Chapter Overview

Rule-Based Reasoning

Rules

Rule-Based Inference

Forward Reasoning

Backward Reasoning

Rule-Based Architectures

Advantages

Problems

Rules

important knowledge representation paradigm

features

natural expression of knowledge

especially empirical associations

format

`if` -portion: condition, premise, antecedent

`then` -portion: action, conclusion, consequent

usage

draw conclusions based on available evidence

- rules are not the same as
  `If ... then ... else ...`
  constructs in procedural languages
- rules are not really logic systems
- rules-based systems are also called production systems

Rule-Based Inference

reasoning method

modus ponens

A is true, and

A $\rightarrow$ B is true

then B is derived to be true

searching

start from available information, and try to combine it so that the desired conclusions are supported

pattern matching

make sure that the chaining between facts and rules is correct

Differences to logic (deduction):

- non-monotonic reasoning
- uncertainty
- often forward reasoning

Forward Reasoning

also: *forward chaining, data-driven reasoning*

principle

combine available data to formulate conclusions

rule selection

premises must be satisfied

usage

few facts, many acceptable conclusions

applications

synthesis; design, configuration, planning, scheduling

close to bottom-up evaluation

Forward Reasoning Process

Rule interpretation

matching

known facts are compared with the premises of rules

if all premises of a rule are true, it is satisfied

conflict resolution

if multiple rules are satisfied, one must be selected

execution

the selected rule is applied

new facts or rules can be generated

execution of rules proceeds forward, from premises (**`if`**) to conclusions (**`then`**)

Backward Reasoning

also: *backward chaining, goal-driven reasoning*

principle

find supporting evidence for a possible conclusion

rule selection

goals that match premises

usage

few possible conclusions, many facts, but only a few are relevant

applications

diagnostic problems

close to top-down evaluation and depth-first search

Backward Reasoning Process

find supporting evidence

top-level goals

collect all goals to be tried

put them on a stack

select goal

take one goal and determine all rules capable of satisfying that goal

consequents of the rules must match the goal

check applicable rules

for each of these rules, examine the premises

a) if all premises are satisfied, a rule can be executed, and the goal is solved

b) if a premise is not satisfied, look for rules that may satisfy it, and place the premise as *sub-goal* on the stack

   continue with the next goal

c) if there is no such rule, query the user;

   if the answer is satisfactory, continue with

the next premise,
otherwise go to the next rule

unsatisfied goals

if no rule satisfies the current goal, it remains undetermined

it is removed from the stack, and the next one is tried

if the stack is empty, we're done

the known fact base initially is empty; at the end, it contains facts supporting the goal

Rule-Based Architectures

for different types of knowledge

Inference Networks

knowledge base can be visualized as a network of interconnected facts and rules

Pattern-Matching Systems

not easily visualizable

conclusions are derived from facts and rules according to the matching of (intermediate) facts and premises

Inference Networks

variations of semantic networks

representation

directed, acyclic graph with facts as nodes
and rules specifying the links
static knowledge structure: all links are
known in advance
often based on taxonomies

principle

results are propagated through the network

implementation

simple and efficient:
predetermined search space
limitations on variable bindings

usage

diagnostics or classification for reasonably well-understood problems

often with uncertainty for parameter values and rules

examples

Mycin, Prospector, GenAID, Personal Consultant

frequently based on backward reasoning

Pattern-Matching Systems

match premises of rules to facts

representation

rules and facts

dynamic knowledge structure: relationships between rules and facts are formed at run-time

principle

intermediate results are created according to matching between the premises of a rule and facts, and the actions in the consequent of a rule

implementation

complex and possibly inefficient

more expressive than inference networks

incrementally developed search space

complex pattern matching with multifield facts, multifield patterns, full variable bindings, multiple instantiations for variables

usage

problems with unbound or plentiful solutions, e.g. in design, planning, synthesis

uncertainty is more difficult to integrate

examples

XCON, OPS-5, ART, CLIPS, KEE

historically, frequently based on forward reasoning

Essential Features

of pattern-matching systems

pattern connectives

relations between the premises of a rule

usually AND (implicit)

wildcards

term that can match any atomic symbol or number within a fact

similar to variable, but no binding occurs

often used to disregard irrelevant fields

field constraints

negation or disjunction to specify (un-)acceptable values for a field

mathematical operators

for calculations of values

test feature

used to evaluate the value of a field

Advantages

of rule-based systems

modularity
: rules are distinct units of knowledge

uniformity
: all knowledge is expressed in the same format

naturalness
: rules are a natural format for expressing knowledge

Problems

of rule-based systems

infinite chaining

cycles between rules

contradictions by new knowledge

new knowledge intended to fix a problem may introduce a unwanted contradictions

modifications of existing rules

in addition to infinite chaining and contradictions, additional rules may result from modifications

inefficiency

search based on pattern matching with each rule is very inefficient

improvements: Rete algorithm, partitioning

opacity

the overall behavior of the system can be difficult to understand because a global perspective is hard to achieve

domain coverage

certain problems are too complex
e.g. too many variations of rules, unclear relationships

Chapter Review

Rule-Based Reasoning

Rules: basic unit of knowledge

Rule-Based Inference

conclusions based on rules and facts

Forward Reasoning

from available data to conclusions

Backward Reasoning

find evidence for a possible conclusion

Rule-Based Architectures

inference networks, pattern-matching systems

Advantages

modularity, uniformity, naturalness

Problems

infinite chaining, contradictions by new knowledge, modifications of existing knowledge; inefficiency, opacity, domain coverage