80. *Standard Form in PGO and Transformation Algorithm: Problem-Solving Machines. II*

By Motokiti KONDÔ*) and Haruo MURATA**)

(Comm. by Zyoiti SUETUNA, M.J.A., May 19, 1965)

1. **Definitions.** In the former paper "Problem-Solving Machines, I", we reported on a method to produce the code expression from any natural language sentence of plane geometry, providing the code system PGO, and to retrieve the proof of any given theorem by machine. Now, we provide below some efficient standardization of any logical expression in plane geometry, named the *standard form*, and give a transformation algolithm.

Definition 1. A formula is a finite sequence of atomic formulas,[1] logical symbols,[2] auxiliary symbols[3] except for comma.

The order of "binding" of logical symbols coincide with that of conventional mathematical usage, that is, in descending order by degree; $\neg, \cap, \cup, \rightarrow$.

Definition 2. An atomic formula is a literal; and if Q is an atomic formula then $\neg Q$ is a literal.

Definition 3. Well formed formula (wff): 1. An atomic formula is a wff. 2. If F is a wff, then $\neg F$ is a wff. 3–5. If E and F are wff's, then $E \cap F$, $E \cup F$, and $E \rightarrow F$ are wff's. 6. The only wff's are those given by 1–5.

2. **Standard form of the formula.** Theorems in plane geometry consist of the hypothesis and the conclusion part, that is, let A and B be wff's, the theorem is usually in the form

$$A \rightarrow B . \tag{2.1}$$

Using the disjunctive normal form in the hypothesis A and the conjunctive normal form in the conclusion B,

$$\bigcup_{i=1}^{m_0} \bigcap_{j=1}^{n_i} A_{ij} \rightarrow \bigcap_{k=1}^{m_1} \bigcup_{l=1}^{n_k} B_{kl} , \tag{2.2}$$

where A_{ij} and B_{kl} are literals.

It is easily seen that the disjunctions in the hypothesis and the conjunctions in the conclusion can be transfered to the front of the formula as conjunctions. Then we have

---

*) Tokyo Metropolitan University. **) IBM Japan, Tokyo.

1) atomic fomula: A string of symbols consisting of a predicate letter followed by n terms.

2) logical symbols: $\neg, \cap, \cup, \rightarrow$, which mean negation, and, or, and imply.

3) auxiliary symbols: (,) and comma.

(2.3) $$\bigcap_{i\ k}^{m_0\ m_1}\left\{\bigcap_{j=1}^{n_i} A_{ij}\rightarrow\bigcup_{l=1}^{n_k} B_{kl}\right\}.$$

Hence, in order to give the proof of the theorem, we have only to execute retrieving the following formula at most $m_0\cdot m_1$ times:

(2.4) $$\bigcap_{j=1}^{n_i} A_{ij}\rightarrow\bigcup_{l=1}^{n_k} B_{kl}.$$

Now, we put

$$\begin{aligned} Q_{ij} &= \neg A_{ij}, && \text{if } \neg A_{ij} \text{ is atomic,}\\ Q'_{ij} &= A_{ij}, && \text{otherwise,}\\ Q_{kl} &= B_{kl}, && \text{if } B_{kl} \text{ is atomic,}\\ Q'_{kl} &= \neg B_{kl}, && \text{otherwise.}\end{aligned}$$

Then, by suitable permutation, we obtain from (2.4)

$$\underbrace{(\neg Q'_{\sigma_1}\cup\neg Q'_{\sigma_2}\cup\cdots\cup\neg Q'_{\sigma_p})}_{\text{negative part}}\cup\underbrace{(Q_{\tau_1}\cup\cdots\cup Q_{\tau_q})}_{\text{affirmative part}}.$$

Therefore, we have

(2.5) $$\bigcap_{\mu=1}^{p} Q'_{\sigma_\mu}\rightarrow\bigcup_{\nu=1}^{q} Q_{\tau_\nu}.$$

The formula (2.5) is called the *standard form* of (2.2).

3. **Transformation procedure.** Now, a procedure transforming the code expression to the standard form is given.

P0: Enclose each of operands of logical operation in parentheses in order of occurrence from the leftmost, according to the "binding" order (hierarchy) of logical symbols.[4)]

Example. $(((Q_1)\cap(\neg(Q_2)))\cap(Q_3))\rightarrow(\neg((Q_4)\cup(Q_5)))\cap(\neg(Q_6))$.

Let $F_1F_2\cdots F_M$ be a wff F resulting from P0, where each F_i is either symbol or atomic formula.

Definition 4. Scope: Let F_i be one of logical symbols, appearing in a wff F, then F_{i+1} is the left parenthesis, "(", and if F_i is either of "$\rightarrow$", "$\cup$" or "$\cap$", then F_{i-1} is the right parenthesis, ")". Let F_p and F_q be the conjugate parenthesis of F_{i-1} and F_{i+1}, respectively. Then, the scope of either of "$\rightarrow$", "$\cup$" of "$\cap$" is the following strings; $F_pF_{p+1}\cdots F_{i-2}F_{i-1}$ and $F_{i+1}F_{i+2}\cdots F_{q-1}F_q$, and the scope of "$\neg$" is the string $F_{i+1}F_{i+2}\cdots F_q$. In particular, the left scope of either logical symbol "$\rightarrow$", "$\cup$" or "$\cap$" is $F_p\cdots F_{i-1}$, and the right scope of it is $F_{i+1}\cdots F_q$.

P1. "$\rightarrow$"-elimination: Let F_i be the symbol "$\rightarrow$", appearing first from the leftmost in a wff F, $3<i<M-3$. Eliminate both the scope of F_i and F_i itself, and produce $(\neg F_pF_{p+1}\cdots F_{i-1})\cup F_{i+1}F_{i+2}\cdots F_q$ in that place. Henceforth, the production like this is to be written as "PROD":

$$F_1\cdots F_p\cdots F_q\cdots F_M \text{ PROD } F_1\cdots F_{p-1}(\neg F_p\cdots F_{i-1})\cup F_{i+1}\cdots F_M.$$

4) This algorithm exists; cf. [3].

Repeat this procedure until all "$\rightarrow$" symbols in each of the hypothesis A and the conclusion B are eliminated.

P2. Let $F_iF_{i+1}F_{i+2}$ be either string "$\neg(($"or"$\neg(\neg$", appearing first from the leftmost of a wff F.

Case 1. The scope of F_i is $(\neg(\Gamma))$: where Γ is a string,

$$F \text{ PROD } F_1 \cdots F_{i-1}F_{i+3}F_{i+4} \cdots F_{q-1}F_{q+1} \cdots F_M .$$

Case 2. The scope of F_i is $((\Gamma) \cap (\Delta))$, where Γ and Δ are strings:

$$F \text{ PROD } F_1 \cdots F_{i-1}((\neg(\Gamma)) \cup (\neg(\Delta)))F_{q+1}F_{q+2} \cdots F_M .$$

Case 3. The scope of F_i is $((\Gamma)) \cup (\Delta))$:

$$F \text{ PROD } F_1 \cdots F_{i-1}((\neg(\Gamma)) \cap (\neg(\Delta)))F_{q+1}F_{q+2} \cdots F_M.$$

Repeat this procedure until all such strings are handled in both of the hypothesis A and conclusion B.

P3–A. Application of the 1st distributive law[5] for the hypothesis A: Let A_i be the "$\cap$" in either string "$\cap(($"or"$))\cap$", appearing first from the leftmost of the hypothesis A.

Case 1. The scope of A_i is $(\Gamma)A_i((\Phi) \cup (\Psi))$, where each of Γ, Φ, and Ψ is a string:

$$A \text{ PROD } A_1 \cdots A_{p-1}((\Gamma) \cap (\Phi)) \cup ((\Gamma) \cap (\Psi))A_{q+1} \cdots A_M,$$

Case 2. The scope of A_i is $((\Phi) \cup (\Psi))A_i(\Gamma)$:

$$A \text{ PROD } A_1 \cdots A_{p-1}((\Phi) \cap (\Gamma)) \cup ((\Psi) \cap (\Gamma))A_{q+1} \cdots A_M.$$

P3–B. Application of the 2nd distributive law[6] for the conclusion B: Let B_i be the "$\cup$", appearing first from the leftmost of a wff B.

Case 1. The scope of B_i is $(\Gamma)B_i((\Phi) \cap (\Psi))$:

$$B \text{ PROD } B_1 \cdots B_{p-1}((\Gamma) \cup (\Phi)) \cap ((\Gamma) \cup (\Psi))B_{q+1} \cdots B_M.$$

Case 2. The scope of B_i is $((\Phi) \cap (\Psi))B_i(\Gamma)$:

$$B \text{ PROD } B_1 \cdots B_{p-1}((\Phi) \cup (\Gamma)) \cap ((\Psi) \cup (\Gamma))B_{q+1} \cdots B_M.$$

Repeat these procedures P3–A and B until all such strings are completed.

P4. Eliminate "("or")", if "("or")" is followed by another "("or")".

The result of P0–P4 is in the form of (2.2).

P5. Let Γ be the left scope of the "$\cup$", appearing first from the leftmost of the hypothesis part A. Let Δ be the left scope of the "$\cap$", appearing first from the leftmost of the conclusion part B. Then

$$\Gamma = (A_1) \cap (A_2) \cap \cdots \cap (A_s), \quad \Delta = (B_1) \cup (B_2) \cup \cdots \cup (B_t),$$

where each of A_j and B_l is a literal. Produce the following formula:

---

5) 1st distributive law: $X \cap (Y \cup Z) \rightarrow (X \cap Y) \cup (X \cap Z)$,
$(X \cup Y) \cap Z \rightarrow (X \cap Z) \cup (Y \cap Z)$.

6) 2nd distributive law: $X \cup (Y \cap Z) \rightarrow (X \cup Y) \cap (X \cup Z)$,
$(X \cap Y) \cup Z \rightarrow (X \cup Z) \cap (Y \cup Z)$.

(3.1) $$\Gamma\rightarrow\Delta\ .$$

And apply P1, P2, and P4 for this formula. Now, the result is in the form

(3.2) $$E_1\cup E_2\cup\cdots\cup E_N\ ,$$

where each of E_i is an parenthesized literal, that is, either (Q_i) or $(\neg(Q_i))$, where Q_i is atomic.

P6. Let (Q_j) be the scope of "$\neg$", appearing first from the leftmost of (3.2):

$$E_1\cup\cdots\cup E_{j-1}\cup(\neg(Q_j))\cup E_{j+1}\cup\cdots\cup E_N$$

$$\text{PROD }(Q_j)\cap E_1\cup E_2\cup\cdots\cup E_{j-1}\cup E_{j+1}\cdots E_N\ .$$

Then P0 is applied for the resulting formula. Repeat this precedure until all of negation symbols in (3.2) is transformed into

(3.3) $$(Q_{i1})\cap(Q_{i2})\cap\cdots\cap(Q_{i\mu})\cup(Q_{k1})\cup(Q_{k2})\cup\cdots\cup(Q_{k\nu})\ .$$

P7. Eliminate the "$\cup$", appearing first from the leftmost of (3.3), and then produce the "$\rightarrow$" in that place.

P8. Eliminate all parentheses except for proper ones within an atomic formula, and then permute elements in each of the left scope and the right scope of "$\rightarrow$" in the resulting formula of P7 in lexicographical order.

The resulting formula is in the form of

(3.4) $$Q_{i1}\cap Q_{i2}\cap\cdots\cap Q_{i\mu}\rightarrow Q_{k1}\cup Q_{k2}\cup\cdots\cup Q_{k\nu}\ .$$

The formula (3.4) is the final result of P0–P8, and this formula is the standard form in PGO.

4. **Proof Retrieval.** Using the standard form of the input formula as an index to the thesaurus, retrieval of the proof is executed by machine. If there exists the matching solution, the input formula is provable. Otherwise, the following procedures are taken.

P9. Eliminate Δ in the conclusion part in the resulting formula of P4 and return to P5. Repeat this procedure until the conclusion part becomes empty.

P10. Eliminate Γ in the hypothesis part in the resulting formula of P4, produce $B=\bigcap_{k=1}^{m_1}\bigcup_{l=1}^{n_k}(B_{kl})$ in place of its conclusion part and return to P5.

The procedures P9–P10 are taken at most $m_0\cdot m_1$ times.

5. **Exsamples.** Let a, b, c, and d be points, respectively. $a=b$ means that one point a equals to one point b. $G_r(a, b, c)$ means that three points a, b, and c lie on one straight line. By the code system PGO, these are represented as follows: a; 00000, b; 00001, c; 00002, d; 00003, $=$; 710, and G_r; 170.

(5.1) $$a=b\rightarrow(a=c\rightarrow b=c)$$

The stanbard form of (5.1) is $a=b\cap a=c\rightarrow b=c$, and its code expression is 710(00000,00001) AND 710(00000,00002)→710(00001,00002).

(5.2) $(G_r(a, b, c) \cap G_r(a, b, d) \cap \neg a=b) \rightarrow G_r(a, c, d)$.

The standard form of (5.2) is $G_r(a, b, c) \cap G_r(a, b, d) \rightarrow a=b \cup G_r(a, c, d)$, and its code expression is

170(00000,00001,00002) AND 170(00000,00001,00003)→
170(00000,00002,00003) OR 710(00000,00001).

Furthermore, redundant formulas in each of the hypothesis and conclusion parts are negligible. That is, let $\Delta'(\Gamma')$ be the proper partial conclusion (hypothesis) in the standard form of a formula $\Gamma \rightarrow \Delta$. Suppose $\Gamma' \rightarrow \Delta' \Rightarrow \Gamma \rightarrow \Delta$, then $\Gamma \rightarrow \Delta$ is called reducible. We have only to make a dictionary for retrieval consisting of only irreducible standard formulas. This problem will be argued in the next paper.

References

[1] Kondô, M., and Murata, H.: On proof retrieval: problem-solving machines. I. Proc. Japan Acad., **41**, 254-259 (1965).

[2] Robinson, J. A.: A machine oriented logic based on the resolution principle. J. ACM **12** (1), (1965).

[3] Sheridan, P. B.: The arithmetic translator. Compiler of the IBM FORTRAN automatic coding system. C. ACM, **2** (2), (1959).