Part Five:
Defeasible Reasoning

- Deductive reasoning guarantees the truth of the conclusion given the truth of the premises.
- Defeasible reasoning makes it reasonable to accept the conclusion, but does not provide an irrevocable guarantee of its truth.
  - conclusions supported defeasibly might have to be withdrawn later in the face of new information.
- All sophisticated epistemic cognizers must reason defeasibly:
  - perception is not always accurate
  - inductive reasoning must be defeasible
  - sophisticated cognizers must reason defeasibly about time, projecting conclusions drawn at one time forward to future times.
  - it will be argued below that certain aspects of planning must be done defeasibly

Defeasible Reasoning

- Defeasible reasoning is performed using defeasible reason-schemas.
- What makes a reason-schema defeasible is that it can be defeated by having defeaters.
- Two kinds of defeaters
  - Rebutting defeaters attack the conclusion of the inference. It is a reason for the negation of the conclusion. (¬ and ~)
  - Undercutting defeaters attack the connection between the premise and the conclusion.
    - An undercutting defeater for an inference from P to Q is a reason for believing it false that P would not be true unless Q were true. This is symbolized $(P \otimes Q)$.
    - More simply, $(P \otimes Q)$ can be read “P does not guarantee Q”.
  - Example: something’s looking red gives us a defeasible reason for thinking it is red.
    - A reason for thinking it isn’t red is a rebutting defeater.
    - It’s being illuminated by red lights provides an undercutting defeater.

Defeasible Reasoning

- Reasoning defeasibly has two parts
  - constructing arguments for conclusions
  - evaluating defeat statuses, and computing degrees of justification, given the set of arguments constructed
    - » OSCAR does this by using a defeat-status computation described in *Cognitive Carpentry*, and discussed shortly.
    - » *Justified beliefs* are those undefeated given the current stage of argument construction.
    - » *Warranted conclusions* are those that are undefeated relative to the set of all possible arguments that can be constructed given the current inputs.

Inference Graphs

Arguments are normally taken to be sequences of conclusions. But some of the ordering may be inessential.

1. (P&Q)
2. P   from 1
3. Q   from 1
4. (Q&P)   from 2 and 3.

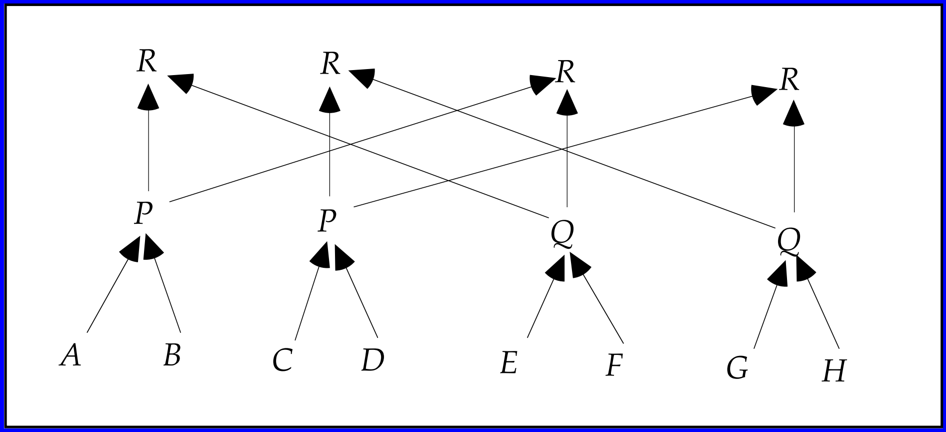
1. (P&Q)
2. Q   from 1
3. P   from 1
4. (Q&P)   from 2 and 3.

We can represent the structure more perspicuously as an ***inference graph***:

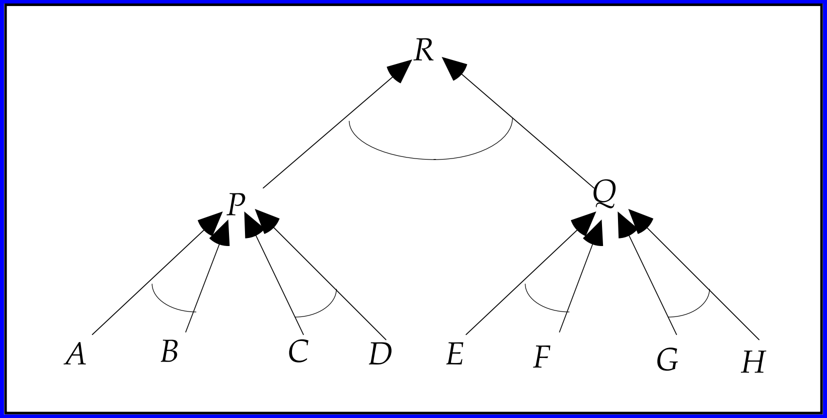
$$\begin{tikzcd}
 & (P\&Q) \arrow[dl] \arrow[dr] & \\
P \arrow[dr] & & Q \arrow[dl] \\
 & (Q\&P) &
\end{tikzcd}$$

Two Kinds of Inference Graphs

- ***Simple inference-graphs*** record multiple arguments for a single conclusion with different nodes. Nodes represent arguments.



- ***And/or inference-graphs*** record multiple arguments for a single conclusion using a single node but linked arcs. Nodes represent conclusions.



Inference Graphs

- When a reasoner reasons, it is natural to regard it as producing a number of different arguments aimed at supporting different conclusions.
- However, we can combine all of the reasoning into a single inference graph that records the overall state of the reasoner's inferences, showing precisely what inferences have been made and how inferences are based upon one another.
- This comprehensive inference graph will provide the central data structure used in evaluating a reasoner's beliefs.
- Accordingly, we can think of the function of reasoning to be that of building the inference graph.

Justification and Warrant

- Let $\mathcal{G}_n$ be the inference graph produced from a fixed input after n steps of reasoning. A conclusion is *justified at stage n* iff it is supported by an undefeated argument in $\mathcal{G}_n$
- Let $\mathcal{G}_\omega$ be the inference graph consisting of all possible arguments constructed from the fixed input. A conclusion is *warranted* iff it is supported by an undefeated argument in $\mathcal{G}_\omega$.
- The warranted conclusions are in a sense the target at which the reasoner is aiming. Ideally, a reasoner would like to draw all and only warranted conclusions.

Justification and Warrant

- Unfortunately, this ideal is impossible. It was first observed by both Israel and Reiter (in 1980) that on almost any conception of defeasible or nonmonotonic reasoning, the set of warranted conclusions is not recursively enumerable.
  - A necessary condition for a defeasibly supported conclusion to be warranted is that its negation not be a theorem of logic.
  - Thus if we are reasoning in a rich enough formalism that logical consistency is undecidable, e.g., in first-order logic, then there can be no effective procedure for ensuring that a conclusion is warranted, and hence it is impossible to build a system that generates all and only warranted conclusions.
  - In other words, the set of warranted conclusions is not recursively enumerable.
  - Familiar automated theorem provers for formal logic generate all and only valid conclusions. This is possible only because the set of valid conclusions for first-order logic is recursively enumerable. This means that an automated defeasible reasoner cannot look like an automated theorem prover.

Automated Defeasible Reasoning

- This has the consequence that it is impossible to build an automated defeasible reasoner that produces all and only warranted conclusions.
  - The most we can require is that the reasoner systematically modify its belief set so that it comes to approximate the set of warranted conclusions more and more closely.
  - The rules for reasoning should be such that:

    (1) if a proposition *P* is warranted then the reasoner will eventually reach a stage where *P* is justified and stays justified;

    (2) if a proposition *P* is unwarranted then the reasoner will eventually reach a stage where *P* is unjustified and stays unjustified.
  - This is possible if the reason-schemas are "well behaved". Then the set of warranted conclusions is Δ_2 in the arithmetic hierarchy.

Two Concepts of Defeasibility

- Human reasoning is *synchronically defeasible* in the sense that a conclusion can be warranted relative to one set of perceptual inputs, and unwarranted relative to a larger set of inputs.

- Human reasoning is also *diachronically defeasible* in the sense that we form beliefs provisionally on the basis of our current reasoning, but we may retract them later just as a result of further reasoning, without any new input.

- It appears that this must be equally true for any sophisticated cognizer.

The Structure of a Defeasible Reasoner

- A defeasible reasoner must build the inference-graph, and then compute which are arguments in it are defeated.
- I assume that the process of constructing arguments is essentially the same as in the deductive case, except that the reasoner employs defeasible reasons as well as deductive ones.
  - Whenever the reasoner makes a defeasible inference, it must adopt interest in constructing arguments for defeaters for that inference.
- In general, we must take account of the fact that we can be more justified in believing some conclusions that others.
  - This effects defeat status, because given an argument for P and an argument for ~P, the "stronger" argument wins.

Uniform Reasons

- But let us begin by assuming that all reasons are equally good, so that we can ignore reason-strengths.

I assume that one argument defeats a second only by supporting either a rebutting defeater or an undercutting defeater:

A node s *rebuts* a node h iff:

(1) h is a pf-node (i.e., a node encoding a defeasible inference) supporting some proposition q relative to a supposition Y; and

(2) s supports ¬q relative to a supposition X, where $X \subseteq Y$.

A node s *undercuts* a node h iff:

(1) h is a pf-node supporting some proposition q relative to a supposition Y; where $p_1,\ldots,p_k$ are the propositions supported by its immediate ancestors; and

(2) s supports $((p_1 \& \ldots \& p_k) \otimes q)$ relative to a supposition X where $X \subseteq Y$.

A node s *defeats* a node h iff s either rebuts or undercuts h.

Computing Defeat Status

A *partial-status-assignment* for an inference-graph G is an assignment of “defeated” and “undefeated” to a subset of the arguments in G such that for each argument A in G:

1. if a defeating argument for an inference in A is assigned “undefeated”, A is assigned “defeated”;
2. if all defeating arguments for inferences in A are assigned “defeated”, A is assigned “undefeated”.

A *status-assignment* for an inference-graph G is a maximal partial-status-assignment, i.e., a partial-status-assignment not properly contained in any other partial-status-assignment.

An argument A is *undefeated* relative to an inference-graph G of which it is a member if and only if every status-assignment for G assigns “undefeated” to A.

A belief is *justified* if and only if it is supported by an argument that is undefeated relative to the inference-graph that represents the agent’s current epistemological state.

(For comparison with other approaches, see Henry Prakken and Gerard Vreeswijk, “Logics for Defeasible Argumentation”, to appear in *Handbook of Philosophical Logic*, 2nd Edition, ed. D. Gabbay.)

Computing Defeat Status

- The justification for this computation is that it gives the intuitively right result in complex examples. To confirm this, we must look at the examples.

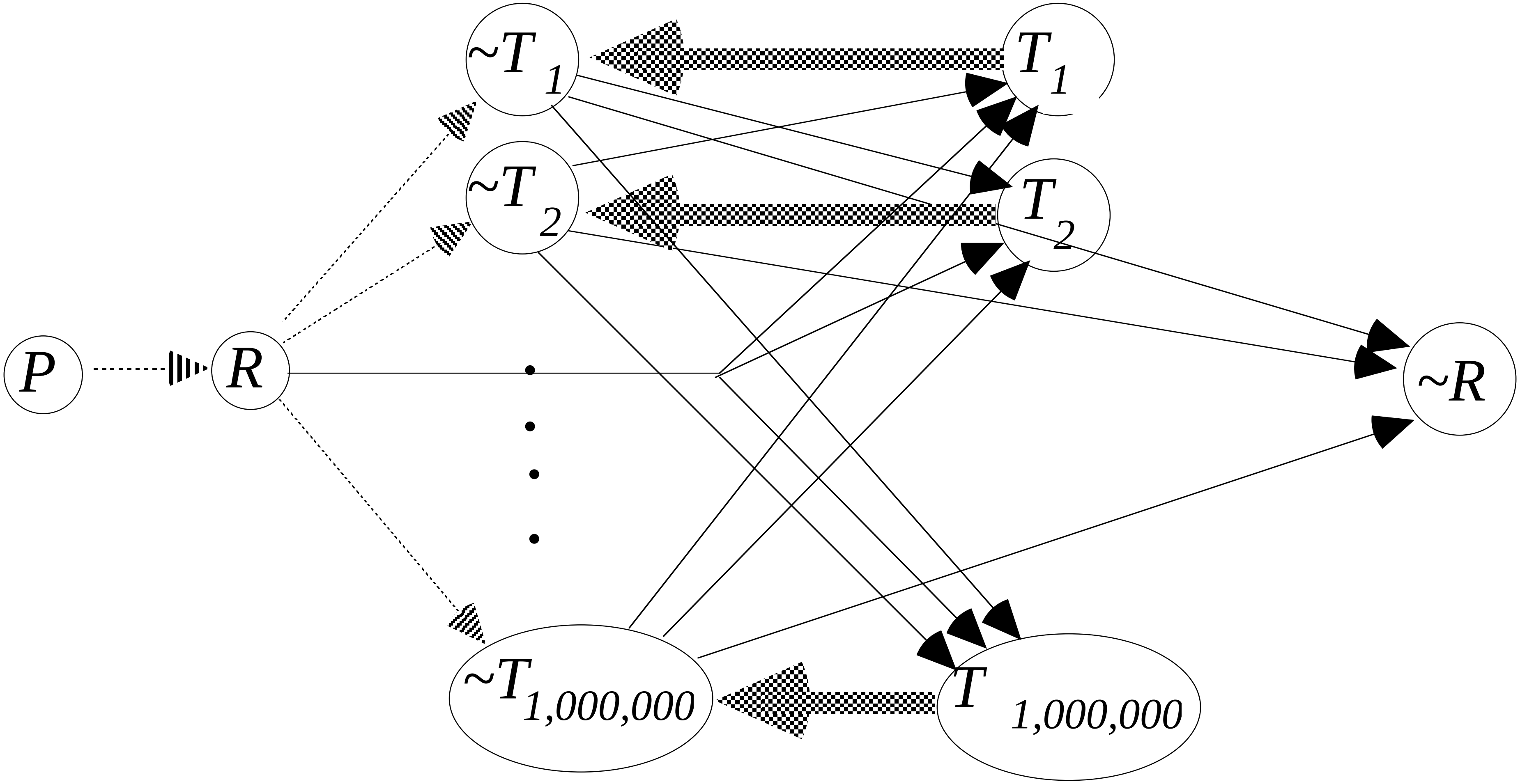


Figure 1. The Lottery Paradox.

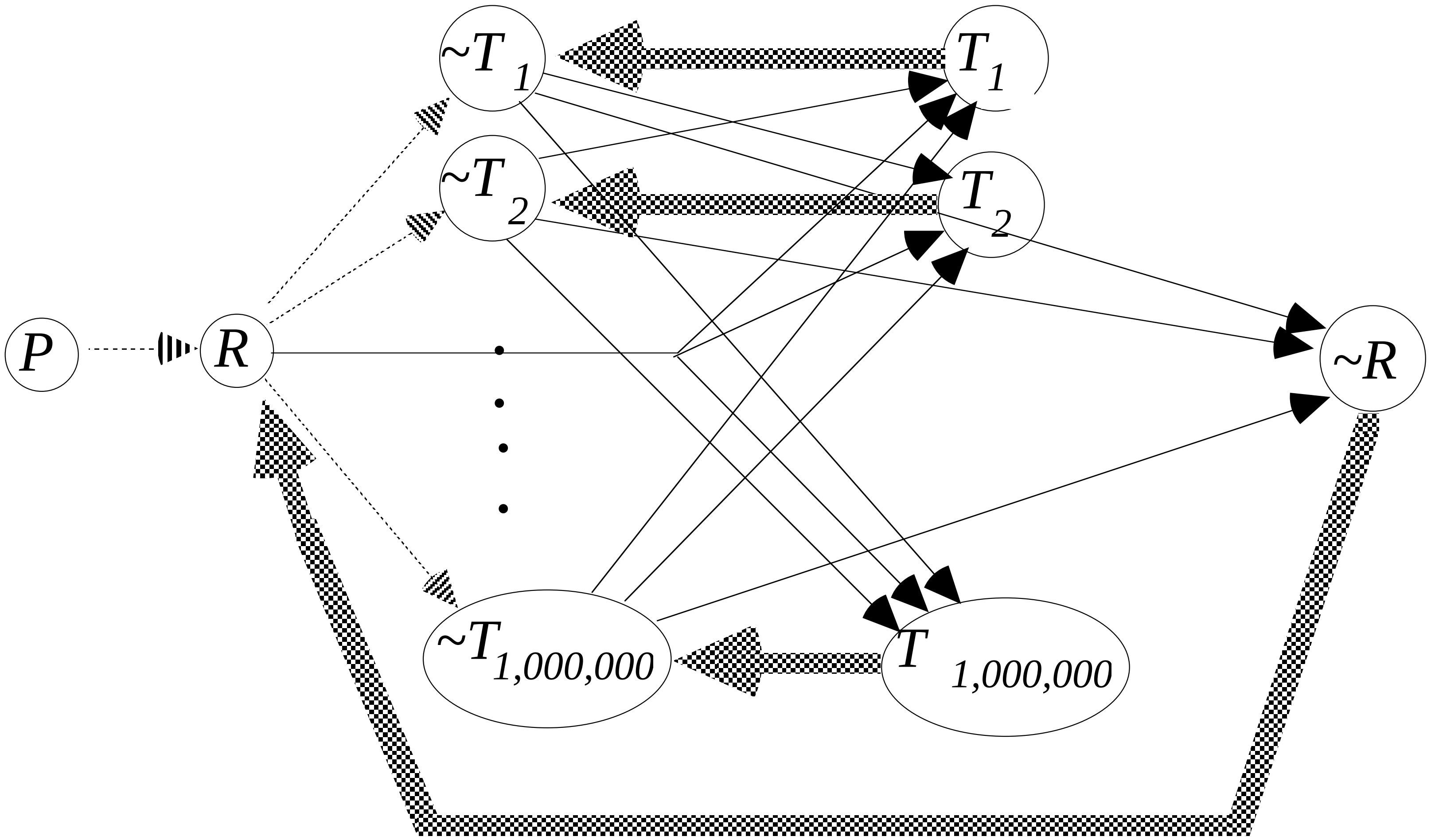


Figure 2. The Lottery Paradox Paradox.

Circumscription gets this example wrong. In circumscribing abnormality, all we can conclude is that one of the defeasible inferences is blocked by abnormality, but it could be the inference to R, so circumscription does not allow us to infer R.

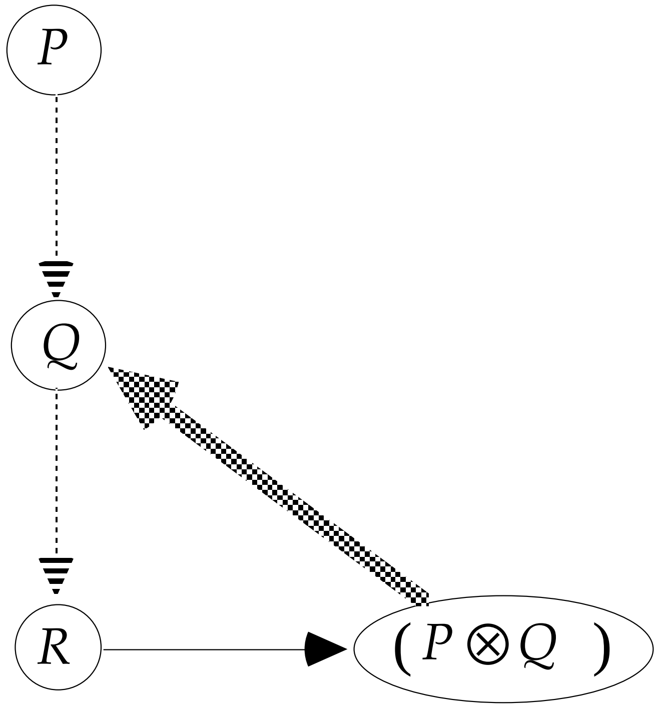


Figure 9. No nearest defeasible ancestor is defeated

Default logic gets this example wrong. There are no extensions, and hence either nothing is justified (including the given premise P) or everything is justified, depending upon how we define justification in such a case.
But does OSCAR get it right?

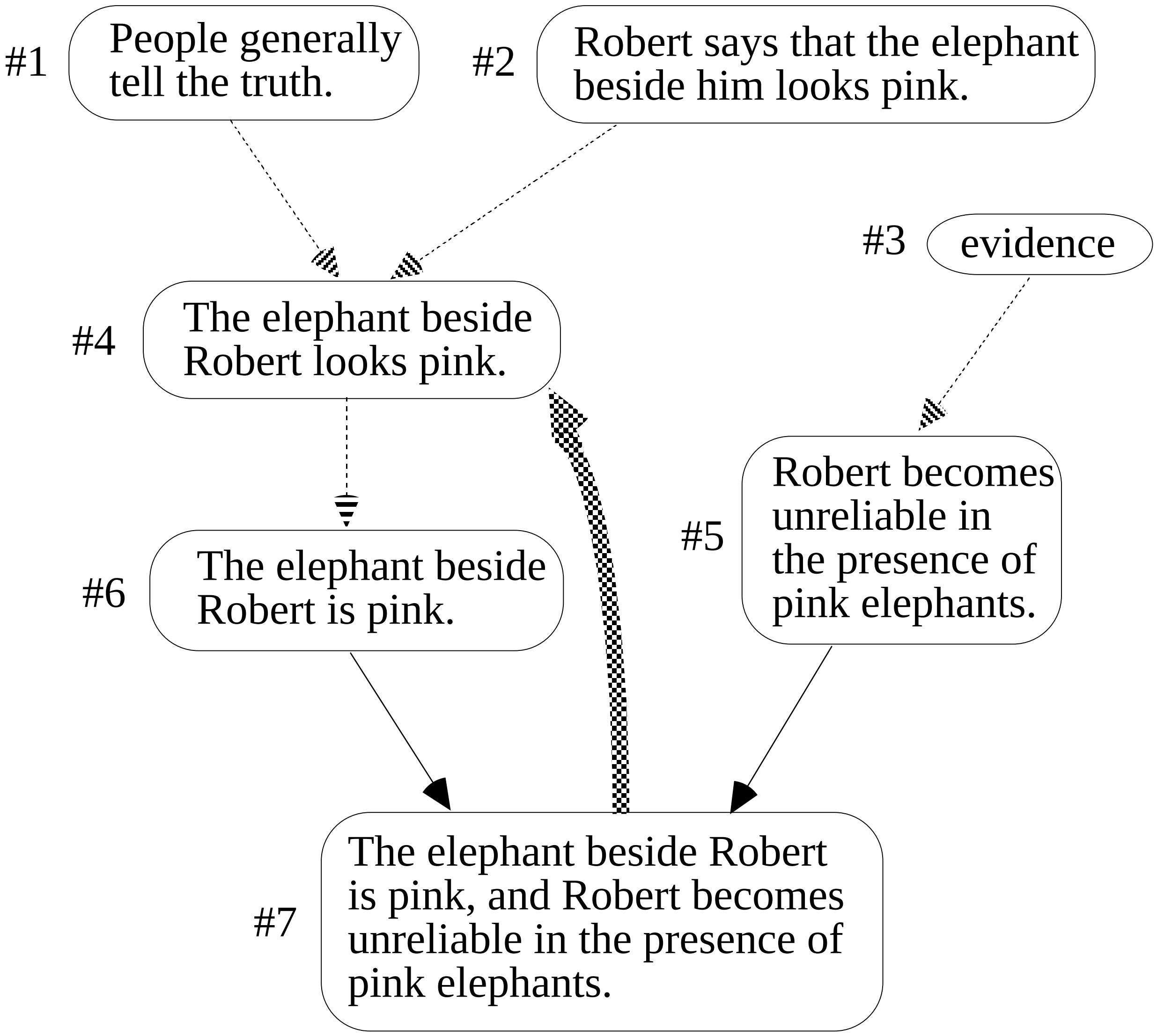

#1
People generally tell the truth.
#2
Robert says that the elephant beside him looks pink.
#3
evidence
#4
The elephant beside Robert looks pink.
#5
Robert becomes unreliable in the presence of pink elephants.
#6
The elephant beside Robert is pink.
#7
The elephant beside Robert is pink, and Robert becomes unreliable in the presence of pink elephants.

Figure 10. Argument with a self-defeating conclusion

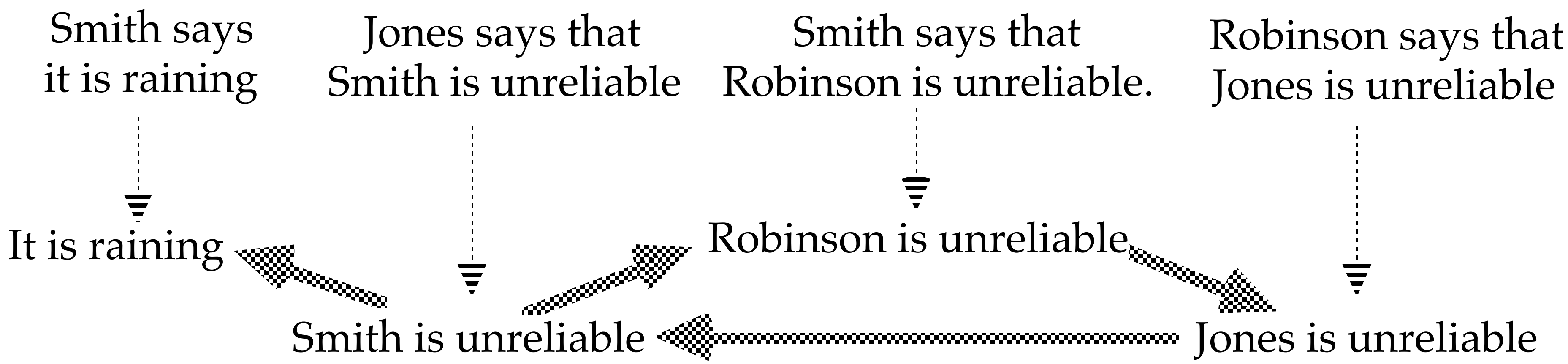


Figure 11. A three-membered defeat cycle.

Taking Strength Seriously

- We need:
  - a way of measuring the strength of a reason
  - a way of computing the strength of an argument in terms of the strengths of the reasons employed in it.
  - a way of computing defeat status that takes account of the relative strengths of the arguments.

Measuring Strength

One way is to compare reasons with a set of standard equally good reasons that have numerical values associated with them in some determinant way. I propose to do that by taking the set of standard reasons to consist of instances of the statistical syllogism.

The Statistical Syllogism:

If $r > 0.5$ then "prob(F/G) $\geq$ r & Gc" is a prima facie reason for Fc, the strength of the reason being a monotonic increasing function of r.

Consequently, for any proposition p, we can construct a standardized argument for ¬p on the basis of the pair of suppositions "prob(F/G) $\geq$ r & Gc" and "(p $\leftrightarrow$ ~Fc)":

1. Suppose prob(F/G) $\geq$ r & Gc.
2. Suppose (p $\leftrightarrow$ ~Fc).
3. Fc — from 1.
4. ¬p — from 2,3.

Measuring Strength

If X is a prima facie reason for p, the strength of this reason is $2\cdot(r - 0.5)$ where r is that real number such that an argument for $\neg p$ based upon the suppositions “$\mathrm{prob}(F/G) \geq r \;\&\; Gc$” and “$(p \leftrightarrow {\sim}Fc)$” and employing the statistical syllogism exactly counteracts the argument for p based upon the supposition X.

The measure $2\cdot(r - 0.5)$ has the convenient consequence that the strength of an instance of statistical syllogism in which $r = 0.5$ is 0, and strengths are normalized to 1.0.

Degrees of Support

- **Distinguish the *degree of support* an argument provides for its conclusion from the *degree of justification*. The latter depends not just on the argument but also on what defeating arguments there are.**
- **Before addressing the computation of degrees of justification, let us focus on degrees of support.**

- **It is often supposed that degrees of support work like probabilities, and a conclusion is well supported by an argument iff it is made sufficiently probable by the reasoning.**
- **This is *Generic Bayesianism*.**

Generic Bayesianism

- The simplest objection to Generic Bayesianism is the one already mentioned — arguments for necessary truths are automatically perfectly good, even if they are intuitively terrible. E.g.,

Given: $\{[P \leftrightarrow (Q \mathbin{\&} {\sim}P)] \rightarrow {\sim}Q\} \rightarrow 2+2 = 4$

Given: $2+2 = 4$

Therefore $[P \leftrightarrow (Q \mathbin{\&} {\sim}P)] \rightarrow {\sim}Q$.

Generic Bayesianism

- However, I will focus on another argument against Generic Bayesianism that generalizes to a large class of theories. I will argue that Generic Bayesianism is "self-defeating".

Probabilistic Validity

- According to generic Bayesianism, our epistemic attitude towards a proposition should be determined by its probability.
- It will generally be necessary to compute such probabilities in order to determine the degree of justification of a belief.
- Some kinds of deductive inference can be applied “blindly” without going through such calculations, but only when the inferences are guaranteed to preserve probability.

Probabilistic Validity

- Let us say that an inference rule

$$\frac{P_1,\ldots,P_n}{Q}$$

  is ***probabilistically valid*** just in case it follows from the probability calculus that prob(Q) ≥ the minimum of the prob(P_i)'s.

- For the generic Bayesian, inference rules can be applied blindly, obviating the need for probability calculations, only if they are probabilistically valid.

Probabilistic Validity

- If P logically entails Q, then it follows from the probability calculus that prob(Q) ≥ prob(P), and hence the generic Bayesian is able to conclude that the degree of justification for Q is as great as that for P.
- Thus deductive inferences from single premises can proceed blindly.
- However, this is not equally true for entailments requiring multiple premises.
- Specifically, it is not true in general that if {P,Q} entails R, then prob(R) ≥ the minimum of prob(P) and prob(Q).
- For instance, {P,Q} entails (P&Q), but prob(P&Q) may be less than either prob(P) or prob(Q).
  - In other words, *adjunctivity* is not probabilistically valid. This has been noted and endorsed by a number of proponents of Generic Bayesianism.

Probabilistic Validity

- What has not been noted, although it is obvious, is that many other inference rules turn out to be probabilistically invalid.
- This includes *modus ponens*, *modus tollens*, etc.
- In fact, *any* inference rule proceeding from multiple premises and using all of the premises essentially will be probabilistically invalid.
- This is extremely counter-intuitive. It means that a reasoner engaging in Bayesian updating is precluded from drawing deductive conclusions from its reasonably held beliefs.

Example

Consider an engineer who is designing a bridge. She will combine a vast amount of information about material strength, weather conditions, maximum load, costs of various construction techniques, and so forth, to compute the size a particular girder must be. These various bits of information are, presumably, independent of one another, so if the engineer combines 100 pieces of information, each with a probability of 0.99, the conjunction of that information has a probability of only $.99^{100}$, which is approximately 0.366. According to generic Bayesianism, she would be precluded from using all of this information simultaneously in an inference—even an inference to a probability about the likelihood of the bridge surviving a certain number of years. But then it would be impossible to build bridges.

Generic Bayesianism is self-defeating

- If generic Bayesianism were correct, it would be impossible to perform the very calculations required by the theory for determining whether a belief ought to be held.
- This arises from the fact that Bayesian updating requires a reasoner to decide what to believe by computing probabilities.
- The difficulty is that the probability calculations themselves cannot be performed by a Bayesian reasoner.
- To illustrate the difficulty, suppose the reasoner has the following beliefs:

prob(P⁄Q) = prob(P)+prob(Q)–prob(P&Q)

prob(P) = 0.5

prob(Q) = 0.49

prob(P&Q) = 0.

Generic Bayesianism is self-defeating

- From this we would like the reasoner to compute that prob(P/Q) = 0.99, and perhaps go on to adopt (P/Q) as one of its beliefs.
- However, generic Bayesianism cannot accommodate this.
- The difficulty is that this is an example of a “blind use” of a deductive inference, and as such it is legitimate only if the inference is probabilistically valid.
- Although the four premises do logically entail the conclusion, the inference is not probabilistically valid for the same reason that *modus ponens*, *modus tollens*, and *adjunction* fail to be probabilistically valid.
- It is an inference from a mutliple premise set, and despite the entailment, the conclusion can be less probable than any of the premises.

Generic Bayesianism is Self-Defeating

Generic Bayesianism is incoherent as a theory of belief updating. The probability calculations required of the reasoner proceed via deductive inferences that are not probabilistically valid, and hence the Bayesian reasoner is precluded from making the very calculations it needs to determine degrees of justification.

Statistical and Epistemic Probability

- If generic Bayesianism is logically incoherent, why is this intuition so compelling?
- We must distinguish between statistical probability and epistemic probability.
  - Statistical probability is concerned with chance.
  - Epistemic probability is concerned with the degree of justification of a belief. We are referring to epistemic probability when we conclude that the butler probably did it. All that means is that there is good reason to think the butler did it.
- The lesson to be learned from the previous discussion is that rules like *modus ponens* and *adjunction* preserve high epistemic probability, and hence epistemic probability cannot be quantified in a way that conforms to the probability calculus.
- This should not be particularly surprising. There was never really any reason to expect epistemic probability to conform to the probability calculus. That is a calculus of statistical probabilities, and the only apparent connection between statistical and epistemic probability is that they share the same ambiguous name.

The Weakest Link Principle for Deductive Arguments

- In place of generic Bayesianism, I propose the ***weakest link principle for deductive arguments***:

  > The degree of support of the conclusion of a deductive argument is the minimum of the degrees of support of its premises.

- The argument for this is that the objections to the Bayesian account can be applied more generally to any account that allows the strength of an argument to be less than its weakest link.
  - On any such account, multi-premise inference rules like modus ponens and adjunction will turn out to be invalid, but then it seems unavoidable that the theory will be self-defeating in the same way as the Bayesian theory—by making it impossible for the reasoner to compute the degrees of support of its conclusions.

The Weakest Link Principle for Defeasible Arguments

- The above formulation of the weakest link principle applies only to deductive arguments, but we can use it to obtain an analogous principle for defeasible arguments. If P is a prima facie reason for Q, then we can use conditionalization to construct a simple defeasible argument for the conclusion (P → Q), and this argument turns upon no premises:
- Suppose P. Then (defeasibly) Q. Therefore, (P → Q).
- Because this argument has no premises, the degree of support of its conclusion should be a function of nothing but the strength of the prima facie reason.
- Any defeasible argument can be reformulated so that prima facie reasons are used only in subarguments of this form, and then all subsequent steps of reasoning are deductive. The conclusion of the defeasible argument is thus a deductive consequence of its premises together with a number of conditionals justified in this way. By the weakest link principle for deductive arguments, the degree of support of the conclusion should then be the minimum of (1) the degrees of justification of the premises used in the argument and (2) the strengths of the prima facie reasons.

The Weakest Link Principle for Defeasible Arguments

The degree of support of the conclusion of a defeasible argument is the minimum of the strengths of the prima facie reasons employed in it and the strengths of the premises to which it appeals.

I will refer to this as the ***strength of the argument***.

The Accrual of Reasons

- The strength of an argument is the degree of support it provides to its conclusion.
- What happens if the agent has more than one argument for the same conclusion? Does that increase the degree of support?
- I will argue that cases seeming initially to illustrate such accrual of support appear upon reflection to be better construed as cases of having a single reason that subsumes the two separate reasons.

The Accrual of Reasons

- If Brown tells me that the president of Fredonia has been assassinated, that gives me a reason for believing it; and if Smith tells me that the president of Fredonia has been assassinated, that also gives me a reason for believing it. Surely, if they both tell me the same thing, that gives me a better reason for believing it.
- There are considerations indicating that my reason in the latter case is not simply the conjunction of the two reasons I have in the former cases.
  - Reasoning based upon testimony is a straightforward instance of the statistical syllogism. We know that people tend to tell the truth, and so when someone tells us something, that gives us a defeasible reason for believing it. This turns upon the following probability being reasonably high:

    (1) prob(P is true / S asserts P).
  - Given that this probability is high, I have a defeasible reason for believing that the president of Fredonia has been assassinated if Brown tells me that the president of Fredonia has been assassinated.
  - When we have the concurring testimony of two people, our degree of justification is not somehow computed by applying a predetermined function to the latter probability. Instead, it is based upon the quite distinct probability

    (2) prob(P is true / S1 asserts P and S2 asserts P and S1 $\neq$ S2).
  - The relationship between (1) and (2) depends upon contingent facts about the linguistic community.

Failure of The Accrual of Reasons

- All examples I have considered that seem initially to illustrate the accrual of reasons turn out in the end to have this same form. They are all cases in which we can estimate probabilities analogous to (2) and make our inferences on the basis of the statistical syllogism rather than on the basis of the original reasons.
- Accordingly, I doubt that reasons do accrue. It is at least simpler to assume that they do not.
- If we have two separate undefeated arguments for a conclusion, the degree of justification for the conclusion is the maximum of the strengths of the two arguments.

Defeat Among Inferences

- The degree of justification of a conclusion is influenced both by the degree of support it receives from supporting arguments and the degrees of support for defeaters of those arguments.
- How does degree of support affect defeat?
- One of the most important roles of the strengths of reasons lies in deciding what to believe when one has conflicting arguments for q and ~q.
  - It is clear that if the argument for q is much stronger than the argument for ~q, then q should be believed.
  - But what if the argument for q is just slightly stronger than the argument for ~q? It is tempting to suppose that the argument for ~q should at least attenuate our degree of confidence in q, in effect lowering its degree of justification.
  - In other words, defeaters that are not strong enough to defeat can still act as ***diminishers***.

Diminishers

Here is an argument against diminishers.

- Suppose weak defeaters can act as diminishers.
- Then if we acquired a second argument for ~q, it would face off against a weaker argument for q and so be better able to defeat it.
- But that is tantamount to taking the two arguments for ~q to result in greater justification for that conclusion, and that is just the principle of the accrual of reasons.
- So it seems that if we are to reject the latter principle, then we should also conclude that arguments that face weaker conflicting arguments are not thereby diminished in strength.
- For now, I will assume this, but I will eventually return to this issue and endorse a somewhat more complex view.

Redefining Defeat

- **On the assumption that there are no diminishers, we can revise our definition of defeat to take account of strength.**
  - **A node of the inference graph can be assigned a strength corresponding to the strength of the argument it encodes.**

A node s *rebuts* a node h iff:

(1) h is a pf-node of some strength α supporting some proposition q relative to a supposition Y; and

(2) s supports ¬q relative to a supposition X with strength β, where $X \subseteq Y$ and $\beta \geq \alpha$.

A node s *undercuts* a node h iff:

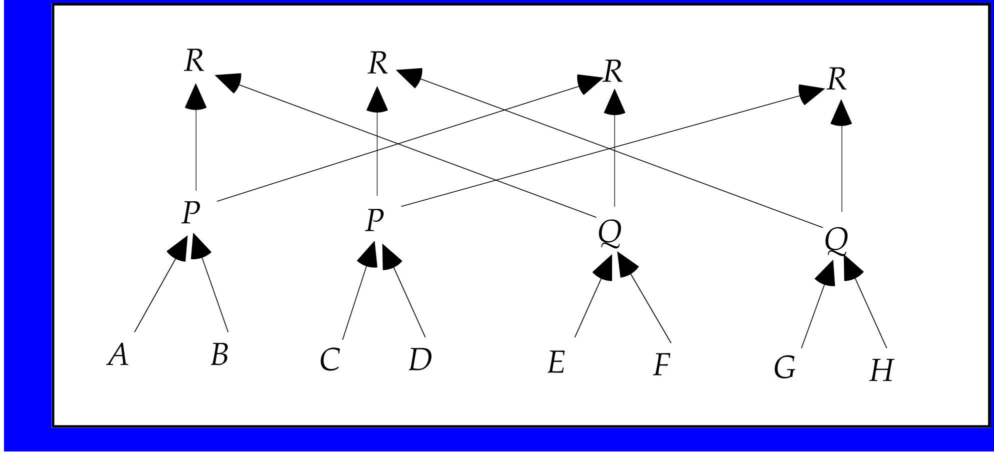
(1) h is a pf-nodeof some strength α supporting some proposition q relative to a supposition Y with strength β; where $p_1,\ldots,p_k$ are the propositions supported by its immediate ancestors; and

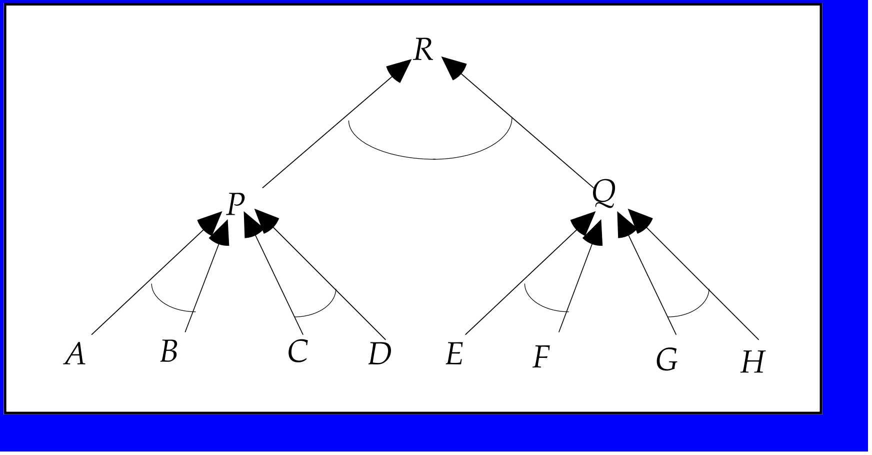
(2) s supports $((p_1 \& \ldots \& p_k) \otimes q)$ relative to a supposition X where $X \subseteq Y$ and $\beta \geq \alpha$.

Degrees of Justification

- The degree of justification of a conclusion (relative to an inference-graph) is the strength of the strongest undefeated argument supporting it.
- This is the strength of the strongest undefeated node supporting it.

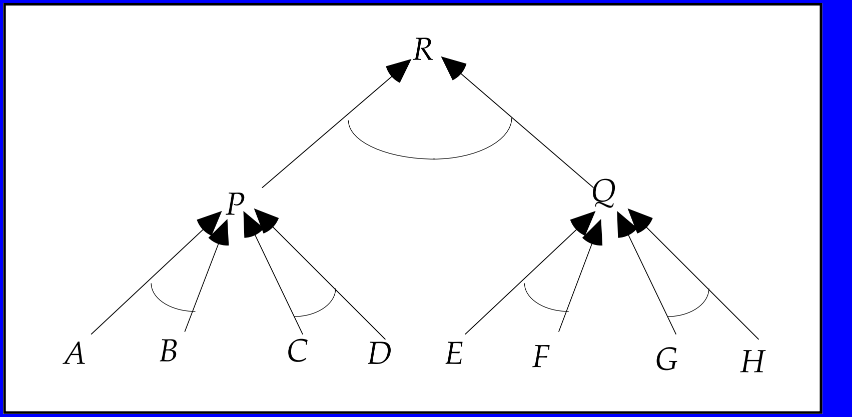
- This only works for simple inference-graphs, but these are inefficient data-structures for storing the agent's reasoning. It would be preferable to use and/or graphs.





And/or Inference-Graphs

- Nodes have support-links linking them to <u>sets</u> of nodes from which they are inferred by single inferences. Thus P has two support-links, one linking it to {A,B} and the other linking it to {C,D}.
- Take the defeat-status of a node to be its undefeated-degree-of-support rather than just "defeated" or "undefeated".
  - If all arguments supporting a node are defeated, then the undefeated-degree-of-support is 0.
  - Otherwise, it is the maximum of the strengths of the undefeated arguments.
- A node of the inference-graph is ***<u>initial</u>*** iff its list of support-links and list of node-defeaters is empty.



And/or Inference-Graphs

σ is a *partial status assignment* iff σ is a function assigning real numbers between 0 and 1 to a subset of the nodes of an inference-graph and to the support-links of those nodes in such a way that:

1. σ assigns its node-strength to any initial node;
2. If σ assigns a value α to a defeat-node for a support-link and assigns a value less than or equal to α to some member of the link-basis, then σ assigns 0 to the link;
3. Otherwise, if σ assigns values to every member of the link-basis of a link and every link-defeater for the link, σ assigns to the link the minimum of the strength of the link-rule and the numbers σ assigns to the members of the link-basis.
4. If every support-link of a node is assigned 0, the node is assigned 0;
5. If some support-link of a node is assigned a value greater than 0, the node is assigned the maximum of the values assigned to its support-links.
6. If every support-link of a node that is assigned a value is assigned 0, but some support-link of the node is not assigned a value, then the node is not assigned a value.

σ is a *status assignment* iff σ is a partial status assignment and σ is not properly contained in any other partial status assignment.

And/or Inference-Graphs

- A ***node is defeated*** iff some status-assignment assigns 0 to it.
- An ***argument*** is a set of nodes linked by support-links.
- An ***argument is undefeated*** iff every node in it is undefeated.
- The ***undefeated-degree-of-support*** of a node (and the degree of justification of its conclusion) is the maximum of the strengths of the undefeated arguments supporting it.

- This is equivalent to the definition provided earlier in terms of simple inference-graphs.

Computing Defeat Statuses

A reasonably efficient generate-and-test algorithm for computing status assignments for and/or graphs:

- First compute assignments to as many nodes and support-links as possible on the basis of the assignments to the initial nodes.
- If some support-links remain unassigned, assignments are made to them arbitrarily and tested for consistency.

An Efficiency Problem

- When this algorithm has to try arbitrary assignments to support-links whose assignments cannot be computed from already-assigned nodes and links, it will consider all possible combinations of assignments to each of the support-links that must be treated in that way.

- This has the result that if we present two copies of the same problem to the algorithm simultaneously, the number of arbitrary assignments it considers will increase exponentially rather than linearly.

An Efficiency Problem — example

Figure 10

0.07 sec

4 assignments

2 instances of #10

1.20 sec(increase by a factor of 17.14)

16 assignments

3 instances of #10

18.24 sec (x260.57)

64 assignments

4 instances of #10

310.67 sec (x4438.14)

256 assignments

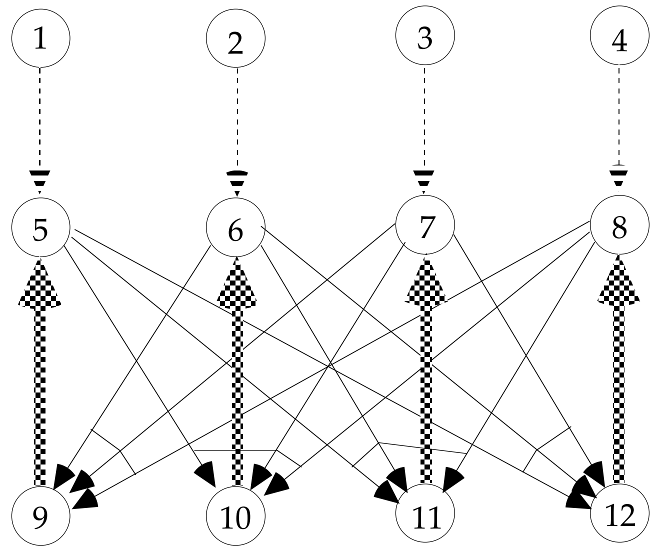


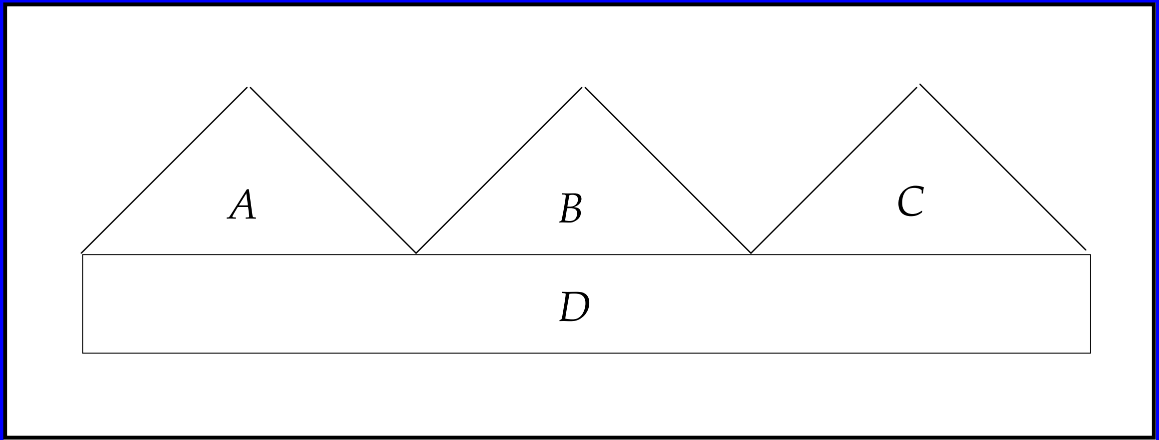
Figure 10. Four-way collective-defeat.

Resolving the Efficiency Problem

- The different parts of the composite problem are really separate subproblems, and it should be possible to separate the computations, resulting in a linear increase in computational difficulty rather than an exponential one.
  - *node* is an inference/defeat-ancestor of *link* iff either (1) *node* is a member of the support-link-basis of *link* or of a support-link that is an inference-defeat-ancestor of *link*, or (2) *node* is a defeater for *link* or for a support-link that is an inference-defeat-ancestor of *link*.
  - A support-link *L* is an inference/defeat-ancestor of *link* iff *L* is a support-link for an inference-node that is an inference/defeat-ancestor of *link*.
  - The inference/defeat-ancestors of a node are its support-links and their inference/defeat-ancestors.
- In computing the defeat-status of a node or link, the only nodes and links that are relevant are its inference/defeat-ancestors.

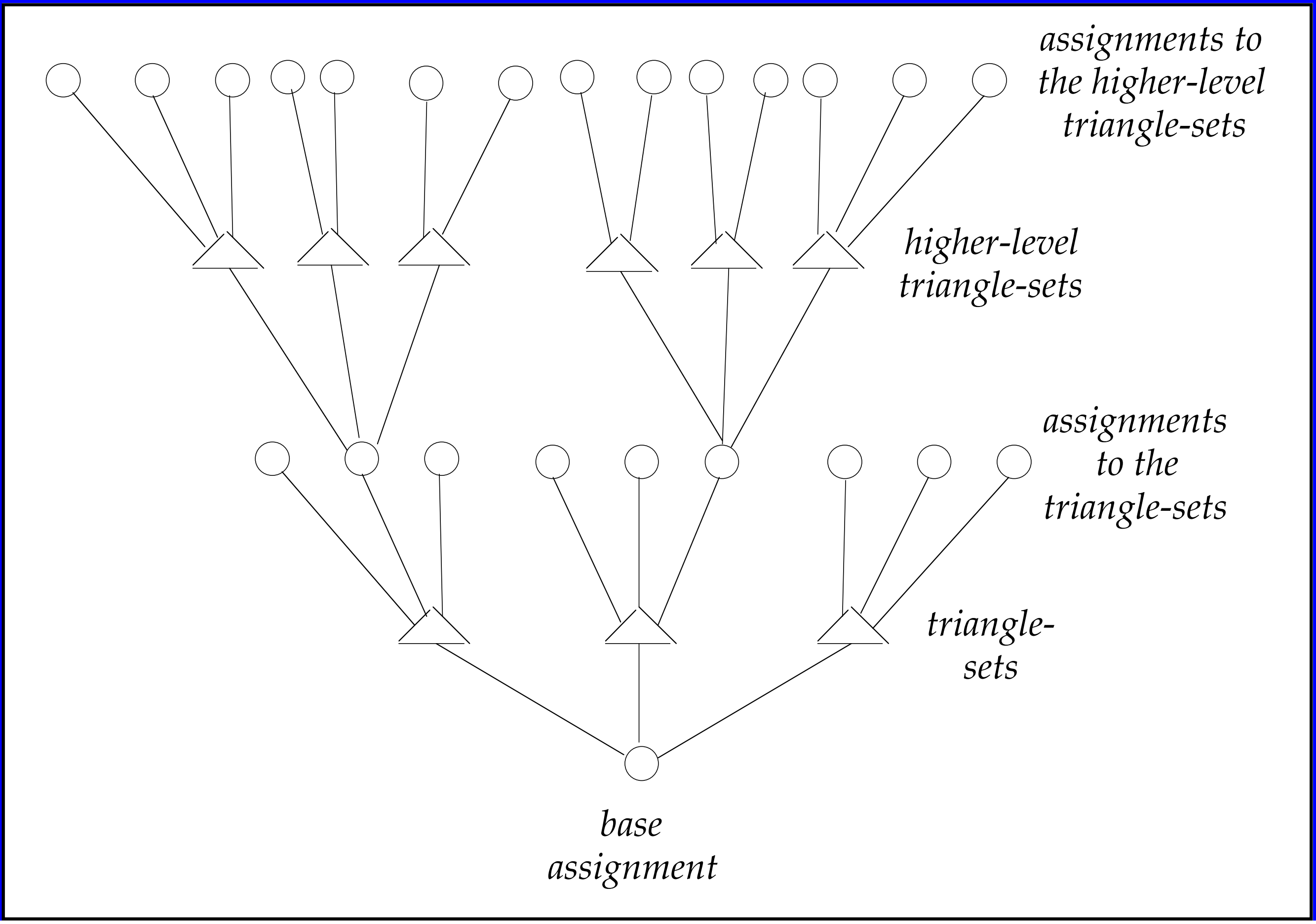
Triangle Sets

- When we are presented with different subproblems embedded in a single inference-graph, we can divide the inference-graph into separate regions:
  - D is the set of nodes and links assigned statuses by the base-assignment.
  - The “triangle-sets” A, B, and C represent isolated subproblems. They are characterized by the fact that they are minimal sets X such that every inference/defeat-ancestor of a member of X is in either X or D.
  - For the purpose of computing the defeat-statuses of the members of the triangle-sets, we can treat the A∪D, B∪D, and C∪D as separate inference-graphs. This will yield the same value for the defeat-status as a computation based upon the entire inference-graph.



Assignment-Trees

- The construction of triangle-sets can be iterated, because there may be further triangle-sets that sprout above the existing ones.
- A simple way to conceptualize this is in terms of an ***assignment-tree***.


assignments to the higher-level triangle-sets
higher-level triangle-sets
assignments to the triangle-sets
triangle-sets
base assignment

Belief Updating

- The generate-and-test algorithm for computing status-assignments for and/or graphs can be modifed to apply to assignment-trees, and this constitutes an efficient way of computing defeat-statuses for an inference-graph that has already been constructed.

- That is an artificial problem.
  - A rational agent must construct its inference-graph one support-link at a time, and update the defeat-statuses with the addition of each link.
  - The result of updating the assignment-tree with the addition of each link should be the same as the result of computing the assignment-tree as above for the completed inference-graph.

Belief Updating

- The ***inference/defeat-descendants*** of a support-link consist of its target and all of the inference/defeat-descendants of its target; the inference/defeat-descendants of an inference-node consist of (1) all the consequent-links of the node and all the inference/defeat-descendants of those consequent-links, and (2) all the node-defeatees of the node and all the inference/defeat-descendants of those node-defeatees.

- When we add new links, the changes to the inference-graph are confined to the new links and their inference/defeat-descendants. Call these the ***affected-nodes*** and ***affected-links***.

Belief Updating

- If the affected-nodes have no defeatees, we can just compute the assignments to the affected-nodes and affected-links at any point in the assignment tree at which they become computable, and add those assignments to the assignment-tree.
- Otherwise, we must update *global-assignment-tree*:
  - In the unusual case in which some of the affected-nodes and affected-links occur in the base assignment, then we must recompute the entire assignment-tree.
  - Normally that will not be the case. Then we can just walk up through the tree until we come to the first triangle-sets containing affected-links. We need only recompute the parts of the tree that branch above such triangle sets.

Efficiency of the Algorithm

The difference between the performance of the algorithms is illustrated dramatically by considering composites of figure ten again. For a single instance of that problem, the original assignments-algorithm is the same speed as the assignment-tree-algorithm. But when we consider composite problems, the comparison changes dramatically:

	assignments-algorithm	assignment-tree-algorithm
Figure 10	0.07 sec 4 assignments	0.07 sec 4 branches
2 instances of #10	1.20 sec (x17.14) 16 assignments	0.15 sec(x2.14) 8 branches
3 instances of #10	8.24 sec (x260.57) 64 assignments	0.25 sec(x3.57) 12 branches
4 instances of #10	310.67 sec (x4438.14) 256 assignments	0.35 sec(x5.0) 16 branches