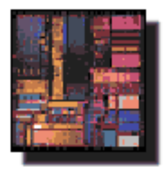
Digital Systems Design Automation

Unit 2: Advanced Boolean Algebra

Lecture 2.4: Conversion of Boolean Function Representations



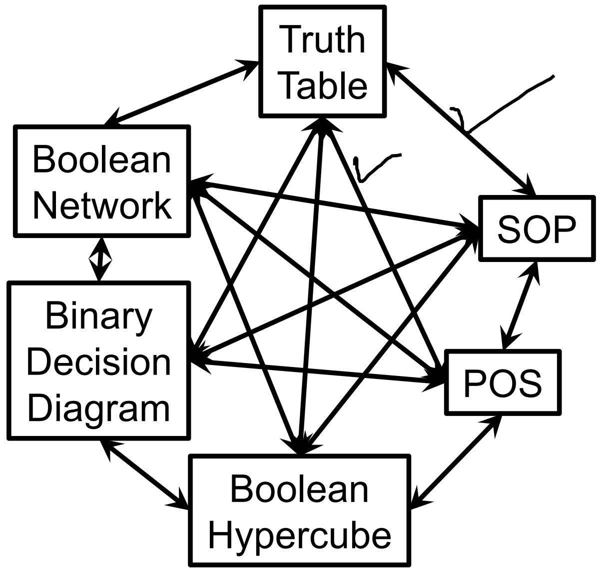
Anand Raghunathan

raghunathan@purdue.edu

Outline

Converting Between Boolean Function Representations

- All of the previously described representations are functionally equivalent…
- But vary in their complexity (size), and ease of performing various operations
  - No single “best” representation
- Need to convert between representations



Conversion: Example #1

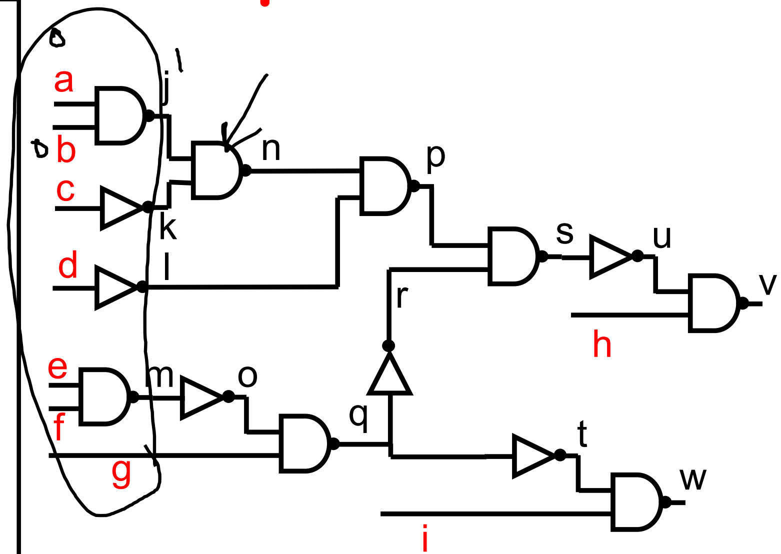
- How do you convert a general Boolean network (multi-level circuit) into SOP form?
  - Quick-and-dirty (exhaustive) algorithm

```
For each input vector v ∈{00…0 to 11…1} {
    Simulate the circuit for input v;
    If (output == 1) {
        encode input vector as a minterm;
    }
}
```

  - Works, but *guaranteed to be exponential* in the number of inputs
- There should be a better algorithm!

Conversion: Example #1

- j = (ab)’ = a’ + b’
- k = c’
- l = d’
- m = e’ + f’
- n = j’ + k’ = (a’+b’)’ + (c’)’ = ab + c
- o = m’ = ef
- p = n’ + l’ = (ab+c)’ + d = a’c’ + b’c’ + d
- q = o’+g’ = e’ + f’ + g’
- r = q’ = efg
- s = p’+r’ = (a+c)(b+c)d’ + e’ + f’ + g’ = abd’ + cd’ + e’ + f’ + g’
- t = q’ = efg
- u = s’ = (a’+b’+d)(c’+d)efg = a’c’efg + b’c’efg + defg
- v = u’ + h’ = abd’ + cd’ + e’ + f’ + g’ + h’
- w = t’ + i’ = e’+f’+g’+i’



Notice the similarity to circuit simulation? Only difference is, we are propagating Boolean expressions, not 0/1 values. This is called **Symbolic Simulation**

Conversion: Example #2

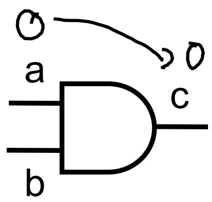
- How do you convert a general Boolean network (multi-level circuit) into a Boolean formula that is linear in the circuit size?
  - Size(formula) = O(M) where M = no. of gates in the circuit
  - SOP may be exponential in the worst case
  - Hints
    - Use variables to represent intermediate signals in the circuit
    - Compose the formula using a 1 : 1 mapping from each gate in the circuit into a piece of the formula

Converting a Boolean Circuit into a CNF Formula

- First, let us see how very simple circuits (single gates) can be expressed as a Boolean formula (in CNF form)

$x \Rightarrow y$

$(x'+y)$



b

∧	0	1
a = 0	0	0
a = 1	0	1

$c = ab$

$\Downarrow$

$c \to ab,\ ab \to c$

$\Downarrow$

$c \to a,\ c \to b,\ ab \to c$

$a = 0 \to c = 0$

$b = 0 \to c = 0$

$a = 1,\ b = 1 \to c = 1$

$\Downarrow$

$a' \to c',\ b' \to c',\ ab \to c$

$\boxed{(a+c')(b+c')(a'+b'+c)}$

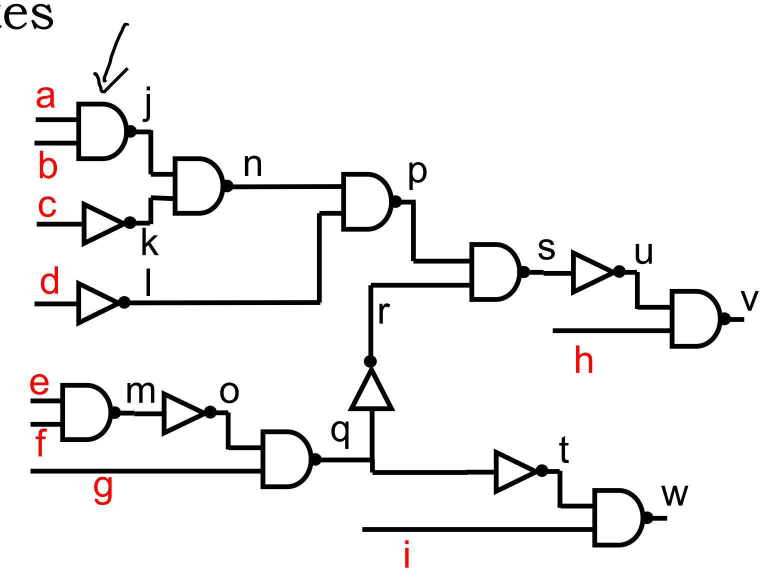
Converting a Boolean Circuit into a CNF Formula

- Rules for converting various basic gates into CNF equivalent

Gate Type	Function	CNF Formula
NOT	c = a’	(a+c)(a’+c’)
AND	c=ab	(a+c’)(b+c’)(a’+b’+c)
NAND	c=a’+b’	(a+c)(b+c)(a’+b’+c’)
OR	c=a+b	(a’+c)(b’+c)(a+b+c’)
NOR	c = a’b’	(a’+c’)(b’+c’)(a+b+c)

Converting a Boolean Circuit into a CNF Formula

- Now, we are ready to convert a multi-level circuit into a CNF formula
  - Conjunction (AND) of formulae representing each of its gates



Known as the Tseitin Transformation

(a+j)(b+j)(a'+b'+j')
(c+k)(c'+k')
(d+l)(d'+l')
(e+m)(f+m)(e'+f'+m')
(m+o)(m'+o')
(j+n)(k+n)(j'+k'+n')
(n+p)(l+p)(n'+l'+p')
(o+q)(g+q)(o'+g'+q')
(q+r)(q'+r')
(p+s)(r+s)(p'+r'+s')
(s+u)(s'+u')
(u+v)(h+v)(u'+h'+v')
(q+t)(q'+t')
(t+w)(i+w)(t'+l'+w')

Terminology Checklist

- Boolean Algebra
- Boolean Function
- Tautology
- Satisfiable / Un-satisfiable
- Cube
- Implicant (of a function)
- Minterm
- Cover (of a function)
- Sum-of-products
- Minterm canonical representation
- Product-of-sums
- Conjunctive Normal Form
- Disjunctive Normal Form
- Binary Decision Tree
- Binary Decision Diagram
- Symbolic Simulation
- Tseitin Transformation