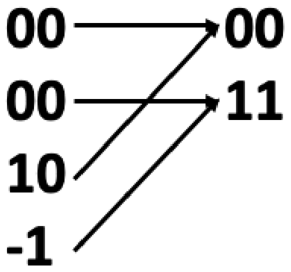
Boolean Relation Determinization

Outline

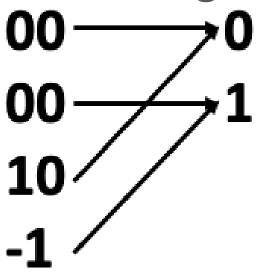
- Introduction
- Problem Formulation
- Preliminaries
- Proposed Method
- Conclusion

Introduction

- Boolean relations are more powerful at representing flexibility than functions.
- Ex:$R = \neg x_1 \neg x_2 \neg y_1 \neg y_2 + \neg x_1 \neg x_2 y_1 y_2 + x_1 \neg x_2 \neg y_1 \neg y_2 + x_2 y_1 y_2$

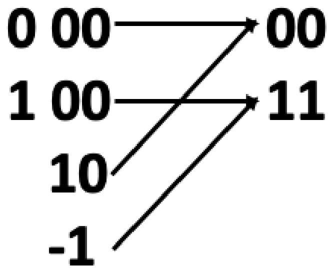


- Boolean relations are a generalization of incompletely specified functions.
- Ex: when it is a single output mapping: $F = x_2$

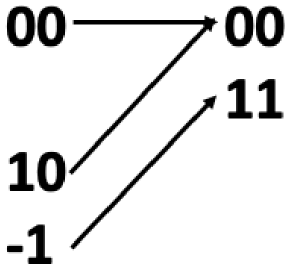


Introduction (cont'd)

- Determinize by introducing parametric parameters
- Range-preserving



- Determinize by choosing one-to-one mapping among one-to-many mappings.
- Deterministically reducing



Problem Formulation

- Given a multi-output relation **R**, where non-determinism exists, what is the minimum number of variables needed to be additionally introduced to determinize **R**?
- It is NP-Complete

Preliminaries

- Lemma1. Given a set of cubes $C_n = \{c_1, c_2, \ldots, c_n\}$ if the pairwise intersection of any two cubes, $c_i \cap c_j$, where $i \neq j$ and $i, j \leq n$, is not empty, then $c_1 \cap c_2 \cap \cdots \cap c_n$ is not empty.
- Lemma2. Maximal Clique is NP-Complete.
- Theorem1. The determinization of a boolean relation is NP-Complete
- Theorem2. Max-SAT is an NP-Complete problem: Given a CNF formula **F** and an integer **k**, is there a truth assignment that can satisfy at least **k** clauses in **F**?

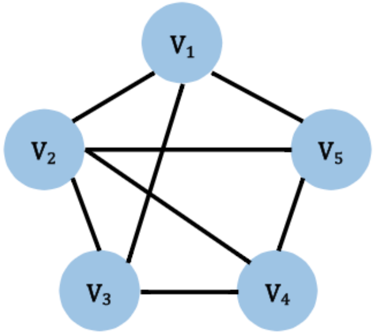
Proposed Method

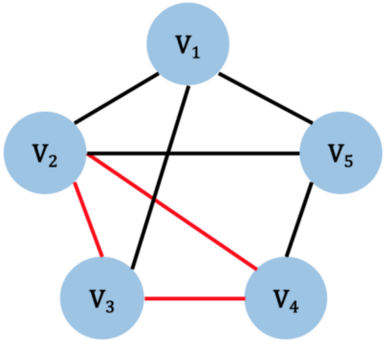
- Conversion to Undirected Graph
- Conversion to Max-SAT Problem
- Unate Recursive with Branch and Bound

Conversion to Undirected Graph

- A relation **R** can be converted into a graph **G**
  a. Expand the output part until no don't care bits
  b. Construct a vertex for each row
  c. An edge is inserted if two rows conflict
  d. Find maximum-clique Q by Bron-Kerbosch algorithm
  e. The minimum number of variables should be $log_2(|Q|)$

	x_1	x_2	y_1	y_2		x_1	x_2	y_1	y_2
r_1	-	-	0	1	r'_1	-	-	0	1
r_2	-	1	1	-	r'_2	-	1	1	0
					r'_3	-	1	1	1
r_3	0	-	-	1	r'_4	0	-	0	1
					r'_5	0	-	1	1





Conversion to Max-SAT

- A relation **R** can be converted into a CNF formula
  a. Allocate one literal ℓ for each row, and finally one dummy literal z
  b. Construct two clauses for each row $i \leq n$
     a. $(\ell_i \vee z)(\ell_i \vee \neg z)$
  c. Iterate through each row $j \leq n, i \neq j$, construct one clause if no conflict
     a. $(\neg \ell_i \vee \neg \ell_j)$
  d. The entire CNF formula should be

$$F = \prod_{\substack{1 \leq i \leq n \\ i < u, v \leq n}} (\ell_i \vee z)(\ell_i \vee \neg z)\ (\neg \ell_i \vee \neg \ell_u) \ldots (\neg \ell_i \vee \neg \ell_v),$$

  e. Feed into any MAX-SAT solver, e.g. QMaxSAT
  f. |Q| = The number of literals assigned to 1 (excepting z)
  g. The minimum number of literals = $log_2|Q|$

	x_1	x_2	y_1	y_2		x_1	x_2	y_1	y_2
r_1	-	-	0	1	r'_1	-	-	0	1
r_2	-	1	1	-	r'_2	-	1	1	0
					r'_3	-	1	1	1
r_3	0	-	-	1	r'_4	0	-	0	1
					r'_5	0	-	1	1

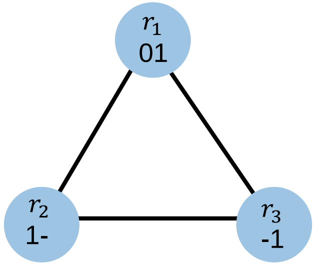
$$\begin{aligned} F = &(\ell_1 \vee z)(\ell_1 \vee \neg z)(\neg \ell_1 \vee \neg \ell_4) \\ &(\ell_2 \vee z)(\ell_2 \vee \neg z) \\ &(\ell_3 \vee z)(\ell_3 \vee \neg z)(\neg \ell_3 \vee \neg \ell_5) \\ &(\ell_4 \vee z)(\ell_4 \vee \neg z) \\ &(\ell_5 \vee z)(\ell_5 \vee \neg z), \end{aligned}$$

$$(\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, z) = (0, 1, 1, 1, 0, 0)$$

Unate Splitting with Branch and Bound

- Idea: Prevent exponential time output expansion.
- Max-Output Clique:
  - Construct graph **G** without expanding the output part
  - Find the clique in which has the maximum number of output minterm

	x_1	x_2	y_1	y_2
r_1	-	-	0	1
r_2	-	1	1	-
r_3	0	-	-	1



Unate Splitting with Branch and Bound

- **Lemma.** Given a set of cubes $C_n = \{c_1, c_2, \ldots, c_n\}$ if the pairwise intersection of any two cubes, $c_i \cap c_j$, where $i \neq j$ and $i, j \leq n$, is not empty, then $c_1 \cap c_2 \cap \cdots \cap c_n$ is not empty.

- **Theorem.** Given a set of cubes $C_n = \{c_1, c_2, \ldots, c_n\}$, C_n is unate iff $c_1 \cap c_2 \cap \cdots \cap c_n$ is not empty.

- **Theorem.** Every unate leaf of input part is a clique in the graph

Unate Splitting with Branch and Bound

- If we only split the variable that is not unate, each unate leaf will be a maximal clique in the graph
    - In each unate leaf, compute the number of output minterm
    - Find the maximum number of output minterm among all unate leaves

- Use branch and bound to skip some branches
    - Bound the branch by currently largest number
    - Each branch must less than the summation of the number of minterms in each row, which has time complexity only $O(n)$

Unate Splitting with Branch and Bound

- Example
  - This example is already unate in input
- Find the number of literals needed
  - Make the output cubes orthogonal to count the number of output minterms
  - Compute the summation of number of minterms in each cube

	x_1	x_2	y_1	y_2
r_1	-	-	0	1
r_2	-	1	1	-
r_3	0	-	-	1

Unate leaf

→

	y_1	y_2
r_1	0	1
r_2	1	-
r_3	-	1

Output cubes

→

	y_1	y_2	$size$
r_1	0	1	1
r_2	1	-	2

(Orthogonalized)

→ 3 minterms

Experimental Result

- Unate splitting method outperformed all other methods

benchmark					Graph	MaxSAT	Unate
Name	NX	NY	NM	NV	Time	Time	Time
c17_po1	2	2	3	2	1.40E-4	7.15E-3	2.63E-5
c432_po0	12	6	64	6	7.95E-2	–	1.59E-4
c432_po1	12	7	511	9	–	–	9.43E+0
3540n728	18	9	240	8	1.40E+2	–	4.30E+0
499n177	9	5	16	4	2.34E+0	–	3.16E-2
880n316	9	5	92	7	1.11E-1	–	1.38E-2
880n359	21	11	2048	12	–	–	3.61E+0

TABLE III: The number of x variables(NX), the number of y variables(NY), max number of y minterns to the same x (NM), the number of required number of additional variables(NV) and the comparisons of CPU times(secs), the notation – means that the tool crashes or could not finish in 1000 secs.