

Two-Level Boolean Minimizer BOOM-II

Petr Fišer, Hana Kubátová

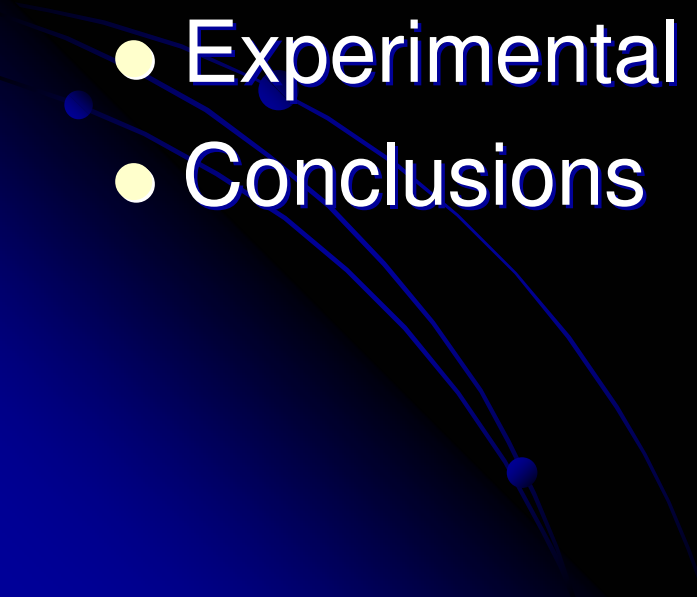
Department of Computer Science and
Engineering

Czech Technical University

Karlovo nám. 13, 121 35 Prague 2

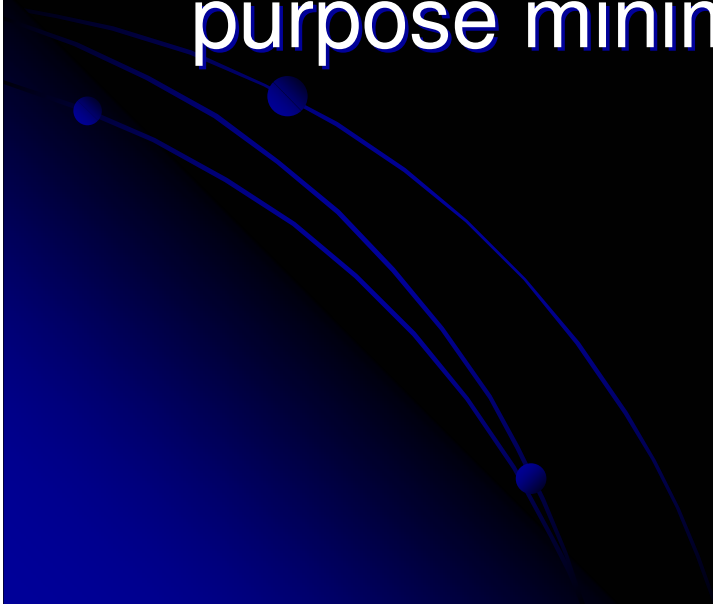
e-mail: fiserp@fel.cvut.cz, kubatova@fel.cvut.cz

Outline

- Introduction
 - Problem Statement
 - Description of BOOM-II
 - The Individual Phases
 - Experimental Results
 - Conclusions
- 

Introduction

- BOOM-II is a heuristic multiple-output two-level Boolean minimizer
- Composition of two minimizers – a general purpose minimizer



Problem Statement

Given:

n -input, m -output function given by a truth table (PLA)

$$F_1(x_1, x_2, \dots, x_n), F_2(x_1, x_2, \dots, x_n), \dots, F_m(x_1, x_2, \dots, x_n)$$

The function is specified by on-set and off-set

Our Aim:

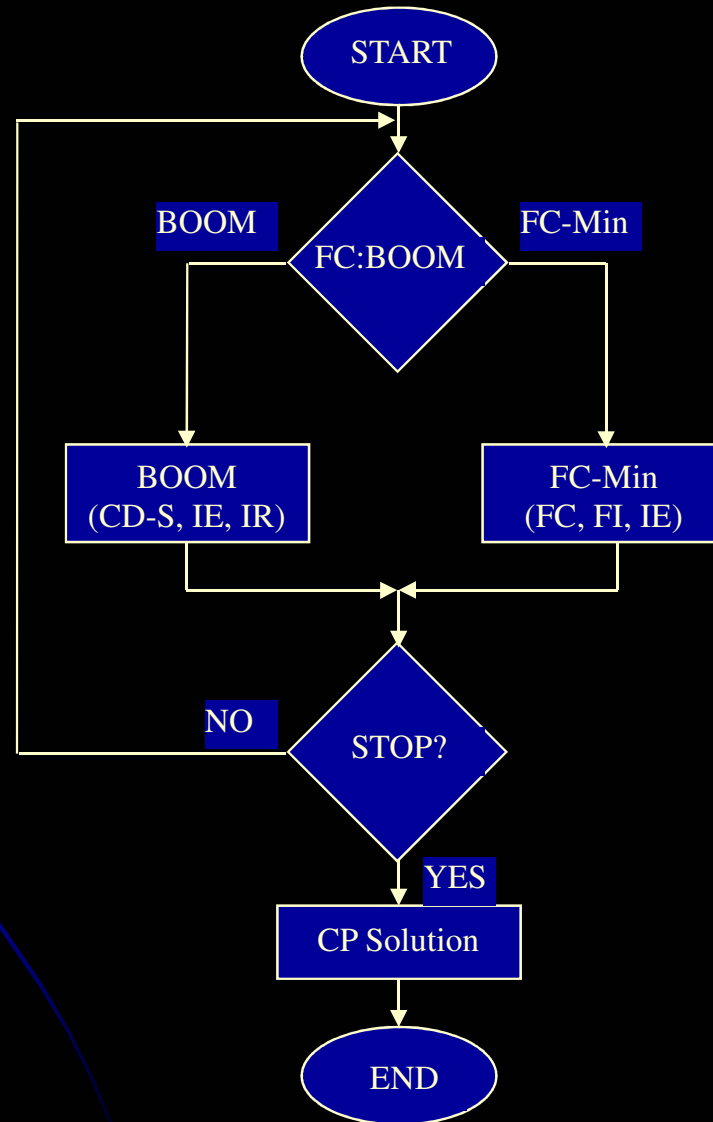
to minimize it, the result is a set of SOP forms

BOOM-II

Composition of BOOM and FC-Min

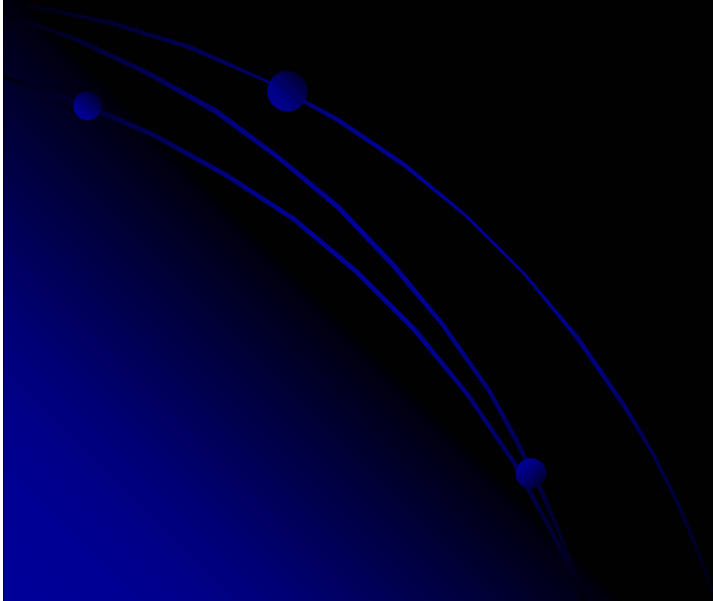
- BOOM is good for functions with many inputs
- FC-Min is good for functions with many outputs
- Iterative minimization – both the minimizers are being alternated

BOOM-II



BOOM

- CD-Search
- Implicant Expansion (IE)
- Implicant Reduction (IR)
- CP Solution (CP)



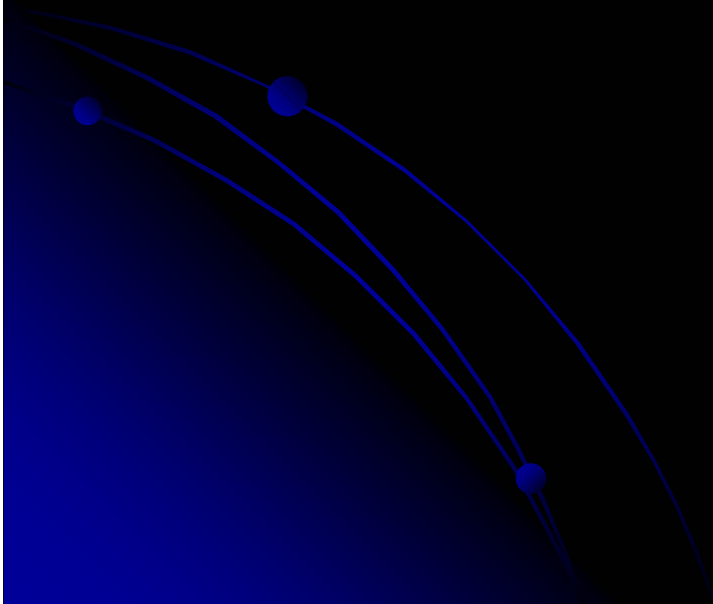
BOOM – CD-Search

- Just for single-output functions – the multiple-output function has to be divided
- Implicants are generated top-down – by reducing the universal hypercube
- We add literals to a term, until it becomes an implicant
- Based on a frequency of occurrence in on-set

BOOM - IE

Implicant Expansion

The implicants from CD-search are expanded into PIs



BOOM - IR

Implicant Reduction

Reduces PIs into group implicants

Very time-consuming for functions with many outputs



BOOM - CP

Covering problem solution

Selects an irredundant set of implicants covering the on-set

greedy heuristic is used

A decorative graphic in the bottom-left corner of the slide. It consists of three blue dots of varying sizes, connected by thin blue lines that curve upwards and to the right. The background of this area is a gradient from black to a dark blue.

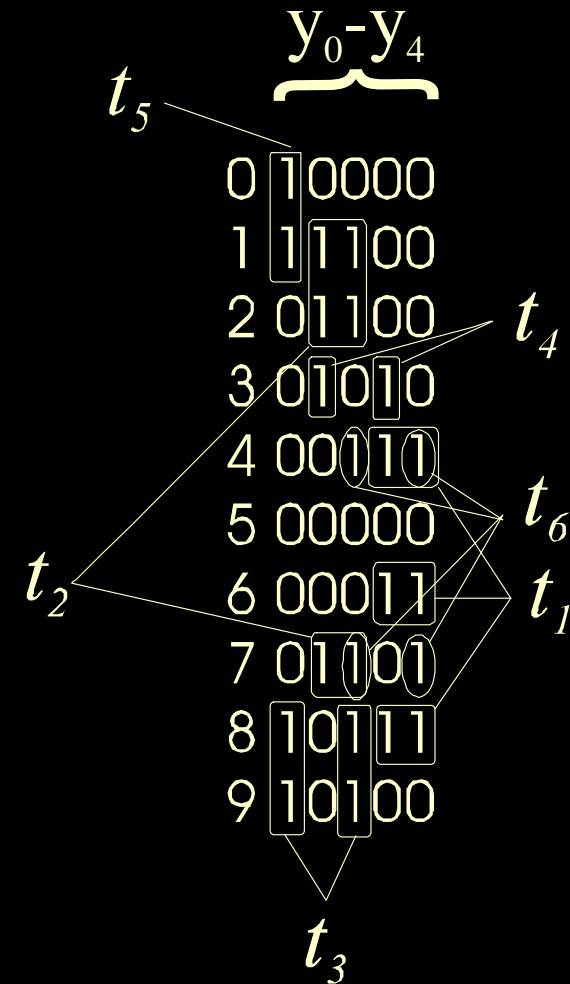
FC-Min

- Completely different approach to minimization
- First, the cover of the on-set is found, then the implicants are generated for this cover
- No PIs are produced, just the group implicants
- Generates *only* the necessary set of group implicants – fast, low memory demands

- Find-Coverage
- Generate Implicants
- Expand Implicants

FC-Min – Find Coverage

- Generates rectangle cover of the on-set
- Determines the number of product terms in the solution, not their structure
- Independent on literals



FC-Min – Find Implicants Phase

Main Idea:

When a term (cube) should cover a particular output vector, the corresponding input vector must be contained in this cube

⇒ Thus the *minimum term* for t_i must be constructed as a minimum ***supercube*** of all the input vectors corresponding to rows of t_i

FC-Min – Find Implicants Phase

PLA:

0	11010	10000
1	10000	11100
2	01001	01100
3	01111	01010
4	00110	00111
5	01110	00000
6	10110	00011
7	00001	01101
8	10101	10111
9	11100	10100

t_1 covers 4, 6 and 8

00110
10110
10101
<hr/>
-01--

FC-Min – Find Implicants Phase

All the implicants:

t_1 : -01-- 00011
 t_2 : --00- 01100
 t_3 : 1-10- 10100
 t_4 : 01111 01010
 t_5 : 1-0-0 10000
 t_6 : 00--- 00101

SOP Forms:

$$y_0 = t_3 + t_5 = x_0 x_2 x_3' + x_0 x_2' x_4'$$

$$y_1 = t_2 + t_4 = x_2' x_3' + x_0' x_1 x_2 x_3 x_4$$

$$y_2 = t_2 + t_3 + t_6 = x_2' x_3' + x_0 x_2 x_3' + x_0' x_1'$$

$$y_3 = t_1 + t_4 = x_1' x_2 + x_0' x_1 x_2 x_3 x_4$$

$$y_4 = t_1 + t_6 = x_1' x_2 + x_0' x_1'$$

FC-Min – Expand Implicants

- The *minimum implicants* can be further expanded
- Literals are removed from the obtained PLA matrix, until some term intersects off-set



BOOM-II

- Most of the phases are randomized heuristics
- Thus, repeated runs of the minimization process could yield different solutions
- We repetitively run the minimizers, while alternating them and put all the implicants into a common pool
- Then we solve CP
- The ratio of FC-Min & BOOM runs can be adjusted – according to the nature of the problem

Experimental Results

Random problem:

100 inputs

15 outputs

125 care terms

	Literals	Oc	Terms	Speedup
ESPRESSO	657	700	110	1
BOOM	765	359	122	52
BOOM-II (1:1)	674	377	102	69
FC-Min	650	374	99	137

Experimental Results

Random problem:

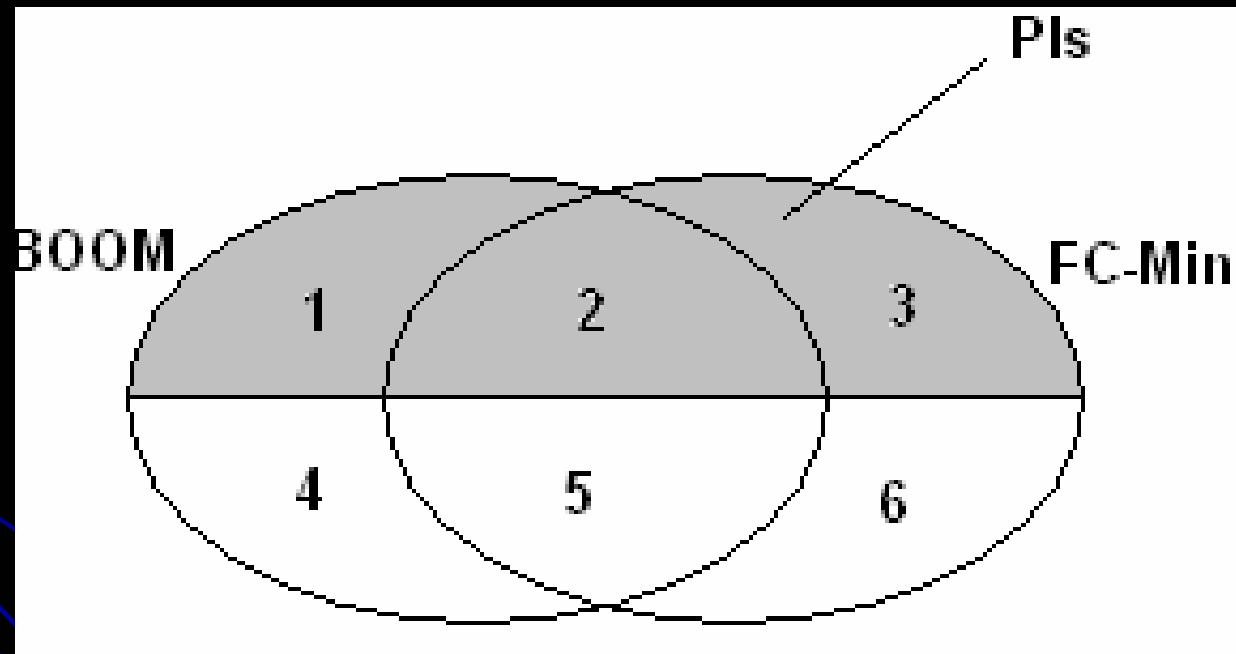
25 inputs

1 output

500 care terms

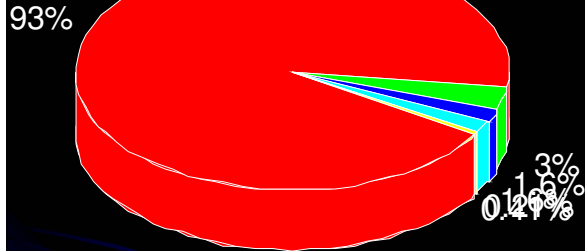
	Literals	Oc	Terms	Speedup
ESPRESSO	881	111	111	1
BOOM	793	98	98	33
BOOM-II (1:1)	852	106	106	19
FC-Min	981	124	124	15

Structure of the Solution

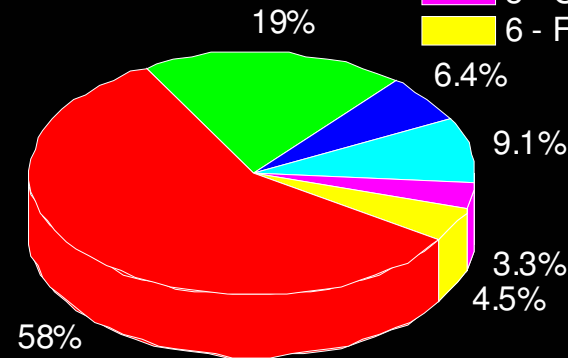
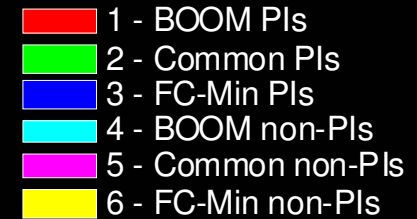


Structure of the Solution

All implicants



Solution implicants



- BOOM is better for PIs
- FC-Min is better for group implicants
- BOOM IR phase can be omitted

Conclusions

- BOOM-II has been presented
- Combination of two different approaches to implicant generation
- Universal Boolean minimizer
- Very scalable (tradeoff between solution quality & runtime)
- Useful for extremely large problems, very fast
- Can be downloaded from
<http://service.felk.cvut.cz/vlsi/prj/BOOM>