Top-down Definite Clause Proof Procedure

Idea: search backward from a query to determine if it is a logical consequence of KB.
An **answer clause** is of the form:

$$yes \leftarrow a_1 \wedge a_2 \wedge \ldots \wedge a_m$$

The **SLD Resolution** of this answer clause on atom a_i with the clause:

$$a_i \leftarrow b_1 \wedge \ldots \wedge b_p$$

is the answer clause

$$yes \leftarrow a_1 \wedge \cdots \wedge a_{i-1} \wedge b_1 \wedge \cdots \wedge b_p \wedge a_{i+1} \wedge \cdots \wedge a_m.$$

Derivations

- An answer is an answer clause with $m = 0$. That is, it is the answer clause $yes \leftarrow$.
- A derivation of query "$?q_1 \wedge \ldots \wedge q_k$" from KB is a sequence of answer clauses $\gamma_0, \gamma_1, \ldots, \gamma_n$ such that
  - γ_0 is the answer clause $yes \leftarrow q_1 \wedge \ldots \wedge q_k$,
  - γ_i is obtained by resolving γ_{i-1} with a clause in KB, and
  - γ_n is an answer.

Top-down definite clause interpreter

To solve the query $?q_1 \wedge \ldots \wedge q_k$:

$ac := \text{“}yes \leftarrow q_1 \wedge \ldots \wedge q_k\text{”}$
repeat
    select atom a_i from the body of ac;
    choose clause C from KB with a_i as head;
    replace a_i in the body of ac by the body of C
until ac is an answer.

Nondeterministic Choice

- Don't-care nondeterminism If one selection doesn't lead to a solution, there is no point trying other alternatives. select
- Don't-know nondeterminism If one choice doesn't lead to a solution, other choices may. choose

Example: successful derivation

$a \leftarrow b \wedge c.$
$c \leftarrow e.$
$f \leftarrow j \wedge e.$

$a \leftarrow e \wedge f.$
$d \leftarrow k.$
$f \leftarrow c.$

$b \leftarrow f \wedge k.$
$e.$
$j \leftarrow c.$

Query: $?a$

γ_0 : $yes \leftarrow a$
γ_1 : $yes \leftarrow e \wedge f$
γ_2 : $yes \leftarrow f$
γ_3 : $yes \leftarrow c$

γ_4 : $yes \leftarrow e$
γ_5 : $yes \leftarrow$

Example: failing derivation

$a \leftarrow b \wedge c.$
$c \leftarrow e.$
$f \leftarrow j \wedge e.$

$a \leftarrow e \wedge f.$
$d \leftarrow k.$
$f \leftarrow c.$

$b \leftarrow f \wedge k.$
$e.$
$j \leftarrow c.$

Query: $?a$

γ_0 : $yes \leftarrow a$
γ_1 : $yes \leftarrow b \wedge c$
γ_2 : $yes \leftarrow f \wedge k \wedge c$
γ_3 : $yes \leftarrow c \wedge k \wedge c$

γ_4 : $yes \leftarrow e \wedge k \wedge c$
γ_5 : $yes \leftarrow k \wedge c$

Search Graph for SLD Resolution

$a \leftarrow b \wedge c.$
$a \leftarrow h.$
$b \leftarrow k.$
$d \leftarrow p.$
$f \leftarrow p.$
$g \leftarrow f.$
$h \leftarrow m.$
$?a \wedge d$

$a \leftarrow g.$
$b \leftarrow j.$
$d \leftarrow m.$
$f \leftarrow m.$
$g \leftarrow m.$
$k \leftarrow m.$
$p.$