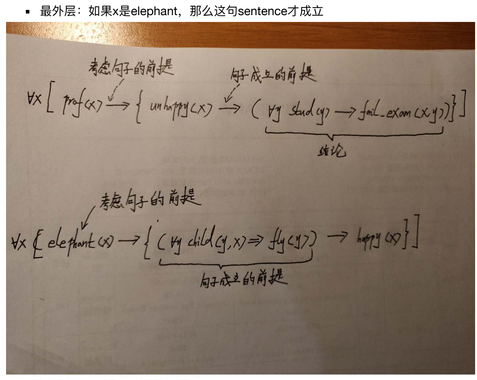
Index	Notes

translate

1. 分清楚逻辑主次
   - 场景一：Let Gov , Univ, Lca, Bill be constant symbols, and let prof(x),stud(x),unhappy(x),cut fund(x,y),fail exam(x,y) be unary and binary predicate symbols.
     - 1. If professors are unhappy all students fail their exams
       - 最外层：如果x是一个教授，那么这句话才是有意义的
   - 场景二：An elephant is happy if all its children can fly
     - 最外层：如果x是elephant，那么这句sentence才成立
   - 

2. 定语从句（不管是修饰主语成分还是宾语成分）
   - 定语从句中也包含implication
     - Everyone who **loves all animals** is loved by someone
       - $\forall x\ [\forall y\ \ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\ \ Loves(y,x)]$.
   - 定语从句只是属性
     - 直接用and就好了
       - If professors are unhappy all students fail their exams
         - $\forall x\ \forall y((prof(x) \wedge unhappy(x) \wedge stud(y)) \Rightarrow fail\_exam(x,y))$
3. Only
   - Mary studies (all and only) easy books

   $$\forall x\ study(Mary,x) \Leftrightarrow easy(x)$$
4. except
   - No animals, except giraffes, are 15 feet or higher.
   - $$\text{(a) } \forall x\ (G(x) \vee \neg F(x))$$

3. 练习

Exercise

Tell which one among the following formulas is a good representation of the sentence.

An elephant is happy if all its children can fly

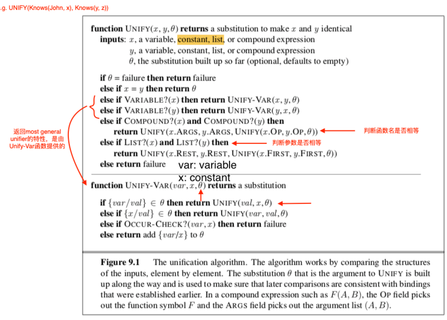
1 $\forall x\ (elephant(x) \Rightarrow ((\forall y\ child(y,x) \Rightarrow fly(y)) \Rightarrow happy(x)))$

2 $\forall x\ ((elephant(x) \wedge happy(x)) \Rightarrow \forall y\ (child(y,x) \Rightarrow fly(y)))$

3 $\forall x\ (elephant(x) \Rightarrow ((\neg\exists y(child(y,x) \wedge \neg fly(y))) \Rightarrow happy(x)))$

4 $\forall x\ (elephant(x) \Rightarrow \forall y\ ((child(y,x) \wedge fly(y)) \Rightarrow happy(x)))$

Important algorithm

- Skolemization
  -
- Unification
  - e.g. UNIFY(Knows(John, x), Knows(y, z))
  - 

  - **Figure 9.1** The unification algorithm. The algorithm works by comparing the structures of the inputs, element by element. The substitution θ that is the argument to UNIFY is built up along the way and is used to make sure that later comparisons are consistent with bindings that were established earlier. In a compound expression such as $F(A, B)$, the OP field picks out the function symbol F and the ARGS field picks out the argument list (A, B).
  -
- Evaluation
  - Backward

    //note: variable bindings are global to the procedure "evaluate"

    bool evaluate(Query Q)

    if Q is empty

    SUCCEED: print bindings of all variables in original query

    if user wants more solutions (i.e. enters ';') return FALSE

    else return TRUE

    else

    remove first predicate from Q, let q1 be this predicate

    for each rule R = h :- r1, r2, ... rj in the program in

    the order they appear in the file

    if(h unifies with q1 (i.e., the head of R unifies with q1))

    put each ri into the query Q so that if the Q was originally (q1, q2, ... qk) it now becomes (r1, r2, ... rj, q2, ... qk)

    NOTE: rule's body is put in front of previous query predicates.

    NOTE: also some of the variables in the ri's and q2...qk might now be bound because of unifying h with q1

    if (evaluate(Q)) //recursive call on updated Q

    return. //succeeded and printed bindings in recusive call.

    else

    UNDO all changes to the variable bindings that arose from unifying h and q1

    end for

    //NOTE. If R's head fails to unify with q1 we move on to try the next rule in the program. If R's head did unify but unable to solve the new query recursively, we also move on to try the next rule after first undoing the variable bindings.
  - Resolution

|

Inference

- Model checking
- Reduction
  - Modus Ponens
  - And-Elimination
  - Resolution
  - All of the logical equivalences
- Sound and complete for Inference
  - Resolution，Model checking——sound and complete
  - Modus Ponens——sound but complete only when the sentences can be represented as Horn Clause

Summary