Puzzle

You meet A, B, and C in the land of knights and knaves.

A says "Either B and I are both knights or we are both knaves."

B says "C and I are the same type."

C says "Either A is a knave or B is a knave."

Who is what?

Methods of Proof for Boolean Connectives

Monday, 13 September

What a truth table can show us

- A sentence is a tautology iff every row of its truth table assigns TRUE to that sentence.
    - A sentence is a contradiction iff it is always false.
- Two sentences are tautologically equivalent iff they have matching truth tables.

WHAT A TRUTH TABLE CAN SHOW US

- A sentence Q is a tautological consequence of a set of sentences $P_1 \ldots P_n$ iff every row of the truth table where $P_1 \ldots P_n$ are all true, Q is also true [i.e. there are NO rows where $P_1 \ldots P_n$ are all true and Q is false].
  - We also say $\{P_1 \ldots P_n\}$ tautologically implies Q
- A set of sentences $P_1 \ldots P_n$ is truth-functionally consistent iff there is at least one row of the truth table where $P_1 \ldots P_n$ are all true.

These terms are interdefinable

- For example, if $\{P_1 \ldots P_n\}$ implies Q iff $\{P_1 \ldots P_{n,}, \neg Q\}$ is inconsistent.

- $\{P_1 \ldots P_{n,}, \neg Q\}$ inconsistent iff $\neg(P_1 \wedge \ldots \wedge P_n \wedge \neg Q)$ is a logical truth.

Conditionals and Logical Consequence

- A sentence Q is a logical consequence of a set of sentences $P_1, P_2 \ldots P_n$ iff it is impossible for the premises to be true and the consequent to be false.
- This is exactly the same as the falsity of $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow Q$
- Therefore: $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow Q$ is a logical truth iff Q is a logical consequence of $P_1, P_2 \ldots P_n$.

Conditionals and Logical Consequence

- $P \leftrightarrow Q$ is a logical truth iff P and Q are logically equivalent (have the same truth values).
- In other words, $P \leftrightarrow Q$ is a logical truth iff $P \Leftrightarrow Q$.
  - NOTE: $P \leftrightarrow Q$ might just happen to be true without P and Q being equivalent
- Recall: $A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$.
- Therefore A is logically equivalent to B iff A is a logical consequence of B and B is a logical consequence of A.

TABLES ARE REALLY POWERFUL

- Knights and Knaves problems reduce to a truth table
  - Find the row where these are all true:
    - Knight(a) ↔ ¬Knave(a)
    - Knight(b) ↔ ¬Knave(b)
      - If A says "Both of us are knaves" then add:
    - Knight(a) ↔ [Knave(a)∧Knave(b)]

Tables are really powerful

- Sudoku problems reduce to a truth table
  - Find a row of the table where these are all true:
    - The first cell is exactly one of 1-9:
      - Exactly one of Cell(1,1), Cell(1,2), ..., Cell(1,9)
    - The second cell is 1-9.... the 81st cell is 1-9
    - The first row has exactly one 1:
      - Exactly one of Cell(1,1), Cell(2,1), ..., Cell(9,1)
    - The second row has.... The upper left box has...

Tables are really powerful

- Determining whether (or in which case) a set of sentences can be simultaneously true is sometimes called 'the satisfiability problem' or 'the Boolean satisfiability problem' or '3-sat' (if 3 variables, etc.)
- This problem is EXTREMELY important in computer science because so many problems are equivalent to solving this problem
- But truth tables are trivial (Microsoft Excel will do them for you) so why is this interesting?

TABLES ARE POWERFUL - BUT REALLY SLOW

- In the sudoku case, as written, each sentence is pretty long and there are lots of sentences, but the real problem is the total number of rows. For the 81x9 = 729 variables there are 2^729 rows in the table ≈10^84. My 2.4 GHZ laptop would take ≈10^70 years at maximum efficiency to finish this table.
- Perhaps the most important problem in computer science - Does P=NP?
  - Very roughly equivalent to: Is there a reasonably fast way solve the satisfiability problem?

Proofs

Why not just use truth tables?

- Truth tables get really HUGE very quickly.
- Truth tables don't mirror the way in which we make arguments.
- Truth tables only show us tautological consequence, for example they are insensitive to identity. We want to capture a broader notion of logical consequence.

Proofs

- We want formal proofs to mirror the kind of reasoning we use informally.
- We will start by looking at some intuitive steps that we use in making valid informal arguments.
- We will then find ways to formalize these steps in our formal system of proof.
- We already have identity introduction (= intro) and identity elimination (= elim).

Formal Proof Rules for ∧

- ∧ Introduction

  From P and Q, we can infer $P \wedge Q$.

1. P	
2. Q	
3. $P \wedge Q$	∧ Intro: 1,2

- ∧ Elimination

  From $P \wedge Q$, we can infer P.

1. $P \wedge Q$	
2. P	∧ Elim: 1

Formal Proof Rules (∧)

Example:

| A ∧ (B ∧ C)
|———
| (A ∧ B) ∧ C

1. A ∧ (B ∧ C)	
2. A	∧ Elim: 1
3. B ∧ C	∧ Elim: 1
4. B	∧ Elim: 3
5. C	∧ Elim: 4
6. A ∧ B	∧ Intro: 2,4
7. (A ∧ B) ∧ C	∧ Intro: 5,6

Main Connectives

- **Incorrect**

1. $\neg(P \rightarrow R)$
2. Q

---

3. $\neg((P \land Q) \rightarrow R)$ $\land$ Intro: 1,2

- **Incorrect**

1. $\neg(P \land Q)$

---

2. $\neg P$ $\land$ Elim: 1

PROOFS

Disjunction Introduction

- Intuitively, if you know that A is true, then you can conclude that either A or B (or both).
- Ex: If Alice will be at the party, then it is true that either Alice or Bill will be there.
- In general, from P we can infer ‘P or Q’.

Formal Proof Rules (∨)

- ∨ Introduction

  From P, we can infer P∨Q.

  | 1. P
  |---
  | 2. P ∨ Q   ∨ Intro: 1

- Another example:

  | 1. P
  |---
  | 2. $P \vee ((Q \leftrightarrow R) \rightarrow \neg S)$ ∨ Intro: 1

PROOF BY CASES

- Intuitively, if you know that A or B is the case, and that C follows from A and C also follows from B, then you know that C is the case.
- Example: I will either go to the bank on Monday or Tuesday. So either way, I will have some money to buy lunch on Wednesday.

Proof by Cases

Disjunction Elimination

- In general, proof by cases (disjunction elimination) is when you start with a disjunction and show for each disjunct that, if you assume its truth, some sentence S follows.
- Note: you don't need to know which disjunct is true.

Proof by Cases

- Disjunction Elimination formalizes proof by cases.
- In order to use proof by cases, we need to be able to make assumptions in our proof.
- To show that certain things follow from a set of assumptions, we use subproofs.
- BUT we can only make assumptions within a subproof.

Proof by Cases

- ∨ Elimination

  If R follows from P, and if R follows from Q, then from P∨Q, we can infer R.

Scope Lines

Scope Lines indicate assumptions that don't necessarily follow from earlier assumptions

```
| 1. P ∨ Q
|----
| | 2. P
| |----
| | ...
| | j. R            ??
| | k. Q
| |----
| | ...
| | m. R            ??
| n. R              ∨Elim: 1,2-j,k-m
```

Proof by Cases

Example:

$(A \wedge B) \vee \neg C$

$B \vee \neg C$

1. $(A \wedge B) \vee \neg C$	
2. $A \wedge B$	
3. B	$\wedge$ Elim: 2
4. $B \vee \neg C$	$\vee$ Intro: 3
5. $\neg C$	
6. $B \vee \neg C$	$\vee$ Intro: 5
7. $B \vee \neg C$	$\vee$ Elim: 1,2-4,5-6