Symmetry in 3-valued Functions

Benjamin Drucker, Craig Files, and Peter Siracusa
Department of Electrical Engineering
Portland State University
Portland, OR 97207-0751, USA
EE 572/672 Final Exam

1 Question 1

Remember the symmetric switch diagram from which we started? It was 2-dimensional for 2-valued logic. Create such diagram, in 3 dimensions, for 3-valued logic.

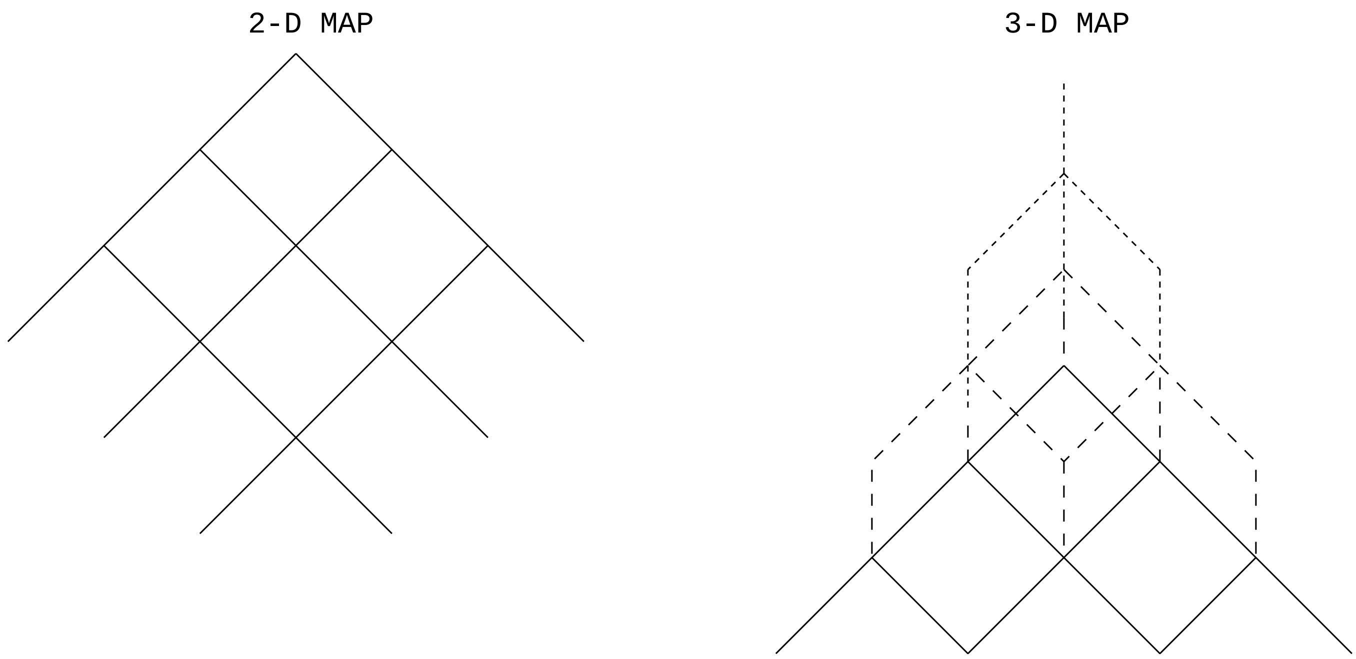

Figure 1: 2-D and 3-D switch diagrams

2 Question 2

Using the diagram from point 1, develop an equivalent network from multiplexers. Try to make it flat and regular. How to realize an arbitrary function in it?

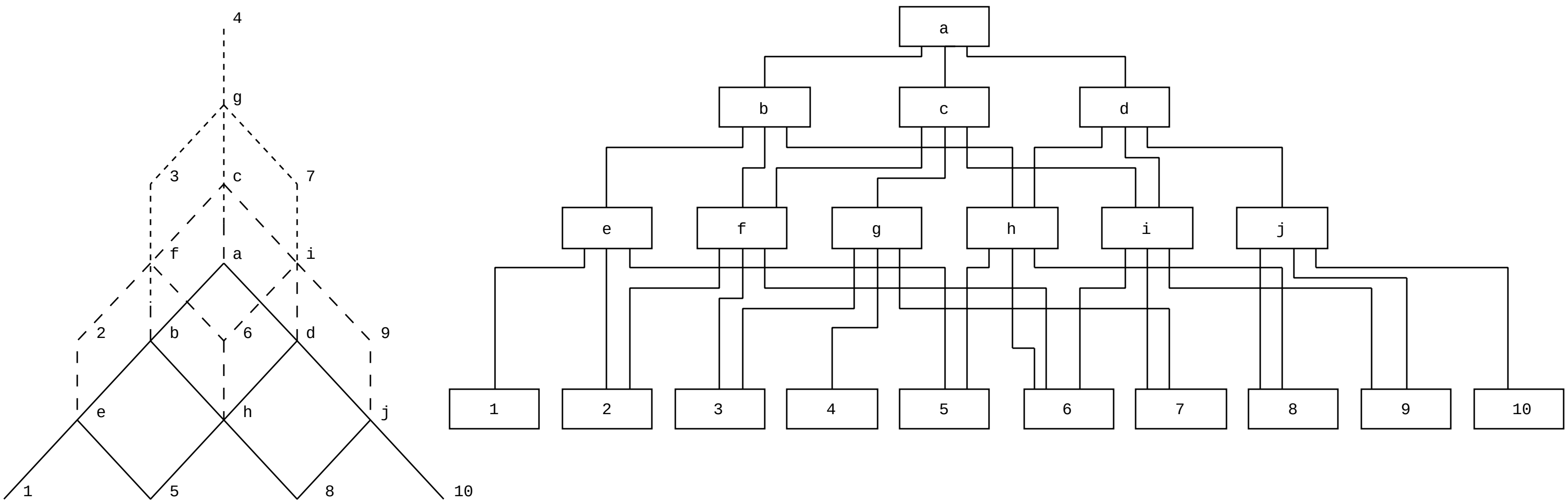

Figure 2: Network of Multiplexers

3 Question 3

Using the diagram from point 1, develop an equivalent network from de-multiplexers and decoders for multi-output functions (we have not done it yet in class for 2-valued logic). Try to make it flat and regular. How to realize an arbitrary function in it?

As far as we can interpret this problem, the only things that need to be changed to make the network applicable for multi-output functions are the inputs to the network and the output. That is, given which function is needed on output, the demultiplex-or selects which function is output to, while the input lines to each of the bottom level multiplexors is changed according to the function that is being requested. Figure 3 shows this representation with a de-multiplexor on the output of the lattice and multiplexors as inputs to all the bottom level nodes in the lattice.

4 Question 4

As you remember, all concepts in binary symmetry are based on cofactors and their negations. In 3-valued logic you still have cofactors, more of them, and you still have negations, more of them. So, how many different partial symmetries we have now? Define all partial and total symmetries.

We have assumed the question means for only two variables. Our group has had many discussions of what symmetry means, with little agreement. For that reason it is reasonable to give foundation to our present approach first giving the example in MV2. The function prototype:

```
The function prototype f =      a\b
                                  A  B
                                  C  D


is equated to

a\b      a\b'   a'\b'   a'\b    b\a    b\a'   b'\a'   b'\a
  A  B     B A     D C     C D    A C    C A     D B     B D
  C  D     D C     B A     A B    B D    D B     C A     A C
```

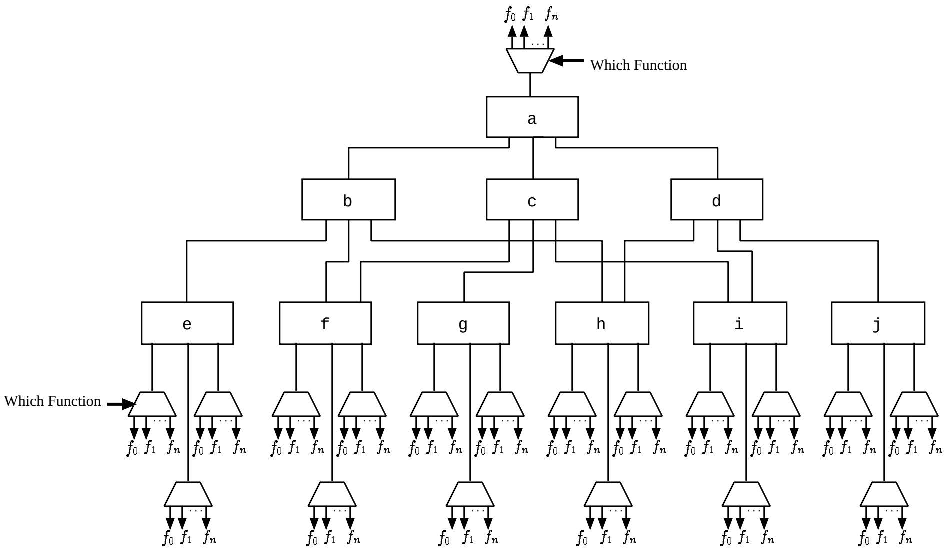

Figure 3: Network for Multi-output functions

```
trivial     A A      A B      A B     A B  trivial    A B  trivial     <=Results
            C C      B A      A B     B D             C A

We note, A B is a specialization of A B and A B
         B A                        B D     C A .

Now we equate f =  a\b     to  negated versions.
                     A  B
                     C  D

a\b       a\b'      a'\b'      a'\b       b\a       b\a'      b'\a'    b'\a
  A' B'     B' A'      D' C'      C' D'     A' C'     C' A'      D' B'    B' D'
  C' D'     D' C'      B' A'      A' B'     B' D'     D' B'      C' A     A' C

                                                                       <=Results
A=A'        A  A'      A  B       A  B      A=A'      A  A'      B=B'     A  A'
Never       C  C'      B' A'      A' B'     Never     A' A       Never    A' A

Note that A  A' is a special case of  A  A'       A  B       A  B
          A' A                        C  C' ,     B' A'  ,   A' B' .
```

We count only 7 independent symmetries with two variables in MV2. It should be clear that our approach and our results are distinct from Sadowska.

For MV3 many more candidates for symmetry arise. A convenient way to designate a bijective transformation in MV3 is by use of the operations of GF(3). For example, the mapping of the variable b: (0,1,2)-¿ (1,2,0) may be represented by b'=1 $\oplus$ b. The mapping (0,1,2)-¿(0,2,1) may be shown with 2b. All the bijective transformations in MV3 are representable by the well known GF(3) operations of multiplication and priming.

An example of a candidate for symmetry is f(a,b)=f'(2b,a). We can enumerate all the candidates for possible symmetries.

- 3 choices primes on f.
- 2 choices of multiplicand 1 or 2 of the first argument.
- 2 for possible reversal of variable order.
- 3 choices of primes on the first argument.
- 2 choices of multiplicand 1 or 2 of the second argument,
- 3 choices of primes on the second argument.

This makes 216 candidates for symmetry in MV3. As in MV2 we find that not all candidates are acceptable. Returning to f(a,b)=f'(2b,a) we find that f(0,0)=f'(0,0) which can never be true.

```
Hereafter we take f(a,b)= a\b
                          A B C
                          D E F
                          G H I.

Studying
f(a,b)= f(b,a'), we find that A B C   D G A                      A A A
                              D E F = E H B gives trivial symmetry A A A
                              G H I   F I C                      A A A .

Exploring further, f(a,b)=f(b,2a) leads to

  A B C   A G D                  A B B
  D E F = B H E resulting in symmetry B E E
  G H I   C I F                  B E E  .

Also interesting, f(a,b)=f'(a,b') becomes

 A B C   C' A' B'              A  A' A''
 D E F = F' D' E' gives the symmetry D  D' D''
 G H I   I' G' H'              G  G' G'' .
```

Like with MV2, in MV3 we find that of the 216 symmetry candidates some are impossible, trivial, or are special cases of another symmetry form. The rich structure of these exceptional cases presently limits our ability to derive a precise enumeration of independent symmetries. Certainly one approach we have not explored is to program a search of such symmetries. 5.Give two ways of deriving lattice diagrams.

5 Question 5

Remember that the concept of symmetry is a concept of a function, not its particular realization with one or another representation (the same as Curtis decompositions). The general symmetries should be then observable in ternary functions regardless how you represent them. So, whatever will be seen in GF(3) should be also seen with Post logic. However, 3-valued Shannon and GF(3) expansions can be useful in exactly the same way as Shannon and two Davio expansions are useful to define various partial symmetries in binary logic. The concept of regular "lattice" structure that we developed for binary, will still hold for ternary. One will be in three dimensions, and one will be flat and regular but with intersecting wires (however still short wires). So, these new lattice diagrams can be derived in two ways. Can you show these ways?

Geometry: The base case for a MV3 lattice is an input node at the origin of R3. To extend an MV3 lattice to one more variable, we feed each input node of coordinates (j,k,l) with the new input nodes (j+1,k,l),(j,k+1,l),(j,k,l+1). Of course, the old input nodes cease to be considered input nodes.

Algebra: Here we are not concerned with spatial relations. We designate a node with the ordered triplet (a,b,c) to convey the number of 0 branches, 1 branches, 2 branches to get to the output. At a depth, say 5, we will see triplets such as (0,5,0), (4,0,1),(1,4,0),(2,1,2) as distinct nodes. All of these triplets sum to 5. At a depth N, all triplets of nonnegative integers summing to N are formed.

Note: A flattened MV3 lattice is no longer a lattice in two dimensions. At depth N, the width is $\frac{N(N+1)}{2}$.

This growth cannot be accommodated on a physical lattice of 2-D. Other problem: Each level of the 3-D lattice forms a triangular grid. Mapping a 2-D grid to each 1-D level of a flat lattice does not maintain nearest neighbor relations. For this reason we conjecture that a flattened MV3 lattice has some wires of length that increases without bound as the number of variables increases.

6 Question 6

When you have 2 binary wires, you can represent a 4-valued logic signal with them. Can you show how to use this concept in case of 4-valued symmetric functions from your definitions. This concept can be realized with two, two valued symmetric functions. Given that one of the symmetric functions produces the most significant bit and the other gives the least significant bit of the 4-values logic signal (just like the concept of 2 binary wires). Figure 4, shows this type of layout, where each of the inputs to the left lattice are the most significant bit, f_1 outputed, and the right lattice has all least significant inputs and f_0 outputed.

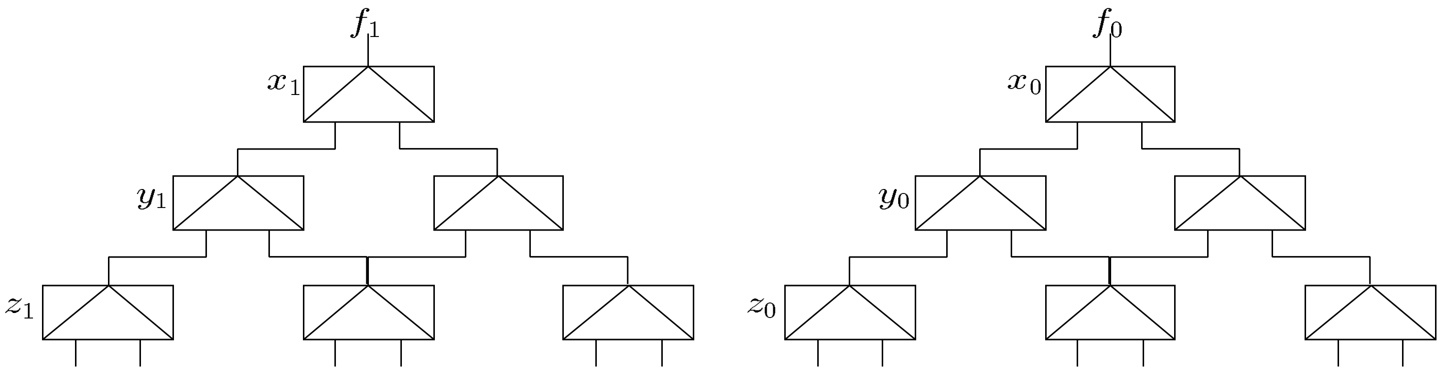

Figure 4: Two symmetric functions, realizing one 4 valued function