Logic and Proof

Computer Science Tripos Part IB
Michaelmas Term

Lawrence C Paulson
Computer Laboratory
University of Cambridge

`lcp@cl.cam.ac.uk`

Contents

Slide 101

Introduction to Logic

Logic concerns *statements* in some *language*

The language can be informal (e.g. English) or *formal*

Some statements are *true*, others *false* or perhaps *meaningless*, . . .

Logic concerns relationships between statements: consistency, entailment, . . .

Logical *proofs* model human reasoning

Slide 102

Statements

Statements are declarative assertions:

Black is the colour of my true love's hair.

They are not greetings, questions, commands, . . . :

What is the colour of my true love's hair?

I wish my true love had hair.

Get a haircut!

Schematic Statements

The *meta-variables* $X, Y, Z, \ldots$ range over 'real' objects

> *Black is the colour of X's hair.*
>
> *Black is the colour of Y.*
>
> *Z is the colour of Y.*

Schematic statements can express general statements, or questions:

> *What things are black?*

Slide 103

Interpretations and Validity

An *interpretation* maps meta-variables to real objects

The interpretation $Y \mapsto \mathsf{coal}$ *satisfies* the statement

> *Black is the colour of Y.*

but the interpretation $Y \mapsto \mathsf{strawberries}$ does not!

A statement A is *valid* if all interpretations satisfy A.

Slide 104

Consistency, or Satisfiability

A set S of statements is *consistent* if some interpretation satisfies all elements of S at the same time. Otherwise S is *inconsistent.*

Examples of inconsistent sets:

$$\{X \text{ part of } Y, \ Y \text{ part of } Z, \ X \text{ NOT part of } Z\}$$

$$\{n \text{ is a positive integer}, \ n \neq 1, \ n \neq 2, \ \ldots\}$$

satisfiable/unsatisfiable = consistent/inconsistent

Slide 105

Entailment, or Logical Consequence

A set S of statements *entails* A if every interpretation that satisfies all elements of S, also satisfies A. We write $S \models A$.

$$\{X \text{ part of } Y, \ Y \text{ part of } Z\} \models X \text{ part of } Z$$

$$\{n \neq 1, \ n \neq 2, \ \ldots\} \models n \text{ is NOT a positive integer}$$

$S \models A$ if and only if $\{\neg A\} \cup S$ is inconsistent

$\models A$ if and only if A is valid

Slide 106

Inference

Want to check A is valid

Checking all interpretations can be effective — but if there are infinitely many?

Let $\{A_1, \dots, A_n\} \models B$. If $A_1, \dots, A_n$ are true then B must be true. Write this as the inference

$$\frac{A_1 \quad \dots \quad A_n}{B}$$

Use inferences to construct finite proofs!

Slide 107

Schematic Inference Rules

$$\frac{X \text{ part of } Y \quad Y \text{ part of } Z}{X \text{ part of } Z}$$

A valid inference:

$$\frac{\text{spoke part of wheel} \quad \text{wheel part of bike}}{\text{spoke part of bike}}$$

An inference may be valid even if the premises are false!

$$\frac{\text{cow part of chair} \quad \text{chair part of ant}}{\text{cow part of ant}}$$

Slide 108

Survey of Formal Logics

propositional logic is traditional *boolean algebra*.

first-order logic can say *for all* and *there exists*.

higher-order logic reasons about sets and functions. It has been
applied to hardware verification.

modal/temporal logics reason about what *must*, or *may*, happen.

type theories support *constructive* mathematics.

Syntax of Propositional Logic

$P, Q, R, \ldots$	propositional letter
t	true
f	false
$\neg A$	not A
$A \wedge B$	A and B
$A \vee B$	A or B
$A \rightarrow B$	if A then B
$A \leftrightarrow B$	A if and only if B

Semantics of Propositional Logic

$\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$ are *truth-functional*: functions of their operands

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
t	t	f	t	t	t	t
t	f	f	f	t	f	f
f	t	t	f	t	t	f
f	f	t	f	f	t	t

Slide 203

Interpretations of Propositional Logic

An *interpretation* is a function from the propositional letters to $\{\mathbf{t}, \mathbf{f}\}$.

Interpretation I *satisfies* a formula A if the formula evaluates to $\mathbf{t}$.

$$\text{Write} \models_I A$$

A is *valid* (a *tautology*) if every interpretation satisfies A

$$\text{Write} \models A$$

S is *satisfiable* if some interpretation satisfies every formula in S

Slide 204

Implication, Entailment, Equivalence

$A \rightarrow B$ means simply $\neg A \vee B$

$A \models B$ means if $\models_I A$ then $\models_I B$ for every interpretation I

$A \models B$ if and only if $\models A \rightarrow B$

Equivalence

$A \simeq B$ means $A \models B$ and $B \models A$

$A \simeq B$ if and only if $\models A \leftrightarrow B$

Equivalences

$$A \wedge A \simeq A$$

$$A \wedge B \simeq B \wedge A$$

$$(A \wedge B) \wedge C \simeq A \wedge (B \wedge C)$$

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

$$A \wedge \mathbf{f} \simeq \mathbf{f}$$

$$A \wedge \mathbf{t} \simeq A$$

$$A \wedge \neg A \simeq \mathbf{f}$$

Dual versions: exchange $\wedge$, $\vee$ and $\mathbf{t}$, $\mathbf{f}$ in any equivalence

Slide 205

Negation Normal Form

1. Get rid of $\leftrightarrow$ and $\rightarrow$, leaving just $\wedge$, $\vee$, $\neg$:

$$A \leftrightarrow B \simeq (A \rightarrow B) \wedge (B \rightarrow A)$$

$$A \rightarrow B \simeq \neg A \vee B$$

2. Push negations in, using de Morgan's laws:

$$\neg\neg A \simeq A$$

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

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

Slide 206

From NNF to Conjunctive Normal Form

3. Push disjunctions in, using distributive laws:

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

4. Simplify:

- Delete any disjunction containing P and $\neg P$

- Delete any disjunction that includes another

- Replace $(P \vee A) \wedge (\neg P \vee A)$ by A

Converting a Non-Tautology to CNF

$$P \vee Q \to Q \vee R$$

1. Elim $\to$: $\neg(P \vee Q) \vee (Q \vee R)$

2. Push $\neg$ in: $(\neg P \wedge \neg Q) \vee (Q \vee R)$

3. Push $\vee$ in: $(\neg P \vee Q \vee R) \wedge (\neg Q \vee Q \vee R)$

4. Simplify: $\neg P \vee Q \vee R$

Not a tautology: try $P \mapsto \mathbf{t}$, $Q \mapsto \mathbf{f}$, $R \mapsto \mathbf{f}$

Slide 209

<div style="border:2px solid black; padding:1em;">

Tautology checking using CNF

$$((P \rightarrow Q) \rightarrow P) \rightarrow P$$

1. Elim $\rightarrow$: $\neg[\neg(\neg P \vee Q) \vee P] \vee P$

2. Push $\neg$ in: $[\neg\neg(\neg P \vee Q) \wedge \neg P] \vee P$

                    $[(\neg P \vee Q) \wedge \neg P] \vee P$

3. Push $\vee$ in: $(\neg P \vee Q \vee P) \wedge (\neg P \vee P)$

4. Simplify: $\mathbf{t} \wedge \mathbf{t}$

                    $\mathbf{t}$ *It's a tautology!*

</div>

A Simple Proof System

Axiom Schemes

K $A \rightarrow (B \rightarrow A)$

S $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

DN $\neg\neg A \rightarrow A$

Inference Rule: Modus Ponens

$$\frac{A \rightarrow B \qquad A}{B}$$

Slide 301

A Simple (?) Proof of $A \rightarrow A$

$(A \rightarrow ((D \rightarrow A) \rightarrow A)) \rightarrow$ (1)

$((A \rightarrow (D \rightarrow A)) \rightarrow (A \rightarrow A))$ by S

$A \rightarrow ((D \rightarrow A) \rightarrow A)$ by K   (2)

$(A \rightarrow (D \rightarrow A)) \rightarrow (A \rightarrow A)$ by MP, (1), (2)   (3)

$A \rightarrow (D \rightarrow A)$ by K   (4)

$A \rightarrow A$ by MP, (3), (4)   (5)

Slide 302

Slide 303

Some Facts about Deducibility

A is *deducible from* the set S of if there is a finite proof of A starting from elements of S. Write $S \vdash A$.

Soundness Theorem. If $S \vdash A$ then $S \models A$.

Completeness Theorem. If $S \models A$ then $S \vdash A$.

Deduction Theorem. If $S \cup \{A\} \vdash B$ then $S \vdash A \rightarrow B$.

Slide 304

Gentzen's Natural Deduction Systems

A varying context of *assumptions*

Each logical connective defined *independently*

Introduction rule for $\wedge$: how to deduce $A \wedge B$

$$\frac{A \quad B}{A \wedge B}$$

Elimination rules for $\wedge$: what to deduce *from* $A \wedge B$

$$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

The Sequent Calculus

Sequent $A_1, \ldots, A_m \Rightarrow B_1, \ldots, B_n$ means,

$$\text{if } A_1 \wedge \ldots \wedge A_m \text{ then } B_1 \vee \ldots \vee B_n$$

$A_1, \ldots, A_m$ are *assumptions*; $B_1, \ldots, B_n$ are *goals*

Γ and Δ are *sets* in $\Gamma \Rightarrow \Delta$

$A, \Gamma \Rightarrow A, \Delta$ is trivially true (*basic sequent*)

Slide 305

Sequent Calculus Rules

$$\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \ (\text{cut})$$

$$\frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta} \ (\neg l) \qquad \frac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A} \ (\neg r)$$

$$\frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta} \ (\wedge l) \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B} \ (\wedge r)$$

Slide 306

Slide 307

More Sequent Calculus Rules

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta} \; (\vee l) \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B} \; (\vee r)$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta} \; (\rightarrow l) \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B} \; (\rightarrow r)$$

Slide 308

Easy Sequent Calculus Proofs

$$\frac{\overline{A, B \Rightarrow A}}{\dfrac{A \wedge B \Rightarrow A}{\Rightarrow A \wedge B \rightarrow A}} \begin{array}{l} (\wedge l) \\ (\rightarrow r) \end{array}$$

$$\frac{\overline{A, B \Rightarrow B, A}}{\dfrac{A \Rightarrow B, B \rightarrow A}{\dfrac{\Rightarrow A \rightarrow B, \; B \rightarrow A}{\Rightarrow (A \rightarrow B) \vee (B \rightarrow A)}}} \begin{array}{l} (\rightarrow r) \\ (\rightarrow r) \\ (\vee r) \end{array}$$

Part of a Distributive Law

$$
\cfrac{
  \cfrac{
    \cfrac{}{A \Rightarrow A, B}
    \qquad
    \cfrac{
      \cfrac{}{\overline{B, C \Rightarrow A, B}}
    }{B \land C \Rightarrow A, B} \; (\land l)
  }{
    \cfrac{
      A \lor (B \land C) \Rightarrow A, B
    }{A \lor (B \land C) \Rightarrow A \lor B} \; (\lor r)
  } \; (\lor l)
  \qquad \text{similar}
}{A \lor (B \land C) \Rightarrow (A \lor B) \land (A \lor C)} \; (\land r)
$$

Second subtree proves $A \lor (B \land C) \Rightarrow A \lor C$ similarly

---

A Failed Proof

$$
\cfrac{
  \cfrac{
    \cfrac{
      A \Rightarrow B, C \qquad \overline{B \Rightarrow B, C}
    }{A \lor B \Rightarrow B, C} \; (\lor l)
  }{A \lor B \Rightarrow B \lor C} \; (\lor r)
}{\Rightarrow A \lor B \rightarrow B \lor C} \; (\rightarrow r)
$$

$A \mapsto \mathbf{t}$, $B \mapsto \mathbf{f}$, $C \mapsto \mathbf{f}$ falsifies unproved sequent!

Ordered Binary Decision Diagrams

Canonical form: essentially decision trees with sharing
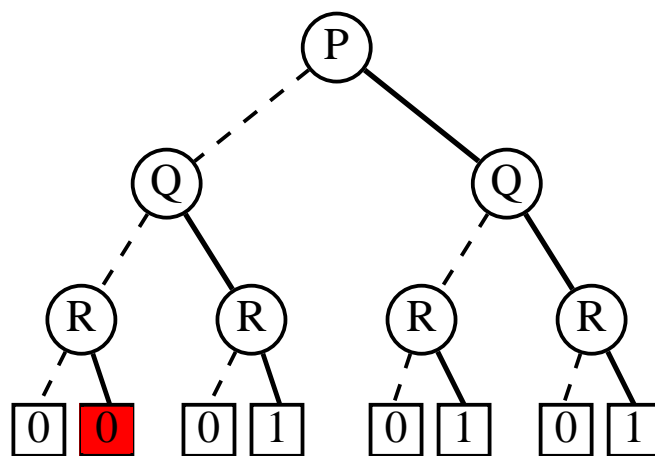
- *ordered* propositional symbols ('variables')

- *sharing* of identical subtrees

- *hashing* and other optimisations

Detects if a formula is tautologous ($\mathbf{t}$) or inconsistent ($\mathbf{f}$)

A **FAST** way of verifying digital circuits, . . .

Slide 401

Decision Diagram for $(P \lor Q) \land R$

Slide 402

Slide 403

Converting a Decision Diagram to an OBDD
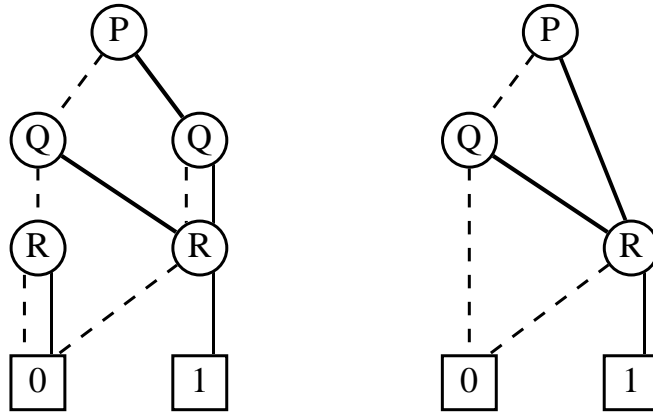
No duplicates   No redundant tests

Slide 404

Building OBDDs Efficiently

Do not construct full tree! (see Bryant, §3.1)

Do not expand $\rightarrow$, $\leftrightarrow$, $\oplus$ (exclusive OR) to other connectives

Treat $\neg Z$ as $Z \rightarrow \mathbf{f}$ or $Z \oplus \mathbf{t}$

Recursively convert operands

Combine operand OBDDs — respecting ordering and sharing

Delete test if it proves to be redundant

Canonical Form Algorithm

To do $Z \wedge Z'$, where Z and Z' are already canonical:

Trivial if either is $\mathbf{t}$ or $\mathbf{f}$. Treat $\vee$, $\rightarrow$, $\leftrightarrow$ similarly!

Let $Z = \mathbf{if}(P, X, Y)$ and $Z' = \mathbf{if}(P', X', Y')$

If $P = P'$ then recursively do $\mathbf{if}(P, X \wedge X', Y \wedge Y')$

If $P < P'$ then recursively do $\mathbf{if}(P, X \wedge Z', Y \wedge Z')$

If $P > P'$ then recursively do $\mathbf{if}(P', Z \wedge X', Z \wedge Y')$

Slide 405

Canonical Form of $P \vee Q$

Slide 406

Slide 407

Canonical Form of $P \vee Q \rightarrow Q \vee R$
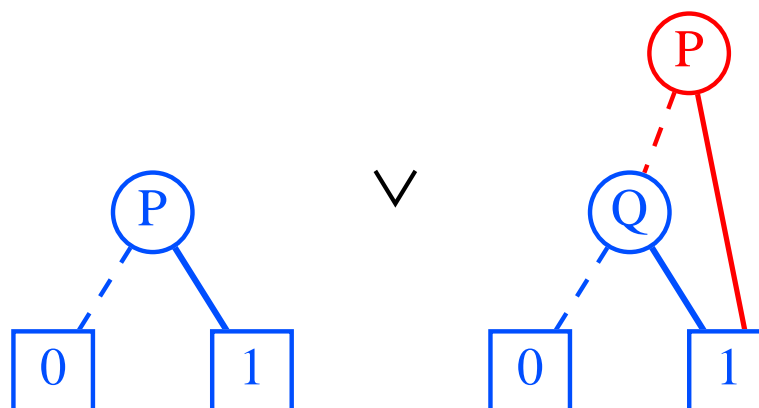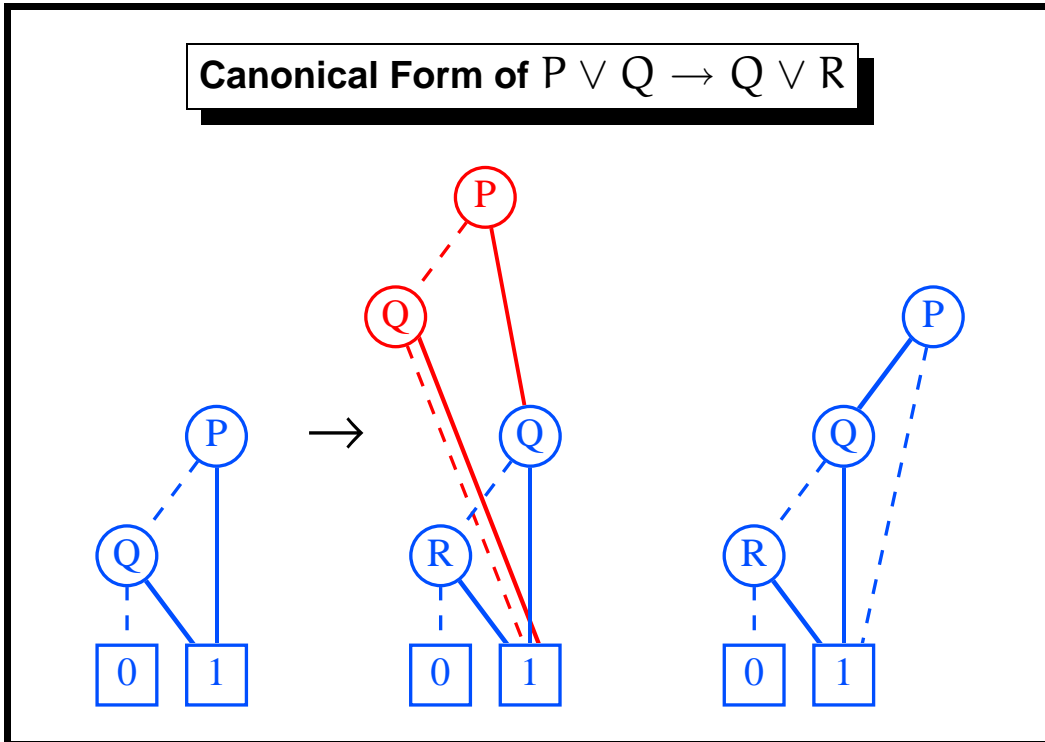
Slide 408

Optimisations Based On Hash Tables

Never build the same OBDD twice: share pointers

- Pointer identity: $X = Y$ whenever $X \leftrightarrow Y$

- Fast removal of redundant tests by $\mathbf{if}(P, X, X) \simeq X$

- Fast processing of $X \wedge X, X \vee X, X \rightarrow X, \ldots$

Never process $X \wedge Y$ twice; keep table of canonical forms

Slide 409

Final Observations

The variable ordering is crucial. Consider

$$(P_1 \wedge Q_1) \vee \cdots \vee (P_n \wedge Q_n)$$

A good ordering is $P_1 < Q_1 < \cdots < P_n < Q_n$

A dreadful ordering is $P_1 < \cdots < P_n < Q_1 < \cdots < Q_n$

Many digital circuits have small OBDDs *(not multiplication!)*

OBDDs can solve problems in hundreds of variables

General case remains intractable!

Outline of First-Order Logic

Reasons about *functions* and *relations* over a set of *individuals*

$$\frac{\text{father}(\text{father}(x)) = \text{father}(\text{father}(y))}{\text{cousin}(x, y)}$$

Reasons about *all* and *some* individuals:

$$\frac{\text{All men are mortal} \quad \text{Socrates is a man}}{\text{Socrates is mortal}}$$

Does not reason about *all functions* or *all relations*, . . .

Slide 501

Function Symbols; Terms

Each *function symbol* stands for an n-place function

A *constant symbol* is a 0-place function symbol

A *variable* ranges over all individuals

A *term* is a variable, constant or has the form

$$f(t_1, \dots, t_n)$$

where f is an n-place function symbol and $t_1, \dots, t_n$ are terms

We choose the language, adopting any desired function symbols

Slide 502

Relation Symbols; Formulae

Slide 503

Each *relation symbol* stands for an n-place relation

Equality is the 2-place relation symbol $=$

An *atomic formula* has the form

$$R(t_1, \ldots, t_n)$$

where R is an n-place relation symbol and $t_1, \ldots, t_n$ are terms

A *formula* is built up from atomic formulæ using $\neg$, $\wedge$, $\vee$, . . .

(Later we add *quantifiers*)

Power of Quantifier-Free FOL

Slide 504

Very expressive, given strong induction rules

Prove equivalence of mathematical functions:

$$p(z, 0) = 1 \qquad\qquad q(z, 1) = z$$
$$p(z, n + 1) = p(z, n) \times z \qquad q(z, 2 \times n) = q(z \times z, n)$$
$$q(z, 2 \times n + 1) = q(z \times z, n) \times z$$

Boyer/Moore Theorem Prover: checked Gödel's Theorem, . . .

Many systems based on *equational reasoning*

Universal and Existential Quantifiers

$$\forall x\, A \quad \text{for all } x, \; A \text{ holds}$$

$$\exists x\, A \quad \text{there exists } x \text{ such that } A \text{ holds}$$

Syntactic variations:

$$\forall xyz\, A \quad \text{abbreviates } \forall x\, \forall y\, \forall z\, A$$

$$\forall z\,.\, A \wedge B \quad \text{is an alternative to } \forall z\, (A \wedge B)$$

The variable x is *bound* in $\forall x\, A$; compare with $\int f(x)\,dx$

Slide 505

Expressiveness of Quantifiers

All men are mortal:

$$\forall x\, (\mathrm{man}(x) \rightarrow \mathrm{mortal}(x))$$

All mothers are female:

$$\forall x\, \mathrm{female}(\mathrm{mother}(x))$$

There exists a unique x *such that* A, written $\exists! x\, A$

$$\exists x\, [A(x) \wedge \forall y\, (A(y) \rightarrow y = x)]$$

Slide 506

Slide 507

How do we interpret mortal(Socrates)**?**

Interpretation $\mathcal{I} = (D, I)$ of our first-order language

D is a non-empty *universe*

I maps symbols to 'real' functions, relations

c a constant symbol $\qquad\qquad$ $I[c] \in D$

f an n-place function symbol $\quad$ $I[f] \in D^n \to D$

P an n-place relation symbol $\quad$ $I[P] \subseteq D^n$

Slide 508

How do we interpret cousin(Charles, y)**?**

A *valuation* supplies the values of free variables

It is a function $V : \mathrm{variables} \to D$

$\mathcal{I}_V[t]$ extends V to a term t by the obvious recursion:

$$\mathcal{I}_V[x] \stackrel{\mathrm{def}}{=} V(x) \qquad \text{if } x \text{ is a variable}$$

$$\mathcal{I}_V[c] \stackrel{\mathrm{def}}{=} I[c]$$

$$\mathcal{I}_V[f(t_1, \dots, t_n)] \stackrel{\mathrm{def}}{=} I[f](\mathcal{I}_V[t_1], \dots, \mathcal{I}_V[t_n])$$

Slide 509

The Meaning of Truth — in FOL

For interpretation $\mathcal{I}$ and valuation V

$$\models_{\mathcal{I},V} P(t) \qquad \text{if } I[P](\mathcal{I}_V[t]) \text{ holds}$$

$$\models_{\mathcal{I},V} t = u \qquad \text{if } \mathcal{I}_V[t] \text{ equals } \mathcal{I}_V[u]$$

$$\models_{\mathcal{I},V} A \wedge B \qquad \text{if } \models_{\mathcal{I},V} A \text{ and } \models_{\mathcal{I},V} B$$

$$\models_{\mathcal{I},V} \exists x\, A \qquad \text{if } \models_{\mathcal{I},V\{m/x\}} A \text{ holds for some } m \in D$$

$$\models_{\mathcal{I}} A \qquad \text{if } \models_{\mathcal{I},V} A \text{ holds for all } V$$

A is *satisfiable* if $\models_{\mathcal{I}} A$ for some $\mathcal{I}$

Free v Bound Variables

All occurrences of x in $\forall x\, A$ and $\exists x\, A$ are *bound*

An occurrence of x is *free* if it is not bound:

$$\forall x\, \exists y\, R(x, y, f(x, z))$$

May *rename* bound variables:

$$\forall w\, \exists y'\, R(w, y', f(w, z))$$

Slide 601

Substitution for Free Variables

$A[t/x]$ means *substitute* t *for* x *in* A:

$$
\begin{aligned}
(B \wedge C)[t/x] \quad &is \quad B[t/x] \wedge C[t/x] \\
(\forall x\, B)[t/x] \quad &is \quad \forall x\, B \\
(\forall y\, B)[t/x] \quad &is \quad \forall y\, B[t/x] \qquad (x \neq y) \\
(P(u))[t/x] \quad &is \quad P(u[t/x])
\end{aligned}
$$

No variable in t may be bound in A!

$(\forall y\, x = y)[y/x]$ *is not* $\forall y\, y = y$!

Slide 602

Some Equivalences for Quantifiers

$$\neg(\forall x\, A) \simeq \exists x\, \neg A$$

$$(\forall x\, A) \wedge B \simeq \forall x\, (A \wedge B)$$

Slide 603

$$(\forall x\, A) \vee B \simeq \forall x\, (A \vee B)$$

$$(\forall x\, A) \wedge (\forall x\, B) \simeq \forall x\, (A \wedge B)$$

$$(\forall x\, A) \rightarrow B \simeq \exists x\, (A \rightarrow B)$$

$$\forall x\, A \simeq \forall x\, A \wedge A[t/x]$$

Dual versions: exchange $\forall$, $\exists$ and $\wedge$, $\vee$

Reasoning by Equivalences

$$\exists x\, (x = a \wedge P(x)) \simeq \exists x\, (x = a \wedge P(a))$$

$$\simeq \exists x\, (x = a) \wedge P(a)$$

$$\simeq P(a)$$

Slide 604

$$\exists z\, (P(z) \rightarrow P(a) \wedge P(b))$$

$$\simeq \forall z\, P(z) \rightarrow P(a) \wedge P(b)$$

$$\simeq \forall z\, P(z) \wedge P(a) \wedge P(b) \rightarrow P(a) \wedge P(b)$$

$$\simeq t$$

Sequent Calculus Rules for $\forall$

$$\frac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x\, A, \Gamma \Rightarrow \Delta} \ (\forall l) \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \forall x\, A} \ (\forall r)$$

Rule $(\forall l)$ can create many instances of $\forall x\, A$

Rule $(\forall r)$ holds *provided* x is not free in the conclusion!

NOT allowed to prove

$$\frac{\overline{P(y) \Rightarrow P(y)}}{P(y) \Rightarrow \forall y\, P(y)} \ (\forall r)$$

Slide 605

Simple Example of the $\forall$ Rules

$$\frac{\dfrac{\overline{P(f(y)) \Rightarrow P(f(y))}}{\forall x\, P(x) \Rightarrow P(f(y))} \ (\forall l)}{\forall x\, P(x) \Rightarrow \forall y\, P(f(y))} \ (\forall r)$$

Slide 606

Not-So-Simple Example of the $\forall$ Rules

$$\dfrac{\dfrac{\overline{P \Rightarrow Q(y), P} \qquad \overline{P, Q(y) \Rightarrow Q(y)}}{\dfrac{\dfrac{P, P \rightarrow Q(y) \Rightarrow Q(y)}{P, \forall x\,(P \rightarrow Q(x)) \Rightarrow Q(y)}}{\dfrac{P, \forall x\,(P \rightarrow Q(x)) \Rightarrow \forall y\, Q(y)}{\forall x\,(P \rightarrow Q(x)) \Rightarrow P \rightarrow \forall y\, Q(y)}}}}{}$$

$(\rightarrow l)$
$(\forall l)$
$(\forall r)$
$(\rightarrow r)$

In $(\forall l)$ we have replaced x by y

Sequent Calculus Rules for $\exists$

$$\dfrac{A, \Gamma \Rightarrow \Delta}{\exists x\, A, \Gamma \Rightarrow \Delta} \;(\exists l) \qquad \dfrac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x\, A} \;(\exists r)$$

Rule $(\exists l)$ holds *provided* x is not free in the conclusion!

Rule $(\exists r)$ can create many instances of $\exists x\, A$

Say, to prove

$$\exists z\,(P(z) \rightarrow P(a) \wedge P(b))$$

Slide 609

Part of the ∃ Distributive Law

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \overline{P(x) \Rightarrow P(x), Q(x)}
      }{P(x) \Rightarrow P(x) \lor Q(x)} \; (\lor r)
    }{P(x) \Rightarrow \exists y\,(P(y) \lor Q(y))} \; (\exists r)
  }{\exists x\, P(x) \Rightarrow \exists y\,(P(y) \lor Q(y))} \; (\exists l)
  \qquad
  \cfrac{similar}{\exists x\, Q(x) \Rightarrow \exists y\,\dots} \; (\exists l)
}{\exists x\, P(x) \lor \exists x\, Q(x) \Rightarrow \exists y\,(P(y) \lor Q(y))} \; (\lor l)
$$

Second subtree proves $\exists x\, Q(x) \Rightarrow \exists y\,(P(y) \lor Q(y))$ similarly

In $(\exists r)$ we have replaced y by x

Slide 610

A Failed Proof

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        P(x), Q(y) \Rightarrow P(x) \land Q(x)
      }{P(x), Q(y) \Rightarrow \exists z\,(P(z) \land Q(z))} \; (\exists r)
    }{P(x), \exists x\, Q(x) \Rightarrow \exists z\,(P(z) \land Q(z))} \; (\exists l)
  }{\exists x\, P(x), \exists x\, Q(x) \Rightarrow \exists z\,(P(z) \land Q(z))} \; (\exists l)
}{\exists x\, P(x) \land \exists x\, Q(x) \Rightarrow \exists z\,(P(z) \land Q(z))} \; (\land l)
$$

We cannot use $(\exists l)$ twice with the same variable

We rename the bound variable in $\exists x\, Q(x)$ and get $\exists y\, Q(y)$

Clause Form

Clause: a disjunction of *literals*

$$\neg K_1 \vee \cdots \vee \neg K_m \vee L_1 \vee \cdots \vee L_n$$

Set notation: $\{\neg K_1, \ldots, \neg K_m, L_1, \ldots, L_n\}$

Kowalski notation: $K_1, \cdots, K_m \rightarrow L_1, \cdots, L_n$

$$L_1, \cdots, L_n \leftarrow K_1, \cdots, K_m$$

Empty clause $\square$

Empty clause means contradiction!

Slide 701

Outline of Clause Form Methods

To prove A, obtain a contradiction from $\neg A$:

1. Translate $\neg A$ into CNF as $A_1 \wedge \cdots \wedge A_m$

2. This is the set of clauses $A_1, \ldots, A_m$

3. Transform the clause set, preserving consistency

Empty *clause* refutes $\neg A$

Empty *clause set* means $\neg A$ is satisfiable

Slide 702

The Davis-Putnam Decision Procedure

1. Delete tautological clauses: $\{P, \neg P, \dots\}$

2. For each unit clause $\{L\}$,

   • delete all clauses containing L

   • delete $\neg L$ from all clauses

3. Delete all clauses containing *pure literals*

4. Perform a *case split* on some literal

Davis-Putnam on a Non-Tautology

Consider $P \vee Q \rightarrow Q \vee R$

Clauses are $\{P, Q\}$ $\{\neg Q\}$ $\{\neg R\}$

$\{P, Q\}$	$\{\neg Q\}$	$\{\neg R\}$	initial clauses
$\{P\}$		$\{\neg R\}$	unit $\neg Q$
		$\{\neg R\}$	unit P (also pure)
			unit $\neg R$ (also pure)

Clauses satisfiable by $P \mapsto \mathbf{t}$, $Q \mapsto \mathbf{f}$, $R \mapsto \mathbf{f}$

Example of a Case Split on P

$\{\neg Q, R\}$ $\{\neg R, P\}$ $\{\neg R, Q\}$ $\{\neg P, Q, R\}$ $\{P, Q\}$ $\{\neg P, \neg Q\}$

$\{\neg Q, R\}$	$\{\neg R, Q\}$	$\{Q, R\}$	$\{\neg Q\}$	if P is true
	$\{\neg R\}$	$\{R\}$		unit $\neg Q$
	$\square$			unit R

$\{\neg Q, R\}$	$\{\neg R\}$	$\{\neg R, Q\}$	$\{Q\}$	if P is false
$\{\neg Q\}$			$\{Q\}$	unit $\neg R$
			$\square$	unit $\neg Q$

The Resolution Rule

From $B \lor A$ and $\neg B \lor C$ infer $A \lor C$

In set notation,

$$\frac{\{B, A_1, \dots, A_m\} \quad \{\neg B, C_1, \dots, C_n\}}{\{A_1, \dots, A_m, C_1, \dots, C_n\}}$$

Some special cases:

$$\frac{\{B\} \quad \{\neg B, C_1, \dots, C_n\}}{\{C_1, \dots, C_n\}} \qquad \frac{\{B\} \quad \{\neg B\}}{\square}$$

Slide 707

Simple Example: Proving $P \wedge Q \rightarrow Q \wedge P$

Hint: use $\neg(A \rightarrow B) \simeq A \wedge \neg B$

$\qquad$ 1. Negate! $\qquad \neg[P \wedge Q \rightarrow Q \wedge P]$

$\qquad$ 2. Push $\neg$ in: $\quad (P \wedge Q) \wedge \neg(Q \wedge P)$

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

$\qquad$ Clauses: $\qquad \{P\} \quad \{Q\} \quad \{\neg Q, \neg P\}$

Resolve $\{P\}$ and $\{\neg Q, \neg P\}$ getting $\{\neg Q\}$

Resolve $\{Q\}$ and $\{\neg Q\}$ getting $\square$

Slide 708

Another Example

Refute $\neg[(P \vee Q) \wedge (P \vee R) \rightarrow P \vee (Q \wedge R)]$

From $(P \vee Q) \wedge (P \vee R)$, get clauses $\{P, Q\}$ and $\{P, R\}$

From $\neg[P \vee (Q \wedge R)]$ get clauses $\{\neg P\}$ and $\{\neg Q, \neg R\}$

Resolve $\{\neg P\}$ and $\{P, Q\}$ getting $\{Q\}$

Resolve $\{\neg P\}$ and $\{P, R\}$ getting $\{R\}$

Resolve $\{Q\}$ and $\{\neg Q, \neg R\}$ getting $\{\neg R\}$

Resolve $\{R\}$ and $\{\neg R\}$ getting $\square$

Refinements of Resolution

Preprocessing: removing tautologies, symmetries . . .

Set of Support: working from the goal

Weighting: priority to the smallest clauses

Subsumption: deleting redundant clauses

Hyper-resolution: avoiding intermediate clauses

Indexing: data structures for speed

Reducing FOL to Propositional Logic

Prenex: Move quantifiers to the front

Skolemize: Remove quantifiers, preserving consistency

Herbrand models: Reduce the class of interpretations

Herbrand's Thm: Contradictions have *finite*, *ground* proofs

Unification: Automatically find the right instantiations

Finally, combine unification with *resolution*

Prenex Normal Form

Convert to Negation Normal Form using additionally

$$\neg(\forall x\, A) \simeq \exists x\, \neg A$$

$$\neg(\exists x\, A) \simeq \forall x\, \neg A$$

Then move quantifiers to the front using

$$(\forall x\, A) \wedge B \simeq \forall x\, (A \wedge B)$$

$$(\forall x\, A) \vee B \simeq \forall x\, (A \vee B)$$

and the similar rules for $\exists$

Slide 803

Skolemization

Take a formula of the form

$$\forall x_1 \, \forall x_2 \, \cdots \forall x_k \, \exists y \, A$$

Choose a new k-place function symbol, say f

Delete $\exists y$ and replace y by $f(x_1, x_2, \ldots, x_k)$. We get

$$\forall x_1 \, \forall x_2 \, \cdots \forall x_k \, A[f(x_1, x_2, \ldots, x_k)/y]$$

Repeat until no $\exists$ quantifiers remain

Slide 804

Example of Conversion to Clauses

For proving $\exists x \, [P(x) \rightarrow \forall y \, P(y)]$

$\neg \, [\exists x \, [P(x) \rightarrow \forall y \, P(y)]]$ negated goal

$\forall x \, [P(x) \wedge \exists y \, \neg P(y)]$ conversion to NNF

$\forall x \, \exists y \, [P(x) \wedge \neg P(y)]$ pulling $\exists$ out

$\forall x \, [P(x) \wedge \neg P(f(x))]$ Skolem term $f(x)$

$\{P(x)\} \quad \{\neg P(f(x))\}$ Final clauses

Correctness of Skolemization

The formula $\forall x\, \exists y\, A$ is consistent

$\Longleftrightarrow$ it holds in some interpretation $\mathcal{I} = (D, I)$

$\Longleftrightarrow$ for all $x \in D$ there is some $y \in D$ such that A holds

$\Longleftrightarrow$ some function $\hat{f}$ in $D \to D$ yields suitable values of y

$\Longleftrightarrow$ $A[f(x)/y]$ holds in some $\mathcal{I}'$ extending $\mathcal{I}$ so that f denotes $\hat{f}$

$\Longleftrightarrow$ the formula $\forall x\, A[f(x)/y]$ is consistent.

Slide 805

Herbrand Interpretations for a set of clauses S

$$H_0 \overset{\text{def}}{=} \text{the set of constants in } S$$

$$H_{i+1} \overset{\text{def}}{=} H_i \cup \{f(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in H_i$$

$$\text{and } f \text{ is an } n\text{-place function symbol in } S\}$$

$$H \overset{\text{def}}{=} \bigcup_{i \geq 0} H_i \quad \textit{Herbrand Universe}$$

$$HB \overset{\text{def}}{=} \{P(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in H$$

$$\text{and } P \text{ is an } n\text{-place predicate symbol in } S\}$$

Slide 806

Example of an Herbrand Model

$$\left. \begin{array}{l} \neg even(1) \\ even(2) \\ even(X \cdot Y) \leftarrow even(X), even(Y) \end{array} \right\} \text{\textit{clauses}}$$

$$H = \{1, 2, 1 \cdot 1, 1 \cdot 2, 2 \cdot 1, 2 \cdot 2, 1 \cdot (1 \cdot 1), \ldots\}$$

$$HB = \{even(1), even(2), even(1 \cdot 1), even(1 \cdot 2), \ldots\}$$

$$I[even] = \{even(2), even(1 \cdot 2), even(2 \cdot 1), even(2 \cdot 2), \ldots\}$$

(for model where · means product; could instead use sum!)

A Key Fact about Herbrand Interpretations

Let S *be a set of clauses.*

S *is unsatisfiable* $\Longleftrightarrow$ *no Herbrand interpretation satisfies* S

- Holds because some Herbrand model mimicks every 'real' model

- We must consider only a small class of models

- Herbrand models are syntactic, easily processed by computer

Slide 809

Herbrand's Theorem

Let S be a set of clauses.

S is unsatisfiable $\iff$ there is a finite *unsatisfiable set S' of* ground instances *of clauses of S.*

- **Finite**: we can compute it

- **Instance**: result of substituting for variables

- **Ground**: and no variables remain: it's propositional!

Unification

Finding a common instance of two terms

- Logic programming (Prolog)

- Polymorphic type-checking (ML)

- Constraint satisfaction problems

- Resolution theorem proving for FOL

- Many other theorem proving methods

Slide 901

Substitutions

A finite set of *replacements*

$$\theta = [t_1/x_1, \ldots, t_k/x_k]$$

where $x_1, \ldots, x_k$ are distinct variables and $t_i \neq x_i$

$$f(t, u)\theta = f(t\theta, u\theta) \qquad \text{(terms)}$$

$$P(t, u)\theta = P(t\theta, u\theta) \qquad \text{(literals)}$$

$$\{L_1, \ldots, L_m\}\theta = \{L_1\theta, \ldots, L_m\theta\} \qquad \text{(clauses)}$$

Slide 902

Slide 903

Composing Substitutions

Composition of ϕ and θ, written $\phi \circ \theta$, satisfies for all terms t

$$t(\phi \circ \theta) = (t\phi)\theta$$

It is defined by (for all relevant x)

$$\phi \circ \theta \stackrel{\text{def}}{=} [\,(x\phi)\theta \,/\, x, \dots\,]$$

Consequences include $\theta \circ [] = \theta$, and *associativity*:

$$(\phi \circ \theta) \circ \sigma = \phi \circ (\theta \circ \sigma)$$

Slide 904

Most General Unifiers

θ is a *unifier* of terms t and u if $t\theta = u\theta$

θ is *more general* than ϕ if $\phi = \theta \circ \sigma$

θ is *most general* if it is more general than every other unifier

If θ unifies t and u then so does $\theta \circ \sigma$:

$$t(\theta \circ \sigma) = t\theta\sigma = u\theta\sigma = u(\theta \circ \sigma)$$

A most general unifier of $f(a, x)$ and $f(y, g(z))$ is $[a/y, g(z)/x]$

The common instance is $f(a, g(z))$

Algorithm for Unifying Two Terms

Represent terms by *binary trees*

Each term is a *Variable* $x, y \ldots$, *Constant* $a, b \ldots$, or *Pair* (t, t')

Constants do not unify with different Constants

Constants do not unify with Pairs

Variable x and term t: unifier is $[t/x]$ — **unless** x occurs in t

Cannot unify $f(x)$ **with** x**!**

Unifying Two Pairs

$\theta \circ \theta'$ unifies (t, t') with (u, u')

if θ unifies t with u and θ' unifies $t'\theta$ with $u'\theta$

$$
\begin{aligned}
(t, t')(\theta \circ \theta') &= (t, t')\theta\theta' \\
&= (t\theta\theta', t'\theta\theta') \\
&= (u\theta\theta', u'\theta\theta') \\
&= (u, u')\theta\theta' \\
&= (u, u')(\theta \circ \theta')
\end{aligned}
$$

Slide 905

Slide 906

Examples of Unification

$f(x, b)$	$f(x, x)$	$f(x, x)$	$j(x, x, z)$
$f(a, y)$	$f(a, b)$	$f(y, g(y))$	$j(w, a, h(w))$
$f(a, b)$	?	?	$j(a, a, h(a))$
$[a/x, b/y]$	**FAIL**	**FAIL**	$[a/w, a/x, h(a)/z]$

We always get a **most general** unifier

Theorem-Proving Examples

$$(\exists y \, \forall x \, R(x, y)) \rightarrow (\forall x \, \exists y \, R(x, y))$$

Clauses after negation are $\{R(x, a)\}$ and $\{\neg R(b, y)\}$

$R(x, a)$ and $R(b, y)$ have unifier $[b/x, a/y]$: *contradiction!*

$$(\forall x \, \exists y \, R(x, y)) \rightarrow (\exists y \, \forall x \, R(x, y))$$

Clauses after negation are $\{R(x, f(x))\}$ and $\{\neg R(g(y), y)\}$

$R(x, f(x))$ and $R(g(y), y)$ are not unifiable: *occurs check*

Formula is not a theorem!

Slide 907

Slide 908

Variations on Unification

Efficient unification algorithms: near-linear time

Indexing & Discrimination networks: fast retrieval of a unifiable term

Order-sorted unification: type-checking in Haskell

Associative/commutative operators: problems in group theory

Higher-order unification: support λ-calculus

Boolean unification: reasoning about sets

Binary Resolution

$$\frac{\{B, A_1, \dots, A_m\} \quad \{\neg D, C_1, \dots, C_n\}}{\{A_1, \dots, A_m, C_1, \dots, C_n\}\sigma} \qquad \textit{provided } B\sigma = D\sigma$$

Slide 1001

First *rename variables apart* in the clauses! — say, to resolve

$$\{P(x)\} \quad \text{and} \quad \{\neg P(g(x))\}$$

Always use a *most general* unifier (MGU)

Soundness? Same argument as for the propositional version

Factorisation

Collapsing similar literals *in one clause*:

$$\frac{\{B_1, \dots, B_k, A_1, \dots, A_m\}}{\{B_1, A_1, \dots, A_m\}\sigma} \qquad \textit{provided } B_1\sigma = \cdots = B_k\sigma$$

Slide 1002

Normally combined with resolution

Prove $\forall x \, \exists y \, \neg(P(y, x) \leftrightarrow \neg P(y, y))$

The clauses are $\{\neg P(y, a), \neg P(y, y)\}$ $\{P(y, y), P(y, a)\}$

Factoring yields $\{\neg P(a, a)\}$ $\{P(a, a)\}$

Resolution yields the empty clause!

A Non-Trivial Example

$$\exists x\, [P \rightarrow Q(x)] \wedge \exists x\, [Q(x) \rightarrow P] \rightarrow \exists x\, [P \leftrightarrow Q(x)]$$

Clauses are $\{P, \neg Q(b)\}$ $\{P, Q(x)\}$ $\{\neg P, \neg Q(x)\}$ $\{\neg P, Q(a)\}$

Resolve $\{P, \underline{\neg Q(b)}\}$ with $\{P, \underline{Q(x)}\}$ getting $\{P\}$

Resolve $\{\neg P, \underline{\neg Q(x)}\}$ with $\{\neg P, \underline{Q(a)}\}$ getting $\{\neg P\}$

Resolve $\{P\}$ with $\{\neg P\}$ getting $\square$

Implicit factoring: $\{P, P\} \mapsto \{P\}$ *Many other proofs!*

Slide 1003

Prolog Clauses and Their Execution

At most one positive literal per clause!

Definite clause $\{\neg A_1, \ldots, \neg A_m, B\}$ or $B \leftarrow A_1, \ldots, A_m.$

Goal clause $\{\neg A_1, \ldots, \neg A_m\}$ or $\leftarrow A_1, \ldots, A_m.$

Linear resolution: a program clause with last goal clause

Left-to-right through program clauses

Left-to-right through goal clause's literals

Depth-first search: backtracks, but still incomplete

Unification without occurs check: fast, but unsound!

Slide 1004

Slide 1005

A (Pure) Prolog Program

```
parent(elizabeth,charles).
parent(elizabeth,andrew).

parent(charles,william).
parent(charles,henry).

parent(andrew,beatrice).
parent(andrew,eugenia).

grand(X,Z) :- parent(X,Y), parent(Y,Z).
cousin(X,Y) :- grand(Z,X), grand(Z,Y).
```

Slide 1006

Prolog Execution

```
                                    :- cousin(X,Y).
                        :- grand(Z1,X), grand(Z1,Y).
        :- parent(Z1,Y2), parent(Y2,X), grand(Z1,Y).
*          :- parent(charles,X), grand(elizabeth,Y).
X=william                    :- grand(elizabeth,Y).
              :- parent(elizabeth,Y5), parent(Y5,Y).
*                              :- parent(andrew,Y).
Y=beatrice                                    :- □.
```

* = backtracking choice point

16 solutions including `cousin(william,william)`

and `cousin(william,henry)`

The Method of Model Elimination

A Prolog-like method; complete for First-Order Logic

Contrapositives: treat clause $\{A_1, \ldots, A_m\}$ as m clauses

$$A_1 \leftarrow \neg A_2, \ldots, \neg A_m$$
$$A_2 \leftarrow \neg A_3, \ldots, \neg A_m, \neg A_1$$
$$\vdots$$

Extension rule: when proving goal P, may assume $\neg P$

A brute force method: efficient but no refinements such as subsumption

Slide 1007

A Survey of Automatic Theorem Provers

Hyper-resolution: Otter, Gandalf, SPASS, Vampire, . . .

Model Elimination: Prolog Technology Theorem Prover, SETHEO

Parallel ME: PARTHENON, PARTHEO

Higher-Order Logic: TPS, LEO

Tableau (sequent) based: LeanTAP, 3TAP, . . .

Slide 1008

Slide 1009

Approaches to Equality Reasoning

Equality is *reflexive*, *symmetric*, *transitive*

Equality is *substitutive* over functions, predicates

- ***Use specialized prover: Knuth-Bendix, . . .***

- ***Assert axioms directly***

- ***Paramodulation rule***

$$\frac{\{B[t], A_1, \ldots, A_m\} \quad \{t = u, C_1, \ldots, C_n\}}{\{B[u], A_1, \ldots, A_m, C_1, \ldots, C_n\}}$$

Modal Operators

W: set of *possible worlds* (machine states, future times, . . .)

R: *accessibility relation* between worlds

(W, R) is called a *modal frame*

$\Box A$ means A is *necessarily true* $\Big\}$ — in all **accessible** worlds
$\Diamond A$ means A is *possibly true*

$\neg\Diamond A \simeq \Box\neg A$ A *cannot be true* $\iff$ A *must be false*

Semantics of Propositional Modal Logic

For a particular frame (W, R)

An *interpretation* I maps the propositional letters to *subsets* of W

$w \Vdash A$ means A *is true in world* w

$$w \Vdash P \qquad \iff w \in I(P)$$
$$w \Vdash A \wedge B \iff w \Vdash A \text{ and } w \Vdash B$$
$$w \Vdash \Box A \quad \iff v \Vdash A \text{ for all } v \text{ such that } R(w, v)$$
$$w \Vdash \Diamond A \quad \iff v \Vdash A \text{ for some } v \text{ such that } R(w, v)$$

Truth and Validity in Modal Logic

For a particular frame (W, R), and interpretation I

$$w \Vdash A \qquad \text{means } A \text{ is true in world } w$$

$$\models_{W,R,I} A \qquad \text{means } w \Vdash A \text{ for all } w \text{ in } W$$

$$\models_{W,R} A \qquad \text{means } w \Vdash A \text{ for all } w \text{ and all } I$$

$\models A$ means $\models_{W,R} A$ for all frames; A is *universally valid*

$\ldots$ but typically we constrain R to be, say, **transitive**

All tautologies are universally valid

A Hilbert-Style Proof System for K

Extend your favourite propositional proof system with

$$\text{Dist} \quad \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$$

Inference Rule: *Necessitation*

$$\frac{A}{\Box A}$$

Treat $\Diamond$ as a *definition*

$$\Diamond A \stackrel{\text{def}}{=} \neg\Box\neg A$$

Variant Modal Logics

Start with pure modal logic, which is called K

Add *axioms* to constrain the accessibility relation:

$$\mathsf{T} \quad \Box A \to A \qquad \textit{(reflexive)} \qquad \text{logic } \mathsf{T}$$

$$\mathsf{4} \quad \Box A \to \Box\Box A \qquad \textit{(transitive)} \qquad \text{logic } \mathsf{S4}$$

$$\mathsf{B} \quad A \to \Box\Diamond A \qquad \textit{(symmetric)} \qquad \text{logic } \mathsf{S5}$$

And countless others!

We shall mainly look at $\mathsf{S4}$

Extra Sequent Calculus Rules for $\mathsf{S4}$

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta} \ (\Box l) \qquad \frac{\Gamma^* \Rightarrow \Delta^*, A}{\Gamma \Rightarrow \Delta, \Box A} \ (\Box r)$$

$$\frac{A, \Gamma^* \Rightarrow \Delta^*}{\Diamond A, \Gamma \Rightarrow \Delta} \ (\Diamond l) \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \Diamond A} \ (\Diamond r)$$

$$\Gamma^* \stackrel{\text{def}}{=} \{\Box B \mid \Box B \in \Gamma\} \qquad \text{Erase } \textit{non-}\Box \text{ assumptions}$$

$$\Delta^* \stackrel{\text{def}}{=} \{\Diamond B \mid \Diamond B \in \Delta\} \qquad \text{Erase } \textit{non-}\Diamond \text{ goals!}$$

Slide 1107

A Proof of the Distribution Axiom

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{
          \overline{A \Rightarrow B, A} \qquad \overline{B, A \Rightarrow B}
        }{A \to B, A \Rightarrow B} \; (\to l)
      }{A \to B, \Box A \Rightarrow B} \; (\Box l)
    }{\Box(A \to B), \Box A \Rightarrow B} \; (\Box l)
  }{\Box(A \to B), \Box A \Rightarrow \Box B} \; (\Box r)
}{}
$$

And thus $\Box(A \to B) \to (\Box A \to \Box B)$

Must apply $(\Box r)$ first!

Slide 1108

Part of an Operator String Equivalence

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{
          \overline{\Diamond A \Rightarrow \Diamond A}
        }{\Box \Diamond A \Rightarrow \Diamond A} \; (\Box l)
      }{\Diamond \Box \Diamond A \Rightarrow \Diamond A} \; (\Diamond l)
    }{\Box \Diamond \Box \Diamond A \Rightarrow \Diamond A} \; (\Box l)
  }{\Box \Diamond \Box \Diamond A \Rightarrow \Box \Diamond A} \; (\Box r)
}{}
$$

In fact, $\Box \Diamond \Box \Diamond A \simeq \Box \Diamond A$ also $\Box \Box A \simeq \Box A$

The $S4$ operator strings are $\Box$ $\Diamond$ $\Box \Diamond$ $\Diamond \Box$ $\Box \Diamond \Box$ $\Diamond \Box \Diamond$

Slide 1109

Two Failed Proofs

$$\frac{\dfrac{\Rightarrow A}{\Rightarrow \Diamond A}\;(\Diamond r)}{A \Rightarrow \Box\Diamond A}\;(\Box r)$$

$$\frac{\dfrac{B \Rightarrow A \wedge B}{B \Rightarrow \Diamond(A \wedge B)}\;(\Diamond r)}{\Diamond A, \Diamond B \Rightarrow \Diamond(A \wedge B)}\;(\Diamond l)$$

Can extract a countermodel from the proof attempt

Simplifying the Sequent Calculus

7 connectives *(or 9 for modal logic)*:

$$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow \quad \forall \quad \exists \quad (\square \quad \Diamond)$$

Left and right: so 14 rules *(or 18)* plus basic sequent, cut

Idea! Work in **Negation Normal Form**

Fewer connectives: $\wedge \quad \vee \quad \forall \quad \exists \quad (\square \quad \Diamond)$

Sequents need *one side only!*

Simplified Calculus: Left-Only

$$\frac{}{\neg A, A, \Gamma \Rightarrow} \text{ (basic)} \qquad \frac{\neg A, \Gamma \Rightarrow \qquad A, \Gamma \Rightarrow}{\Gamma \Rightarrow} \text{ (cut)}$$

$$\frac{A, B, \Gamma \Rightarrow}{A \wedge B, \Gamma \Rightarrow} \; (\wedge l) \qquad \frac{A, \Gamma \Rightarrow \qquad B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \; (\vee l)$$

$$\frac{A[t/x], \Gamma \Rightarrow}{\forall x\, A, \Gamma \Rightarrow} \; (\forall l) \qquad \frac{A, \Gamma \Rightarrow}{\exists x\, A, \Gamma \Rightarrow} \; (\exists l)$$

Rule $(\exists l)$ holds *provided* x is not free in the conclusion!

Slide 1203

Left-Only Sequent Rules for $S4$

$$\frac{A, \Gamma \Rightarrow}{\Box A, \Gamma \Rightarrow} \ (\Box l) \qquad \frac{A, \Gamma^* \Rightarrow}{\Diamond A, \Gamma \Rightarrow} \ (\Diamond l)$$

$$\Gamma^* \stackrel{\text{def}}{=} \{\Box B \mid \Box B \in \Gamma\} \qquad \text{Erase non-}\Box\text{ assumptions}$$

From 14 *(or 18)* rules to 4 *(or 6)*

Left-side only system uses **proof by contradiction**

Right-side only system is an exact *dual*

Slide 1204

Proving $\forall x \, (P \rightarrow Q(x)) \Rightarrow P \rightarrow \forall y \, Q(y)$

Move the right-side formula to the left and convert to NNF:

$$P \wedge \exists y \, \neg Q(y), \ \forall x \, (\neg P \vee Q(x)) \Rightarrow$$

$$\frac{\dfrac{\overline{P, \neg Q(y), \neg P \Rightarrow} \quad \overline{P, \neg Q(y), Q(y) \Rightarrow}}{\dfrac{P, \neg Q(y), \neg P \vee Q(y) \Rightarrow}{\dfrac{P, \neg Q(y), \forall x \, (\neg P \vee Q(x)) \Rightarrow}{\dfrac{P, \exists y \, \neg Q(y), \forall x \, (\neg P \vee Q(x)) \Rightarrow}{P \wedge \exists y \, \neg Q(y), \forall x \, (\neg P \vee Q(x)) \Rightarrow}}}}{}}$$

$(\vee l)$

$(\forall l)$

$(\exists l)$

$(\wedge l)$

Slide 1205

Adding Unification

Rule $(\forall l)$ now inserts a **new** free variable:

$$\frac{A[z/x], \Gamma \Rightarrow}{\forall x\, A, \Gamma \Rightarrow} \; (\forall l)$$

Let unification instantiate *any free variable*

In $\neg A, B, \Gamma \Rightarrow$ try unifying A with B to make a basic sequent

Updating a variable affects *entire proof tree*

What about rule $(\exists l)$? *Skolemize!*

Slide 1206

Skolemization from NNF

Follow tree structure; don't pull out quantifiers!

$[\forall y\, \exists z\, Q(y,z)] \wedge \exists x\, P(x) \quad$ to $\quad [\forall y\, Q(y, f(y))] \wedge P(a)$

Better to push quantifiers in (called **miniscoping**)

Proving $\exists x\, \forall y\, [P(x) \rightarrow P(y)]$

Negate; convert to NNF: $\quad \forall x\, \exists y\, [P(x) \wedge \neg P(y)]$

Push in the $\exists y$: $\quad \forall x\, [P(x) \wedge \exists y\, \neg P(y)]$

Push in the $\forall x$: $\quad \forall x\, P(x) \wedge \exists y\, \neg P(y)$

Skolemize: $\quad \forall x\, P(x) \wedge \neg P(a)$

Slide 1207

A Proof of $\exists x \, \forall y \, [P(x) \to P(y)]$

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{y \mapsto f(z)}{P(y), \neg P(f(y)), P(z), \neg P(f(z)) \Rightarrow} \text{(basic)}
      }{P(y), \neg P(f(y)), P(z) \wedge \neg P(f(z)) \Rightarrow} (\wedge l)
    }{P(y), \neg P(f(y)), \forall x \, [P(x) \wedge \neg P(f(x))] \Rightarrow} (\forall l)
  }{P(y) \wedge \neg P(f(y)), \forall x \, [P(x) \wedge \neg P(f(x))] \Rightarrow} (\wedge l)
}{\forall x \, [P(x) \wedge \neg P(f(x))] \Rightarrow} (\forall l)
$$

Unification chooses the term for $(\forall l)$

Slide 1208

A Failed Proof

Try to prove $\forall x \, [P(x) \vee Q(x)] \Rightarrow \forall x \, P(x) \vee \forall x \, Q(x)$

NNF: $\exists x \, \neg P(x) \wedge \exists x \, \neg Q(x), \; \forall x \, [P(x) \vee Q(x)] \Rightarrow$

Skolemize: $\neg P(a) \wedge \neg Q(b), \; \forall x \, [P(x) \vee Q(x)] \Rightarrow$

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{y \mapsto a}{\neg P(a), \neg Q(b), P(y) \Rightarrow} \qquad \cfrac{y \mapsto b???}{\neg P(a), \neg Q(b), Q(y) \Rightarrow}
    }{\neg P(a), \neg Q(b), P(y) \vee Q(y) \Rightarrow} (\vee l)
  }{\neg P(a), \neg Q(b), \forall x \, [P(x) \vee Q(x)] \Rightarrow} (\forall l)
}{\neg P(a) \wedge \neg Q(b), \forall x \, [P(x) \vee Q(x)] \Rightarrow} (\wedge l)
$$

The World's Smallest Theorem Prover?

```
prove((A,B),UnExp,Lits,FreeV,VarLim) :- !,
        prove(A,[B|UnExp],Lits,FreeV,VarLim).
prove((A;B),UnExp,Lits,FreeV,VarLim) :- !,
        prove(A,UnExp,Lits,FreeV,VarLim),
        prove(B,UnExp,Lits,FreeV,VarLim).
prove(all(X,Fml),UnExp,Lits,FreeV,VarLim) :- !,
        \+ length(FreeV,VarLim),
        copy_term((X,Fml,FreeV),(X1,Fml1,FreeV)),
        append(UnExp,[all(X,Fml)],UnExp1),
        prove(Fml1,UnExp1,Lits,[X1|FreeV],VarLim).
prove(Lit,_,[L|Lits],_,_) :-
        (Lit = -Neg; -Lit = Neg) ->
        (unify(Neg,L); prove(Lit,[],Lits,_,_)).
prove(Lit,[Next|UnExp],Lits,FreeV,VarLim) :-
        prove(Next,UnExp,[Lit|Lits],FreeV,VarLim).
```