A Connectionist Architecture for Representing and Reasoning about Structured Knowledge

Mark Derthick
Carnegie-Mellon University

Abstract

μKLONE is the first sub-symbolic connectionist system for reasoning about high level knowledge to approach the representational power of current symbolic AI systems. The algorithm for building a network takes as input a knowledge base definition in a language very similar to that of KL2, which has previously been implemented only in Lisp. In μKLONE, a concept is more than a set of features: it is a complex of required and optional subparts filling well defined roles, each of which may have its own type restrictions. In addition to being able to use complex structured descriptions in its reasoning, μKLONE exhibits a facility for plausible inference due to its inherently parallel constraint satisfaction algorithm that is not shared by symbolic systems. This paper describes how the system answers a query that requires both of these characteristics. It is hoped that this is the beginnings of a response to (McDermott, 1986)'s challenge that connectionists should pay more attention to architectural issues and rely less on learning.

1. Previous Work

The semantic networks of (Quillian, 1968) were developed to model common sense reasoning using spreading activation. The degree to which concepts are related determines the influence they exert on one another. (Hinton, 1981) develops a more structured semantic network influenced by the theory of case frames in which concepts play one of three different roles in a frame.[1] In addition, the concepts themselves have a meaningful substructure: they are composed of *micro-features*. ELEPHANT is represented as the conjunction of the *<big>*, *<gray>*, and *<mammal>* micro-features. This gives the flexibility to reason about novel concepts that can be represented as conjunctions of existing micro-features. It also performs property inheritance automatically: since the ELEPHANT pattern contains the MAMMAL pattern, properties of mammals apply to elephants, unless they are overridden by the other micro-features of elephants.

(Shastri, 1985) is an extension of early semantic nets in a different direction. He retains the simple localist encoding of concepts, but adds a formal theory of evidential reasoning which can be encoded with connection strengths and thus efficiently executed. Structured knowledge can be represented using properties and values. Properties, like concepts, are definable in a high level language and are organized into an ISA hierarchy. In Shastri's system the formal elements, units, correspond to knowledge level entities like concepts. I will call such systems symbolic to distinguish them from systems like Hinton's in which the formal elements are at a lower level.

Although the micro-feature based representations of Hinton's system allow automatic inheritance and

[1]The theories of frames, schemas, and structured inheritance networks each have their own terminology. I will use "concept" uniformly instead of schema or frame and "role" instead of slot or case.

the representation of novel concepts, previously no system using them has had the expressive power of symbolic systems like Shastri's. μKLONE resembles current AI semantic networks in expressive power, particularly KL2 (Vilain, 1985), after which it was consciously modeled. It incorporates a third ontological category, individuals, in addition to concepts and roles. The language for defining a knowledge base (KB) makes it possible to specify *value restrictions* on the fillers of a role (for instance, the JOBs of MILLIONAIREs must be ARMCHAIR-ACTIVITIES) and *minimum restrictions* (for instance, a MILLIONAIRE must have at least one JOB). Appendix I gives a brief description of the KB language.

2. An Example Plausible Inference

2.1. Informal Description

The following scenario motivates the need to reason about conflicting beliefs. June has made some unwarranted conclusions about Ted from his conversation and appearance; specifically he is walking along a pier wearing a captain's hat and knowledgeably discussing the influence of independent producers on television programming. June assumes that Ted is a sailor, and that he must be deeply interested in television. The next week she sees Ted's picture in the newspaper with the caption "Millionaire Playboy Ted Turner." She now concludes that sailing is only a hobby of Ted's, since millionaires don't have jobs requiring vigorous activity but could afford an expensive sailboat. Millionaires usually have a job, so perhaps Ted is a high level television executive.

2.2. Formal Domain Description

The formal description of June's initial assumptions and general world knowledge is given in appendix I. In the knowledge base Ted is asserted to be a sailor, someone whose job is sailing, and a TV-Buff, someone who has an interest that is a television-related activity. A millionaire-playboy is defined to be a person with a hobby that is an activity requiring an expensive prop. Millionaire-playboys must have at least one job, and all their jobs must be armchair activities.

The primary query used as an example in this paper is given at the end of appendix I, and can be paraphrased "If Ted were a millionaire-playboy, what would his job and hobby be?" The system answers that sailing would be Ted's hobby and that TV-Network-Management would be his job. To demonstrate that the presupposition that Ted is a millionaire-playboy affects the reasoning process, the system can be asked "What are Ted's job and hobby?" In this case the answer is sailing is Ted's job, and he has no hobbies (TV-Watching is seen as an interest, but not necessarily either a hobby or a job).

2.3. High Level Description of Reasoning Process

The presupposition "If Ted were a millionaire-playboy" 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 is that sailing is likely to 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 props are in the KB, either flying or sailing are most likely to be chosen. Hobbies and jobs are both interests, so scenarios in which Ted's hobby is sailing are seen to be more plausible than those where it is flying.

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 more plausible because it is known that Ted is interested in television. The distractors TV-acting and TV-watching are in the knowledge base to demonstrate that all three factors (being TV-related, profitable, and an armchair-activity) are taken into account.

This kind of reasoning requires a structured treatment of properties. It is crucial to distinguish that vigorous-activity applies to sailing, that millionaire-playboy applies to Ted, and that Ted's relation to sailing, has-job, has its own properties, such as requiring sailing to be a profitable-activity. However, the structural properties alone are not sufficient to determine that sailing, rather than flying, is Ted's hobby. This decision is the result of residual activation of sailing as Ted's job. Default logic approaches (Reiter, 1980) are limited to binary decisions on the consistency of a theory, and have difficulty making choices like this based on relative plausibilities.

3. System Implementation

3.1. Architecture

One reason powerful non-symbolic systems are difficult to build is the problem of representing more than one concept at a time, a capability that is required to infer Ted's hobby. Since each concept is represented as a pattern over many units, it is not practical to represent very many. The best approach seems to be to build a small number of special purpose registers. This has the advantage over the localist approach that a concept can be represented in different registers in different contexts, and even in multiple places at the same time. The latter is especially difficult for localist systems.

At any one time, μKLONE can represent a single individual and information relevant to it. This includes the concepts it instantiates, any number of other individuals to which it is related, the concepts they instantiate, the relations involved, and value restrictions (VRs), value permissions (VPs), and minimum restrictions (MinRs) on the relations. The modular architecture shown in figure 1 reflects these distinctions. There are five modules primarily responsible for the representation, three auxilliary modules which mediate some of the inter-module constraints, a variable number of modules used for input and output, and a variable number of auxiliary I/O modules. There is no limit to the number of concepts, relations, or related individuals that can be represented in the five modules and used in reasoning, but the number simultaneously accessible to the user is limited by the number of I/O modules. To represent the example question of section 2.2, the pattern for Ted is clamped in the **subject** module, the pattern for MILLIONAIRE-PLAYBOY in a **subject-type-I/O** module, the pattern for HAS-JOB in one **role-I/O** module, and the pattern for HAS-HOBBY in a second **role-IO** module. After an annealing search (Kirkpatrick, 1983), the pattern for {sailing} is found in the first **role-fillers** I/O module and for {TV-Network-Management} in the second. Also represented internally are the VR that all the subject's jobs are armchair-activities, and the facts that sailing is a vigorous-activity, that TV-network-management is a profitable-activity and an armchair-activity, as well as (irrelevant) type information about the five individuals that have no known relations to Ted.

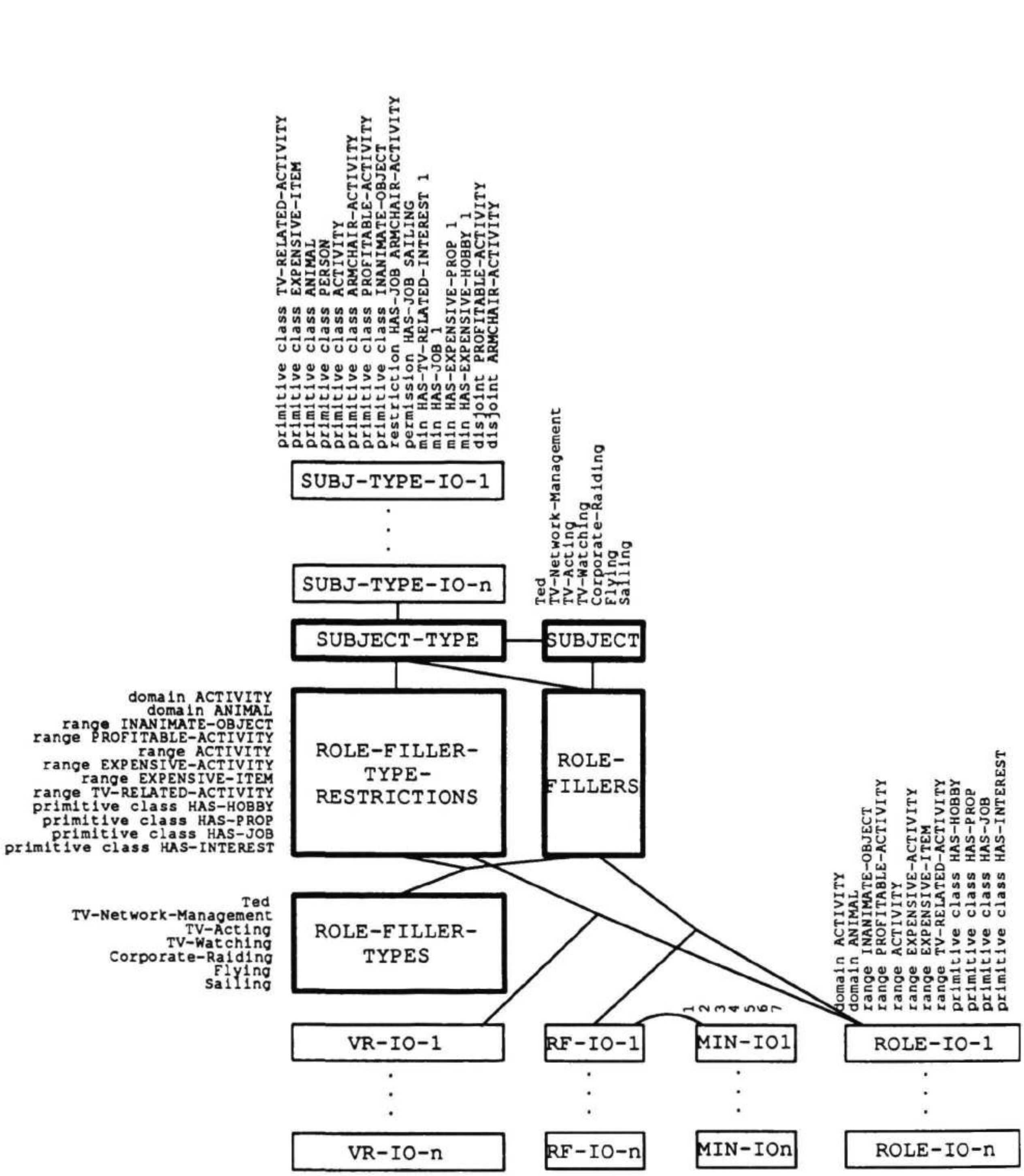

Figure 1: μKLONE has five important central modules (dark rectangles) and a variable number of subject-type, role, role-filler, minimum restriction, and value restriction IO modules. There are also auxiliary modules, which are not shown here. Those modules which directly constrain one another are connected. The meaning of each unit can be deduced from the printed descriptions. For modules with a single row of units, the meaning is the description in the unit's column. For modules with many rows, the meaning involves the conjunction of the descriptions in the 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.

3.2. Representations

3.2.1. Individuals

The seven individuals in the KB are described by the concepts they instantiate and the relations they participate in, but have no internal structure. A localist representation is used, with each bit corresponding to an individual. This makes it easy to represent sets of individuals; the pattern for {Ted Flying} is seven bits long, and exactly two of the bits are on.

3.2.2. Concepts

A concept pattern has one bit for every micro-feature derived from the KB. Every primitive concept has its own micro-feature; in addition there is a micro-feature for every distinct value restriction, value permission, minimum restriction, and disjointness restriction mentioned in the KB. The active bits in the pattern for SAILOR, for instance, are those corresponding to the *<primitive class person>* and *<permission has-job sailing>* micro-features. It is apparent that a very simple algorithm is sufficient to derive the required micro-features. The concept micro-features for the example domain are listed above the **subject-type** module in figure 1.

3.2.3. Roles

Roles are also represented as sets of micro-features. In this case the possible micro-features come from primitive roles, domain restrictions, and range restrictions, and are listed to the left of the **role-filler-type-restrictions** module in figure 1.

3.2.4. Minimum Restrictions

The MinR IO groups use a unary encoding of the minimum number of fillers a role must have: zero is represented by turning all bits off, two is represented by turning the "1" and "2" bits on, etc. The maximum number that needs to be represented is the number of individuals in the KB, in this case seven.

3.2.5. Derived Representations

Value restrictions are identified by the combination of the role that is being restricted and the concept which any fillers of that role must instantiate, for example HAS-JOB ARMCHAIR-ACTIVITY. Value permissions are a combination of a role and a set of individuals filling the role, such as HAS-JOB {sailing}. Binding two entities together is not straightforward in distributed connectionist systems. μKLONE uses a kind of *coarse coding*, a technique described in (Hinton, 1986a). To bind a set of individuals whose pattern has length I with a role whose pattern has length R requires a register with R rows of I units (see figure 2). For each i and j, the unit at coordinates i,j is turned on if and only if both the ith bit in the role pattern and the jth bit in the individual-set pattern are on.

Using coarse coding a number of pairs can be stored simultaneously by superimposing their patterns. Normally there is a limit to this number because patterns begin to interfere with one another, making them difficult to reconstruct. In μKLONE the patterns are constructed so that the combination of two or more patterns always produces a pattern whose meaning is a necessary consequence of the meanings of the constituent patterns. Interference is not degrading retrieval ability, it is actually performing inferences!

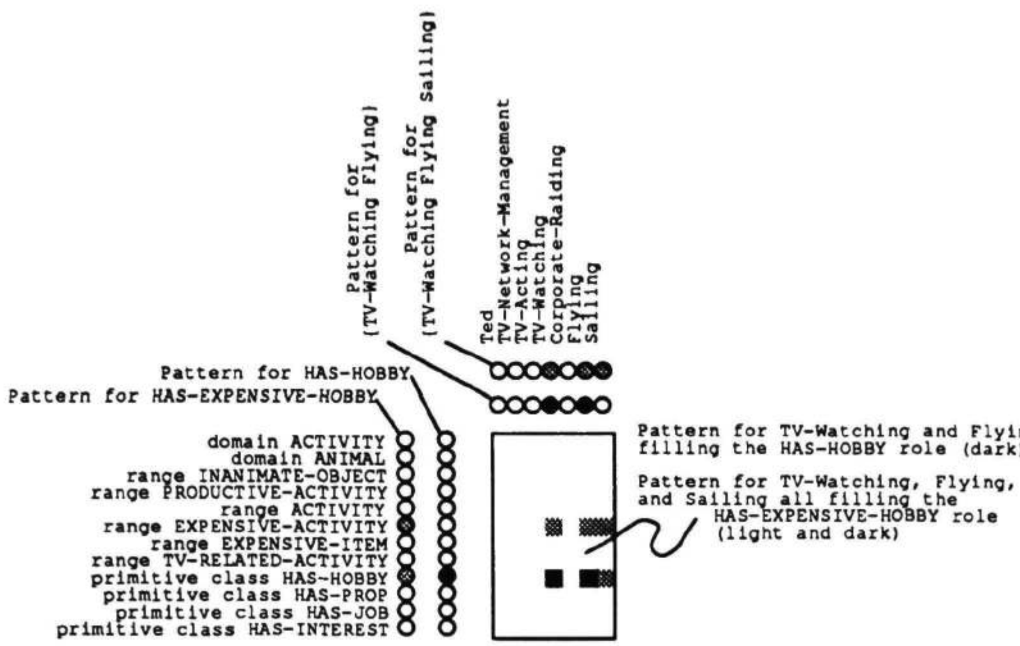

Figure 2: Two examples illustrating how pairs of patterns are conjunctively coded. The first example (black) combines the two bit pattern for (TV-Watching Flying) with the one bit pattern for HAS-HOBBY, producing a 2×1 bit pattern in the **role-fillers** module representing the fact that (TV-Watching 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 of the latter.

The reason for this can be inferred from figure 2, which illustrates that the pattern for the value permission HAS-EXPENSIVE-HOBBY (TV-Watching Flying Sailing) contains that for HAS-HOBBY (TV-Watching Flying). Not coincidently the latter VP is a necessary consequence of the former. In the more general case where neither pattern contains the other, their superposition will yield new VPs.

The pattern for a VP, *A*, contains that for VP *B* if and only if the role and individual patterns used to form *A* contain those forming *B*. This happens if *B*'s role is more general than *A*'s, or if *B*'s individual set is contained by *A*'s, which are just the conditions under which *B* is a necessary consequence of *A*. The weaker VP is always represented by fewer bits.

In the **role-filler-type-restrictions** module, multiple concept/role pairs are represented and the same kind of inference by interference is desirable. Unfortunately a value restriction entails other VRs with more general concepts or more *specific* roles. For example if all your hobbies are profitable-activities, then surely all your expensive-hobbies are activities. If the normal role representation is used, however, the weaker VR may be represented by *more* bits than the stronger one. To fix this, patterns for VRs are calculated as in figure 2 while pretending that role patterns are complemented. This way more specific roles have fewer bits on.

3.3. Constraints

All the constraints on the state of the network are currently implemented as symmetric, weighted, pairwise links between units, and biases on individual units. Units linked with a positive weight tend to excite each other, and units linked by negative weights inhibit each other. The μKLONE algorithm defines a network of units and links, which can then be run as either a Boltzmann Machine (Hinton, 1986b) or a Hopfield network (Hopfield, 1984). As discussed below, using only pairwise links is proving to be a problem, and I plan to use a more powerful model in the future.

3.3.1. Constraints on Coherent Concepts and Roles

There are three places where constraints are required to ensure that coherent concepts and roles are represented: the **subject-type** module must correctly describe the subject, each column in the **role-fillers** module must correctly describe some role being filled, and each row in the **role-filler-types** module must correctly describe an individual.[2]

Two types of assertions require links to ensure a coherent pattern is represented: "specializes" clauses require the pattern for the more general concept or role to be present if the more specific one is, and "disjoint" clauses forbid two concept or role patterns to be present at the same time. In the simple case of primitive concepts or roles where each pattern has only one bit on, a single implication link (see figure 3a) or a single inhibitory link (3c) suffices to effect the constraint. Otherwise, pairwise links cannot implement the constraint and, in the Boltzmann or Hopfield networks, extra units must be used (figure 3b and d). No links are required to implement *defined* specialization relations. The pattern for MILLIONAIRE-PLAYBOY, for instance, already contains that for PERSON.

3.3.2. Other Constraints

Describing all the constraints embedded in μKLONE in detail would be exceedingly tedious. Using micro-feature based representations, all the requirements for a plausible interpretation of the query reduce to fairly local relations between unit states of the type illustrated in figure 3. These logical relations are used to implement the following constraints:

- **Role Domain and Range Constraints:** A unit in the **role-fillers** module in the "sailing" column and the "domain ANIMAL" row should only come on if the *<primitive class animal>* micro-feature is active in the **subject-type** module. A unit in the "sailing" column and the "range ACTIVITY" row should only come on if the *<sailing is an activity>* micro-feature is active in the **role-filler-types** module.
- **Individual Type Constraints:** If Ted is asserted to be a SAILOR in the KB, and Ted is the subject, then the SAILOR pattern must be active in the **subject-type** module. The SAILOR pattern must be active in the row representing Ted's type in the **role-filler-types** module no matter what the subject is.
- **Value Permission Constraints:** If SAILOR is represented in the **subject-type** module, then the HAS-JOB pattern must be active in the "sailing" column of the **role-fillers** module. There are no "instantiate-role" assertions in the example KB, but if Ted had been asserted to have an interest in flying, then the Ted unit in the **subject** module would activate the HAS-INTEREST

[2]Because incompatible roles may be restricted to be filled with the same type of individual, columns in the **role-filler-type-restrictions** module do not necessarily represent coherent roles. Incompatible type restrictions may be imposed on a role, so rows in this group do not necessarily represent coherent concepts.

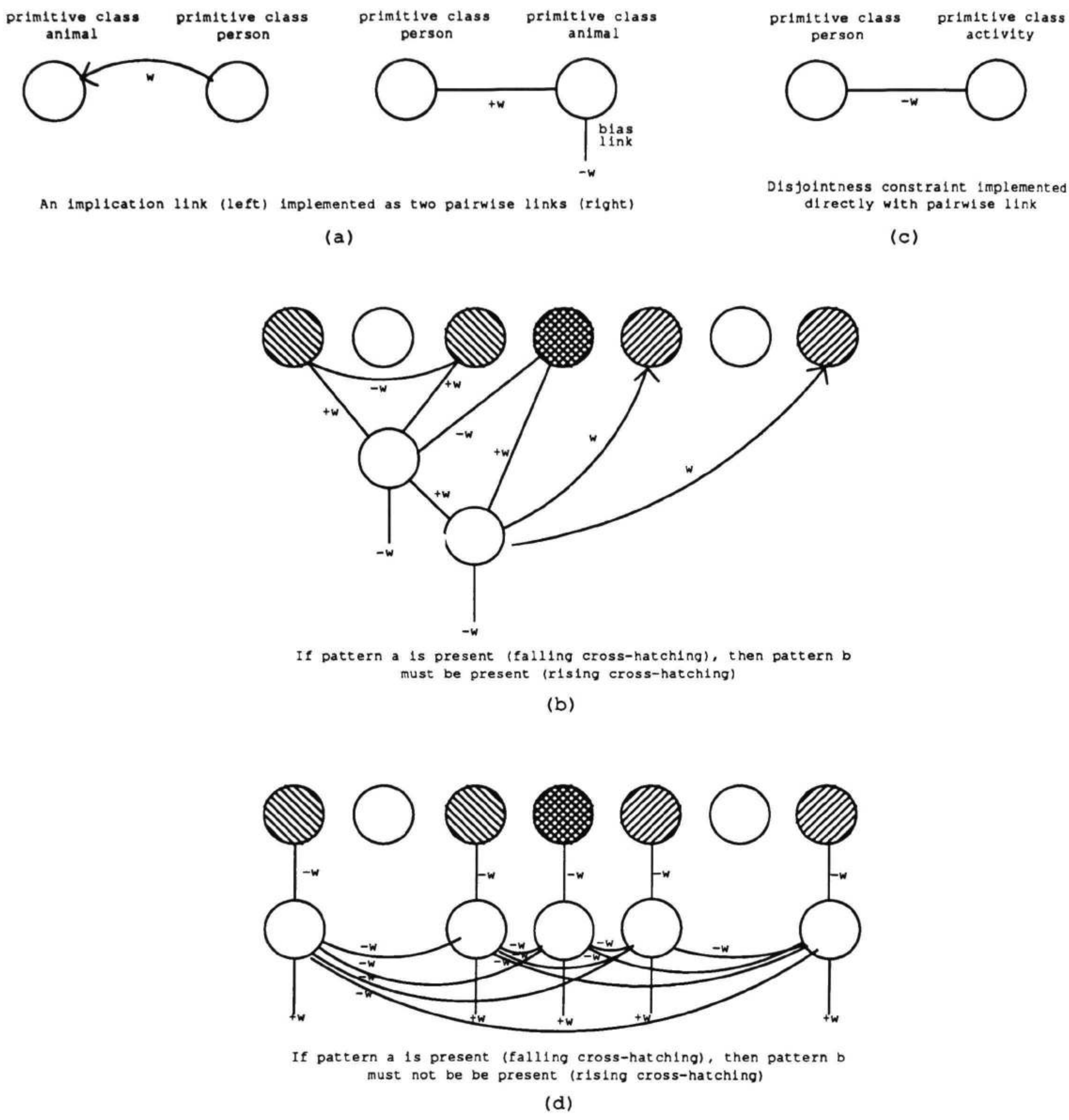

Figure 3: (*a*) Specializes constraints are implemented with *implication* links, which require two pairwise links. The contribution to the global energy function is $+w$ if the <*primitive class person*> micro-feature is on and the <*primitive class animal*> micro-feature is off; otherwise the contribution is zero. Implication links are drawn with an arrowhead and labeled with an unsigned weight. (*b*) When the pattern of the more specific concept has more than one active bit, extra units are required to detect its presence; and when the pattern of the more general concept has more than one active bit not shared by the more specific concept, more than one implication link is required. (*c*) Disjoint constraints are implemented with inhibitory pairwise links if both concept patterns have a single bit on. (*d*) Otherwise a technique discovered by Steve Nowlan requiring a "winner take all" circuit attached with inhibitory links is used to make sure the union of the two patterns results in a high energy state.

pattern in the "flying" column of the **role-fillers** module.

- **Value Restriction Constraints:** The *<restriction has-job armchair-activity>* micro-feature in the **subject-type** module must excite the pattern for this VR in the **role-filler-type-restrictions** module. If a VR in the latter module applies to a role being filled by some individual, then the individual must be of the appropriate type, as represented in the **role-filler-types** module. Detecting whether a VR applies to a role-filler cannot be done with pairwise links alone, so there is an auxiliary module to implement these constraints. This module is actually larger than all the others put together, and so is the limiting factor on the network's speed. I plan to give up the simplicity of the current Boltzmann/Hopfield model and include more complex relations among units directly as terms in the energy equation rather than implementing them with pairwise links. This will eliminate the need for all auxiliary modules.
- **Minimum Restriction Constraints:** If the *<minimum has-job 1>* micro-feature is on in the **subject-type** module, then the HAS-JOB pattern must be active in at least one of the seven columns in the **role-fillers** module.

3.4. Input/Output Implementation

Producing and interpreting μKLONE's distributed and coarse coded representations requires rather elaborate machinery. The exception to this complexity is the **subject** module, where a local representation is used, and exactly one unit is always active. Input and output take place directly in this module. In the **subject-type** module, which uses distributed but not coarse coded representations, several concept patterns may be combined. To allow the system this freedom, constraints on the subject's type imposed by presuppositions of a query are expressed by clamping patterns into one or more **subject-type-IO** modules, rather than clamping the **subject-type** module directly. The concept represented in the **subject-type** module must be subsumed by each concept in the **subject-type-IO** modules.

Each "with" clause in a query requires a set of IO modules to represent the role, MinR, role fillers, and VR. Each **MinR-IO** module interacts solely with a **role-fillers-IO** module to ensure that a sufficient number of individuals are represented there. Using the technique of figure 2, the role and role-fillers patterns in IO modules are combined in an auxilliary module, which is constrained to pull out of the central **role-fillers** module. Pull out is a technique introduced in (Mozer, 1984) for extracting meaningful constituents from coarse-coded modules in which several patterns have been superimposed. With this arrangement, the set of individuals represented in the **Role-Fillers-IO** module filling the role represented in the **role-IO** module must be a subset of all the role fillings represented in the central **role-fillers** module. Using the same technique of having an auxiliary pull-out group, the concept represented in the **VR-IO** module as a restriction on the role represented in the **role-IO** module is forced to be one of the VRs represented in the central **role-filler-type-restrictions** module. Though described as performing input, any of these IO modules can perform output if they are not clamped. Positive biases on all the input/output units result in the retrieval of the most specific answers compatible with the scenario represented in the central modules.

3.5. Performance

The network derived from the example KB has 2531 units and 16,959 links. Using the Hopfield model for annealing is generally more efficient than using the Boltzmann model,[3] and requires an annealing

[3] (Marroquin, 1985) reports an order of magnitude improvement.

schedule of 500 time steps for this network. At each time step the state of each unit is updated, and the entire search takes 10 minutes on a Symbolics Lisp Machine.

4. A More Detailed Description of the Reasoning Process

Posing the example query entails clamping patterns in some of the modules. One unit clamped on is the Ted unit in the **subject** module. Since Ted is known to be a SAILOR and a TV-BUFF, this unit excites those patterns in the **subject-type** module. At the same time the pattern for MILLIONAIRE-PLAYBOY is clamped into **subject-type-IO-1**, which excites the same pattern in the **subject-type** module, where three concept patterns are now superimposed. The conflict between being a sailor and being a millionaire playboy is not felt here, though, but in the coarse coded modules. The *<permission has-job sailing>* micro-feature excites the pattern for HAS-JOB in the sailing column of the **role-fillers** module. Since sailing is asserted to be a VIGOROUS-ACTIVITY, this pattern has already become active in the sailing row of the **role-filler-types** module. The *<restriction has-job armchair-activity>* micro-feature in the **subject-type** module excites the corresponding pattern in the **role-filler-type-restrictions** module. Since sailing is filling this role, the ARMCHAIR-ACTIVITY pattern is excited in the sailing row of the **role-filler-types** module. Now two contradictory patterns, VIGOROUS-ACTIVITY and ARMCHAIR-ACTIVITY, are excited in this row. The conflict can be worked out by violating any of several constraints: that vigorous-activities and armchair-activities are contradictory, that sailing is a vigorous activity, that playboys have sedentary jobs, that sailors' jobs include sailing, that Ted is a playboy, or that Ted is a sailor. Each of these constraints has an associated cost for violation. In a learning system these costs would be chosen automatically. Here I have chosen them so the requirement that sailing must be the job of a sailor is weakest. After being excited initially, the pattern for HAS-JOB is inhibited in the sailing column of the **role-fillers** module.

The *<minimum Has-Expensive-Hobby 1>* **subject-type** micro-feature is active, so some individual must be found to fill the role. The one chosen is the one for which the HAS-EXPENSIVE-HOBBY pattern is already most active. Since flying and sailing are both known to be ACTIVITIES-REQUIRING-EXPENSIVE-PROPS, they have a head start. In addition, sailing has some residual activation of HAS-EXPENSIVE-HOBBY because of the commonality with having a job, which was formerly active, and so sailing is chosen.

The *<minimum Has-TV-Related-Interest 1>* **subject-type** micro-feature seeks a filler for its role. The leading candidates are TV-Watching, TV-Acting, and TV-Network-Management. Simultaneously, the *<minimum Has-Job 1>* **subject-type** micro-feature, part of the MILLIONAIRE-PLAYBOY pattern, seeks a filler for its role. Jobs must be profitable activities, so Corporate-Raiding, TV-Acting, and TV-Network-Management are leading candidates. TV-Acting, however, is a vigorous-activity and is inhibited by the "VR HAS-JOB ARMCHAIR-ACTIVITY" pattern in the **role-filler-type-restrictions** module. Since TV-NETWORK-MANAGEMENT is excited by both the HAS-TV-RELATED-INTEREST and HAS-JOB minimum restrictions and these roles have similar properties, it is chosen to satisfy both requirements.

Now the central modules have reached a stable configuration in which only one constraint is violated, which is the best that can be done for this query. The only remaining task is for the **role-filler-IO** modules to pull out the individuals filling the HAS-JOB and HAS-HOBBY roles.

5. Conclusion

μKLONE demonstrates that sophisticated knowledge representation systems can be developed which retain the advantages of micro-feature based representations. The Ted Turner example demonstrates that a sub-symbolic implementation can solve common sense reasoning problems which cause difficulties for current symbolic systems.

Acknowledgments

Geoff Hinton and Dave Touretzky have been very helpful with the design of μKLONE and the preparation of this paper. This research is supported by NSF grants IST-8520359 and IST-8516330, and an ONR Graduate Fellowship.

I. Formal Domain Definition

The following input was used by the network building algorithm to produce a Hopfield network for answering queries. The syntax derives from that of KL2's definition language. 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. (This is rather different from KL2, where a completely different language is used for assertions.)

```
((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-Prop))               ;a PLAYBOY must have some HOBBY
                                                                    ;which is an ACTIVITY-REQUIRING-AN-EXPENSIVE-PROP
 (DEFCONCEPT Activity-Requiring-Expensive-Prop (SPECIALIZES Activity)
  (SOME Has-Prop Expensive-Item))
 (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 SAILORs JOBs
 (DEFCONCEPT Activity (PRIMITIVE))
 (ASSERT-CONCEPT Activity (DISJOINT Inanimate-Object) (DISJOINT Animal))
 (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))
 (DEFCONCEPT TV-Buff (SOME Has-Interest Television-Related-Activity))
 (ASSERT-CONCEPT TV-Buff (SPECIALIZES Person))
```

```
(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-Prop (PRIMITIVE))
(ASSERT-ROLE Has-Prop (DOMAIN Activity) (RANGE Inanimate-Object))

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

Hinton, G. E. Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel Models of Associative Memory.* Hillsdale, NJ: Erlbaum, 1981.

Hinton, G. E., McClelland, J. L. & Rumelhart, D. E. Distributed representations. In D. E. Rumelhart, J. L. McClelland, & the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition..* Cambridge, MA: Bradford Books, 1986.

Hinton, G. E., & Sejnowski, T. J. Learning and relearning in boltzman machines. In D. E. Rumelhart, J. L. McClelland, & the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge, MA: Bradford Books, 1986.

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.*, May 1984, *81*, 3088-3092.

S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi. Optimization by simulated annealing. *Science*, 1983, *220*, 671-680.

Marroquin, Jose Luis. *Probabilistic Solution of Inverse Problems.* Doctoral dissertation, MIT, September 1985.

McDermott, Drew. What AI Needs from Connectionism. August 1986. Unpublished notes accompanying a lecture given at AAAI-86.

Mozer, M. C. The perception of multiple objects: A parallel, distributed processing approach. 1984. Unpublished thesis proposal.

Quillian, M. R. Semantic memory. In M. Minsky (Ed.), *Semantic information processing.* Cambridge, Mass: MIT Press, 1968.

Reiter, R. A Logic for Default Reasoning. *Artificial Intelligence*, 1980, Vol. 13(1,2).

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

Vilain, M.B. The Restricted Language Architecture of a Hybrid Representation System. In *IJCAI-85.* Morgan Kaufmann, 1985.