A short introduction to Espresso

Silvia Chiusano
Politecnico di Torino
<Silvia.Chiusano@polito.it>

Index

1. Introduction

The present document summarizes a subset of the Espresso functionalities. Please refer to the manual in Appendix for more details.

2. Espresso

The purpose of Espresso minimization program is to find a logically equivalent set of product-terms to represent the ON-set and optionally minterms that lie in the DC-set, without containing any minterms of the OFF-set.

Espresso takes as input a two-level representation of a two-valued (or multiple-valued) Boolean function, and produces a minimal equivalent representation. The algorithms used are new and represent an advance in both speed and optimality of solution in heuristic Boolean minimization.

Espresso reads the file provided, performs the minimization, and writes the minimized result to standard output. Espresso automatically verifies that the minimized function is equivalent to the original function.

3. Input Format

The format of the Espresso input file is compatible with the Berkeley standard format for the physical description of a PLA. The Espresso input file has extension *.pla*

Example. The input file ex1.pla is the following

```
.i 3
.o 3
000 000
001 000
010 -00
011 001
100 010
101 011
110 100
111 110
.e
```

where

- `.i 3`, specifies the number of input variables
- `.o 3`, specifies the number of output variables

- `.e` , specifies the end of file.

The Boolean equation is represented using the *PLA matrix formalism*; each row includes:

- an input part which corresponds to the *input plane* of the PLA
- an output part which corresponds to the *output plane* of the PLA.

In the PLA matrix, starting from the left hand side, the first columns belong to the input plane and the remaining columns to the output plane. The number of columns belonging to the input and output plane is specified in the *.pla* file using the keywords `.i` and `.o`. In the example, the first three columns correspond to the input plane and the remaining three to the output plane.

```
.i 3
.o 3
000 000
001 000
010 -00
011 001
100 010
101 011
110 100
111 110
.e
```

4. Main commands

4.1. Minimize a function and return it in cubic notation

espresso *inputfile*

- It minimizes the input function and generates an equivalent representation with a lower cost. The minimized function is returned in cubic notation.

Example. In the case of ex1.pla, the minimized function computed with Espresso is the following:

```
.i 3
.o 3
.p 5
   011 001
```


101 001
11- 100
10- 010
1-1 010
.e

4.2. Minimize a function and return it in algebraic notation

espresso –o eqntott *inputfile*

- With the option "–o eqntott" the minimized function is returned in algebraic notation (See Slide 24 in Lecture 3.2).

Example. In the case of ex1.pla, the minimized function is

```
.i 3
.o 3
.p 5
011 001
101 001
11- 100
10- 010
1-1 010
.e
```

and its representation in algebraic notation, obtained with the option "–o eqntott", is the following:

v3.0 = (v0&v1);
v3.1 = (v0&!v1) | (v0&v2);
v3.2 = (!v0&v1&v2) | (v0&!v1&v2);

Espresso uses the following notations:

- The input variables are labeled `vj`, where `j`∈`[0, i-1]` and `i` is the number of input variables. In the input plane the column in the left-hand side is labeled `v0`, the adjacent column `v1` and so on.
- The output variables are labeled `vi.k`, where `k`∈`[0, o-1]`, `i` is the number of input variables, and `o` the number of output variables. In the output plane the

column in the left-hand side is labeled `vi.0`, the adjacent column `vi.1` and so on.

- The basic Boolean functions are expressed as

Basic Boolean Function	Notation in Espresso	Example
AND	`&`	`(v0&v2)`
OR	`|`	`(v0|v2)`
NOT	`!`	`!v0`

In the case of the ex1.pla example, the cubic and algebraic notations for the input and output planes are the followings:

Input plane		Output plane	
Cubic notation	**Algebraic notation**	**Cubic notation**	**Algebraic notation**
`v0 v1 v2`		`v3.0 v3.1 v3.2`	
`0 1 1`	`(!v0&v1&v2)`	`0 0 1`	`v3.2 = (!v0&v1&v2) | (v0&!v1&v2);`
`1 0 1`	`(v0&!v1&v2)`	`0 0 1`	
`1 1 -`	`(v0&v1)`	`1 0 0`	`v3.0 = (v0&v1);`
`1 0 -`	`(v0&!v1)`	`0 1 0`	`v3.1 = (v0&!v1) | (v0&v2);`
`1 - 1`	`(v0&v2)`	`0 1 0`	

5. Examples

Lecture 6.4 contains several examples related to the usage of Espresso.

6. Source code availability

The Espresso source code is available at the Berkeley ftp site:

ftp://ic.eecs.berkeley.edu/pub/Espresso/

