# A Probabilistic Analysis of Marker-Passing Techniques for Plan Recognition

Glenn Carroll and Eugene Charniak

Department of Computer Science
Brown University
Providence, Rhode Island 02912

**CS-91-44**
June 1991

# A Probabilistic Analysis of Marker-Passing Techniques for Plan-Recognition *

Glenn Carroll
Eugene Charniak
Department of Computer Science, Brown University

June 23, 1991

**Abstract**

Useless paths are a chronic problem for marker-passing techniques. We use a probabilistic analysis to justify a method for quickly identifying and rejecting useless paths. Using the same analysis, we identify key conditions and assumptions necessary for marker-passing to perform well.

*This work has been supported by the National Science Foundation under grant IRI-8911122 and by the Office of Naval Research, under contract N00014-88-K-0589. ©1990 Eugene Charniak

# 1 Introduction

A recognition problem is one of inferring the presence of some entity from some input, typically from observing the presence of other entities and the relations between them. We will make the common assumption that high-level recognition is accomplished by selecting an appropriate *schema* from a schema library. A schema is a generalized internal description of a class of entities in terms of their parts, their properties, and the relations between them. In the schema selection paradigm, to recognize a "foo" in the input is to create a schema instance foo1 of type foo and assign a high degree of belief in the proposition that foo1 exists. (Henceforth we will assign the degree of belief in the existence of a schema instance to the proposition that the instance is of the appropriate type, e.g., that foo1 is of type foo.) In plan recognition, the generalized plans are schemas. While the system which we will discuss has been applied to plan recognition in the context of story understanding, we will continue to talk of schema, since we wish to emphasize that our system is applicable to high-level recognition in general.

A crucial problem faced by schema selection is that of searching the schema library for the right schema; typically a single piece of local evidence is multiply ambiguous as to the schema which it could indicate. For example, an act of getting a rope might fit into many schemas.

One of the few concrete proposals to solve this problem has been *marker passing* (Alterman [1985]; Charniak [1983]; Charniak [1986a]; Collins & Quillian [1969]; Hendler [1988]; Norvig [1987b]). Marker-passing uses a breadth-first search to find paths between concepts in an associative network made up of concepts and their part-subpart relations. In our case, the concepts will be schemas, i.e. plans and/or actions. The idea is that a path between two schemas suggests which schema(s) to consider for recognition. For example, a knob instance and a hinge instance might suggest a door (instance); since there are links between the schemas door and knob and between door and hinge in the associative network (they are part-subpart relations), there is therefore a path from knob through door to hinge. Unfortunately, most marker-passer systems have found many more bad paths, suggesting incorrect schemas, than good ones (Charniak [1986a]; Norvig [1987a]). We will show in this paper that the good/bad path ratio can be raised quite high by exploiting probability information; we realize this benefit by (cheaply) controlling the marker-passer's search, extending it in promising directions and terminating it in unpromising ones.

In this paper we will give a probabilistic account of marker passing. This account will have two goals — first, as described above, it should show how to improve the performance of marker-passing algorithms by increasing the likelihood that the paths generated will, in fact, suggest the correct schema, and second it should shed further light on when marker-passing is an appropriate technique. In section 2 we will discuss previous work in this area. In section 3 we will consider schema evaluation within a probabilistic framework. That is, given we have a potential schema, how do we evaluate the probability that it is the correct explanation of the input. In particular, we adopt a Bayesian network (or belief network) formulation of the problem, so the probability distributions correspond to DAGs with probabilities

associated with each node. Section 4 will then be concerned with schema selection, i.e., how our marker passing system works, and how the schema suggestions (paths) from the marker-passer map to Bayesian networks. In Section 5 we will show how to use probability information from the knowledge base to intelligently limit the marker-passing search. Specifically, we will describe how to calculate on the fly a measure which is an upper bound on the joint probability of the schemas which a candidate path suggests. Thanks to properties of the marker-passer paths and our probabilistic model, we can avoid constructing and evaluating a Bayesian network to evaluate each path, an NP-hard problem (Cooper [1987]), so our evaluation need not be expensive. This section has the bulk of the new work in the paper. We summarize and explain results in section 6.

The opening sections of this paper alternate somewhat irregularly between the marker-passer and the Bayesian network; while it might appear to be simpler to fully describe first one and then the other, this would leave much of what we have to say either unmotivated or obscure or both. Once we reach section 4, we treat the two systems together, showing how paths map to Bayesian networks, and how path calculations yield an upper bound on the joint probability of the nodes in the Bayesian network.

# 2 Previous Work

Allen proposed a scheme (Allen [1979]) to do plan recognition in the context of cooperative dialogue. The search for candidate plans was essentially carried out using a forward-chaining search up the plan hierarchy, unifying proposed plan alternatives with the system's expectations and current plan information. He used a heuristic scheme of non-probabilistic rating numbers to prune the search and control the order in which plan alternatives were investigated. For example, unifying a plan alternative with an expectation increased the rating of the alternative; proposing several alternative plans to explain the current one would result in the current plan's rating being divided among each of the explanations. Allen's system could not correctly deal with a case in which a single piece of evidence fit a great many schemas. For example, the act of going somewhere presumably fits into a great many schemas: one goes to work, one goes grocery shopping, one goes to concerts for entertainment, one goes home for dinner, etc, etc. In Allen's system, the probability of each of these explanations would share an equal piece of the probability of the act of going, and none of them would end up high enough to drive further processing.

Calistri developed a system for recognizing plan-based misconceptions in (Calistri [1989]) which required a modified form of plan recognition. He used a modified form of the A* graph search algorithm to search for (possibly flawed) plan explanations, again in the context of a cooperative dialogue. Calistri assumed that a high-level plan would be made available to his system. While this may be plausible for a system searching for plan errors in a cooperative dialogue setting, it is patently untrue that explanations are generally made explicit in stories. Indeed, if the explanations were given in the text, we could skip searching for explanatory

schemas entirely.

# 3 Probabilistic Schema Evaluation

We adopt a standard first-order theory of schemas in which a schema is a set and asserting that an entity is an instance of that schema is asserting that it is an element of the set. We use the predicate inst for this purpose.

(inst *instance schema*).

Schemas are related in the usual isa-hierarchy (subset) as in

(isa *specific-schema general-schema*).

In this paper we assume that isa relations form a tree, not a lattice, and thus all the immediate isa subsets of a given parent are disjoint.

Slots or roles in schemas are represented using functions from a schema instance to the slot-filler schema instance. Equality is used to assert that a particular entity fills that role. For example, to assert that a particular store store-25 fills the store-of role in supermarket-shopping-3 we assert

(== (store-of supermarket-shopping-3) store-25)

Facts about the relations between the parts of a schema are then universally quantified facts about the corresponding functions. For example, to say that every store-of a supermarket-shopping must be filled with an instance of a supermarket (another schema) we would say

(inst ?x supermarket-shopping) → (inst (store-of ?x) supermarket)

To abbreviate this we will write: (role supermarket-shopping store-of supermarket). More generally,

(role $schema_1$ *slot* $schema_2$)

states that anything which fills *slot* in $schema_1$ must be an instance of $schema_2$. Note that role is not a predicate of our plan recognition language, but is rather an abbreviation for formulas of the above form.

In our probabilistic version we will determine the probability of a plan by embedding inst and == statements in a Bayesian network. We will not attempt to summarize Bayesian networks but rather will assume the reader has a working knowledge of them. (See (Pearl [1988]) for a good introduction.) Equality (==) statements will become random variables with possible values 1 and 0 (true and false). inst statements become random variables which can take any maximally specific schema type as their value[1]. Thus the probability of the statement (inst sms1 supermarket-shopping) would become the probability that the instance sms1 takes on the value supermarket-shopping. However, often we will talk as if the statement (inst sms1 supermarket-shopping) appears in the network (with values 1 and 0). Most of the time the two representations are interchangeable.

We intend our prior and conditional probabilities to come from a sample space of explanations for some large corpus of stories. For example, the prior probability of a supermarket-shopping plan would be the number of supermarket-shopping plans

[1] Thus, the sum over all possible values is 1.0, which would not be true if they could take on non-maximally specific types.

that appear in our set of explanatory plans, divided by the total number of explanatory plans. See (Goldman [1991]) for a detailed description of the probability model.

# 4 Probabilistic Schema Selection

## 4.1 Marker-Passing

Marker-passing searches for paths between schemas in a graph whose nodes are schemas and whose arcs are isa and role statements. Marker-passing works as follows: the marker-passer is given some schema, derived from a new inst statement, e.g., (inst supermarket1 supermarket). It places a mark on that schema, and then proceeds in breadth-first order to place marks on all the neighbors in the graph, their neighbors' neighbors, and so on. For example, our supermarket schema has two neighbors, supermarket-shopping, which is connected by the statement (role supermarket-shopping store-of supermarket) and store-, which is connected by the statement (isa supermarket store-). Both of these would be marked after supermarket, and then their neighbors would be marked, and so on. Each mark has a numeric value, which generally diminishes according to its distance from the starting point (c.f., "zorch" in (Charniak [1986a])). This value is used to cut off marker-passing, since otherwise we would continue until the entire graph was covered. For our value, we use an upper bound on the schemas and relations suggested by the path; we will be precise in section 4. After marking a node, the marker-passer checks for marks from some other origin on the same node. If such a mark is found, both it and the new mark are back-traced to their respective origins, and the resulting lists of statements are glued together to form a path. For example, suppose we found a mark on supermarket-shopping which had originated at go. We would have as a path:

```
(inst supermarket2 supermarket)
(role supermarket-shopping store-of supermarket)
(isa shopping supermarket-shopping)
(role shopping go-step go)
(inst go1 go)
```

We include the original inst statements, even though the marker-passer does not, strictly speaking, pass marks over these links. It will be convenient for us to refer to them as part of the path, and they serve to disambiguate this path from other paths which may have the same links but different origins.

The marker-passer returns a list of all the paths which it found once the marking has terminated. For more detail on how marker-passing works, see (Hendler [1988]).

## 4.2 Valid Paths and Interpretations

Intuitively, we wish to interpret a path as a claim about how its ends are related to each other. In order to do this, we need to translate the path through the semantic network (a list of inst, isa, and role statements) into a set of Bayesian network nodes

(inst and equality statements) and arcs. Before we describe this mapping, we must elaborate our marker-passing system somewhat. We employ a DFA at each node in our network to control the marker-passing, allowing us to restrict the form of paths which we generate and report. This allows us to skip paths which are malformed in the sense that either they cannot be translated into a consistent set of statements in our schema theory, or they embody demonstrably bad schema suggestions. A valid path is one which is not malformed in the above sense.

NOTATION. By isa- we mean isa with the order of arguments reversed. So,

(isa *specific-frame general-frame*) iff (isa- *general-frame specific-frame*)

Similarly for role and role-:

(role *filler-frame filled-frame slot*) iff (role- *filled-frame filler-frame slot*)

□

In future discussions we will often fail to distinguish between the predicates and their "-" versions.

DEFINITION 4.1. *A* valid path *from* $i_1$ *to* $i_2$ *has the form*

(inst $i_1$ $s_1$) (*pred*$_1$ $s_1$ $s_2$) ... (*pred*$_n$ $s_n$ $s_{n+1}$) (inst $i_2$ $s_{n+1}$)

*where*

- *pred*$_i$ *may be one of* isa, isa-, role *and* role-,
- *at least one* role *appears among the pred*$_i$.
- *no sequence* (isa ...)(isa- ...) *appears among the pred*$_i$
- *if pred*$_i$ *is a* role-, *then no pred*$_k$ *where* $k > i$ *can be a* role

Our last two restrictions prohibit *isa-plateaus*, where an isa is followed by an isa-, and slot-filler valleys, where a role- is followed by a role, possibly with isa's between them. We have calculated, off-line, the joint probability of the statements associated with paths which violate the above restrictions; in all cases the joint probability falls below our threshold for being worth computing[2].

We will now define the set of statements associated with a path $P$, written $S(P)$.

NOTATION. By $P[n]$ we mean the nth statement of a path $P$. □

DEFINITION 4.2. *The* relevant instance *at* $P[n]$ *(written* $I(P[n])$*) is defined to be:*

1. *If* $P[n]$ = *(inst i s), then* $I(P[n]) = i$.
2. *If* $P[n]$ = *(isa* $s_1$ $s_2$*) and* $I(P[n-1]) = i$, *then* $I(P[n]) = i$. *Similarly for isa-.*
3. *If* $P[n]$ = *(role* $s_1$ *slot* $s_2$*) then* $I(P[n]) = i'$, *where* $i'$ *is a new constant term.*

DEFINITION 4.3. $S(P)$ *is defined to be:*

1. *If (inst i s)* $= P[n]$, *then (inst i s)* $\in S(P)$.
2. *If (isa* $s_1$ $s_2$*)* $= P[n]$, *then (inst i* $s_2$*)* $\in S(P)$, *where* $i = I(P[n-1])$. *Similarly for isa-.*
3. *If (role* $s_1$ *slot* $s_2$*)* $= P[n]$, *then* $\{$*(inst* $i_n$ $s_2$*) (==(slot* $i_n$*)* $i_{n-1}$*))*$\} \subset S(P)$, *where* $i_n = I(P[n])$ *and* $i_{n-1} = I(P[n-1])$. *Similarly for role-.*

[2] To summarize: for isa-plateaus, the left and right side of the Bayesian net in which we embed the statements are independent, so they cannot support one another. For the slot-filler valleys, there is no evidence to support the object which must appear in two schemas. See (Charniak & Carroll [1991]) for details.

```
(inst supermarket2 supermarket)
(role supermarket-shopping store-of supermarket)
(isa- shopping supermarket-shopping)
(role go shopping go-step) \nopagebreak
(inst go1 go)
```

FIG. 4.1. An Example Marker-passer Path.

```
(inst supermarket2 supermarket)
(== (store-of shopping3) supermarket2)
(inst shopping3 supermarket-shopping)
(inst shopping3 shopping)
(== (go-step shopping3) go1)
(inst go1 go)
```

FIG. 4.2. $S(P)$ For Example Path.

Intuitively we wish to interpret a path $P$ as a claim about how its ends are related to each other. "Every $s \in S(P)$ is true" is intended to be the formalization of this claim. For example, the path in Figure 4.1 would have as its $S(P)$ the statements shown in Figure 4.2. We actually need only a subset of $S(P)$, namely the *relevant statements* associated with $P$, (written $RS(P)$) which we will define below. First we give two more definitions necessary for defining $RS(P)$.

DEFINITION 4.4. $\forall t, t'$(isa* $t$ $t'$) *iff* $((isa\ t\ t')$ *or* $\exists t''[(isa\ t\ t'')$ *and* $(isa*\ t''\ t')])$.

We define the *relevant type* of an inst (written $RT(i)$) to be the most specific schema type of the node.

DEFINITION 4.5. $RT(i) = t$ *such that* (inst $i$ $t$) $\in S(P)$ *and* $\forall t'((inst\ i\ t') \in S(P) \rightarrow (isa*\ t\ t')$ *or* $t = t')$.

DEFINITION 4.6. *The set of* relevant statements *associated with* $P$, *written* $RS(P)$ *is defined as*

1. *If* $(inst\ i\ t) \in S(P)$ *and* $RT(i) = t$ *then* $(inst\ i\ t) \in RS(P)$.
2. *If* $(==\ (slot\ i)\ j) \in S(P)$, *then* $(==\ (slot\ i)\ j) \in RS(P)$.

In effect, we remove superfluous instance statements from $S(P)$, whose statements are still implied by those retained in $RS(P)$.

Our formal measure of a path $P$ is defined by embedding the members of $RS(P)$ in a Bayesian network, and then evaluating the probability that each node is true, given the evidence. In general there may be several paths for the same entities, indicating alternative possible plans. In what follows we will be looking at Bayesian networks in which there is only one $RS(P)$. The idea is that we are interested in getting a preliminary guess as to how likely a particular interpretation–an $RS(P)$–is, and we can do this without detailed comparisons with its competitors.

## 4.3 Vertebrate Bayesian Networks

With each path $P$ we will associate a Bayesian network with a particular structure which will prove important to our calculations. We call such networks *vertebrate* Bayesian networks, because they have *spines.*

DEFINITION 4.7. *A Bayesian network is a "vertebrate" Bayesian network iff it consists of two parts, to be defined below, called the* spine, *and the* interior.

Intuitively, a spine is the geometrical backbone of its Bayesian network.

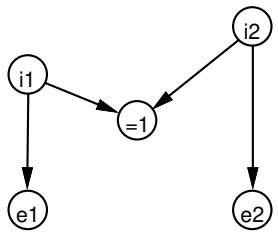


FIG. 4.3. The basic spine.

NOTATION. We use $i_j$ to name inst nodes. □

Recall that equality nodes represent slot-filler relationships for us.

NOTATION. We use $=_j$ to name equality nodes, indicating in this case that the parent node $i_j$ is the slot filler. □

There are other types of nodes which act as evidence for our equality and inst nodes. For example, the appearance of a word in text provides evidence for the existence of a particular inst. Although these nodes come in several flavors, we can treat them generically for our purposes, so we will name all of them simply "evidence" nodes.

NOTATION. We use $e_j$ to name evidence nodes. □

Our definition describes a structure topologically; thus, we imply that a node subscripted by $j$ is not equal to any node subscripted by $k$, where $k \neq j$.

Legal spines are recursively defined as follows:

DEFINITION 4.8.

**Base step:** *Any Bayesian network whose node set is* $\{i_1,\ i_2,\ =_1,\ e_1,\ e_2\}$ *and whose edge set is* $\{i_1 \rightarrow e_1,\ i_2 \rightarrow e_2,\ i_1 \rightarrow =_1,\ i_2 \rightarrow =_1\}$ *is a spine. See Figure 4.3.*

**Recursion:** *If* $S$ *is a spine with the nodes* $i_1$ *and* $e_1$ *and the edge* $i_1 \rightarrow e_1$, *then* $S'$ *is a spine, where*

$$N(S') = N(S) \cup \{i', =_{i'}\} \tag{1}$$

*and*

$$\begin{aligned} E(S') \quad = \quad & \{E(S) - \{i_1 \rightarrow e_1\}\} \cup \\ & \{i_1 \rightarrow i',\ i' \rightarrow e_1, \\ & i_1 \rightarrow =_{i'},\ i' \rightarrow =_{i'}\} \end{aligned} \tag{2}$$

*See Figure 4.4.*

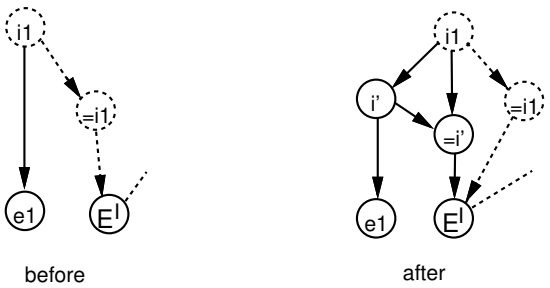


FIG. 4.4. Before and after nodes are added to a spine.

As for the "interior" of a vertebrate Bayesian network, intuitively it is the evidence supporting the equality nodes of the vertebrate Bayesian network. We may assume, without loss of generality that there is only once such evidence node $E^I$. More formally:

DEFINITION 4.9. *If $V$ is a vertebrate Bayesian network with spine $S$, and $S$ has the non-evidence nodes $N$ and the evidence nodes $E$, then the interior of $V$ is an evidence node $E^I$ disjoint from $E$ and a set of edges $D$ from every equality node in $N$ to $E^I$.*

The assumption that there will be supporting evidence, some $E^I$ node, is the crucial one for marker-passing. When there is no evidence, the posterior probabilities of the abductive hypotheses generated from our paths turn out to be abysmally low; plainly put, they are bad guesses. In general, we believe that

CLAIM 4.1. *A domain is suitable for search by means of marker-passing only if there will usually be supporting evidence for paths returned by the marker-passer.*

We will support our claim by showing that, in our domain, which meets the evidence condition, we can increase the ratio of good to bad paths returned from the marker-passer to better than 90%. Our claim should not be interpreted as saying that the marker-passer has no responsibility for the quality of paths it returns; quite the opposite is true. Most of the remainder of this paper will focus on the calculations which allow us to determine whether or not a path is worthwhile, *assuming that there is evidence for it.* If we could not make these calculations, then doubtless many of the paths which would be returned would in fact *fail* to have associated evidence. We can safely throw them out because they are bad paths *regardless* of whether they have evidence.

## 4.4 Relating Paths to Networks

We will now show that each path corresponds to a unique vertebrate Bayesian network, and that the joint probability of $S(P)$ in the network can be calculated from each step of the path.

THEOREM 4.1. *If $P$ is a valid path, then there exists a unique vertebrate Bayesian network $V$ such that the statements in $RS(P)$ have formulas in one-to-one correspondence with the non-evidence nodes of $V$.*

*Proof.* The proof is by induction on the length of the path.

The basis follows from the definition of the simplest spine, and $RS(P)$ for the shortest valid path. The spine and $RS(P)$ have one == node (statement, respectively) and two inst nodes (statements, respectively). For the induction step, suppose we have proved there is a unique network for $P$. Let $P'$ have the same structure as $P$, with an extra isa statement inserted in some location which does not violate our constraints for path validity. We first note that isa statements do not add statements to $RS(P)$, they only change the schemas named in statements already there. Hence, if we have a vertebrate Bayesian network for $P$, we can use the same structure (with different relevant types for some nodes) for $P'$.

If $P'$ is $P$ with an extra role statement (again, inserted in some location which respects path validity), this corresponds to applying clause 2, the recursion step, of the definition of a spine. The role statement adds an == statement and an inst statement to $RS(P)$, and applying clause 2 adds the corresponding nodes to the network. Since each step determines a unique transformation and each transformation results in a unique vertebrate Bayesian network, $P$ corresponds to a unique vertebrate Bayesian network, which we call $V(P)$, or, where unambiguous, simply $V$. □

From the above, it should be obvious that we can construct a vertebrate Bayesian network by sequentially processing a path from left to right, adding new nodes and arcs for each role statement we come across, and changing the relevant type of our last added node when we encounter an isa-. We do not need any global information about the path to construct the corresponding vertebrate Bayesian network. We will use this fact, together with some distribution properties of vertebrate Bayesian networks to calculate our measure of path utility without actually constructing and evaluating the vertebrate Bayesian network corresponding to the path.

## 4.5 Some Partial Distributions for Vertebrate Bayesian Networks

We need some partial distribution information to carry out our calculations based on paths. These will not be complete distributions, only enough to allow us to compute our measure.

We give our distribution information on a node-type-by-node-type basis:

**Inst nodes.**
These nodes may or may not have a parent in the network. If not, we define the value of interest to be $p(RT(i_1))$, [3] the prior of the relevant type of the node.

[3] Note that, for the marker-passer, the links it traverses connect nodes which are schemas; for the Bayesian network, the links connect nodes which are instances. Thus, in our short-hand notation we write $p(i_1)$, meaning $p(i_1 = s_1)$, when it is clear that there is only one schema under consideration. When we have no instance to refer to, or it is otherwise convenient, we may write simply $p(s_1)$.

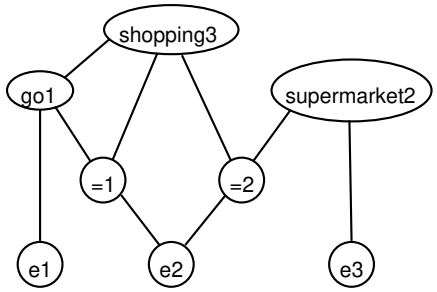


FIG. 4.5. A typical vertebrate Bayesian network.

If the node does have a parent, it will be connected with an *existential* arc, such as the arc between shopping3 and go1 in figure 4.5. The existential arc indicates that we create the parent shopping3 in order to explain go1; hence, the probability of go1 given shopping3 is 1.0. (Recall that the existential arc corresponds to a role statement in a path.) More formally, if an existential arc connects $i_1$ and $i_2$, then it specifies schemas, $s_1$ and $s_2$ where $s_1$ is the schema for the slot-filler, and $s_2$ is the schema for the plan in which the slot is filled, and

$$p(i_1 = s_1 | i_2 = s_2) = 1.0 \tag{3}$$

We also assume that

$$p(i_1 = s_3 | i_1 = s_1, i_2 = s_2) = p(i_1 = s_3 | i_1 = s_1) \tag{4}$$

where $i_1$ and $i_2$ are connected exactly as above, and $s_3$ is some schema type more specific than $s_1$. In effect, we are assuming that the existential link influences $p(i_1 = s_1)$, but has no influence beyond that. Finally, we assume that

$$p(i_1 = s_1 | i_2 = s_4) = p(i_1 = s_1 | i_2 = s_2) \tag{5}$$

Where $s_4$ is some schema more specific than the schema which the existential arc indicates for the parent node. Since the specific schema implies the general schema, it is not unreasonable to assume the same distribution for both. We will make virtually identical assumptions for equality and evidence nodes; that is, we will assume that the distribution for a node with a more specific schema is the same as that for the more general. For more detailed discussion of the existential arc, see (Charniak & Goldman [1989]).

**Equality nodes.**
The distribution of these nodes depends on whether or not an existential bridges the parent nodes; for example, the $=_2$ node in figure 4.5 has no existential, while $=_1$ does. In the former case, we want to know the probability that the filler of the store-of slot of shopping3 is equal to supermarket2, given that both the slot-filler and supermarket2 are supermarkets. This is just the probability that any two supermarkets are the same, which is 1/number of supermarkets in our universe;

the prior of supermarket is the number of supermarkets in the universe divided by the total number of objects in the universe, so we can divide out and get $p(=_2 |\mathsf{shopping3}, \mathsf{supermarket2}) = p(=)/p(\mathsf{supermarket})$, where $p(=)$ is the probability that any two things in the universe are equal. In general, such an equality statement indicates a schema $s_1$ (derived from the role-inst statement it represents), and its distribution is

$$p(=_1 |i_1, i_2) = p(=)/p(s_1) \tag{6}$$

If there is an existential arc bridging the parents of an equality node, then its distribution is

$$p(=_1 |i_1, i_2) = 1.0 \tag{7}$$

since we are creating $i_2$ to explain $i_1$ and we therefore believe that $i_1$ must fill this slot, given $i_2$.

In both existential and non-existential cases, it is possible that the node for the parent plan, e.g. shopping3 in (== (store-of shopping3) supermarket2), has a schema type more specific than that indicated by the existential arc. We assume that this has no effect, that is,

$$p(=_1 |i_1, i_2 = s_3) = p(=_1 |i_1, i_2 = s_2) \tag{8}$$

where $s_3$ is more specific than $s_2$.

**Evidence nodes.**
For evidence nodes underneath inst nodes, namely $e_1$ and $e_3$, we make only the assumption that, given schema $s_1$ is less specific than $s_2$,

$$p(e|i_1 = s_1) = p(e|i_1 = s_2) \tag{9}$$

as we did for inst nodes and equality nodes.

For $e_2$ we make the assumption above, and also assume that

$$p(e_2|e_1, e_3) \leq p(=) \tag{10}$$

and

$$p(e_2| =_1, =_2, ...) = 1.0 \tag{11}$$

In effect, we are assuming that $e_2$ has a very low prior probability, that it is independent of $e_1$ and $e_3$, and that it is true given our path interpretation, $RS(P)$. [4] We will come back to the implications of this later.

[4] Strictly speaking, we should write

$$p(e_2|i_1, =_1, i_2, =_2, ...) = 1.0 \tag{12}$$

to indicate that our evidence node is, in principle, conditional upon the inst nodes as well. For the networks our system currently generates, $e_2$ is independent of the inst nodes, given the equality nodes, so we omit connections to the former.

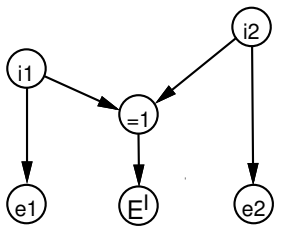


FIG. 5.1. The basic vertebrate Bayesian network.

# 5 Path Calculations

## 5.1 The Spinal Contribution

Our calculations will not compute the exact joint probability of the network which we construct. Instead we will compute an upper bound on the joint probability, under assumptions to be detailed below, which we call the *spinal contribution*. We will define the spinal contribution momentarily, in terms of the joint probability of the network. To begin with, the exact joint probability of the simplest vertebrate Bayesian network (the basic spine with an added interior evidence node, $E^I$, pictured in Figure 5.1) is

$$\frac{p(e_1|i_1)p(E^I|=_1)p(e_2|i_2)p(i_1)p(i_2)p(=_1|i_1,i_2)}{p(e_1,E^I,e_2)} \tag{13}$$

We use conditional probability and independence to transform the denominator into

$$p(e_1)p(e_2)p(E^I|e_1,e_2) \tag{14}$$

For the numerator, we note that

$$p(e_1|i_1) = p(i_1|e_1)p(e_1)/p(i_1) \tag{15}$$

and similarly for $p(e_2|i_2)$.

The slot-filler term, $p(=_1|i_1,i_2)$, requires some discussion. Recall that role statements specify a particular type for each slot that a schema has. The probability that $i_1$ fills this particular slot in $i_2$ is the probability that any two things of the specified type are equal. This, in turn, is equal to the prior probability of any two things being equal, divided by the prior probability of a thing being the specified type.

NOTATION. We write $p(==)$ for the prior probability of any two things being equal. □

This allows us to rewrite the last term in our formula as follows:

$$p(=_1|i_1,i_2) = p(==)/p(i_1) \tag{16}$$

Applying the substitutions in equations 14, 15, and 16 to 13, and then cancelling and regrouping yields

$$\left(\frac{p(i_1|e_1)p(i_2|e_2)}{p(i_1)}\right)\left(\frac{p(==)p(E^I|=_1)}{p(E^I|e_1,e_2)}\right) \tag{17}$$

The right-hand group of terms will appear in the exact calculations for every vertebrate Bayesian network, with the difference that $E^I$ may be conditioned on more nodes, if they are present. By our earlier assumptions about the distributions for $E^I$, this group has an upper bound of 1.0.

The left-hand group of terms we call the *spinal contribution* of our joint probability; more generally, any terms not included in the bounded group, will be part of the spinal contribution, and calculating it will be the focus of the rest of this section. Since those terms not in the spinal contribution have an upper bound of 1.0, the spinal contribution is an upper bound on the joint probability of the network.

## 5.2 Calculations

THEOREM 5.1. *As a path $P$ is traversed, our measure of the spinal contribution of the corresponding vertebrate Bayesian network fragment, $SC(V)$, can be computed recursively in the following manner:*

1. *The initial value, corresponding to the* (inst $i_1$ $s_1$) *node/statement* [5] *is* $p(i_1|e_1)$, *our current belief in the node.*
2. *As each subsequent statement is traversed, we compute the new value by multiplying the current value by the number given in table 5.1.*

TABLE 5.1. Spinal Contribution Multipliers

| *Link* | *Multiplier* |
|---|---|
| (role $s_1$ slot $s_2$) | $p(s_1)/p(s_2)$ |
| (role- $s_1$ slot $s_2$) | *1.0* |
| (isa $s_1$ $s_2$ ) | *1.0* |
| (isa- $s_1$ $s_2$ ) | $p(s_2)/p(s_1)$ |
| (inst $i_1$ $s_1$ ) | $p(i_1|e_1)/p(s_1)$ |

*Proof.* The proof is by induction on the length of the path.

**Base case:**

For the statement (inst $i_1$ $s_1$), our initial value for $SC(V)$ is which is $p(i_1|e_1)$, and the corresponding network fragment is as shown in figure 5.2. The joint probability of this network fragment is

$$p(e_1|i_1)p(i_1) \tag{18}$$

[5] This is a node in the Bayesian network, and a statement in the path. Since we will be discussing probabilities from now on, we will generally call them nodes.

FIG. 5.2. The starting vertebrate Bayesian network.

Applying Bayes' law and cancelling yields:

$$p(i_1|e_1)p(e_1) \tag{19}$$

Recall that in the computation of $SC(V)$ for the entire network, we divide by $1/p(e_1, e_2, e_3)$, which, under our independence assumptions, resolves into $1/p(e_1)p(e_3)p(e_2|e_1, e_3)$. The term $p(e_1)$ above cancels with a term from this group, so that our spinal contribution is

$$p(i_1|e_1) \tag{20}$$

which is what we claimed for our initial value.

**Induction Step:**

For the induction step, we assume that we have computed $SC(V)$ correctly for all the links we have seen so far. We proceed by enumerating the various possible link types we can encounter, and showing that for each type we correctly compute $SC(V')$ for the extended network fragment. Our procedure for role statements, which add nodes, will be to simply multiply the current value by the conditional probabilities of the nodes added to the network; for isa statements, which only change the relevant types, we will simply correct for our new understanding of what the relevant type of the current inst node is. As a general rule, we will use $i_1$ for the last inst node added to the network fragment, and $i_2$ for the added inst node.

**Role statements:**

There are two cases for role statements, depending on whether or not an existential is added to the network, as shown in figure 5.3. [6] For the non-existential case,

$$SC(V') = SC(V)p(=_1 |i_1, i_2)p(i_2) \tag{21}$$

Note that $e_2$ does not appear in the spinal contribution, so it does not appear in our calculations.

[6] Note that we assume that $i_2$ is not directly connected to some other evidence, as would be the case if it turned out that the next statement was (inst $i_2$ $s_2$). If that is the case, it will be taken care of when we cross the inst link.

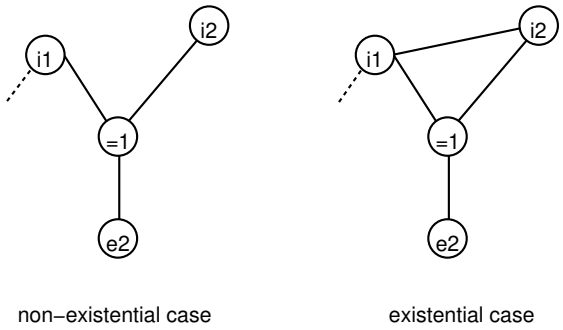


FIG. 5.3. Network additions for the statement (role $i_2$ *slot* $i_1$) .

From our earlier discussion, we know that

$$p(=_1 | i_1, i_2) = p(=)/p(i_1) \tag{22}$$

The term $p(=)$ is part of our bounded group of terms in 17, so we ignore it. Equations 21 and 22 give us a multiplier of $p(i_2)/p(i_1)$, which is exactly what we claimed.

In the above, we have ignored the relevant type of nodes $i_1$ and $i_2$, because they don't change the calculations. The relevant type of node $i_2$ after crossing the role statement is the type named by the role. For example, if we crossed (role supermarket-shopping store-of supermarket), the relevant type of the new node, say *shopping*3 would be shopping.

The relevant type of node $i_1$, say $s_1$, may be more specific than the slot-filler type named by the role, say $s_2$. However, we have assumed that the distribution for the specific type is the same as that for the general type, so

$$p(=_1 | i_1 = s_2) = p(=_1 | i_1 = s_1) \tag{23}$$

In general, since we have made the assumption that the distribution for the specific type is always the same as for the general type, anything conditioned on a node having a general type will have the same conditional probability if it is conditioned on the specific type.

The existential case is more complicated. Again, the node $i_2$ must have the relevant type named by the role statement, since this is the only information we have on it so far. Again, $i_1$ may have a type more specific than the slot-filler type named by the role statement, but this time it changes the calculations. We begin with the case where the relevant type is the same as that named by the role statement. In our shorthand notation, we have

$$SC(V') = SC(V)p(=_1 | i_1, i_2)p(i_2)p(i_1|i_2)/p(i_1) \tag{24}$$

The term $p(i_1)$ appears because we previously assumed that $i_1$ had no parents, and we now know that to be wrong. The semantics of our existential arc tell us that the first and last term are 1.0; hence, our multiplier is $p(i_2)/p(i_1)$, exactly as before.

If, however, the relevant type of $i_1$, say $s_s$, is more specific than the schema type named by the role statement, say $s_g$, then the term $p(i_1|i_2)$ above is not 1.0. Letting $X_1$ stand for the proposition $i_1 = s_s$ and $X_2$ stand for $i_1 = s_g$, and using our expanded notation, we rewrite the term $p(i_1|i_2)$ as

$$p(i_1 = s_s|i_2) = p(X_1|i_2) \tag{25}$$

$$p(X_1|i_2) = p(X_1|i_2, X_2)p(X_2|i_2) + p(X_1|i_2, \overline{X_2})p(\overline{X_2}|i_2) \tag{26}$$

The term $p(X_1|i_2, \overline{X_2})$ is 0.0, since the specific $i_1$ cannot be a member of the specific schema and not the general one. That leaves us

$$p(X_1|i_2, X_2)p(X_2|i_2) \tag{27}$$

The last term here is 1.0, by the semantics of our existential arc. This leaves us

$$p(X_1|i_2, X_2) \tag{28}$$

which, by our assumption about distributions under existential arcs equals

$$p(X_1|X_2) = p(X_1)/p(X_2) = p(i_1 = s_s)/p(i_1 = s_g) \tag{29}$$

since the membership in the general schema makes membership in the specific schema more likely, in exactly this ratio.

For the equality term, $p(=_1 |X_1, i_2)$, we note that, just as above, the specific type implies the general type, and the semantics of our existential arc still apply, so

$$p(=_1 |X_1, i_2) = p(=_1, X_1, i_2) = 1.0 \tag{30}$$

This leaves us with

$$p(i_2)p(i_1 = s_s)/p(i_1 = s_g) \tag{31}$$

all of which should be divided by $p(i_1)$ from 24, as before. In expanded notation, we should divide by $p(i_1 = s_s)$, which cancels with the same term in the numerator, leaving us with $p(i_2)/p(i_1 = s_g)$. This looks slightly differently than it did before, because we have had to use the expanded notation to be precise. It is, however, the same multiplicative factor.

**Role- statements:**

When we cross a statement (role- $s_1$ *slot* $s_2$), the added network fragment looks like that illustrated in figure 5.4. As for the role case, only the nodes $=_1$ and $i_2$ are new, and there are two cases, depending on whether or not an existential arc is added. For the non-existential case,

$$SC(V') = SC(V)p(=_1 |i_1, i_2)p(i_2) \tag{32}$$

The relevant type of $i_2$ must be that named by the role statement. The relevant type of $i_1$ may be more specific, but this doesn't change the probability of equality statement, which is still $p(=)/p(i_2)$, leaving us with

$$p(=) \tag{33}$$

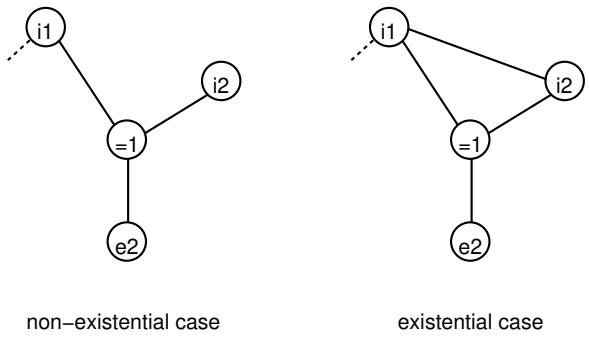


FIG. 5.4. Network additions for the statement (role- $s_2$ *slot* $s_1$).

As before, this term is not in our spinal contribution, so our multiplier is 1.0.

In the existential case, we multiply by

$$p(=_1 |i_1, i_2)p(i_2|i_1) \tag{34}$$

The equality statement has probability 1.0, regardless of the relevant type of $i_1$. Similarly for $p(i_2|i_1)$.

**Isa statements:**
Unlike role and role- statements, traversing the statement (isa $s_1$ $s_2$) adds nothing to the network. Isa statements change the values in which we are interested, namely the relevant types of inst nodes, but when we traverse them from specific types to general types, even those don't change. Hence, our multiplier is 1.0.

**Isa- statements:**
When we traverse an isa- statement, (isa- $s_1$ $s_2$), we change the relevant type of the node we are at. If this node has no parents, we need only multiply by the prior of the new type, and divide by the prior of the previous type. We are cancelling our previous assumption about the relevant type, and making a new one. Since the specific type implies the general type, no node which is conditional upon this one is affected by revising the relevant type. Therefore, we need make no other calculations.

If we are not at a peak, then there must be an existential connecting this node to a parent. Let the parent be $i_1$, the new node be $i_2$, the previous relevant type $s_1$ and the new relevant type $s_2$. See figure5.4. We are interested in

$$p(i_2 = s_2|i_1) \tag{35}$$

Earlier we calculated for the case of a role statement following an isa statement and creating an existential link; this is the converse case, an isa- following a role-. Letting $s_g$ be the (more general) slot-filler type named by the role statement, and $s_s$ be the relevant (more specific) type of the child inst node, we earlier showed that

$$p(i_1 = s_s|i_2) = p(s_s)/p(s_g) \tag{36}$$

Now, before we crossed this isa- link, we had calculated

$$p(s_1)/p(i_2) \tag{37}$$

We now believe the correct calculation should be

$$p(s_2)/p(i_2) \tag{38}$$

So we multiply by $p(s_2)$ and divide by $p(s_1)$, exactly as we do in the non-peak case.

**Inst statements:**

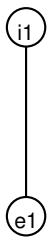


FIG. 5.5. Network additions for the statement (inst $i_1$ $s_1$).

An inst statement terminates the path, and indicates the last evidence node should be added to complete the spine, as shown in figure 5.5. The network additions add the multiplier term

$$p(e_1|i_1) = p(i_1|e_1)p(e_1)/p(i_1) \tag{39}$$

The term $p(e_1)$ is included in our bounded term, so we drop it, leaving the multiplier

$$p(i_1|e_1)/p(i_1) \tag{40}$$

□

## 5.3 Internal Calculations

Marker-passing produces whole paths as output; internally, however, it builds these from two half-paths which resulted from passing marks from two differing origins (at different times). We have proved that we can calculate our path metric given an entire path; what like to do is compute our measure of spinal contribution to cut off the depth of marker-passing, which requires that we compute it as the half-path is built, before the two halves are put together. We now show that this is possible, and that the calculations for an entire path, above, comprise the bulk of the work for computing half-paths. Our lemma concerns the spine, not the entire vertebrate Bayesian network, since the interior evidence node is not part of our spinal contribution.

DEFINITION 5.1. *By* cleaving *a vertebrate Bayesian network graph at some* inst *node n, we mean that*

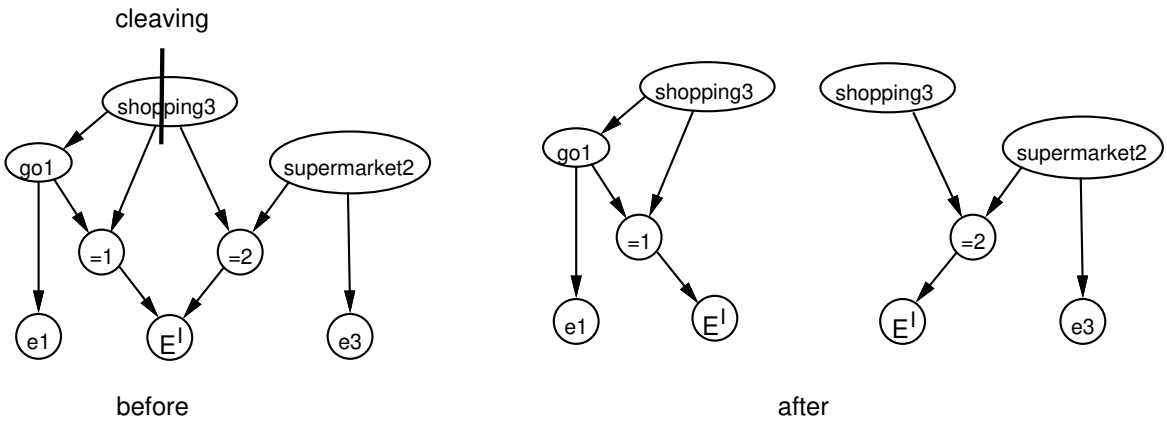


FIG. 5.6. Cleaving a vertebrate Bayesian network.

- *the cleaved node n appears in both halves*
- *the left half includes the evidence node* $e_1$*, all* inst *nodes* $i_1$ *through* $i_n$ *and all arcs between them, and similarly for the right half*
- *equality nodes with both parents among* inst *nodes* $i_1$ *through* $i_n$ *and all arcs incident to them are in the left half, and similarly for the right*

*See Figure 5.6.*

LEMMA 5.2. *A spine can be cleaved into two halves* $H_1$ *and* $H_2$ *at any* inst *node, such that the spinal contribution of the whole graph is given by*

$$SC(V) = SC(H_1) \times SC(H_2)/p(n) \tag{41}$$

*where n is the node at which the graph is cleaved.*

*Proof.*

If there were no isa statements, the proof would be straightforward, since there are no arcs between nodes in the left and right sides of the cleaved network. We are concerned solely with the spinal contribution, we can ignore the terms $p(e_2| =_1 , =_2, ...)$ on both sides. The cleaved node appears to be a peak exactly one more time than it really is. If it really is a peak in the entire network, then it appears to be one twice, once on each side. See, for example, shopping3 in figure 5.6. If the cleaved node is not a global peak, then it appears to be so on one half of the cleaved network. If we cleaved at go1 instead, it would be a peak on the left side in our example. In order to avoid over-counting this node we always divide by its prior.

Isa statements mean that one side of the cleaved graph may misjudge the cleaved node's relevant type. There are two possible cases, depending on whether the node is a global peak or not. In the former case, our $SC(V)$ formula will have no terms of the form

$$p(i_1|...) \tag{42}$$

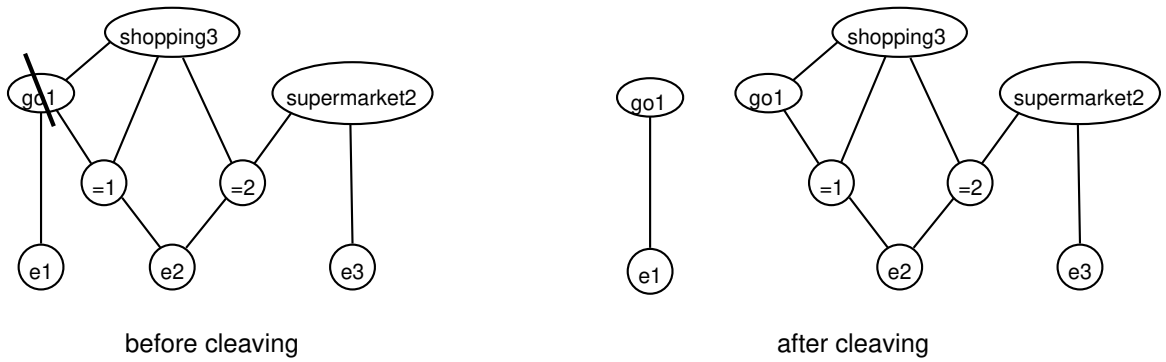


FIG. 5.7. Cleaving a vertebrate Bayesian network at a non-peak node.

where $i_1$ is the cleaved node, since this is a global peak. We need only worry about terms of the form

$$p(X|i_1, ...) \tag{43}$$

where $X$ may be any other node. Recall that the genuine relevant type of a node may only be *more* specific than the type currently considered for that node. We have already assumed for each node type that the distribution for a more specific type will be the same as for the general type. More formally, no matter what type $X$ is

$$p(X|i_1 = s_1, ...) = p(X|i_1 = s_2, ...) \tag{44}$$

where $s_2$ is a type more specific than $s_1$. This suffices for the peak case.

If the node is not a global peak, then one side or the other will believe it is anyway. There are two possibilities: the side which knows the node is not a peak also knows the node's relevant type, and the other side knows neither, or the side which knows the node is not a peak does not know its relevant type. To give an example: suppose we cleaved our graph in figure 5.6 at go1. The left side now believes go1 is a peak, whereas the right side knows it is not. One side or the other may be wrong about its relevant type, but not both.

Let us suppose that the right-hand (non-peak) side knows the cleaved node's genuine relevant type. Let $i_1$ be the cleaved node, $s_1$ the genuine relevant type, and $s_2$ the node at which the paths meet (which is also the type erroneously believed to be the relevant type on the left side). Then on the left-hand side we will have calculated a multiplicative term $p(s_2)$ for the peak node, which cancels out when we divide by the prior of the current node, $s_2$. We may also have terms for an equality node or inst node connected to $i_1$, say $p(X|i_1 = s_2, ...)$. As before, we assume in our distributions that

$$p(X|i_1 = s_1, ...) = p(X|i_1 = s_2, ...) \tag{45}$$

so terms on the left side do not change. Nothing changes for the right half-path, since it already has the correct relevant type, and no structural changes are made over there. Hence, we correctly calculate $SC(V)$ for the entire network.

In the final case, the left-hand (peak) side knows the relevant type of the cleaved node. Using the same variables, what we will have calculated on the peak side will be

$$p(i_1 = s_1) \tag{46}$$

On the non-peak side we will have

$$p(i_1 = s_2 | i_2) \tag{47}$$

It is possible that $s_2$ is more specific than the type specified by the existential arc, say $s_3$. It is easiest to imagine this as cleaving a path in the middle of a chain of isa statements between role statements. If this is the case, then this reduces to

$$p(s_2)/p(s_3) \tag{48}$$

as we showed for our calculations for crossing role statements, in the case in which an existential arc is added. If $s_2$ is not more specific than $s_3$, i.e. $p(s_2) = p(s_3)$, then our multiplier is simply 1.0.

Our final calculation should be

$$p(i_1 = s_1 | i_2) = p(s_1)/p(s_3) \tag{49}$$

which will be produced if we multiply the two together and divide by $p(s_1)$, as directed. Q.E.D. □

Lemma 5.2 means that we can track the spinal contribution incrementally as we extend a half-path. When the measure drops below a threshold, $T$, we can cut safely cut off marker-passing; that is, we will miss no complete paths whose measure would be above $T^2$. Currently we have $T$ set at 30. We have arrived at this value through experimentation with a set of paths generated from a set of stories which we use for debugging and tuning our system. Generally, there is a large gap (approximately a factor of 10) between the spinal contribution of those paths which have the right explanations and those which do not. While our system does rely on the prior probabilities for our schema, this gap suggests that we can get by with priors that are only approximately correct.

# 6 Results

We have employed our marker-passer in the Wimp3 story understanding system (Goldman [1991]) to find explanatory plans. The results quoted in table 6.1 are those obtained both on Wimp3's debugging corpus of 25 1-to-4 line "stories" and on its evaluation corpus[7] of 25 stories. We counted paths at four points in the flow of control. First, we counted paths which left the marker passer. As described above, we integrated the probability valuation of paths into the marker-passing mechanism itself, using it to control the spread of marks. This makes it impossible to

[7] The debugging corpus is used for testing and tuning of parameters. The evaluation corpus is for evaluation only, and is therefore a cleaner test, in some sense.

TABLE 6.1. Results Summary

| corpus | paths reported | paths asserted | paths evaluated | paths approved |
|---|---|---|---|---|
| debug | 985 | 115 | 83 | 78 |
| test | 747 | 109 | 68 | 64 |

estimate how many paths were eliminated due to low probability values. Likewise, we cannot say how many paths were eliminated by employing DFA's to prohibit generation of invalid paths. We can only give the total number of paths returned, with the invalid and low probability paths already weeded out. Second, we counted paths which were "asserted", i.e., used for forward-chaining and Bayesian network construction. These are paths which passed various secondary filters reported in (Carroll & Charniak [1989]). Third, we counted paths whose resulting statements were actually evaluated using our Bayesian network evaluation mechanism. Some paths could be eliminated without evaluation, as we will describe shortly. Finally, we counted those paths which we approved after evaluation; for a path to be approved the posterior probability of the suggested plans given the evidence had to be 1,000 times higher than the prior probability for the plans. The ratio of approved paths to asserted paths was 68% for the debugging corpus, and 59% for the test corpus, a significant improvement on the 10% good to bad path ratio reported earlier, (Charniak [1986b]; Norvig [1987a]) and strong support for our claim that the key to the viability of marker-passing is the supporting evidence, our $E^I$.

Our analysis suggested one more improvement which could be made. As we commented earlier, the marker passer's probability calculations are an upper bound, based on the assumption that there is evidence in favor of the path (other than the nodes at either end). That is, after the marker passer produces an acceptable path $P$, Wimp3 constructs a Bayesian network which includes $RS(P)$. Among the paths which were not approved, we found that there was often no evidence supporting some of the statements in $RS(P)$. While we cannot determine whether evidence is missing before network construction, we *can* do so before network evaluation. While the former takes time linear in the size of the network (making reasonable assumptions about the process), the latter, in general, takes exponential time (and is NP-hard). Indeed, network evaluation accounts for approximately 90% of Wimp3's running time, and thus the only really bad paths are those which cannot be removed before network evaluation and are not approved afterwards. Weeding out paths with no evidence accounts for the difference between paths which were asserted–used for network construction–and paths which were evaluated. A combined total of 151 paths had to be evaluated by Wimp3, and, of these, all but 9 were good, for a percentage of about 94%. This leads us to extend our claim to say that: *A domain is suitable for search by means of marker-passing only if there will usually be supporting evidence for paths returned by the marker-passer, or paths without evidence can be cheaply identified.*

## References


Allen, James F. [1979], "A Plan-Based Approach to Speech Act Recognition," University of Toronto, Ph.D. Thesis.

Alterman, Richard. [1985], "A dictionary based on concept coherence," *Artificial Intelligence* 25, 153–186.

Calistri, Randall J. [1989], "Classifying and Detecting Plan-Based Misconceptions," TR CS-89-16 Brown University Department of Computer Science.

Carroll, Glenn & Charniak, Eugene [1989], "Finding plans with a marker-passer," *Proceedings of the Plan-recognition Workshop, 1989*.

Charniak, Eugene [1983], "Passing markers: A theory of contextual influence in language comprehension," *Cognitive Science* 7.

Charniak, Eugene [1986a], "A Single-Semantic-Process Theory of Parsing," Department of Computer Science, Brown University, Technical Report.

Charniak, Eugene [1986b], "A neat theory of marker passing," *AAAI-86*.

Charniak, Eugene & Carroll, Glenn [1991], "A Probabilistic Analysis of Marker-Passing Techniques for Plan-Recognition," Department of Computer Science, Brown University, Technical Report.

Charniak, Eugene & Goldman, Robert P. [1989], "A semantics for probabilistic quantifier-free first-order lanuages, with particular application to story understanding," *IJCAI-89*.

Collins, Alan & Quillian, M. Ross [1969], "Retrieval time from semantic memory," *Journal of Verbal Learning and Verbal Behavior* 8, 240–248.

Cooper, Gregory F. [1987], "Probabilistic Inference Using Belief Networks is NP-hard," Stanford University, Technical Report KSL-87-27 Medical Computer Science Group.

Goldman, Robert [1991], "A Probabilistic Approach to Language Understanding," Department of Computer Science, Brown University, Technical Report.

Hendler, James A. [1988], *Integrated Marker-Passing and Problem-Solving*, Lawrence Erlbaum Associates, Hillsdale New Jersey.

Norvig, Peter [1987a], "Unified Theory of Inference for Text Understanding," Computer Science Division, University of California Berkeley, Report No. 87/339.

Norvig, Peter [1987b], "Inference in text understanding," *Proceedings of the Seventh National Conference on Artificial Intelligence*.

Pearl, Judea [1988], *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Los Altos, Calf..