Knowledge Representation and Reasoning

# 0. Opening

| Logic | A language without the nuances (sfumature). It is an artificial language. There are multiple logics, we use it to express and manipulate knowledge |
| --- | --- |

# 1. What is Knowledge Representation

| Sound | True premises derive true conclusions. If two premises are true the conclusion is always true. Don't create an error |
| --- | --- |
| Complete | Whatever follows can be derived. It is partially true |
| Syllogism | Handle Knowledge about classes. Very limited (only two classes at a time, no disjunctions, no complex properties, no relationship between objects) |
| Propositional logic | Simple logic, only speaks about prepositions (properties). It is too inexpressive |
| Predicate logic | It is too complex, no sound and complete derivation system |
| Machine-Tailored KR formalisms | Has no formal semantics (can be ambiguous) |
| Frames | Part of Machine-Tailored KR Formalisms. Are blocks with open "windows" to fill in properties. No formal semantics |
| Logic based KR | Language with clear syntax and formal semantics |
| DLs | Description logics are a family of KR language. There are many DLs |

# 2. Boolean Algebra and Propositional Logic

| Boolean Algebra | Formalism for manipulating truth values. True (1) False (0) |
| --- | --- |
| And | Symbol ^. A conjunction of two truth values is true if and only if both values are true |
| Or | Symbol v. A disjunction of truth values is true iff at least one value is true |
| Not | Symbol ¬. The negation of a truth value is true iff the value is false |
| Propositional Logic | Formalism for combining propositions (sentences, statements) to make conclusion about their truth value |
| Atomic Propositions | State one specific fact or property |
| Formulas | Combine propositions to make complex statements. It can be true or false depending on the subformulas. Two formulas are equivalent iff they have the same truth table |
| Propositional variables | Same as atomic propositions |
| Operator precedence | ¬ (not) ^(and) v(or) |
| Subformulas | Components of a formulas |
| Valuation | An assignment of a truth value to each variable |
| Tautologies | Are always evaluated to 1 regardless of the valuation |
| Contradictions | Are always evaluated to 0 regardless of the valuation |
| Non-tautological satisfiables | Formulas that are not tautologies or contradictions |
| Function | A formula with n variables |
| De Morgan 1 | ¬ (X ^ Y) = ¬ X v ¬ Y |
| De Morgan 2 | ¬ (X v Y) = ¬ X ^ ¬ Y |
| Distribuitivity1 | X ^ ( Y v Z) = (X ^ Y) v (X ^ Y) |
| Distributiviy2 | X v (Y ^ Z) = (X v Y) ^ (X v Y) |
| Involution | ¬ ¬ X = X |

# 3. Propositional Rules

| Clause | Disjunction of literals (variables or negated variables) |
| --- | --- |
| Horn clauses | Clause with exactly one positive literal |
| Knowledge base | Finite set of rules |
| Entails | if K entails B; B is a consequence of K |
| Redux | The KB without facts in the body |

# 4. Predicate Logic and Rules

| Propositional Logic | Deal with one object |
| --- | --- |
| Predicate | P(t) or Q(t,s), careful with the arity of predicate symbols |
| Literal | A predicate or its negation |
| Predicate Horn clause | Disjunction of literals with exactly one positive |
| Interpretation | An interpretation has two components I = (delta^i , .^i) a domain an interpretation function |
| Domain | Says which world we are talking about DELTA^i |
| Interpretation function .^i | Tells how to read the symbols in this domain we are talking about |
| Graph | Collection of points (nodes) and arrows (edges) connecting them |
| Predicate rule | Is a predicate clause P(t) <- Q1(t1),... Qn(Tn) such that all variables P(t) appear in the body |
| Canonical model | Propagate the knowledge building a minimal representation of all possible models. Minimal in terms of constraints |
| Complete | Every consequence is found |
| Sound | Every fact follows from all models |
| Homomorphism |  |
| Constraint | All variables in the head of a rule appear also in the body |
| Grounding | The grounding of a rule is the set of all its instantiations given a fixed set of constants |

# 5. Extensions of Rules and Introduction to DLs

| unidirectional | H <- B. Propagates knowledge from B to H |
| --- | --- |
| Constraints | May lead to contradictory knowledge |
| DLs | Description logic is a family of KR language. Characterised by clear syntax and formal unambiguous semantics. The scope is to provide reasoning |
| AL | First DL were called Attribute Language |
| ALC | Attribute Language extended to include components |

# 6. A light-Weight Description Logic (EL)