Artificial Intelligence Fundamentals

Problem Solving

Goal Trees

Rule-Based Expert Systems

Example 1

- Resolve the following indefinite integral

$$\int \frac{-5x^4}{(1-x^2)^{5/2}} dx$$

- Can you do it in your head?
- A program that can do that, it's an *intelligent* program?
- In 1963, Slagle James, "A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus.", presents SAINT (Symbolic Automation INTegrator)

The Problem-Reduction method

- Convert difficult goals into one or more easier-to-achieve subgoals
- Each subgoal may be divided still more finely into one or more lower-level subgoals
- Recognize goals -> convert them into appropriate subgoals
- Problem reduction -> goal reduction

Standards forms

- If an integral is into a “standard form” the goal is immediately achieved by substitution
- SAINT uses 26 standard forms

$$(a)\quad \int \frac{1}{x}\,dx = \ln(x)$$

$$(b)\quad \int x^n\,dx = \frac{x^{n+1}}{n+1}$$

$$(c)\quad \int \cos(x)\,dx = \sin(x)$$

$$\vdots$$

Simple (Safe) transformation

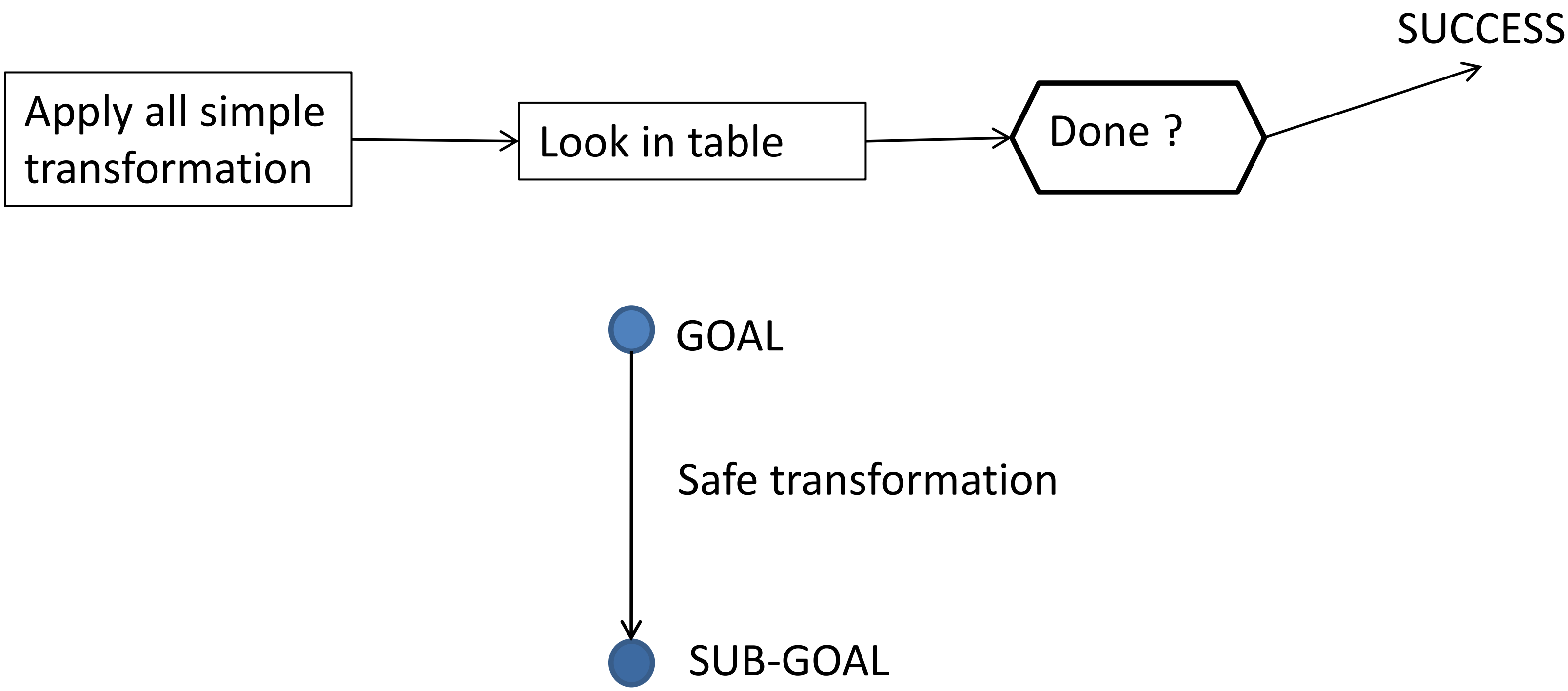
1. Deal with *minus* sign $$\int -g(x)dx = -\int g(x)dx$$
2. Take the constant out $$\int cg(x)dx = c\int g(x)dx$$
3. Decompose the sum $$\int \sum g_i(x)dx = \sum \int g_i(x)dx$$
4. Polynomial division $$\int \frac{P(x)}{Q(x)}dx \to DIVIDE$$

$\vdots$

Simple transformation

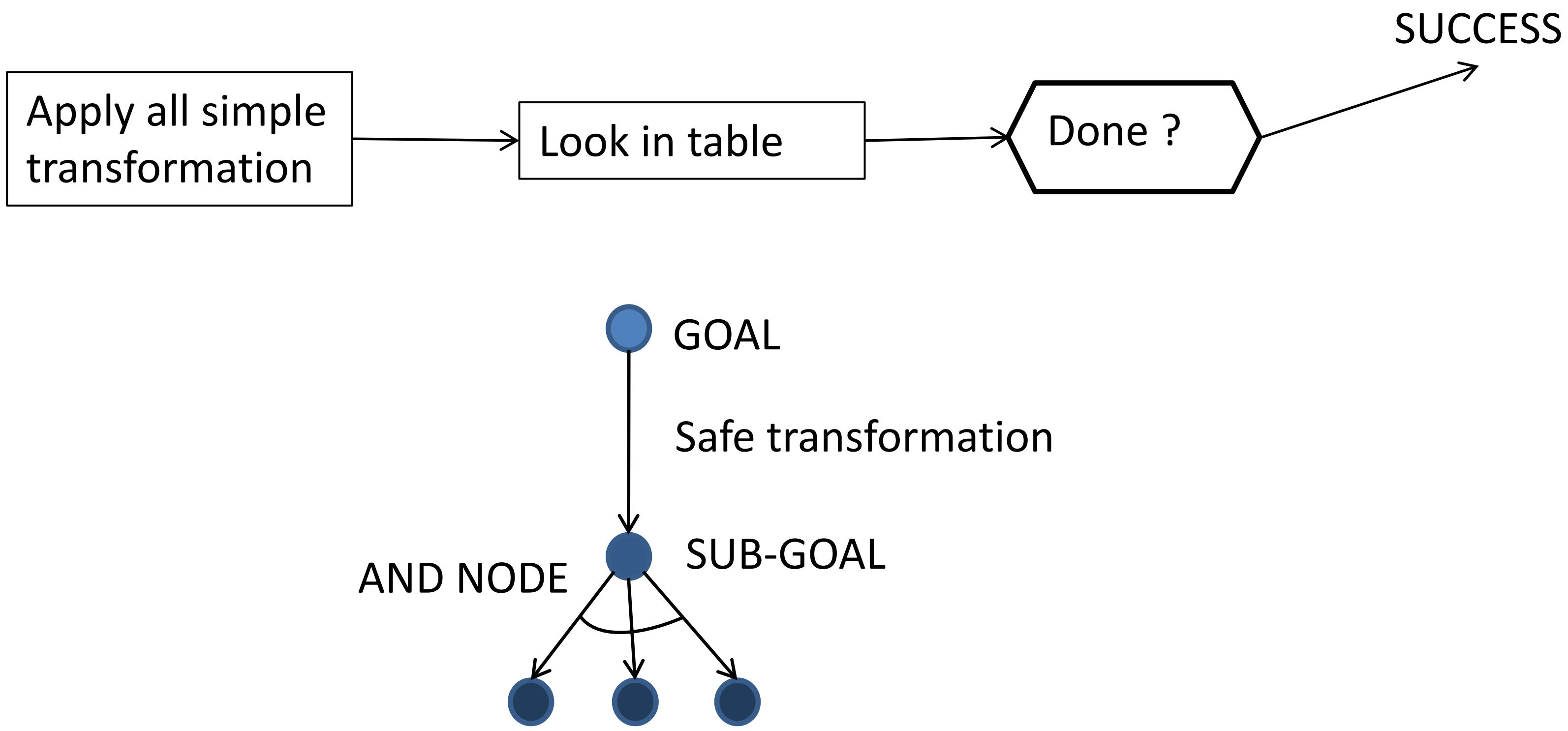
- a safe transformation – a transformation which, when applicable, is always or almost always appropriate
- for a goal, a transformation is appropriate if it's the correct next step to bring that goal nearer to achievement
- SAINT uses 8 simple transformations

The framework



- This version of framework doesn’t tell us anything about what happens for transformation no. 3

The framework - update



Problem reduction

- Apply all simple transformations

$$\int \frac{-5x^4}{(1-x^2)^{5/2}} \xrightarrow{1} \int \frac{5x^4}{(1-x^2)^{5/2}} \xrightarrow{2} \int \frac{x^4}{(1-x^2)^{5/2}}$$

Heuristic transformations

- Heuristic method - method that often work; isn’t guaranty that will always works
- Is not an algorithm – it’s an attempt

Heuristic transformation

A. Transformation from trigonometric form into other trigonometric form

$$f(sin(x), cos(x), tg(x), ctg(x), sec(x), cosec(x))$$
$$= g_1(sin(x), cos(x))$$
$$= g_2(tg(x), cosec(x))$$
$$= g_3(ctg(\mathrm{x}), sec(x))$$

B. Transformation from trigonometric form into polynomial form

$$\int f(tg(x))dx = \int \frac{f(y)}{1+y^2} dy$$

C. Transformation from polynomial form into trigonometric form

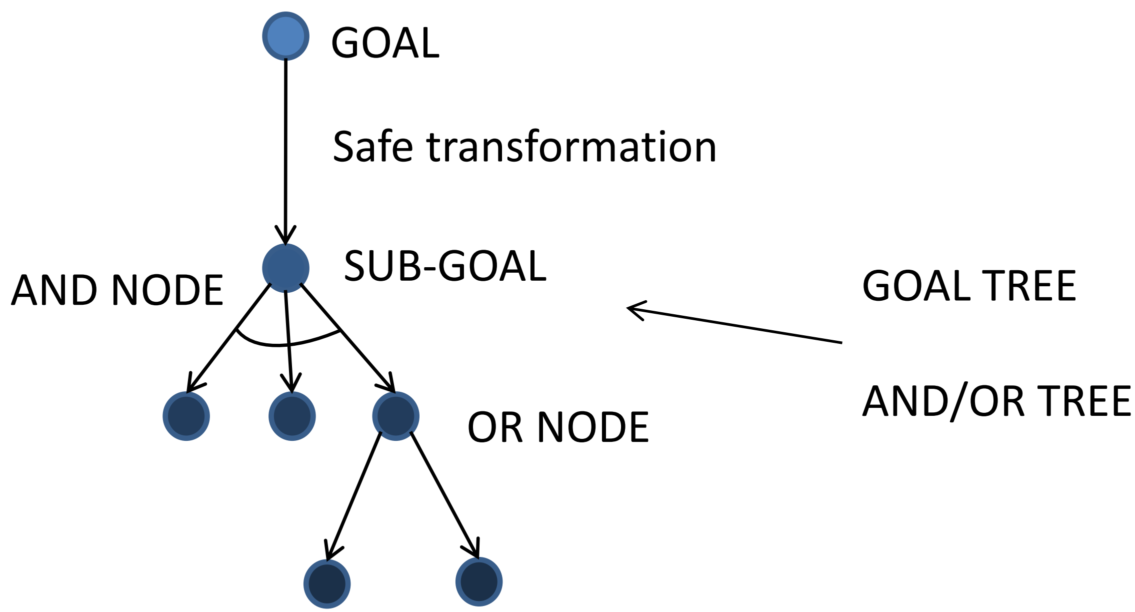
$$1 - x^2 \rightarrow x = sin(y) \qquad 1 + x^2 \rightarrow x = tg(y)$$

Problem reduction - continue

$$\int \frac{x^4}{(1-x^2)^{5/2}}\,dx \xrightarrow[\substack{x = \sin(y) \\ dx = \cos(y)dy}]{C} \int \frac{\sin^4(y)}{\cos^4(y)}\,dy$$

$$\int \frac{\sin^4(y)}{\cos^4(y)}\,dy \xrightarrow{A} \int \frac{1}{ctg^4(z)}\,dz$$

$$\int \frac{\sin^4(y)}{\cos^4(y)}\,dy \xrightarrow{A} \int tg^4(z)\,dz$$



Which one is easier ?

- Measure the depth of function composition
- The winner is *tg(x)*

$$\int tg^4(z)dz \xrightarrow[t=tg(z)]{B} \int \frac{t^4}{1+t^2}dt \xrightarrow{4} \int \left(t^2-1+\frac{1}{1+t^2}\right)dt$$

$$\xrightarrow{3}$$

$$\underset{b}{\int t^2 dt} \qquad \int -1dt \qquad \int \frac{1}{1+t^2}dt$$

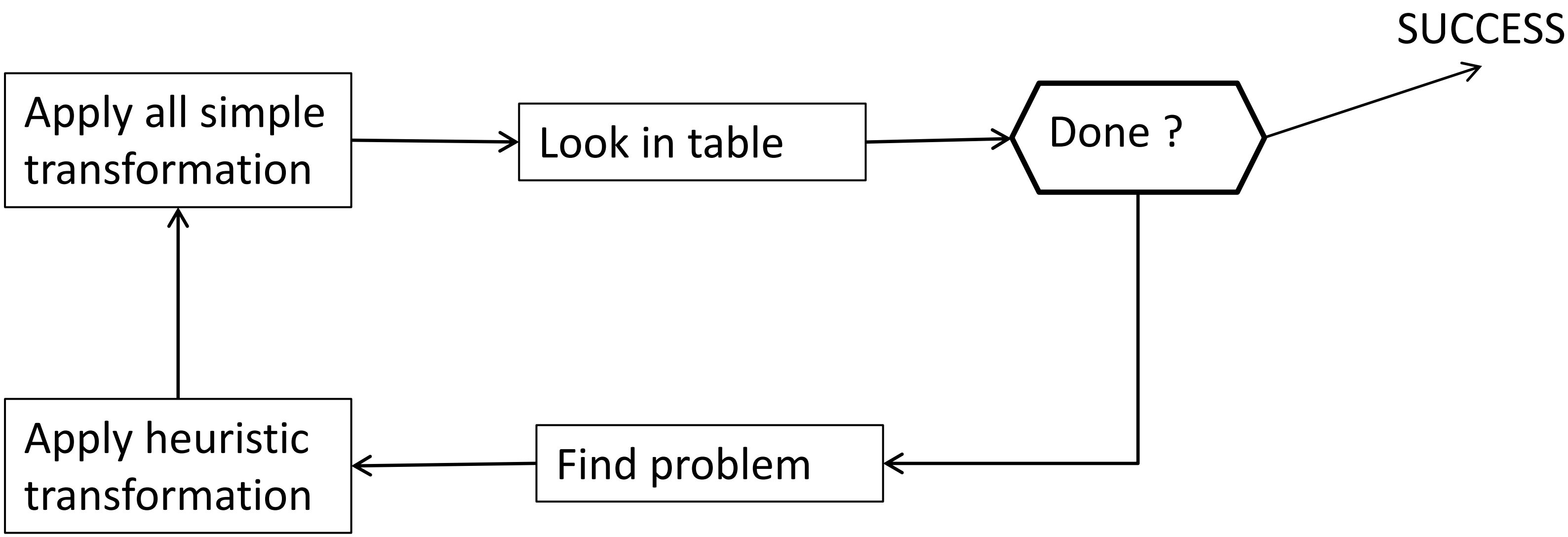
$$\int -1dt \xrightarrow{1} \underset{b}{\int dt}$$

Problem reduction - continue

$$\int \frac{1}{1+t^2}\,dt \xrightarrow[w = arctg(t)]{c} \int dw \quad b$$

- Before doing this transformation, SAINT compute the function composition (which is 3) and return to a previous choice $\int \frac{1}{ctg^4(z)}\,dz$ which have 2
- After doing some transformation, the function composition raises to 4, and because of that, SAINT returns to this transformation

The framework - update



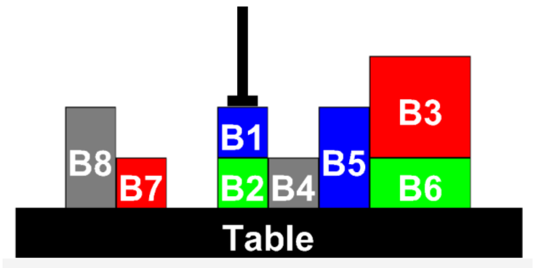
Questions

- How good the integration program is it?
  - 32K memory
  - Resolves 54 from 56 of the hardest problem (lacks 2 transformations)
  - Depth of the tree – 7 levels
  - Average depth – approx. 3
  - Branches unused – approx. 1
- What kind of knowledge is involved in this?
  - Knowledge about transformations
  - How goal tree works
  - Knowledge about standard forms
  - Knowledge about domain

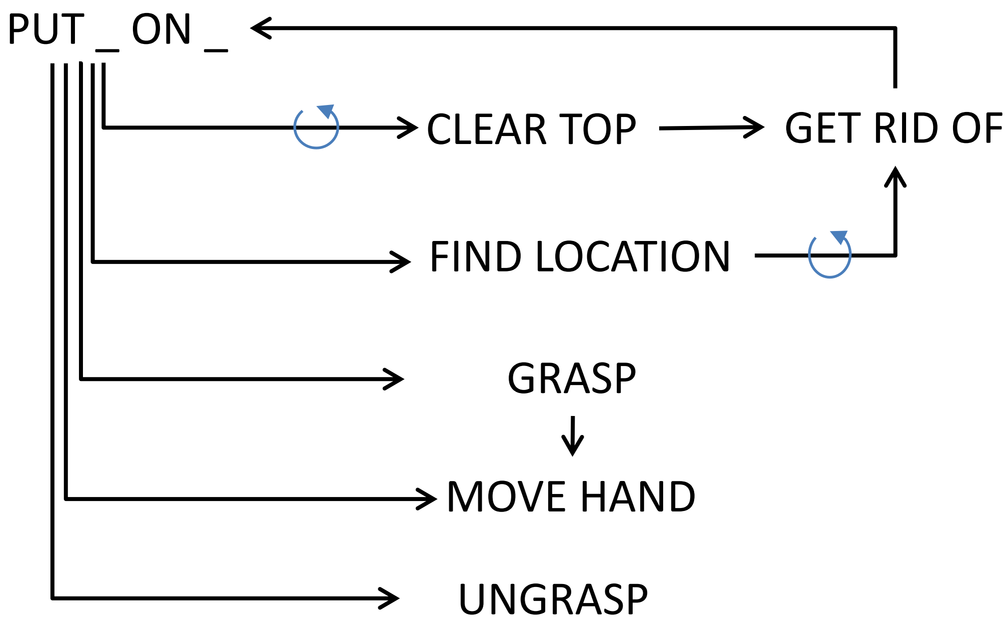
Questions

- How is the knowledge represented?
  - Table expressions
  - LISP relations
  - Goal tree information is embedded into procedures
- How much knowledge is required?
  - Table of integrals – standard form – 26
  - Simple transformations – 8
  - Heuristics transformations - 12

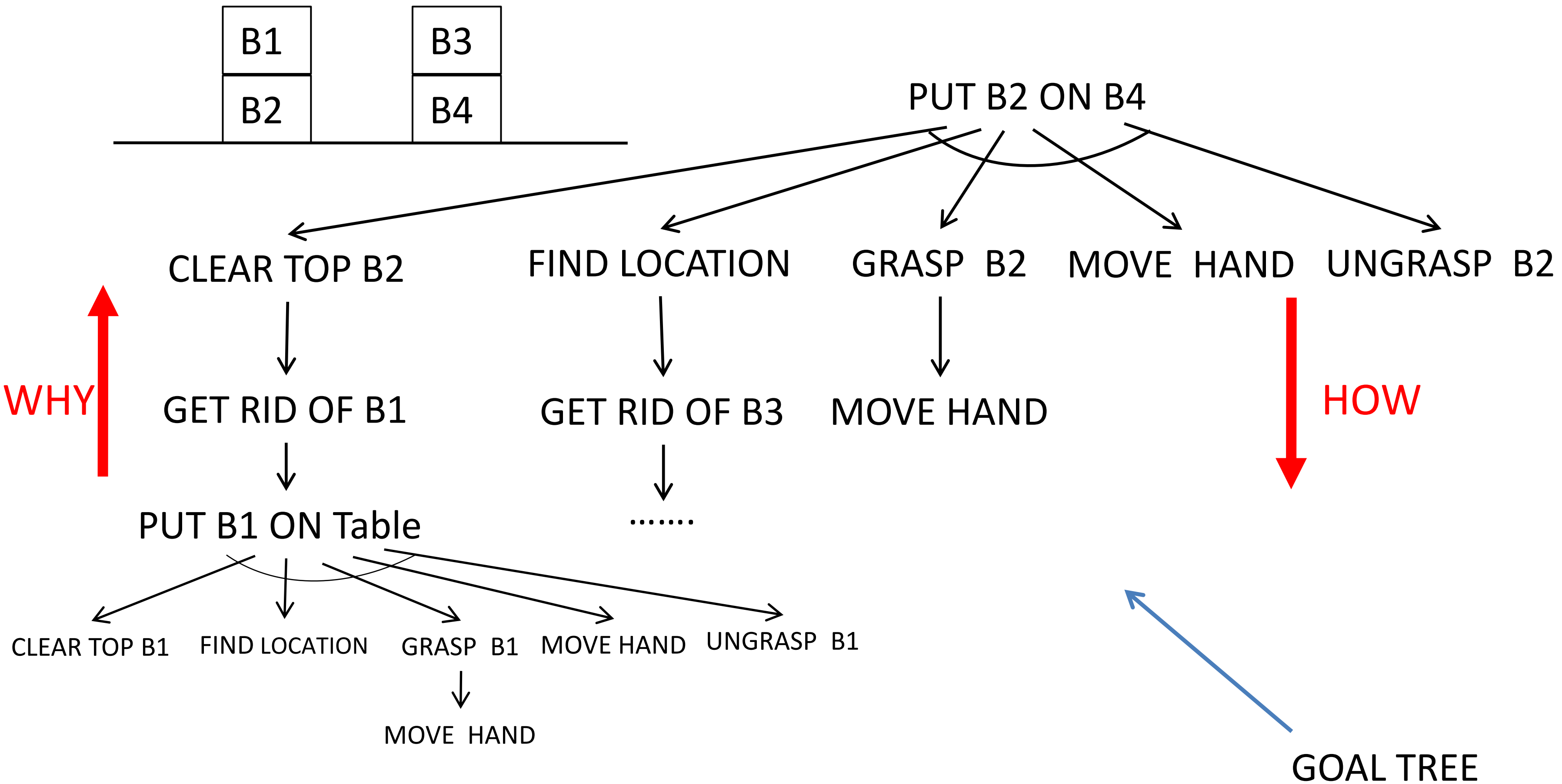
Blocks world



- Demo
- Written by Terry Winograd – professor at Stanford



Example



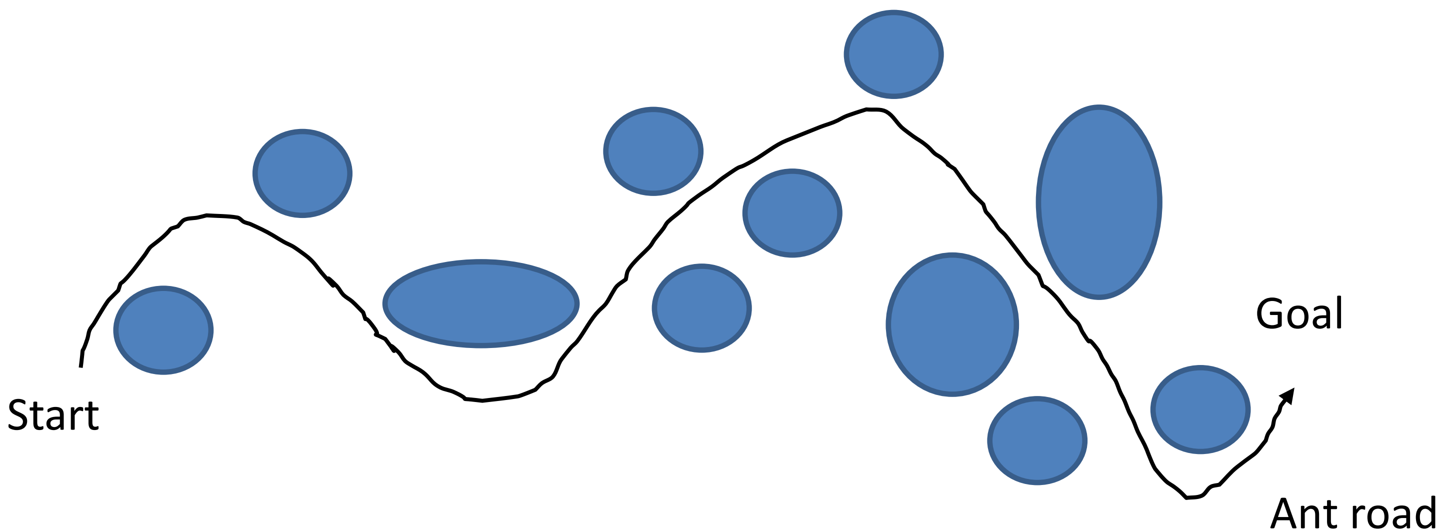
B1
B2
B3
B4
PUT B2 ON B4
CLEAR TOP B2
FIND LOCATION
GRASP B2
MOVE HAND
UNGRASP B2
WHY
GET RID OF B1
GET RID OF B3
MOVE HAND
HOW
PUT B1 ON Table
....
CLEAR TOP B1
FIND LOCATION
GRASP B1
MOVE HAND
UNGRASP B1
MOVE HAND
GOAL TREE

How and Why questions

- Deal with “*how*” questions:
  - Identify the goal involved in the goal tree
  - If the goal is an AND node report all immediate subgoals
  - If the goal is an OR node report the immediate subgoal that was achieved
- Deal with “*why*” questions:
  - Identify the goal involved in the goal tree
  - Report the immediate supergoal

Complexity

- The complexity of the behavior is largely a consequence not of the complexity of the program, but the complexity of a problem (the environment)

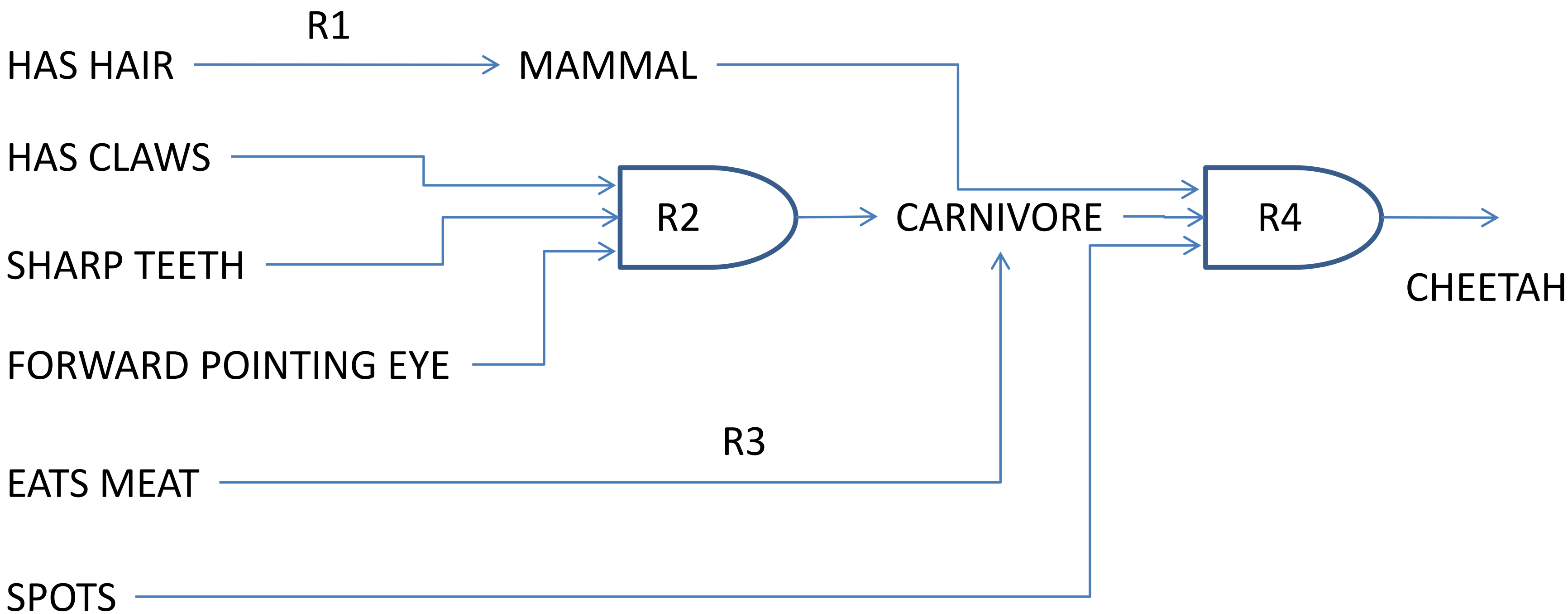


C(behavior) = max(C(program), C(environment))

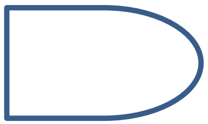
Rule-based expert systems

- Emulate the decision-making ability of a human expert
- All the knowledge in a form of simple rules: “*If .....then...*”

Example – Animals from Zoo



FORWARD CHAINING RULE-BASED EXPERT SYSTEM

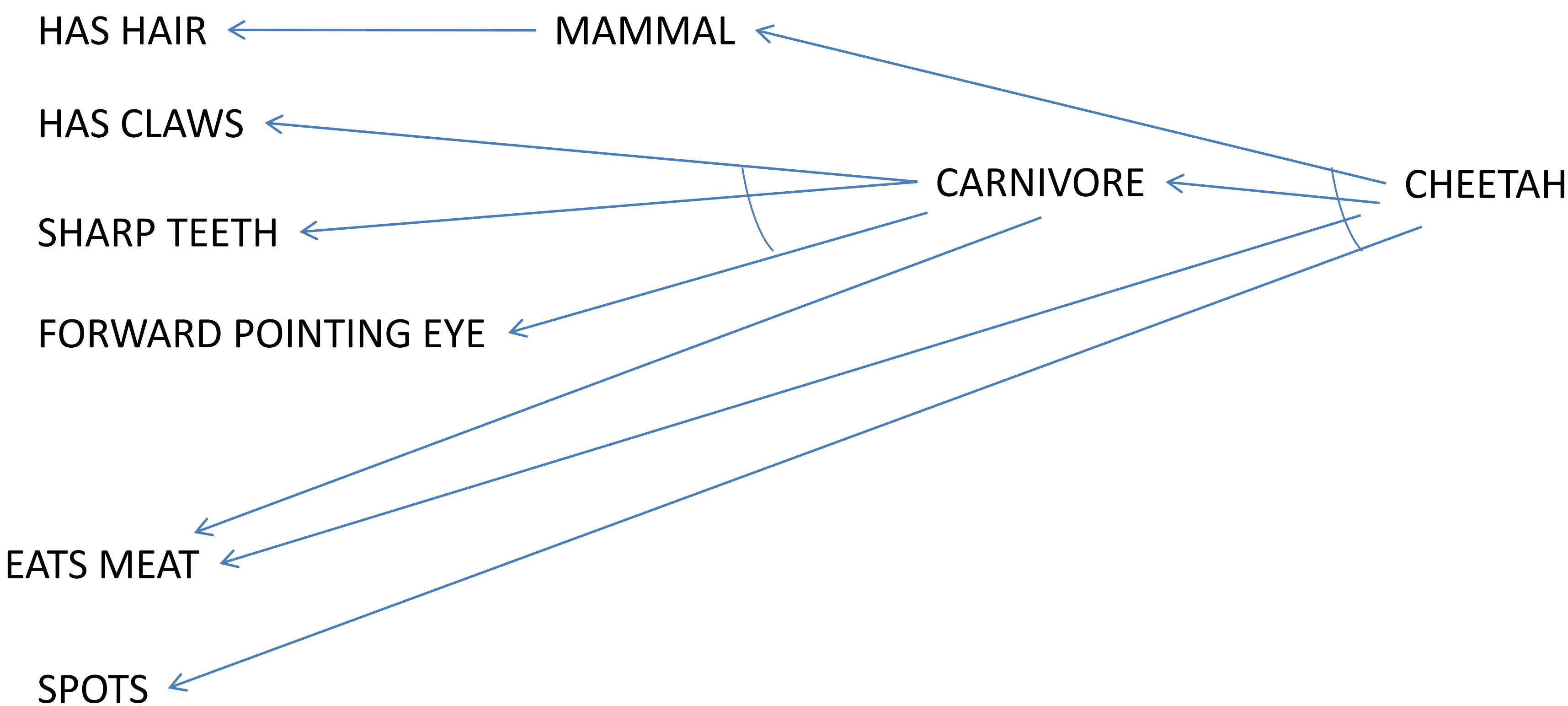


AND NODE

Rule based expert systems

- Is not an expert in the real meaning of the word because it didn’t have the experience of previous cases
- Only follows some already defined rules
- The systems can answers to the questions about the behavior
  - *Why are you interested in the animal claws?*
  - *Because I will try to see if it’s a carnivore.*

Backward chaining



BACKWARD CHAINING RULE-BASED EXPERT SYSTEM

Expert systems

- Are deduction systems – working with facts to produce new facts
- One of the first expert system – MYCIN
  - Rule example: **If** x's type is primary bacteremia and x's suspected portal is gastrointestinal and the site of the culture of x is sterile **Then** there is evidence that x is bacteroides

Principles of knowledge engineering

- Deal with specific cases – learn knowledge they otherwise they missed
- Ask questions about the things that appear to be the same but actually they are handled differently -> new words in my domain
- Build a system and see when it crack -> missing rules

Questions

- Is an expert system real smart ?
- He doesn’t know about the knowledge involves into an expert system
- Rule based systems doesn’t have anything to do with common sense

Sample exam problem

- Due to constant pressure from the AIF staff, J. K. Rowling decides to write an 8th Harry Potter book. But, she's suffering from a bad case of writer's block and decides to use a rule-based system to help her with the plot for Harry Potter and the Deadhorse Principle. She's given you a set of rules and assertions and would like your help with developing key plot points.

Rules:

P0:	IF (AND('(?x) is ambitious', '(?x) is a squib') THEN '(?x) has a bad term')
P1:	IF ('(?x) lives in Gryffindor Tower') THEN ('(?x) is a protagonist')
P2:	IF (('(?x) lives in Slytherin dungeon') THEN ('(?x) is a villain'), '(?x) is ambitious))
P3:	IF (AND(OR('(?x) is a protagonist', '(?x) is a villain'), '(?x) is ambitious') THEN ('(?x) studies a lot'))
P4:	IF (AND('(?x) studies a lot', '(?x) is a protagonist') THEN ('(?x) becomes Hermione's friend'))
P5:	IF (AND('(?x) snogs (?y)', '(?x) lives in Gryffindor Tower', '(?y) lives in Slytherin dungeon') THEN ('(?x) has a bad term'))

- ?x, ?y – variables waiting to be bound
- after IF we have some *antecedents* that must be true in order to match the rule
- after THEN we have *consequences* that will be added into DB

Assertions:

A0:	(Millicent lives in Slytherin dungeon)
A1:	(Millicent is ambitious))
A2:	(Seamus lives in Gryffindor Tower)
A3:	(Seamus snogs Millicent)

- Check assertions before using a rule

Backward chaining

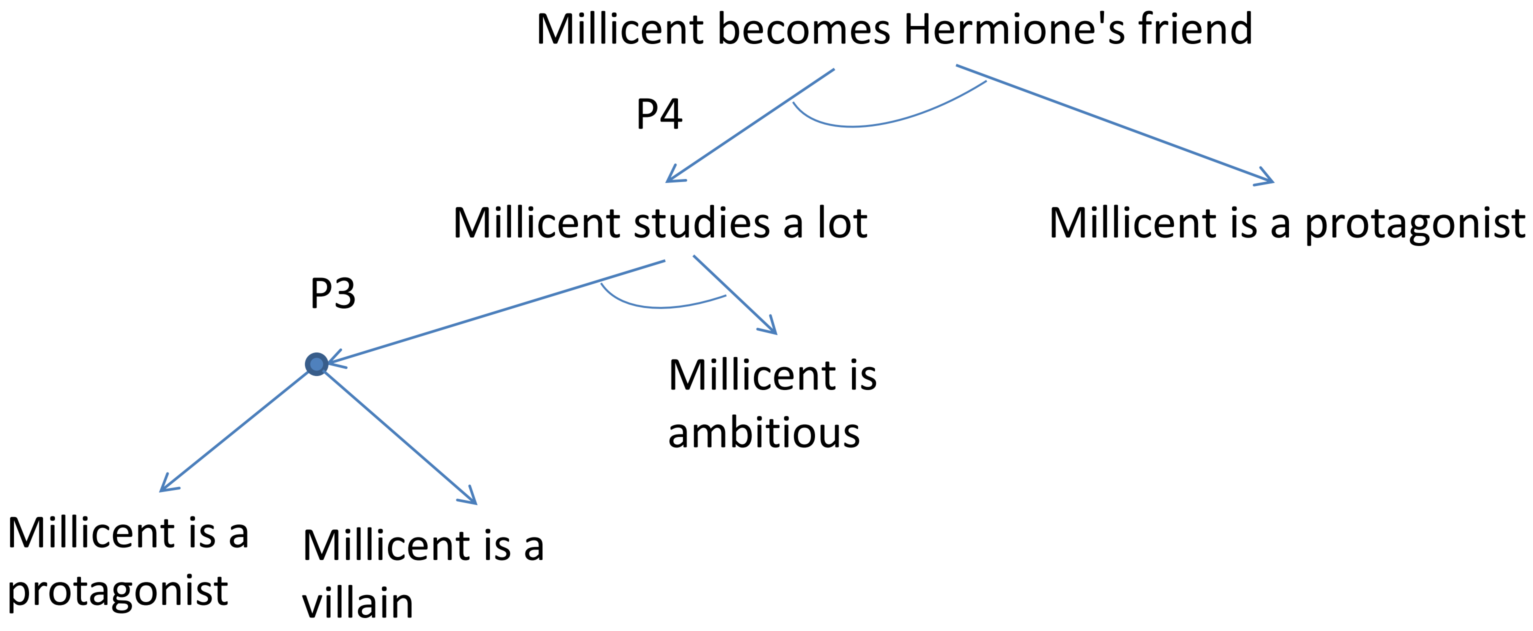
- When working on a hypothesis, the backward chainer tries to find a matching assertion in the list of assertions.
  - If we must demonstrate that “Seamus snogs Millicent” then we done. Because it’s in assertion list then we prove it.
- If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent.
  - If we must demonstrate that “Seamus is a protagonist” then based on P1 we can demonstrate that someone is a protagonist if is living in GT, and we have that Seamus lives in GT, so again we done.
- In case none are found, then the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear
- The goal tree is traverse in depth-first order

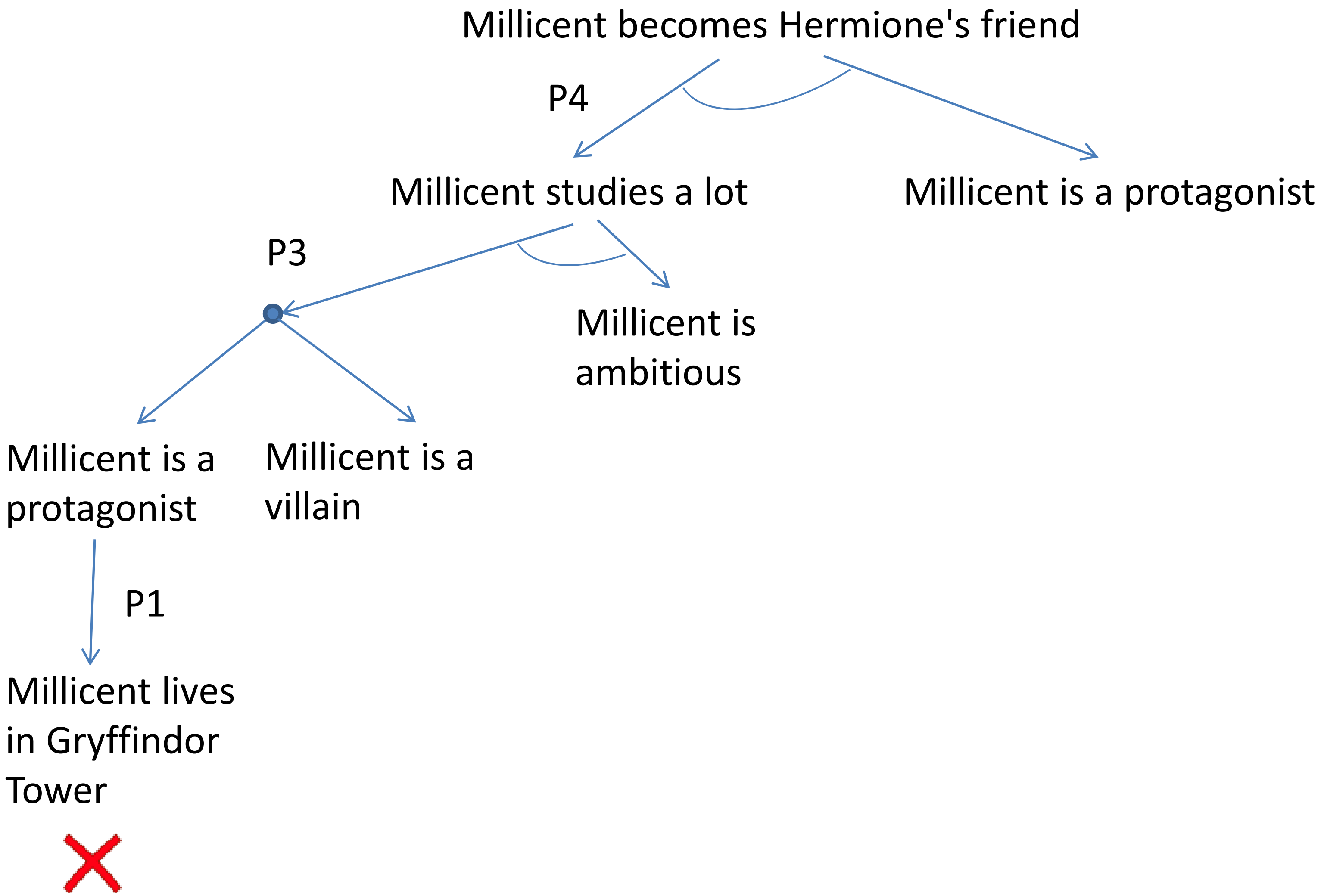
Millicent becomes Hermione's friend

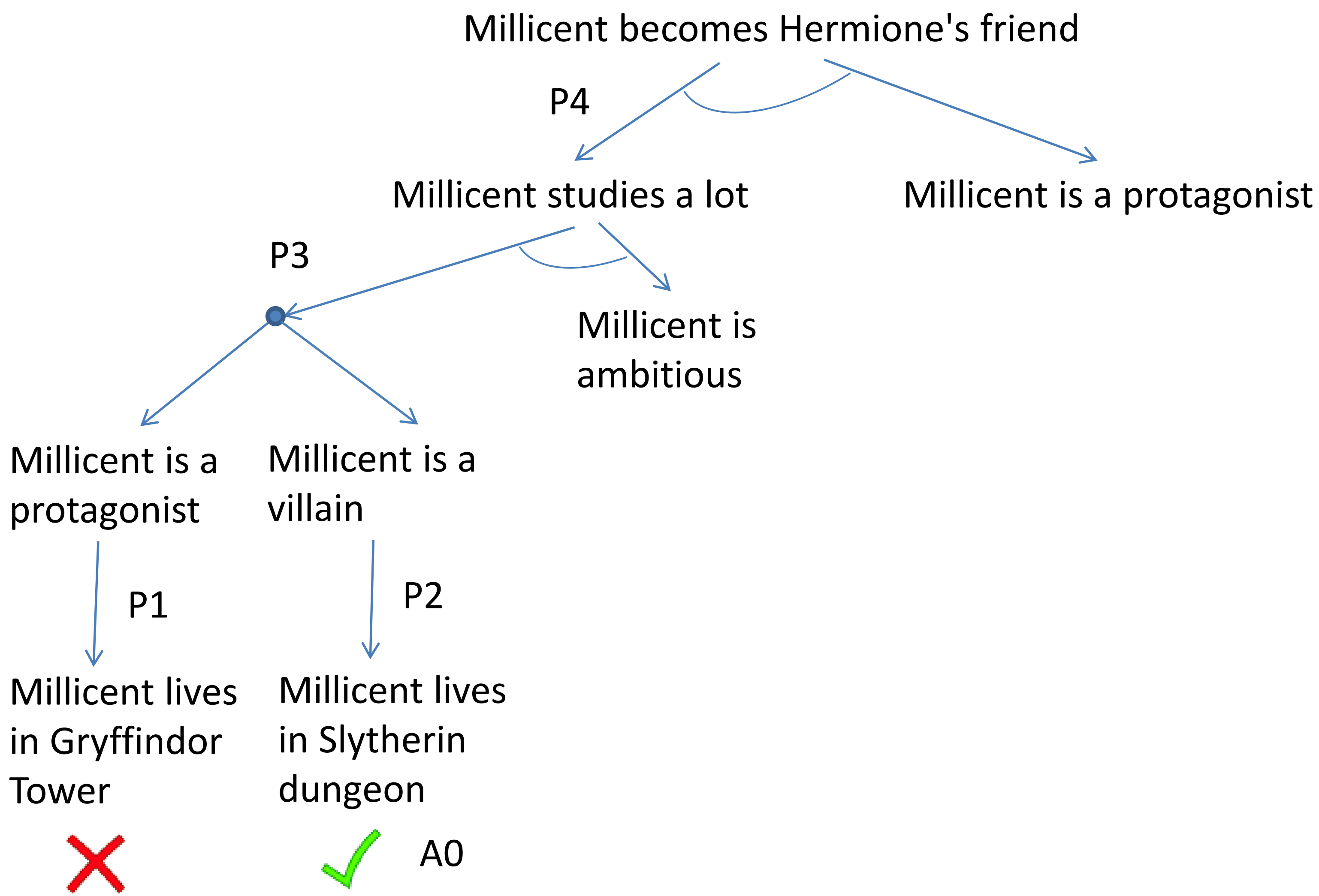
P4

Millicent studies a lot

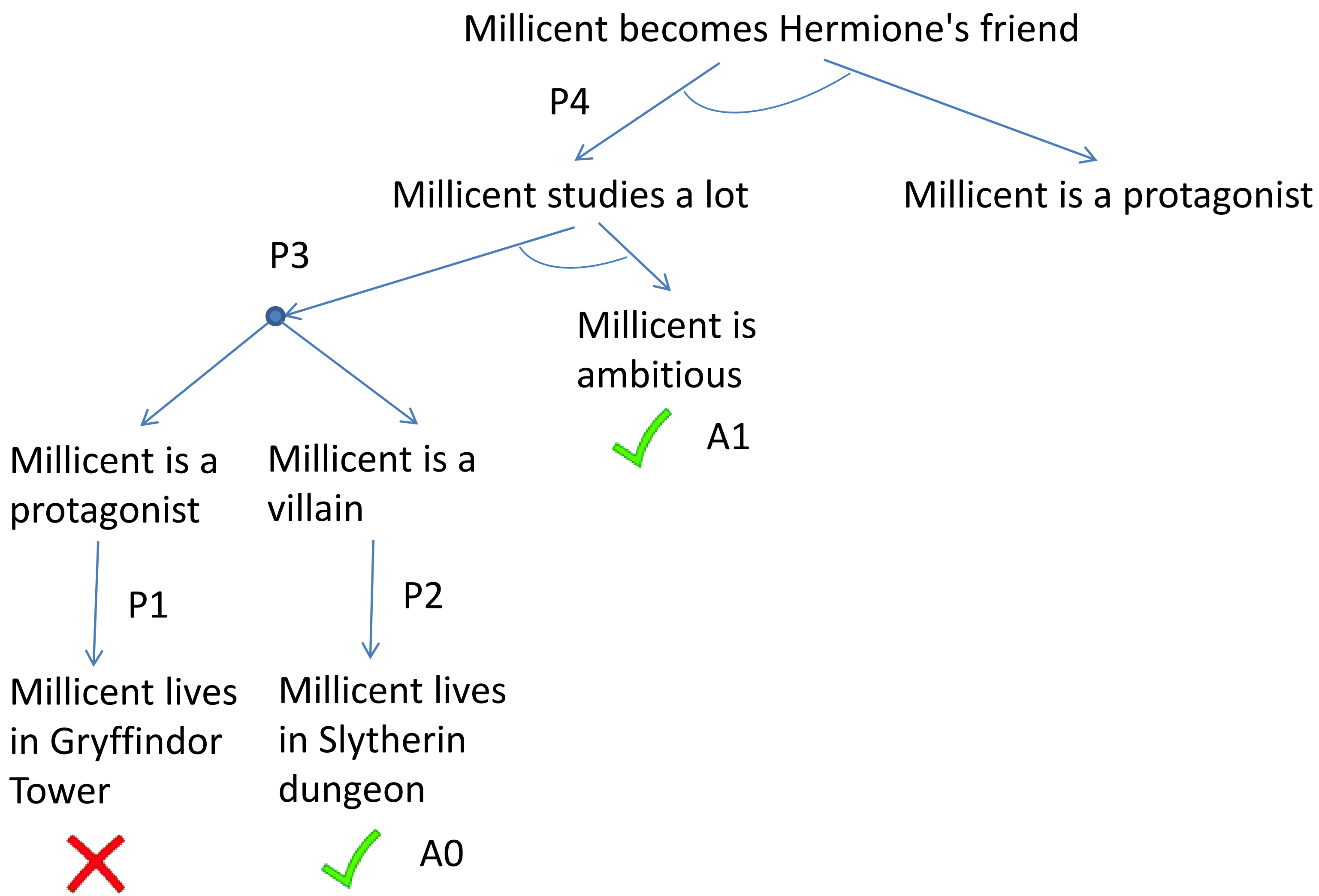
Millicent is a protagonist

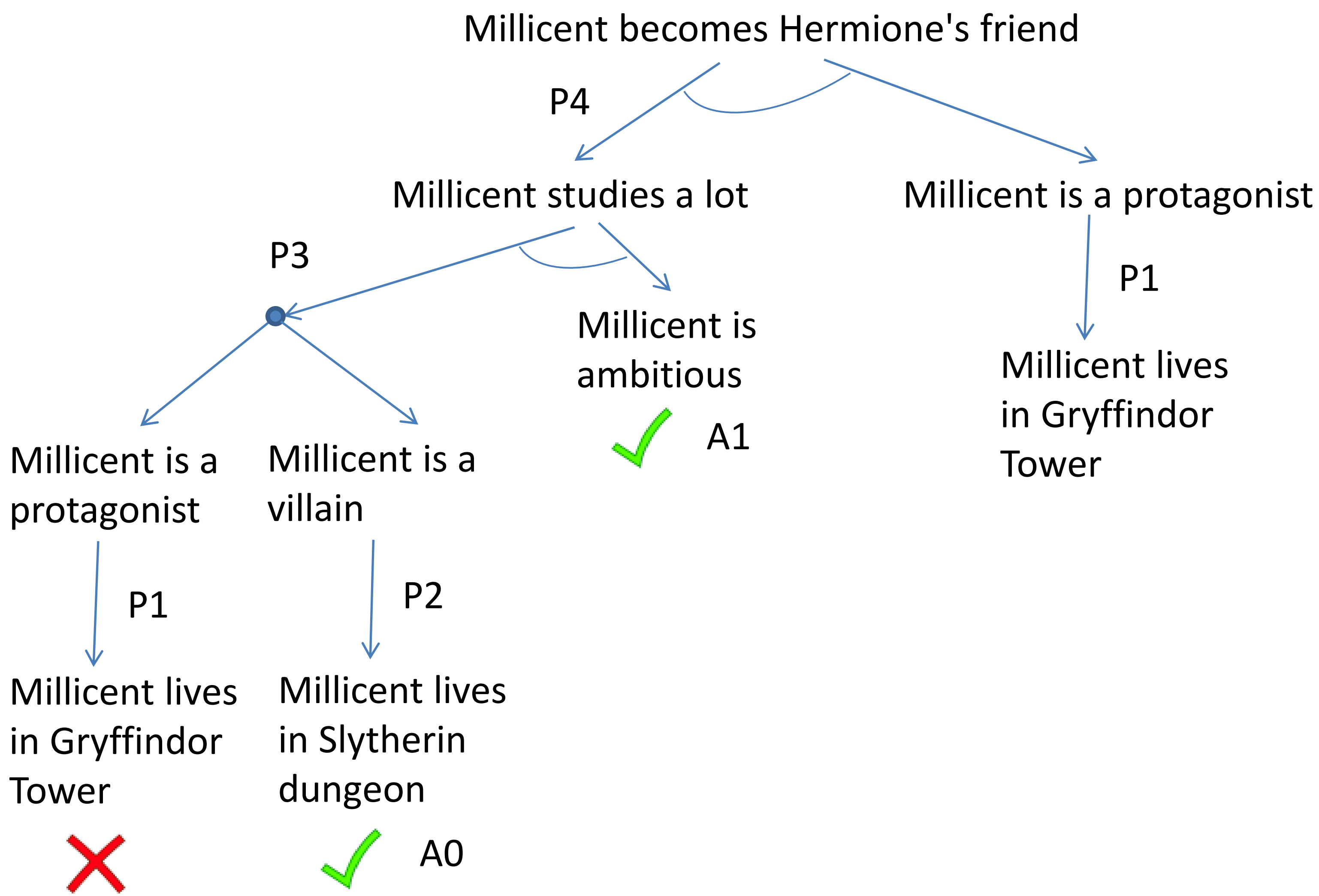

Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain

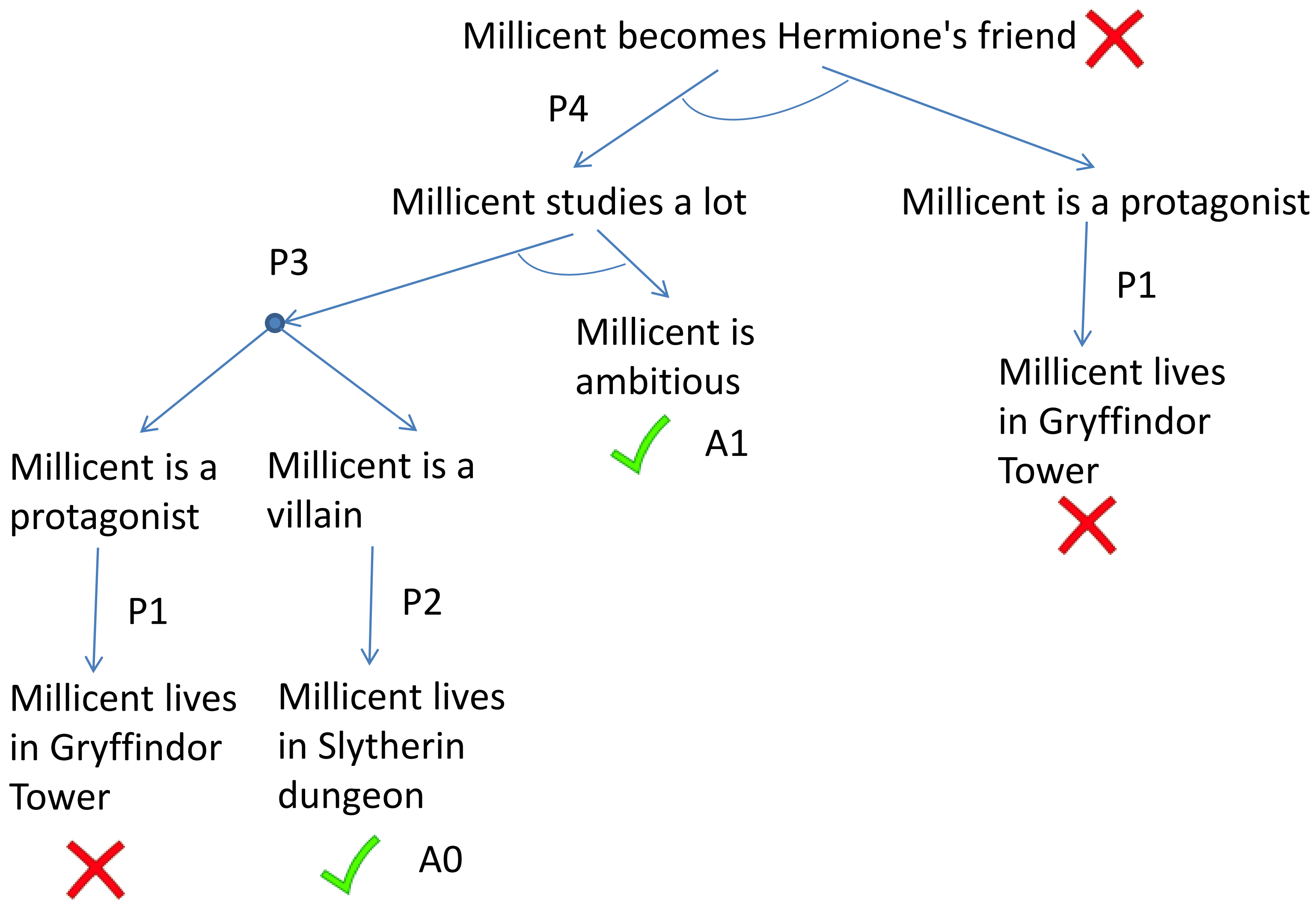

Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain
P1
Millicent lives in Gryffindor Tower



Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
A1
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
A1
P1
Millicent lives in Gryffindor Tower
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0



Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
A1
P1
Millicent lives in Gryffindor Tower
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0

A2 part

- Now, determine the minimum number of additional assertions required for Millicent to become Hermione's friend and list those assertions. Include no assertion that matches the consequent of a rule.

Millicent lives in Gryffindor Tower

- Your solution to Part A2 creates an uncommon situation. What is that uncommon situation and if J. K. considers the situation to be a problem, what should she do to the list of assertions to solve the problem?

remove the A0

Forward chaining

- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches assertions in the list of assertions in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule has an antecedent that matches both A1 and A2, the match with A1 is considered first.

- Run forward chaining on the rules and assertions. For the first two iterations, fill out the first two rows in the table below, noting the rules whose antecedents match the data, the rule that fires, and the new assertions that are added by the rule. For the remainder, supply only the fired rules and new assertions.

Matched	Fired	New Assertions Added to List of Assertions
P1,P2,P5	P1	Seamus is a protagonist
P1,P2,P5	P2	Millicent is a villain
P1,P2,P3,P5	P3	Millicent studies a lot
P1,P2,P3,P5	P5	Seamus has a bad term

Related resources

- http://aitopics.org/sites/default/files/classic/Feigenbaum_Feldman/Computers_And_Thought-Part_1_SAINT.pdf
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/exams/MIT6_034F10_quiz1_2009.pdf

Readings

- Artificial Intelligence (3rd Edition), Patrick Winston, pp. 53-60