Design of Logic-based Intelligent Systems

Lecture 1: Introduction – Part 1

K. Truemper
University of Texas at Dallas

Acknowledgements
(Names in Alphabetical Order)

Organizations:

Alexander von Humboldt-Stiftung
Deutsche Forschungsgemeinschaft
IASI-CNR, Rome
Office of Naval Research
University of Augsburg
University of Texas at Dallas
Zuse Institut, Berlin

People:

G. Felici
M. Groetschel
M. Juenger
R. Karpelowitz
G. Rinaldi
M. Stoer
D. K. Wagner

PhD Students:

H. Almubaid
A. Remshagen
J. Straach
F.-S. Sun
Y. Zhao

Reviewers ("Design" book):

T. Achterberg
C. Buchheim
S. Hachul
F. Liers
J. Mugan
M. Percan
B. Randerath

. . . . and many, many more (colleagues, students, friends, family)

Intelligent Systems

Definition

A system is *intelligent* if it accomplishes feats that, when carried out by humans, require a substantial amount of intelligence.

Example Tasks

Medical diagnosis, processing of natural language, supervision of complex processes.

Definition

An *expert system* is an intelligent system which in an interactive setting asks a person for information and, based upon the response, draws conclusions or gives advice.

Definition

An *intelligent agent* is an intelligent system which perceives its environment by sensors and which uses that information to act upon the environment.

Observation

Expert systems and intelligent agents are special cases of intelligent systems.

An example of an intelligent system that is not an intelligent agent: an intelligent system that constructs expert systems from data. That intelligent system is not an intelligent agent, unless one accepts the odd notion that creating an expert system constitutes acting upon an environment.

Terminology

"Model": In propositional logic, a satisfying solution. Here, a mathematical formulation of a real-world situation.

"Valid": In propositional logic, a formula that always evaluates to *True*. Here, a formulation that correctly represents a part of the real world of interest.

Levels of Thinking, Informal Definition

First Level (= thinking about problems)

Direct reasoning about problems.

Typical questions:
"Do symptoms s and t imply that disease d or e is present?"
"Is this log-in behavior typical for a hacker ?"

Second Level (= thinking about thinking)

We think about which logic module of the first level is to be used.

Third Level (= thinking about thinking about thinking)

We think about which selection process of the second level is to be used.

Reduction to thinking at lower level generally cannot be achieved by a polynomial algorithm. Note: “cannot” is meant pragmatically in the sense that presently nobody knows of a polynomial reduction.

Link to polynomial hierarchy of theory of computation: kth level of thinking corresponds to kth-level of hierarchy.

Quantified Boolean Formula (QBF)

a, b, c, $d, \ldots$ = vectors of Boolean variables
$\forall a \ \exists b \ \forall c \ \exists d \ldots \ \ D(a, b, c, d, \ldots)$

Each quantifier introduces another level.

Logic Tools

Production Rules

If-then relationships with single conclusion.

Example: "If symptoms s and t are present, then disease d is present."

Prolog

Elaboration of production rule concept where *universal quantification* is allowed.

Example: "For all x, y, and z: If x is an ancestor of y, and if y is a parent of z, then x is an ancestor of z."

Propositional Logic

True, *False*, variables, $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\rightarrow$ (only if), $\leftrightarrow$ (if and only if), formulas.

CNF (conjunctive normal form): conjunction of disjunctions.
Example: $(x \vee \neg y) \wedge (x \vee z)$.

DNF (disjunctive normal form): disjunction of conjunctions.
Example: $(x \wedge \neg y) \vee (x \wedge z)$.

Negation converts CNF to DNF, and conversely.

First-order Logic

Extension of propositional logic by predicates, also called truth-functions, and quantifications $\exists$ (there exists) and $\forall$ (for all).

Example: $\forall x, y\ [f(x, y) \rightarrow \exists z\ g(x, z)]$.

Fuzzy Logic

Production rules plus uncertainty of facts.

Example: “If a person uses an umbrella, then it likely is raining.”

Bayesian Networks

Directed acyclic graph plus probability tables at nodes represent probabilistic rules.

An Extension of Propositional Logic

- Cost of *True* or *False* for a variable; call this Truecost and Falsecost.
- Representation of unknown values by two values: *Absent* and *Unavailable*.
- Likelihood of clauses being applicable.
- Predicates with finite quantifications $\exists$ (there exists) and $\forall$ (for all).
- Quantification of propositional variables. For example, for all *True/False* values of a subset of the variables, there exist *True/False* values for the remaining variables so that the formula has a specified value. Can be viewed as special case of predicate with finite quantification.

Examples

For variable x: Truecost $= 10$, Falsecost $= -5$

U, V finite sets

$\forall u \in U \ \forall v \in V \ [\neg g(u,v) \vee h(u,v)]$

Equivalent:

$\bigwedge_{\substack{u \in U \\ v \in V}} [\neg g(u,v) \vee h(u,v)]$

$x = \mathit{Absent}$: value of x unknown, but could be obtained

$x = \mathit{Unavailable}$: value of x cannot be obtained

Quantification of propositional variables

CNF formula R with vectors q and x of variables.

CNF formula S with vectors q and y of variables.

For all *True/False* values of the q_i of vector q:
if R is satisfiable, then S is satisfiable as well.

Write as: $\forall q \; (\exists x \; R \rightarrow \exists y \; S)$

Problems SAT and MINSAT

We often use CNF *system* instead of CNF *formula*.

CNF system: list of variables, list of CNF clauses, each of which uses a subset of the variables.

Special Cases

trivial system: variables given, but no clauses; or clauses given, but no variables.

empty system: no variables and no clauses.

Problem SAT

Instance: CNF system S.

Solution: Either: A satisfying solution of S. Or: The conclusion that S is not satisfiable.

Problem MINSAT

Instance: CNF system S. For each variable of S, two rational cost values associated with the values *True* and *False* for that variable.

Solution: Either: A satisfying solution of S for which the total cost is minimum. Or: The conclusion that S is unsatisfiable.

Theorem Proving

Systems S and T; all variables of T occur in S

Equivalent Questions

Is every satisfying solution of S also a satisfying solution of T?

Does $S \rightarrow T$ always evaluate to *True*?

Is $S \rightarrow T$ a tautology?

Is $\neg S \vee T$ a tautology?

Is $\neg(\neg S \vee T)$ a contradiction?

Is $S \wedge \neg T$ unsatisfiable?

In case of “yes,” T is a *theorem* of S.

When S models a real-world problem

T is a theorem of S if and only if, for every possible case of the real-world setting, T evaluates to $True$.

Counterexample

If T is not a theorem of S, then $S \wedge \neg T$ has a satisfying solution, which demonstrates that T is not a theorem. That solution is a *counterexample* to the claim "T is a theorem."

Typical Case

$S \wedge \neg T$ is a CNF system. Indeed, S usually is a CNF system, and T is a CNF clause or is trivially equivalent to such a clause.

Example

S: $(\neg p \vee q) \wedge (\neg \vee r)$
T: $\neg p \vee r$

$\neg T = \neg(\neg p \vee r) = p \wedge \neg r$

$S \wedge \neg T = (\neg p \vee q) \wedge (\neg q \vee r) \wedge p \wedge \neg r$

Can enforce $p \wedge \neg r$ by $p = \mathit{True}$ and $r = \mathit{False}$. These values reduce S to $S' = q \wedge \neg q$, which is unsatisfiable. Hence, T is a theorem.

Observation

If S is a CNF system and T is CNF clause, then $S \wedge \neg T$ is equivalent to S with variables fixed to the *True/False* values implied by $\neg T$.

Other Reasoning

Abduction

Inverse use of an implication.

Example:

Implication: $infection \rightarrow elevated\_temperature$

Abduction: $elevated\_temperature \rightarrow infection\ frequently$

We view abduction as deduction with likelihoods.

Induction

One claims general result from some instances.

Nonmonotonic Reasoning

Conclusion may become invalid when additional information is obtained.

Default Reasoning

Reasoning in the absence of specific information.

Metareasoning

Decide which reasoning to employ.

Note: Default reasoning and metareasoning are examples of thinking at the second level.