If fido goes wherever John goes, and if John is at school, where is fido

$(\forall x)$ (Location (John, x) → Location (Fido, x)) — ①

Location (John, SCHOOL) — ②

v$(\exists x)$ (Location (Fido, x)) — ③

① $\exists x$ (~ Location(John, x) ∨ Location (Fido, x))

③ $\forall x$ (~Location (fido, x))

Resolve

∨ Locate(fido ?x)

~Location (fido, x)   ~Location (John, y) ∨ Location(fido, y)

/ y/x

~Locate (John, x) ∨ Locate (fido, school)

~ Location (John, x)   Location (John, SCHOOL)

Locate (fido, school)

x/SCHOOL

NIL

~ Locate (fido, x) ∨ Locate (fido, x)

$(\forall x)$ (Read/$R(x) \rightarrow L(x)$) (Peopl who can not are intelligence)

$(\forall x)$ ($\sim R(x)$

Some dolphin are intelligent $(\exists x)$ (dol(x) $\wedge$ intell(x))

Some who are intelligent cannot

$(\exists x)$ ($I(x) \wedge \sim R(x)$

$\sim$ ( )

$\sim \forall(x)$ ($\sim I(x) \vee \sim R(x)$

Remov

$\sim I(z) \vee \sim R(z)$

1. $\sim R(x) \vee L(x)$
2. $\sim D(y) \vee \sim L(y)$

$D(A)$, $\cancel{I(A)}$

$I(A)$

$\sim I(z) \vee R(z)$

Pick up any two clauses and keep resolving.

$\sim R(x) \vee L(x)$ $\sim D(y) \vee \sim L(y)$

y/x

$\sim I(z) \vee R(z)$

$\sim R(x) \sim D(y)$

z/y

$D(A)$

$\sim D(y) \vee \sim I(y)$

$I(A)$

y/A

$\sim I(A)$

NIL

Question: Skolemization

- $(\forall x)(\exists y)loves(x,y)$
- $\forall x1\ \forall x2\ \forall x3\ \exists y\ P(\ldots\ y\ \ldots)$

Produce Resolvent from 2 Parent Clauses

Parent Clauses	Resolvent	Comment
P and ~P or Q	Q	Modus ponens
(P or Q) and (~P or Q)	Q	Merge operation
(P or Q) and (~P or ~Q)	Q or ~Q and P or ~P	tautologies
P and ~P	NIL	empty clause; sign of contradiction
(~P or Q) and (~Q or R)	~P or R	Chaining

Resolution Refutation System

- One type of theorem proving system.
- Designed to produce proofs by contradictions/ refutations

We have a set, S, of wffs from which we wish to prove some goal wff, W

1. Negate W and add it to S
2. Convert new S to a set of clauses

--- attempt to derive a contradiction represented by NIL

Key idea: If W logically follows from S, the set S U{~W} is unsatisfiable.

Therefore, if empty clause NIL is produced from S U{~W}, then W logically follows from S.

Production Systems for Resolution Refutations

- Let S be the set of clauses (base set)

Algo:

- 1. Clauses = S
- 2. until NIL is a member of Clauses do:
- 3. begin
- 4. Select two distinct resolvable clauses, Ci and Cj, from Clauses
- 5. Compute a resolvent r_{ij} to Clauses
- 6. Clauses = The set produced by adding r_{ij} to Clauses
- 7. end

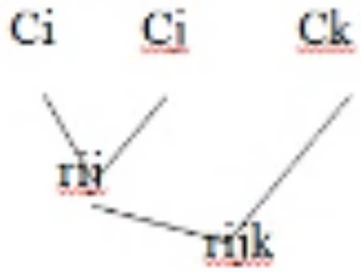
Control Strategies for Resolution Refutation

- The decision about which two clauses in Clauses to resolve and which resolution of these clauses to perform, are done irrevocably by the control strategy.
- Control strategy uses derivation graph

  nodes – clauses

Initially, for every clause in the base set, a node exists

From Ci and Cj create rij →



In refutation tree root is NIL

A control strategy for a refutation tree is said to be complete if its use results in a procedure that will find a contradiction (eventually) whenever one exists

Example

1. Whoever can read is literate
2. Dolphins are not literate
3. Some dolphins are intelligent

Prove that,

- Some who are intelligent cannot read

Sound and Complete

- Resolution is a sound rule of inference---- the resolvent of a pair of clauses also logically follows from the pair of clauses

- When resolution is used in a special kind of theorem proving system, and called a refutation system, it is also complete ----- Every wff that logically follows from a set of wffs, can be derived from that set of wffs using resolution

- Breadth First Strategy

Compute all first level resolvents (between two clauses in the base set), then second level… and so on

ith level resolvent → deepest parent is an (i-1)th level resolvent

Complete but inefficient

- Linear Input Form Strategy

Each resolvent has at least one parent belonging to the base set. At subsequent levels, it reduces the number of clauses produced

Not complete

Example:

~Q(x) U ~P(x) , Q(y) U ~P(y), ~Q(w) U P(w), Q(u) U P(A)

Control Strategies for Resolution Refutation

- Set of Support Strategy

At least one parent of each resolvent is selected from negation of goal wff or from their descendants.

Complete

- Unit Preference Strategy

Modification of set of support; instead of filling out each level in breadth first, try to select a single literal clause (unit) to be a parent in resolution

Complete, typically increases efficiency

- Ancestry Filtered Form Strategy

Each resolvent has a parent that is either in the base set or ancestor of other parent

- Combination of strategies

Extracting Answers from Resolution Refutation

- If Fido goes wherever John goes, and if John is at School, where is Fido?

1. Append to each clause arising from the negation of goal wff, its own negation
2. Following the structure of the refutation tree, perform the same resolutions as before until some clause is obtained at the root
3. Use the clause at the root as an answer statement