AI: Reasoning

Introduction to Cognitive Science

Normative and Descriptive Theories of Reasoning

- Psychology of reasoning is a scientific study of how humans reason:
  - What do humans infer from what?
  - What is the mechanism behind human reasoning?
- As such, psychologists (and cognitive scientists) come up with *descriptive* theories of reasoning: *hypotheses* as to how humans reason based on *empirical* studies.
- Logicians (philosophers), however, try to come up with *normative* theories of reasoning:
  - What *actually* follows from what?

Logic

- Logic is the study of valid reasoning
- What do we mean by this?
  - When I reason, I try to infer something (conclusion, theorem, result) from something else (premises, axioms, givens).
  - The reasoning is said to be valid when the conclusion does indeed follow from the premises
- Logic tries to provide guidelines for when reasoning is valid, and when it is not.

Deductive Validity

- Sometimes, the truth of the conclusion is 100% guaranteed by the truth of the premises:
  - “I am either purple or orange. I am not orange. So, I am purple”
- Other times, the truth of the conclusion is merely made more likely given the truth of the premises:
  - “He came from the tropics, for his face is dark, and that is not the natural color of his skin, as his wrists are fair”
- The first argument is deductively valid, the second is not.
- In a formal logic course, you study deductive validity.
- An informal logic course covers all aspects of reasoning:
  - Non-deductive reasoning (statistical, causal, scientific)
  - Well-foundedness of reasoning (credibility, sources)
  - Effects of rhetoric, emotions, cognitive and social biases

Argument Forms

- “If I win the lottery, then I am poor. I win the lottery. Hence, I am poor.”
- This argument has the following abstract structure or *form*: “If P then Q. P. Hence, Q”
- *Any* argument of the above form is valid, including “If flubbers are gook, then trugs are brig. Flubbers are gook. Hence, trugs are brig.”!
- Hence, we can look at the abstract form of an argument, and tell whether it is valid without even knowing what the argument is about!!

Formal Logic

- Formal logic studies the validity of arguments by looking at the abstract form of arguments.
- Formal logic thus works in 2 steps:
  - Step 1: Use certain symbols to express the abstract form of premises and conclusion.
  - Step 2: Use a certain procedure to figure out whether the conclusion follows from the premises based on their symbolized form alone.

Example

- “Either the housemaid or the butler killed Mr. X. However, if the housemaid would have done it, the alarm would have gone off, and the alarm did not go off. Therefore, the butler did it.”

Example Step 1: Symbolization

- We can do this using propositional logic:
- Use symbols to represent simple propositions:
  - H: The housemaid did it
  - B: The butler did it
  - A: The alarm went off
- Use further symbols to represent complex claims:
  - $H \vee B$: The housemaid or the butler did it
  - $H \rightarrow A$: If the housemaid did it, the alarm would go off
  - $\neg A$: The alarm did not go off

Step 2: Evaluation

- Many different techniques have been developed:
  - Truth-Tables
  - Algebra’s
  - Formal Proofs
  - …

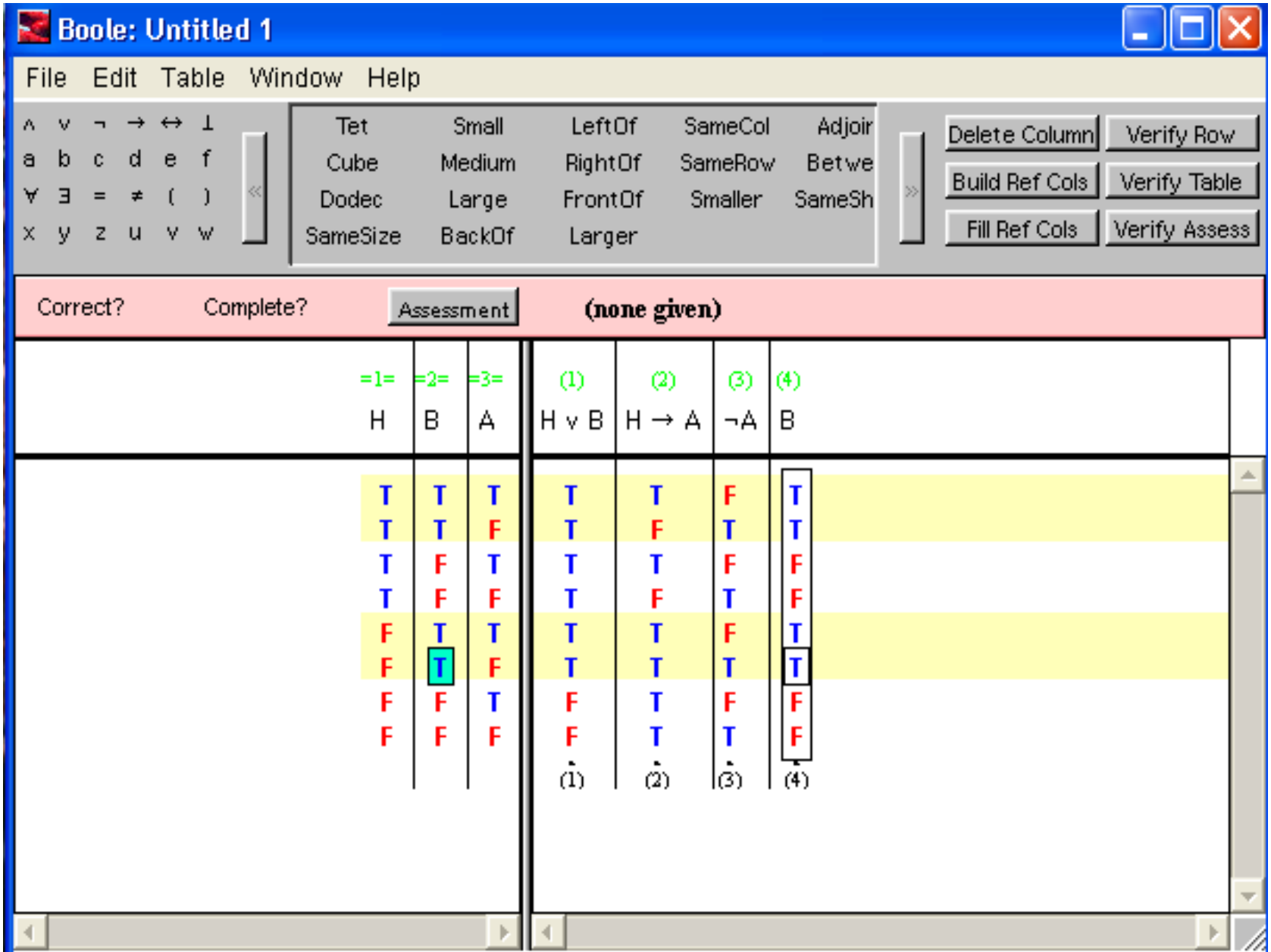
Truth-Tables

P	$\neg P$
T	F
F	T

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Housemaid, Butler, and Alarm



De Morgan's Laws

P	Q	$\neg(P \wedge Q)$		$\neg P \vee \neg Q$			$\neg(P \vee Q)$		$\neg P \wedge \neg Q$		
T	T	F	T	F	F	F	F	T	F	F	F
T	F	T	F	F	T	T	F	T	F	F	T
F	T	T	F	T	T	F	F	T	T	F	F
F	F	T	F	T	T	T	T	F	T	T	T
		↑			↑		↑			↑	

So: $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$

$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$

Some Logical Equivalences

- Double Negation:
  - $P \Leftrightarrow \neg \neg P$
- Commutation:
  - $P \wedge Q \Leftrightarrow Q \wedge P$
  - $P \vee Q \Leftrightarrow Q \vee P$
- Association:
  - $P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$
  - $P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$
- Idempotence:
  - $P \wedge P \Leftrightarrow P$
  - $P \vee P \Leftrightarrow P$
- DeMorgan:
  - $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$
  - $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$
- Distribution:
  - $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$
  - $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$
- Reduction:
  - $P \wedge (\neg P \vee Q) \Leftrightarrow P \wedge Q$
  - $P \vee (\neg P \wedge Q) \Leftrightarrow P \vee Q$
- Subsumption:
  - $P \wedge (P \vee Q) \Leftrightarrow P$
  - $P \vee (P \wedge Q) \Leftrightarrow P$
- Contradiction and Tautology:
  - $P \wedge \neg P \Leftrightarrow \bot$
  - $P \vee \neg P \Leftrightarrow \top$
  - $P \wedge \bot \Leftrightarrow \bot$
  - $P \vee \top \Leftrightarrow \top$
- Implication:
  - $P \rightarrow Q \Leftrightarrow \neg P \vee Q$
  - $\neg(P \rightarrow Q) \Leftrightarrow P \wedge \neg Q$
- Transposition (Contraposition):
  - $P \rightarrow Q \Leftrightarrow \neg Q \rightarrow \neg P$
- Equivalence:
  - $P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$

Housemaid, Butler, and Alarm

$$(H \vee B) \wedge (H \rightarrow A) \wedge \neg A \Leftrightarrow \text{(Implication)}$$

$$(H \vee B) \wedge (\neg H \vee A) \wedge \neg A \Leftrightarrow \text{(Reduction)}$$

$$(H \vee B) \wedge \neg H \wedge \neg A \Leftrightarrow \text{(Reduction)}$$

$$B \wedge \neg H \wedge \neg A$$

Note: the result is not just ‘B’

Modus Ponens

$P \rightarrow Q$

P

Q

P	Q	$P \rightarrow Q$	P	Q	
T	T	T	T	T	←
T	F	F	T	F	
F	T	T	F	T	
F	F	T	F	F	

We see that whenever the premises ($P \rightarrow Q$ and P) are true, the conclusion (Q) is also true. Hence, given the premises, the conclusion is necessarily true as well. So, this argument is deductively valid.

Some Valid Inferences

$$\frac{P \rightarrow Q \quad P}{Q}$$

Modus Ponens

$$\frac{P \vee Q \quad \neg P}{Q}$$

Disjunctive Syllogism

$$\frac{P \rightarrow Q}{(P \wedge R) \rightarrow Q}$$

Strengthening the Antecedent

$$\frac{P \rightarrow Q \quad \neg Q}{\neg P}$$

Modus Tollens

$$\frac{P \rightarrow Q \quad Q \rightarrow R}{P \rightarrow R}$$

Hypothetical Syllogism

$$\frac{P \rightarrow (Q \wedge R)}{P \rightarrow Q}$$

Weakening the Consequent

Some *Invalid* Inferences

$\varphi \rightarrow \psi$

ψ

φ

Affirming the Consequent

$\varphi \rightarrow \psi$

$\neg\varphi$

$\neg\psi$

Denying the Antecedent

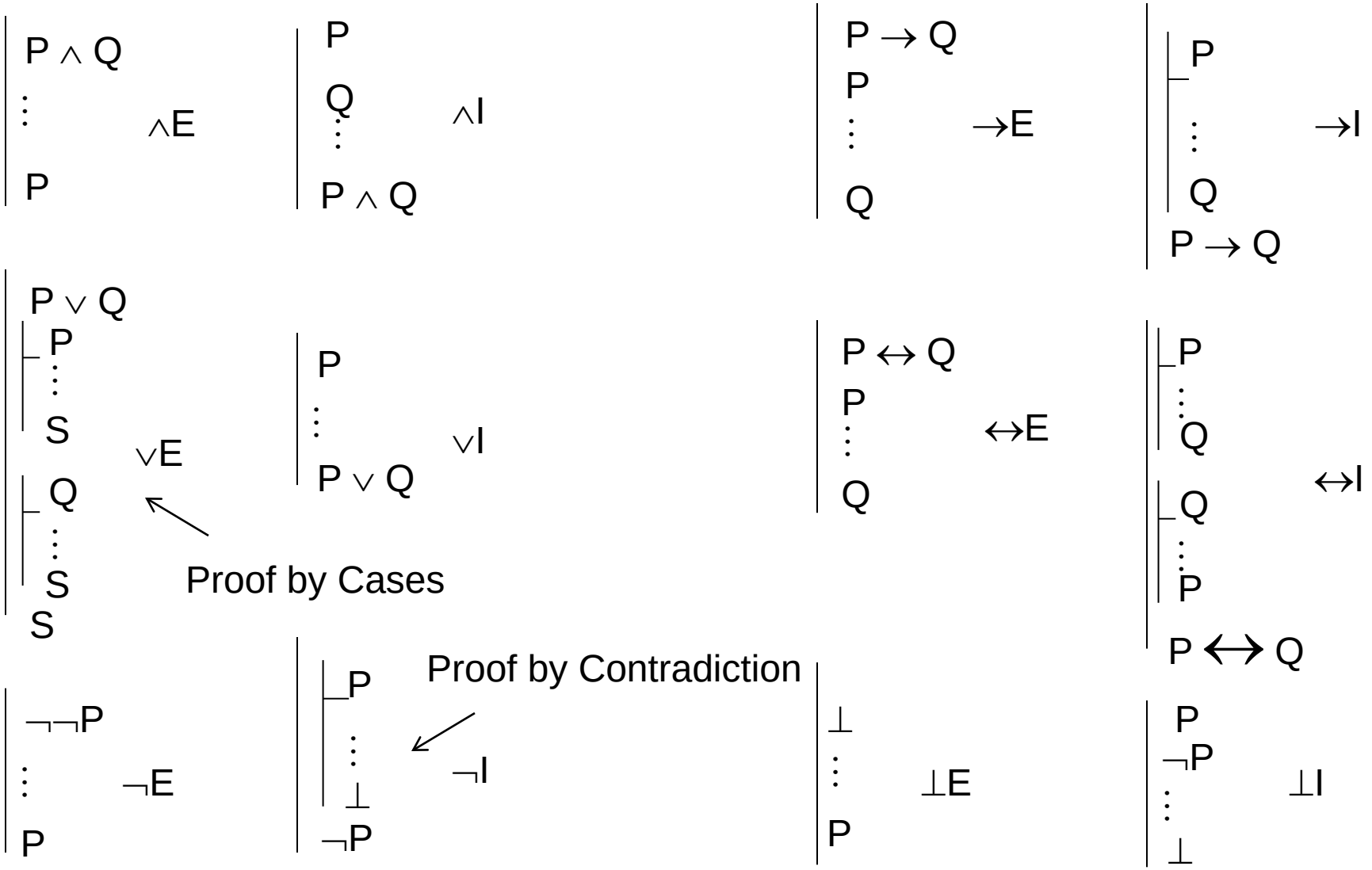
Proof by Contradiction

- Proofs by contradiction follow the following pattern: ‘Assume P to be the case. Then [after some work] I get some kind of impossibility or contradiction. Hence, contrary to my assumption, P cannot be the case.’
- This pattern of reasoning goes by many names:
  - *Proof by Contradiction*
  - *Indirect Proof*
  - *Reductio ad Absurdum*
  - *Reductio Proof*

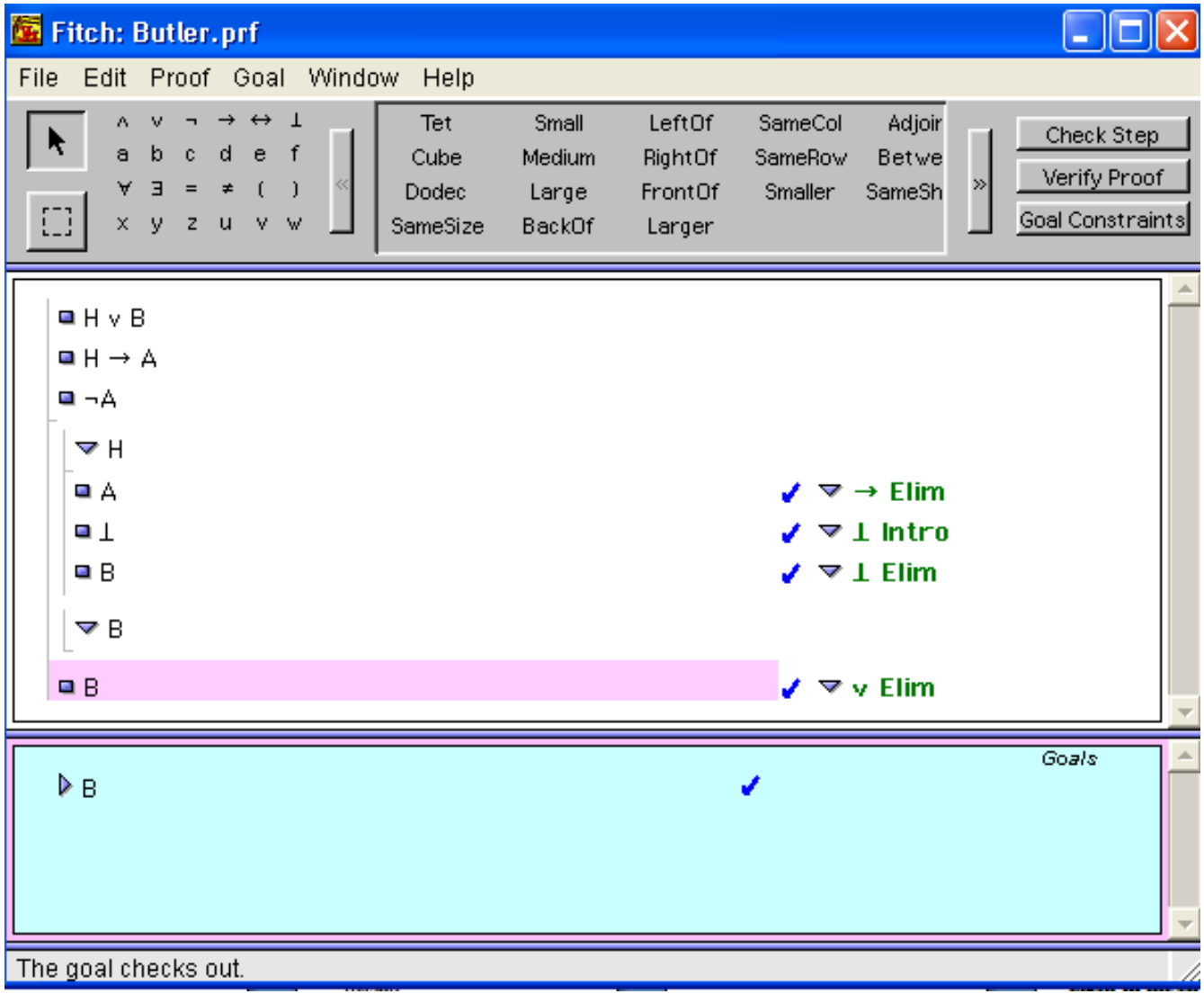
Proof by Cases

- Proofs by Cases follow the following pattern: ‘Either P is the case or Q is the case. However, if P is the case, then S is the case, and if Q is the case, then S is the case as well. Either way, S is therefore the case. Hence, S is the case.’
- This pattern of reasoning is called *Proof by Cases*
- Obviously, the above pattern can be generalized to disjunctions with any number of disjuncts.
- However, a very common form is to start with: ‘Either P is the case or P is not the case’.

Fitch Rules


P ∧ Q
⋮
P
∧E
P
Q
⋮
P ∧ Q
∧I
P → Q
P
⋮
Q
→E
P
⋮
Q
P → Q
→I
P ∨ Q
P
⋮
S
Q
⋮
S
S
∨E
Proof by Cases
P
⋮
P ∨ Q
∨I
P ↔ Q
P
⋮
Q
↔E
P
⋮
Q
Q
⋮
P
P ↔ Q
↔I
¬¬P
⋮
P
¬E
P
⋮
⊥
¬P
Proof by Contradiction
¬I
⊥
⋮
P
⊥E
P
¬P
⋮
⊥
⊥I

Housemaid, Butler, and Alarm



Some Formal Logic Axioms for Set Theory

- Identity:
  - $\forall x\ \forall y\ (x = y \leftrightarrow \forall z\ (z \in x \leftrightarrow z \in y))$
- Subset:
  - $\forall x\ \forall y\ (x \subseteq y \leftrightarrow \forall z\ (z \in x \rightarrow z \in y))$
- Intersection:
  - $\forall x\ \forall y\ \forall z\ (z \in x \cap y \leftrightarrow (z \in x \wedge z \in y))$
- Union:
  - $\forall x\ \forall y\ \forall z\ (z \in x \cup y \leftrightarrow (z \in x \vee z \in y))$

Some Theorems that can be proven from this

- $\forall x\ x \subseteq x$
- $\forall x\ \forall y\ (x = y \leftrightarrow (x \subseteq y \wedge y \subseteq x))$
- $\forall x\ \forall y\ x \cap y = y \cap x$
- $\forall x\ \forall y\ x \cup y = y \cup x$
- $\forall x\ \forall y\ \forall z\ x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$
- $\forall x\ \forall y\ \forall z\ x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$

Formal Proof Demo

Automated Theorem Proving

- Formal proofs seem perfect for automation: proofs require tediously many applications of precisely defined rules: just something a computer would be good at!
- Problem: the rules of logic are like the rules of chess: they tell you what you *can* do, but not what you *must* do.
- In Automated Theorem Proving (a branch of Artificial Intelligence) researchers try and come up with *algorithms* to create formal proofs.

How good are Automated Theorem Provers?

- Well, pretty disappointing, really!
- In 1956, things looked promising: the Logic Theorist was able to prove a theorem from Russell and Whitehead's book Principia Mathematica using a shorter proof than Russell and Whitehead themselves had found.
- This, by the way, is often seen as the birth of AI.
- However, 50 years later, the best ATP's around still can't prove that $P(\varnothing) = \{\varnothing\}$ given basic set theory axioms.
- Some researchers see this as evidence that human thought cannot be captured through computation (i.e. that AI is a pipe dream), but others say it's too early to tell.

Automated Theorem Proving Demo

Prolog

- Prolog ("PROgramming in LOGic") is a language often used by AI
- A Prolog program consists of 2 types of lines:
  - Facts: P. In Prolog: P.
  - Rules: $(P_1 \wedge \ldots \wedge P_n) \rightarrow Q$. In Prolog: Q :- P_1 , … , P_n.
- A Prolog program is run by asking whether some atomic statement Q follows from the facts and rules. In Prolog: Q?
- The Prolog program will answer 'Yes' or 'No'. (so 'No' means that Prolog is unable to infer Q, which is not the same as Q being false)

The Prolog Algorithm

- Prolog checks whether Q follows from the facts or rules as follows:
  - 1. Make a stack of goals G, starting with Q.
  - 2. If G is empty, stop with answer ‘Yes’.
  - 3. If a statement P is in G that is a fact, remove (‘pop’) P from G.
  - 4. If P is in G and there is a rule P :- P_1 , … , P_n, then remove (‘pop’) P from G, and add (‘push’) each P_i to G.
  - 5. If you get stuck, try a different rule P :- P_1 , … , P_n.
  - 6. If all options fail, stop with answer ‘No’.

Prolog Example

Putting into Prolog:

$H \Rightarrow H$

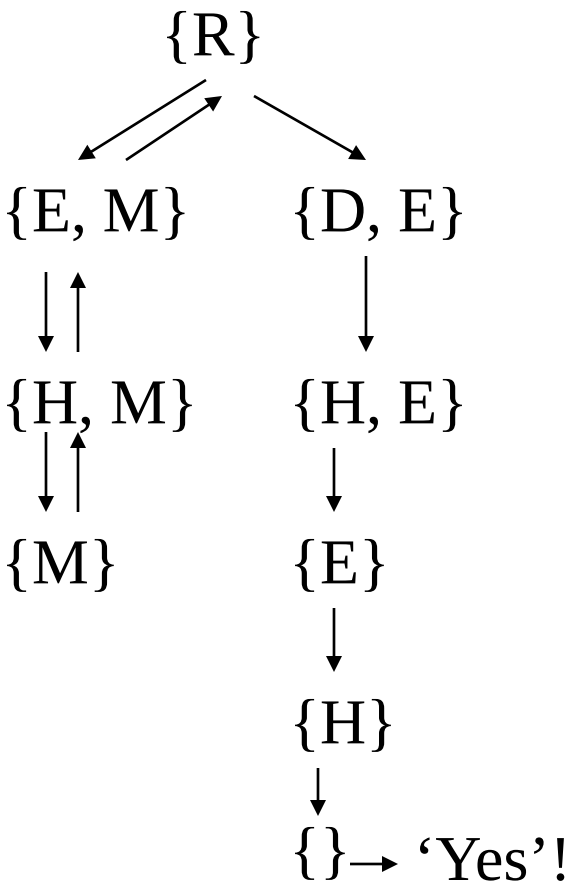
$H \rightarrow E \Rightarrow E$:- H.

$H \rightarrow D \Rightarrow D$:- H.

$(E \wedge M) \rightarrow R \Rightarrow R$:- E, M.

$(D \wedge E) \rightarrow R \Rightarrow R$:- D, E.

Query: R?



Prolog Demo

Summary

- Reasoning can be studied by logic
  - Deductive reasoning can be studied through formal logic
- In addition to being a tool for *analysis*, formal logic can be used as a tool for *synthesis*: use logic to *do* reasoning
  - Formal logic can be automated -> Automated Theorem Proving!
- However:
  - Formal proofs are long, and can be hard to construct. Indeed, Automated Theorem Provers still struggle with pretty elementary theorems (at least in context of mathematics)
  - Automated Theorem Proving = Automated Theorem Verification ≠ Automated Proof Generation ≠ Automated Theorem Generation
  - What about non-deductive reasoning?