COS402- Artificial Intelligence
Fall 2015

Lecture 6: Theorem Proving & Resolution Algorithm

Outline

- **Logical equivalence, validity and satisfiability**
- **Inference rules**
- **Resolution rule**
- **Conjunction Normal Form(CNF)**
- **Resolution algorithm**
- **Construct a model for satisfiable clauses (all clauses in KB when resolution algorithm stops.)**

Some points

- **A clause is a disjunction of literals.**
- **A CNF is a conjunction of clauses.**
- **Resolution algorithm is both complete and sound.**
- **Theorem proving does not need to consult models.**
- **Every sentence can be written in CNF.**
- **KB |= α if and only if (KB => α) is valid. (Deduction Theorem)**
- **KB |= α if and only if KB ∧ ¬α is unsatisfiable.**

Inference rules

- **Modus Ponens**
- **And elimination**
- **Reverse And elimination**
- **All logical equivalence rules**

Construct a model

- **Consider the set of all clauses (KB, ~α and all derived clauses)**
- **For k=1,2,…,n**
  - **If P_k is “forced” to be true or false, set P_k accordingly.**
  - **Else set P_k arbitrarily**

Construct a model(example)

- **KB: $(P_1 \vee P_2) \wedge (\sim P_1 \vee \sim P_3) \wedge (\sim P_3 \vee P_4)$**
- **α: P_1**
- **The set of all sentences: (KB, ~α, derived clauses (5) and (6))**
  - **(1) $(P_1 \vee P_2)$**
  - **(2) $(\sim P_1 \vee \sim P_3)$**
  - **(3) $(\sim P_3 \vee P_4)$**
  - **Add ~α, (4) P_1**
  - **Resolve (1) and (4) to derive (5) P_2**
  - **Resolve (1) and (2) to derive (6) $(P_2 \vee \sim P_3)$**
- **Follow the algorithm on previous slide, can find a model (P_1=F, P_2=T, P_3=T, P_4=F). Note: P_3 is set to T arbitrarily, all others are "forced".**

Proof for constructing a model(Will not get stuck when setting $P_1, P_2, \ldots, P_n$)

- **Proof idea: (by contradiction)**
  - **Assume first get stuck when setting P_k**
  - **Then will have P_k and $\sim P_k$ after setting $P_1, P_2, \ldots, P_{k-1}$**
  - **Must come from original form α v P_k and β v $\sim P_k$, respectively. α and β are clauses over $P_1, P_2, \ldots, P_{k-1}$, α=F and β=F after setting $P_1, P_2, \ldots, P_{k-1}$. So α v β is False.**
  - **However, α v β is in the set of all clauses. (we can derive α v β by resolving (α v $P_{k)}$ and (β v $\sim P_k$), so α v β should be True after setting $P_1, P_2, \ldots, P_{k-1}$.**
  - **Contradiction.**

Review questions: true or false

1. **Theorem proving is a technique which applies inference rules on known facts in order to derive new facts.**
2. **Modus Ponens is one of the inference rules that are used in resolution algorithm.**
3. **Resolution algorithm can be used to determine whether a sentence is satisfiable.**
4. **Resolution algorithm is used to determine whether KB $\models$ α.**

Review questions: true or false(con'd)

5. **Finding proofs can be converted into a search problem.**
6. **By using resolution rule on (¬A ∨ B) and (A ∨ ¬B), an empty clause is derived.**
7. **All sentences can be written in CNF.**
8. **The first step of resolution algorithm is to convert KB ∧ α into CNF.**

Announcement & Reminder

- **P1 (first programming assignment) has already been released. It is due on Tuesday Oct. 13th.**

  --- due by midnight, upload your files to CS dropbox

- **W2 is released today and is due on Tuesday Oct. 20th**

  --- Due in class, hard copies.