Venkata Krishna Mohan Sunkara
65528957

HOME WORK-7

2.

1. TT-Entails:

It is an algorithm for proving entailment. It is a representation of the truth table enumeration. It consists of a function which takes input as the knowledge base which is a collection of sentences in propositional logic and a query α which needs to be proven whether it is entailed or not. The algorithm consists of a list of symbols which consist of all the propositional symbols in KB and α.

It then recursively enumerates each symbol to T and F, and continues until all symbols in the list are enumerated. Now it checks the models in which the KB is true and returns the truth value of α in those models where the KB is true. If the KB is false in a model then it returns true as it doesn't matter whether α is true or false in that model. By this we can get the entailment of a query α.

The algorithm stops when all the symbols have been assigned a truth value, that is the symbols list is empty. The TT-entails algorithm is a representation of truth table enumeration theorem.

2. PL-Resolution:

It is an algorithm which takes input as the Knowledge base and a query α. All the sentences in KB are represented as a sentence of propositional logic. Now the sentences in KB and α are represented in CNF clauses. These clauses are stored in a set called clauses.

Now the algorithm compares each pair of clauses in the clauses set and apply the resolution rule if possible and then adds the new resolvents to the new set. If the clauses can't be resolved then return true. Now if the newly added resolvents are already present in the clauses set then return false that is don't add them to the clause set. If not, then add them to the clauses set.

The algorithm stops when a resolution closure is obtained that is even on applying resolution rules no new resolvents are obtained. The Pl-resolution algorithm is an implementation of resolution theorem.

3.PL-FC entails:

It is an algorithm which takes input as the knowledge base and a query q, all of which are represented as propositional statements. It also consists of some lists where count list is used for storing the number of symbols in a statement c's premise, inferred list consists of false for all symbols initially, an agenda list which consists of the symbols which are known to be true initially in the KB.

The algorithm iterates until the agenda list is empty. It selects a symbol from the agenda list and checks if it is equals to the query q, if they are equal then it returns true.

Otherwise, it checks whether the symbol is inferred or not using the inferred list. If the symbol is not inferred then it is made true and it checks every clause in the Kb where the symbol is in premise of the clause. If it is, then the count for that symbol is decremented and if the count is 0, then the conclusion part of the clause is added to the agenda list.

The algorithm terminates after a fixed point is reached that is when no more new inferences are possible and the PL-FC entails algorithm is a representation of forward chaining theorem.

4. DPLL-satisfiable:

It is an algorithm which is used for solving the satisfiability problem. It takes the entire sentence as input. The sentence is represented as a conjunction of clauses. The clauses in a sentence are stored in the clauses list and the propositional symbols are stored in the symbols list. The DPLL algorithm finds the pure symbols which are the symbols that appear in the same sign in all clauses and the unit clauses which are the clauses with just one literal remaining to be assigned a value.

If every clause in the sentence is true then the algorithm returns true and if at least one clause is false then the algorithm returns false. Otherwise, it finds the pure symbols and unit clauses from the set of clauses and add it to the model along with its value. Now it obtains the first symbol in the symbols list and calls the DPLL algorithm in a recursive fashion with the symbol's value assigned to either true or false.

The Unit clauses and the pure symbols are obtained due to unit propagation which occurs due to cascading of forced assignments. The algorithm terminates if either all the symbols are assigned truth assignments or if any one clause is false. It is representation of TT-entails algorithm but an improvement so that the algorithm terminates early.

5. Walk-Sat Algorithm:

It is an algorithm which is used for solving the satisfiability problem using local search technique. It uses the concepts of min-conflict and random-walk. It takes the set of clauses, the probability of choosing a random walk and the number of max flips allowed. It uses the min-conflict heuristic to ensure that there are minimum number of unsatisfied clauses at a given state.

Now the algorithm selects a clause which is false from the set of clauses randomly and with the specified probability it flips the truth value of a randomly selected symbol from that clause. It flips the truth value of a symbol in a clause which maximizes the satisfiability of more number of clauses. It continues until the number of flips is equal to max_flips. If the model is satisfiable then it returns true.

The algorithm is an implementation of a SAT solver and it terminates if the model is satisfiable in the specified number of max flips. If the model is not satisfiable we do not know as it doesn't terminate if the max flips is set to infinite. The algorithm is not sound as it doesn't determine whether the model is satisfiable or not in a limited number of max flips.

Hence, we can say that the Walk-Sat algorithm is not sound and doesn't terminate everytime.

1.

Description Logic:

It is a formal language which is used for the representation of Knowledge. It is more expressive than the propositional logic but less expressive than that of the First order logic. The fundamental modeling concept of a description logic is the axiom. An axiom is defined as a logical statement relating roles. A concept in Description logic can be related to a unary predicate in first order logic and a role in description logic can be related to a binary predicate of first order logic. Concepts in Description logic can be related to classes and the roles can be related to properties or relationships.

Description logic was first introduced to overcome the lack of formal semantics. The first DL- based knowledge representation system was KL-ONE. Later several DL-based systems were developed like KRYPTON, LOOM, BACK and CLASSIC. The use of structural subsumption algorithms provided the description logic with relatively efficient reasoning.

The modern Description logics has been the basis of widely used ontology languages. These add meaning to the web content by annotating it with terms defined in ontologies. The description logic consists of various operators which are similar to those in the First order logic. ⊓,⊔ are similar to conjunction and disjunction operators in the first order logic. There are also two special concepts $\bot$ and T which represent the inconsistent concept and most general concept respectively. The relationships in first order logic are represented as roles in description logic. The implications in first order logic are represented as concept inclusions using the ⊑ operator. The data axioms in description logic is equivalent to ground facts of first order logic.

A set of axioms is called a Tbox and a set of facts is called an Abox in description logic. A knowledge base is just a union of a Tbox and Abox. There are various types of description logic and each of them is defined by the concept and role axioms, operators. EL, ALC are some of the types of description logics.
– H for role hierarchy (e.g., hasDaughter ⊑ hasChild)
– R for role box (e.g., hasParent ∘ hasBrother ⊑ hasUncle)
– O for nominals/singleton classes (e.g., {Italy})
– I for inverse roles (e.g., isChildOf ≡ hasChild–)
– N for number restrictions (e.g., >2hasChild, < hasChild)
– Q for qualified number restrictions (e.g., >2hasChild.Doctor)
– F for functional number restrictions (e.g., < hasMothe

The semantics of description logics are defined by interpreting concepts as a set of individuals and roles as set of ordered pairs of individuals. The semantics of a description logic can be directly related to that of first order logic. A knowledge base k is said to be satisfied iff there exists a model which entails the knowledge base k. A KB entails an axiom ax iff for every model M of K, M entails ax.

Many description logics are decidable fragments of first-order logic and some DL's have features that are not present in FOL such as concrete domains or an operator on roles for the transitive closures. Fuzzy description logics combines fuzzy logic with DLs. Fuzzy logic is needed to deal with notions of vagueness and imprecision, so this gives the DL a motivation towards dealing with imprecise and vague concepts.

There are reasoning algorithms for Description logic. Most of these algorithms are based on the tableau proof techniques. The tableau algorithm for concept satisfiability tries to construct a finite interpretation I that satisfies C. We can also use the ABox assertion for theorem proving in description logic. We add formulas to the tableau by applying the previous rules. We apply the rules until either a contradiction is generated in every branch, or there is a branch where no contradiction appears and no rule is applicable. We can also check the satisfiability of ABoxes using tableau techniques. This method is sound and always preserve the satisfiability.

In the early stages, the Description logic's inference has been undecidable. Kl-One and Loom are the undecidable. The fact, DLP and Race types of description logic are Exponential in time. The Crack and Kris types of description logic are in polynomial space in time. The classic form of description logic from AT&T has been in polynomial time complexity.

The description logic has many applications in semantic web and uses ontologies to provide common terms of reference with clear semantics. A web based ontology language has well defined semantics and is built on existing web standards and the resulting language is based on a description language. DLR, a description language can capture semantics of many conceptual modeling methodologies in a database schema. DL Abox can also capture semantics of conjunctive queries.

Web ontology language is motivated by semantic web activity. It adds meaning to web content by annotating it with terms defined in ontologies. OWL is based on SROIQ which is an ALC extended with transitive roles. An ontology is usually considered to be a Tbox but OWL ontology is a mixed set of TBox and ABox axioms. It has well defined semantics.

REFERENCES:

https://en.wikipedia.org/wiki/Description_logic

https://www.cs.ox.ac.uk/ian.horrocks/Seminars/download/Horrocks_Ian_pt1.pdf

http://www.cse.iitd.ernet.in/~kkb/DL-1.pdf

http://www.inf.ed.ac.uk/teaching/courses/kmm/PDF/L3-L4-DL.pdf