McCluskey
Fišer
Hlavička
monotone
positive
notations: $\subseteq$
Quine
satisfiability–satisfiable
P=NP?

In many simple cases, the disjunctive prime form is the shortest possible disjunctive normal form that a function can have. But we can often do better, because we might be able to cover all the necessary points with only a few of the maximal subcubes. For example, the prime implicant $(y \wedge z)$ is unnecessary in (27). And in expression (30) we don't need both $(\bar{w} \wedge \bar{y} \wedge \bar{z})$ and $(x \wedge \bar{y} \wedge \bar{z})$; either one is sufficient, in the presence of the other terms.

Unfortunately, we will see in Section 7.9 that the task of finding a best disjunctive normal form is NP-complete, thus quite difficult in general. But many useful shortcuts have been developed for sufficiently small problems, and they are well explained in the book *Introduction to the Theory of Switching Circuits* by E. J. McCluskey (New York: McGraw–Hill, 1965). For later developments, see Petr Fišer and Jan Hlavička, *Computing and Informatics* **22** (2003), 19–51.

There's an important special case for which the shortest DNF is, however, easily characterized. A Boolean function is said to be *monotone* or *positive* if its value does not change from 1 to 0 when any of its variables changes from 0 to 1. In other words, f is monotone if and only if $f(x) \le f(y)$ whenever $x \subseteq y$, where the bit string $x = x_1 \ldots x_n$ is regarded as contained in or equal to the bit string $y = y_1 \ldots y_n$ if and only if $x_j \le y_j$ for all j. An equivalent condition (see exercise 21) is that the function f either is constant or can be expressed entirely in terms of $\wedge$ and $\vee$, without complementation.

Theorem Q. *The shortest disjunctive normal form of a monotone Boolean function is its disjunctive prime form.*

Proof. [W. V. Quine, *Boletín de la Sociedad Matemática Mexicana* **10** (1953), 64–70.] Let $f(x_1, \ldots, x_n)$ be monotone, and let $u_1 \wedge \cdots \wedge u_s$ be one of its prime implicants. We cannot have, say, $u_1 = \bar{x}_i$, because in that case the shorter term $u_2 \wedge \cdots \wedge u_s$ would also be an implicant, by monotonicity. Therefore no prime implicant has a complemented literal.

Now if we set $u_1 \leftarrow \cdots \leftarrow u_s \leftarrow 1$ and all other variables to 0, the value of f will be 1, but all of f's other prime implicants will vanish. Thus $u_1 \wedge \cdots \wedge u_s$ must be in every shortest DNF, because every implicant of a shortest DNF is clearly prime. ▮

Corollary Q. *A disjunctive normal form is the disjunctive prime form of a monotone Boolean function if and only if it has no complemented literals and none of its implicants is contained in another.* ▮

Satisfiability. A Boolean function is said to be *satisfiable* if it is not identically zero — that is, if it has at least one implicant. The most famous unsolved problem in all of computer science is to find an efficient way to decide whether a given Boolean function is satisfiable or unsatisfiable. More precisely, we ask: Is there an algorithm that inputs a Boolean formula of length N and tests it for satisfiability, always giving the correct answer after performing at most $N^{O(1)}$ steps?

When you hear about this problem for the first time, you might be tempted to ask a question of your own in return: "What? Are you serious that computer scientists still haven't figured out how to do such a simple thing?"