ，Γはsequent，特にΦはprimary formulaかalah formulaから成る．つまりsequent中のformulaは上記2種を除いて左にあるものから分解していく．

alah formula同志では左にあるものから分解する．左端のalah formulaを分解した結果が再びalah formulaである場合は，なるべく他のformulaの分解に移る．

alah formulaを分解すると，既出の（その枝の下式に含まれている）free variableのうちのいずれかを代表するものとして，未使用のalah variable（例えばi）が現われる．iは最後に（証明可能ならば）uppermost sequentにおいて決定される．（§2.の2.1を参照）

alah formula自身にはmarkが付されてsequentの最後尾のformulaとして加えられ，再び使用されるべく待機する．

aは未使用の（その枝の下式にあらわれない）free variableをあらわす．

(1) －（でない）

$$\frac{\Phi, \mathcal{A}, \Gamma}{\Phi, --\mathcal{A}, \Gamma}$$

(2) &（そして）

$$\frac{\Phi, -\mathcal{A}, -\mathcal{B}, \Gamma}{\Phi, -(\mathcal{A}\,\&\,\mathcal{B}), \Gamma} \qquad \frac{\Phi, \mathcal{A}, \Gamma \qquad \Phi, \mathcal{B}, \Gamma}{\Phi, (\mathcal{A}\,\&\,\mathcal{B}), \Gamma}$$

(3) V（あるいは）

$$\frac{\Phi, -\mathcal{A}, \Gamma \qquad \Phi, -\mathcal{B}, \Gamma}{\Phi, -(\mathcal{A}\,\mathrm{V}\,\mathcal{B}), \Gamma} \qquad \frac{\Phi, \mathcal{A}, \mathcal{B}, \Gamma}{\Phi, (\mathcal{A}\,\mathrm{V}\,\mathcal{B}), \Gamma}$$

(4) →（ならば）

$$\frac{\Phi, \mathcal{A}, \Gamma \qquad \Phi, -\mathcal{B}, \Gamma}{\Phi, -(\mathcal{A}\to\mathcal{B}), \Gamma} \qquad \frac{\Phi, -\mathcal{A}, \mathcal{B}, \Gamma}{\Phi, (\mathcal{A}\to\mathcal{B}), \Gamma}$$

(5) ≡（同等である）

$$\frac{\Phi, \mathcal{A}, \mathcal{B}, \Gamma \qquad \Phi, -\mathcal{A}, -\mathcal{B}, \Gamma}{\Phi, -(\mathcal{A}\equiv\mathcal{B}), \Gamma} \qquad \frac{\Phi, -\mathcal{A}, \mathcal{B}, \Gamma \qquad \Phi, \mathcal{A}, -\mathcal{B}, \Gamma}{\Phi, (\mathcal{A}\equiv\mathcal{B}), \Gamma}$$

(6) 〔　〕（すべての）

$$\frac{\Phi, -\mathcal{A}(i), \Gamma, -[x]\mathcal{A}(x)\,(\text{mark付})}{\Phi, -[x]\mathcal{A}(x), \Gamma} \qquad \frac{\Phi, \mathcal{A}(a), \Gamma}{\Phi, [x]\mathcal{A}(x), \Gamma}$$

(7) 〔E〕(存在する)

$$\frac{\Phi, -\mathfrak{A}(a), \Gamma}{\Phi, -[\mathrm{E}_x]\mathfrak{A}(x), \Gamma} \qquad \frac{\Phi, \mathfrak{A}(i), \Gamma, [\mathrm{E}_x]\mathfrak{A}(x)\ (\text{mark付})}{\Phi, [\mathrm{E}_x]\mathfrak{A}(x), \Gamma}$$

(8) definition

$$\frac{\Phi, -\mathfrak{A}(b_1, b_2, \cdots, b_m), \Gamma}{\Phi, -P_{b_1 b_2 \cdots b_m}, \Gamma} \qquad \frac{\Phi, \mathfrak{A}(b_1, b_2, \cdots, b_m), \Gamma}{\Phi, P_{b_1 b_2 \cdots b_m}, \Gamma}$$

ただし，$[x_1]\cdots[x_m](\mathrm{P}_{x_1 \cdots x_m} \equiv \mathfrak{A}(x_1, \cdots, x_m))$

(9) cut

$$\frac{\Phi, \mathfrak{A}, \Lambda \qquad \Pi, -\mathfrak{A}, \Gamma}{\Phi, \Pi, \Lambda, \Gamma}$$

8. lemmaの作成

上記推論規則のうち，cutはいわゆる三段論法に当るものであるが，こゝではcut formula $\mathfrak{A}$ をprime formulaに制限して我々のtheoremに適用することによつてlemmaを作り出す．theoremとlemmaは計算機内部では全く同じ形をもち，対等に扱われる．

各lemmaのfound variableは，x，y，z，u，v，w，s，tの順に使用され，構成要素であるprime formulaは，predicate及びそのvariableのmachine wordとしての大小順に並べられる．

存在変数はrに置換えられる．

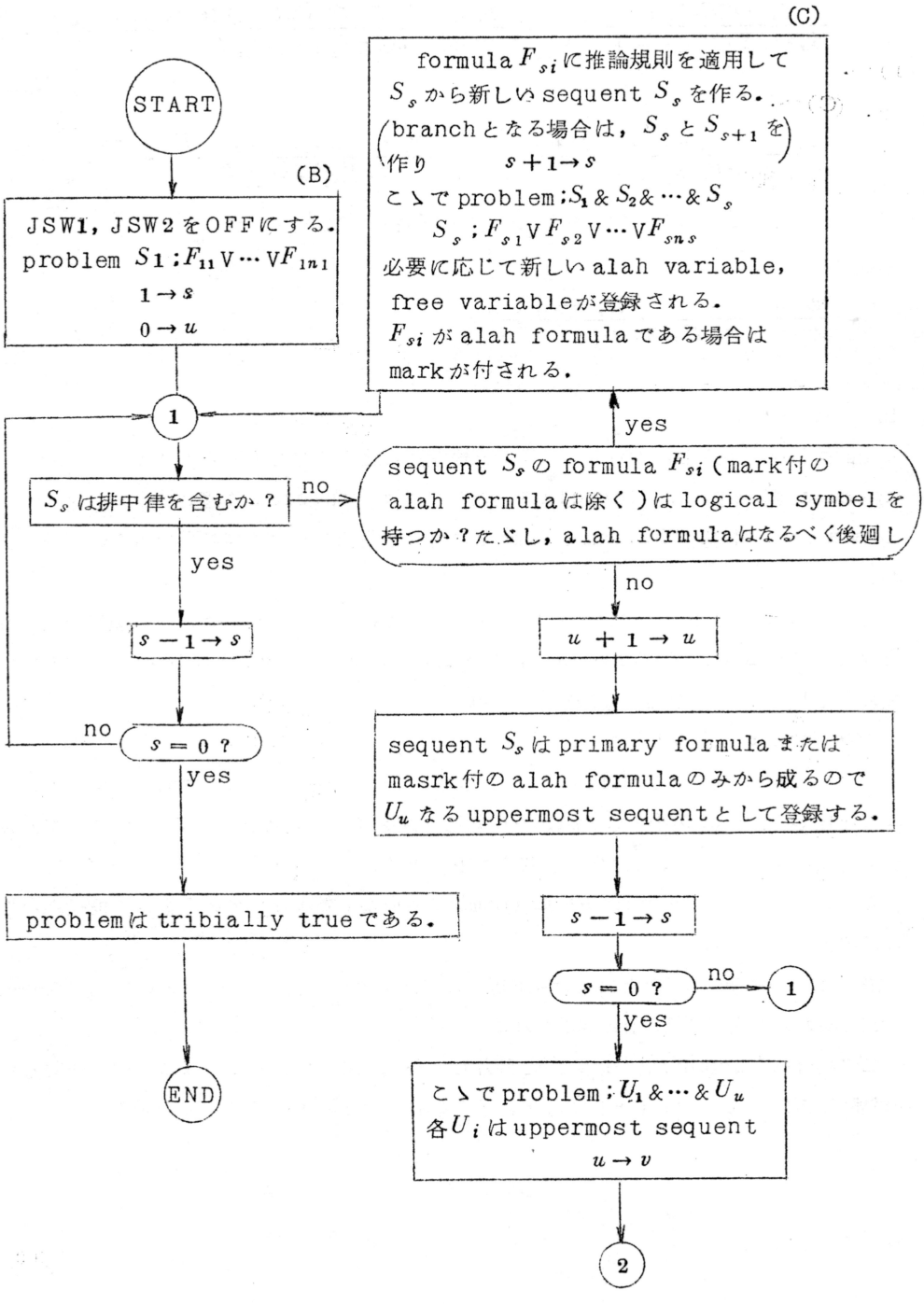
lemma作成の仕事としては，2つのlemmas間におけるcut formulaの検索と，$\mathfrak{A}$の特殊化($\mathfrak{A}$，$-\mathfrak{A}$の間のvariableの対応を調整する)，及び作成されたlemmaの登録(既成のものと同じならば登録不要)が主となる．

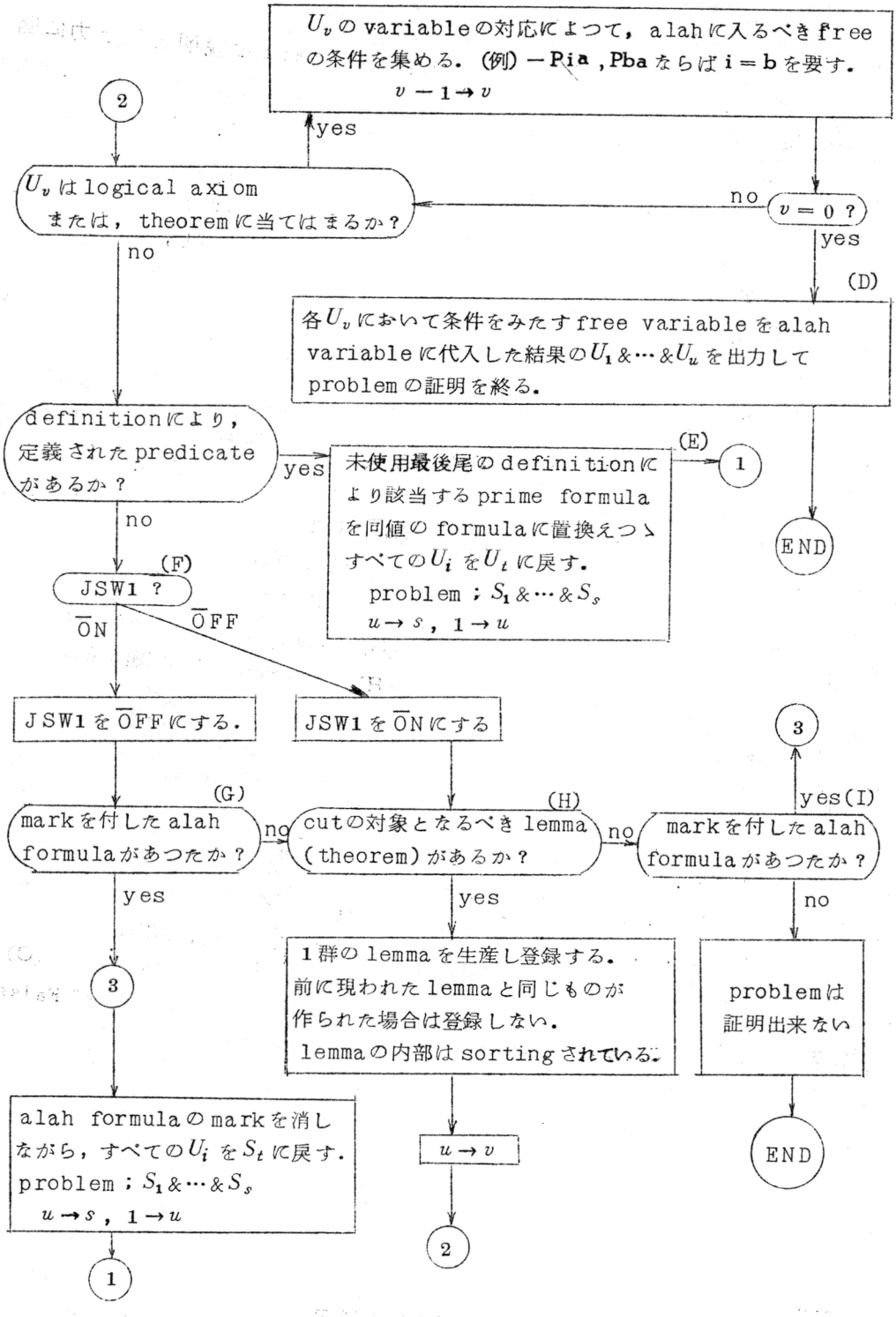
なお，作成されたlemmaが既成のlemmaの特別なものになつている場合の検査が今後の宿題となつている．

帰謬法　相矛盾するtheoremsが始めに与えられた場合は，cutによつて，formulaのない(空の)lemmaが現われる．

この場合は"Inconsistent"と出力して終る．従つて，問題によつては，証明したい結論の否定をtheoremとして与え，矛盾を引き出すという利用法もある．

9. Block Chart




U_v の variable の対応によつて，alah に入るべき free の条件を集める．(例) －Pia，Pba ならば i＝b を要す．
$v-1 \rightarrow v$
2
yes
U_v は logical axiom または，theorem に当てはまるか？
no
$v=0$ ？
yes
(D)
各 U_v において条件をみたす free variable を alah variable に代入した結果の $U_1 \& \cdots \& U_u$ を出力して problem の証明を終る．
definition により，定義された predicate があるか？
yes
(E)
未使用最後尾の definition により該当する prime formula を同値の formula に置換えつゝすべての U_i を U_t に戻す．
problem ; $S_1 \& \cdots \& S_s$
$u \rightarrow s$, $1 \rightarrow u$
1
END
no
(F)
JSW1 ？
$\overline{\text{O}}$N
$\overline{\text{O}}$FF
JSW1 を $\overline{\text{O}}$FF にする．
JSW1 を $\overline{\text{O}}$N にする
3
yes(I)
(G)
mark を付した alah formula があつたか？
no
(H)
cut の対象となるべき lemma (theorem) があるか？
no
mark を付した alah formula があつたか？
yes
yes
no
3
1群の lemma を生産し登録する．前に現われた lemma と同じものが作られた場合は登録しない．lemma の内部は sorting されている．
problem は証明出来ない
alah formula の mark を消しながら，すべての U_i を S_t に戻す．
problem ; $S_1 \& \cdots \& S_s$
$u \rightarrow s$, $1 \rightarrow u$
$u \rightarrow v$
END
1
2

block chartの説明と入出力に関する補足

(A) 問題テープ読込部――入力をPTR，出力をHSPとし，問題テープを読みながら，形式をととのえて出力する．(B)へ．

問題テープの作り方が規定通りでない場合は"Error."と出力して(J)へ．

(B) JSW1及びJSW2をOFFにする．各種initializationを行なう．(C)へ．

(C) 各sequentを分解し，証明の中間結果を出力する．(D)へ．

特にalah formulaの場合は，新たなalah variableが指定されるが，18種のalah variableがすべて指定済となつている場合は"No more alah."と出力して(J)へ．

(D) uppermost sequentのcheck――logical axiomまたはlemma(theorem)が適用出来る場合は"True."及びその理由を出力して(J)へ．

適用出来ない場合は(E)へ．

(E) 未使用のDefinitionがあれば，uppermost sequentの概当するprime formulaに代入して(C)へ．

(F) JSW1が$\overline{\text{O}}$FFならば$\overline{\text{O}}$Nにして(H)へ．$\overline{\text{O}}$Nならば$\overline{\text{O}}$FFにして(G)へ．

(G) alah formulaがあれば再び(C)へ．alah formulaがなければ(H)へ．

(H) theoremのない問題ならば(I)へ．theorem付の問題の場合は，theoremsの組合せにより1群のlemmaを作成し，JSW2が$\overline{\text{O}}$FFならば(D)へ．JSW2が$\overline{\text{O}}$Nならば再び(H)へ．

(H－1) 1群のlemma作成の途中において，formulaのないlemmaが現われたとき――Theorem群の与え方に矛盾がある場合――には，"Inconsistent."と出力して(J)へ．

(H－2) すべての組合せが終つて，lemmaを作りつくした場合は，"No more lemma."と出力して，JSW2が$\overline{\text{O}}$FFならば(I)へ．JSW1が$\overline{\text{O}}$Nならば(D)へ．

(I) alah formulaがあれば(C)へ．alah formulaがない場合は，残つているuppermost sequent及び"False."を出力して(J)へ．

(J) BREAK HALTが$\overline{\text{O}}$FFならば，引続き次の問題を読むべく(A)へ．

BREAK HALTが$\overline{\text{O}}$Nならば，一旦停止する．(OPERATIONを押せば(A)へ．)

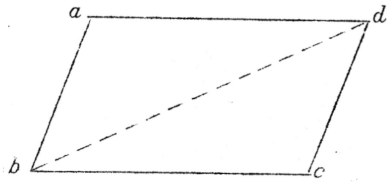
§2．実行例と解説

2.1. われわれは，初等幾何のつぎの問題を考える．

「図のように4辺形$abcd$が与えられたとき，

$ad=bc$ かつ $ad \nparallel bc$

ならば，$ab=cd$でかつ$ab/\!/cd$である」



この問題の証明には，図のような補助線が用いられるのがふつうである．われわれのプログラムも，同様の補助線を用いて証明を完成する．その過程をある程度追跡してみることとする．

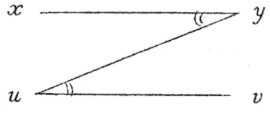
まず問題の formulation から考えよう．

〝図のような4辺形$abcd$〟とはどう表現すべきか．これは，〝線分acとbdが交わる〟と理解してよいであろう．ここでは$Cacbd$と表わすことにしよう．この問題の証明には，当然のことながら，初等幾何の既成の若干の定理が必要である．つぎに列挙してみよう．

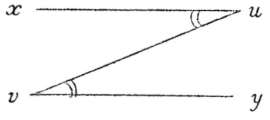
(1) $(x)(y)xy=yx$

(2) $(x)(y)(u)(v)(s)(t)(\neg xy=uv \vee \neg uv=st \vee xy=st)$

(3) $(x)(y)(u)(v)(\neg xy/\!/uv \vee \neg Cxvuy \vee \angle uyx=\angle yuv)$



(4) $(x)(y)(u)(v)(\neg\angle xuv=\angle yvu \vee \neg Cxyuv \vee xu/\!/yv)$



(5) $(x)(y)(z)(u)(v)(w)(\neg xy=uv \vee \neg zy=wv \vee \neg\angle zyx=\angle wvu \vee (xz=uw \ \& \ \angle xzy=\angle uwv))$

〝二辺と**夾角**が等しければ2つの三角形は合同である.〟

これらの前提のもとで，

$\neg ad=bc, \neg ad/\!/bc, \neg Cacbd, (ab=cd \ \& \ ab/\!/cd)$

を証明することになる．これを行なうには，種々の Input の形式があるが，われわれはいま，つぎの形のものを考えることにする．すなわち，列挙した(1)，(2)，(3)，(4)を，**Theorem 1.**，**Theorem 2.**，**Theorem 3.**，**Theorem 4.**とする．そして，(5)はつぎの形で Problem に入れる．

Problem. $\neg(x)(y)(z)(u)(v)(w)(((xy=uv \ \& \ zy=wv) \ \& \ \angle zyx=\angle wvu)\rightarrow(xz=uw \ \& \ \angle xzy=\angle uwv))$,

$\neg ad=bc, \neg ad/\!/bc, \neg Cacbd, (ab=cd \ \& \ ab/\!/cd)$

さて，証明の実行に移る．われわれは，(1)，(2)，(3)，(4)，(5)に現われるx，y，z……等に対して，a，b，c，dのどの変数をも自由に代入することができる．すなわち，a，

b，c，dが自由に代入されるべき場所は，ぜんぶで22ケ所ある．したがつて，可能性の総数は

$$4^{22}=2^{44}$$

あるわけである．このすべてをしらみつぶしに試してみて，われわれの目的にかなう唯1つのものをみつけるのが，もつとも原始的な形でのアルゴリズムである．

この問題のInput形式にすると，これはまず第1にTheoremの活用によつて，

$$4^{12}=2^{24}$$

まで一きよに減少する．つぎに，これから説明する方法を用いると，われわれが試すべき場合の総数は，再び大きく減少し，われわれの手許の計算機で，容易に行ない得，補助線を用いた証明をも得ることができるのである．

まずProblemを推論規則により分解する．これは，こうした問題を，加法標準形に変形する操作に相当する．その結果として，つぎの8つのdisjunctionが得られる．

(1) $ij=\ell m \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab=cd$

(2) $kj=nm \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab=cd$

(3) $\angle kji=\angle nm\ell \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab=cd$

(4) $-ik=\ell n \ \vee\ -\angle ikj=\angle \ell nm \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab=cd$

(5) $op=i'j' \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab/\!/cd$

(6) $qp=k'j' \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab/\!/cd$

(7) $\angle qpo=\angle k'j'i' \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab/\!/cd$

(8) $-oq=i'k' \ \vee\ -\angle oqp=\angle i'k'j' \ \vee\ -ad=bc \ \vee\ -ad/\!/bc \ \vee\ -Cacbd \ \vee\ ab/\!/cd$

ここに現われたi，j，k，ℓ，m，n，o，p，q，i'，k'，j'はa，b，c，dが自由に代入されるべき場所を表わす．これら(8)つの式が，i，j，…，k'，j'にa，b，c，dのうちの適当なものを代入したとき，すべて正しくなるとき，Problemは証明されたことになるわけでする．さて，まず(1)の式をみてみよう．(1)にはi，j，ℓ，mが現われる．試みるべき対象は

$$4^4=2^8$$

通りある．しかし，(1)が正しくなるのは，これが排中律を含むか，あるいはTheorem 1 ～ Theorem 4のどれかに含まれるときに限る．これはどんなときに可能か．

(1.1)　$i:=a$，$j:=d$，$\ell:=b$，$m:=c$

のときに排中律を含む．

さらに，(1)に含まれるPredicateの種類から，これがTheorem 1 ～ Theorem 4になる可能性があるのは，

（1.2）　$i:=m$, $j:=n$

のときに限ることもすぐわかる．

このようにして，

$$2^8$$

は，

$$2^1$$

に減つてしまう．このようにして，おのおのの式に対して，順次代入の可能性を列記していく．この可能性が，矛盾なく成立していけばProblemの証明が得られるわけである．このProblemでは，じつは途中で矛盾に出くわす．矛盾が出てくるとは，可能性をつなぐと，たとえば，1つのiに異なつたaとbを同時に代入しなければならなくなることをさす．こうして，このTheorem 1～Theorem 4の範囲でProblemが正しくならなくても，まだProblemが間違つているとは即断できない．つぎに行なうのは，Theorem同志を用いて，三段論法によつてlemmaをつくり出し，これとの照合を行なうことである．それぞれのTheoremを1回ずつ用いて，lemmaを生産すれば，8つのlemmaが得られる．たとえば，Theorem 1とTheorem 2を用いると，

Lemma 1.　$(x)(y)(z)(u)(-zu=xy \ \vee \ zu=yx)$

といつたlemmaが得られる．

こうして得られたlemmaに対して，ふたたび代入の可能性に対する条件をしらべていくことになる．すると，つぎのような，矛盾のない1つの代入が得られる．

$i:=a$	$o:=a$
$j:=d$	$p:=d$
$k:=b$	$q:=b$
$l:=c$	$i':=c$
$m:=b$	$y':=b$
$n:=d$	$k':=d$

このとき，

(1)は　$-ad/\!/bc \ \vee \ -ad=bc \ \vee \ -Cacbd \ \vee \ ab=cd \ \vee \ ad=cb$:
is true by Lemma 1.

(2)は　$-ad/\!/bc \ \vee \ -ad=bc \ \vee \ -Cacbd \ \vee \ ab=cd \ \vee \ bd=db$:
is true by Theorem 1.

(3)は　$-ad/\!/bc \ \vee \ -ad=bc \ \vee \ -Cacbd \ \vee \ \angle bda=\angle dbc \ \vee \ ab=cd$:
is true by Theorem 3.

(4)は　$-ab=cd \ \vee \ -\angle abd=\angle cdb \ \vee \ -ad=bc \ \vee \ -ad/\!/bc \ \vee \ -Cacbd \ \vee \ ab=cd$:　**is logical axiom.**

(5)は　$-ad/\!/bc \vee -ad=bc \vee -Cacbd \vee ad=cb \vee ab/\!/cd$:
is true by Lemma 1.

(6)は　$-ad/\!/bc \vee -ad=bc \vee -Cacbd \vee bd=db \vee ab/\!/cd$:
is true by Theorem 1.

(7)は　$-ad/\!/bc \vee -ad=bc \vee -Cacbd \vee \angle bda=\angle dbc \vee ab/\!/cd$:
is true by Theorem 3.

(8)は　$-ad/\!/bc \vee -ab=cd \vee -ad=bc \vee -Cacbd \vee -\angle abd=\angle cdb \vee ab/\!/cd$:　is true by Theorem 4.

となるわけである。

ここでは，補助線 bd を引くことは，表面的には目立たない．しかし，得られた証明を観察すれば，補助線 bd を引いて証明がなされていることがわかる．これは，前提のなかの(5)，"２辺と夾角が等しければ２つの３角形は合同である"という定理を，３角形 adb と３角形 cbd の間に適用したことにあたる．すなわち，$i, j, k, l, m, n, o, p, i', j', k'$ に対して，a, b, c, d のなかのなにを代入すべきかを探す操作のなかに，補助線を探すことも含まれていたわけである．

証明に要した時間は２分１０秒であつた．（Inputや outputに要した時間は省く）

2.2. このプログラムでは，１つの問題の証明を行なわせるのに，input の形を変えることにより，証明獲得の過程に変化を与えることができる．それによつて，証明を得るに要する時間も異なる．以下で若干の実行例について述べる．

2.2.1　命題論理では，Principia Mathematicaにあるはじめ５章の約２００題を連続的に試みた．証明に要した時間は合計で約３分である．

2.2.2　「２等辺３角形の両底角は相等しい」

(1)　$(x)(y)(z)(u)(v)(w)(-xy=uv \vee -xz=uw \vee -\angle yxz=\angle vuw \vee \angle xyz=\angle uvw)$

(2)　$(x)(y)(z)\angle yxz=\angle zxy$

(3)　$(x)(y)(u)(v)(-xy=uv \vee uv=xy)$

の仮定の下で

$$-ab=ac \vee \angle abc=\angle acb$$

を証明させる．

Theorem として，(1)のみを用い，(2)，(3)を Problem のなかに入れた場合，所要時間７秒．

Theorem として，(1)，(2)，(3)を用いた場合，１３ケの lemmaをつくり出す．所要時間３３秒．

2.2.3 「3角形abcにおいて，$\angle abc<\angle acb$ならば$ac<ab$である」

前提(1) $uv<xy \ \vee \ uv=xy \ \vee \ xy<uv$

(2) $-\angle xyz<\angle uvw \ \vee \ -\angle xyz=\angle uvw$

(3) $-\angle xyz<\angle uvw \ \vee \ -\angle uvw<\angle xyz$

(4) $-xy=xz \ \vee \ \angle xzy=\angle xyz$

(5) $-xy<xz \ \vee \ \angle xzy<\angle xyz$

この前提のもとで

$$-\angle abc<\angle acb \ \vee \ ac<ab$$

を証明する．

Theorem として(1)，(2)，(3)を用い，(4)，(5)をProblem に入れたとき，所要時間12秒．

Theorem として(1)，(2)，(3)，(4)，(5)を用いたとき，73ケのlemmaをつくり出し，所要時間10分30秒で完了．

いづれの場合も，ふつうの幾何の教科書通りの証明である．

2.2.4 「3角形abcの辺ab，bd，caの垂直2等分線は1点に会する」（外心）

前提(1) $-Bxyz \ \vee \ -xy=zy \ \vee \ -Ruyx \ \vee \ xu=zu$

(2) $-xy=uv \ \vee \ uv=xy$

(3) $-xu=zu \ \vee \ -xy=zy \ \vee \ -Bxyz \ \vee \ Ruyx$

(4) $-xy=uv \ \vee \ -uv=st \ \vee \ xy=st$

のもとで，

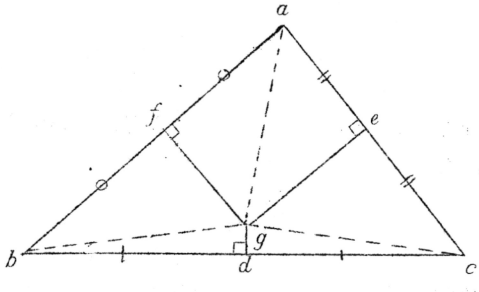
$$-Bbdc \ \vee \ -Bcea \ \vee \ -Bafb \ \vee \ -af=bf \ \vee \ -bd=cd \ \vee \ -ce=ae$$
$$\vee \ -Rgfa \ \vee \ -Rgdb \ \vee \ Rgec$$

を証明する．ここで

$Bxyz$ ：点yは点xと点zの間にある

$Rxyz$ ：$\angle xyz$＝直角

これは8つのlemmaをつくり，所要時間は9分55秒．証明は図から推られる方法である．点線は補助線である．



2.2.5　Definition を用いる例としては，たとえばつぎのようなものがある．

Definition 1.　$x=y \equiv (u)(u \in x \equiv u \in y)$

Definition 2.　$z \in \{xy\} \equiv (z=x \ \vee \ z=y)$

Definition 3.　$\{xy\}=\{uv\}=(z)(z \in \{xy\} \equiv z \in \{uv\})$

Theorem 1.　$x=x$

Theorem 2.　$\neg x=y \ \vee \ y=x$

Problem.　$\neg \{ab\}=\{cd\} \ \vee \ ((a=c \ \& \ b=d) \ \vee \ (b=c \ \& \ a=d))$

最終的には20ケのuppermost sequentに分解される．所要時間は45秒．

2.2.6　その他，初等幾何で

「3角形の内角の和は2直角に等しい」

(inputの与え方により35秒ないし1分32秒)

等いくつかの例を試みた．

2.2.7　代数的な問題では，

「left solutionとright solutionをもつ，結合法則のなりたつ体系では，右単位元が存在する」

(所要時間23秒，あるいは1分34秒)

等若干の例を試みた．

§3．問題点

1.　pattern recognitionの良い方法を見つけること．

　○　既成lemmaと新lemmaの関係

　○　lemmaのuppermost sequentに対する当てはめ

2.　述語論理の体系自身からの制約を解くこと．

　○　equality axion

　○　function

3.　variableとして使用出来る文字を増加すること．

その他，現状の計算機の不備(memory，speed，I／$\overline{O}$などについて)，及び計算機の利用技術の未熟さも相俟つて，なかなか前途遠大なるものがある．

§4．参考文献

西村敏男：　計算機による数学の開発 ― 計算機に数学の証明を行なわせる試み ―

日本電子工業振興協会，昭和40年1月

本 PDF ファイルは 1967 年発行の「第 8 回プログラミング―シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間： 2020 年 12 月 18 日 ～ 2021 年 3 月 19 日
掲載日： 2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html