Review: Group Exercise 1

Are the following sets identical?

- **(*A* Ç *B*') È (*A*' Ç *B*)**
- **(*A* Ç *B*)' Ç (*A* È *B*)**

Schematic vs. formal analysis

Review: Group Exercise 2

Are the following statements equivalent?

- **(*p* Ù Ø*q*) Ú (Ø*p* Ù *q*)**
- **Ø(*p* Ù *q*) Ù (*p* Ú *q*)**

Semantic vs. proof-based analysis

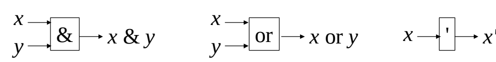
C2: Boolean Algebra

Today

- Compare three structures and analyze difference/similarity
  - Digital circuits
  - Sets
  - Propositional logic
- Understand Boolean Algebra and its variations
- Take-home exercises
  - Pseudo-electrical engineer, Exotic Logic and Circuits

Section 1

Digital Circuits



x	*y*	*x* & *y*
1	1	1
1	0	0
0	1	0
0	0	0

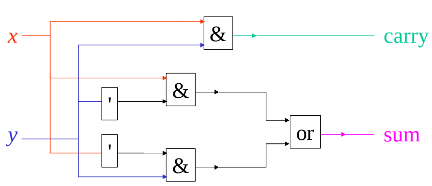
x	*y*	*x* or *y*
1	1	1
1	0	1
0	1	1
0	0	0

x	*x*'
1	0
0	1

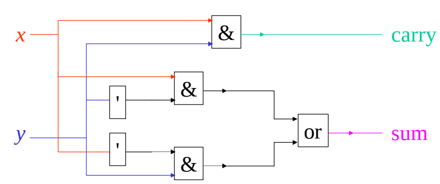
Addition (1)

0	0	1	1
0	1	0	1
00	01	01	10

$10_2 = 2_{10}$

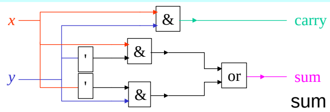


Addition (2)

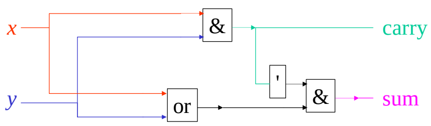


- carry = ***x* & *y***
- sum = **(*x* & *y*') or (*x*' & *y*)**

Addition (3)



sum = **(*x* & *y*') or (*x*' & *y*)**



sum = **(*x* & *y*)' & (*x* or *y*)**

equivalent?

Comparison

- Digital circuits

  (*x* & *y*') or (*x*' & *y*) = (*x* & *y*)' & (*x* or *y*)
- Sets

  (*A* Ç *B*') È (*A*' Ç *B*) = (*A* Ç *B*)' Ç (*A* È *B*)
- Propositional logic

  (*p* Ù Ø*q*) Ú (Ø*p* Ù *q*) Û Ø(*p* Ù *q*) Ù (*p* Ú *q*)

What can we say?

Section Summary

- The structures for digital circuits, sets, and propositional logic are indistinguishable (except for the use of different symbols).
  - Indistinguishable structures are said to be *isomorphic*.
- If we understand one structure, we can translate the results to other structures.

Section 2

Representative Structure

- **Bool** = (*B*, &, or, ')
  - *B* = {1, 0}
  - &: *B* × *B* → *B*

    & = {((0, 0), 0), ((0, 1), 0), ((1, 0), 0), ((1, 1), 1)}
  - or: *B* × *B* → *B*

    or = {((0, 0), 0), ((0, 1), 1), ((1, 0), 1), ((1, 1), 1)}
  - ': *B* → *B*

    ' = {(0, 1), (1, 0)}

Group Exercise 3

x	*y*	*x* & *y*
1	1	1
1	0	0
0	1	0
0	0	0

x	*y*	*x* or *y*
1	1	1
1	0	1
0	1	1
0	0	0

For each operation:

- Commutative?
- Associative?
- Identity?
- Additional properties?

Additional Properties

- De Morgan's law
  - **(*a* or *b*)' = *a*' & *b*', (*a* & *b*)' = *a*' or *b*'**
- Distributive
  - ***a* or (*b* & *c*) = (*a* or *b*) & (*a* or *c*)**
  - ***a* & (*b* or *c*) = (*a* & *b*) or (*a* & *c*)**

Boolean Algebra (definition)

- A class of structures **Bool** = (B, &, or, ', 0, 1) such that:
  - &: $B \times B \rightarrow B$, or: $B \times B \rightarrow B$, ': $B \rightarrow B$, 0: $\rightarrow B$, 1: $\rightarrow B$
  - & and or are associative, commutative, and with an identity.
  - Existence of complements (i.e., ' is defined).
  - & and or are distributive.

What about De Morgan?
What about if $|B| \neq 2$?

Isomorphic Structures

- **Bool** = ({0, 1}, &, or, ')
- **Set** = ({∅, {1}}, ∩, ∪, ')
- **Prop** = ({F, T}, ∧, ∨, ¬)

x	y	x or y
1	1	1
1	0	1
0	1	1
0	0	0

x	y	$x \cup y$
{1}	{1}	{1}
{1}	∅	{1}
∅	{1}	{1}
∅	∅	∅

x	y	$x \vee y$
T	T	T
T	F	T
F	T	T
F	F	F

different spec of same structure?

Isomorphism

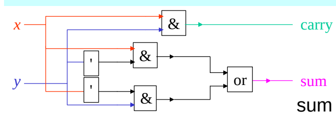
- Transformation of a structure to another structure such that
  - Structure components correspond in the order
  - Structures are identical except for the use of symbols

Bool = ({0, 1}, &, or, ')
Set = ({∅, {1}}, ∩, ∪, ')
Prop = ({F, T}, ∧, ∨, ¬)

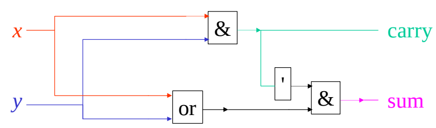
If structures are represented as sets instead of n-tuples, ...

Revisited

Justification of Equivalence



sum = **(*x* & *y*') or (*x*' & *y*)**



sum = **(*x* & *y*)' & (*x* or *y*)**

Equivalence of Expressions

- Digital circuits
  - Computer Engineering?
  - CMSC325?
- Sets
  - Convert to set expressions and justify the result schematically
- Propositional logic
  - Convert to propositional logic and
    - Check $exp_1 \leftrightarrow exp_2$ is a tautology [semantics]
    - Prove exp_2 from exp_1 (and vice versa) [syntax]

Revisited

Review: Group Exercise 1

Are the following sets identical?

- **(*A* Ç *B*') È (*A*' Ç *B*)**
- **(*A* Ç *B*)' Ç (*A* È *B*)**

Schematic vs. formal analysis

Revisited

Review: Group Exercise 2

Are the following statements equivalent?

- **(*p* Ù Ø*q*) Ú (Ø*p* Ù *q*)**
- **Ø(*p* Ù *q*) Ù (*p* Ú *q*)**

Semantic vs. proof-based analysis

Proof Approach

Hyp: $(x \wedge \neg y) \vee (\neg x \wedge y)$, To prove: $\neg(x \wedge y) \wedge (x \vee y)$

1.	$(x \wedge \neg y) \vee (\neg x \wedge y)$	[hyp]
2.	$\neg\neg(x \wedge \neg y) \vee \neg\neg(\neg x \wedge y)$	[double neg]
3.	$\neg(\neg x \vee y) \vee \neg(x \vee \neg y)$	[De Morgan]
4.	$\neg((\neg x \vee y) \wedge (x \vee \neg y))$	[De Morgan]
5.	$\neg((\neg x \wedge (x \vee \neg y)) \vee (y \wedge (x \vee \neg y)))$	[distributive]
6.	$\neg((\neg x \wedge x) \vee (\neg x \wedge \neg y)) \vee ((y \wedge x) \vee (y \wedge \neg y)))$	[distributive]
7.	$\neg((F \vee (\neg x \wedge \neg y)) \vee ((y \wedge x) \vee F))$	[contradiction (subformula)]
8.	$\neg((\neg x \wedge \neg y) \vee (y \wedge x))$	[use of '∨']
9.	$\neg(\neg x \wedge \neg y) \wedge \neg(y \wedge x)$	[De Morgan]
10.	$(x \vee y) \wedge \neg(y \wedge x)$	[De Morgan]
11.	$\neg(x \wedge y) \wedge (x \vee y)$	[commutative]

Section Summary

- Boolean algebra is a class of structures that can be used to represent digital circuits, set theory, and propositional logic.

Summary Exercise

- Did you know Boolean Algebra before this class? Briefly comment on the following:
  - If yes, did your idea change after this class?
  - If no, do you think you will use it?
- Questions/Comments/Suggestions