HW2 Problem 8: Yet More On Metaproducts

What you know

- Metaproducts are a way to represent SOP forms as BDDs
- Some BDD-like logic manipulations are "supposed to work…"
- …but results are very difficult to interpret
- HW2 problem 8: a 4-variable function is *too big* to see what's going on.

What you don't know

- That metaproducts really represent ***sets of things*** (like we introduced in Lec05 on Formal Verification)
- How to *really* interpret ops like NOT(BDD in metaproduct form)

More On Metaproducts

Suppose I have a Boolean function of 2 vars: F(x,y)

- If I want to consider writing an SOP equation for F(x,y), how many possible product terms could there be?

Can enumerate: there are $3^2 = 9$ terms:

- Every product has 2 "slots" for literals in it
- The first slot can be one of {ε, x, x'} where "ε" means "empty"
- The second slot can be one of {ε, y, y'} where "ε" also means "empty"
- Why 9 terms max? |{ε, x, x'}| x |{ε, y, y'}| = 3x3 = 9

Examples

- Term xy' == (1st slot is x)(2nd slot is y')
- Term x' == (1st slot is x')(2nd slot is ε -- empty)

More on Metaproducts

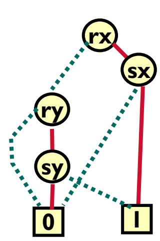
- Well, what are *all* 9 of these possible product terms?

1^{st} slot	2^{nd} slot	Product Term Represented
ε	ε	ε = empty
x	ε	x
x'	ε	x'
ε	y	y
ε	y'	y'
x	y	xy
x	y'	xy'
x'	y	x'y
x'	y'	x'y'

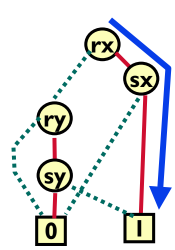
- OK, what does this have to do with metaproducts…?

More on Metaproducts

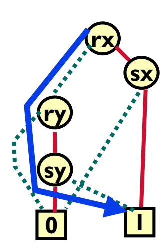
- A metaproduct is really a BDD that represents *a set*
  - The set it represents is some arbitrary *set of product terms,* chosen from the complete set of 9 (in this 2-variable case) on previous slide
- Example: F(x,y) = x + y'



Metaproduct BDD



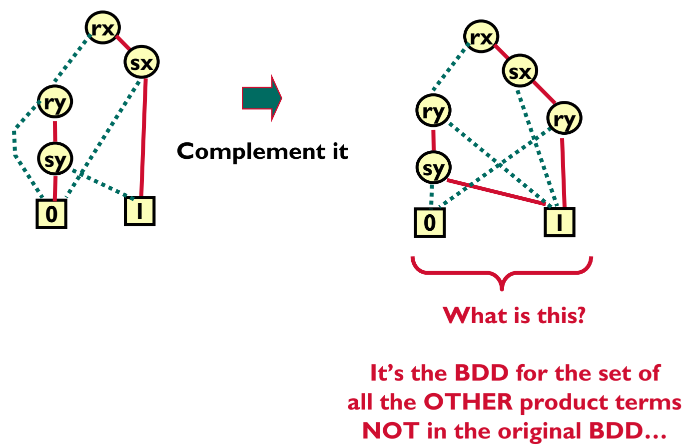
1^{st} path to "1" node represents term " x "



2^{nd} path to "1" node represents term " y' "

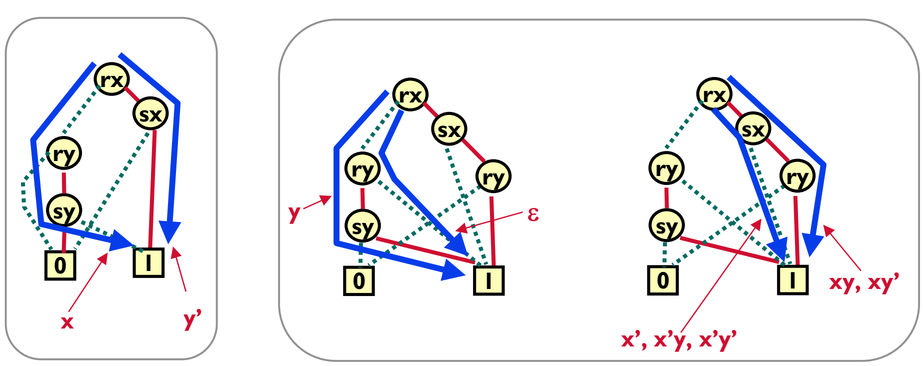
More on Metaproducts

- So, what really happens when you complement this BDD?



More on Metaproducts

- So, what really happens when you complement this BDD?
  - You get a new BDD that represents the 7 other products NOT in original set

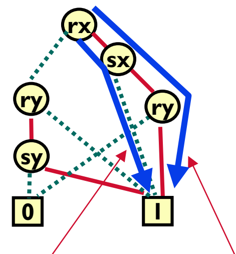


2 product terms in original set

7 other product terms that were NOT in original set

More on Metaproducts

- Subtle stuff
  - Interpreting what happens when you see missing variables

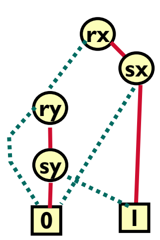


x is here and negative, but no y occurrence var. Interpret as: all values of y are possible, including the empty "ε" y value. Result is: x', x'y, x'y'

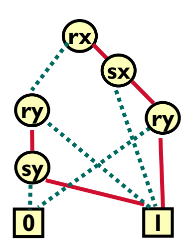
x is here and positive, y is here, but no y sign var. Interpret as: all "signed" values of y are possible, but not the empty "ε" y value. Result is: xy, xy'

More on Metaproducts

- So both original metaproduct BDD and its complement are just sets of stuff. They represent subsets of these 9 terms
  - When you complement one of these, you **don't** get F'( ). You get a set that represents all the other terms you didn't represent originally



1st slot	2nd slot	Product Term Represented
ε	ε	ε = empty
x	ε	x
x'	ε	x'
ε	y	y
ε	y'	y'
x	y	xy
x	y'	xy'
x'	y	x'y
x'	y'	x'y'



Represents these 2 of 9 possible terms

Represents these 7 of 9 possible terms

Back to Homework

About HW2 Problem 8

- The part about "..complement it and explain it" was aimed at this, but with F( ) = 4 variables, its just way too complicated to see. (Sorry…)

- Do this instead of the complicated 4-variable function:
  - Let F(x,y) = x'y + xy'
  - Draw the BDD for the metaproduct form for F( )
  - Draw the complement BDD for this metaproduct BDD
  - Like in these notes, show that the complement really does represent all of the *other* product terms not in the original BDD.