Informatics 1

Computation and Logic

Lecture 19

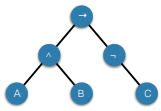
Logic: The Big Ideas

A formal language, without the vagueness and ambiguity of natural language

Syntax:
expressions are built up from atomic propositions using logical connectives

∧∨¬→

expressions are trees, with atomic propositions as leaf nodes and other nodes labelled with connectives

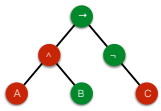


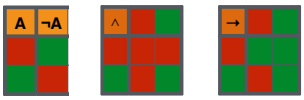
A ∧ B → ¬C

precedence
¬
∧
∨
→
⬌

A formal language, without the vagueness and ambiguity of natural language

Semantics:
the truth value expressions is built up "compositionally" from the truth values of its atomic propositions using logical operators





A ∧ B → ¬C
A ∧ B → ¬C
A ∧ B → ¬C

Formal Inference

proofs are built from assumptions using sound rules
proofs are trees, with assumptions as leaves, and other nodes labelled with instances of rules

a deduction rule

$$\frac{A \rightarrow B \quad \neg B}{\neg A}$$

an entailment

$A \rightarrow B, \neg B \vdash \neg A$

Formal Inference

proofs are built from assumptions using sound rules
proofs are trees, with assumptions as leaves, and
other nodes labelled with instances of rules

a proof

$$\dfrac{\dfrac{A \to B \quad \neg B}{\neg A} \qquad \dfrac{B \to C \quad \neg C}{\neg B}}{\neg A \to \neg B}$$

cut rule
a rule for composing proofs

$$\dfrac{A \to B,\ \neg B \vdash \neg A \qquad B \to C,\ \neg C \vdash \neg B}{A \to B,\ B \to C,\ \neg C \vdash \neg A}$$

?

$$(A \to B) \to (A \to C) \vdash A \to (B \to C)$$

Natural Deduction

one natural way to prove A→B is to assume A and prove B

a proof

$$\dfrac{\dfrac{A\to B \quad \dfrac{B\to C \quad \neg C}{\neg B}}{\neg A}}{}$$

→ introduction

a rule of inference

$$\dfrac{\Gamma, X \vdash Y}{\Gamma \vdash X\to Y}$$

a proof?

$$\dfrac{\dfrac{A\to B \quad \dfrac{B\to C \quad \not{\neg C}}{\neg B}}{\neg C \to \neg A}}{}$$

$$\dfrac{\dfrac{A\to B, \neg B \vdash \neg A \qquad B\to C, \neg C \vdash \neg B}{A\to B, B\to C, \neg C \vdash \neg A}}{A\to B, B\to C \vdash \neg C \to \neg A}$$

Natural Deduction

one natural way to prove X→Y is to assume X and prove Y

and if we can prove X→Y then from X we can infer Y

→ introduction & elimination

a 2-way rule of inference

$$\frac{\Gamma, X \vdash Y}{\Gamma \vdash X \to Y}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A,B \vdash B}{B \vdash A \to B} \quad \dfrac{(A \to B) \to (A \to C) \vdash (A \to B) \to (A \to C)}{A \to B,\ (A \to B) \to (A \to C) \vdash A \to C}}{B,\ (A \to B) \to (A \to C) \vdash A \to C} \quad \dfrac{A \to C \vdash A \to C}{A \to C,\ A \vdash C}}{(A \to B) \to (A \to C),\ A,\ B \vdash C}}{(A \to B) \to (A \to C),\ A \vdash B \to C}}{(A \to B) \to (A \to C) \vdash A \to (B \to C)}$$

The proofs may be natural, but sometimes they are hard to find!

Gentzen's idea

Instead of just entailments, $\Gamma \vdash X$

(where X is an expression and Γ is a finite set of expressions)

allow **sequents**, $\Gamma \vdash \Delta$

(where both Γ and Δ are finite sets of expressions)

Of course, every entailment 'is' a sequent

(where Δ is a singleton)

but the sequent calculus is much simpler than natural deduction

KEEP CALM & FOLLOW THE RULES

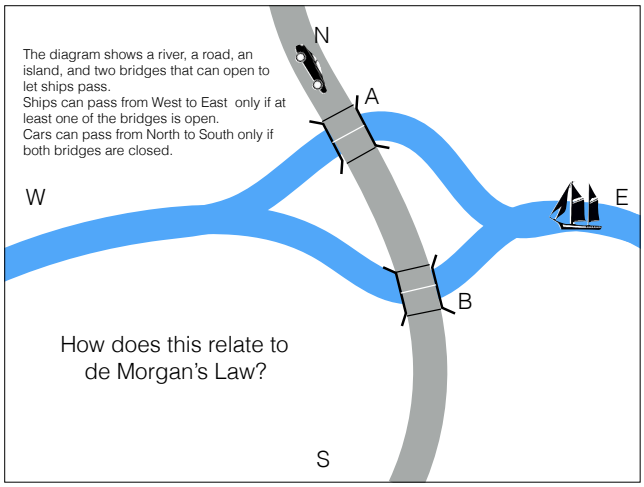
$$\frac{}{\Gamma, A \vdash \Delta, A}\ (I)$$

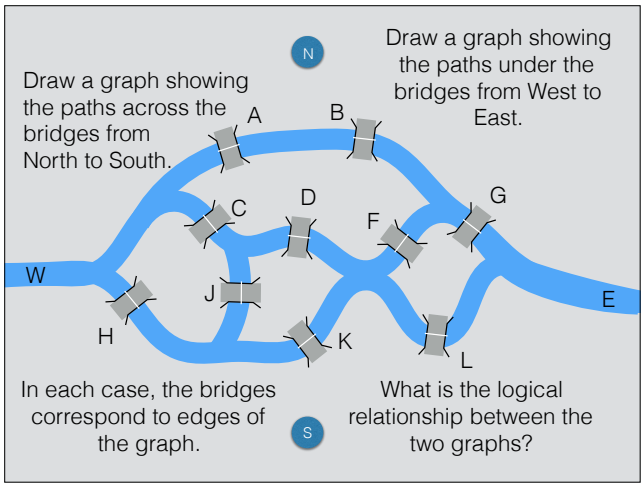
$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ (\vee R)$$

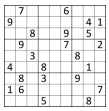
$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ (\wedge R)$$

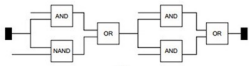
$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\ (\to L) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\ (\to R)$$

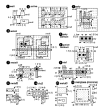
$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ (\neg L) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ (\neg R)$$


The diagram shows a river, a road, an island, and two bridges that can open to let ships pass.
Ships can pass from West to East only if at least one of the bridges is open.
Cars can pass from North to South only if both bridges are closed.
N
A
W
E
B
How does this relate to de Morgan's Law?
S


N
Draw a graph showing the paths across the bridges from North to South.
Draw a graph showing the paths under the bridges from West to East.
A
B
C
D
F
G
W
J
E
H
K
L
In each case, the bridges correspond to edges of the graph.
What is the logical relationship between the two graphs?
S







We can express many combinatorial problems in propositional logic

(eg Sudoku, but also more practical problems, such as circuit design)

We can use resolution to check whether a set of clauses is consistent.

If we can derive the empty clause the set is inconsistent, and we can invert the proof to produce a refutation tree

If we cannot derive the empty clause we can construct a satisfying valuation from the failed attempt to prove a contradiction

Generating all the resolvants takes space and time

We can express many combinatorial problems in propositional logic
(eg Sudoku, but also more practical problems)

We can search for solutions to a set of constraints expressed in propositional logic

We convert the problem to clausal form
and check partial valuations V against our constraints
if V contradicts any clauses our search must backtrack.

Checking these potential solutions costs time and space

We can narrow the search by **unit propagation**:
identifying literals whose siblings are all falsified,
and making them true

Keeping track of unit literals takes time

The search for solutions to a set of constraints

We convert the problem to clausal form
and check partial valuations V against our constraints
if V contradicts any clauses our search must backtrack.

Checking these potential solutions costs time and space

We speed up the search by watching one or two literals in each clause and checking whether we can maintain an invariant

for each clause,
if any watched literal is false then some watched literal is true
if we cannot maintain the invariant we must backtrack

if we watch two literals, then unit propagation is included in the procedure we use to maintain the invariant