Checking solutions of tasks on expressibility in Boolean algebra of sets

R. Prank

University of Tartu, Estonia, rein.prank@ut.ee

The paper describes some steps in a trial to computerize a new type of exercises in Predicate Logic. Many introductory courses contain exercises on expression of predicates using first order formulas in some given signature of constant, functional and predicate symbols. Most exploited mathematical topic is here arithmetic of natural numbers using signature $\langle\, 0\; ;\, ', +, \cdot\; ; = \rangle$ (or similar). For example, "x is even", "$x/y = z$" and "$x \leq y$ are quite easy tasks but "x is prime" or "x is greatest common divisor of y and z" are harder for students.

In this paper we consider another quite reasonable exercise topic - predicates defined on subsets of a fixed set, for example of set of natural numbers N. Books on Boolean algebras or lattices show how the elementary statements of these theories can be formulated in the signature of Boolean operations $\langle\, ', \cap, \cup\; ; = \rangle$ or in the signature of order relation $\langle \subseteq \rangle$. For example,

$$X \subseteq Y \Leftrightarrow X \cup Y = Y, \tag{1}$$

$$X \cup Y = Z \Leftrightarrow (X \subseteq Z) \wedge (Y \subseteq Z) \wedge \forall W [(X \subseteq W) \wedge (Y \subseteq W) \rightarrow (Z \subseteq W)]. \tag{2}$$

In our course we use both signatures for expression of predicates like "$X = Y$", "$X = \emptyset$", "$X' = Y$", "$X \setminus Y = Z$", "$|X| = m$" but also for their combinations: "$X \cap (Y \cup Z) = W$" or "X is union of Y and some 2-element set".

The weaker students compose often wrong answers to expressibility tasks. It would be desirable to create a computerized solution environment. Existing general-purpose methods of expression handling enable detection of (quite frequent) technical errors: incorrect syntax, superfluous or missing free variables, using symbols that do not belong to the required signature, confusion of set-theoretic expressions and formulas. But the main problem is checking of correctness of answer. Correctness of answer of expressibility tasks means equivalence with the 'correct' formula. For arithmetic of natural numbers, equivalence of first order formulas is undecidable. For being able to evaluate student answers, the Tarski's World [1] uses exercises with predicates on finite domains. For the class of all Boolean algebras and also for any particular Boolean algebra the problem of equivalence is decidable [2, 3]. In our project we investigate the question whether the equivalence (in algebra $P(N)$) can be checked sufficiently quickly.

The following propositions describe what defines the truth-value of formulas of the signature $\sigma = \langle\ \emptyset;\ ',\cap,\cup\ ;=\rangle$. For sets $X_1,...,X_n$ let $\pi_i(X_1,...,X_n)$ denote their "Venn intersections" $X_1 \cap ... \cap X_n,\ ...,\ X_1' \cap ... \cap X_n'$ (where $1 \le i \le 2^n$).

Propositon 1. Let $F(X_1,...,X_n)$ be any quantifier-free formula with free variables $X_1,...,X_n$ in signature σ. If $A_1,...,A_n$ and $B_1,...,B_n$ are collections of sets having equal Venn diagrams, i.e.

$\pi_i(A_1,...,A_n) = \emptyset \Leftrightarrow \pi_i(B_1,...,B_n) = \emptyset \qquad (1 \le i \le 2^n)$.

Then $\quad F(A_1,...,A_n) = t \Leftrightarrow F(B_1,...,B_n) = t$.

Quantified formulas enable describe also finite cardinalities of sets. For $1 \le i \le 2^n$ we have

$|A| = m \Leftrightarrow \exists Y_1...\exists Y_n(A \cap \pi_1(Y_1,...,Y_n) \neq \emptyset \wedge ... \wedge A \cap \pi_m(Y_1,...,Y_n) \neq \emptyset) \wedge$
$\neg\exists Y_1...\exists Y_n(A \cap \pi_1(Y_1,...,Y_n) \neq \emptyset \wedge ... \wedge A \cap \pi_{m+1}(Y_1,...,Y_n) \neq \emptyset)$.

The following proposition tells that the only expressible predicates are combinations of cardinalities of regions of Venn diagrams.

Propositon 2. Let $G(X_1,...,X_n)$ be any formula of signature σ that does not contain other free variables beside $X_1,...,X_n$ and where the maximal number of nested quantifiers is k. If $A_1,...,A_n$ and $B_1,...,B_n$ are such collections of sets that for every i (where $1 \le i \le 2^n$) the following holds:

1) $|\pi_i(A_1,...,A_n)| \ge 2^k \Leftrightarrow |\pi_i(B_1,...,B_n)| \ge 2^k$

2) if $\quad |\pi_i(A_1,...,A_n)| < 2^k \quad$ then $\quad |\pi_i(A_1,...,A_n)| = |\pi_i(B_1,...,B_n)|$.

Then $\quad G(A_1,...,A_n) = t \Leftrightarrow G(B_1,...,B_n) = t$.

Corollary. For characterization of any formula it is sufficient to find its truth-values for all combinations of cardinalities $0,...,2^k$ of regions of Venn diagram of the free variables.

If the formula has n free variables and maximal number of nested quantifiers is k then the Venn diagram contains 2^n regions and we should examine $2^k + 1$ possible cardinalities for each region i.e. our 'extended column' of truth-values contains $(2^k + 1)^{2^n}$ bits. Numbers of necessary truth-values are presented in the following table:

Table 1. Free variables, nested quantifiers and numbers of truth-values

k (nested quantifiers) **n (free variables)**	**0**	**1**	**2**	**3**	**4**
1	4	9	25	81	289
2	16	81	625	6561	83521
3	256	6561	390625	43046721	
4	65536	43046721			

Most of 'elementary' predicates in usual student exercises have 1-3 arguments. Bigger numbers appear when we express composite predicates. Typical examples are here predicates describing combined set-theoretical expressions like $(X \cup Y) \cap Z = W$ but also $X \setminus Y = Z$. They require in signature $\langle \subseteq \rangle$ formulas with 4+2 and 3+3 variables. In case of algebraic signature the formulas are less complex and this allows also simplifying of internal representations. At the moment of composing the abstract we work on calculation of cases 4+1 and 3+3 for acceptable time.

Fortunately the natural solution strategy of expressibility tasks is not immediate input of the final formula. Already in paper-and-pencil technology we recommend the students to solve tasks step by step building some intermediate predicates. In computer environment we can predict this approach more efficiently, proposing appropriate choice of intermediate predicates (we can also allow substitution of formulas from earlier tasks). This way allows also checking the intermediate formulas step by step and reducing the number of nested quantifiers.

What kinds of feedback can be provided, using our computing engine? First the program can use traditional methods for checking the syntax, free variables, usage of signature symbols and expressions of correct type. Next, the described above technical reasons enforce the program to reject the formulas that are too complex (contain too much nested quantifiers). After that the main loop of the program counts the truth-values of etalon formula and student formula for all cardinality cases from the Proposition 2. If some distribution of cardinalities of regions of Venn diagram gives a wrong truth-value then the program can use these cardinalities for building a concrete example of sets where the formula fails. For example, if the student enters for the predicate $X \cup Y = Z$ the formula $(X \subseteq Z) \wedge (Y \subseteq Z)$ instead of formula (2) then the program can respond with the simplest counterexample $X = \emptyset$, $Y = \emptyset$, $Z = \{0\}$.

References

[1] J. Barwise and J. Etchemendy, *Language, Proof and Logic*, Stanford : CSLI Publications, (2007).

[2] A. Tarski, *Arithmetical classes and types of Boolean algebras*, Bull. Amer. Math. Soc., **55**, pp. 64 (1949).

[3] Yu. Ershov, *Decidability of the elementary theory of relatively complemented distributive lattices and of the theory of filters*, Algebra i Logica, **3**, 3, pp. 17-38 (1964) (Russian).