2. *Subroutine program between above Steps* 9 *and* 10, 11 *and* 12, *etc.*

2.1 For each term left over after the completion of each main minimisation search, look at ALL terms prior to this last minimisation search, looking for a triplet of terms with 0, 1, and 2 in the position of any '—' variable of the left-over term.

2.2 If one or more such triplet of terms can be found, check agreement of all remaining ($n - 1$) variables of each term of each triplet with the left-over term, where (i) any additional '—' variable in the left-over term must match exactly with a '—' variable in each of the triplet terms, and (ii) all remaining 0 or 1 or 2 variables of the left-over term must match exactly with the corresponding variable in each of the triplet terms, where '—' in any triplet term is taken as equal to 0 or 1 or 2 as desired.

2.3 If such full agreement of all these ($n - 1$) variables of a triplet and the left-over term is found, the left-over term is redundant and should be deleted.

2.4 If no triplets or matching of triplet variables can be found, the left-over term is irredundant and must therefore be printed out as a necessary IRREDUNDANT PRIME IMPLICANT.

(*Note*: Having found redundant and deleted a left-over term, this term is no longer available in the list of ALL terms prior to the last minimisation, see 2.1 above. Thus if procedures 2.1 to 2.4 above have to be repeated for further left-over terms, any deleted term is no longer available in these subsequent search procedures.)

Book Review

Machine Intelligence 3. (Ed.) D. Michie, 1968; 405 pages. (*Edinburgh University Press*, 70s.)

The first attempts to get machines seeking proofs for mathematical assertions were aimed at putting mathematicians out of business and remedying Fermat's deplorable carelessness. Ten years and a few pages of college mathematics later the day nevertheless looks not far off when theorem-proving will indeed be a workaday occupation for computers in banks and universities, but not to lay Goldbach's ghost. The light is slowly dawning that even a modest inferential capacity would be an immense improvement on today's lumpen responses, and that mathematics is not the only illogical human activity that lends itself to analysis in the terms of current logic.

In five papers of this book leading participants in the theorem-proving field write about techniques for proof-seeking, especially J. A. Robinson and others about developments of Robinson's 'resolution principle'. To make a comparison with another field, their work is like the development of techniques for dealing with simultaneous linear constraints. For applications we must await the next instalment (at least). This comparison is likely to be justified when proof-seeking acquires a similarly central position to that now occupied by linear programming.

The analogy doesn't stop here. Operational research began as the ragbag for unclassifiable applications of mathematics, and it spawned linear programming as its earliest specific methodology. For OR the search for self-identification is now over (or just too boring) and its boundaries have hardened. The focus for the unclassifiable has shifted from the analysis of corporate behaviour to the analysis of individual behaviour. The mantle of OR seems to have fallen on artificial intelligence (AI), although the shift is by no means complete, witness here Varshavsky's survey of recent Russian work in Collective Behaviour and Control.

Theorem-proving started as an application-study and is becoming a foundational tool. Another less surprising such tool is the exploration of trees and graphs. Also, reaching out unsurely towards the proof-seeking techniques, is the business of formulating in logic topics that might one day yield to mechanical inference-making, for example Laski's and Park's discussions of data-structures. These shade off from logic into programming because programming languages seem destined to evolve towards logic. AI can take a lot of the credit for this. It has always excelled at spot-lighting the deficiencies that most of us merely suffer inarticulately (see Burstall's 'Alternative Expressions', Foster's 'Assertions').

As ever more tricks are found for introducing implicit forms of description instead of the explicit forms forced on us by strictly algorithmic languages, something has to be done to clean up the mess, and logicians are technology's sanitary inspectors.

If AI has inherited some of the angst of OR it has also inherited a sympton—polarisation between methodological preoccupations and undigested engineering descriptions. This book bridges the gap with some thoughtful case-studies, especially Amarel exhibiting the effect of alternative formulations of a problem (missionaries and cannibals).

One serious study for its own sake is the automatic English parser of Thorne *et al.* that does not rely on a complete dictionary of words encountered. Language processing is another application area that has won independence. Perhaps AI is destined to remain fuzzy because each study that becomes well-defined claims autonomy. Some practitioners might draw the bounds so tight as to include only one piece of work in this book—Hilditch's automatic inspection of photographs of chromosomes.

Fortunately this volume represents no such narrow view. The preface acknowledges the difficulty of assimilating 'the interconnections of such a ramifying field of subject matter'. Reading the book brings home how important doing just that is going to be as AI crystallises out into techniques and application-areas the shapes of which are not now predictable.

P. J. Landin (London)

Downloaded from https://academic.oup.com/comjnl/article/11/3/286/416405 by guest on 19 April 2024