The QBKG System: Generating Explanations from a Non-Discrete Knowledge Representation[1]

Hans J. Berliner
David H. Ackley

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

The QBKG system produces critical analyses of possible moves for a wide variety of backgammon positions, using a hierarchically structured, non-discrete form of knowledge representation. This report compares discrete and continuous representations and reasoning systems, addressing issues of competence, robustness, and explainability. The QBKG system is described and demonstrated.

Discrete systems vs Continuous systems

Most work in knowledge representation for artificial intelligence systems has used some variety of "discrete" representation and control structure, from the condition-action rules of production systems [9], to a variety of frame-based systems [11, 4], to various sorts of semantic network [10, 7]. These systems have in common the property that at any given time there is an unequivocal distinction between what knowledge is *relevant* and what is not (with relevance criteria such as "those productions whose condition portion is satisfied", "those scripts that are activated", or "those nodes with marker 2 set."). This all-or-none assumption increases the efficiency of these systems by reducing the effective size of the knowledge base, and makes construction of the knowledge base simpler by guaranteeing modularity. The price exacted for this simplicity can be high, however, in terms of system behavior. As discussed below, such systems tend towards anomalous behavior in certain circumstances, and are typically very sensitive to noise. Some more recent work with these representations has centered on relaxing the all-or-none assumption in various ways, such as allowing for partial matches in the condition-part of production rules [2] and the various spreading-activation theories in semantic nets [5].

Another reason that discrete systems seem natural stems from the fact that all systems must eventually make basically all-or-none decisions about their actions. The traditional view seems to assume that the discreteness of the ultimate action implies that it will have discrete justifications, with discrete reasons for the justifications and so on, until the discrete inputs are reached. A major alternative scheme was advanced by workers interested in game-playing systems, in the construction of knowledge-intensive evaluation functions for games such as backgammon, where the element of chance introduces a branching factor that makes substantial exhaustive search infeasible. In this paradigm, knowledge is represented by appropriate mathematical combination of observations on the world, and the control mechanism is the evaluation of the overall function on a selection of legal successor states, interpreted as a "gathering of evidence" procedure. Discretization is held off until the last possible moment, when the evaluations of the successor states are compared and the one with the largest evaluation is chosen. In this scheme, the relevance criterion for a knowledge item is fuzzy, with potentially all of the knowledge base implicated to some degree in each evaluation. For a large enough knowledge base, this might suggest that implementation on a uniprocessor would be slow; however, the structure is well suited to a parallel implementation.

It should be noted that even the choice of "discrete" versus "continuous" representations is not a discrete choice; in fact, there are varying degrees of continuity possible, from the two-valued propositional representation (e.g., "John is an adult" vs "John is not an adult"), to a finer grain representation ("John is in the 18 to 34 group"), to an essentially continuous representation ("John is 26.087 years old"). In many cases, the two-valued approach seems completely adequate ("John is a male"), and it seems easy to ignore the odd boundary cases that occasionally crop up. Unfortunately, in many areas of practical interest there is a wide "gray area" between the extremes (e.g., for age, duration, size, shape, color, beliefs, desires), and two-state (or *n*-state, for small[2] *n*) systems tend to manifest undesirable behavior near the boundaries between the states. In society, for example, discretization leads to surprising behaviors such as pretending to be "over 18", or driving from Massachusetts to New Hampshire to buy liquor. Berliner has shown [3] that Samuel's use of non-linearity in an attempt to improve his checkers program probably foundered on this problem.

The important point is that too large a *grain size* for an observation can have a disastrous effect on system behavior. The extra information in a fine-grained observation can always be discarded higher in the knowledge structure if it is not needed, such as when the digital watch tells us it is "4:56:34" and we think "five o'clock." In other circumstances, such as timing an egg, we would maintain a finer grain. It is clear that the needed grain size varies depending on the task at

[1] This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551.

[2] "small" here implying that large numbers of usefully distinguishable observations are being "bucketed" together.

hand. In some circumstances, the simple two-state assumption is adequate, but in general, no *a priori* grain size assumption can be made.

For this reason, discrete systems tend to be fragile in the face of noisy or erroneous inputs. Simply put, in a two-valued system, if you are wrong, you are *very* wrong. In a discrete medical diagnosis system using production rules, for example, an erroneous result on a test could prevent the system from ever making an accurate diagnosis, because the knowledge relating to the actual disease is not used, due to the non-satisfaction of the condition portions of the relevant productions. This could lead to an inaccurate diagnosis (or no diagnosis at all) despite a preponderance of evidence, excepting one test, pointing to the actual malady.[3]

Shifting to a more continuous representation can alleviate both of these problems. The boundary problem is handled by removing the hard boundaries and replacing them with non-linear functions which provide context-sensitivity (as in the way that scores on IQ tests are divided by the subject's age, as opposed to, say, interpreting the raw scores on different scales depending on whether the subject is over or under 10 years old). The fragility problem is handled in two ways: on the one hand, input error or uncertainty is not magnified by the "bucketing" process, and on the other hand, the "gathering of evidence" control structure ensures that the most reasonable hypothesis based on *all* available data will not be missed due to a small miscue.

One strong advantage of some discrete systems is that they are very well suited to the task of explaining what they are doing, a task at which humans are frequently quite adept. Given a subgoal structure, it is very simple to explain *why* a particular fact is needed (to prove the next higher goal in the structure) and *how* a particular fact is to be established (by proving all needed subgoals immediately below.) [12, 6] Given only a continuous evaluation function, it is not immediately clear how to explain why one evaluation is better than another or what the significance of a particular observation is in the overall scheme of evaluation.

The QBKG system is an example of a continuous knowledge representation system that plays backgammon and provides a mechanism for explaining some of what it does. It is derived from the BKG system, which demonstrated expert-level abilities in human competition and introduced SNAC [3], which forms the basis of the method used for structuring knowledge in QBKG. This paper presents the high-level issues addressed by the system and describes the fundamental mechanisms used. For a more extensive treatment of the system, discussion of the limitations of the method, and possible extensions to a learning system, see [1].

[3] The MYCIN system [6] dealt with this problem by generalizing production rules to function with many-valued logic rather than two-valued. This important step towards continuous knowledge representation is sometimes overlooked.

The QBKG system

The explanation mechanism of QBKG must handle two main issues. First, it must isolate the backgammon knowledge relevant to any particular query from the (usually large amount of) knowledge that does not bear on a given situation. The second issue is to provide some mechanism for deciding when quantitative changes should be viewed as qualitative change; in essence, to provide the judgemental ability that discrete systems enjoy by virtue of their all-or-none assumption. (There was relatively little effort expended in the generation of natural language output; in the example below, the output has been left "in the rough" as the system generated it. Most "language issues" have been ignored.)

QBKG is oriented around answering the question "Why did you make *that* move, as opposed to *this* move?" This reduces the explanation task to one of accounting for the *differences* between a pair of moves. The evaluation function is structured in a hierarchical fashion as shown in Figure 1. At the leaves of the tree are *primitive observations* (*Prim* in Figure 1) constituting the system's source of knowledge about the world. The primitives are combined into *concepts* using a variety of mathematical operators provided by the system. Related concepts are collected higher in the tree by scaling each concept non-linearly by multiplying by an *application coefficient* (AC) [3] and summing the results to produce a new concept. This can be thought of as a unit conversion operation: the subconcept is converted into units of the concept, with the application coefficient giving the current conversion rate. In the case of IQ test scores, for example, units of raw score are converted into units of IQ by multiplying by 1/**min**(*age,cutoff-value*).[4] In QBKG, ultimately all values are converted into units of *heuristic value* and a single value, **Heur**, is available at the top of the tree.

The fundamental assumption of the explanation process is that important differences between a pair of moves will be reflected by "large" changes in the values of the highest level concepts that are related to the differences. Letting *Move1* denote the move with the larger **Heur** and *Move2* the one with the smaller, define δ*concept* = value of *concept* for *Move1* - value of *concept* for *Move2*. Referring to Figure 1, this assumption implies that if δ**Blocking** is "small", then backgammon knowledge related to blocking is not relevant to this comparison and should not be mentioned. If only one subconcept of **Heur**, say, **Tactical**, is not small, then all interesting differences are with respect to **Tactical** concepts, and the level of discourse for comparison can be narrowed to just tactical knowledge.

This is the method by which the relevant backgammon knowledge is isolated. Beginning at **Heur**, the system searches down the tree until a level is reached at which more than one significant difference is found. As desired, if the two moves are radically different in their effects, the commentary will begin at a relatively abstract level (e.g., tactical and

[4] It is somewhat sobering that this cutoff-value is typically 16.

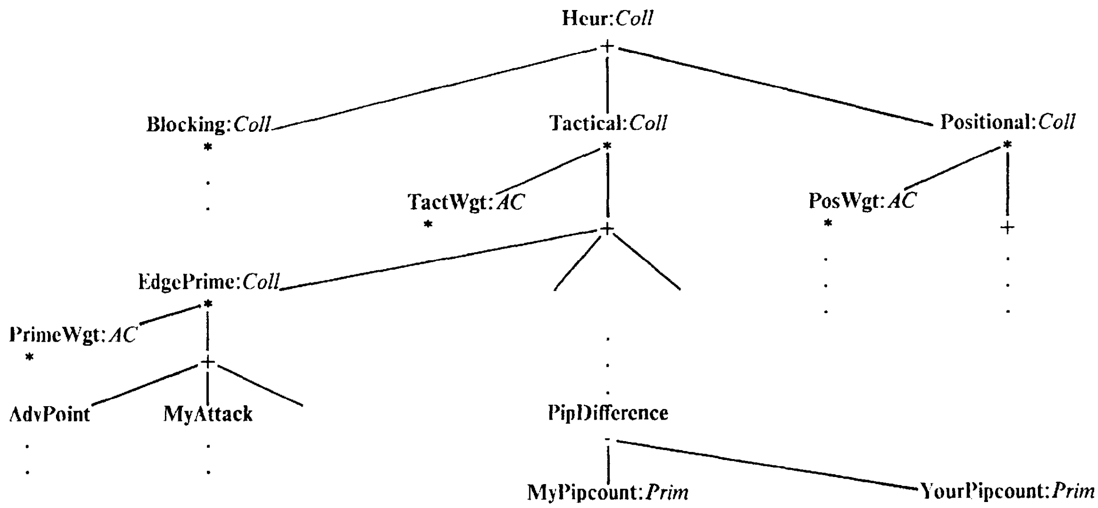

Figure 1: Schematic view of portions of the QBKG knowledge base.

positional issues) and if the two moves are quite similar, the commentary will focus on the crucial differences at whatever level they are found.

If the discussion is at a fairly narrow level, it is usually sufficient to just describe the differences and their magnitudes; at higher levels, this leads to unsatisfying, "hand waving" commentaries. The level at which an explanation feels satisfying varies from person to person and topic to topic[5], so we have adopted a simple heuristic. The broad concepts at the top of the tree are denoted *collections* (*Coll* in Figure 1), and the system is built to automatically "look inside" of any collections that are mentioned; in effect, supplying for free the question "Why is there a difference in that collection?"

The above discussion is predicated upon having the ability to recognize "large" or "significant" differences in the values of concepts. In a two-valued system, any difference is a large one (on the order of *True* versus *False*), and the process of recognizing significant differences is done *outside* of the system, during the generation of discrete primitive observations of a continuous world. Unfortunately, it is impossible to determine what should be considered significant without considering the *context* within which the judgement is to be made. A difference of ten feet, for example, is much larger in the context of "Distance I am from the ground" than in the context of "Distance I am from the moon." A context provides a means of classifying differences into fuzzy classes such as "about the same", "somewhat larger", and so on. The number of classes will vary depending on personal taste as well as the degree of refinement of the knowledge base. Our assumption of six classes ("not significantly", "slightly", "somewhat", "much", "very much", and "vastly") seems to work quite well.

In terms of the QBKG structure, the context of a concept is the set of more general concepts of which it is a part. Some of the more primitive concepts, such as **MyPipcount**, appear in several places in the knowledge structure and can therefore be judged in several different contexts. To judge a difference in context, it is necessary to determine how that difference affects the value at the top of the knowledge structure. Given the sign and magnitude of a difference which is to be considered a significant improvement for **Heur**, this *goodness metric* can be propagated down through the tree to determine how much better or worse one move is than another with respect to a given concept in a given position. In Figure 1, for example, if the goodness metric for **Heur** is assumed to be $+10$, and in a given position **TactWgt** equaled 3, then the goodness metric for **EdgePrime** would be $+10/3$, and a δ**EdgePrime** of less than 10/3 would be judged "about the same", a δ**EdgePrime** between 10/3 and 20/3 would be "somewhat better", and so on.

This procedure requires an *a priori* goodness metric for **Heur.** In QBKG, probably the most satisfying overall context would be "What is the expected value of the game?" with a goodness metric of perhaps a tenth of a point. Such an evaluation function could be built, and in fact the system has an independent computation used to approximate the expected value, which is used in making doubling decisions.[6] The original BKG evaluation function only needed to order the possible moves with respect to a given initial position, and this "relative" nature remained through the translation to the QBKG-style evaluation function, so **Heur** values resulting from different initial positions are not directly comparable. With respect to a given position, however, various heuristics have been devised which empirically give satisfactory results in the determination of significant differences.

[5] As evidenced by the child who responds to every explanation with "Why?"

[6] The approximation of the expected value is too crude to provide adequate discrimination between individual moves, however, thus motivating the "two function" scheme used in BKG.

The QBKG system is now able to produce cogent commentary on about 70% of positions randomly presented to it. Its principal flaws at this point are idiosyncrasies in the knowledge base due largely to historical reasons. For example, the system is unable to comment on the relative merits of two moves where one move separates the opposing armies (creating a non-interfering race to the finish of the game) and the other move does not.

An example of its ability is shown in Figure 2, taken from a set of problems by Holland [8]. QBKG has chosen 17-24 as its move, and the user has asked for a comparison with 12-18,17-18. Holland comments on this position, "The correct play is to move one man from [the 17 to the 24 point], hitting Black's blot. You must try to prevent Black from establishing [the 24 point]. If you were to make [the 18 point] in lieu of hitting, Black would have 11 chances out of 36 to roll a 1, giving him a position from which he will still be able to win the game." (p. 66, paraphrased into QBKG's notation.) Figure 3 shows QBKG's commentary on this choice. Part (1) is some general comments about the situation, based on **PipDifference** and the independent expected value computation. Part (2) is QBKG's opinion on the worth of the two moves, based on δ**Heur** and some knowledge about the range of possible **Heurs** in this position. Part (3) is the result of the focusing mechanism discussed above and shows the extreme importance of hitting the lone Black man. The crucial issue of stopping Black from making an advanced point (**AdvPoint** in Figure 1) is discovered and reported, while irrelevant differences between the moves, such as the added risk that White may be hit (11 chances for Move 1 vs 1 for Move 2), are ignored. The bracketed numbers in part (3) are reference numbers by which the user may request further commentary on the specified topics. The system responds to such requests by recursively entering the focusing system using the selected topic as the root of the search, in the same manner as it handles topics which are denoted collections.

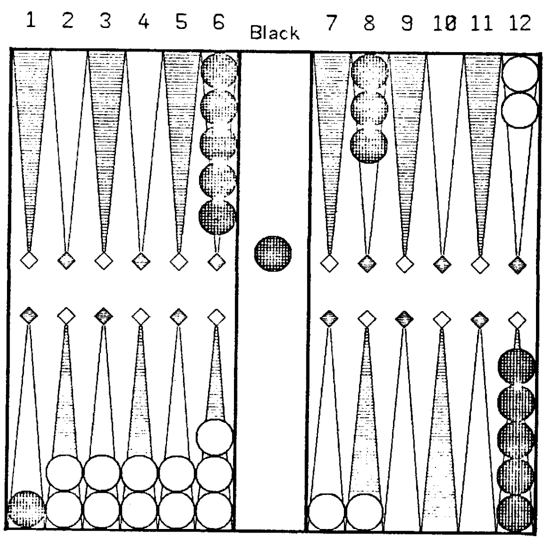

Figure 2: Sample position, White to play 6,1

```
(1)In the given position, White is far ahead in
   the race, and has a winning advantage, with
   substantial gammon chances.
(2)The actual move, 17-23,23-24(Move 1), is much
   better than the suggested move,
   12-18,17-18(Move 2).
(3)There is nothing to recommend Move 2.The
   advantages of Move 1 are:
      o vastly better chances of keeping Black
        from making an advanced point [1].
      o very much better attack by White [2].
```

Figure 3: QBKG's commentary on two moves in Figure 2.

Conclusions

We view discretization as a simplifying assumption that becomes less and less workable as AI systems begin to tackle real-world tasks, with the accompanying problems of noise, uncertainty, and shifting notions of what is true and what is relevant. Using a fine-grained representation, hierarchical knowledge structuring and the context-sensitivity provided by application coefficients, knowledge-intensive evaluation functions of the form used in QBKG provide a means of avoiding the difficulties introduced by an excessively discrete view of the world, while still providing the benefits of explainability and uniformity of representation which are demonstrated advantages for an artificial intelligence system.

References

1. Ackley, D. H. & Berliner, H. J. The QBKG System. Carnegie-Mellon University, 1982. *in preparation*

2. Anderson, J. R. *A.C.T.* Harvard University Press, 1983. (forthcoming)

3. Berliner, H. J. "Backgammon Program beats World Champion." *Artificial Intelligence 14* (1980).

4. Bobrow, D. G. & Winograd, T. "An Overview of KRL." *Cognitive Science 1*, 1 (1977).

5. Collins, A. M. & Loftus, E.F. "A spreading activation theory of semantic processing." *Psychological Review 82* (1975).

6. Davis, R., Buchanan, B., Shortliffe, E. "Production Rules as a Representation for a Knowledge-Based Consultation Program." *Artificial Intelligence 8* (1977).

7. Fahlman, S. E. *NETL: a system for representing and using real-world knowledge.* MIT Press, 1979.

8. Holland, T. *Better Backgammon.* Reiss Games, Inc., New York, 1974.

9. Newell, A. & Simon, H. A. *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, N. J., 1972.

10. Quillian, M. R. Semantic memory. In *Semantic Information Processing*, Minsky, M., Ed.,MIT Press, 1968.

11. Schank, R. C. & Abelson, R. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale, N.J., 1977.

12. Winograd, T. *Understanding Natural Language.* Academic Press, 1972.