Functional Decomposition of MVL Functions using Multi-Valued Decision Diagrams

Craig Files[1], Rolf Drechsler[2], Marek A. Perkowski[1]

[1]Department of Electrical Engineering
Portland State University

[2]Institute of Computer Science
Albert-Ludwigs-University

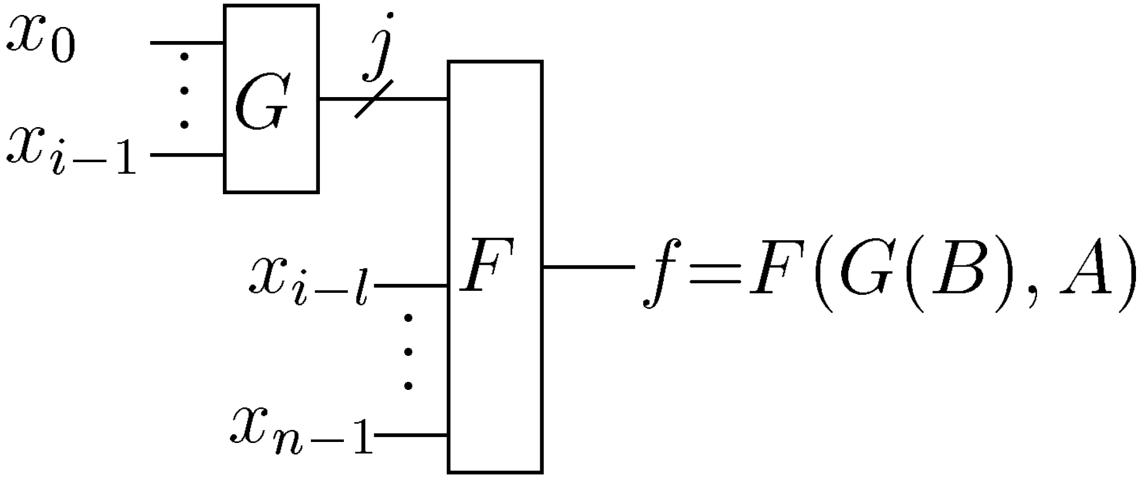
Introduction

- Functional decomposition for multi-valued logic
  - Multi-valued inputs/outputs
  - Machine learning (99.9% don't cares)
- Multi-valued decision diagrams

Generalized Functional Decomposition

A function $f(x_0, x_1, \ldots, x_{n-1})$ is decomposable under *bound set* $\{x_0, \ldots, x_{i-1}\}$ and *free set* $\{x_{i-l}, \ldots, x_{n-1}\}$, $0 < i < n, 0 \leq l$ if and only if f can be represented as the composite function $F(G_0(x_0, \ldots, x_{i-1})$, $\ldots$, $G_{j-1}(x_0, \ldots, x_{i-1})$, $x_{i-l}, \ldots, x_{n-1})$, where $0 < j < i - l$. If l equals 0 then f is said to be *disjunctively decomposable*, otherwise, it is known as *non-disjunctively decomposable.*

Curtis Decomposition

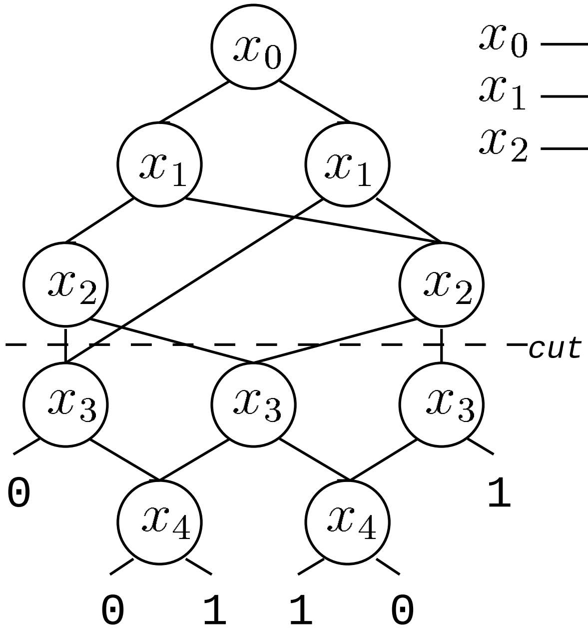


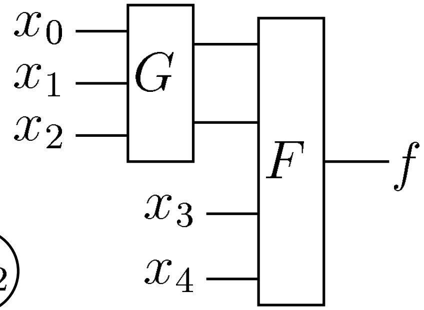
Bound set B $= \{x_0, \ldots, x_{i-1}\}$

Free set A $= \{x_{i-l}, \ldots, x_{n-1}\}$

Boolean Decomposition Example

x_3x_4 \ $x_0x_1x_2$								
	0	0	0	1	0	0	0	1
	0	1	1	0	0	0	1	0
	0	1	1	1	0	0	1	1
	1	0	0	1	1	1	0	1
	A	B	B	C	A	A	B	C

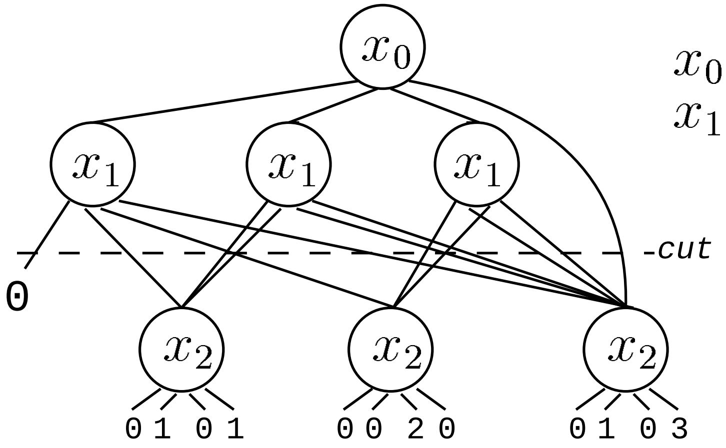


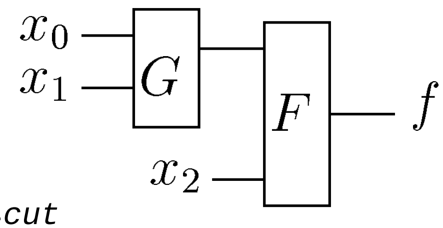


The number of distinct columns is three, thus f is represented as a composite function with $\lceil log_2 3\rceil$=2, G functions. The resulting decomposition is $F(G_0(x_0,x_1,x_2),G_1(x_0,x_1,x_2),x_3,x_4)$.

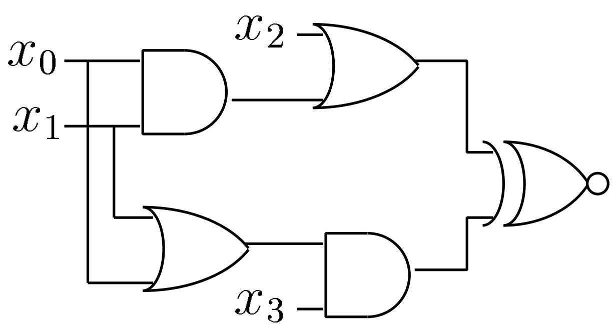
Multi-Valued Decomposition

x_2 \ x_0x_1	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1
2	0	0	2	0	0	0	0	0	2	0	2	0	0	0	0	0
3	0	1	0	3	1	1	3	3	0	3	0	3	3	3	3	3
	A	B	C	D	B	B	D	D	C	D	C	D	D	D	D	D





Don't Cares In Decomposition



x_0x_1 (columns), x_2x_3 (rows)

0	0	0	1
1	1	1	1
0	1	1	0
1	0	0	0
A	B	B	C

G_0G_1 (columns), x_2x_3 (rows)

0	0	X	1
1	1	X	1
0	1	X	0
1	0	X	0

x_3G_1 (columns), x_2G_0 (rows)

0	1	X	X	B
0	1	1	1	B
0	1	X	X	B
1	0	0	0	A
A	B	B	B	

Decomposed Function Cardinality(DFC)

- Abu-Mostafa theory for binary functions
- A function $f(x_0, x_1, \ldots, x_{n-1})$ with k valued logic and m outputs, has *cardinality* $k^n * m$. DFC is defined as the sum of the function cardinality of each decomposed partial block.
- DFC may be used to find the minimum combined component cardinality of all the decompositions of a function.
- It is also a “good” figure of merit for evaluating function implementations.

DFC Example

Given the multi-level function
$f{=}F(G_0(x_0,x_1),G_1(x_2,x_3),x_4,x_5)$.

- f has six primary inputs, is 4-valued, and has 1 output.
- cardinality of f is $4^6 * 1{=}4096$.
- DFC of f is $4^2 + 4^2 + 4^{(1+1+2)} = 288$.

Algorithm

```
int decompose(function f) {
  if (Num_vars<3)  then return cardinality of f;
  evaluate all possible partitions;
  // given partition with smallest DFC
  // create F and G0 through Gn-1
  DFC=decompose(F);
  for i=0 to (n-1)
    DFC=DFC+decompose(G(i));
  // if function is non-decomposable
  // return cardinality of the f
  if (DFC < cardinality of f) return DFC;
  else  return cardinality of f;
}
```

Results

name	in	out	k	card.	DFC	time
balance	4	1	5	625	375	0.1
breastc	9	1	10	10^9	7025	207
flare1_1	10	1	7	7^{10}	1797	7
flare1_2	10	1	7	7^{10}	3512	8
flare1_3	10	1	7	7^{10}	1751	8
flare2_1	10	1	7	7^{10}	1413	7
flare2_2	10	1	7	7^{10}	2867	9
flare2_3	10	1	7	7^{10}	1751	8
hayes	4	1	5	625	314	0.1

Conclusions

- MDD data structure
- Multi-level minimization with don't cares

Future

- encoding the MDD onto a BDD