Knowledge Representation & Reasoning

Shyamanta M Hazarika
CSE, SoE
Tezpur University

Goal

Deductive reasoning in language as close as possible to full FOL

$\neg,\ \wedge,\ \vee,\ \exists,\ \forall$

Knowledge Level:

given KB, α, determine if KB |= α.

or given an open $\alpha[x_1,x_2,...x_n]$, find $t_1,t_2,...t_n$ such that KB |= $\alpha[t_1,t_2,...t_n]$

When KB is finite $\{\alpha_1, \alpha_2, ..., \alpha_k\}$

KB |= α

iff |= $[(\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_k) \supset \alpha]$

iff KB $\cup \{\neg\alpha\}$ is unsatisfiable

iff KB $\cup \{\neg\alpha\}$ |= FALSE

where FALSE is something like $\exists x.(x \neq x)$

So want a procedure to test for validity, or satisfiability, or for entailing FALSE.

Will now consider such a procedure (first without quantifiers)

Clausal representation

Formula = set of clauses

Clause = set of literals

Literal = atomic sentence or its negation

positive literal and negative literal

Notation:

If ρ is a literal, then $\overline{\rho}$ is its complement

$\overline{p} \Rightarrow \neg p \qquad \overline{\neg p} \Rightarrow p$

To distinguish clauses from formulas:

[and] for clauses: $[p, \overline{r}, s]$ { and } for formulas: $\{ [p, \overline{r}, s], [p, r, s], [\overline{p}\ ] \}$

[] is the empty clause {} is the empty formula

So {} is different from {[]}!

Interpretation:

Formula understood as conjunction of clauses

Clause understood as disjunction of literals

Literals understood normally

$\{[p,\neg q], [r], [s]\}$ represents $((p \vee \neg q) \wedge r \wedge s)$

[ ] represents FALSE

CNF and DNF

Every propositional wff α can be converted into a formula α' in Conjunctive Normal Form (CNF) in such a way that $\models \alpha \equiv \alpha'$.

1. eliminate $\supset$ and $\equiv$ using $(\alpha \supset \beta) \rightarrow (\neg\alpha \vee \beta)$ etc.
2. push $\neg$ inward using $\neg(\alpha \wedge \beta) \rightarrow (\neg\alpha \vee \neg\beta)$ etc.
3. distribute $\vee$ over $\wedge$ using $((\alpha \wedge \beta) \vee \gamma) \rightarrow ((\alpha \vee \gamma) \wedge (\beta \vee \gamma))$
4. collect terms using $(\alpha \vee \alpha) \rightarrow \alpha$ etc.

Result is a conjunction of disjunction of literals.

an analogous procedure produces DNF,
a disjunction of conjunction of literals

We can identify CNF wffs with clausal formulas

$$(p \vee \neg q \vee r) \wedge (s \vee \neg r) \rightarrow \{ [p, \neg q, r], [s, \neg r] \}$$

So: given a finite KB, to find out if KB $\models \alpha$, it will be sufficient to

1. put $(KB \wedge \neg\alpha)$ into CNF, as above
2. determine the satisfiability of the clauses

Resolution rule of inference

Given two clauses, infer a new clause:

From clause $\{ p \} \cup C_1$,

and $\{\neg p \} \cup C_2$,

infer clause $C_1 \cup C_2$.

$C_1 \cup C_2$ is called a resolvent of input clauses with respect to p.

Example:

clauses $[w, r, q]$ and $[w, s, \neg r]$ have $[w, q, s]$ as resolvent wrt r.

Special Case:

$[p]$ and $[\neg p]$ resolve to [] (the C_1 and C_2 are empty)

A derivation of a clause c from a set S of clauses is a sequence $c_1, c_2, ..., c_n$ of clauses, where $c_n = c$, and for each c_i, either

1. $c_i \in S$, or
2. c_i is a resolvent of two earlier clauses in the derivation

Write: $S \rightarrow c$ if there is a derivation

Rationale

Resolution is a symbol-level rule of inference, but has a connection to knowledge-level logical interpretations

Claim: Resolvent is entailed by input clauses.

Suppose $\mathfrak{I} \models (p \vee \alpha)$ and $\mathfrak{I} \models (\neg p \vee \beta)$

Case 1: $\mathfrak{I} \models p$

then $\mathfrak{I} \models \beta$, so $\mathfrak{I} \models (\alpha \vee \beta)$.

Case 2: $\mathfrak{I} \not\models p$

then $\mathfrak{I} \models \alpha$, so $\mathfrak{I} \models (\alpha \vee \beta)$.

Either way, $\mathfrak{I} \models (\alpha \vee \beta)$.

So: $\{(p \vee \alpha), (\neg p \vee \beta)\} \models (\alpha \vee \beta)$.

Special case:

$[p]$ and $[\neg p]$ resolve to $[\ ]$,

so $\{[p], [\neg p]\} \models$ FALSE

that is: $\{[p], [\neg p]\}$ is unsatisfiable

Derivations and entailment

Can extend the previous argument to derivations:

If $S \rightarrow c$ then $S \models c$

Proof: by induction on the length of the derivation.
Show (by looking at the two cases) that $S \models c_i$.

But the converse does not hold in general

Can have $S \models c$ without having $S \rightarrow c$.

Example: $\{[\neg p]\} \models [\neg p, \neg q]$ i.e. $\neg p \models (\neg p \vee \neg q)$
but no derivation

However.... Resolution *is* refutation complete!

Theorem: $S \rightarrow []$ iff $S \models []$

sound and complete when restricted to [ ]

Result will carry over to quantified clauses (later)

So for any set S of clauses: S is unsatisfiable iff $S \rightarrow []$.

Provides method for determining satisfiability: search all derivations for [].

So provides a method for determining all entailments

A procedure for entailment

To determine if KB |= α,

- put KB, $\neg\alpha$ into CNF to get *S*, as before
- check if *S* $\rightarrow$ [].

If KB = {}, then we are testing the validity of α

Non-deterministic procedure

1. Check if [] is in *S*.

   If yes, then return **UNSATISFIABLE**

2. Check if there are two clauses in *S* such that they resolve to produce a clause that is not already in *S*.

   If no, then return **SATISFIABLE**

3. Add the new clause to *S* and go to 1.

Note: need only convert KB to CNF once

- can handle multiple queries with same KB
- after addition of new fact α, can simply add new clauses α' to KB

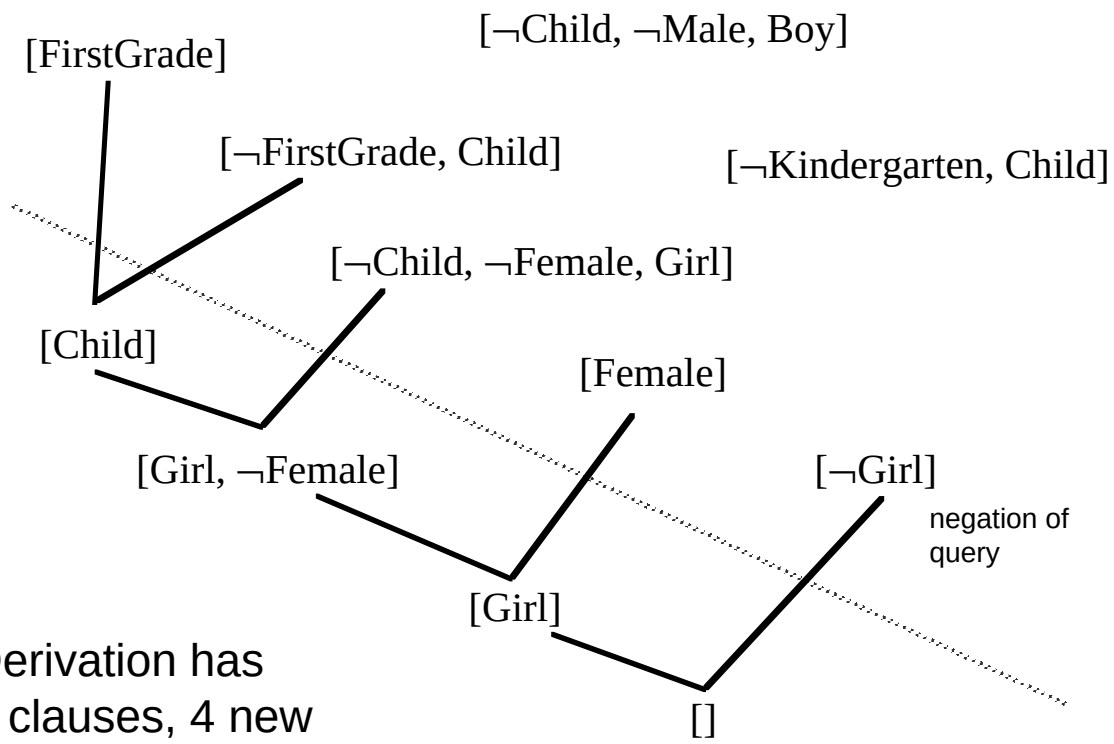
So: good idea to keep KB in CNF

Example 1

KB

FirstGrade
FirstGrade $\supset$ Child
Child $\wedge$ Male $\supset$ Boy
Kindergarten $\supset$ Child
Child $\wedge$ Female $\supset$ Girl
Female

Show that KB |= Girl



Derivation has
9 clauses, 4 new

Example 2

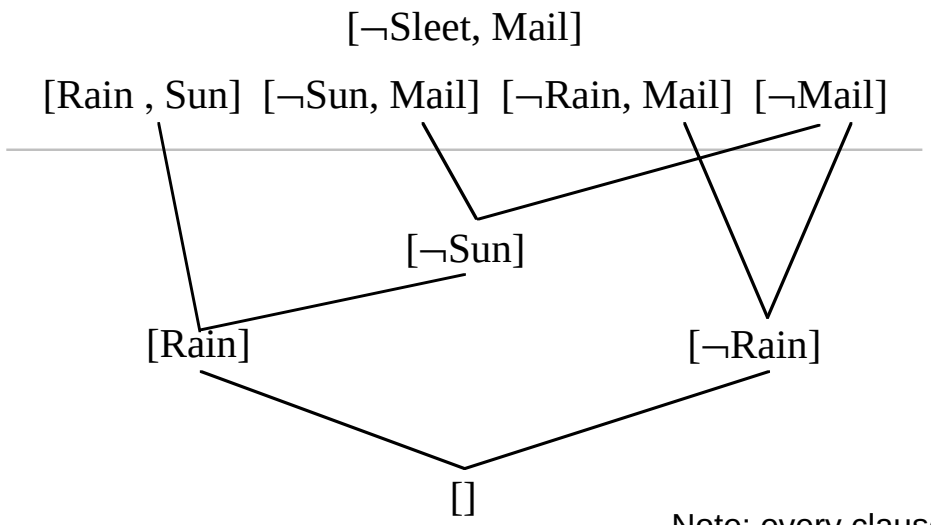
KB

(Rain $\vee$ Sun)

(Sun $\supset$ Mail)

((Rain $\vee$ Sleet) $\supset$ Mail)

Show KB |= Mail



Note: every clause not in *S* has 2 parents

Similarly KB |≠ Rain

Can enumerate all resolvents given ¬Rain, and [] will not be generated

Quantifiers

Clausal form as before, but atom is $P(t_1, t_2, ..., t_n)$, where t_i may contain variables

Interpretation as before, but variables are understood *universally*

Example: $\{ [P(x), \neg R(a,f(b,x))], [Q(x,y)] \}$

interpreted as

$\forall x \forall y \{ [R(a,f(b,x)) \supset P(x)] \wedge Q(x,y) \}$

Substitutions: $\theta = \{v_1/t_1, v_2/t_2, ..., v_n/t_n\}$

Notation: If ρ is a literal and θ is a substitution, then $\rho\theta$ is the result of the substitution (and similarly, $c\theta$ where c is a clause)

Example: $\theta = \{x/a, y/g(x,b,z)\}$

$P(x,z,f(x,y))\,\theta = P(a,z,f(a,g(x,b,z)))$

A literal is ground if it contains no variables.

A literal ρ is an instance of ρ', if for some θ, $\rho = \rho'\theta$.