SYMBOLIC LOGIC OPTIMIZATION AND ENCODING

©*Giovanni De Micheli*

Stanford University

Outline

- Symbolic minimization.
  - Simplification of interconnected logic blocks.
  - Encoding of *finite-state machines*
- Encoding problems:
  - Input encoding.
  - Output encoding.

Symbolic minimization

- Minimize tables of *symbols* rather than binary tables.

- Extension to bvi and mvi function minimization.

- Applications:

  - Encoding of *op-codes*.

  - State encoding of *finite-state machines*

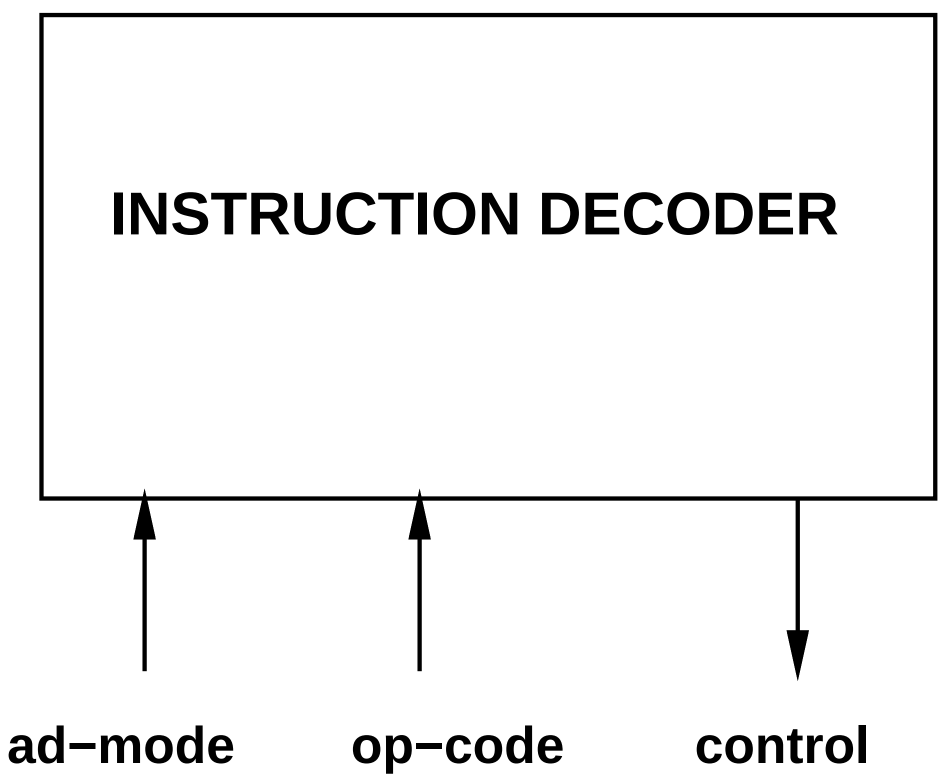
- Problems:

  - Input encoding.

  - Output encoding.

  - Mixed encoding.

Example
(input encoding)



Example

ad-mode	op-code	control
INDEX	AND	CNTA
INDEX	OR	CNTA
INDEX	JMP	CNTA
INDEX	ADD	CNTA
DIR	AND	CNTB
DIR	OR	CNTB
DIR	JMP	CNTC
DIR	ADD	CNTC
IND	AND	CNTB
IND	OR	CNTD
IND	JMP	CNTD
IND	ADD	CNTC

Definitions

- Symbolic cover:
  - List of symbolic implicants.
  - List of rows of a table.
- Symbolic implicant:
  - Conjunction of symbolic literals.
- Symbolic literals:
  - Simple: one symbol.
  - Compound: the disjunction of some symbols.

Input encoding problem
Rationale

- Degrees of freedom in encoding the symbols.

- Goal:
  - Reduce size of the representation.

- Approach:
  - Encode to minimize number of rows.
  - Encode to minimize number of bits.

Input encoding problem

- Represent each string by 1-hot codes.
- Table with positional cube notation.
- Minimize table with mvi minimizer.
- Interpret minimized table:
  - Compound mvi-literals.
  - Groups of symbols.

Example

- Encoded cover:

100	1000	1000
100	0100	1000
100	0010	1000
100	0001	1000
010	1000	0100
010	0100	0100
010	0010	0010
010	0001	0010
001	1000	0100
001	0100	0001
001	0010	0001
001	0001	0010

- Minimum cover:

100	1111	1000
010	1100	0100
001	1000	0100
010	0011	0010
001	0010	0010
001	0110	0001

Example

- Minimum symbolic cover:

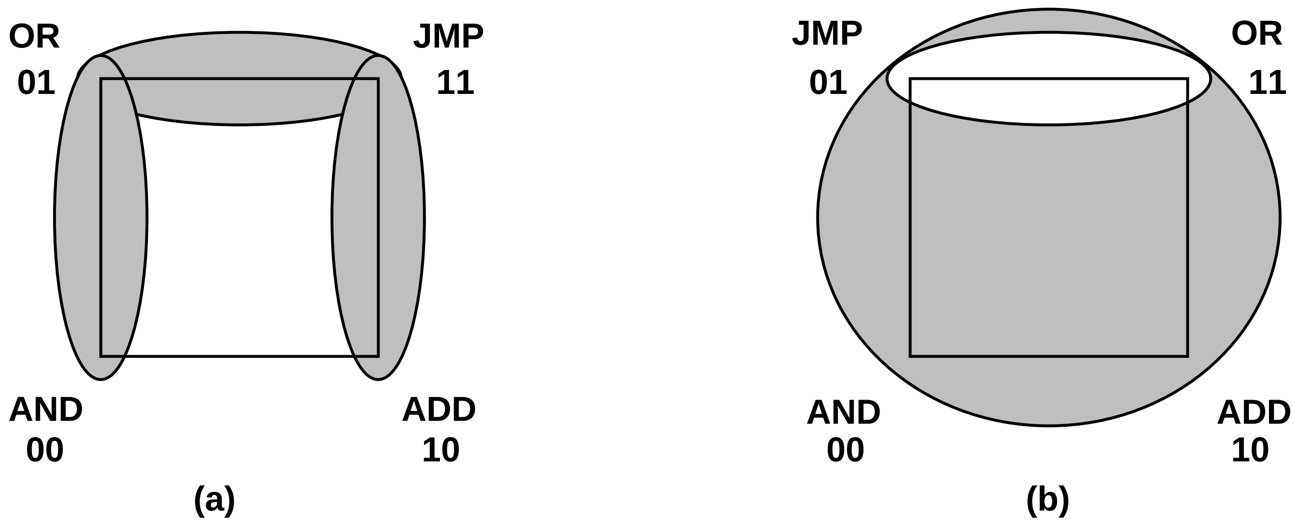
INDEX	AND,OR,JMP,ADD	CNTA
DIR	AND,OR	CNTB
IND	AND	CNTB
DIR	JMP,ADD	CNTC
IND	ADD	CNTC
IND	OR,JMP	CNTD

- Examples of:
  - Simple literal: AND
  - Compound literal: AND,OR

Input encoding problem

- Transform minimum symbolic cover into minimum bv-cover.
- Map symbolic implicants into bv implicants (one to one).
- Compound literals:
  - Encode corresponding symbols so that their supercube does not include other symbol codes.
- Replace encoded literals into cover.

Example



- Compound literals:
  - AND,OR,JMP,ADD
  - AND,OR
  - JMP,ADD
  - OR,JMP

Example

- Valid codes:

AND	00
OR	01
JMP	11
ADD	10

- Replacement in cover:

$$
\begin{array}{lcl}
1111 & \rightarrow & ** \\
1100 & \rightarrow & 0* \\
1000 & \rightarrow & 00 \\
0011 & \rightarrow & 1* \\
0010 & \rightarrow & 10 \\
0110 & \rightarrow & *1
\end{array}
$$

Input encoding algorithms

- Problem specification:
  - *Constraint matrix* **A**:
  - $a_{ij} = 1$ iff symbol j belongs to literal i.

- Solution sought for:
  - *Encoding matrix* **E**:
    * As many rows as the symbols.
    * Encoding length n_b.

Example

- Constraint matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

- Encoding matrix:

$$\mathbf{E} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Input encoding problem

- Given constraint matrix **A**
  - Find encoding matrix **E** satisfying all input encoding constraints (due to compound literals)
  - With minimum number of columns (bits).

Dichotomy theory

- Dichotomy:
  - Two sets (L, R).
  - Bipartition of a subset of the symbol set.
- Encoding:
  - Set of columns of **E**.
  - Set of bipartitions of symbol set.
- Rationale:
  - Each row of the constraint matrix implies some choice on the codes.

- *Dichotomy* associated with row $\mathbf{a}^T$ of $\mathbf{A}$:
  - A set pair (L, R):
    - L has the symbols with the 1s in $\mathbf{a}^T$
    - R has *the* symbols with the 0s in $\mathbf{a}^T$

- *Seed dichotomy* associated with row $\mathbf{a}^T$ of $\mathbf{A}$:
  - A set pair (L, R):
    - L has the symbols with the 1s in $\mathbf{a}^T$
    - R has *one* symbol with a 0 in $\mathbf{a}^T$

Example

- Dichotomy associated with constraint $\mathbf{a}^T = 1100$:
  - $(\{AND, OR\}; \{JMP, ADD\})$.

- The corresponding seed dichotomies are:
  - $(\{AND, OR\}; \{JMP\})$
  - $(\{AND, OR\}; \{ADD\})$.

Definitions

- *Compatibility:*
  - $(L_1; R_1)$ and $(L_2; R_2)$ are compatible if:
    - $L_1 \cap R_2 = \emptyset$ and $R_1 \cap L_2 = \emptyset$ or
    - $L_1 \cap L_2 = \emptyset$ and $R_1 \cap R_2 = \emptyset$.
- *Covering:*
  - Dichotomy (L_1, R_1) covers (L_2, R_2) if:
    - $L_1 \supseteq L_2$ and $R_1 \supseteq R_2$ or
    - $L_1 \supseteq R_2$ and $R_1 \supseteq L_2$.
- *Prime dichotomy:*
  - Dichotomy that is not covered by any compatible dichotomy of a given set.

Exact input encoding

- Compute all prime dichotomies.
- Form a prime/seed table.
- Find minimum cover of seeds by primes.

Example

- Seed dichotomies:

$$
\begin{array}{c|lcl}
s_1 & (\{\ \mathsf{AND,OR}\} & ; & \{\ \mathsf{JMP}\ \}) \\
s_2 & (\{\ \mathsf{AND,OR}\} & ; & \{\ \mathsf{ADD}\ \}) \\
s_3 & (\{\ \mathsf{JMP,ADD}\} & ; & \{\ \mathsf{AND}\ \}) \\
s_4 & (\{\ \mathsf{JMP,ADD}\} & ; & \{\ \mathsf{OR}\ \}) \\
s_5 & (\{\ \mathsf{OR,JMP}\} & ; & \{\ \mathsf{AND}\ \}) \\
s_6 & (\{\ \mathsf{OR,JMP}\} & ; & \{\ \mathsf{ADD}\ \})
\end{array}
$$

- Prime dichotomies:

$$
\begin{array}{c|lcl}
p_1 & (\{\ \mathsf{AND,OR}\} & ; & \{\ \mathsf{JMP,ADD}\ \}\ ) \\
p_2 & (\{\ \mathsf{OR,JMP}\} & ; & \{\ \mathsf{AND,ADD}\ \}\ ) \\
p_3 & (\{\ \mathsf{OR,JMP,ADD}\} & ; & \{\ \mathsf{AND}\ \}\ ) \\
p_4 & (\{\ \mathsf{AND,OR,JMP}\} & ; & \{\ \mathsf{ADD}\ \}\ )
\end{array}
$$

Example

- Table:

	s_1	s_2	s_3	s_4	s_5	s_6
p_1	1	1	1	1	0	0
p_2	0	0	0	0	1	1
p_3	0	0	1	0	1	0
p_4	0	1	0	0	0	1

- Minimum cover:
  - p_1 and p_2.

- Encoding:

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Heuristic encoding

- Determine dichotomies of rows of **A**.

- Column-based encoding:

  - Construct **E** column by column.

- Iterate:

  - Determine maximum compatible set.

  - Find a compatible encoding.

  - Use it as column of **E**.

Example

- Dichotomies:

$$\begin{array}{c|llll}
d_1 & (\; \{ \text{ AND,OR } \} & ; & \{ \text{ JMP,ADD } \} \;) \\
d_2 & (\; \{ \text{ JMP,ADD } \} & ; & \{ \text{ AND,OR } \} \;) \\
d_3 & (\; \{ \text{ OR,JMP } \} & ; & \{ \text{ AND,ADD } \} \;)
\end{array}$$

- First two dichotomies are compatible.

- Encoding column $[1100]^T$ satisfies d_1, d_2.

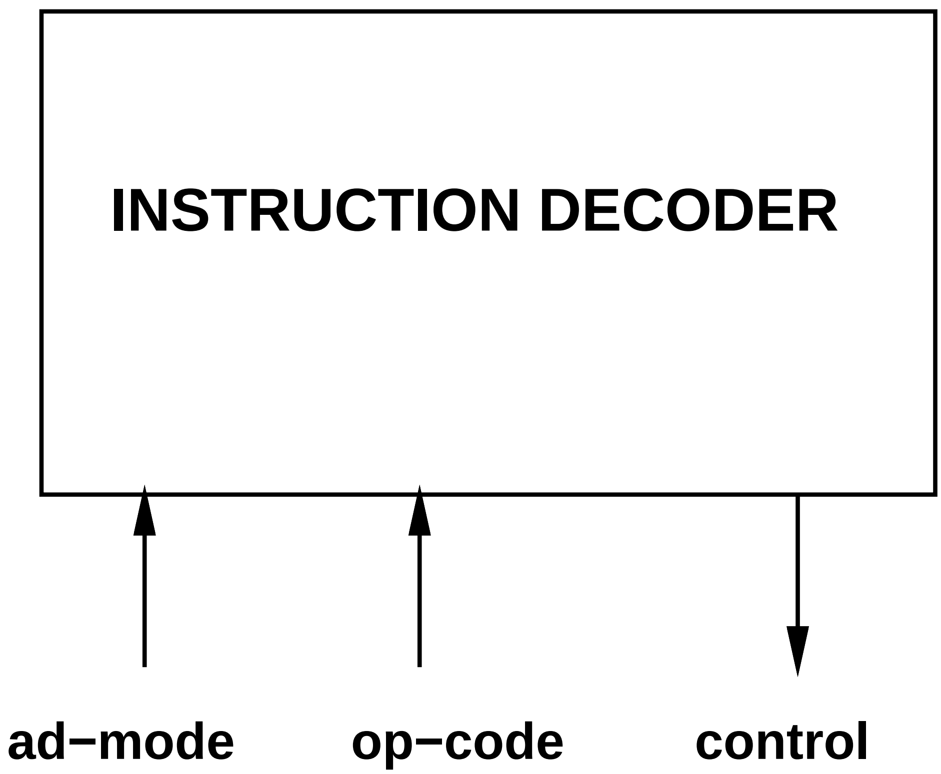
- Need to satisfy d_3.

- Second encoding column $[0110]^T$.

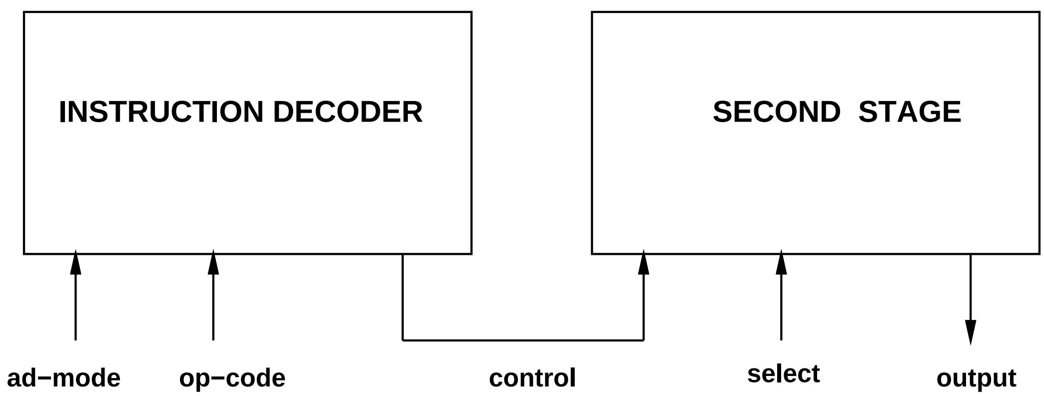
Output and mixed encoding

- Output encoding:
  - Determine encoding of output symbols.
- Mixed encoding:
  - Determine both input and output encoding
  - Examples:
    - Interconnected circuits.
    - Circuits with feedback.

Example



Example


INSTRUCTION DECODER
SECOND STAGE
ad−mode
op−code
control
select
output

Symbolic minimization

- Extension to mvi-minimization.

- Accounts for:

  - *Covering* relations.

  - *Disjunctive* relations.

- Exact and heuristic minimizers.

Example

- Minimum symbolic cover computed before:

INDEX	AND,OR,JMP,ADD	CNTA
DIR	AND,OR	CNTB
IND	AND	CNTB
DIR	JMP,ADD	CNTC
IND	ADD	CNTC
IND	OR,JMP	CNTD

- Can we use fewer implicants?

- Can we merge implicants?

Example
covering relations

- Assume the code of $CNTD$ covers the codes of $CNTB$ and $CNTC$.

100	1111	CNTA
011	1100	CNTB
011	0011	CNTC
001	0110	CNTD

- Possible codes:
  - $CNTA = 00$, $CNTB = 01$, $CNTC = 10$ and $CNTD = 11$.

Example
disjunctive relations

- Assume the code of $CNTD$ is the *or* of the codes of $CNTB$ and $CNTC$.

100	1111	CNTA
010	1100	CNTB
010	0011	CNTC
001	1110	CNTB
001	0111	CNTC

- Possible codes:
  - $CNTA =$ 00, $CNTB =$ 01, $CNTC =$ 10 and $CNTD =$ 11.

Output encoding algorithms

- Often solved in conjunction with input encoding.

- Exact algorithms:

  - Prime dichotomies compatible with output constraints.

  - Construct prime/seed table.

  - Solve covering problem.

- Heuristic algorithms:

  - Construct **E** column by column.

Example

- Input constraint matrix of second stage:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

- Output constraint matrix of first stage:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

- Assume the code of $CNTD$ covers the codes of $CNTB$ and $CNTC$.

Example

- Seed dichotomies associated with **A**

s_1	({ CNTA,CNTB }	;	{ CNTC })
s_2	({ CNTA,CNTB }	;	{ CNTD })
s_3	({ CNTC }	;	{ CNTA,CNTB }
s_4	({ CNTD }	;	{ CNTA,CNTB }
s_5	({ CNTB,CNTD }	;	{ CNTA })
s_6	({ CNTB,CNTD }	;	{ CNTC })
s_7	({ CNTA }	;	{ CNTB,CNTD }
s_8	({ CNTC }	;	{ CNTB,CNTD }

- Seed dichotomies s_2, s_7 and s_8 are not compatible with **B**.

Example (2)

- Prime dichotomies compatible with **B** :

p_1	({ CNTC,CNTD }	;	{ CNTA,CNTB })
p_2	({ CNTB,CNTD }	;	{ CNTA,CNTC })
p_3	({ CNTA,CNTB,CNTD }	;	{ CNTC })

- Cover: p_1 and p_2

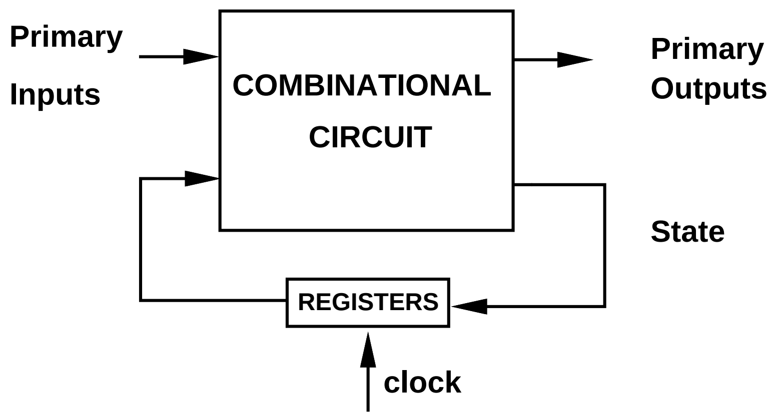
- Encoding matrix:

$$\mathbf{E} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

State encoding of *finite-state machines*

- Given a state table of a *finite-state machine*
  - With symbols representing:
    - present-states.
    - next-states.
- Find a consistent encoding of the states
  - That minimizes the size of the cover.
  - With minimum number of bits.

Example



INPUT	P-STATE	N-STATE	OUTPUT
0	s_1	s_3	0
1	s_1	s_3	0
0	s_2	s_3	0
1	s_2	s_1	1
0	s_3	s_5	0
1	s_3	s_4	1
0	s_4	s_2	1
1	s_4	s_3	0
0	s_5	s_2	1
1	s_5	s_5	0

Example

- Minimum symbolic cover:

$$\begin{array}{llll} * & s_1 s_2 s_4 & s_3 & 0 \\ 1 & s_2 & s_1 & 1 \\ 0 & s_4 s_5 & s_2 & 1 \\ 1 & s_3 & s_4 & 1 \end{array}$$

- Covering constraints:
  - s_1 and s_2 cover s_3
  - s_5 is covered by all other states.
- Encoding constraint matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Example

- Encoding matrix (one row per state):

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Encoded cover of combinational component:

```
*  1**  001  0
1  101  111  1
0  *00  101  1
1  001  100  1
```

Summary

- Symbolic minimization:
  - Reduce size of tabular representations where symbols in table can be encoded.
- Requires solving encoding problems:
  - Find minimum-length encoding that is valid for a minimum symbolic cover.
- Applicable to optimizing:
  - Interconnected combinational blocks.
  - Combinational part of *finite-state machines*