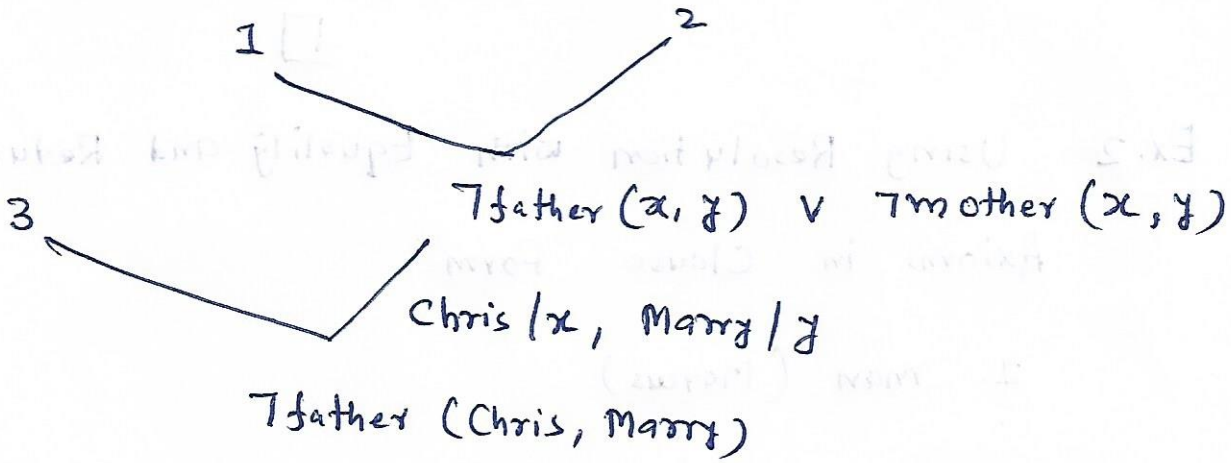
Needs to standardize variables

Example (Without standardizing)

1. father $(x, y) \rightarrow \neg$ woman (x)
2. mother $(x, y) \rightarrow$ woman (x)
3. mother (Chris, Marry)
4. father (Chris, Bill)

Clauses

1. $\neg$father $(x, y) \vee \neg$ woman (x)
2. $\neg$mother $(x, y) \vee$ woman (x)
3. mother (Chris, Marry)
4. father (Chris, Bill)

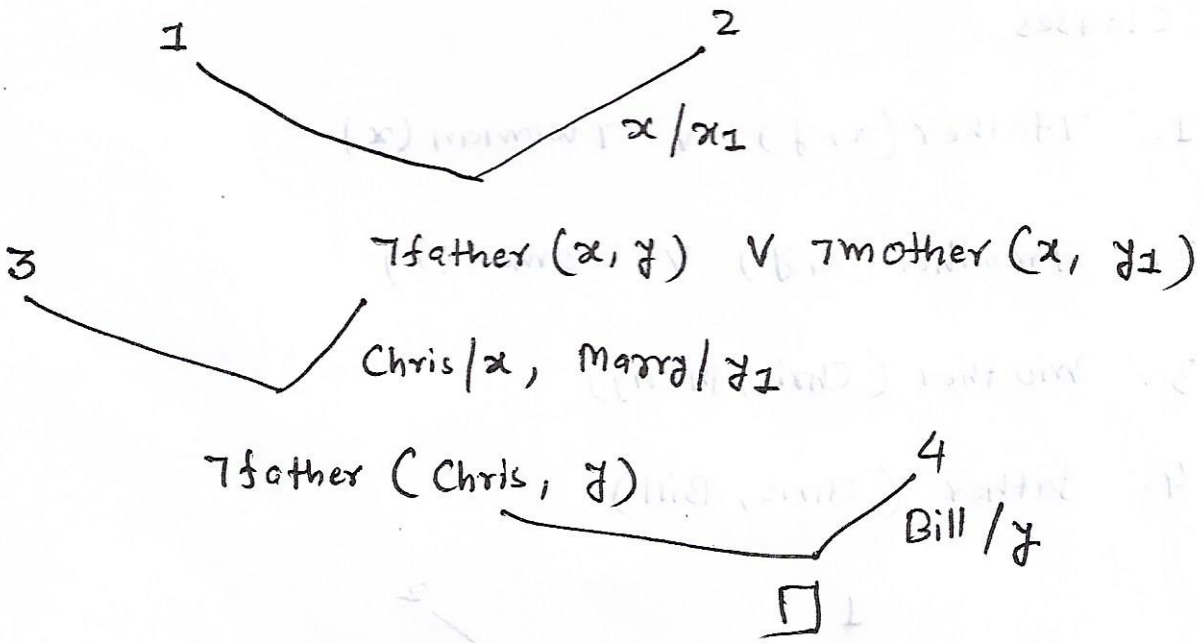


↳ Can't be resolved further

(Contradiction present in the KB can't be detected)

Same Example (With standardizing)

1. $\neg father(x, y) \vee \neg woman(x)$
2. $\neg mother(x_1, y_1) \vee woman(x_1)$
3. mother (Chris, Marry)
4. father (Chris, Bill)



Ex. 2 Using Resolution with Equality and Reduce Axioms in Clause Form

1. man (Marcus)
2. pompeian (Marcus)
3. born (Marcus, 40)
4. $\neg man(x_1) \vee mortal(x_1)$

5. $\neg pompeian(x_2) \vee died(x_2, 79)$

6. $erupted(volcano, 79)$

7. $\neg mortal(x_3) \vee \neg born(x_3, t_1) \vee \neg gt(t_2 - t_1, 150) \vee dead(x_3, t_2)$

8. $now = 2020$

alive → ¬dead

9a. $\neg alive(x_4, t_3) \vee \neg dead(x_4, t_3)$

Not Req^d

~~9b~~. $dead(x_5, t_4) \vee alive(x_5, t_4)$

10. $\neg died(x_6, t_5) \vee \neg gt(t_6, t_5) \vee dead(x_6, t_6)$

To prove : $\neg alive(Marcus, now)$

$alive(Marcus, now)$ — 9a — Marcus/x_4, now/t_3

$\neg dead(Marcus, now)$ — 10 — Marcus/x_6, now/t_6

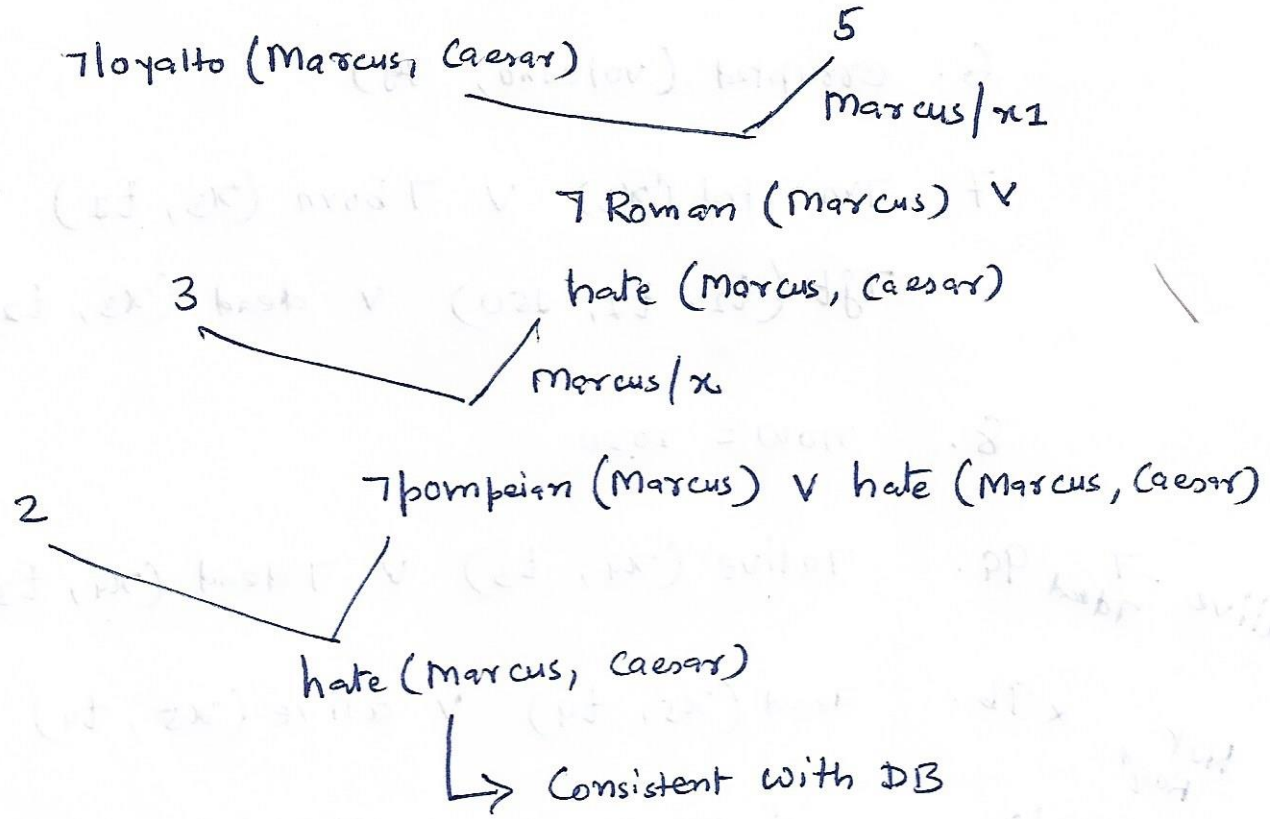
$\neg died(Marcus, t_5) \vee \neg gt(now, t_5)$ — 5 — Marcus/x_2, 79/t_5

$\neg pompeian(Marcus) \vee \neg gt(now, 79)$ — {None} — Substitute Equal (8)

$\neg pompeian(Marcus) \vee \neg gt(2020, \ldots)$ — {None} — Reduce

$\neg pompeian(Marcus)$ — 2 — □

Unsuccessful attempt at Resolution



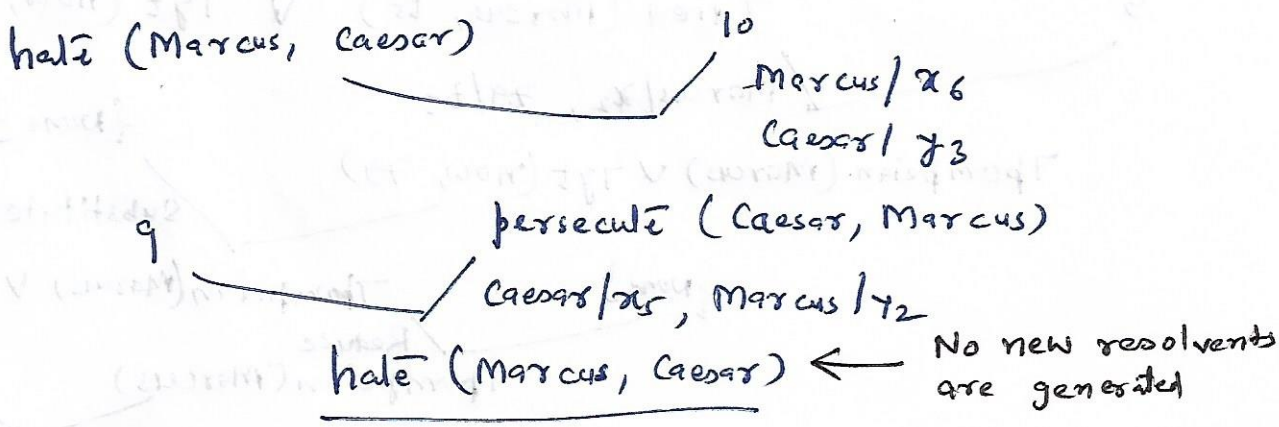
New wff

9. $persecute(x, y) \rightarrow hate(y, x)$

$\neg persecute(x_5, y_2) \vee hate(y_2, x_5)$

10. $hate(x, y) \rightarrow persecute(y, x)$

$\neg hate(x_6, y_3) \vee persecute(y_3, x_6)$



Trying several substitutions

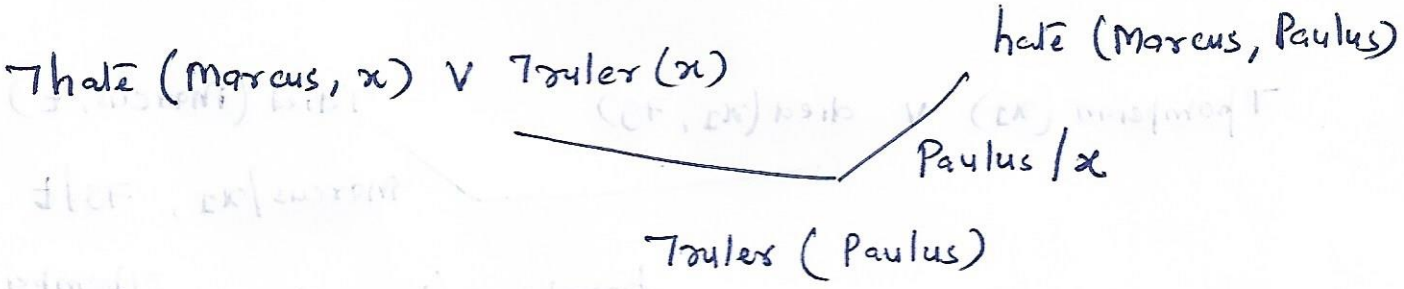
Que: Is there any ruler who is being hated by Marcus?

To prove: $\exists x$: hate (Marcus, x) $\wedge$ ruler (x)

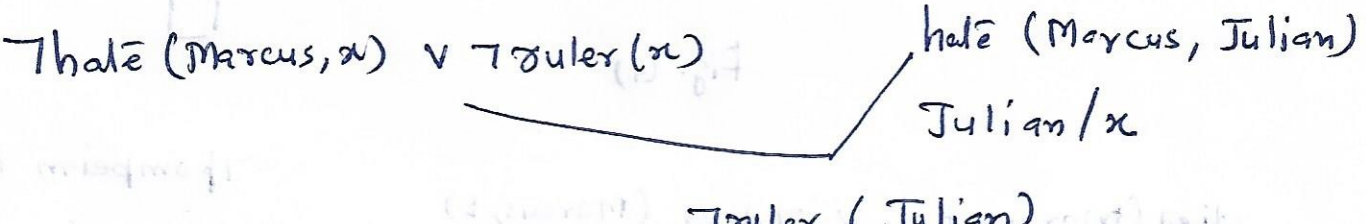
Negate: $\neg \exists x$: hate (Marcus, x) $\wedge$ ruler (x)

$\equiv$ $\forall x$: $\neg$ [hate (Marcus, x) $\wedge$ ruler (x)]

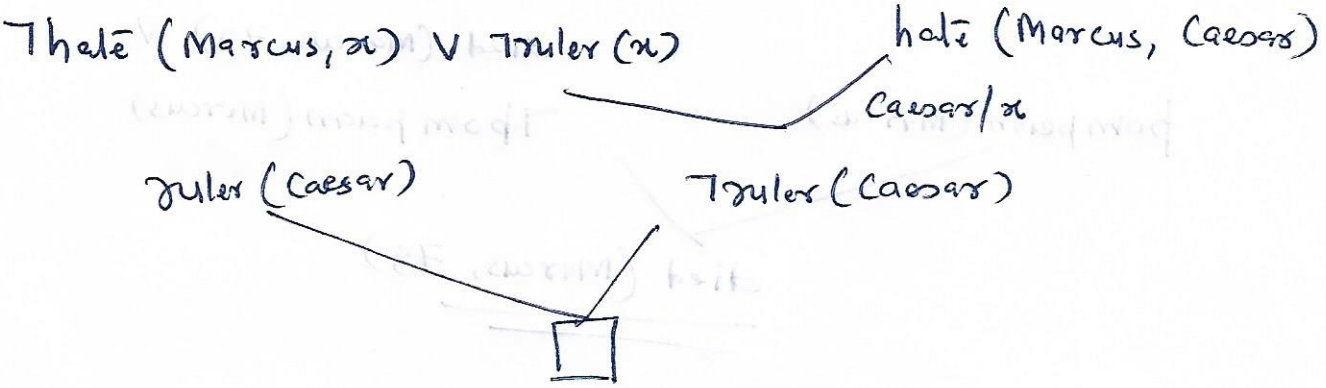
$\equiv$ $\neg$hate (Marcus, x) $\vee$ $\neg$ruler (x)



(a)



(b)



Answer Extraction using Resolution

Fill in the blank type question

When did Marcus die?

$\exists t$: died (Marcus, t)

Negate : $\neg \exists t$: died (Marcus, t)

$\equiv \forall t$: $\neg$ died (Marcus, t)

$\equiv \neg$ died (Marcus, t)

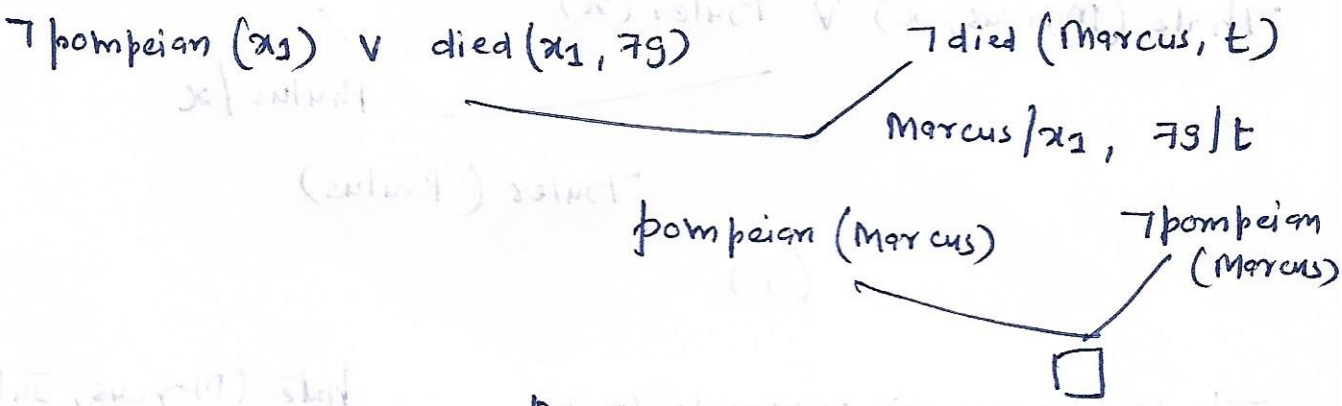
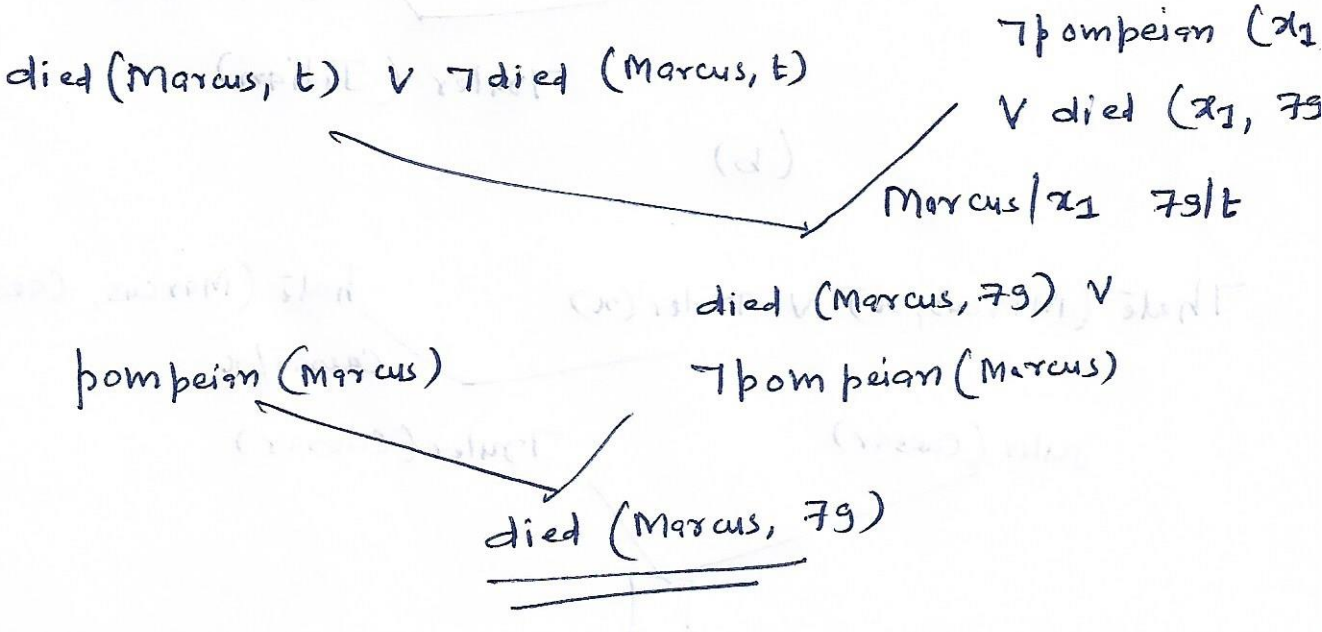


Fig. (4)



To answer question

What courses would Steve like?

1. Steve likes easy courses

$\forall x$: easycourse $(x) \rightarrow$ likes (steve, x)

2. Science courses are hard

$\forall x$: sciencecourse $(x) \rightarrow$ hard (x)

3. Basket weaving courses are easy.

$\forall x$: bwcourse $(x) \rightarrow$ easycourse (x)

4. BK 301 is a Basketweaving course

bwcourse (BK 301)

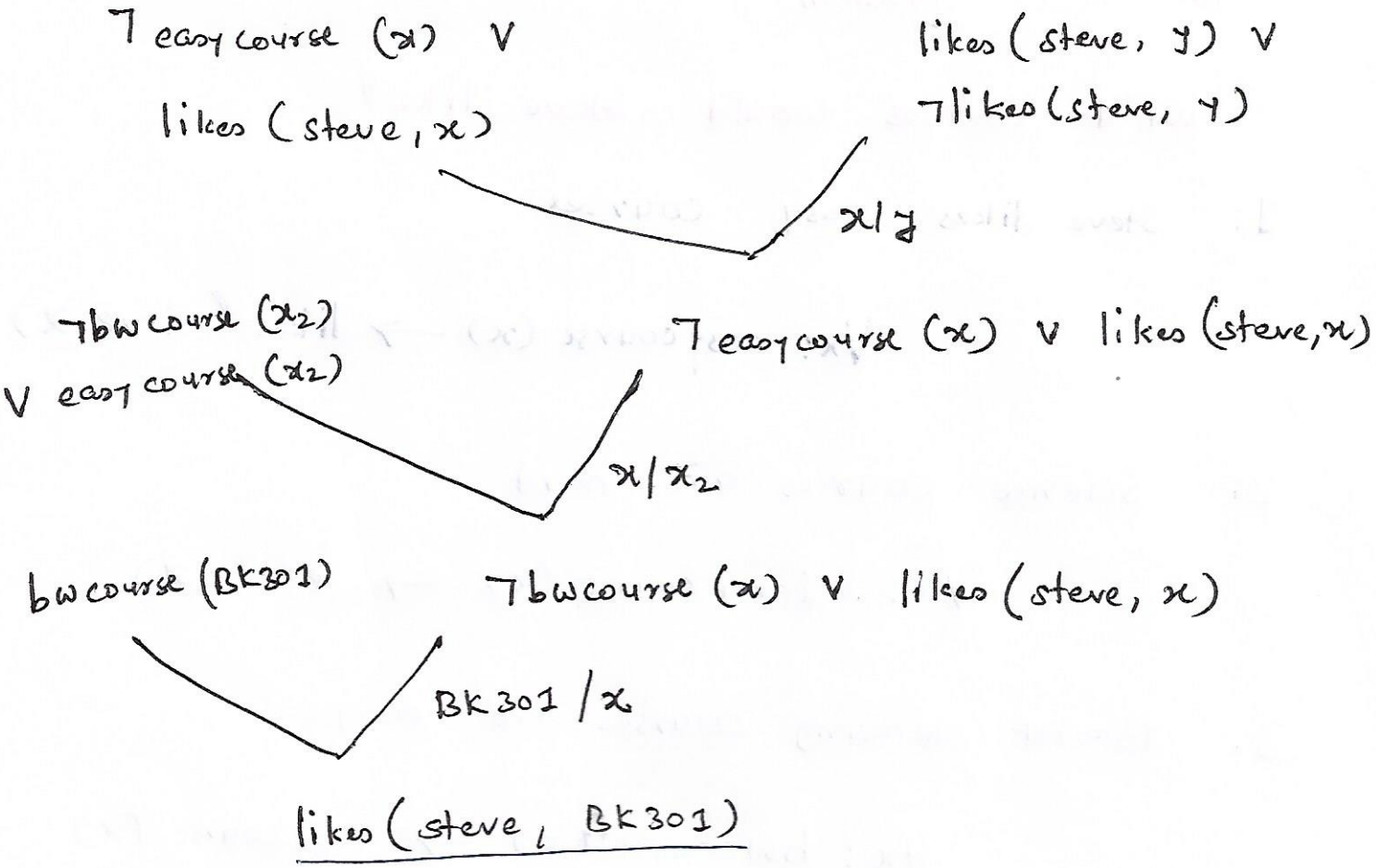
Clauses

1. $\neg$ easycourse (x) $\vee$ likes (steve, x)

2. $\neg$sciencecourse (x_1) $\vee$ hard (x_1)

3. $\neg$bwcourse (x_2) $\vee$ easycourse (x_2)

4. bwcourse (BK 301)



Can we answer the question

"What happened in 79 A.D.?"

We need higher order predicate calculus;

i.e. quantification over predicates / functions.

event (erupted (volcano), 79)

Then, we can extract answer using Resolution

event (x, 79)
V ¬event (x, 79)

event (errupted (volcano), 79)

event (errupted (volcano), 79)

Limitation of Resolution

- The procedure is not natural to humans.

If the procedure is used to prove a theorem, Then possibly, it would require interaction with human and more likely humans would not be able to help it.

Forward chaining

From Wikipedia, the free encyclopedia
This article is about forward chaining inference engines. For forward chaining as an instructional procedure, see Chaining.

Forward chaining (or **forward reasoning**) is one of the two main methods of reasoning when using an inference engine and can be described logically as repeated application of *modus ponens*. Forward chaining is a popular implementation strategy for expert systems, business and production rule systems. The opposite of forward chaining is backward chaining.

Forward chaining starts with the available data and uses inference rules to extract more data (from an end user, for example) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (**If** clause) is known to be true. When such a rule is found, the engine can conclude, or infer, the consequent (**Then** clause), resulting in the addition of new information to its data.[1]

Inference engines will iterate through this process until a goal is reached.

For example, suppose that the goal is to conclude the color of a pet named Fritz, given that he croaks and eats flies, and that the rule base contains the following four rules:

1. **If** X croaks and X eats flies - **Then** X is a frog
2. **If** X chirps and X sings - **Then** X is a canary
3. **If** X is a frog - **Then** X is green
4. **If** X is a canary - **Then** X is yellow

Let us illustrate forward chaining by following the pattern of a computer as it evaluates the rules. Assume the following facts:

- Fritz croaks
- Fritz eats flies

With forward reasoning, the inference engine can derive that Fritz is green in a series of steps:

1. Since the base facts indicate that "Fritz croaks" and "Fritz eats flies", the antecedent of rule #1 is satisfied by substituting Fritz for X, and the inference engine concludes:

```
Fritz is a frog
```

2. The antecedent of rule #3 is then satisfied by substituting Fritz for X, and the inference engine concludes:

```
Fritz is green
```

The name "forward chaining" comes from the fact that the inference engine starts with the data and reasons its way to the answer, as opposed to backward chaining, which works the other way around. In the derivation, the rules are used in the opposite order as compared to backward chaining. In this example, rules #2 and #4 were not used in determining that Fritz is green.

Because the data determines which rules are selected and used, this method is called data-driven, in contrast to goal-driven backward chaining inference. The forward chaining approach is often employed by expert systems, such as CLIPS.

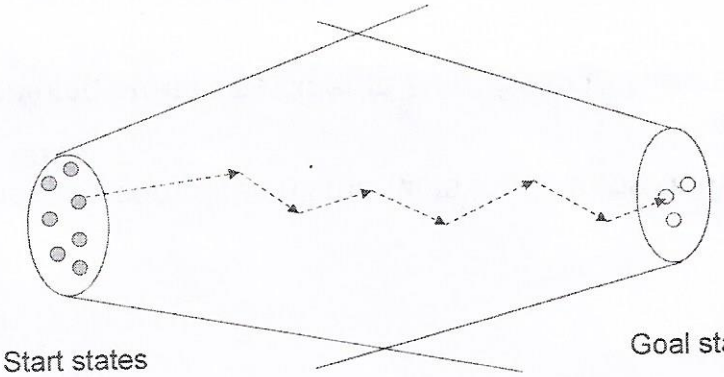
One of the advantages of forward-chaining over backward-chaining is that the reception of new data can trigger new inferences, which makes the engine better suited to dynamic situations in which conditions are likely to change.[2][3]

Forward Chaining Strategies

- Forward chaining computes all the facts that can be derived from the knowledge base
- Forward chaining strategies differ in step „Select". Here are some examples of strategies:
  - Apply the rules sequentially
  - Randomly select a rule
  - Apply more specific rules first
  - Prefer rules where conditions match a recently derived fact
  - Derive consequences of a set of starting facts: Only apply rules where at least one condition matches either with a starting fact or a derived fact
    - Fact base contains facts that are generally true, e.g. insurance product
    - Starting facts describe a concrete situation, e.g. customer data

Choosing Forward or Backward Chaining

- Backward Chaining
  - If you already know what you are looking for
- Forward Chaining
  - If you don't necessarily know the final state of your solution



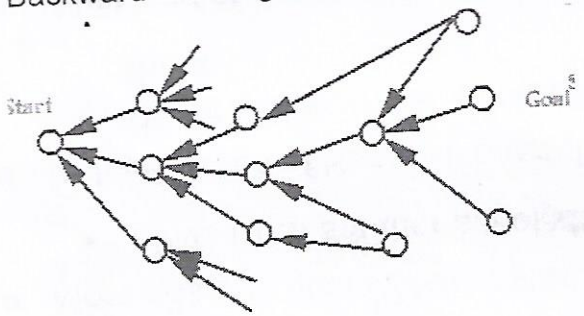
Decision Criteria for Forward or Backward Reasoning

- More possible goal states or start states?
  - Move **from smaller** set of states **to the larger**
- Is Justification of Reasoning required?
  - Prefer direction that corresponds **more closely to the way users think**
- What kind of events triggers problem-solving?
  - If it is **arrival of a new fact**, forward chaining makes sense.
  - If it is **a query to which a response is required**, backward chaining is more natural.
- In which direction is branching factor greatest?
  - Go in direction with lower branching factor

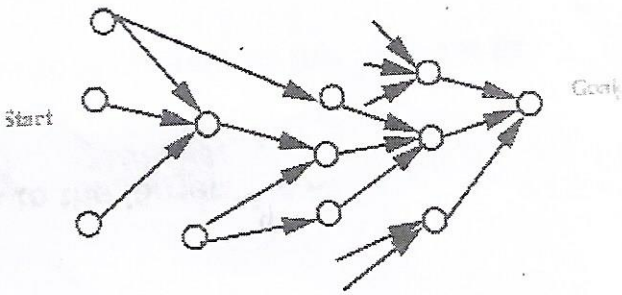
Source: Kerber (2004), http://www.cs.bham.ac.uk/~mmk/Teaching/AI/l2.html

Branching Factor

Backward chaining more appropriate



Forward chaining more appropriate



a conflict resolution by taking the first rule found in the knowledge base, and then followed the results of that rule. The forward-chaining starts with an empty WM.

Thus, we examine if the premise of R1 "The applicant's UG is receiving a bachelor's degree from US" is askable, and the answer of this query is "true", according to the applicant's details. Then "The applicant's UG is receiving a bachelor's degree from US" is placed in the WM, as is the conclusion of R1 "The language requirement met" because one of the two OR premises is true as in Fig. 12.

Working memory

The applicant's UG is receiving a bachelor's degree from US
The language requirement met

Fig. 12. After R1 is placed in WM.

Now the control moves to R2. The premise "The language requirement met" is not askable, so rule 2 fails and the control moves to R3. The premise of R3 "The language requirement met" is also not askable, so R3 fails and the control moves to R4. The premise of R4 "Conditional decision" is also not askable, so R4 fails and the control moves to R5. The premise of R5 "Conditional decision" is also not askable, so R5 fails and the control moves to the next rule. In Rules 6 to 8, the premises are not askable so they also fail and the control moves to R9. The premise of R9 "The applicant is not prior graduate work completed" is askable and according to the applicant's details the answer of this query is "true", so "The applicant is not prior graduate work completed" and the conclusion "The undergraduate GPR should be considered = true" are placed in the WM as in Fig. 13.

Working memory

The applicant's UG is receiving a bachelor's degree from US
The language requirement met
The applicant is not prior graduate work completed
The undergraduate GPR should be considered = true

Fig. 13. After R9 is placed in WM.

The control now moves to next rule. The premises of R10, 11 and 12 are not askable so they fail. By the end of the first pass we find that two rules are proved: R1 and R9.

At this point all rules have been considered, so the search now returns with the new contents of WM in order to consider the rules in their correct order a second time.

Since R1 is proved, the second pass will start with R2, and then compare the premises of R2 with the WM. According to the content of WM, the premise "The language requirement met" is "true", and the second premise "The applicant's final undergraduate transcript is not available" is askable, but its value is "false" because the applicant's final undergraduate transcript is available, according to the applicant's details. Therefore R2 has failed and the control moves to R3. The premise "The language requirement met" is "true" based on the content of the WM. The second premise "The applicant's final undergraduate transcript is available" is askable, and its value is "true" based on applicant's details, so this premise and the conclusion "Full decision" will be placed in the WM, and the control moves to the next rule. The R4, 5, 6 and 7 have the premise "Conditional Decision" which is the conclusion of R2, and R2 failed to match the applicant's details. This means that the 4 rules also will also fail because of one AND premise, and the control moves to R8. The first two premises of R8 match the content of the WM, and the other 4 premises are askable. Based on the applicant's details, all four premises are true, so the goal of R8 is placed in the WM, and the control then moves to R10 because R9 was proved in the first pass. The premise of R10 does not match the content of WM, so it fails and the control moves to R11. The first premise of R11 "The undergraduate GPR should be considered = true" matches the content of the WM. The other premises of this rule are askable. Based on the applicant's details, the premise "The applicant's GPR on last 60 undergraduate hours >2.49" is "true", the premise "The applicant is not recipient of special honors" is "true", the premise "The applicant's GRE score>899" is "true", the premise "The applicant's GRE score< 1100" is "false", and the premise "The applicant's GRE verbal score < 400" is "false". Since two AND premises are false, the rule fails and the control moves to R12. The premise of R12 "The undergraduate GPR should be considered = true" matches the content of WM. The second premise is ask able but it does not match with the applicant's details, so this rule also fails and the second pass is completed.

By the end of pass two, all rules are fired and their premises are checked. We find that R8 matches the details of the applicant, so its goal is the best decision for this applicant, which is "Admit with full Status".

VI. Result

The main points needed for a comparison of forward-chaining and backward-chaining are summarised in the Table II.

TABLE II: The Comparison between Forward-Chaining and Backward-Chaining

Forward-chaining	Backward-chaining
Starts with the initial facts.	Starts with some hypothesis or goal.
Asks many questions.	Asks few questions.
Tests all the rules.	Tests some rules.
Slow, because it tests all the rules.	Fast, because it tests fewer rules.
Provides a huge amount of information from just a small amount of data.	Provides a small amount of information from just a small amount of data.
Attempts to infer everything possible from the available information.	Searches only that part of the knowledge base that is relevant to the current problem.
Primarily data-driven	Goal-driven
Uses input, searches rules for answer	Begins with a hypothesis, seeks information until the hypothesis is accepted or rejected.
Top-down reasoning	Bottom-up reasoning
Works forward to find conclusions from facts	Works backward to find facts that support the hypothesis
Tends to be breadth-first	Tends to be depth-first
Suitable for problems that start from data collection, e.g. planning, monitoring, control	Suitable for problems that start from a hypothesis, e.g. diagnosis
Non-focused because it infers all conclusions, may answer unrelated questions	Focused; questions all focused to prove the goal and search as only the part of KB that is related to the problem
Explanation not facilitated	Explanation facilitated
All data is available	Data must be acquired interactively (i.e. on demand)
A small number of initial states but a high number of conclusions	A small number of initial goals and a large number of rules match the facts
Forming a goal is difficult	Easy to form a goal

A wff can be:

(i) Valid (Tautology): It is true under all interpretations.

eg. $P \vee \neg P$

Above formula is always true irrespective of the interpretation. Hence it is Tautology.

(ii) Invalid: Formula is False under atleast one interpretation

(iii) Contradiction: " " " " all interpretations.

(iv) Satisfiable: Formula is true under atleast one interpretation

(v) Not Satisfiable; " " False under all interpretations.

Valid → Satisfiable

¬ Satisfiable → Invalid

¬ satisfiable ↔ Contradiction

An interpretation is a model of wff it is true under that interpretation.

If it is false, then it is a counter-model for that wff.

- For Tautologies, all interpretations are <u>models</u>.
- A formula is Valid if its negation is a Contradiction (Used in Resolution)

Examples

1. $\forall x : P(x)$

$P(x)$ is relation x is odd.

$D = \{0, 1, 2, \ldots\}$

The formule is false under above interpretation.

(∵ All numbers are not odd)

2 a. $\forall x : \exists y : P(x, y)$

b. $\exists y : \forall x : P(x, y)$

(i) $D = \{0, 1, 2\}$

and $P(x, y)$ is $x \leq y$

2a. True [∵ $x = 0$ $y = 0$ or 1 or 2

$x = 1$ $y = 1$ or 2

$x = 2$ $y = 2$]

2b. True [∵ $y = 2 \geq 0, 1, 2$]

(ii) $D = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

$P(x, y)$ is $x \leq y$

2a. True

2b. False

3. $\forall x : P(x, f(x))$

Interpretations

(i) Domain $D = \{0, 1, 2, \ldots\}$

Function : $f(x) = x + 1$

Predication : $P(x, y)$ is True iff $x \leq y$

True ($\because$ y is always $x+1$)

(ii) Domain $D = \{a, f(a), f(f(a)), \ldots\}$ ($f^1(a)$ $f^2(a)$ $f^3(a)$ etc. when f^4 is applied to it)

Function : $a = f^0(a)$ ... added afterwards

$f(a) = f^1(a)$

$f(f^n(a)) = f^{n+1}(a)$ $\forall x : P(x, f(x))$

Predicate $P(x, y)$ is True iff

$x = f^n(a)$, $y = f^m(a)$ and

$n \leq m$

True.

(PMI) 4. $(P(a) \wedge (\forall x : \underbrace{P(x) \rightarrow P(S(x))}_{I})) \rightarrow \forall x : P(x)$

Interpretations

(i) $D = \{0, 1, 2, \ldots\}$

$a = \phi$

$S \rightarrow$ Successor Function $S(x) = x + 1$

$P \rightarrow$ Any property