Chapter Overview

Logic and Reasoning

Trees and Graphs

State and Problem Spaces

Logic and Knowledge

Propositional Logic

Predicate Logic

Inference Methods

PROLOG

Trees and Graphs

frequently used for reasoning

tree

hierarchical data structures

nodes (vertices) store information or knowledge

branches (edges, links) connect nodes

special type of graph or semantic net

graph

mathematical structure, frequently used to describe networks

nodes and edges as in trees

edges may have weights

cycles are paths through the graph ending at the initial node

a *connected graph* has links to all its nodes

a *directed acyclic graph* (DAG; also lattice) has directed links and no cycles

State and Problem Spaces

search as problem solving method

state space

graph describing *states* and *transitions* between them

a problem is solved by progressing from one state to another

transitions describe admissible actions to move to another state

problem space

closely related to state space

a solution corresponds to a valid *path* from start (problem statement) to answer (solution)

finite state machines are formal descriptions of a class of state-based systems

Formal Logic

for knowledge representation and reasoning

syntax

defines the language for statements
a *well-formed fomula* (wff) is a legitimate expression

semantics

establishes the connection between the language and the problem domain
provides an *interpretation* of a formula

axioms

represent the basic assumptions

inference rules

specify when a new formula can be derived from existing ones

calculus

set of rules for the derivation of new formulae (*theorems*)

proof of a theorem

sequence of rule applications during the derivation of a theorem

Logic Systems

and their properties

interpretation

assignment of truth values to a wff

model

interpretation in which the wff is true

satisfiability

there is an interpretation which makes the wff true

validity

the wff is true in all interpretation

correctness of a calculus

only sematically valid formulae can be deduced syntactically

completeness of a calculus

each sematically valid formula can also be deduced syntactically

Propositional Logic

logical treatment of simple statements

syntax

propositional symbols, logical connectives

semantics

a truth value is assigned to each symbol (interpretation)

evaluation

truth tables, semantic trees, etc.

decidable: there are systematic procedures to check the validity of any propositional formula

limitations

expressiveness: no quantifiers, variables, terms, functions

Predicate Logic

logical treatment of complex statements

syntax quantifiers, predicates, constants, variables, functions, terms
several notational variants (normal forms, clause form)

semantics a mapping is defined between objects in a domain and symbols (interpretation)
far more complex than for propositional logic

evaluation undecidable: there can be no systematic procedures to check the validity of an arbitrary predicate logic formula
various calculi and proof methods, especially for limited subsets (Horn clause logic, first order predicate logic)

limitations efficiency, understandability

Inference Methods

ways to come to conclusions

deduction sound

conclusions must follow trom their premises
prototype of logical reasoning

induction unsound

inference from specific cases (examples) to the general

abduction unsound

reasoning from a true conclusion to premises that may have caused the conclusion

resolution sound

find two clauses with complementary literals, and combine them

generate and test unsound

a tentative solution is generated and tested for validity
often used for efficiency (trial and error)

default reasoning unsound

general or common knowledge is assumed in the absence of specific knowledge

analogy unsound

a conclusion is drawn based on similarities to another situation

heuristics unsound

rules of thumb based on experience

intuition unsound

typically human reasoning method

nonmonotonic reasoning unsound

new evidence may invalidate previous knowledge

autoepistemic unsound

reasoning about your own knowledge

Metaknowledge

knowledge about knowledge

abstraction

similarities or patterns in the knowledge itself are found

evaluation

the computation process is observed, and knowledge about it is gathered and applied

verification

new knowledge is in the correct form

"Am I doing things right?"

validation

a chain of correct inference steps leads to the correct answer

"Am I doing the right thing?"

Important Concepts

non-monotonicity

axioms can be retracted, and new ones introduced

truth maintenance systems

maintain the integrity of the knowledge base

intermediate conclusions based on retracted facts are withdrawn

closed world assumption

if something is not explicitly stated as an axiom, it is assumed to be false

refutation "reductio ad absurdum"

a statement is proven by assuming that it is false, and showing that this leads to a contradiction

frame problem

recognition of changes over time

inspired by movies as sequences of frames

PROLOG

PROgramming in LOGic

syntax

modified Horn clauses

semantics

e.g. operational, model theoretic, fixed point

evaluation

modified resolution method, refutation

limitations

unsound ("occurs" check)

incomplete (depth-first search strategy)

negation only "as failure"

Resolution

sound inference method

resolution rule

If $(A \vee B)$ is true and $(\neg B \vee C)$ is true, then $(A \vee C)$ is true

application

find two clauses with complementary literals (e.g. $B, \neg B$), eliminate all the instances of the literals, and combine the rest of the clauses into a new one

more general than modus ponens, modus tollens

Unification

matching terms

substitutions

assignment of values (constants, variables, terms) to variables

unifier for a set of patterns

variable substitution such that all patterns are identical after its application

Prolog Notation

predicates, variables, constants

variables start with a capital letter

facts

basic axioms

```
male(nicholas).
```

rules

axioms stated as implications

```
sisters(X,Z) :-  child_of(X,Y),
                 child_of(Z,Y),
                 female(X),
                 female(Z).
```

query

statement to be proven

```
?- sisters(nicholas, marie).
```

Advantages

of logic

correctness

consistency can be checked automatically

completeness

all possible solutions are guaranteed to be found

expressiveness

in principle, all formalisms can be translated into logic

higher order logic might be required

declarative style

does not require implementation-dependent details

Limitations

of logic

efficiency

evaluation time unknown, often no intermediate results

formalization

can be tedious

uncertainty

only true and false

control

heuristics for evaluation either are extra-logical or meta-level concepts

nonmonotonicity

not for deductive approaches

Chapter Review

Logic and Reasoning

Trees and Graphs

important for problem solving strategies

State and Problem Spaces

basis for search as problem solving method

Logic and Knowledge

logic as basis for knowledge representation and reasoning

Propositional Logic

formal treatment of simple statements

Predicate Logic

formal treatment of complex propositions

Inference Methods

different ways of doing reasoning

PROLOG

a programming language based on logic