Counterfactual Reasoning with Direct Models

Mark Derthick
Department of Computer Science
Carnegie-Mellon University

Abstract

Most of the effort AI has put into common sense reasoning has involved inference by sequential rule application. This approach is most effective in well characterized domains where any valid chain of inference from a set of observations leads to an acceptable interpretation. In more realistic cases where there are multiple consistent interpretations that are not equally good, or where there are no consistent interpretations, it seems more natural to choose the best alternative based on the interpretations themselves rather than the chains of inference used to derive them. μKLONE is a connectionist network which uses simulated annealing to search the space of interpretations, or models. Inconsistent theories lead to generation of models which come as close as possible to satisfying all of the axioms, so counterfactual reasoning can be accomplished by the same mechanism as factual reasoning. An example involving conflicting information is presented for which μKLONE finds an intuitively plausible interpretation.

I. Introduction

The model based approach described below embodies three key ideas taken from other work on common sense reasoning. In a possible worlds semantics a counterfactual implication $A > B$ is true if B holds in the most plausible world where A is true [Ginsberg, 1986]. The hard part of reasoning this way is finding the appropriate world. This is the task of constructing a vivid knowledge base examined by [Levesque, 1986], which suggests using defaults and other heuristics. [Johnson-Laird, 1983] also finds one or a few models of a scenario, and improves the efficiency of the search by using models whose structure is analogous to the problem domain. Such models are called direct [Hayes, 1985]. For a system that can find plausible models, this kind of reasoning is easier than ordinary implication, which would require checking whether B holds in all consistent models where A does. μKLONE finds plausible analog models as determined by a continuous evaluation function which maximizes the number of assertions in the knowledge base (KB) that hold in the model. All assertions can therefore be treated as defaults.

Expert systems such as ISIS [Fox, 1983] also use real valued constraints to guide the search for a good solution, but the search is over paths to solutions rather than the solutions themselves. The disadvantage is that the search control knowledge does not give a process independent semantics for characterizing the correct solution.

Previous spreading activation models have been less expressive than μKLONE. In finding similarities between words, the algorithm of [Quillian, 1968] spreads activation along all types of links identically. [Shastri, 1985] treats concepts as atomic propositions rather than predicates, and can simultaneously consider only a single token of any type.

The remainder of this paper presents an example of counterfactual reasoning, describes how it can be accomplished within the model based framework, and gives a detailed explanation of how this kind of reasoning can be implemented on a connectionist architecture.

II. An Example of Plausible Inference

At a Newport bar, June meets Ted, who is dressed like a sailor. Ted is excited about the approaching television season, and tells June how the schedule reflects the evolution of TV programming. June concludes that Ted is a sailor, and that he must spend a lot of time becalmed to be so interested in television. The next week she sees Ted's picture in the newspaper with the caption "Millionaire Playboy Ted Turner." June concludes that sailing must be only a hobby of Ted's, since millionaires don't have manual labor jobs but often have ostentatious pastimes. They also are unlikely to spend all day watching TV, so perhaps Ted has a job as a high level television executive.

Appendix I contains the full μKLONE description used to approximate June's beliefs before seeing the newspaper. In the knowledge base Ted is asserted to be a (professional) sailor and it is asserted that one of Ted's interests is a television-related activity. Sailors are defined to be people one of whose jobs is sailing. Millionaire-playboys are defined to be people who have an expensive hobby, and it is asserted that all of their jobs are armchair-activities, and they must have at least one job.

The following is an abstract description of the way the final model is chosen by μKLONE. The description is abstract because it refers to high level rule-like causal relations which do not correspond in a simple way to the changing relations among unit states, determined by the simulated annealing search algorithm [Smolensky, 1986].

When a μKLONE network constructed from the knowledge base is asked "If Ted were a millionaire-playboy, what

would his job and hobby be?"[1] the system must try to reconcile being a millionaire-playboy with its previous knowledge about Ted, that he is a sailor and is interested in TV. The counterfactual premise conflicts with the knowledge base because sailing is a vigorous-activity, and the jobs of millionaire-playboys must be armchair-activities. The initial impact of this conflict on the selection of a model is that sailing is likely to still be one of Ted's interests, but perhaps not his job. Since millionaire-playboys must have expensive hobbies and only two activities known to require expensive equipment are in the KB, flying and sailing are the most likely candidates. Sailing is chosen because it is already thought to be an interest. The plausible substitution that sailing is Ted's job rather than his hobby is made because HAS-JOB and HAS-HOBBY are both subsumed by HAS-INTEREST, making it is relatively easy to slip between them.

Millionaire-playboys must have a job that is an armchair activity and a profitable activity. Both TV-network-management and Corporate-Raiding fit this category, but the former is chosen because it is known that Ted is interested in television. TV-acting is rejected because it is not an armchair-activity, and TV-watching is rejected because it is not a profitable-activity.

If the knowledge base did not specify that millionaire-playboys had expensive hobbies, the bias towards having sailing as an interest would not be sufficient for its being picked out as a hobby. Similarly, if millionaire-playboys did not have to have jobs none would be picked out. And if the query had been simply "What are Ted's job and hobby?" no contradictory information would have been introduced. The answer, that sailing is Ted's job and he has no hobbies, would be constructed from knowledge in the KB alone.

III. A Model Based Approach

μKLONE answers wh- questions of the form $A > B(x)$. A is a set of propositions and $B(x)$ is a set of proposition templates in which either predicate symbols or individual constants are left out, to be filled in by the system. The system searches for the most plausible model in which A holds, and answers by filling in the missing predicates and individuals in $B(x)$. Since the response is filling in rather than assenting, the yes/no questions of [Ginsberg, 1986] must be recast into wh- form: "If Ted were a millionaire-playboy, what would his job be?" rather than "If Ted were a millionaire-playboy, would he be a sailor?"

μKLONE is able to make very fine grained distinctions between models because the definitions and assertions in the KB are decomposed into many constraints among *micro-features* [Hinton, 1981]. Axioms which mention defined predicates are expanded by replacing the predicate with its definition, and all axioms with conjunctions on the right hand side are broken up into multiple axioms. For instance, the similarity of HAS-JOB and HAS-HOBBY necessary for answering the example query is evidenced in the micro-features they both excite: *<domain animal>*, *<range activity>* and *<primitive class has-interest>*. In addition, HAS-JOB has the micro-features *<range profitable-activity>* and *<primitive class has-job>* while HAS-HOBBY has the micro-feature *<primitive class has-hobby>*.

Because defined concepts are expanded out before the connectionist network is built, definitional knowledge is not represented explicitly. Instead, it is represented *directly*, in the relationships between patterns. Direct representations [Hayes, 1985] have properties isomorphic to formal properties of the entities they represent. μKLONE directly represents explicitly defined subsumption relations among both concepts and roles, and subset relations among sets of role fillers (see section IV B.). Thus it is impossible for a μKLONE network to represent that, for instance, Ted is a MILLIONAIRE-PLAYBOY but not a PERSON. In addition to making definitions (as distinct from assertions) non-defeasible, it improves the efficiency of the system because certain contradictory models are eliminated from the search space.

The query language is highly constrained in that all predicates in both the premise and the consequent must be about the same individual (Ted in the example). This way inhibition can be hard-wired between, for instance, the value restriction that all jobs be armchair activities, that sailing is a job, and that sailing is a vigorous activity. This would be inappropriate if the restriction applied to one individual, but another was the sailor. Hard-wiring units to enforce very specific constraints produces a simple network topology, maximizes the independence of the units, and increases the effectiveness of parallelism.

The constraints implement the model evaluation function, and are of two main types: each axiom has some associated cost for violation; and there is a penalty for including tuples in the extension of a predicate. Each axiom or tuple contributes additively to the evaluation function independently of which other axioms hold. Using this cost function, μKLONE's simulated annealing search algorithm generally finds a good approximate solution early, which it then refines. In simulated annealing, all constraints are continually considered and contribute in accordance with their strength. The importance of satisfying constraints of any given strength is gradually raised. This way, more important constraints are generally satisfied first, except when contradicted by a number of less important ones.

An advantage of the annealing search is that models are not evaluated entirely in isolation. At a given moment, the state of the system may represent a superposition of models. To the extent that two models overlap, they reinforce one another, so that models incorporating propositions which hold in the greatest number of competing interpretations are preferred. Even if the "wrong" model is chosen, this maximizes the probability that individual beliefs are correct. This heuristic is necessary for correctly answering the example query. Models in which flying is Ted's interest are evaluated as plausible only when this is necessary to fulfill the hobby requirement of playboys. There are two groups of plausible models in which Ted's interest is sailing: those in which it fulfills the hobby

[1] μKLONE uses a formal query language, but English paraphrases are used in this paper. The formal version of this query is given in Appendix I.

requirement of playboys and those in which it fulfills the job requirement of sailors. Therefore, more minimal models in which sailing is Ted's interest are plausible. This biases the evaluation toward Ted's hobby being sailing. Unfortunately there is no process independent semantics to model the effect of this heuristic.

IV. System Implementation

A. Architecture

A μKLONE KB is compiled into a connectionist network of very simple processors. Each processor has an graded activity level in [0, 1] which it asynchronously updates based on the activity levels of its neighbors. Vectors of activity levels have meaningful interpretations in terms of concepts, roles, and individuals. The architecture of the network supports direct representations of an individual and a set of other individuals to which it is related, together with the relations involved. There are five important modules: The **subject** module represents the individual the query is about; the **subject-type** module represents the subject's type, which is a concept; the **role-fillers** module represents the set of individual/role pairs for all individuals directly related to the subject; the **role-filler-type-restrictions** module represents the set of value restrictions imposed on role fillers by the concept represented in the **subject-type** module; the **role-filler-types** module represents the type of each individual in the **role-fillers** module, whether or not it is currently filling any roles. The communication pathways between modules are shown in figure 1. The constraints among the **role-filler-types**, **role-fillers**, and **role-filler-type-restrictions** modules are too complex to be captured directly with pairwise links, so there is another module to mediate the interaction.[2] There are also modules which do not participate in the direct representation, but serve only for input and output. The shape of the modules in figure 1 is a clue to what is represented: the modules with a single row of units represent a single entity, either a concept in the **subject-type** module, or an individual in the **subject** module. The modules with multiple rows represent a set of pairs of entities: the **role-fillers** module represents sets of individual/role pairs; the **role-filler-type-restrictions** module represents sets of concept/role pairs; the **role-filler-types** module represents sets of individual/concept pairs.

B. Representations

Sets of Individuals Each individual maps to a unique bit in patterns representing sets of individuals. For the example domain there are seven individuals, so the patterns have seven bits. Set containment, an important relation in μKLONE, is represented directly. In figure 2 it is evident that {TV-Watching Flying Sailing} contains {TV-Watching Flying} because the pattern for the former contains that for the latter.

[2] I plan to use a more powerful connectionist model in the future which will not require this module.

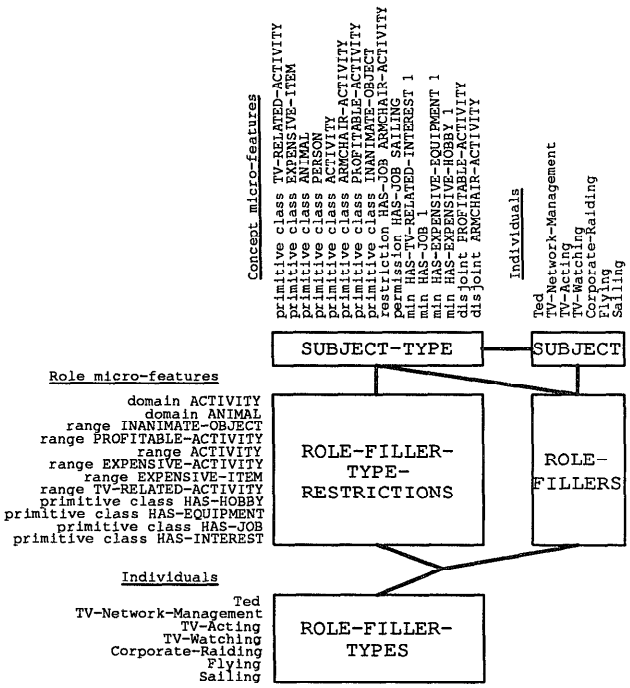

Figure 1: μKLONE has five important modules. Those modules which directly constrain one another are connected. The meaning of each unit can be deduced from the printed descriptions. For the **subject** and **subject-type** modules, the meaning is the description above the unit. For the other modules, the meaning involves the conjunction of the descriptions in the unit's row and column. For instance, the top left unit in the **role-fillers** module means that Ted is filling a role whose domain is ACTIVITY.

Roles The KB is examined to find all the properties which are either used to define a role or asserted to be true of one. Each role corresponds to a subset of these properties, or *micro-features*. Role patterns have one bit for each micro-feature. For the example domain there are 12 role micro-features, among which are *<range inanimate-object>* and *<primitive class has-interest>*.

For roles, the relation of subsumption is directly represented as the relation of set containment of patterns. For example, the pattern representing the role HAS-HOBBY is the set of micro-features {*<primitive class has-hobby>*} while HAS-EXPENSIVE-HOBBY has the micro-features {*<primitive class has-hobby>* *<range expensive-activity>*}. HAS-HOBBY subsumes HAS-EXPENSIVE-HOBBY, and its pattern is contained by the pattern for HAS-EXPENSIVE-HOBBY, as illustrated in figure 2. [Hinton, 1981] originated this technique, which results in very efficient representations. Information relevant to a micro-feature need only be attached locally to a single unit, and yet it is used by all concepts having the property associated with the micro-feature. This avoids the dilemma facing conventional semantic network implementations: either cache information together with each concept that needs it, requiring duplication of information, or search up the inheritance hierarchy each time the information is needed.

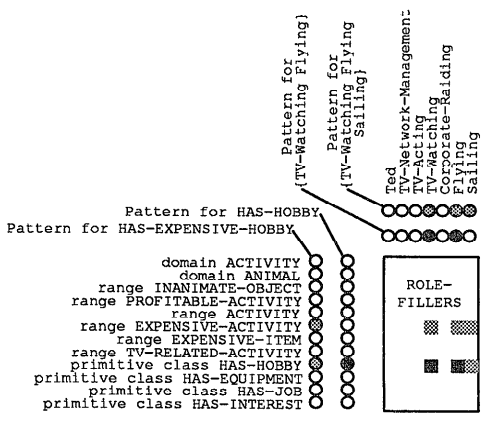

Figure 2: Two examples illustrating how pairs of patterns are conjunctively coded. The first example (black) combines the two bit pattern for {TV-Watching Flying} with the one bit pattern for HAS-HOBBY, producing a 2×1 bit pattern in the **role-fillers** module representing the fact that {TV-Watching Flying} is the set of fillers of the HAS-HOBBY role. The second example (gray) combines the three bit pattern for {TV-Watching Flying Sailing} with the two bit pattern for HAS-EXPENSIVE-HOBBY, producing a six bit pattern in the **role-fillers** module. Since the former value permission necessarily follows from the latter, its two bit pattern is contained by the six bit pattern for the latter.

Concepts Concepts also use a micro-feature representation, and each micro-feature corresponds to a clause in a DEFCONCEPT or ASSERT-CONCEPT statement in the KB. For the example domain there are 16 concept micro-features, such as *<primitive class person>*, *<minimum has-job 1>*, and *<restriction has-job armchair-activity>*. This makes subsumption direct for concepts as well.

Pair Representations Figure 2 illustrates the technique used for representing pairs of patterns. The size of the module required to represent a pair of entities is the product of the pattern lengths of the two entities. To represent the pairing *AB*, for each *i* and *j* the unit at coordinates *i, j* is turned on if and only if the *i*th bit in the pattern for *A* is on and the *j*th bit in the pattern for *B* is on. To store multiple pairs, the patterns for each pair are superimposed. This is a variation on the technique of *coarse coding* [Hinton *et al.*, 1986].

The direct relation between subsumption and set containment of patterns carries over to the modules representing pairs as well. Figure 2 illustrates this for the the **role-fillers** module. If {TV-watching Flying Sailing} is the set of fillers of the HAS-EXPENSIVE-HOBBY role, it automatically becomes the case that, for example, each of {TV-watching Flying} is filling the HAS-HOBBY role. This works because the implications of set *A* filling role *R* involve subsets of *A* and subsumers of *R*, either of which have patterns with fewer units on.

C. Constraint Types

There are five types of constraints that must hold between modules (see figure 1), and five more types that must hold within modules. One illustrative constraint of each type is described here. See [Derthick, 1987] for the complete description.

Constraints Within Subject-Type Module These constraints ensure that a coherent concept is represented. If the pattern for a concept is present, then the patterns for all concepts asserted to subsume it must be present, and the patterns for all concepts asserted to be disjoint from it must be absent. This can usually be done with pairwise links among the units in a group. For example, an inhibitory link between the *<primitive class animal>* and *<primitive class activity>* micro-features expresses that ANIMAL and ACTIVITY are disjoint. When more than one bit is required for discrimination, extra units are created to express the constraint. With the more powerful connectionist architecture mentioned in section A., these extra units will not be required.

Constraints among the Role-Fillers, Role-Filler-Types, and Role-Filler-Type-Restrictions Modules These constraints ensure that all type restrictions on a role are satisfied by fillers of the role, and are the most complicated part of μKLONE. A **role-filler-types** micro-feature represents the conjunction of an individual and a concept micro-feature. The former determines the relevant column in the **role-fillers** module, and the latter determines the relevant column in the **role-filler-type-restrictions** module. If the individual is filling a role at least as specific as the one to which the value restriction applies, then the **role-filler-types** micro-feature must come on. This condition can be determined by first ORing the two columns together, and then ANDing the resulting column. If the result is true, then the value restriction applies to this individual.

D. Size and Speed of the Connectionist Network

When the network building algorithm is given the KB of Appendix I, a Hopfield and Tank network [Hopfield, 1984] with 2531 units and 16,959 connections results. Empirically it was found that an annealing schedule exponentially increasing the gain for 500 time steps was sufficient for answering the queries mentioned above. (One time step involves updating the state of each unit.) This takes about ten minutes of CPU time on a Symbolics 3600.

The number of units scales as the third power of the size of the knowledge base, and the number of links scales as the fourth power. With the envisioned, more powerful connectionist model, this will be reduced to the second power and third power, respectively. The only known theoretical bound on the number of time steps in the annealing schedule required for good performance is exponential, however if μKLONE generally produces networks with smooth energy surfaces the results may be much better. Only two KBs have been compared to date: a 34% increase in KB size required a 16% increase in the annealing schedule length.

V. Details of Reasoning Process

A more detailed description of the inference process outlined in section II. can now be given. The input/output modules excite the Ted unit in the **subject** module, which in turn excites SAILOR in the **subject-type** module. Meanwhile, MILLIONAIRE-PLAYBOY also receives external excitation. Although incompatible, these concepts were not explicitly made disjoint in the KB, and so no links were built within the **subject-type** module to inhibit the combination of the two patterns. In μKLONE, relationships between concepts are maintained indirectly through the effect of each on the model anyway, so there is no need to precompute them.

The SAILOR pattern contains the *<permission has-job sailing>* micro-feature, which excites the pattern for HAS-JOB in the sailing column of the **role-fillers** module. The MILLIONAIRE-PLAYBOY pattern contains micro-features for *<minimum has-expensive-hobby 1>* and *<restriction has-job armchair-activity>*. The latter excites a pattern in the **role-filler-type-restrictions** module. At this point, the micro-feature in the **role-filler-types** module for *<sailing is an armchair-activity>* has a problem. On the one hand, sailing is known to be a VIGOROUS-ACTIVITY, so the *<sailing is a vigorous-activity>* micro-feature in the **role-filler-types** module has a positive bias and is active. Since VIGOROUS-ACTIVITY and ARMCHAIR-ACTIVITY are disjoint, the *<sailing is an armchair-activity>* micro-feature is inhibited. But the relevant columns in the **role-fillers** module and the **role-filler-type-restrictions** module indicate that sailing must indeed be an armchair activity.

There is no way to satisfy all the constraints simultaneously. The system's choice depends on the relative strengths of the constraints, which are free parameters chosen by the experimenter. Logically they are part of the KB, but as of now they are constants hidden in Lisp code. In a connectionist system these strengths can, in principle, be learned automatically. I have adjusted the strengths of the links so the constraint from the *<permission has-job sailing>* micro-feature in the **subject-type** module to the "sailing is a HAS-JOB" pattern in the **role-fillers** module is weakest. Therefore, the pattern for has-job in the sailing column of the **role-fillers** module is not sustainable. From this point, the choice of Sailing as Ted's hobby and TV-Network-Management as his job result from the similarity of their patterns, independent of any constraint strengths. The unit that differentiates HAS-JOB from HAS-INTEREST is forced off, but the remaining activation of the "sailing is a HAS-INTEREST" pattern leads to the eventual choice of sailing as Ted's hobby. Space limitations prevent a detailed description of the selection of TV-Network-Management as Ted's job.

VI. Conclusion

This paper introduced a novel semantics for question answering based on finding an explicit partial model which plausibly reconciles long term knowledge with situation specific information. The Ted Turner example demonstrates that this method is effective for a non-trivial problem involving counterfactual reasoning. Finding a plausible model is well suited to parallel constraint satisfaction using a special purpose architecture. The structure of the solution is constant so the models can take advantage of direct representations to reduce the search space. Explicitly represented constraints contribute independently to the evaluation function, so parallelism can be used effectively with units connected heterogeneously to enforce particular constraints. Representing concepts and roles as sets of micro-features results in many more constraints in the connectionist network than there are statements in the KB. This, along with the continuous activation levels of units, increases the smoothness of the evaluation function so that simulated annealing is a good search technique.

Future work will examine: knowledge bases of many sizes to better determine empirically how the search time scales; learning as an alternative to setting weights by hand; and the possibility of giving a semantics to the weights in terms of probabilities of models.

Acknowledgments

Geoff Hinton and Dave Touretzky have been very helpful with the design of μKLONE and the preparation of this paper. Discussions with Ron Brachman resulted in a more coherent KB language. I thank Oren Etzioni, Craig Knoblock, David Plaut, Roni Rosenfeld, David Steier, and the anonymous referees for providing useful comments. This research is supported by NSF grants IST-8520359 and IST-8516330, and an ONR Graduate Fellowship.

Appendix I Formal Domain Definition

The following input was used by the network building algorithm to produce a Hopfield and Tank network for answering queries. The syntax derives from that of KL2's definition language [Vilain, 1985]. Three ontological categories are used: *concepts* are classes of *individuals*. *Roles* are classes of two-place relations between individuals. DEFCONCEPT and DEFROLE statements normally give necessary and sufficient conditions for determining whether an individual instantiates a concept or whether an ordered pair of individuals instantiates a role. Alternatively, if the language is not powerful enough to provide sufficient conditions for recognizing membership, a concept or role can be defined to be *primitive*. In this case, the extension of the concept or role must be explicitly declared using INSTANTIATE-CONCEPT or INSTANTIATE-ROLE statements. Conditions which necessarily hold of instances of concepts or roles, but are not part of the recognition criteria are asserted with ASSERT-CONCEPT or ASSERT-ROLE statements.

```
(DEFCONCEPT Animal (PRIMITIVE))
    ;ANIMAL is a natural kind — you can't define it
(DEFCONCEPT Person (PRIMITIVE))
(ASSERT-CONCEPT Person (SPECIALIZES Animal))
    ;PERSONs always turn out to be ANIMALs
(DEFCONCEPT Millionaire-Playboy (SPECIALIZES Person)
    (SOME Has-Hobby Activity-Requiring-Expensive-Equipment))
    ;a PLAYBOY must have some HOBBY
```

```
    ;which is an ACTIVITY-REQUIRING-EXPENSIVE-EQUIPMENT
(ASSERT-CONCEPT Millionaire-Playboy (MIN Has-Job 1)
    ;a PLAYBOY must have a JOB
    (RESTRICTION Has-Job Armchair-Activity))
    ;a PLAYBOY's JOBs must be ARMCHAIR-ACTIVITYs
(DEFCONCEPT Sailor (SPECIALIZES Person)
    (PERMISSION Has-Job Sailing))
    ;sailing must be one of a SAILOR's JOBs
(DEFCONCEPT TV-Buff
    (SOME Has-Interest Television-Related-Activity))
(ASSERT-CONCEPT TV-Buff (SPECIALIZES Person))
(DEFCONCEPT Activity (PRIMITIVE))
(ASSERT-CONCEPT Activity
    (DISJOINT Inanimate-Object) (DISJOINT Animal))
(DEFCONCEPT Activity-Requiring-Expensive-Equipment
    (SPECIALIZES Activity)
    (SOME Has-Equipment Expensive-Item))
(DEFCONCEPT Armchair-Activity (PRIMITIVE))
(ASSERT-CONCEPT Armchair-Activity (SPECIALIZES Activity))
(DEFCONCEPT Vigorous-Activity (SPECIALIZES Activity)
    (DISJOINT Armchair-Activity))
(DEFCONCEPT Profitable-Activity (PRIMITIVE))
(ASSERT-CONCEPT Profitable-Activity (SPECIALIZES Activity))
(DEFCONCEPT UnProfitable-Activity
    (DISJOINT Profitable-Activity) (SPECIALIZES Activity))
(DEFCONCEPT Television-Related-Activity (PRIMITIVE))
(ASSERT-CONCEPT Television-Related-Activity
    (SPECIALIZES Activity))
(DEFCONCEPT Inanimate-Object (PRIMITIVE))
(ASSERT-CONCEPT Inanimate-Object (DISJOINT Animal))
(DEFCONCEPT Expensive-Item (PRIMITIVE))
(ASSERT-CONCEPT Expensive-Item
    (SPECIALIZES Inanimate-Object))

(DEFROLE Has-Interest (PRIMITIVE))
(ASSERT-ROLE Has-Interest (DOMAIN Animal)
    ;only ANIMALs can have INTERESTs
    (RANGE Activity))
    ;only ACTIVITYs can be INTERESTs
(DEFROLE Has-Job (PRIMITIVE))
(ASSERT-ROLE Has-Job (SPECIALIZES Has-Interest)
    (RANGE Profitable-Activity))
(DEFROLE Has-Hobby (PRIMITIVE))
(ASSERT-ROLE Has-Hobby (SPECIALIZES Has-Interest)
    (DISJOINT Has-Job))
(DEFROLE Has-Equipment (PRIMITIVE))
(ASSERT-ROLE Has-Equipment (DOMAIN Activity)
    (RANGE Inanimate-Object))

(INSTANTIATE-CONCEPT (Activity-Requiring-
    Expensive-Equipment Vigorous-Activity) Sailing)
(INSTANTIATE-CONCEPT
    Activity-Requiring-Expensive-Equipment Flying)
(INSTANTIATE-CONCEPT (Profitable-Activity Armchair-Activity)
    Corporate-Raiding)
(INSTANTIATE-CONCEPT (Television-Related-Activity
    Armchair-Activity UnProfitable-Activity) TV-Watching)
(INSTANTIATE-CONCEPT (Television-Related-Activity
    Vigorous-Activity Profitable-Activity) TV-Acting)
(INSTANTIATE-CONCEPT (Television-Related-Activity
    Armchair-Activity Profitable-Activity) TV-Network-Management)
(INSTANTIATE-CONCEPT (Sailor TV-Buff) Ted)
```

The query discussed in the paper, "If Ted were a millionaire-playboy, what would his job and hobby be?" is written:

```
((SUBJECT Ted)
 (SUBJECT-TYPE Millionaire-Playboy)
 (WITH (ROLE Has-Hobby) (FILLERS ?))
 (WITH (ROLE Has-Job) (FILLERS ?)))
```

References

[Derthick, 1987] Mark A. Derthick. *A Model Based Approach to Knowledge Representation and Reasoning.* Technical Report, CMU, Pittsburgh, PA, 1987. Forthcoming.

[Fox, 1983] Mark Fox. *Constraint-directed search: a case study of job-shop scheduling.* PhD thesis, CMU, 1983.

[Ginsberg, 1986] M. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–79, 1986.

[Hayes, 1985] P. J. Hayes. Some problems and non-problems in representation theory. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.

[Hinton, 1981] G. E. Hinton. Implementing semantic networks in parallel hardware. In *Parallel Models of Associative Memory*, Erlbaum, Hillsdale, NJ, 1981.

[Hinton *et al.*, 1986] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Distributed representations. In *Parallel distributed processing: Explorations in the microstructure of cognition.*, Bradford Books, Cambridge, MA, 1986.

[Hopfield, 1984] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences U.S.A.*, 81:3088–3092, May 1984.

[Johnson-Laird, 1983] Philip N. Johnson-Laird. *Mental Models.* Harvard University Press, 1983.

[Levesque, 1986] Hector J. Levesque. Making believers out of computers. *Artificial Intelligence*, 30:81–108, 1986.

[Quillian, 1968] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic information processing*, MIT Press, Cambridge, Mass, 1968.

[Shastri, 1985] Lokendra Shastri. *Evidential Reasoning in Semantic Networks: A Formal Theory and its Parallel Implementation.* PhD thesis, University of Rochester, September 1985. Available as TR 166.

[Smolensky, 1986] P. Smolensky. Foundations of harmony theory: cognitive dynamical systems and the subsymbolic theory of information processing. In *Parallel distributed processing: Explorations in the microstructure of cognition*, Bradford Books, Cambridge, MA, 1986.

[Vilain, 1985] M.B. Vilain. The restricted language architecture of a hybrid representation system. In *IJCAI-85*, Morgan Kaufmann, August 1985.