PROGRESSIVE MODELLING FOR THE HENDRIX SIMULATION SYSTEM

Lawrence O. Kilgore
Daniel P. Friedman

August, 1976

PREFACE

This paper presents approaches to and models for the proposed Hendrix Simulation System and its subsequent implementation by John Lowrance. To present these concepts it is recommended the user have a working knowledge of the Lowrance paper and of LISP 1.6. The paper is divided into three parts: 1) a brief discourse about world modelling technique, 2) the implementation of all the worlds proposed by Gary Hendrix, and 3) a series of new worlds designed to provide evidence of the power and accuracy of the system. Included are appendices with a series of new LISP functions useful in certain types of simulation and debugging.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER I - OVERVIEW

As the first user of the Lowrance implementation I hope to provide insight into using it efficiently and successfully. A unique group of problems arise from the use of this system which forewarning can diminish. In addition, certain types of simulation tend to work well with this system while others fail miserably. In demonstrating new worlds it is hoped that one can ascertain the types of worlds that should be attempted. Finally, it will become evident that the true power of the system lies in that it is imbedded in LISP and quite understandably the better the LISP programmer, the better the simulator.

CHAPTER II - PROGRESSIVE MODELLING

Problems

The single most important thing to remember about modelling is that model world processes and events are pure representational simulation. In order to attain a logical flow to simultaneous and chain reaction processes, the system must be presented with a world model and its related, preplanned, and intercoordinated scenarios. The system triggers these scenarios by the existence of a set of satisfiable conditions of which (to the system) readability has little or no importance. However, the human simulator must be able to see and understand results (and failures) of any preconceived processes. In this sense, guidelines may be set down to facilitate a high degree of visual cohesion. In addition, representations of facts and relationships within a world and its associated scenarios, specifically the atoms within the n-tuples in the state of the world model,

have no intrinsic value (in the LISP PNAME/VALUE sense). It is this final fact that emphisizes the importance of readability and rapid transitions between world model and scenario.

The importance of human visual simplicity in modelling should not be underrated. Unlike many languages, the semantics in scenarios are not at all concerned with things such as function manipulation, variable procedures, and logical control, rather, the words and symbols are in themselves of crucial importance. In that sense then, to view the world and its processes is to view the world model and its scenarios. If scenarios work the world list will change and the simulator will look at the results. If scenarios do not work the world list will remain unchanged and the simulator must look back and forth between it and its scenarios. Either way, unless one is modelling very simple processes with very tiny world models you inevitably spend vast amounts of time searching through lines of relations.

Any simulations being preformed are entirely represented by updated snapshots of tuples so it is important to sketch visual aids or blueprints to help conceptualize each portion of the simulation. One big problem in designing a detailed group of scenarios is world conceptualization. In the planning stages this saves a great deal of time determining what is required for scenarios. These blueprints not only aid in catching errors and testing functions but can later be extended to a practical application or construction design for the simulated machine or mechanical process. In fact, any user that would be performing any serious applicational simulation would already have a detailed design or blueprint available and would be working from it to

test its feasibility or to trouble-shoot. For this reason, the modeler of a nonapplicational simulation (like the ones presented in this paper) should also have his ideas well in hand; mainly a picture.

Although at first thought, specific provisions for appearance seem rather trivial, it becomes a crucial dilema on complex worlds. Problems, syntax errors, and general frustrations increase exponentially with an increase in scenario complexity (which increases with the addition of each new scenario). For example, a very closely modelled process, without careful planning, may turn into an incoherent nightmare of 10-15 scenarios, 50 State of the World Model tuples, and dozens of LISP functions all stubbornly failing or refusing to do anything at all.

Another temptation during modelling is to define all scenarios at once and debug the entire list in conjunction with its packet of help functions. This should be avoided at all costs. LISP functions can be debugged or tested in the normal manner (without loading the simulation program). As for debugging scenarios this will be dealt with in the BUGOFF section.

In summary there are four basic problems:

1) representational readibility(for examination and symbolization)
2) conceptualization (blueprints)
3) scenario development
4) world debugging

Solutions

One solution to the problem of readability is the simple indentation at every level within each scenario. This means that every subitem of equal value is indented. Specifically this causes a list like ordering on all items such as initiation conditions.

Each condition is on a seperate line and ICS, ICN, EID, EIA (and others) and scenarios (see Figure 2.1) deliberately begin to take on a resemblance to the State of the World Model (see Figure 2.2). As can be seen, the structure in Figure 2.2 provides quick orientation for debugging, adding, and deleting tuples as required.

To represent the atoms themselves, the spelling is-arbitrary, however the much finer model requires many more parameters. For example in the GOTO scenario, a one-dimensional model (robot travels linearly from points 1-10) the only positional variables required are CFROM and CTO. In a two-dimensional model (X-Y coordinates) this becomes CFROMX, CFROMY, CTOX, and CTOY. In a three-dimensional model that locates objects (X-Y-Z coordinates) the scenario would require CFROMX, CFROMY, CFROMZ, CTOX, CTOY, CTOZ (for the Robot), CATX, CATY, CATZ (for located objects), CLX, CLY, CLZ (for last search position), CSFX, CSFY, CSFZ (for initial search position), CSTX, CSTY, CSTZ (for next search position), and all just for positional information, discounting numerous other necessary parameters. All three processes are nearly identical yet with the simple addition of a finer modelling aspect the number of positional parameters jumps from 2 to 4 to 18!

Clearly two problems arise from this proliferation. First, in any complex model you must expect a large ungainly set of parameters. Second, and in direct opposition to the first, is that with large numbers of similar parameters, a unique and easily recognizable PNAME must be adopted for each, yet, this generally results in a longer representation. Often this may be remedied by a clever representation. Many numerical parameters can easily

```
(TURNVALVE (PAR A V CRATE R : CMAXRATE N)
    (ICS(ALLOCATE-ACTIVATE R A TURNVALVE V CRATE)(AT R N)(AT V N)
        (TYPE A ARM)(MAXRATE V CMAXRATE)(STATE R AWAKE))
    (ICN(GT(*DIF CMAXRATE CRATE)0)(GE CRATE 0))
    (EID(RATE V *)(ALLOCATE-ACTIVATE R A TURNVALVE V CRATE))
    (EIA(RATE V CRATE)))
```

Turnvalve Regular
Figure 2.1

```
(TURNVALVE (PAR A V CRATE R : CMAXRATE N)
    (ICS(ALLOCATE-ACTIVATE R A TURNVALVE V CRATE)
        (AT R N)
        (AT V N)
        (TYPE A ARM)
        (MAXRATE V CMAXRATE)
        (STATE R AWAKE))
    (ICN(GT(*DIF CMAXRATE CRATE)0)
        (GE CRATE 0))
    (EID(RATE V *)
        (ALLOCATE-ACTIVATE R A TURNVALVE V CRATE))
    (EIA(RATE V CRATE)))
```

Turnvalve Indented
Figure 2.2

be eliminated by incorporating them directly into LISP functions. This type of situation arises in models where certain data is surely not to change yet is crucial for certain equations. For example, this has been done in the Billiards World (in this paper) where the dimensions of the Billiards table is standard and unchangeable. One LISP function which returns the location a rolling Billiard ball will hit the bunk uses these dimensions included directly in the code. This can be used as often as necessary and will tend to reduce the number of parameters in any one scenario.

Conceptualization of any world model can be as brief as possible. Only the minimum of sketching is necessary and only those refinements relevant to each specific model should be included (the movement of atoms would be a ridiculous inclusion in the Voltage World but would be paramount in a model of an Atom Smasher). If specific purposes are to be acheived or models are to be patterned after real world examples then a blueprint can be used. The X-Y or X-Y-Z coordinate axes will suffice for problems involving movement. Mechanical simulation generally needs only a sketch of the tool or machine while problems involving unviewable phenomena, as in the case of the Voltage World, are usually drawn like a circuit board.

World debugging assumes two problems, syntax errors and faulty scenario/world model coordination. The latter usually arises through poor conceptualization. It is important not to model too coarsely. Problems and inaccuraces will result if the physical phenomena is not understood. If for example formulas computing X-Y coordinates are inaccurate, results may appear correct (by

returning some coordinate) but are in fact not close.

BUGOFF

As for syntax errors a package of functions has been provided (APPENDIX B) that will scan scenarios and list all unrecognized words (with respect to parameters, keywords, defined functions, and state of the world model tuples). This will catch nearly all typographical errors and unwritten functions while giving their location in the scenario. In addition, it will locate parenthesis errors and illegal keywords. Such problems are extremely difficult to locate because unlike LISP functions which can be closed individually by brackets, scenarios are defined in groups (assigned to a variable). If a parenthesis error exists then its location is very vague. Likewise with syntax errors, faulty scenarios just don't fail to work they fail to do anything at all! A very subtle and hard to find spelling error (like a ∅ in place of an O) in a large packet of scenarios is like a needle in a haystack.

To use BUGOFF, simply call the function (BUGOFF SLIST SWM NUM) where SLIST is bound to the scenario list, SWM to the state of the world model list, and NUM to an integer equal to the number of scenarios in SLIST. For a sample run see Appendix B.

Weaknesses of the System

Certain types of scenarios, because of the large sets of combinations needed to be processed are subsequently handled very poorly by the system. One of the best examples of this is the Sort World (see Figure 2.3). This world has only 1 scenario and three types of n-tuples. With such a simple foundation it is surpris-

```
(SETQ SLIST 0)
(SORT (PAR B1 B2 L1 L2 : J1 J2 V1 V2)
    (ICS(AT B1 L1)
         (AT B2 L2)
         (CONTENTS B1 J1)
         (CONTENTS B2 J2)
         (VAL J1 V1)
         (VAL J2 V2))
    (ICN(LT L1 L2)
         (GT V1 V2))
    (EID(CONTENTS B1 J1)
         (CONTENTS B2 J2))
    (EIA(CONTENTS B1 J2)
         (CONTENTS B2 J1)))]
(SETQ SWM 0(
(AT BOX1 1)
(AT BOX2 2)
(AT BOX3 3)
(AT BOX4 4)
(AT BOX5 5)
(AT BOX6 6)
(CONTENTS BOX1 JACK6)
(CONTENTS BOX2 JACK5)
(CONTENTS BOX3 JACK4)
(CONTENTS BOX4 JACK3)
(CONTENTS BOX5 JACK2)
(CONTENTS BOX6 JACK1)
(VAL JACK1 1)
(VAL JACK2 2)
(VAL JACK3 3)
(VAL JACK4 4)
(VAL JACK5 5)
(VAL JACK6 6))]
```

Sort World SWM and single Scenario

Figure 2.3

ing to discover its gross inefficiencies. The state of the world model presents six Jack-in-the-boxes located at positions 1 through 6. Each "Jack" in a box is assigned a value 1 through 6 for JACK1 through JACK6 respectively. The "Jacks" are then placed out of order with JACK6 in BOX1 and JACK1 in BOX6 and so forth. The scenario SORT simply finds 2 boxes with positions in order but with "Jack" values out of order and then reverses them.

Because of the large combinations of initiation conditions that are satisfiable the system requires 13 minutes, 22 seconds of CPU time to sort 6 items! (see APPENDIX A) As can be imagined this amount of time would jump sharply with multiple scenarios. Worlds that depend upon similar structures should be avoided!

CHAPTER III - THE HENDRIX WORLD IMPLEMENTATION

Within the Lowrance paper the introductory worlds and Hendrix World include the bulk of the proposed worlds in the Hendrix paper. Certain gaps were not filled however and in all fairness to the creator these remaining unwritten items are combined to give an updated final working model. New inclusions are an X-Y axis equation (see Figure 3.1) solver to eliminate the circular track, a coordinated LOC to work along with the new GOTO, a gradual TURNVALVE, and the final version of FILLBUCKET. Some ideas on things like MONITORALARM and MONITORSET have been borrowed from the Lowrance paper.

Before introducing the sample run it may help to introduce each scenario and provide any relevant details (see Figure 3.2)

1) SETALARM remains the same as presented in the Lowrance paper but with the inclusion of X-Y locational parameters.

2) MONITORSET also remains identical with this same change.

3) AWAKENROBOT awakens the sleeping robot. Like all sleepy robots it immediately dashes to the clock and turns it off (e.g. effects added are GOTO the clock and OFFALARM once its there).

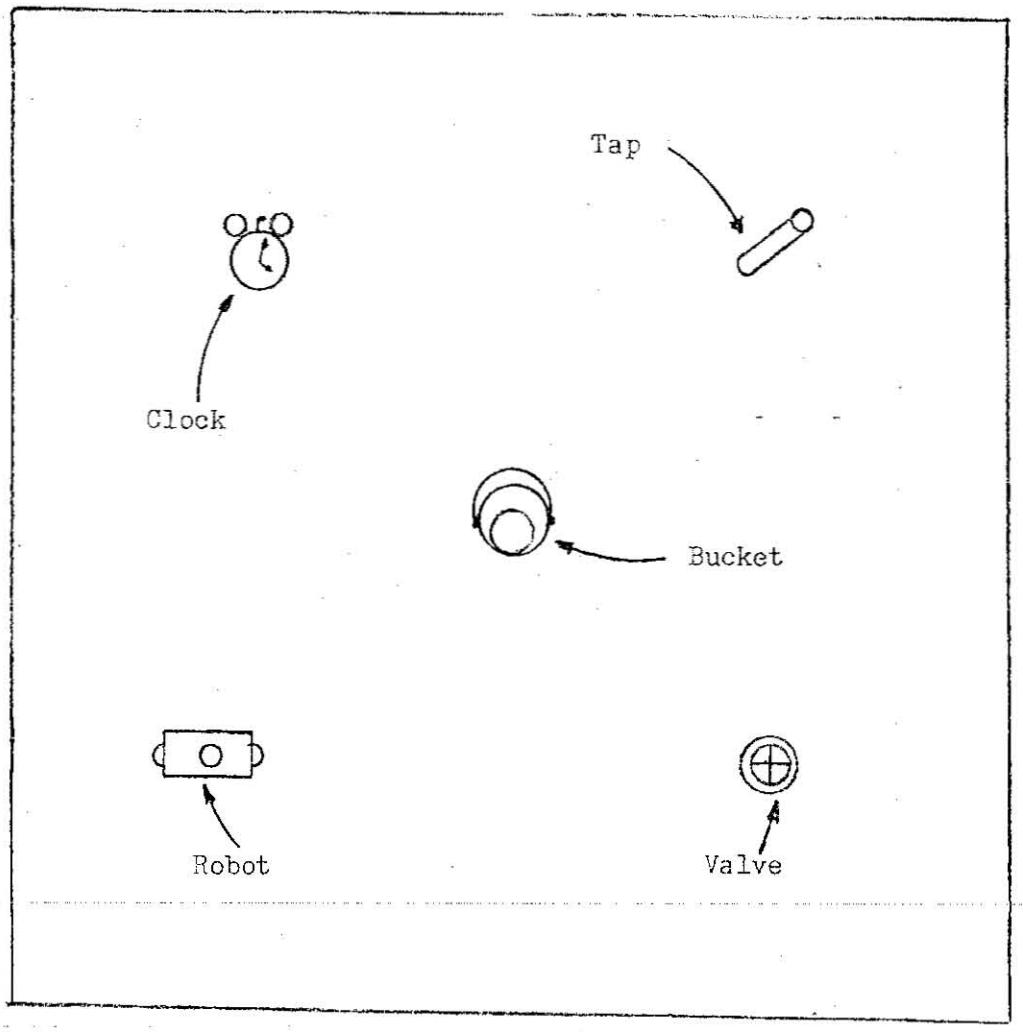
4) SOUNDALARM is the same as in the Lowrance paper.

5) MONITORALARM is also the same.

6) OFFALARM is the same as OFFALARM1 in the Lowrance paper but with the inclusion of X-Y coordinates.

7) SLEEPROBOT will put the robot asleep. Since all scenarios in which he is invloved require him to be awake, they will be (henceforth) nonfunctional.

8) TURNVALVE differs from the Lowrance version in that it is gradual. The possibility that its associated tap may already be flowing is taken into consideration and since the valve is turned gradually, the flowrate of the tap also increases/decreases gradually.




```
(AT CLK 50 150)
(AT RBT 40 50)
(AT BKT 100 100)
(AT VLV 160 50)
(AT TAP1 150 150)

(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MU MOBILITYUNIT)
(TYPE RBT-ARM ARM)

(MOVABLE CLK)
(MOVABLE RBT)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)

(ALARM OFF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 0)
(CAPACITY BKT 100)

(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0)
(TURNRATE VLV 0)
(MAXTURNRATEABS VLV 5)
(XRATE RBT 0)
(YRATE RBT 0)
(SPEEDLIMIT RBT 20)
(STATE RBT AWAKE)

(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED CLK)
(NOTGRASPED BKT)
(NOTGRASPED VLV)
(NOTGRASPED TAP1)
```

Hendrix World and SWM

Figure 3.1

```
(SETALARM (PAR R A K CSTIME : CX CY)
    (TCS(ALL-ACT R A SETALARM K CSTIME)
        (STATE R AWAKE)
        (TYPE A ARM)
        (GRASPING R A K)
        (ALARM DEF K)
        (AT R CX CY)
        (AT K CX CY))
    (ICN(LE # CSTIME)
        (LT(*DIF CSTIME #)12))
    (EID(ALARM DEF K)
        (ALL-ACT R A SETALARM K CSTIME))
    (EIA(ALARM SET K CSTIME)))

(MONITORSET (PAR R K CTM : CX CY)
    (TCS(ALL-ACT R SETALARM K CTM)
        (AT R CX CY)
        (AT K CX CY))
    (ICN(LT # CTM)
        (LE 12(*DIF CTM #)))
    (CCS(ALL-ACT R SETALARM K CTM)
        (AT R CX CY)
        (AT K CX CY))
    (CCN FUNC(*DIF CTM(*DIF 12(*TIMES 1.1 PASTLONG)))))

(AWAKENROBOT (PAR R K : CX CY)
    (TCS(STATE R ASLEEP)
        (ALARM SOUNDING K)
        (AT K CX CY))
    (EID(STATE R ASLEEP))
    (EIA(ALL-ACT R RBT-ARM OFFALARM K)
        (ALL-ACT R RBT-ARM GRASP K)
        (ALL-ACT R RBT-MU GOTO CX CY PA)
        (STATE R AWAKE)))

(SOUNDALARM (PAR K CSTIME)
    (TCS(ALARM SET K CSTIME))
    (ICN(EQUAL # CSTIME))
    (EID(ALARM SET K CSTIME))
    (EIA(ALARM SOUNDING K)))

(MONITOFFALARM (PAR K CTM)
    (TCS(ALARM SET K CTM))
    (ICN(NOT(EQUAL CTM #)))
    (CCS(ALARM SET K CTM))
    (CCN FUNC CTM))

(OFFALARM (PAR R A K : CX CY)
    (TCS(ALL-ACT R A OFFALARM K)
        (STATE R AWAKE)
        (TYPE A ARM)
        (ALARM SOUNDING K)
        (GRASPING R A K)
        (AT K CX CY)
        (AT R CX CY))
    (EID(ALL-ACT R A OFFALARM K)
        (ALARM SOUNDING K))
    (EIA(ALARM DEF K)))
```

Hendrix World Scenarios (Part I)

Figure 3.2

```
(SLEEPPROB (PAR P)
    (IC (ALL-ACT P SLEEPPROB)
        (STATE P AWAKE))
    (FID (STATE P AWAKE)
        (ALL-ACT P SLEEPPROB))
    (FIA (STATE P ASLEEP)))

(TURNVALVE (PAR P A V CTURNRATE CDESIREDFLOWRATE : CINITIALFLOWRATE
                                    MAXFLOWRATE CMAXTURNRATE CX CY)
    (ICS (ALL-ACT P A TURNVALVE V CTURNRATE CDESIREDFLOWRATE)
        (TYPE A ARM)
        (SPRAYING P A V)
        (RATE V CINITIALFLOWRATE)
        (MAXRATE V CMAXFLOWRATE)
        (MAXTURNRATEABS V CMAXTURNRATE)
        (AT P CX CY)
        (AT V CX CY)
        (STATE P AWAKE))
    (ICN (GE CMAXFLOWRATE CDESIREDFLOWRATE)
        (GE CDESIREDFLOWRATE 0)
        (GE CMAXTURNRATE CTURNRATE)
        (GE (*TIMES (*DIF CDESIREDFLOWRATE CINITIALFLOWRATE)
                    CTURNRATE) 0))
    (FID (TURNRATE V *)
        (RATE V *))
    (FIA (TURNRATE V CTURNRATE))
    (FGS (RATE V YFLOWRATE))
    (EGN (:= YFLOWRATE (*PLUS CINITIALFLOWRATE (TIMES CTURNRATE $)))
         (:= $ (QUO (*DIF YFLOWRATE CINITIALFLOWRATE) CTURNRATE))))
    (CCS (ALL-ACT P A TURNVALVE V CTURNRATE CDESIREDFLOWRATE)
        (AT P CX CY))
    (CCN FUNC (*PLUS (QUO (*DIF CDESIREDFLOWRATE CINITIALFLOWRATE)
                CTURNRATE) %))
    (EFD (TURNRATE V CTURNRATE)
        (ALL-ACT P A TURNVALVE V CTURNRATE CDESIREDFLOWRATE))
    (EFA (TURNRATE V 0)))

(FILLBUCKET (PAR V T P : CINITIALFLOWRATE CTURNRATE CCAPACITY
                         CINITIALCONTENT CX CY)
    (ICS (CONTENT V T)
        (RATE V CINITIALFLOWRATE)
        (TURNRATE V CTURNRATE)
        (CAPACITY P CCAPACITY)
        (CONTENT P CINITIALCONTENT)
        (AT T CX CY)
        (AT P CX CY)
        (ORIENTATION P UP))
    (ICN (NOT (AND (ZEROP (ABS CINITIALFLOWRATE)) (ZEROP (ABS CTURNRATE))))
        (GT CCAPACITY CINITIALCONTENT))
    (FID (CONTENT P *))
    (FGS (CONTENT P YCONTENT))
    (EGN (:= YCONTENT (PLUS (PLUS CINITIALCONTENT (*TIMES CINITIALFLOWRATE
                $)) (TIMES (TIMES .5 CTURNRATE) ($D $)))
         (:= $ (QUO (PLUS (MAS CINITIALFLOWRATE) (SQRT (PLUS (*DIF (SQ
                        CINITIALFLOWRATE) (*TIMES 2 YCONTENT))
                        (*TIMES 2 CINITIALCONTENT)))) CTURNRATE))))
    (CCS (TURNRATE V CTURNRATE)
        (AT P CX CY)
        (ORIENTATION P UP))
    (CCN FUNC (*PLUS (QUO (*DIF CCAPACITY CINITIALCONTENT)
                        CINITIALFLOWRATE) %)))
```

Hendrix World Scenarios (Part II)

Figure 3.2

```
(GRASP (PAR A E P : CX CY)
     (TCS(ALL-ACT P A GRASP P)
          (TYPE P ARM)
          (GRASPABLE P)
          (NOTGRASPED P)
          (STATE P AWAKE)
          (AT P CX CY)
          (AT P CX CY))
     (ETD(NOTGRASPED P)
          (ALL-ACT P A GRASP P))
     (ETA(GRASPING P A P)))

(RELEASE (PAR A P P)
     (TCS(ALL-ACT P A RELEASE P)
          (TYPE A ARM)
          (STATE P AWAKE)
          (GRASPING P A P))
     (ETD(GRASPING P A P)
          (ALL-ACT P A RELEASE P))
     (ETA(NOTGRASPED P)))

(MOVABILITY (PAR P : A P)
     (TCS(GRASPING P A P)
          (IMMOVABLE P))
     (ETD(MOVABLE P))
     (ETA(IMMOVABLE P))
     (TCS(GRASPING P A P)
          (IMMOVABLE P))
     (ETD(IMMOVABLE P))
     (ETA(MOVABLE P)))
```

Hendrix World Scenarios (Part III)

Figure 3.2

```
(GOTO (PAR R M CXT CYT CSPD : (SPDL CXF CYF ED EXR EYR)
    (IFS(ALL-ART R M GOTO CXT CYT CSPD)
        (STATE R AWARE)
        (MOVABLE R)
        (TYPE M MOBILITYUNIT)
        (SPEEDLIMIT R CSPDL)
        (AT R CXF CYF))
    (IFN(GE CXT 0)
        (GE CYT 0)
        (GE (*DIF 200 CXT) 0)
        (GE (*DIF 200 CYT) 0)
        (GT CSPD 0)
        (GE (*DIF CSPDL CSPD) 0)
        (:= ED (SQRT (+PLUS (SQ (*DIF CXT CXF)) (SQ (*DIF CYT CYF)))))
        (:= EXR (ROUND (*TIMES (*DIF CXT CXF) (SPD ED))))
        (:= EYR (ROUND (*TIMES (*DIF CYT CYF) (SPD ED)))))
    (FID (XRATE R *)
        (YRATE R *)
        (AT R * *))
    (FIA (YRATE R EXR)
        (YRATE R EYR))
    (FAS (AT R XX YY))
    (FGN ((:= XX (+PLUS (*TIMES EXR $) CXF))
          (:= $ (QUO (*DIF XX CXF) EXR)))
        ((:= YY (+PLUS (*TIMES EYR $) CYF))
          (:= $ (QUO (*DIF YY CYF) EYR))))
    (FFS (ALL-ART R M GOTO CXT CYT CSPD)
        (MOVABLE R))
    (FCN FUNC (+PLUS (QUO ED CSPD) $))
    (FFD (XRATE R *)
        (YRATE R *)
        (ALL-ART R M GOTO CXT CYT CSPD))
    (FFA (XRATE R 0)
        (YRATE R 0)))

(LOC (PAR R A B : CXF CYF CXR CYR)
    (IFS (GRASPING R A B)
        (MOVABLE B)
        (AT R CXF CYF)
        (XRATE R CXR)
        (YRATE R CYR))
    (IFN (NOT (AND (ZERO? (ABS CYR)) (ZERO? (ABS CYR)))))
    (FID (AT R * *))
    (FAS (AT R XX YY))
    (FGN ((:= XX (+PLUS CXF (*TIMES CYR $)))
          (:= $ (QUO (*DIF XX CXF) CYR)))
        ((:= YY (+PLUS CYF (*TIMES CYR $)))
          (:= $ (QUO (*DIF YY CYF) CYR))))
    (IFS (GRASPING R A B)
        (YRATE R CYR)
        (YRATE R CYR)))
```

Hendrix World Scenarios (Part IV)

Figure 3.2

9) FILLBUCKET is nearly identical to the Lowrance version except it takes into account the gradually turning valve and adjusts its bucket-filling appropriately.

10 & 11) GRASP and RELEASE now include X-Y coordinates for the grasped/released object and robot.

12) MOVABILITY is identical to the Lowrance version.

13) GOTO is entirely different. It solves equations which compute distance, X-Y rates and X-Y positioning. This eliminates the need for a track and thus the robot can move directly to locations therby foregoing a lot of needless travel.

14) LOC is similar to the Lowrance version but is now activated by the X-Y rates of the robot. It has no CCN for as long as the robot moves, so does whatever it is grasping.

As a preface to the output demonstrating all scenarios in the Hendrix World Implementation it may be noted first that the dimensions of the world are represented by an X-Y grid of 200 by 200 units (APPENDIX D). Second, to facilitate the time consuming evaluation of TRACE output (control block creation/destruction) the following itinerary is provided.

ITINERARY

The robot will first GOTO the clock, GRASP it, SETALARM to 11, and then RELEASE it. Returning to his former position he will then take a snooze, awaken at 11, move to the clock, GRASP it, shut it off, and RELEASE it (time for work!). After moving to the bucket and GRASPing it he moves to the tap where he RELEASEs it. Travelling next to the valve he GRASPs this, turns it until the bucket is full, turns the valve off, and then RELEASEs it. Moving back to the tap he GRASPs the bucket again and moves to the buckets original position (very slowly so he won't spill any). After

RELEASing the bucket, his long day has drawn to an end so he returns to his "home" position and promptly falls asleep.

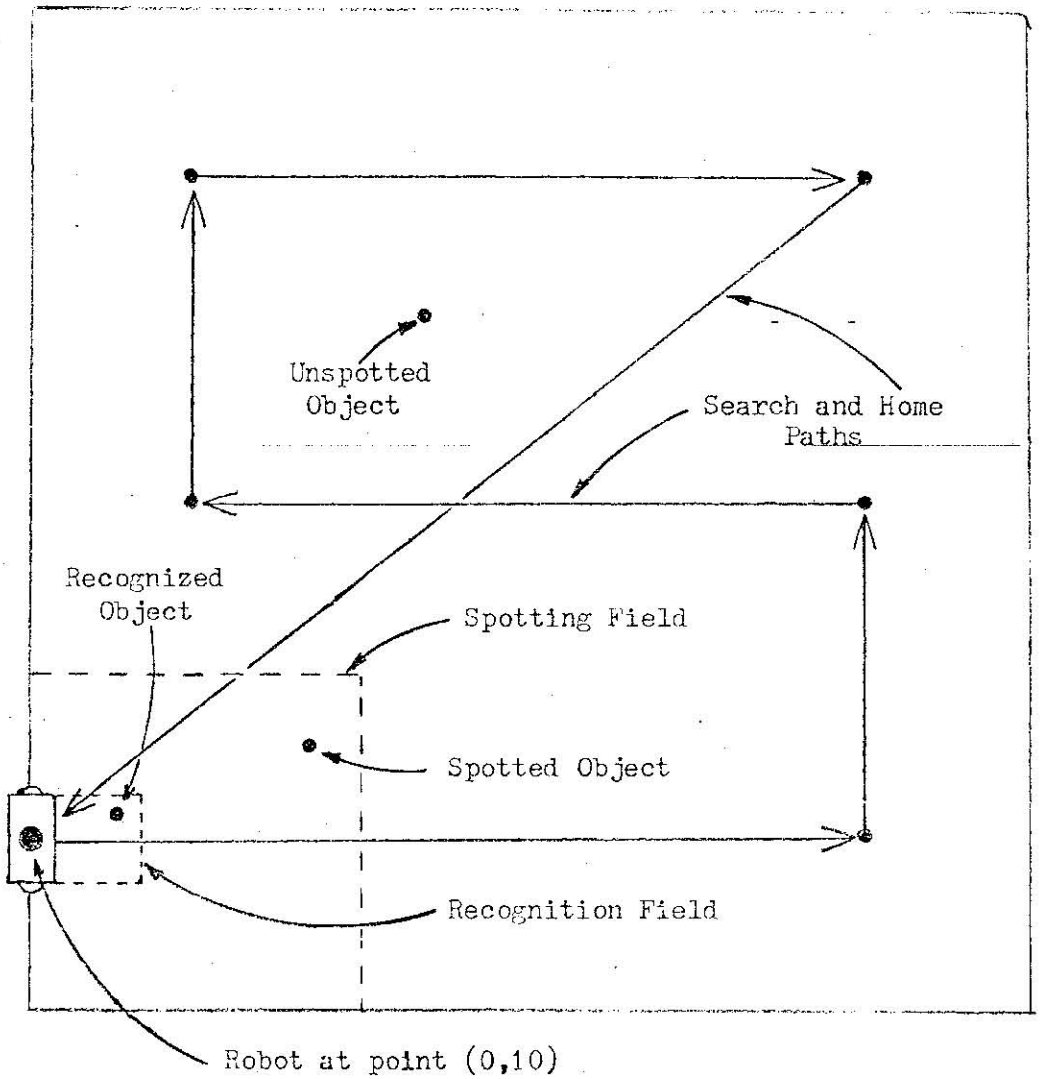
CHAPTER IV - NEW WORLD IMPLEMENTATIONS

The Robot Eye World

The Robot Eye World, though not in a sense a model of simulated vision, does resemble some of the capabilities of sight. This world can best be described as a "search and locate" system utilizing blind reference of the SWM by certain scenarios. Since any object found in the SWM could at any time be referenced immediately by a scenario, the scenarios in this world are designed so as to not violate this convenience. Just as human vision can neither spot nor recognize objects beyond a certain range likewise the robot in this world can't either.

The world is represented as a 60 x 60 foot X-Y axis (see Figure 4.1) where locational points are represented by cartesian coordinates. The Home or starting point of all searches is at point (0,10). The VISION of the robot is 20/20 or in terms of this simulation 20' forward and 10' to either side. This region symbolizes the area in which an object can be seen or "spotted". Any object outside this zone is ignored. A field of resolution identical to the field of vision (but smaller) is provided 5' forward and 2.5' to either side of the robot. This resolution or recognition field is the range within which any object can be recognized. All objects outside may be seen (if within field of vision) but not recognized.

Each object when placed in the world is given four distinct n-tuples. The first is (TYPE OBJ OBJECT) identifying the fact that it is an object. Next is its locational tuple (AT OBJ X Y) where X and Y are its cartesian coordinates. The third is its characteristic tuple (CHAR OBJ CHAR1) where CHAR1 could be any thing from color




```
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT ROB 0 10)
(STATE ROB NORMAL)
(ANGLE ROB 0)
(XRATE ROB 0)
(YRATE ROB 0)
(PATH ROB ON)
(OLD-PT ROB 0 10)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 20 20)
(PATH-PTS 0 10)
(PATH-ANG 0)
```

Robot Eye World and SWM

Figure 4.1

to shape to even radioactivity. This particular tuple is crucial in location for the robot. When commanded to search for a specific object, the robot is only told to look for some object with characteristic CHAR1. The fourth tuple (NEW OBJ) determines whether the object has been looked at already. This tuple changes to (OLD OBJ) after OBJ is looked at.

In order to cover the entire search field without missing any area. Scenarios have been set up to guide the robot systematically along the field at the very limits of his visual capabilities. Upon locating the desired object or running to the end of the field without locating it, scenarios return the robot to his Home position reporting either success or failure.

Travelling along the search path, the robot stops only when any object comes into view. At that point he exits the search path toward the object until it enters his field of resolution. Upon recognizing an object with the correct search characteristic, he has finished so he returns Home. Otherwise he returns to his last search path location to resume the search.

Before looking at the sample run it may help to briefly describe each scenario (see Figure 4.2).

1) SPOT is activated basically by the tuples starting with FIND, STATE, SPOTTING, and NEW. The command to FIND an object must of course be present. The robot must be in the STATE of SPOTTING (as opposed to the NORMAL STATE) a particular object at a given point (see MONITOR-SPOT). Lastly the object to spot must be NEW or previously unspotted. In addition, the rate of the robot must be zero (stopped) and the object must be in the field of vision. After initiation the tuples relating that the object is NEW and that the robot is SPOTTING are deleted and the tuples saying that OBJ is OLD and that the robot CANSEE the OBJ are added.

2) MONITOR-SPOT provides the monitoring for objects the

```
(SPOT (PAR R REYE OBJ CDX CDY CEX CEY : CANG CX CY CXR CYR CB CD)
     (IFS (FIND R *)
          (ANGLE R CANG)
          (STATE R SPOTTING)
          (TYPE R ROBOT)
          (AT R CX CY)
          (XRATE R CXR)
          (YRATE R CYR)
          (TYPE REYE ROBOT-EYE)
          (VISION REYE CB CD)
          (SPOTTING OBJ CDX CDY)
          (AT OBJ CDX CDY)
          (DIMENSION FIELD CEX CEY)
          (TYPE OBJ OBJECT)
          (NEW OBJ))
     (IFN (ZERO CYR)
          (ZERO CYR)
          (INFIELD CX CY CDX CDY CANG CB CD CEX CEY))
     (FIN (NEW OBJ)
          (SPOTTING OBJ * *))
     (FIN (CANSEE OBJ)
          (OLD OBJ)))

(MONITOR-SPOT (PAR R REYE CEX CEY OBJ CDX CDY : CANG CX CY CXR CYR
                                                CB CD EDIS)
     (IFS (FIND R *)
          (ANGLE R CANG)
          (TYPE R ROBOT)
          (FROM R CX CY)
          (XRATE R CXR)
          (YRATE R CYR)
          (TYPE REYE ROBOT-EYE)
          (VISION REYE CB CD)
          (DIMENSION FIELD CEX CEY)
          (AT OBJ CDX CDY)
          (TYPE OBJ OBJECT)
          (NEW OBJ))
     (IFN (NOT (AND (ZERO CXR) (ZERO CYR)))
          (INF CX CY CDX CDY CANG CB CEX CEY)
          (:= EDIS (DIST CDX CDY CANG CX CY CD)))
     (CON FUNC (PLUS (QUO EDIS 9) 4))
     (FIN (GOTO R * *)
          (STATE R *))
     (FIN (STATE R SPOTTING)
          (SPOTTING OBJ CDX CDY)))
```

Robot Eye World Scenarios (Part I)

Figure 4.2

```
(GOTO (PAR R CXT CYT : CXF CY  ED EXR EYR)
     (IC?(GOTO R CXT CYT)
          (AT R CXF CYF))
     (ICN(NOT(AND(EQUAL CXT CXF)(EQUAL CYT CYF)))
          (:= ED(SQRT(+PLUS(SQ(+DIF CXT CXF))(SQ(+DIF CYT CYF))))
          (:= EXR(QUO(*TIMES(+DIF CXT CXF)5)ED))
          (:= EYR(QUO(*TIMES(+DIF CYT CYF)5)ED)))
     (EIT(XRATE R *)
          (YRATE R *)
          (AT R * *))
     (EIA(XRATE R EXR)
          (YRATE R EYR)
          (FROM R CXF CYF))
     (EQS(AT R XX YY))
     (EGN((:= XX(+PLUS(*TIMES EXR $)CXF))
          (:= $(QUO(+DIF XX CXF)EXR)))
          ((:= YY(+PLUS(*TIMES EYR $)CYF))
          (:= $(QUO(+DIF YY CYF)EYR))))
     (ICS(GOTO R CXT CYT))
     (ICN FUNC(+PLUS(QUO ED 5)*))
     (EPT(XRATE R *)
          (YRATE R *)
          (FROM R * *)
          (GOTO R CXT CYT))
     (EPA(XRATE R 0)
          (YRATE R 0)))

(RESPOT (PAR R CTX CTY REYE OB CD OBJ CDX CDY CEX CEY : CANG
                                                    CX CY CXR CYR)
     (IC?(FIND R *)
          (PATH-POS CTX CTY)
          (ANGLE R CANG)
          (STATE R SPOTTING)
          (AT R CX CY)
          (XRATE R CXR)
          (YRATE R CYR)
          (VISION REYE OB CD)
          (SPOTTING OBJ CTX CTY)
          (AT OBJ CDX CDY)
          (DIMENSION FIELD CEX CEY)
          (NEW OBJ))
     (ICN(ZERO CXR)
          (ZERO CYR)
          (NOT(INFIELD CX CY CDX CDY CANG OB CD CEX CEY)))
     (EIT(SPOTTING OBJ * *)
          (STATE R *))
     (EIA(GOTO R CTX CTY)
          (STATE R NORMAL)))
```

Robot Eye World Scenarios (Part II)

Figure 4.2

```
(MOVETO (PAR OBJ R I DEX DEY   DDX DDY DX DY DANG DRES FANG EX EY) -
    (ICS(CANSEE OBJ)
        (AT OBJ DDX DDY)
        (PATH R ON)
        (TYPE R ROBOT)
        (AT R DX DY)
        (ANGLE R DANG)
        (TYPE I ROBOT-EYE)
        (DIMENSION FIELD DEX DEY)
        (RESOLUTION I DRES))
    (ICN(NOT(INFIELD DX DY DDX DDY DANG DRES DRES DEX DEY))
        (:= FANG(SLANG DX DY DDX DDY))
        (:= EX(XDIS DX FANG(BUCKET(DIFFERENCE(DIS DX DY DDX DDY)
                                              DRES -1))))
        (:= EY(YDIS DY FANG BUCKET)))
    (FID(ANGLE R *)
        (PATH R *)
        (OLD-PT R * *))
    (EIA(ANGLE R FANG)
        (PATH R DEF)
        (OLD-PT R DX DY)
        (GOTO R EX EY)))

(EXAMINE (PAR OBJ OBJ-CHAR R I : DDX DDY DX DY DANG DRES)
    (ICS(CANSEE OBJ)
        (AT OBJ DDX DDY)
        (CHAR OBJ OBJ-CHAR)
        (TYPE R ROBOT)
        (AT R DX DY)
        (ANGLE R DANG)
        (TYPE I ROBOT-EYE)
        (RESOLUTION I DRES))
    (ICN(INFS DX DY DDX DDY DANG DRES))
    (FID(CANSEE OBJ))
    (EIA(DFS(CREATED OBJ OBJ-CHAR))))

(RECOGNIZE (PAR R OBJ-CHAR OBJ : DX DY EDUM)
    (ICS(FIND R OBJ-CHAR)
        (DFS(CREATED OBJ OBJ-CHAR)
        (AT OBJ DX DY))
    (ICN(:= EDUM(SETQ SHIFT NIL)))
    (FID(FIND R OBJ-CHAR)
        (PATH R *))
    (EIA(CREATED OBJ-CHAR OBJ DX DY)
        (GOHOME R))))

(STOP-SEARCH (PAR DX DY DLX DLY R OBJ-CHAR)
    (ICS(PATH-POS DX DY)
        (LAST-POS DLX DLY)
        (FIND R OBJ-CHAR))
    (ICN(PROGE(SETQ EPSILON .001)(RETURN(AND
                (EQUAL DX DLX)(EQUAL DY DLY))))
    (FID(GOTO R * *)
        (FIND R *))
    (EIA(NOT(CREATED OBJ-CHAR OBJECT WITHIN FIELD)
        (GOHOME R))))
```

Robot Eye World Scenarios (Part III)

Figure 4.2

```
(SEARCH (PAR P : CX CY CPX CP  CB CD CDV CANG ENDANG EX EY)
    (IOS(FIND P *)
        (STATE P NORMAL)
        (PATH P DN)
        (AT P CX CY)
        (PATH-POS CPX CPY)
        (DIMENSION FIELD CB CD)
        (VISION EYE * CDV)
        (XRATE P 0)
        (YRATE P 0)
        (PATH-ANG CANG))
    (IGN(PROGF() (SETQ EPSILON .001) (RETURN(AND
                    (EQUAL CPX CX)(EQUAL CPY CY))))
        (:= ENDANG(BUCKET(SHIFT CANG)))
        (:= EX(CAR(BUCKET(ROUND CX CY BUCKET CB CD(OUT CDV P)))))
        (:= EY(CADR BUCKET)))
    (FIN(PATH P *)
        (PATH-POS * *)
        (LAST-POS * *)
        (ANGLE P *)
        (PATH-ANG *))
    (FIA(PATH P DN)
        (PATH-POS EX EY)
        (LAST-POS CX CY)
        (ANGLE P ENDANG)
        (PATH-ANG ENDANG)
        (GOTO P EX EY)))

(GOAHOME (PAR P REYE CB : CX CY EY EANG)
    (IOS(GOAHOME P)
        (TYPE P POINT)
        (AT P CX CY)
        (VISION REYE CB *))
    (IGN(:= EY(MID CB 2))
        (:= EANG(SLANG CX CY 0 EY)))
    (FIN(GOAHOME P)
        (LAST-POS * *))
    (FIA(GOTO P 0 EY)
        (LAST-POS CX CY)))

(REXSEARCH (PAR P : CRX CRY CX CY ENDANG)
    (IOS(FIND P *)
        (PATH P DEF)
        (AT P CRX CRY)
        (OLD-PT P CX CY)
        (XRATE P 0)
        (YRATE P 0))
    (IGN(PROGF() (SETQ EPSILON .001) (RETURN(OR(NE CRX CX)(NE CRY CY))))
        (:= ENDANG(SLANG CRX CRY CX CY)))
    (FIN(ANGLE P *))
    (FIA(GOTO P CX CY)
        (ANGLE P ENDANG)))
```

Robot Eye World Scenarios (Part IV)

Figure 4.2

```
(RESTART (PAR R : CX CY CDX C| . CXT CYT CANG)
    (IFS(FIND R *)
        (PATH-SNS CANG)
        (PATH R DEF)
        (AT R CX CY)
        (DLD-PT R CDX CDY)
        (PATH-POS CXT CYT))
    (ICN(OR(NE CX CXT)(NE CY CYT))
        (PROGF()(SETG EPSILON .001)(RETURN(AND(EQUAL CDX CX)
                                              (EQUAL CDY CY)))))
    (ELD(PATH R *)
        (STATE R *)
        (ANGLE R *))
    (ELA(GOTO R CXT CYT)
        (ANGLE R CANG)
        (STATE R NORMAL)
        (PATH R ON)))
```

Robot Eye World Scenarios (Part V)

Figure 4.2

```
(DE DEF(EX EY TX TY)(COND
    ((EQUAL EX TX)(*DIF EY TY))
    (T(*DIF EX TX))))

(DE SLANG(EX EY TX TY)(PROG(SL X)
    (SETQ SL (SLOPE(-(FIX EX EY TX TY)))
    (SETQ X(DEF EX EY TX TY))
    (COND((LT SL -1)(SETQ X(NEG X))))
    (RETURN(CONV(COND((EQUAL SAN(SETQ Y(THIN(*PLUS
    (COND((NULL SL)(SETQ X(NEG X))-90)(T(INVTAN SL)))
    (COND((LE X 0)360)(T 180))))360))))(T X))))

(DE DECP(CDX CDY CANG CX CY D)(DIED D(COND
    ((EQUAL CANG 90)(DIS CX CY CY CDY))
    (T(DIS CX CY CDX CY)))

(DE DIED(A B)(COND
    ((LT B* D)A)
    (T(DIFFERENCE B A -1)))

(DE IN2(CX CY CDX CDY CANG CR CEX CEY)(COND
    ((EQUAL CANG 90)(INFIELD CX CY CDX CDY CANG CR CEX CEX CEY))
    (T(INFIELD CX CY CDX CDY CANG CR CEX CEX CEY))))

(DE CANR(L N)(COND
    ((ONEP N)(CAR L))
    (T(CONS(CAR L)(SUB1 N)))))
```

Robot Eye World Help Functions (Part I)

Figure 4.3

```
(SETQ SHIFT NIL)

(DF SHIFT(A)(PROG()
     (COND((NULL SHIFT)(SETQ SHIFT -1)(RETURN 0))
     ((AND(OR(EQUAL A 180)
              (EQUAL A 0))(SETQ SHIFT(NEG SHIFT))))
     (RETURN(*PLUS A (*TIMES SHIFT 90)))))

(DF BOUND(X Y A FX FY DV)(COND
     ((EQUAL A 90)(LIST X(COND
          ((GE(*PLUS Y(*TIMES DV F))(*DIF FY DV))(*DIF FY DV))
          (T(*PLUS Y(*TIMES DV F))))))
     ((EQUAL A 180)(LIST(*PLUS 0 DV)Y))
     (T(LIST(*DIF FX DV)Y))))

(DF INFRS(X Y DX DY A R)(PROGF()(SETQ EPSILON .01)
     (RETURN(OR
          (AND(LE(DIS X Y DX DY)R)
               (EQUAL(CLANG X Y DX DY)A))
          (INFIELD X Y DX DY A R F 60 A0)))))

(DF INFIELD(X Y AX AY A CB CD FX FY)(COND
     ((OR(NOT(WITHIN FX AX 0))(NOT(WITHIN FY AY 0)))NIL)
     ((EQUAL(THIN A 180)0 180)(AND(WITHIN CD(ABS(*DIF AX X)))0)
                              (WITHIN(QUO CB 2)(ABS(*DIF AY Y)))0)))
     (T(AND(WITHIN CD(ABS(*DIF AY Y)))0)
          (WITHIN(QUO CB 2)(ABS(*DIF AX X)))0))))
```

Robot Eye World Help Functions (Part II)

Figure 4.3

searching robot will come across. Initiation indicates a potentially Spottable object ahead in the direction of the search path and will through its CCN activate the appropriate interrupt time for GOTO. Stopped with the object just within vision the robot then enters the SPOTTING STATE (in turn activating SPOT).

3) GOTO remains nearly identical to that in the Hendrix World with the deletion of certain superfluous initiation conditions.

4) RESPOT is initiated in the event that two potential objects activate MONITOR-SPOT (one further away). The interrupt time for the first allows the robot to travel and recognize it. This time change when added to the interrupt time for the second object will place the robot short of the SPOTTING range for this second object thus necessitating reinitiation of the search (inturn activating a new updated MONITOR-SPOT).

5) MOVETO is activated when SPOT responds with a tuple stating the robot CANSEE an object. It inturn moves (GOTO) the robot toward the object and stops it when the object is within the resolution field.

6) EXAMINE is activated when the robot both CANSEE an object and when that object is within the resolution field. It deletes the CANSEE tuple and adds a DESCRIBED tuple. It is important to note that this is the only scenario that is allowed (through its ICS) to access the characteristic tuples associated with the search characteristic thus providing the realistic vision abilities.

7) RECOGNIZE can only be initiated after EXAMINE has DESCRIBED an object and its characteristic is identical to the search characteristic. When this occurs the search is over, the FIND command tuple is deleted, and a tuple stating that the robot has LOCATED the appropriate object is added. Finally, a GOHOME tuple is added (see GOHOME scenario).

8) STOP-SEARCH occurs when the end of the field is encountered without finding the search object. A tuple stating that the robot has NOTLOCATED the object is added and GOHOME is activated.

9) SEARCH will initiate a search from HOME position (with GOTO) when the FIND tuple is found in the SWM. In addition it will turn the robot and send him in a new search direction when he has come to the edge of the field.

10) GOHOME simply needs the GOHOME tupe to initiate a direct GOTO to Home position (Ø,1Ø).

11) RESEARCH sends the robot back to the original search position from which he left to EXAMINE an object.

12) RESTART is activated after the robot EXAMINEs an object and returns to the original search path position. Using tuples stored upon leaving the search path (in MOVETO), the robot is sent in the same direction as was previously established by SEARCH.

The associated help functions used in some of the above scenarios (see Figure 4.3) are briefly described. All other functions not mentioned are either LISP system functions or can be found in AUXFUN (APPENDIX G).

(DEP FX FY TX TY) - Used by SLANG (see below) to provide directional information about the points from (FX,FY) to (TX,TY).

(SLANG FX FY TX TY) - Given the points from (FX,FY) to (TX,TY) this returns the angle (in degrees) based upon the slope of the line between the two points.

(DEC2 COX COY CANG CX CY D) - When the robot is facing CANG degrees at point (CX,CY) and a search object is at (COX,COY), the depth D of the robot vision, determines how far the robot must travel forward until the object is within vision. This distance is what is returned.

(DIFD D DS) - Is used by DEC2 to determine if the object is already in the vision field in which case the distance to travel is Ø; otherwise it returns the distance that will place the search object just inside the far edge of the robot vision.

(IN2 CX CY COX COY CANG CB CFX CFY) - This function calls INFIELD with identical arguments changing only the depth of vision to the full extent of the search field. This is used by MONITOR-SPOT to determine possible objects ahead of the robot vision.

(CANR L N) - Is simply CAnR for n = 1,...,n n > Ø (e.g. CADR, CADDR, CADDDR,...).

(SETQ SHIFT NIL)
(SHIFT A) - When given angle A, the function returns the

new search angle (A = Ø, 18Ø, 9Ø, Ø,...).

(BOUND X Y A FX FY DV) - Given robot position (X,Y), new search angle A, dimensions FX and FY of the search field, and the depth of the robot vision, this function returns the X-Y coordinates that the robot is BOUND to hit at the edge of the serach field.

(INRES X Y OX OY A R) - Returns T if the robot is facing angle A at (X,Y) with an object at (OX,OY) within resolution field of depth of R; otherwise it returns NIL.

(INFIELD X Y AX AY A CB CD FX FY) - With the robot at (X,Y) facing angle A (A = multiples of 90° only), the search object at (AX,AY), the vision or resolution field of dimension CB x CD and the search field FX x FY if (AX,AY) is within the CB x CD field and within the FX x FY field the function returns T; otherwise it returns NIL.

Executions of the Robot Eye World - Summary (see APPENDIX C)

In Run No. 1 only one object with characteristic RED is placed in the search field. To find, the robot searches to the right along the bottom, turns up along the right edge to about half way, and then moving left he locates the object. After locating he travels Home.

In Run No. 2 a block and a sphere are placed along the bottom. This is to demonstrate the simultaneous initiations of MONITOR-SPOT (with different interrupt times). The object-characteristic searched for is the sphere so the first examination will fail causing search continuation. After recognizing the sphere, the robot goes Home.

In Run No. 3 four objects a bicycle, automobile, motorcycle, and female-robot are used. All are placed within the field except the female-robot which is placed just outside the field in the upper right corner. Although (of course) the robot never finds the female-robot, inadvertantly a few interesting results occur (coincidentally) very near the area she is at. After examining the bicycle and automobile the robot turns in the direction of the motorcycle and

female-robot. The interrupt time provided by the CGN in MONITOR-SPOT is given and the robot stops, turns and moves toward the motorcycle. Unfortunately the angle returned by SLANG to move toward turned out to be a bit inaccurate and EXAMINE was never activated. Continuing the search, the robot moves immediately toward the female-robot. Since this is the end of the search path, he stops. Still seeing the motorcycle the robot again moves toward it and now with a less acute angle, SLANG returns a more accurate result and EXAMINE is successfully initiated. Returning to continue the search (but at both the end of the search path and the end of the field) RESTART would not reinitiate the search because it had ended and SEARCH would not be initiated without RESTART adding the fact that the robot was back on the path. This had to be done by hand.

Two ramifications, though purely accidental but interesting, stem from this. First, the system overcame an inaccuracy of one equation solver by providing a position with which the equation solver could respond accurately. Second, although the fact that the female-robot was near by was sheer luck, it was interesting to see the robot stubbornly stop at the closest point possible to her and refuse to GOHOME!

The Billiards World

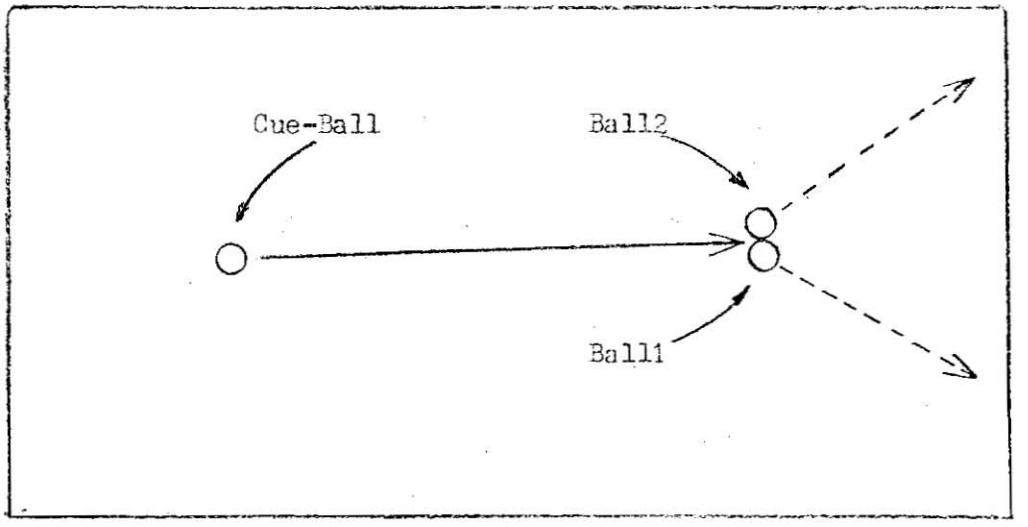
The Billiards World is a simulation of certain basic actions found in Regulation Billiards (not Pocket Billiards). Though in this version scoring is not involved it may be noted that Billiards is generally played with 3 balls (cue-ball and two scoring balls). In order to score a player must strike 1 or both of the scoring balls with the cue-ball while first bouncing it off of $\emptyset$, 1, 2, or 3 banks (depending upon the version) gaining various points according to each version.

The range of this simulation covers:

1) movement of a ball
2) decay in acceleration of movement
3) derivation of bounce point on bank
4) subsequent course and angle change after bounce
5) replusion of ball (with excessive speed) off the table
6) collision of any moving ball with another still ball (provided collision is eminent)
7) subsequent transfer and reduction of inertia between colliding balls
8) subsequent course change (with respect to percentage off-center) between colliding balls

The Regulation Billiards table of dimensions 304.80 cm. x 154.40 cm. is represented by an X-Y axis (see Figure 4.4). The balls of diameter 6.0325 cm. are represented and activated through 5 similar n-tuples for each. Though not all balls are used in each demonstration, these tuples are identical for all. Any ball is located (AT BALL X Y) where (X,Y) would be its cartesian coordinate. The STATE of the ball tuple can represent the states MOVING, STOPPED, or LIMBO(a transition state where the ball is either hitting a ball or bank or off the table).

One tuple tells whether the ball is ON or OFF the table, another tells (RATE BALL n) where $n \geq \emptyset$ is the MOVING rate of the ball.




```
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325)
(SPEED BALL 100)
(DIMENSION TABLE 304.80 152.40)
 AT CUE-BALL 75 75)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
```

Billiards World and SWM

Figure 4.4

The final tuple tells whether the ball is BEHIND or NOTBEHIND a ball it has just hit (or been hit by). This last tuple is necessary if the ball BEHIND will stop sooner than the ball it has just hit (which is usually the case since the front ball gains its momentum and starts across the table while the rear ball now slowed rolls to a stop). If this were not included the rear ball would have its movement-control block destroyed before the front ball and would pass through the front ball and stop ahead of it (only to be hit by the ball it just hit!) In early runs two balls were seen passing through each other and rehitting each other across the table repeatedly until momentum was lost.

A monitoring scenario MAYHIT (similar to MONITOR-SPOT in the Robot Eye World) activates interrupts in ball movement at crucial points where potential collisions may occur. At interrupt time other scenarios (HIT and NOHIT) check whether a collision has occured (or perhaps whether the ball has slowed and stopped short of collision).

A brief summary of each scenario is given below (see Figure 4.5).

1) HIT is activated basically after MAYHIT has been destroyed. It requires two tuples supplied by this scenario. These tuples OFFCENTER and MAYHIT supply information to HIT in the event it is actually activated. This happens when the balls in question have points within the diameters of the two balls. In this case two new rates are computed for the new movement of these two balls. These rates are based upon the old rate of the moving ball and the percentage OFFCENTER they have hit. In deciding whether the balls would collide, MAYHIT had already computed the new angles after collision and this is communicated in the OFFCENTER tuple. The two balls are then shot (by the SHOOT scenario) at the appropriate directions and rates. The CCS of HIT continues until the front ball has lost its SHOOT command. This is necessary because the tuple BEHIND is added in connection with the rear ball so as to not reinitiate MAYHIT in the event the rear ball stops ahead of the front ball.

```
(HIT (PAR B CDAM : B2 CX CY CR  CBY CRAT CA1 CA2 COFF ERAT1 ERAT2)
    (ICS(STATE B LIMBO)
        (MAYHIT B B2 CX CY)
        (AT B CBX CBY)
        (DIAMETER BALL CDAM)
        (RATE B CRAT)
        (OFFCENTER B B2 CA1 CA2 COFF))
    (ICN(LE(DIS CX CY CBX CBY)CDAM)
        (:= ERAT1(CAR(BUCKET(PROPSPD CDAM(ABS COFF)CRAT))))
        (:= ERAT2(CADR BUCKET)))
    (EID(MAYHIT B B2 * *)
        (NOTBEHIND B)
        (RATE B *)
        (OFFCENTER B B2 * * *))
    (EIA(SHOOT B ERAT1 CA1)
        (HASHIT B B2 CX CY)
        (RATE B 0)
        (BEHIND B)
        (SHOOT B2 ERAT2 CA2))
    (CCS(SHOOT B2 ERAT2 CA2))
    (EPD(BEHIND B)
        (HASHIT B B2 * *))
    (EPA(NOTBEHIND B)))

(NOHIT (PAR B STAT CDAM : B2 CX CY CBX CBY)
    (ICS(STATE B STAT)
        (MAYHIT B B2 CX CY)
        (AT B CBX CBY)
        (DIAMETER BALL CDAM))
    (ICN(OR(EQF STAT LIMBO)(EQF STAT STOPPED))
        (GT(DIS CX CY CBX CBY)CDAM))
    (EID(MAYHIT B B2 * *)
        (OFFCENTER B B2 * * *)))

(OFF-THE-TABLE (PAR B CANG CSPD CDAM : CBX CBY CRAT)
    (ICS(ATWALL B CANG)
        (STATE B LIMBO)
        (AT B CBX CBY)
        (SPEED BALL CSPD)
        (RATE B CRAT)
        (DIAMETER BALL CDAM)
        (ON B TABLE))
    (ICN(GT CRAT CSPD))
    (EID(ATWALL B *)
        (ON B TABLE)
        (AT B * *)
        (RATE B *))
    (EIA(OFF B TABLE)))

(OFF-THE-WALL (PAR B CR : CR CS)
    (ICS(ATWALL B CR)
        (STATE B LIMBO)
        (RATE B CR)
        (SPEED BALL CS))
    (ICN(LE CR CS))
    (EID(ATWALL B *))
    (EIA(SHOOT B 0 CA)))
```

Billiards World Scenarios (Part I)

Figure 4.5

```
(SHOOT (PAR B CSPD CANG : CX (-: TAB CRAT ERAT EX EY EDIS CDAM)
    (ICS(SHOOT B CSPD CANG)
        (AT B CX CY)
        (ON B TAB)
        (TYPE TAB BILLIARDS)
        (RATE B CRAT)
        (DIAMETER BALL CDAM))
    (ICN(WITHIN 0 CANG 360)
        (:= ERAT(*PLUS CSPD CRAT))
       (:= EX(CAR(BUCKET(WALL CX CY CANG))))
        (:= EY(CADR BUCKET))
        (:= EDIS(*DIF(DIS EX EY CX CY)(QUO CDAM 2))))
    (EID(RATE B *)
        (STATE B *)
        (AT B * *))
    (EIA(ROLLING B EX EY CANG)
        (STATE B MOVING)
        (FROM B CX CY ERAT))
    (EGS(RATE B YRAT)
        (AT B YX YY))
    (EGN(:= YRAT(DECAY ERAT $))
         (:= $(GROW EDIS ERAT)))
        ((:= YX(CAR(BUCKET(LOOP EX CY CANG $ # ERAT EDIS))))
         (:= $ $))
        ((:= YY(CADR BUCKET))(:= $ $)))
    (OCS(ROLLING B EX EY CANG))
    (OCN FUNC(*PLUS(GROW EDIS ERAT)#))
    (EPD(STATE B *)
        (FROM B * * *)
        (SHOOT B CSPD CANG))
    (EPA(STATE B LIMBO)))

(MAYHIT (PAR B1 CX CY CANG B2 C2X C2Y CDAM : CFX CFY CRAT
                                             EDS EOFF EANG1 EANG2)
    (ICS(ROLLING B1 CX CY CANG)
        (FROM B1 CFX CFY CRAT)
        (AT B2 C2X C2Y)
        (RATE B2 0)
        (NOTBEHIND B2)
        (DIAMETER BALL CDAM))
    (ICN(ONLINE C2X C2Y CFX(BUCKET(SLOPE(LIST CFX CFY CX CY)))
             (Y-INT CX CY BUCKET)(QUO CDAM 2))
        (:= EDS(*DIF(DIS C2X C2Y CFX CF))(QUO CDAM 2)))
        (:= EOFF BUCKET)
        (:= EANG1(CAR (BUCKET(OFFLINE CANG EOFF CDAM))))
        (:= EANG2(CADR BUCKET)))
    (OCN FUNC(GRO(*PLUS(GROW2 EDS CRAT)#)))
    (EPD(ROLLING B1 CX CY CANG))
    (EPA(MAYHIT B1 B2 C2X C2Y)
        (OFFCENTER B1 B2 EANG1 EANG2 EOFF)))
```

Billiards World Scenarios (Part II)

Figure 4.5

```
(BOUNCE (PAR B CX CY CANG : CBX CBY CDAM EANG)
    (ICS(ROLLING B CX CY CANG)
        (AT B CBX CBY)
        (DIAMETER BALL CDAM))
   (ICN(ENUF CBX CBY CX CY(*PLUS(QUO CDAM 2).1))
        (:= EANG(MIRROR CANG CX CY)))
    (EID(ROLLING B * * *))
    (EIA(ATWALL B EANG)))

(STOP-ROLL (PAR B : CRAT)
    (ICS(STATE B LIMBO)
        (RATE B CRAT))
    (ICN(EQUAL CRAT 0))
    (EID(STATE B LIMBO)
        (ROLLING B * * *))
    (EIA(STATE B STOPPED)))
```

Billiards Worlds Scenarios (Part III)

Figure 4.5

```
(SETQ TMM NIL)
(SETQ DSS NIL)

(DE GRO(TM)(SETQ TMM TM))

(DE GROW2(D P)(PROG2(SETQ DSS D)(GROW D P)))

(DE WALL(X Y D)(PROG(M)(SETQ M(SLOPE D))
    (RETURN(ZERNIL(DECIDE
        (X-INT X Y M)
        (Y-INT X Y M)
        (X-INT X(*DIF Y 152.4)M)
        (Y-INT (*DIF X 304.8)Y M D)))))

(DE DECIDE(A B C D E)(COND
    ((WITHIN 0 E 90)(LEGIT NIL NIL C D))
    ((WITHIN 90 E 180)(LEGIT NIL B C NIL))
    ((WITHIN 180 E 270)(LEGIT A B NIL NIL))
    (T(LEGIT A NIL NIL D))))

(DE LEGIT(A B C D)(COND
    ((AND A(LX A))(LIST A D))
    (A(COND(B(LIST 0 B))(T(LIST 304.8 D))))
    ((AND B(LY B))(LIST 0 B))
    ((OR B(LX C))(LIST C 152.4))
    (T(LIST 304.8 D))))

(DE ZERNIL(L)(MAPCAR 'ZIL L))

(DE ZIL(A)(COND(A A)(T 0)))
```

Billiards World Help Functions (Part I)

Figure 4.6

```
(DE LX(N)(COND((NULL N)NIL)(T (WITHIN 304.2 N 0))]

(DE LY(N)(COND((NULL N)NIL)(T (WITHIN 152.4 N 0))]

(DE Y-INT(X Y M)(COND((NULL M)NIL)(T(*DIF Y(*TIMES M X)))]

(DE X-INT(X Y M)(COND
    ((NULL M)X)
    ((EQUAL M 0)NIL)
    (T(NEG(QUO(Y-INT X Y M)M)))]

(DE DIS(X Y Z W)(SQRT
    (*PLUS(SQ(*DIF X Z))(SQ(*DIF Y W)))]

(DE MIRROR(A X Y)(COND
    ((AND(EQALL X 0 304.2)(EQALL Y 0 152.4))(THIN(*PLUS 180 A)360))
   ((EQALL X 0 304.2)(THIN(*DIF 180 A)360))
   (T(THIN(*DIF 360 A)360)))]

(DE FIN(V)(PROG2(SETQ TMM NIL)V(SETQ DSS NIL)))]

(DE LOOP(X Y A TM TM# R DIS)(COND
    ((AND(NOT(ZEROP(DECAY R TM)))
        (EQUAL TM# TMM))(RETURN(FIN(XYDIS X Y A DSS))))
    ((OR(NE(GROW DIS R)TM)
        (ZEROP(DECAY R TM)))(XYDIS X Y A(DIZ TM R 0)))
    (T(XYDIS X Y A DIS)))]

(DE DIZ(TM R TD)(COND
    ((AND(ZEROP TM)(ZEROP TD))0)
    ((WITHIN 0 TM 1)(*TIMES(AVG(DECAY R TD)(DECAY R(*PLUS TD TM)))TM))
    (T(*PLUS(AVG(DECAY R TD)(DECAY R(ADD1 TD)))
        (DIZ(SUB1 TM)R(ADD1 TD))))]

(DE AVG(A B)(QUO(*PLUS A B)2))]

(DE DECAY(R TM)(PROG(RT TM2)
    (SETQ RT R)(SETQ TM2 TM)
    (REPEAT WHILE (GE TM2 1)
        (SETQ RT(*TIMES RT .60))
        (SETQ TM2(SUB1 TM2))
        UNTIL (LE RT 1))
    (COND((NE TM2 0)
          (SETQ RT(*TIMES(*PLUS(*DIF .40(*TIMES TM2 .40))).60)RT))))
    (RETURN(COND((LE RT 1)0)(T RT)))]

(DE ONLINE(X Y FX M B EP)(PROGE()
    (SETQ EPSILON EP)
    (COND((NULL M)(RETURN(EQUAL 0(BUCKET(*DIF FX X))))))
    (RETURN(EQUAL 0(ABS(BUCKET(*DIF(*PLUS(*TIMES M X)B)Y)))))]

(DE OFFLINE(A B C)(LISNEG(SNUM B)
    (*PLUS(*TIMES(QUO 180 C)B)A))]
```

Billiards World Help Functions (Part II)

Figure 4.6

```
(DE LISNEG(B N)(LIST(THIN N B)(THIN(*DIF B)(*TIMES B 90))360))]

(DE ENUF(X Y A B EP)(PROGE()
    (SETQ EPSILON EP)
    (RETURN(AND(EQUAL X A)(EQUAL Y B)))]

(DE PROPSPD(D O R)(LEFTOV R
        (TIMES(OUO(*DIF(OUO D 2)O)(OUO D 2)R .90))]

(DE LEFTOV(R F)(LIST(*DIF R F)F)]

(DE CROSS(X1 Y1 X2 Y2 A1 A2)(PROG(D)
    (SETQ D(INVTAN(SLOPE(LIST X1 Y1 X2 Y2))))
    (SETQ D(FIXX(LIST(*DIF D A1)(*DIF D A2))))
    (SETQ D(OUO(*TIMES(DIS X1 Y1 X2 Y2)(SIN(CAR D))(
            (SIN(*DIF 180(*PLUS(CAR D)(CADR D))))))
    (RETURN(LIST(XDIS X1 A1 D)(YDIS Y1 A1 D)))]

(DE FIXX(L)(DIV(THIN(CAR L)180)(THIN(CADR L)180))]

(DE DIV(A B)(COND
    ((GT A 90)(LIST(*DIF 180 A)B))
    ((GT B 90)(LIST A(*DIF 180 B)))
    (T(LIST A B)))]

(SETQ GLO NIL)

(DE GROW(D R)(PROG(DZ H N)
    (COND((ZEROP R)(RETURN 0)))
    (SETQ N 1)(SETQ H 0)
    (REPEAT (SETQ DZ(DIF N R 0))
        UNTIL (EQUAL H DZ)
        WHILE (LT DZ D)
            (SETQ H DZ)(SETQ N(ADD1 N)))
    (COND((NE H DZ)(SETQ N(SUB1 N))))
   (RETURN(*PLUS N(PARTZ(ABS(*DIF D(DIF N R 0)))(DECAY R N))))]

(DE PARTZ(D R)(COND
    ((ZEROP R)0)
    (T(OUO D R)))]
```

Billiards World Help Functions (Part III)

Figure 4.6

2) NOHIT like HIT is also activated by MAYHIT. This activation occurs only if the balls are not within the distance of their diameters. If this occurs then the information supplied by MAYHIT (for a potential collision) are deleted.

3) OFF-THE-TABLE is initiated basically by the BOUNCE scenario (the ball either BOUNCEs OFF-THE-TABLE or OFF-THE-WALL). The tuple involved relates that a ball is ATWALL. The ball is in the STATE LIMBO (that instant between which the system changes its direction). In this case the ball is travelling above an allowable limit dictated by (SPEED BALL n). Since this is indeed the case the ball remains in LIMBO and it goes from ON to OFF the table. Its RATE and AT tuples are also deleted.

4) OFF-THE-WALL is initiated identically as OFF-THE-TABLE with the exception that it is within its legal BOUNCE speed. The tuple ATWALL added by the BOUNCE scenario conveniently includes the new angle which it will now travel. The ball is then simply shot (with SHOOT) in that angle.

5) SHOOT handles any activation of ball movement. The ball needs only to be on the table with the SHOOT command tuple present. This scenario is similar to GOTO (in the Hendrix World) except that a gradually decaying rate is included. The function DECAY is used to derive the decay of the rate given any rate and elapsed time. GROW which is used in CCN is an estimating function which loops with DECAY until it becomes zero while simultaneously adding up seconds (to a fraction) of elapsed time close to actual decay time. I say close because these are not physics based formulas in the least, they are simply quick and easy divisional formulas to demonstrate <u>some</u> decay. Depending upon back ground and ability, a user could simply replace these functions with precise physics based formulas returning exact answers. This is also the case with LOOP, a function which determines the distance the ball travels given elapsed time and initial rate. LOOP is used to place the ball the proper distance after travel. When the CCN deactivates this scenario the STATE of the ball changes from MOVING to LIMBO. 1 of 3 things has happened upon deactivation. The ball has stopped (see STOP-ROLL), the ball is waiting to bounce, or the ball has collided.

6) MAYHIT uses the tuple ROLLING provided by SHOOT to decide through ICN equations whether the present ROLLING angle will bring it in line with another

ball. If this is the case the scenario is activated and its CCN interrupts SHOOT at a predetermined time to allow HIT or NOHIT to operate. Angles the ball (and its colliding partner) will go after collision are computed according to how far OFFCENTER the stopped ball is in relation to the line upon which the rolling ball is travelling.

7) BOUNCE is activated simply through SHOOT's CCN. SHOOT stops the ball at the time it would hit a bank and subsequently places it in LIMBO. BOUNCE simply checks its stopped location (to within half the diameter of the ball) with the appropriate point on the wall. If this holds true, then an ATWALL tuple giving a (ICN computed) new bounce angle for OFF-THE-WALL (or OFF-THE-TABLE).

8) STOP-ROLL simply checks if a ball is in LIMBO and has a zero rate. If this is the case then it simply changes the STATE to STOPPED.

Although not immediately evident, this relatively simple looking packet of scenarios is backed by a complicated series of LISP functions. In order to understand the effects of these scenarios it is important to summarize these functions (see Figure 4.6).

(GRO TM) - Is a function used to set a global time variable TMM (used in LOOP).

(GROW2 D R) - Is simply a call to (GROW D R) which first sets a global distance variable DSS to D (used in FIN).

(WALL X Y D) - Incorporates the dimensions of the billiards table in the code. Given the (X,Y) coordinate and angle D it computes what point on some bank a ball travelling from that point will hit.

(DECIDE A B C D E) - This help function accepts possible intercepts (A B C D) from the function WALL. Using angle E it filters out unlikely candidates and passes the results to LEGIT.

(LEGIT A B C D) - This help function uses some predetermined combination (from DECIDE) of two NIL's and two wall-intercept candidates to eventually pick between the latter two returning what is ultimately the correct bank position needed in WALL.

(ZERNIL L) & (ZIL A) - Are used together to change any NIL element in L to $\emptyset$.

(LX N) & LY N) - Determine whether N is a legal X or Y coordinate within ∅ and table dimensions.

(Y-INT X Y M) - Returns the Y-Intercept of the line of slope M traveling through the point (X,Y).

(X-INT X Y M) - Returns the X-Intercept of the line of slope M travelling through the point (X,Y).

(DIS X Y Z W) - Returns the distance between points (X,Y) and (Z,W).

(MIRROR A X Y) - Returns the angle a ball will bounce off a wall given the initial angle and its bounce point (X,Y). This angle is similar to that made when a flashlight is shone in a mirror.

(FIN V) - Is called by LOOP after determining a value. This function returns that value while setting globals TMM and DSS to NIL.

(LOOP X Y A TM TM# R DIS) - This function can do 1 of 3 things. It can return the coordinates at global distance DSS from (X,Y), the same coordinate for distance computed by DIZ or the coordinate at distance DIS.

(AVG A B) - Averages A and B.

(DECAY R TM) - Iteratively decreases R (rate) by a constant until either R or TM are zero. If TM is zero the remaining rate is returned; otherwise ∅.

(ONLINE X Y FX M B EP) - This function uses PROGE to save EPSILON while it resets it to EP. It then returns T if the coordinate Y (of (X,Y) location for a collidable ball) is equal to (within EP) the Y coordinate computed (with X) in the line of slope M and Y-intercept B; otherwise NIL.

(OFFLINE A B C) - Returns a list of two new angles (for colliding balls) when A is the initial angle, B the percentage offcenter, and C the ball diameter.

(ENUF X Y A B EP) - Simply checks whether both X = A and Y = B where EPSILON is reset to EP.

(PROPSPD D O R) - Returns the new speeds divided between colliding objects. R is the rate of the rolling ball, D is the diameter of the ball, and O is the percentage offcenter.

(LEFTOV R F) - Simply determines the speed left over after a ball, run into, takes away fraction F of the initial rate R.

(GROW D R) - This function iteratively expands the time (in seconds) that R is taking to decay to zero within the distance D (without surpassing D with the total distance that rate R allows a ball to travel in that time).

(PARTZ D R) - Is called when less than 1 second will decay R to zero (from GROW). That fraction is returned.

S-Variables

A new entity called an S-Variable has been utilized in the NOHIT scenario. Similar to C-, E-, and Y-Variables, S-Variables begin with the letter S. Where C-Variables represent numerical quantities used in the ICN, S-Variables represent symbolic quantities also used in the ICN. In the case of NOHIT both the STATE LIMBO and STOPPED could initiate the scenario. To eliminate the writting of two nearly identical scenarios a variable STAT is bound up to either LIMBO or STOPPED. Since STAT has no LISP binding the ICN function EQ would cause an error message when checking (OR(EQ STAT LIMBO)(EQ STAT STOPPED)). To avoid this a new function (EQF FEXPR) which simply performs an EQ with unevaluated variables is used. The logical extension to this is an addition of E-Variables which are assigned quoted values and then used in ADDed tuples in EIA and EPA to represent state or other symbolic entities. An example of this type of idea can be utilized through the function OFF (see Figure 4.10 and description in the Robot Arm World).

Execution of the Billiards World - Summary (see APPENDIX E)

Beginning at a basic break position, the cue-ball is shot alone from that point to demonstrate the decay of the rolling rate. STOP-ROLL is initiated (once halted) to remove the LIMBO state. Next the cue-ball is shot from the stop point toward a bank with a rate deliberately too high. BOUNCE is activated but OFF-THE-TABLE detects

the velocity violation and removes the ball from the table.

Replacing the ball on the table at break position the ball is shot toward a bank this time with a lower velocity. Again BOUNCE is activated and OFF-THE-WALL completes the bounce and activates SHOOT again with a new (slower) rate and mirrored angle. The ball is next shot from stop position toward a corner to demonstrate the capabilities of double banking.

With the ball now replaced in break position, a second ball is included, at the other end. The cue-ball is sent toward the other ball dead center. They collide and the second ball begins rolling on the same angle. The second ball rebounds directly back from the bank while the cue-ball slowsto a halt. Returning on the same line the second ball hits the cue-ball and sends it back toward where it came from.

Next the cue-ball is shot on collision course again except slightly off angle. This collision produces a response of both balls rolling in different directions. The cue-ball rolls to a stop short of the wall while the second ball (with more momentum) bounces off the wall.

Adding a third ball, the cue-ball is returned to break position and the other two balls are racked together for a break shot shooting the cue-ball at just the right angle results in collision with BALL1 and a richochet into BALL2 (a 2 point shot!).

Finally, at command level, by hand the help function SLANG is used to determine the precise angle from the cue-ball to one of the balls. This shot is then made accurately. This hints at the possibility of developing equation solvers for the command level which, when given ball locations, could return angles for double hit shots or even single, double, and triple bank double hit shots.

The Robot Arm World

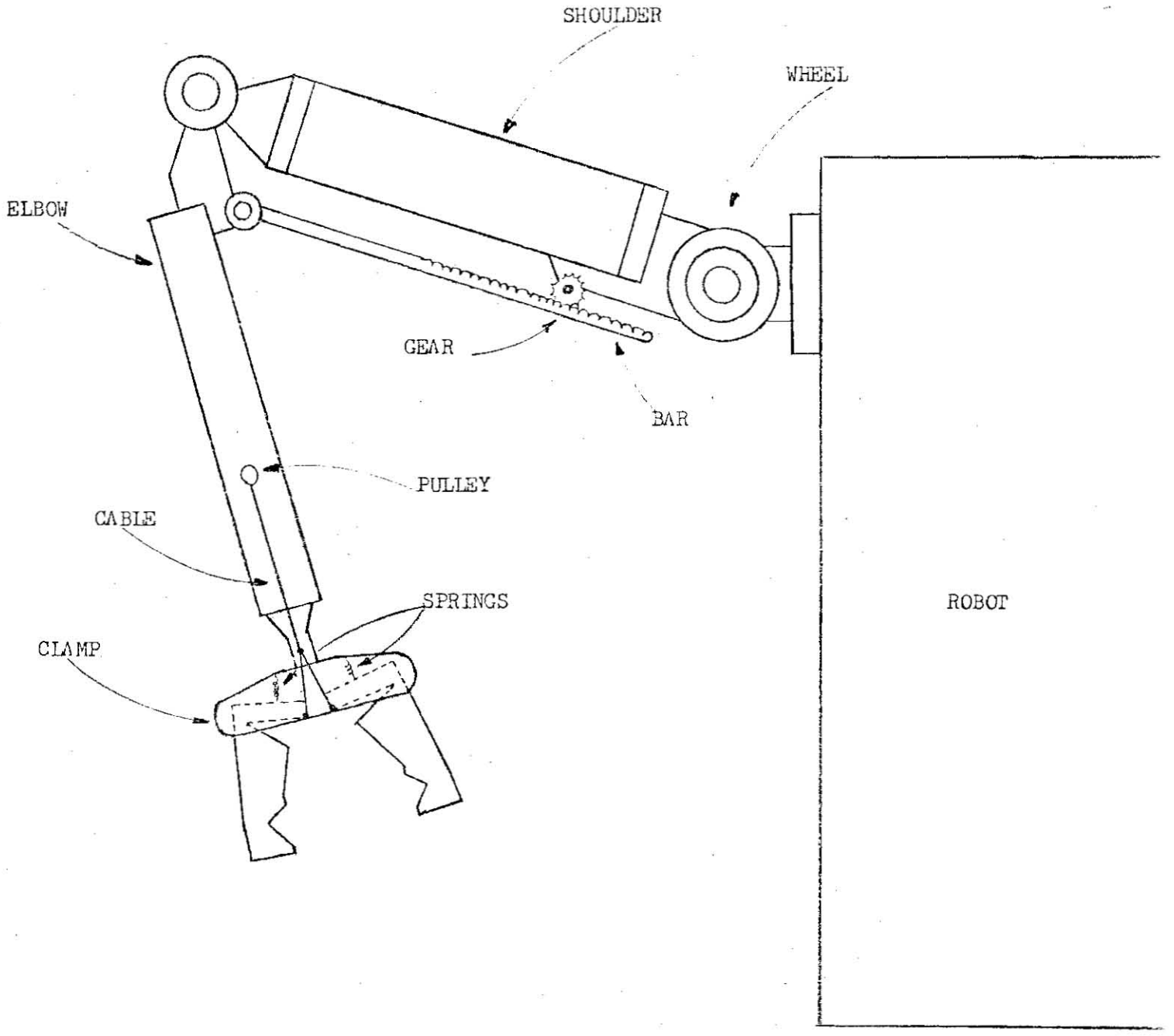
"In factories throughout the world there are several thousand devices loosely called robots. Most of those mechanisisms exhibit few of the characteristics the average person would associate with the term robot; they are mostly pick and place machines that are capable of only the simplest kinds of motion."* The Robot Arm World is a simulation of an example of such a device (see Figure 4.7).

Connected at shoulder position on the side of the robot torso is a simple yet potentially functional robot arm. Beginning at the shoulder area is a wheel. A simple servo-motor (shaft driven) is connected to this wheel. This motor then turns the wheel thus lowering the shoulder shaft. Below the shoulder shaft is attached a motor driven servo-gear with its associated bar. The bar is inlaid with identical teeth to which responds any forward or backward gear movement. The opposite end of this shaft is hinged (below the elbow) with the forearm. This forearm swings back and forth in response to bar movements. Slightly over midway down the forearm is mounted a pulley or electrical winch which drum coils a flexible steel cable. This cables' extended length shortens as the winch turns and winds. Within the base of the forearm a Y-connector splits the cable into two segments each attached to movable, spring loaded, retractable grip pads which respond in unison (as a clamp) to any cable tension.

The SWM (see Figure 4.8) describes these characteristics in the form of related tuples. The TYPE tuples describe the hardware parts and motors. CONTROLS tuples relate which part responds to which motors. STATE tuples indicate the states in which the motors are in

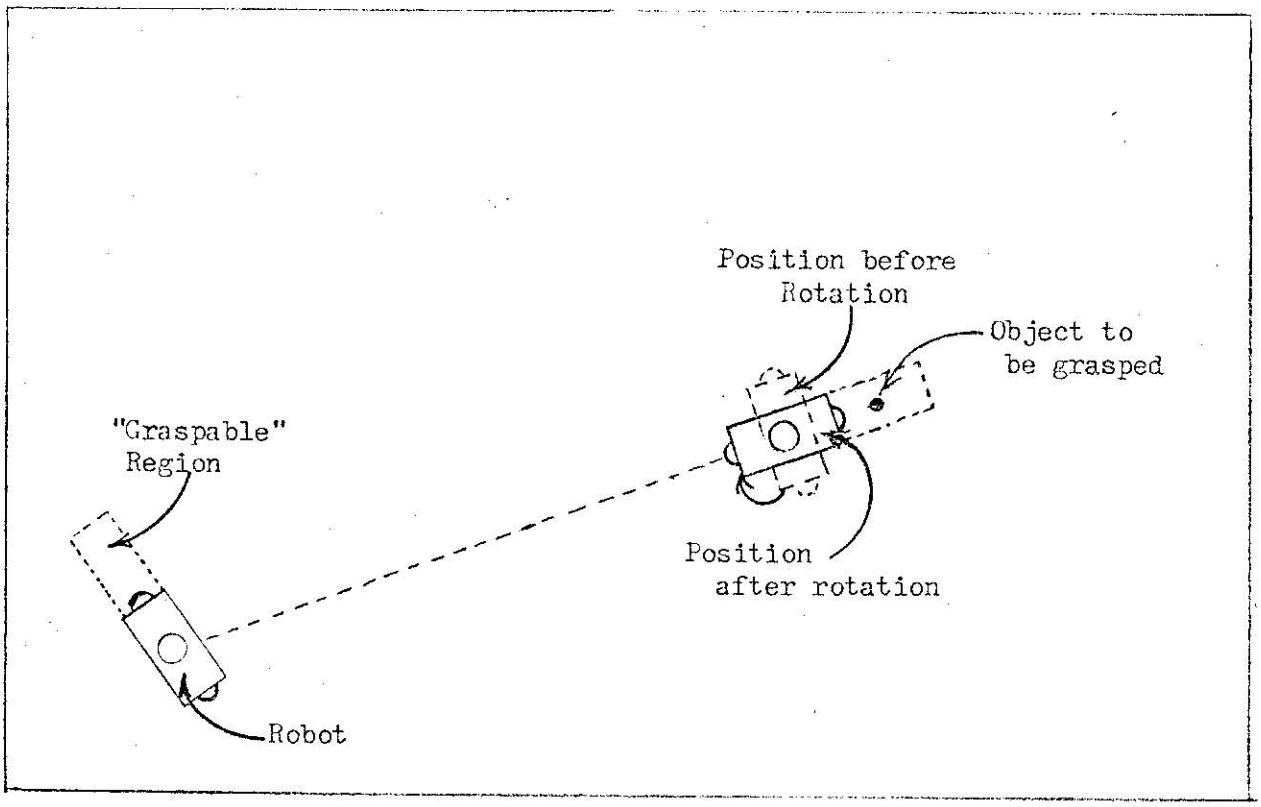
---

* "ROBOT SYSTEMS", Albus, James S. and Evans, John M., Scientific American, Feb. 1976, p. 77.



Robot Arm in Robot Arm World

Figure 4.7




```
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)

(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)

(STATE ARM DOWN)
(STATE GEAR 0)
(STATE WINCH 0)
(STATE WHEEL 0)

(SPEED GEAR 10)
(SPEED CLAMP .5)
(SPEED WINCH 5)
(SPEED WHEEL 3)

(RATE BAR 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(RATE SHOULDER 0 0)

(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)

(LENGTH BAR 90)
(LENGTH CABLE 50)
(LENGTH SHOULDER 5)

(ANGLE RBT 90)
(ANGLE FOREARM 270)
(ANGLE SHOULDER 180)

(GOAL CLAMP 0)
(APART CLAMP 5)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)

(XRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)

(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)

(AT SPHERE 5 10 5)
(AT RBT 12.5 10)
(AT CLAMP 5 10 5)
(AT ELBOW 5 10 10)
(AT SHOULDER 10 10 10)
```

Robot Arm World and SWM

Figure 4.8

(1 = forward, -1 = reverse, $\emptyset$ = stopped) except the ARM DOWN tuple which changes to ARM UP in order to lift objects. The SPEED tuples give the revolution per millisecond speed of the motors. These various speeds have been carefully tuned so as to allow the different arm parts to move in similar time intervals. The RATE tuples are crucial in activating the part controlled by a particular motor. Although the world is updated by the STATE of the motor, the RATE of the affected device is what creates the chain reaction in movement. The EXTENT tuples tell the degree ranges of the arms ($\emptyset^{\circ}$ = due east), and the distances the CLAMP and CABLE can go. The LENGTH tuples coordinate the location of the point at the end of the various devices when in movement. The ANGLE tuples (except for robot angle) tell the current position within the EXTENT ranges of any part.

Other tuples worth noting are first the GOAL tuple. This tuple is used when something is to be grasped. The clamp opens and closes to its full extent as long as the GOAL is $\emptyset$ but when the GOAL is not zero then this (stopping) point is identical to the diameter of the object to be grasped. The APART tuple responds respectively to the distance the clamp should be apart. The ROTATION tuple is like a SPEED tuple but for the rotating speed of the robot. (NOTE: AT is 3 dimensional tuple for some parts)

The following is a brief summary of each scenario (see Figure 4.9).

1) SHOULDER when initiated, responds by gradually changing the angle of the shoulder when the STATE of the WHEEL is non-zero (the WHEEL is an affected device controlled by SERVO-TURN).

2) ELBOW responds to the tuple added by SHOULDER telling that the RATE of the shoulder is non-zero. The elbow's height (Z coordinate of X,Y,Z) changes with respect to shoulder movement.

3) CLAMP is activated basically when the RATE of the cable is non-zero. Its gradual effect is the distance its pads are APART (the CABLE is an affected device

```
(SHOULDER (PAR S W : CRAT CANG CEL CLO ERAT ESTP)
    (ICS (TYPE S W)
        (TYPE W SERVO-MOTOR)
        (STATE W CTAT)
        (SPEED W CSPD)
        (ANGLE S CANG)
        (EXTENT S CHI CLO))
    (ICN (NOT (ZEROP CTAT))
        (:= ERAT (*TIMES CTAT CSPD))
        (:= ESTP (COND ((LT CTAT 0) CLO) (T CHI))))
    (EID (ANGLE S *)
        (RATE S * *))
    (EIA (RATE S ERAT CANG))
    (EGS (ANGLE S YANG))
    (EGN (:= YANG (*PLUS (*TIMES ERAT $) CANG))
         (:= $ (QUO (*DIF YANG CANG) ERAT))))
    (CDS (STATE W CTAT))
    (CCN FUNC (*PLUS (ABS (QUO (*DIF ESTP CANG) ERAT)) #))
    (EPD (STATE W *)
        (RATE S * *))
    (EPA (STATE W 0)
        (RATE S 0 0)))

(ELBOW (PAR S W CRAT CANG : LBO CX CY CZ CLEN CSZ)
    (ICS (TYPE S W)
        (TYPE W SERVO-MOTOR)
        (RATE S CRAT CANG)
        (MOVES W LBO)
        (AT LBO CX CY CZ)
        (LENGTH S CLEN)
        (AT S * * CSZ))
    (ICN (NOT (ZEROP CRAT)))
    (EID (AT LBO * * *))
    (EGS (AT LBO CX CY YZ))
    (EGN (:= YZ (*PLUS (*TIMES (COS (*PLUS (*TIMES CRAT $) CANG)) CLEN) CSZ))
         (:= $ (QUO (*DIF (INVCOS (QUO (*DIF YZ CSZ) CLEN)) CANG) CRAT))))
    (CCS (RATE S CRAT CANG)))

(CLAMP (PAR S CRAT : K CDIS CHI CLO CG CSPD ERAT ESTP)
    (ICS (RATE S CRAT)
        (CONTROLS S K)
        (TYPE K CLAMP)
        (APART K CDIS)
        (EXTENT K CHI CLO)
        (GOAL K CG)
        (SPEED K CSPD))
    (ICN (NOT (ZEROP CRAT))
        (:= ERAT (*TIMES CSPD (DIF CRAT)))
        (:= ESTP (COND ((NOT (ZEROP CG)) CG) ((LT ERAT 0) CLO) (T CHI))))
    (EID (APART K *)
        (RATE K *))
    (EIA (RATE K ERAT))
    (EGS (APART K YDIS))
    (EGN (:= YDIS (*PLUS (*TIMES $ ERAT) CDIS))
         (:= $ (*QUO (*DIF YDIS CDIS) ERAT))))
    (CCS (RATE K ERAT))
    (CCN FUNC (*PLUS (ABS (*QUO (*DIF CDIS ESTP) ERAT)) #))
    (EPD (RATE K *)
        (RATE S *))
    (EPA (RATE K 0)))
```

Robot Arm World Scenarios (Part I)

Figure 4.9

```
(SERVO-TURN (PAR $ DEV : CTAT CSPD AFD CLEN CHI CLO ESTP ERAT)
    (ICS(TYPE $ DEV)
        (STATE $ CTAT)
        (SPEED $ CSPD)
        (CONTROLS $ AFD)
        (LENGTH AFD CLEN)
        (EXTENT AFD CHI CLO))
    (ICN(NOT(ZEROP CTAT))
        (:= ESTP(COND((LT CTAT 0)CLO)(T CHI)))
        (:= ERAT(*TIMES CTAT CSPD)))
    (EID(LENGTH AFD *)
        (RATE AFD *))
    (EIA(RATE AFD ERAT))
    (EGS(LENGTH AFD YLEN))
    (EGN((:= YLEN(*PLUS(*TIMES $ ERAT)CLEN))
         (:= $(*QUO(*DIF CLEN ESTP)ERAT))))
    (CCS(RATE AFD ERAT))
    (CCN FUNC(*PLUS(ABS(*QUO(*DIF CLEN ESTP)ERAT))#))
    (EPD(RATE AFD *)
        (STATE $ *))
    (EPA(STATE $ 0)
        (RATE AFD 0)))

(BAR (PAR B CRAT : F CHI CLO CANG ESTP)
    (ICS(RATE B CRAT)
        (TYPE B EXTENSION-BAR)
        (CONTROLS B F)
        (EXTENT F CHI CLO)
        (ANGLE F CANG))
    (ICN(NOT(ZEROP CRAT))
        (:= ESTP(COND((LT CRAT 0)CLO)(T CHI))))
    (EID(ANGLE F *))
    (EGS(ANGLE F YANG))
    (EGN((:= YANG(*PLUS(TIMES CRAT $(QUO 1 3))CANG))
         (:= $(QUO(*DIF YANG CANG)(*TIMES CRAT(QUO 1 3))))))
   (CCS(RATE B CRAT))
    (CCN FUNC(*PLUS(ABS(QUO(*DIF CANG ESTP)(*TIMES CRAT(QUO 1 3))))#))
    (EPD(RATE B *)))

(UPARM (PAR P G W : B S)
    (ICS(UPARM P)
        (STATE ARM DOWN)
        (TYPE G SERVO-GEAR)
        (TYPE W SERVO-MOTOR)
        (CONTROLS G B)
        (MOVES W S))
    (EID(STATE G *)
        (UPARM P)
        (STATE W *))
    (EIA(STATE W -1))
    (CCS(STATE W -1))
    (EPD(STATE ARM DOWN))
    (EPA(STATE G -1)
        (STATE ARM UP)))
```

Robot Arm World Scenarios (Part II)

Figure 4.9

```
(DOWNARM (PAR P G W : B S)
    (ICS(DOWNARM P)
        (STATE ARM UP)
        (TYPE G SERVO-GEAR)
        (TYPE W SERVO-MOTOR)
        (CONTROLS G B)
        (MOVES W S))
    (EID(STATE G *)
        (DOWNARM P)
        (STATE W *))
    (EIA(STATE G 1))
    (CCS(STATE G 1))
    (EPD(STATE ARM UP))
    (EPA(STATE ARM DOWN)
        (STATE W 1)))

(ROTATE (PAR P CANG : COLD CR ERAT)
    (ICS(ROTATE P CANG)
        (NOTROTATING P)
        (ANGLE P COLD)
        (ROTATION P CR))
    (ICN(:= ERAT(*TIMES CR (DIR CANG))))
    (EID(NOTROTATING P)
        (ROTATE P CANG)
        (ANGLE P *))
    (EIA(ROTATING FROM P COLD ERAT))
   (EGS(ANGLE P YANG))
    (EGN((:= YANG(THIN(*PLUS(*TIMES $ ERAT)COLD)360))
         (:= $(QUO(*DIF YANG COLD)ERAT))))
    (CCS(ROTATING FROM P COLD ERAT))
    (CCN FUNC(*PLUS(ABS(*QUO CANG ERAT))*))
    (EPD(ROTATING FROM P COLD ERAT))
    (EPA(NOTROTATING P)))

(LOCR (PAR R CA CRAT P : CRX CRY CX CY CZ EDIS)
    (ICS(ROTATING FROM R CA CRAT)
        (HASASPART P P)
        (AT P CRX CRY)
        (AT P CX CY CZ))
    (ICN(:= EDIS(DIS CX CY CRX CRY)))
    (EID(AT P * * *))
    (EGS(AT P YX YY CZ))
    (EGN((:= YX(*PLUS(*TIMES(COS(THIN(PLUS(*TIMES $ CRAT)CA 90)360))
                              EDIS)CRX))
         (:= $(QUO(DIFFERENCE(INVCOS(QUO(*DIF YX CRX)EDIS))CA 90)CRAT)))
         ((:= YY(*PLUS(*TIMES(SIN(THIN(PLUS(*TIMES $ CRAT)CA 90)360))
                              EDIS)CRY))
         (:= $(QUO(DIFFERENCE(INVSIN(QUO(*DIF YY CRY)EDIS))CA 90)
                   CRAT))))
    (CCS(ROTATING FROM R CA CRAT)))

(TURNTO (PAR P CANG : COLD EANG)
    (ICS(TURNTO P CANG)
        (ANGLE P COLD))
    (ICN(NE CANG COLD)
        (:= EANG(DIFANG CANG COLD)))
    (EID(TURNTO P CANG))
    (EIA(ROTATE P EANG)))
```

Robot Arm World Scenarios (Part III)

Figure 4.9

```
(GOTO (PAR R CX CY : CXF CYF E) EXR EYR)
    (ICS(GOTO R CY CY)
        (AT R CXF CYF))
    (ICN(GE CX 0)
        (GE CY 0)
        (:= ED(SQRT(+PLUS(SQ(+DIF CX CXF))(SQ(+DIF CY CYF))))
        (:= EXR(QUO(+TIMES(+DIF CY CXF)5)ED))
        (:= EYR(QUO(+TIMES(+DIF CY CYF)5)ED)))
    (EID(XRATE R +)
        (YRATE R +)
        (AT R + +))
    (EIA(XRATE R EXR)
        (YRATE R EYR))
    (ES(AT R YX YY))
    (EGN((:= YX(+PLUS(+TIMES EXR $)CXF))
         (:= $(QUO(+DIF YX CXF)EXR)))
        ((:= YY(+PLUS(+TIMES EYR $)CYF))
         (:= $(QUO(+DIF YY CYF)EYR))))
    (CCS(GOTO R CX CY))
    (CCN FUNC(+PLUS(QUO ED 5)6))
   (EPD(XRATE R +)
        (YRATE R +)
        (GOTO R CX CY))
    (EPA(XRATE R 0)
        (YRATE R 0)))

(LOOG (PAR R P : CXR CYR CX CY CZ)
    (ICS(HASASPART R P)
        (XRATE R CXR)
        (YRATE R CYR)
        (AT P CX CY CZ))
    (ICN(NOT(AND(ZEROP CXR)(ZEROP CYR))))
    (SID(AT P + + +))
    (ES(AT P YX YY CZ))
    (EGN((:= YX(+PLUS CX(+TIMES CXR $)))
         (:= $(QUO(+DIF YX CX)CXR)))
        ((:= YY(+PLUS CY(+TIMES CYR $)))
         (:= $(QUO(+DIF YY CY)CYR))))
    (CCS(XRATE R CXR)
        (YRATE R CYR)))

(CLAMPTO (PAR R OBJ KLMP : CX CY CRX CRY EANG EX EY)
    (ICS(GRASP R OBJ)
        (NOTGRASPING KLMP)
        (XRATE R 0)
        (YRATE R 0)
        (AT OBJ CX CY +)
        (AT R CRX CRY))
    (ICN(PROGE 0)(SETQ EPSILON .001)(RETURN(NE(DIS CX CY CRX CRY)7.5)))
        (:= EANG(THIN(+DIF(BUCKET(SLANG CRX CRY CX CY))90 360)))
       (:= EX(XDIS CX(THIN(+PLUS BUCKET 180 360)7.5)))
        (:= EY(YDIS CY(THIN(+PLUS BUCKET 180 360)7.5)))
    (EIA(GOTO R EX EY))
    (CCS(GOTO R EX EY))
    (EPA(TURNTO R EANG)))
```

Robot Arm World Scenarios (Part IV)

Figure 4.9

```
(GRASP (PAR P OBJ KLMP : CDIS  AB W CDAM CX CY CZ CKX CKY CKZ)
    (ICS(GRASP P OBJ)
        (NOTGRASPING KLMP)
        (STATE ARM DOWN)
        (APART KLMP CDIS)
        (CONTROLS CAB KLMP)
        (CONTROLS W CAB)
        (DIAMETER OBJ CDAM)
        (AT OBJ CX CY CZ)
        (AT KLMP CKX CKY CKZ))
    (ICN(LE CDAM CDIS)
        (PROGE() (SETQ EPSILON 1)(RETURN(AND(EQUAL CX CKX)(EQUAL CY CKY)
                                            (EQUAL CZ CKZ))))))
    (EID(GRASP P OBJ)
        (GOAL KLMP *)
        (STATE W *)
        (NOTGRASPING KLMP))
    (EIA(STATE W -1)
        (GOAL KLMP CDAM))
    (CCS(STATE W -1))
    (EPD(GOAL KLMP *))
    (EPA(GRASPING KLMP OBJ)))

(RELEASE (PAR OBJ : KLMP CAB W)
   (ICS(RELEASE OBJ)
        (GRASPING KLMP OBJ)
        (CONTROLS KLMP CAB)
        (CONTROLS W CAB))
    (EID(RELEASE OBJ)
        (GOAL KLMP *)
        (STATE W *)
        (GRASPING KLMP OBJ))
    (EIA(NOTGRASPING KLMP)
        (STATE W 1)
        (GOAL KLMP 0)))
```

Robot Arm World Scenarios (Part V)

Figure 4.9

```
(DE OFF() (SETQ OFF #OFF))

(DE DIFANG(A B)(PROG(H D)
    (SETQ D(*DIF 360(ABS(SETQ H(*DIF A B)))))
    (RETURN(COND
        ((LT(ABS H)D)H)
        ((LT H 0)D)
        (T(NEG D))))))

(DE DIR(N)(COND((LT N 0)-1)(T 1)))
```

Robot Arm World Help Functions

Figure 4.10

controlled by SERVO-TURN).

4) SERVO-TURN is a motor controlling device which not only turns any motor in the direction its STATE commands but also controls the gradual effect of whatever affected device the motor controls.

5) BAR is initiated when the bar is in movement. This creates a gradual angle change in the forearm (BAR is an affected device controlled by SERVO-TURN).

6) UPARM is activated when the ARM is DOWN and the command to UPARM is given. This initiates the lifting of the shoulder and the swinging out of the forearm (one at a time through the use of CCS)

7) DOWNARM is activated when the ARM is UP and the command to DOWNARM is given. This initiates the swinging in of the arm and the lowering of the shoulder.

8) ROTATE is a gradual scenario which simply turns the angle the robot is facing plus or minus a certain number of degrees (determined by the ROTATE tuple).

9) LOCR is like LOC of the Hendrix World except is for robot rotation. Only the parts of the arm are involved for clearly if the robot turns the shoulder, elbow, and clamp will gradually have changing cartesian coordinates (although their height will remain unchanged).

10) TURNTO computes the desired angle given in its command tuple and activates ROTATE with a degree shift which will take the robot to that given TURNTO angle.

11) GOTO is identical to that in the Robot Eye World.

12) LOCG like LOCR moves the robot appendages but for GOTO movement. This is very similar to the LOC in the Hendrix World but height is unaffected.

13) CLAMPTO is initiated when the GRASP tuple is found but the robot is too far away (or too close) to GRASP the object. It sends the robot the appropriate distance in the correct direction and subsequently turns it to just the right GRASP position.

14) GRASP is only initiated when the CLAMP and object to GRASP are at the same location (X,Y,Z). This is where the GOAL tuple (telling GRASP object diameter) described earlier comes from.

15) RELEASE can release a GRASPed object at any time provided the command tuple is present. It simply opens the clamp and updates the new NOTGRASPED state.

These scenarios unlike the Billiards world with its large library of help functions do nearly all the simulation work themselves. Three functions described below (see Figure 4.10) however are used for simple ICN manipulations.

(OFF) - This function simply sets a quote binding of @OFF for the variable OFF. Although not used in this demonstration, this could replace the STATE of a motor from Ø to OFF. This idea is discussed in the Billiards World section on S-Variables.

(DIFANG A B) - Is used by TURNTO to compute the angle necessary to ROTATE (positive or negative) in order to attain a desired angle.

(DIR N) - Returns -1 if N < Ø; else 1.

Execution of the Robot Arm World - Summary (see APPENDIX F)

Placing a sphere at point (50,50) the command to GRASP it is given. Because the robot is too far away CLAMPTO is initiated and the robot travels toward the object. LOCG is initiated after GOTO to track the moving arm. Arriving at the sphere, the robot turns, placing the arm at the point where the sphere is. LOCR is initiated after ROTATE to track the changing points of the arm. ROTATE inturn activates GRASP which then suceeds after activation of the winch, cable, and clamp. After GRASPING, the RELEASE scenario is called to reverse the winch, cable, and clamp and let go of the sphere. Finally UPARM and DOWNARM are called in a row. This is to demonstrate the functioning of the wheel, shoulder, elbow, gear, bar, and forearm.

CHAPTER V - CONCLUSIONS

The worlds in this paper were hoped to illustrate the different methods needed to model different types of simulations. As can be seen, in some cases (like the Billiards World) simulation is highly dependent upon LISP functions performing calculations. In this type of world the Simulation System does very little work. In other worlds (like the Robot Arm World) LISP function evaluation is held to a minimum and vast loads of work are placed upon the system. Either alternative is best for its type of simulation and reversal of these roles would probably result in inefficient failures.

In all worlds presented and any future worlds of at least the same or greater complexity, modelling is by no means a trivial matter. Coordination of scenarios and state of the world model is a very meticulous and time consumming affair. This is true even without considering debugging. To add the additional time needed for debugging, the user, after attempting to model worlds such as these, would appreciate the discussion presented in CHAPTER II. BUGOFF becomes a time saving little gem when complex scenarios are involved, and insignificant looking changes like identation in scenarios becomes very crucial.

Finally, the potential for S-variables it seems is very great. In addition to the symbolic variable additions to the state of the world model the best advancement in the system would be for an evaluation option in symbolic conditions so that LISP functions can be evaluated, returning results used in initiating scenarios. Given this option in some easy form, the Hendrix Simulation System would surely gain a flexibility and especially a clarity far surpassing its present implementation. The main advantage (given this new option) would be that little training would be necessary to convert the competent LISP programmer into a competent Hendrix Simulator.

APPENDIX A

Execution of the Sort World

```
<HSI*M>

================================
HENDRIX SIMULATING SYSTEM
================================

INPUT SCENARIO LIST: *(EVAL SLIST)

SORT

INPUT SWM RELATION LIST: *(EVAL SWM)


COMMAND: *PICTURE


****TIME****
0
****EXPRS****
(AT BOX1 1)
(AT BOX2 2)
(AT BOX3 3)
(AT BOX4 4)
(AT BOX5 5)
(AT BOX6 6)
(CONTENTS BOX1 JACK6)
(CONTENTS BOX2 JACK5)
(CONTENTS BOX3 JACK4)
(CONTENTS BOX4 JACK3)
(CONTENTS BOX5 JACK2)
(CONTENTS BOX6 JACK1)
(VAL JACK1 1)
(VAL JACK2 2)
(VAL JACK3 3)
(VAL JACK4 4)
(VAL JACK5 5)
(VAL JACK6 6)
****SKLRS****
******************

COMMAND: *
*DAY: :02:39 RUN: 5.58 RD:334 WR:4 L 62+8P TI  PC:405754
INPUT WAIT FOR TTY1:
DAY: 5.86 RUN: 0.00 RD:0 WR:0 L 62+8P TI  PC:405754
INPUT WAIT FOR TTY1:
GO

COMMAND: *
*DAY: :16:45 RUN: :13:22 RD:0 WR:0 L 62+8P TI  PC:405754
INPUT WAIT FOR TTY1:
PICTURE
```

CPU Time
Before and After

```
****TIME****
0
****EXPRS****
(AT BOX1 1)
(AT BOX2 2)
(AT BOX3 3)
(AT BOX4 4)
(AT BOX5 5)
(AT BOX6 6)
(CONTENTS BOX6 JACK6)
(CONTENTS BOX5 JACK5)
(CONTENTS BOX4 JACK4)
(CONTENTS BOX3 JACK3)
(CONTENTS BOX2 JACK2)
(CONTENTS BOX1 JACK1)
(VAL JACK1 1)
(VAL JACK2 2)
(VAL JACK3 3)
(VAL JACK4 4)
(VAL JACK5 5)
(VAL JACK6 6)
****SKLPS****
******************

COMMAND: *STOP

(****TERMINATED-AT-TIME**** 0)
```

APPENDIX B

Execution of BUGOFF

NOTE: For demonstration purposes, the following following errors were placed in the Hendrix World (see Figure 3.2)

1) 1 missing right paren in the CCN of MONITORSET
2) 1 missing right paren in the ICS of TURNVALVE
3) 1 extra right paren in the ICS of FILLBUCKET
4) Ø for an O (word OFF) in the ICS of SETALARM
5) O for a Ø (number 2Ø) in the EIA of AWAKENROBOT
6) ALL-HCT for ALL-ACT in the ICS of OFFALARM
7) all E-variables (ED, EXR, EYR) deleted from the parameter list in GOTO
8) EDI in place of EID in LOC (effects initial delete)

These errors demonstrate the use of BUGOFF yet at the same time are models for the type of typical blunders occurring in modelling. After each run, LISP was exited and only those changes spotted as errors by BUGOFF were dealt with before the next run.

```
********** RUN NO. 1 **********

*(BUGOFF 'LIST 'MM 12)


***BUGOFF***

*PARENTHESIS ERROR*
2 SCENARIOS DEFINED:
     1 SETALARM
     2 MONITORSET


***END-BUGOFF***
```

```
********** RUN NO. 2 **********

*(BUGOFF SLIST SWM 12)


***BUGOFF***


*PARENTHESIS ERROR*
8 SCENARIOS DEFINED:
      1 SETALARM
      2 MONITORSET
      3 AWAKENROBOT
      4 SOUNDALARM
      5 MONITORALARM
      6 OFFALARM
      7 SLEEPROBOT
      8 TURNVALVE



***END-BUGOFF***


********** RUN NO. 3 **********

*(BUGOFF SLIST SWM 12)


***BUGOFF***


*PARENTHESIS ERROR*
15 SCENARIOS DEFINED:
      1 SETALARM
      2 MONITORSET
      3 AWAKENROBOT
      4 SOUNDALARM
      5 MONITORALARM
      6 OFFALARM
      7 SLEEPROBOT
      8 TURNVALVE
      9 FILLBUCKET
     10 ICN
     11 EID
     12 EGS
     13 EGN
     14 CCS
     15 CCN



***END-BUGOFF***
```

```
********** RUN NO. 4 *********

*(BUGOFF SLIST SWM 13)


***BUGOFF***


*PARENTHESIS ERROR*
14 SCENARIOS DEFINED:
      1 SETALARM
      2 MONITORSET
      3 AWAKENROBOT
      4 SOUNDALARM
      5 MONITORALARM
      6 OFFALARM
      7 SLEEPROBOT
      8 TURNVALVE
      9 FILLBUCKET
     10 GRASP
     11 RELEASE
     12 MOVABILITY
     13 GOTO
     14 LOC



***END-BUGOFF***

*(BUGOFF SLIST SWM 14)


***BUGOFF***



1 SCENARIO NAME: SETALARM
       *UNBOUND IDENTIFIERS*
              ICS: FF GRASPING

2 SCENARIO NAME: MONITORSET
       *UNBOUND IDENTIFIERS*
              CCN: EPSILON

3 SCENARIO NAME: AWAKENROBOT
       *UNBOUND IDENTIFIERS*
              ICS: SOUNDING ASLEEP
              EID: ASLEEP
              EIA: 0 GOTO GRASP OFFALARM

4 SCENARIO NAME: SOUNDALARM
       *UNBOUND IDENTIFIERS*
              EIA: SOUNDING

5 SCENARIO NAME: MONITORALARM

6 SCENARIO NAME: OFFALARM
       *UNBOUND IDENTIFIERS*
              ICS: GRASPING SOUNDING ALL-HOT
              EID: SOUNDING
```

```
7 SCENARIO NAME: SLEEPDUCT
      *UNBOUND IDENTIFIERS*
            EIA: ASLEEP

8 SCENARIO NAME: TURNVALVE
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EGS: YFLOWRATE
            EGN: YFLOWRATE

9 SCENARIO NAME: FILLBUCKET
      *UNBOUND IDENTIFIERS*
            EGS: YCONTENT
            EGN: YCONTENT

10 SCENARIO NAME: GRASP
      *UNBOUND IDENTIFIERS*
            EIA: GRASPING

11 SCENARIO NAME: RELEASE
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EID: GRASPING

12 SCENARIO NAME: MOVABILITY
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            OCS: GRASPING

13 SCENARIO NAME: GOTO
      *UNBOUND IDENTIFIERS*
            ICN: EYR EXR
            EIA: EYR EXR
            EGS: YY YX
            EGN: EYR YY EXR YX

14 SCENARIO NAME: LOC
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EGS: YY YX
            EGN: YY YX
            OCS: GRASPING
ILLEGAL KEYWORDS:  EDI


***END-BUGOFF***

********** RUN NO. 5 **********

*(BUGOFF SLIST SWM 14)


***BUGOFF***


1 SCENARIO NAME: SETALARM
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING

2 SCENARIO NAME: MONITORSET
      *UNBOUND IDENTIFIERS*
            OCN: EPSILON
```

```
3 SCENARIO NAME: AWAKENROBOT
      *UNBOUND IDENTIFIERS*
            ICS: SOUNDING ASLEEP
            EID: ASLEEP
            EIA: GOTO GRASP OFFALARM

4 SCENARIO NAME: SOUNDALARM
      *UNBOUND IDENTIFIERS*
            EIA: SOUNDING

5 SCENARIO NAME: MONITORALARM

6 SCENARIO NAME: OFFALARM
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING SOUNDING
            EID: SOUNDING

7 SCENARIO NAME: SLEEPROBOT
      *UNBOUND IDENTIFIERS*
            EIA: ASLEEP

8 SCENARIO NAME: TURNVALVE
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EGS: YFLOWRATE
            EGN: YFLOWRATE

9 SCENARIO NAME: FILLBUCKET
      *UNBOUND IDENTIFIERS*
            EGS: YCONTENT
            EGN: YCONTENT

10 SCENARIO NAME: GRASP
      *UNBOUND IDENTIFIERS*
            EIA: GRASPING

11 SCENARIO NAME: RELEASE
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EID: GRASPING

12 SCENARIO NAME: MOVABILITY
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            CCS: GRASPING

13 SCENARIO NAME: GOTO
      *UNBOUND IDENTIFIERS*
            EGS: YY YX
            EGN: YY YX

14 SCENARIO NAME: LOC
      *UNBOUND IDENTIFIERS*
            ICS: GRASPING
            EGS: YY YX
            EGN: YY YX
            CCS: GRASPING


***END-BUGOFF***
```

BUGOFF Functions

```
          ********** BUGOFF **********

(DE FILTER (SWML) (COND
     ((NULL SWML) NIL)
     ((ATOM (CAR SWML)) (COND
          ((MEMBER (CAR SWML) (CDR SWML)) (FILTER (CDR SWML)))
          (T (CONS (CAR SWML) (FILTER (CDR SWML))))))
     (T (FILTER (APPEND (CAR SWML) (CDR SWML))))))
(DE MEMBER (A L) (COND
     ((NULL L) NIL)
     ((ATOM (CAR L)) (OR (EQ A (CAR L)) (MEMBER A (CDR L))))
     (T (MEMBER A (APPEND (CAR L) (CDR L))))))
(DE BUGOFF (SLIST SWM BUG) (PROG (NUM SLST)
     (SETQ WLIST (APPEND (FILTER SWM)
     $(ALLOCATE-ACTIVATE ALL-ACT NOT EQUAL GT LT GE LE *TIMES QUO ABS
                    ABDIF SQ SORT * ME $ ; *PLUS *DIF := PLUS TIMES
                    FUNC T COND *DUO OR NUMBERP AND BUCKET %)))
     (PL 2) (PRINC $***BUGOFF***) (PL 2)
     (SETQ CNT 0)
     (SETQ NUM (LEN SLIST))
     (COND
          ((EQUAL NUM BUG) (SPLIT-SCENES SLIST))
          (T (CLOSE NUM) (PL 3)))
     (RETURN $***END-BUGOFF***)))
(DE PL (N) (COND ((ZEROP N) T) ((TERPRI T) (PL (SUB1 N)))))
(SETQ BLK $/ )
(DE SPLIT-SCENES (L) (COND
     ((NULL L) (PL 3))
     ((ONE-SCENE (CAR L)) (SPLIT-SCENES (CDR L)))))
(SETQ EFCON $(ICS ICN EID EIA EGS EGN CCS CCN EPD EPA))
(DE ONE-SCENE (L) (PROG ()
     (PL 2) (SETQ MESS $(*UNBOUND IDENTIFIERS*))
     (SETQ CNT (ADD1 CNT)) (PRIN1 CNT) (PR ))
     (PRLIST2 $(SCENARIO NAME:))
     (SETQ WLIST (CONS (CAR L) WLIST))
     (SETQ ILLIST NIL)
     (PRIN1 (CAR L))
     (RETURN (COND
          ((EQ (CADDR L) $PAR) (SETQ WLIST (APPEND (CDADDR L) WLIST))
                               (LEGAL (CDDDR L)))
          (T (LEGAL (CDR L)))))))
(DE LEGAL (REST) (COND
     ((NULL REST) (ILL-CHECK))
     ((MEMQ (CAAR REST) EFCON) (LSTOUT (CAR REST))
                               (LEGAL (CDR REST)))
    (T (SETQ ILLIST (CONS (CAAR REST) ILLIST))
       (LEGAL (CDR REST)))))
(DE ILL-CHECK () (COND
     ((NULL ILLIST) T)
     (T (PRLIST $(ILLEGAL KEYWORDS:))
       (PRINC BLK) (PRLIST2 ILLIST))))
```

```
(DE PB(N)(COND((ZEROP N)T)((PRINC BLK)(PB(SUB1 N))))))
(DE LSTOUT(LST)(PROG(BLIST)
    (SETQ BLIST(BADLIST(CDR LST)))
    (COND
        ((NULL BLIST)(RETURN T))
        (T(COND
            ((NULL MESS)(TERPRI)(PB 13))
            (T(TERPRI)(PB 6)(PRLIST2 MESS)(TERPRI)(PB 13)
              (SETQ MESS NIL)))
        (PRIN1(CAR LST))
        (PRIN1 ":)
        (PB 1)
        (PRLIST2 BLIST)))
    (RETURN T)))
(DE BADLIST(LST)(PROG(BLST FLST LAMLST)
    (SETQ LAMLST '(SUBR FSUBR LSUBR FEXPR MACRO EXPR))
    (SETQ FLST(FILTER LST))
    (REPEAT WHILE FLST
        (COND
            ((MEMQ(CAR FLST)EFCON)(RETURN(PAPERR(CAR FLST))))
            ((NUMBERP(CAR FLST))(SETQ FLST(CDR FLST)))
            ((OR(MEMQ(CAR FLST)WLIST)
                 (MEMQ(CADR(CAR FLST))LAMLST))(SETQ FLST(CDR FLST)))
            (T(SETQ BLST(CONS(CAR FLST)BLST))
              (SETQ FLST (CDR FLST)) )))
        (RETURN BLST)))
(DE END() (COND(T(PL 2)'***END-BUGOFF***))))
(DE CLOSE(N)(PROG(SSLST N2)
    (PRLIST '(*PARENTHESIS ERROR*))(TERPRI)
    (PRIN1 N)(PB 1)(PRLIST2 '(SCENARIOS DEFINED:))
    (TERPRI)(SETQ SSLST SLIST)(SETQ N2 0)
    (REPEAT WHILE SSLST
        (COND((>GREAT N2 8)(PB 4))(T(PB 5)))
        (SETQ N2 (ADD1 N2))(PRIN1 N2)(PB 1)(PRIN1(CAAR SSLST))
        (SETQ SSLST(CDR SSLST))(TERPRI))
    (RETURN T)))
(DE LEN(L)(COND
    ((NULL L)0)
    (T(ADD1(LEN(CDR L))))))
(DE PRLIST(L)(COND((TERPRI T)(PRLIST2 L)L)))
(DE PRLIST2(L)(COND((NULL L)T)(T(PRIN1(CAR L))(PB 1)(PRLIST2(CDR L))))))
(DE PAPERR(AT)(PROG()
    (PRLIST '(*MISSING RIGHT PARENTHESIS*))
    (TERPRI)(PB 2)(PRLIST2 '(FOUND BEFORE:))
    (PRINC BLK)(PRINC AT)
    (RETURN NIL)))
(SETQ BLK '| |)
```

APPENDIX C

Execution of the Robot Eye World

Run No. 1

```
*(HSIM)

==============================
HENDRIX SIMULATING SYSTEM
==============================

INPUT SCENARIO LIST: *(EVAL SLIST)
SPOT MONITOR-SPOT GOTO RESPOT MOVETO EXAMINE RECOGNIZE STOP-SEARCH SEARC
H GOHOME RESEARCH RESTART

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(ADD
*(TYPE OBJ OBJECT)
*(AT-OBJ 33 33)
*(CHAR OBJ RED)
*(NEW OBJ))

COMMAND: *(TRACE *)

COMMAND: AUTOSNAP

COMMAND: *(ADD (FIND ROB RED))

COMMAND: GO
////ERROR////
(LAST-POS * *) NOT FOUND

<<<CREATING CB>>> TIME = 0
SEARCH ** (R ROB) (CX 0) (CY 10) (CB 60) (CD 60) (CDV 20) (CANG 0) (E
NDANG 0) (EX 50.0) (EY 10)

<<<DESTROYING CB>>> TIME = 0
SEARCH ** (R ROB) (CX 0) (CY 10) (CB 60) (CD 60) (CDV 20) (CANG 0) (E
NDANG 0) (EX 50.0) (EY 10)

<<<CREATING CB>>> TIME = 0
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0) (CYF 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)
```

```
****TIME****
0
****EXPRS****
(FROM ROB 0 10)
(GOTO ROB 50.0 10)
(LAST-POS 0 10)
(FIND ROB RED)
(NEW OBJ)
(CHAR OBJ RED)
(TYPE OBJ OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ 33 33)
(STATE ROB NORMAL)
(ANGLE ROB 0)
(XRATE ROB 5.0)
(YRATE ROB 0.0)
(PATH ROB ON)
(OLD-PT ROB 0.10)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 10)
(PATH-ANG 0)
****SKLRS****
(AT ROB 0.0 10.0)
******************

<<<DESTROYING CB>>> TIME = 10.0
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0) (CYF 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 10.0
SEARCH ** (R ROB) (CX 50.0) (CY 10.0) (CR 60) (CD 60) (CDV 20) (CANG
0) (ENDANG 90) (EX 50.0) (EY 20.0)

<<<DESTROYING CB>>> TIME = 10.0
SEARCH ** (R ROB) (CX 50.0) (CY 10.0) (CR 60) (CD 60) (CDV 20) (CANG
0) (ENDANG 90) (EX 50.0) (EY 30.0)

<<<CREATING CB>>> TIME = 10.0
GOTO ** (R ROB) (CXT 50.0) (CYT 30.0) (CXF 50.0) (CYF 10.0) (ED 20.0)
 (EXR 0.0) (EYR 5.0)
```

```
****TIME****
10.0
****EXPRS****
(FROM ROB 50.0 10.0)
(GOTO ROB 50.0 30.0)
(LAST-POS 50.0 10.0)
(FIND ROB RED)
(NEW OBJ)
(CHAR OBJ RED)
(TYPE OBJ OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ 33 33)
(STATE ROB NORMAL)
(ANGLE ROB 90)
(XRATE ROB 0.0)
(YRATE ROB 5.0)
(PATH ROB ON)
(OLD-PT ROB 0 10)
(VISION EYE 80 80)
(RESOLUTION EYE 5)
(DIMENSION FIELD 80 80)
(PATH-POS 50.0 30.0)
(PATH-ANG 90)
****SKLRS****
(AT ROB 50.0 10.0)
******************

<<<DESTROYING CB>>> TIME = 14.0
GOTO ** (R ROB) (XT 50.0) (YT 30.0) (XF 50.0) (YF 10.0) (ED 20.0)
 (EXR 0.0) (EYR 5.0)

<<<CREATING CB>>> TIME = 14.0
SEARCH ** (R ROB) (X 50.0) (Y 30.0) (CR 60) (CD 60) (CDV 20) (CANG
90) (ENDANG 180) (EX 10.0) (EY 30.0)

<<<DESTROYING CB>>> TIME = 14.0
SEARCH ** (R ROB) (X 50.0) (Y 30.0) (CR 60) (CD 60) (CDV 20) (CANG
90) (ENDANG 180) (EX 10.0) (EY 30.0)

<<<CREATING CB>>> TIME = 14.0
GOTO ** (R ROB) (XT 10.0) (YT 30.0) (XF 50.0) (YF 30.0) (ED 40.0)
 (EXR -5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 14.0
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ) (CDX 3
3) (CDY 33) (CANG 180) (X 50.0) (Y 30.0) (XR -5.0) (YR 0.0) (CR 2
0) (CD 20) (EDIS 0)

<<<DESTROYING CB>>> TIME = 14.0
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ) (CDX 3
3) (CDY 33) (CANG 180) (X 50.0) (Y 30.0) (XR -5.0) (YR 0.0) (CR 2
0) (CD 20) (EDIS 0)

<<<DESTROYING CB>>> TIME = 14.0
GOTO ** (R ROB) (XT 10.0) (YT 30.0) (XF 50.0) (YF 30.0) (ED 40.0)
 (EXR -5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 14.0
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ) (CDX 33) (CDY 33) (CANG 180) (X
 50.0) (Y 30.0) (XR 0) (YR 0) (CR 20) (CD 20)
```

```
<<<DESTROYING CB>>> TIME = 14.0
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ) (CDX 33) (CDY 33) (CANG 180) (CX
 50.0) (CY 30.0) (CXR 0) (CYR 0) (CB 20) (CD 20)

<<<CREATING CB>>> TIME = 14.0
MOVETO ** (OBJ OBJ) (R ROB) (I EYE) (CDX 33) (CDY 33) (CX 50.0) (CY 3
0.0) (CANG 180) (CRES 5) (EANG 169.99201) (EX 36.939741) (EY 32.30487
5)

<<<DESTROYING CB>>> TIME = 14.0
MOVETO ** (OBJ OBJ) (R ROB) (I EYE) (CDX 33) (CDY 33) (CX 50.0) (CY 3
0.0) (CANG 180) (CRES 5) (EANG 169.99201) (EX 36.939741) (EY 32.30487
5)

<<<CREATING CB>>> TIME = 14.0
GOTO ** (R ROB) (CXT 36.939741) (CYT 32.304875) (CXF 50.0) (CYF 30.0)
 (ED 13.262081) (EXR -4.9239097) (EYR 0.86897210)

****TIME****
14.0
****EXPRS****
(OLD OBJ)
(CANSEE OBJ)
(FROM ROB 50.0 30.0)
(GOTO ROB 36.939741 32.304875)
(LAST-POS 50.0 30.0)
(FIND ROB REL)
(CHAR OBJ REL)
(TYPE OBJ OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ 33 33)
(STATE ROB SPOTTING)
(ANGLE ROB 169.99201)
(XRATE ROB -4.9239097)
(YRATE ROB 0.86897210)
(PATH ROB OFF)
(OLD-PT ROB 50.0 30.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 10.0 30.0)
(PATH-ANG 180)
****SKIRS****
(AT ROB 50.0 30.0)
****************

<<<DESTROYING CB>>> TIME = 16.652416
GOTO ** (R ROB) (CXT 36.939741) (CYT 32.304875) (CXF 50.0) (CYF 30.0)
 (ED 13.262081) (EXR -4.92390997) (EYR 0.86897210)

<<<CREATING CB>>> TIME = 16.652416
EXAMINE ** (OBJ OBJ) (OBJ-CHAR REL) (R ROB) (I EYE) (CDX 33) (CDY 33)
 (CX 36.939740) (CY 32.304875) (CANG 169.99201) (CRES 5)

<<<DESTROYING CB>>> TIME = 16.652416
EXAMINE ** (OBJ OBJ) (OBJ-CHAR REL) (R ROB) (I EYE) (CDX 33) (CDY 33)
 (CX 36.939740) (CY 32.304875) (CANG 169.99201) (CRES 5)

<<<CREATING CB>>> TIME = 16.652416
RECOGNIZE ** (R ROB) (OBJ-CHAR REL) (OBJ OBJ) (CX 33) (CY 33) (EDUM N
IL)
```

```
<<<DESTROYING CB>>> TIME = 16.652416
RECOGNIZE ** (R ROB) (OBJ-CHAR RED) (OBJ OBJ) (CX 33) (CY 33) (EDUM N
IL)

<<<CREATING CB>>> TIME = 16.652416
GOHOME ** (R ROB) (REYE EYE) (CB 20) (CX 36.939740) (CY 32.304875) (E
Y 10.0) (EANG 211.12435)

<<<DESTROYING CB>>> TIME = 16.652416
GOHOME ** (R ROB) (REYE EYE) (CB 20) (CX 36.939740) (CY 32.304875) (E
Y 10.0) (EANG 211.12435)

<<<CREATING CB>>> TIME = 16.652416
GOTO ** (R ROB) (CXT 0) (CYT 10.0) (CXE 36.939740) (CYE 32.304875) (E
D 43.151499) (EXR -4.2802383) (EYR -2.5044844)

<<<DESTROYING CB>>> TIME = 25.282716
GOTO ** (R ROB) (CXT 0) (CYT 10.0) (CXE 36.939740) (CYE 32.304875) (E
D 43.151499) (EXR -4.2802383) (EYR -2.5044844)

COMMAND: PICTURE

****TIME****
25.282716
****EXPRS****
(LOCATED RED OBJ 33 33)
(DESCRIBED OBJ RED)
(OLD OBJ)
(LAST-POS 36.939740 32.304875)
(CHAR OBJ RED)
(TYPE OBJ OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT ROB -0.47683715E-6 9.9999997)
(AT OBJ 33 33)
(STATE ROB SPOTTING)
(ANGLE ROB 169.99901)
(XRATE ROB 0)
(YRATE ROB 0)
(OLD-PT ROB 50.0 30.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD A0 60)
(PATH-POS 10.0 30.0)
(PATH-ANG 180)
****SKIPS****
******************

COMMAND: STOP

(****TERMINATED-AT-TIME**** 25.282716)
```

Run No. 2

```
*(HSIM)

==============================
HENDRIX SIMULATING SYSTEM
==============================

INPUT SCENARIO LIST: *(EVAL SLIST)
SPOT MONITOR-SPOT GOTO RESPOT MOVETO EXAMINE RECOGNIZE STOP-SEARCH SEARC
H GOHOME RESEARCH RESTART

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(ADD
*(TYPE OBJ1 OBJECT)
*(TYPE OBJ2 OBJECT)
*(AT OBJ1 25 7)
*(AT OBJ2 43 28)
*(CHAR OBJ1 BLOCK)
*(CHAR OBJ2 SPHERE)
*(NEW OBJ1)
*(NEW OBJ2))

COMMAND: *(TRACE *)

COMMAND: *AUTOSNAP

COMMAND: *(ADD (FIND ROB SPHERE))

COMMAND: *GO
////ERROR////
(LAST-POS * *) NOT FOUND

<<<CREATING CB>>> TIME = 0
SEARCH ** (R ROB) (CX 0) (CY 10) (CB 60) (CD 60) (CDV 20) (CANG 0) (E
NDANG 0) (EX 50.0) (EY 10)

<<<DESTROYING CB>>> TIME = 0
SEARCH ** (R ROB) (CX 0) (CY 10) (CB 60) (CD 60) (CDV 20) (CANG 0) (E
NDANG 0) (EX 50.0) (EY 10)

<<<CREATING CB>>> TIME = 0
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0) (CYF 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 0
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ1) (CDX
25) (CDY 7) (CANG 0) (CX 0) (CY 10) (CXR 5.0) (CYR 0.0) (CB 20) (CD 2
0) (EDIS 6.0)
```

```
****TIME****
0
****EXPRS****
(FROM ROB 0 10)
(GOTO ROB 50.0 10)
(LAST-POS 0 10)
(FIND ROB SPHERE)
(NEW OBJ2)
(NEW OBJ1)
(CHAR OBJ2 SPHERE)
(CHAR OBJ1 BLOCK)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ2 43 28)
(AT OBJ1 25 7)
(STATE ROB NORMAL)
(ANGLE ROB 0)
(XRATE ROB 5.0)
(YRATE ROB 0.0)
(PATH ROB ON)
(OLD-PT ROB 0 10)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 10)
(PATH-ANG 0)
****SKLRS****
(AT ROB 0.0 10.0)
****************

<<<DESTROYING CR>>> TIME = 1.2000000
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEY 60) (CEY 60) (OBJ OBJ1) (CDX
25) (CDY 7) (CANG 0) (CX 0) (CY 10) (CXR 5.0) (CYR 0.0) (CR 20) (CD 2
0) (EDIS 6.0)

<<<DESTROYING CR>>> TIME = 1.2000000
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0) (CYF 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)

<<<CREATING CR>>> TIME = 1.2000000
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ1) (CDX 25) (CDY 7) (CANG 0) (CX 6
.0) (CY 10.0) (CXR 0) (CYR 0) (CR 20) (CD 20)

<<<DESTROYING CR>>> TIME = 1.2000000
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ1) (CDX 25) (CDY 7) (CANG 0) (CX 6
.0) (CY 10.0) (CXR 0) (CYR 0) (CR 20) (CD 20)

<<<CREATING CR>>> TIME = 1.2000000
MOVETO ** (OBJ OBJ1) (R ROB) (I EYE) (CDX 25) (CDY 7) (CX 6.0) (CY 10
.0) (CANG 0) (CRES 5) (EANG 351.02736) (EX 21.048948) (EY 7.6238707)

<<<DESTROYING CR>>> TIME = 1.2000000
MOVETO ** (OBJ OBJ1) (R ROB) (I EYE) (CDX 25) (CDY 7) (CX 6.0) (CY 10
.0) (CANG 0) (CRES 5) (EANG 351.02736) (EX 21.048948) (EY 7.6238707)

<<<CREATING CR>>> TIME = 1.2000000
GOTO ** (R ROB) (CXT 21.048948) (CYT 7.6238707) (CXF 6.0) (CYF 10.0)
(ED 15.235381) (EXR 4.9288158) (EYR -0.77990430)
```

```
<<<DESTROYING CB>>> TIME = 4.2470762
GOTO ** (R ROB) (CXT 21.048948) (CYT 7.6238707) (CXF 6.0) (CYF 10.0)
(FD 15.235981) (FXR 4.9388158) (FYR -0.7798068)

<<<CREATING CB>>> TIME = 4.2470762
EXAMINE ** (OBJ OBJ1) (OBJ-CHAR BLOCK) (R ROB) (I EYE) (CDX 25) (CDY
7) (CX 21.048948) (CY 7.6238707) (CANG 351.02736) (CRES 5)

<<<DESTROYING CB>>> TIME = 4.2470762
EXAMINE ** (OBJ OBJ1) (OBJ-CHAR BLOCK) (R ROB) (I EYE) (CDX 25) (CDY
7) (CX 21.048948) (CY 7.6238707) (CANG 351.02736) (CRES 5)

<<<CREATING CB>>> TIME = 4.2470762
RESEARCH ** (R ROB) (CRX 21.048948) (CRY 7.6238707) (CX 6.0) (CY 10.0
) (ENDANG 171.02744)

<<<DESTROYING CB>>> TIME = 4.2470762
RESEARCH ** (R ROB) (CRX 21.048948) (CRY 7.6238707) (CX 6.0) (CY 10.0
) (ENDANG 171.02744)

<<<CREATING CB>>> TIME = 4.2470762
GOTO ** (R ROB) (CXT 6.0) (CYT 10.0) (CXF 21.048948) (CYF 7.6238707)
(FD 15.235981) (FXR -4.9388158) (FYR 0.7798068)

<<<DESTROYING CB>>> TIME = 7.2941524
GOTO ** (R ROB) (CXT 6.0) (CYT 10.0) (CXF 21.048948) (CYF 7.6238707)
(FD 15.235981) (FXR -4.9388158) (FYR 0.7798068)

<<<CREATING CB>>> TIME = 7.2941524
RESTART ** (R ROB) (CX 5.9999998) (CY 10.0) (CDX 6.0) (CDY 10.0) (CXT
 50.0) (CYT 10) (CANG 0)

<<<DESTROYING CB>>> TIME = 7.2941524
RESTART ** (R ROB) (CX 5.9999998) (CY 10.0) (CDX 6.0) (CDY 10.0) (CXT
 50.0) (CYT 10) (CANG 0)

<<<CREATING CB>>> TIME = 7.2941524
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 5.9999998) (CYF 10.0) (FD 44
.0) (FXR 5.0) (FYR 0.0)
```

```
<<<DESTROYING CB>>> TIME = 16.094152
GOTO ** (R ROB) (XT 50.0) (YT 10) (XF 5.9999998) (YF 10.0) (D 44
.0) (XR 5.0) (YR 0.0)

<<<CREATING CB>>> TIME = 16.094152
SEARCH ** (R ROB) (X 50.0) (Y 10.0) (B 60) (D 60) (DV 20) (ANG
0) (ENDANG 90) (X 50.0) (Y 30.0)

<<<DESTROYING CB>>> TIME = 16.094152
SEARCH ** (R ROB) (X 50.0) (Y 10.0) (B 60) (D 60) (DV 20) (ANG
0) (ENDANG 90) (X 50.0) (Y 30.0)

<<<CREATING CB>>> TIME = 16.094152
GOTO ** (R ROB) (XT 50.0) (YT 30.0) (XF 50.0) (YF 10.0) (D 20.0)
 (XR 0.0) (YR 5.0)

<<<CREATING CB>>> TIME = 16.094152
MONITOR-SPOT ** (R ROB) (REYE EYE) (EX 60) (EY 60) (OBJ OBJ2) (DX
43) (DY 28) (ANG 90) (X 50.0) (Y 10.0) (XR 0.0) (YR 5.0) (B 20
) (D 20) (EDIS 0)

<<<DESTROYING CB>>> TIME = 16.094152
MONITOR-SPOT ** (R ROB) (REYE EYE) (EX 60) (EY 60) (OBJ OBJ2) (DX
43) (DY 28) (ANG 90) (X 50.0) (Y 10.0) (XR 0.0) (YR 5.0) (B 20
) (D 20) (EDIS 0)

<<<DESTROYING CB>>> TIME = 16.094152
GOTO ** (R ROB) (XT 50.0) (YT 30.0) (XF 50.0) (YF 10.0) (D 20.0)
 (XR 0.0) (YR 5.0)

<<<CREATING CB>>> TIME = 16.094152
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ2) (DX 43) (DY 28) (ANG 90) (X
 50.0) (Y 10.0) (XR 0) (YR 0) (B 20) (D 20)

<<<DESTROYING CB>>> TIME = 16.094152
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ2) (DX 43) (DY 28) (ANG 90) (X
 50.0) (Y 10.0) (XR 0) (YR 0) (B 20) (D 20)

<<<CREATING CB>>> TIME = 16.094152
MOVETO ** (OBJ OBJ2) (R ROB) (I EYE) (DX 43) (DY 28) (X 50.0) (Y
10.0) (ANG 90) (RES 5) (EANG 111.25052) (EX 44.449805) (EY 24.27198
1)

<<<DESTROYING CB>>> TIME = 16.094152
MOVETO ** (OBJ OBJ2) (R ROB) (I EYE) (DX 43) (DY 28) (X 50.0) (Y
10.0) (ANG 90) (RES 5) (EANG 111.25052) (EX 44.449805) (EY 24.27198
1)

<<<CREATING CB>>> TIME = 16.094152
GOTO ** (R ROB) (XT 44.449805) (YT 24.271981) (XF 50.0) (YF 10.0)
 (D 15.313200) (XR -1.8122255) (YR 4.6400255)
```

```
****TIME****
16.094152
****EXPRS****
(DESCRIBED OBJ1 BLOCK)
(OLD OBJ2)
(OLD OBJ1)
(CANSEE OBJ2)
(FROM ROB 50.0 10.0)
(GOTO ROB 44.449805 24.271981)
(LAST-POS 50.0 10.0)
(FIND ROB SPHERE)
(CHAR OBJ2 SPHERE)
(CHAR OBJ1 BLOCK)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ2 43 28)
(AT OBJ1 25 7)
(STATE ROB SPOTTING)
(ANGLE ROB 111.25052)
(XRATE ROB -1.8122255)
(YRATE ROB 4.6600255)
(PATH ROB DEF)
(OLD-PT ROB 50.0 10.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 30.0)
(PATH-ANG 90)
****SKLRS****
(AT ROB 50.0 10.0)
**********

<<<DESTROYING CB>>> TIME = 19.156792
GOTO ** (R ROB) (CXT 44.449805) (CYT 24.271981) (CXF 50.0) (CYF 10.0)
 (ED 15.313900) (EXR -1.8122255) (EYR 4.6600255)

<<<CREATING CB>>> TIME = 19.156792
EXAMINE ** (OBJ OBJ2) (OBJ-CHAR SPHERE) (R ROB) (I EYE) (CDX 43) (CDY
 28) (CX 44.449805) (CY 24.271981) (CANG 111.25052) (CRES 5)

<<<DESTROYING CB>>> TIME = 19.156792
EXAMINE ** (OBJ OBJ2) (OBJ-CHAR SPHERE) (R ROB) (I EYE) (CDX 43) (CDY
 28) (CX 44.449805) (CY 24.271981) (CANG 111.25052) (CRES 5)

<<<CREATING CB>>> TIME = 19.156792
RECOGNIZE ** (R ROB) (OBJ-CHAR SPHERE) (OBJ OBJ2) (CX 43) (CY 28) (ED
UM NIL)

<<<DESTROYING CB>>> TIME = 19.156792
RECOGNIZE ** (R ROB) (OBJ-CHAR SPHERE) (OBJ OBJ2) (CX 43) (CY 28) (ED
UM NIL)

<<<CREATING CB>>> TIME = 19.156792
GOHOME ** (R ROB) (REYE EYE) (CB 20) (CX 44.449805) (CY 24.271981) (E
Y 10.0) (EANG 197.80084)

<<<DESTROYING CB>>> TIME = 19.156792
GOHOME ** (R ROB) (REYE EYE) (CB 20) (CX 44.449805) (CY 24.271981) (E
Y 10.0) (EANG 197.80084)
```

```
<<<CREATING CP>>> TIME = 19.156792
GOTO ** (R ROB) (XT 0) (YT 10.0) (XF 44.449805) (YF 24.271981) (E
D 46.684844) (EXR -4.7606248) (EYR -1.5285455)

<<<DESTROYING CP>>> TIME = 28.493761
GOTO ** (R ROB) (XT 0) (YT 10.0) (XF 44.449805) (YF 24.271981) (E
D 46.684844) (EXR -4.7606248) (EYR -1.5285455)

COMMAND: *PICTURE

****TIME****
28.493761
****EXPRS****
(LOCATED SPHERE OBJ2 43 28)
(DESCRIBED OBJ2 SPHERE)
(DESCRIBED OBJ1 BLOCK)
(OLD OBJ2)
(OLD OBJ1)
(LAST-POS 44.449805 24.271981)
(CHAR OBJ2 SPHERE)
(CHAR OBJ1 BLOCK)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT ROB -0.47689715E-6 9.9999957)
(AT OBJ2 43 28)
(AT OBJ1 25 7)
(STATE ROB SPOTTING)
(ANGLE ROB 111.25052)
(XRATE ROB 0)
(YRATE ROB 0)
(OLD-PT ROB 50.0 10.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 30.0)
(PATH-ANG 90)
****SKIRS****
******************

COMMAND: *STOP

(****TERMINATED-AT-TIME**** 28.493761)
```

Run No. 3

```
*(HSIM)

==============================
HENDRIX SIMULATING SYSTEM
==============================

INPUT SCENARIO LIST: *(EVAL PLIST)
SPOT MONITOR-SPOT GOTO RESPOT MOVETO EXAMINE RECOGNIZE STOP-SEARCH SE
ARCH GOHOME RESEARCH RESTART

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(ADD
*(TYPE OB.1 OBJECT)
*(TYPE OB.2 OBJECT)
*(TYPE OB.3 OBJECT)
*(TYPE OB.4 OBJECT)
*(AT OB.1 15 9)
*(AT OB.2 49 11)
*(AT OB.3 59 59)
*(AT OB.4 65 20)
*(CHAR OB.1 BICYCLE)
*(CHAR OB.2 AUTOMOBILE)
*(CHAR OB.3 MOTORCYCLE)
*(CHAR OB.4 FEMALE-ROBOT)
*(NEW OB.1)
*(NEW OB.2)
*(NEW OB.3)
*(NEW OB.4))

COMMAND: *(TRACE *)

COMMAND: *AUTOSNAP

COMMAND: *(ADD (FIND ROB FEMALE-ROBOT))

COMMAND: *GO
////ERROR////
(LAST-POS * *) NOT FOUND

<<<CREATING CB>>> TIME = 0
SEARCH ** (B ROB) (X 0) (DX 10) (BX 0) (BY 10) (CB 60) (CD 60) (CD
V 20) (ANG 0) (ENDANG 0) (EX 50.0) (EY 10)

<<<DESTROYING CB>>> TIME = 0
SEARCH ** (B ROB) (X 0) (DX 10) (BX 0) (BY 10) (CB 60) (CD 60) (CD
V 20) (ANG 0) (ENDANG 0) (EX 50.0) (EY 10)

<<<CREATING CB>>> TIME = 0
GOTO ** (B ROB) (EXT 50.0) (EYT 10) (EXB 0) (EYB 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 0
MONITOR-SPOT ** (B ROB) (BEYE EYE) (BX 60) (BY 60) (OBJ OB.2) (OBX
49) (OBY 11) (ANG 0) (DX 0) (DY 10) (DXR 5.0) (DYR 0.0) (CB 20) (CD
20) (EDIS 30.0)
```

```
<<<CREATING CB>>> TIME = 0
MONITOR-SPOT ** (P BOB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ1) (CDX
15) (CDY 9) (CANG 0) (CX 0) (CY 10) (CXR 5.0) (CYR 0.0) (CB 20) (CD 2
0) (EDIS 0)

<<<DESTROYING CB>>> TIME = 0
MONITOR-SPOT ** (P BOB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ1) (CDX
15) (CDY 9) (CANG 0) (CX 0) (CY 10) (CXR 5.0) (CYR 0.0) (CB 20) (CD 2
0) (EDIS 0)

<<<DESTROYING CB>>> TIME = 0
SPOT ** (P BOB) (CXT 50.0) (CYT 10) (CXF 0) (CYF 10) (ED 50.0) (EXR 5
.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 0
SPOT ** (P BOB) (REYE EYE) (OBJ OBJ1) (CDX 15) (CDY 9) (CEX 60) (CEY
60) (CANG 0) (CX 0.0) (CY 10.0) (CXR 0) (CYR 0) (CB 20) (CD 60)

<<<DESTROYING CB>>> TIME = 0
SPOT ** (P BOB) (REYE EYE) (OBJ OBJ1) (CDX 15) (CDY 9) (CEX 60) (CEY
60) (CANG 0) (CX 0.0) (CY 10.0) (CXR 0) (CYR 0) (CB 20) (CD 20)

<<<CREATING CB>>> TIME = 0
MOVETO ** (OBJ OBJ1) (P BOB) (I EYE) (CEX 60) (CEY 60) (CDX 15) (CDY
9) (CX 0.0) (CY 10.0) (CANG 0) (CDES 5) (EANG 358.19598) (EX 11.00885
9) (EY 9.966809E)

<<<DESTROYING CB>>> TIME = 0
MOVETO ** (OBJ OBJ1) (P BOB) (I EYE) (CEX 60) (CEY 60) (CDX 15) (CDY
9) (CX 0.0) (CY 10.0) (CANG 0) (CDES 5) (EANG 358.19598) (EX 11.00885
9) (EY 9.966809E)

<<<CREATING CB>>> TIME = 0
SPOT ** (P BOB) (CYT 11.008859) (CYT 9.966809E) (CXF 0.0) (CYF 10.0)
(ED 11.033995) (EXR 4.9889962) (EYR -0.3329984E)
```

```
****TIME****
0
****EXPRS****
(OLD OBJ1)
(ANSEE OBJ1)
(FROM-ROB 0.0 10.0)
(GOTO ROB 11.008859 9.2668906)
(LAST-POS 0 10)
(FIND ROB FEMALE-ROBOT)
(NEW OBJ4)
(NEW OBJ3)
(NEW OBJ2)
(CHAR OBJ4 FEMALE-ROBOT)
(CHAR OBJ3 MOTORCYCLE)
(CHAR OBJ2 AUTOMOBILE)
(CHAR OBJ1 BICYCLE)
(TYPE OBJ4 OBJECT)
(TYPE OBJ3 OBJECT)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ4 65 60)
(AT OBJ3 59 59)
(AT OBJ2 49 11)
(AT OBJ1 15 9)
(STATE ROB SMOKING)
(ANGLE ROB 356.18599)
(XRATE ROB 4.9999962)
(YRATE ROB -0.33259984)
(PATH ROB OFF)
(OLD-PT ROB 0.0 10.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 10)
(PATH-ANG 0)
****SKIPS****
(AT ROB 0.0 10.0)
******************

<<<DESTROYING CB>>> TIME = 2.2066591
GOTO ** (R ROB) (CXT 11.008859) (CYT 9.2668906) (CXF 0.0) (CYF 10.0)
(ED 11.033295) (EYR 4.9999962) (EYR -0.33259984)

<<<CREATING CB>>> TIME = 2.2066591
EXAMINE ** (OBJ1 OBJ1) (OBJ1-CHAR BICYCLE) (R ROB) (I EYE) (COX 15) (CO
Y 9) (CX 11.008859) (CY 9.2668906) (CANG 356.18599) (CRES 5)

<<<DESTROYING CB>>> TIME = 2.2066591
EXAMINE ** (OBJ1 OBJ1) (OBJ1-CHAR BICYCLE) (R ROB) (I EYE) (COX 15) (CO
Y 9) (CX 11.008859) (CY 9.2668906) (CANG 356.18599) (CRES 5)

<<<CREATING CB>>> TIME = 2.2066591
RESEARCH ** (R ROB) (CRX 11.008859) (CRY 9.2668906) (CX 0.0) (CY 10.0
) (ENDANG 176.18599)

<<<DESTROYING CB>>> TIME = 2.2066591
RESEARCH ** (R ROB) (CRX 11.008859) (CRY 9.2668906) (CX 0.0) (CY 10.0
) (ENDANG 176.18599)
```

```
<<<CREATING CB>>> TIME = 2.20-4591
GOTO ** (R ROB) (CXT 0.0) (CYT 10.0) (CXF 11.0082591) (CYF 9.2660906)
(ED 11.0332995) (EXR -4.9889862) (EYR 0.3325994E)

<<<DESTROYING CB>>> TIME = 4.1133182
GOTO ** (R ROB) (CXT 0.0) (CYT 10.0) (CXF 11.0082591) (CYF 9.2660906)
(ED 11.0332995) (EXR -4.9889862) (EYR 0.3325994E)

<<<CREATING CB>>> TIME = 4.4133182
RESTART ** (R ROB) (CX 0.0) (CY 10.0) (CDX 0.0) (CDY 10.0) (CXT 50.0)
 (CYT 10) (CANG 0)

<<<DESTROYING CB>>> TIME = 4.4133182
RESTART ** (R ROB) (CX 0.0) (CY 10.0) (CDX 0.0) (CDY 10.0) (CXT 50.0)
 (CYT 10) (CANG 0)

<<<CREATING CB>>> TIME = 4.4133182
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0.0) (CYF 10.0) (ED 50.0) (E
XR 5.0) (EYR 0.0)

<<<DESTROYING CB>>> TIME = 6.0
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ2) (CDX
 49) (CDY 11) (CANG 0) (CX 0) (CY 10) (CXR 5.0) (CYR 0.0) (CB 20) (CD
 20) (EDIS 30.0)

<<<DESTROYING CB>>> TIME = 6.0
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 0.0) (CYF 10.0) (ED 50.0) (E
XR 5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 6.0
RESPOT ** (R ROB) (CTX 50.0) (CTY 10) (REYE EYE) (CB 20) (CD 20) (OBJ
 OBJ2) (CDX 49) (CDY 11) (CEX 60) (CEY 60) (CANG 0) (CX 7.9334086) (C
Y 10.0) (CXR 0) (CYR 0)

<<<DESTROYING CB>>> TIME = 6.0
RESPOT ** (R ROB) (CTX 50.0) (CTY 10) (REYE EYE) (CB 20) (CD 20) (OBJ
 OBJ2) (CDX 49) (CDY 11) (CEX 60) (CEY 60) (CANG 0) (CX 7.9334086) (C
Y 10.0) (CXR 0) (CYR 0)

<<<CREATING CB>>> TIME = 6.0
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 7.9334086) (CYF 10.0) (ED 42
.066591) (EXR 5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 6.0
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ2) (CDX
 49) (CDY 11) (CANG 0) (CX 7.9334086) (CY 10.0) (CXR 5.0) (CYR 0.0) (C
B 20) (CD 20) (EDIS 22.066591)

<<<DESTROYING CB>>> TIME = 10.413318
MONITOR-SPOT ** (R ROB) (REYE EYE) (CEX 60) (CEY 60) (OBJ OBJ2) (CDX
 49) (CDY 11) (CANG 0) (CX 7.9334086) (CY 10.0) (CXR 5.0) (CYR 0.0) (C
B 20) (CD 20) (EDIS 22.066591)

<<<DESTROYING CB>>> TIME = 10.413318
GOTO ** (R ROB) (CXT 50.0) (CYT 10) (CXF 7.9334086) (CYF 10.0) (ED 42
.066591) (EXR 5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 10.413318
SPOT ** (R ROB) (REYE EYE) (OBJ OBJ2) (CDX 49) (CDY 11) (CEX 60) (CEY
 60) (CANG 0) (CX 30.0000000) (CY 10.0) (CXR 0) (CYR 0) (CB 20) (CD 20
)
```

```
<<<DESTROYING CB>>> TIME = 10.413318
SPOT ** (P POB) (PEYE EYE) (OBJ OBJ0) (CDY 49) (CDY 11) (CEX 60) (CEY
 60) (CANG 0) (CX 30.000000) (CY 10.0) (CXR 0) (CYR 0) (CR 20) (CD 20
)

<<<CREATING CB>>> TIME = 10.413318
MOVETO ** (OBJ OBJ0) (P POB) (I EYE) (CEX 60) (CEY 60) (CDX 49) (CDY
11) (CX 30.000000) (CY 10.0) (CANG 0) (CRES 5) (FANG 3.0127906) (EX 4
5.005528) (EY 10.789784)

<<<DESTROYING CB>>> TIME = 10.413318
MOVETO ** (OBJ OBJ0) (P POB) (I EYE) (CEX 60) (CEY 60) (CDX 49) (CDY
11) (CX 30.000000) (CY 10.0) (CANG 0) (CRES 5) (FANG 3.0127906) (EX 4
5.005528) (EY 10.789784)

<<<CREATING CB>>> TIME = 10.413318
GOTO ** (P POB) (CXT 45.005528) (CYT 10.789784) (CXF 30.000000) (CYF
10.0) (FD 15.026998) (FXR 4.9930887) (FYR 0.2628008)

<<<DESTROYING CB>>> TIME = 13.418578
GOTO ** (P POB) (CXT 45.005528) (CYT 10.789784) (CXF 30.000000) (CYF
10.0) (FD 15.026998) (FXR 4.9930887) (FYR 0.2628008)

<<<CREATING CB>>> TIME = 13.418578
EXAMINE ** (OBJ OBJ0) (OBJ-CHAR AUTOMOBILE) (P POB) (I EYE) (CDX 49)
(CDY 11) (CX 45.005528) (CY 10.789784) (CANG 3.0127906) (CRES 5)

<<<DESTROYING CB>>> TIME = 13.418578
EXAMINE ** (OBJ OBJ0) (OBJ-CHAR AUTOMOBILE) (P POB) (I EYE) (CDX 49)
(CDY 11) (CX 45.005528) (CY 10.789784) (CANG 3.0127906) (CRES 5)

<<<CREATING CB>>> TIME = 13.418578
RESEARCH ** (P POB) (CRX 45.005528) (CRY 10.789784) (CX 30.000000) (C
Y 10.0) (ENDANG 183.01286)

<<<DESTROYING CB>>> TIME = 13.418578
RESEARCH ** (P POB) (CRX 45.005528) (CRY 10.789784) (CX 30.000000) (C
Y 10.0) (ENDANG 183.01286)

<<<CREATING CB>>> TIME = 13.418578
GOTO ** (P POB) (CXT 30.000000) (CYT 10.0) (CXF 45.005528) (CYF 10.78
9784) (FD 15.026998) (FXR -4.9930887) (FYR -0.2628008)
```

```
****TIME****
13.418578
****EXPRS****
(DESCRIBED OBJ2 AUTOMOBILE)
(DESCRIBED OBJ1 BICYCLE)
(OLD OBJ2)
(OLD OBJ1)
(FROM ROB 45.005528 10.789784)
(GOTO ROB 30.000000 10.0)
(LAST-POS 0 10)
(FIND ROB FEMALE-ROBOT)
(NEW OBJ4)
(NEW OBJ3)
(CHAR OBJ4 FEMALE-ROBOT)
(CHAR OBJ3 MOTORCYCLE)
(CHAR OBJ2 AUTOMOBILE)
(CHAR OBJ1 BICYCLE)
(TYPE OBJ4 OBJECT)
(TYPE OBJ3 OBJECT)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ4 45 40)
(AT OBJ3 59 56)
(AT OBJ2 49 11)
(AT OBJ1 15 9)
(STATE ROB SPOTTING)
(ANGLE ROB 180.01284)
(XRATE ROB -4.993808B7)
(YRATE ROB -0.2828888B)
(PATH ROB OFF)
(OLD-PT ROB 30.000000 10.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 50.0 10)
(PATH-ANG 0)
****SKLRS****
(AT ROB 45.005528 10.789784)
******************

<<<DESTROYING (B)>>> TIME = 16.423837
GOTO ** (B ROB) (CXT 30.000000) (CYT 10.0) (CXF 45.005528) (CYF 10.78
9784) (ED 15.02829B) (EXR -4.993808B7) (EYR -0.2828888B)

<<<CREATING (B)>>> TIME = 16.423837
RESTART ** (B ROB) (CX 29.999999) (CY 10.0) (CDX 30.000000) (CDY 10.0
) (CXT 50.0) (CYT 10) (CANG 0)

<<<DESTROYING (B)>>> TIME = 16.423837
RESTART ** (B ROB) (CX 29.999999) (CY 10.0) (CDX 30.000000) (CDY 10.0
) (CXT 50.0) (CYT 10) (CANG 0)

<<<CREATING (B)>>> TIME = 16.423837
GOTO ** (B ROB) (CXT 50.0) (CYT 10) (CXF 29.999999) (CYF 10.0) (ED 20
.000000) (EYR 5.0) (EYR 0.0)
```

```
<<<DESTROYING CB>>> TIME = 20.423837
GOTO ** (R RDB) (XT 50.0) (YT 10) (XF 99.999994) (YF 10.0) (ED 20
.000000) (XR 5.0) (YR 0.0)

<<<CREATING CB>>> TIME = 20.423837
SEARCH ** (R RDB) (X 50.0) (Y 10.0) (PX 50.0) (PY 10) (B 60) (D
60) (DW 20) (ANG 0) (ENDANG 90) (X 50.0) (Y 30.0)

<<<DESTROYING CB>>> TIME = 20.423837
SEARCH ** (R RDB) (X 50.0) (Y 10.0) (PX 50.0) (PY 10) (B 60) (D
60) (DW 20) (ANG 0) (ENDANG 90) (X 50.0) (Y 30.0)

<<<CREATING CB>>> TIME = 20.423837
GOTO ** (R RDB) (XT 50.0) (YT 30.0) (XF 50.0) (YF 10.0) (ED 20.0)
(XR 0.0) (YR 5.0)

<<<CREATING CB>>> TIME = 20.423837
MONITOR-SPOT ** (R RDB) (REYE EYE) (EX 60) (EY 60) (OB | OB 0) (DX
.59) (DY 59) (ANG 90) (X 50.0) (Y 10.0) (XR 0.0) (YR 5.0) (CB 20
) (CD 20) (ENTS 30.0)

<<<DESTROYING CB>>> TIME = 24.423837
GOTO ** (R RDB) (XT 50.0) (YT 30.0) (XF 50.0) (YF 10.0) (ED 20.0)
(XR 0.0) (YR 5.0)

<<<CREATING CB>>> TIME = 24.423837
SEARCH ** (R RDB) (X 50.0) (Y 30.0) (PX 50.0) (PY 30.0) (B 60) (
D 60) (DW 20) (ANG 90) (ENDANG 180) (X 10.0) (Y 30.0)

<<<DESTROYING CB>>> TIME = 24.423837
SEARCH ** (R RDB) (X 50.0) (Y 30.0) (PX 50.0) (PY 30.0) (B 60) (
D 60) (DW 20) (ANG 90) (ENDANG 180) (X 10.0) (Y 30.0)

<<<CREATING CB>>> TIME = 24.423837
GOTO ** (R RDB) (XT 10.0) (YT 30.0) (XF 50.0) (YF 30.0) (ED 40.0)
(XR -5.0) (YR 0.0)

<<<DESTROYING CB>>> TIME = 24.423837
MONITOR-SPOT ** (R RDB) (REYE EYE) (EX 60) (EY 60) (OB | OB 0) (DX
59) (DY 59) (ANG 90) (X 50.0) (Y 10.0) (XR 0.0) (YR 5.0) (CB 20
) (CD 20) (ENTS 30.0)

<<<DESTROYING CB>>> TIME = 24.423837
GOTO ** (R RDB) (XT 10.0) (YT 30.0) (XF 50.0) (YF 30.0) (ED 40.0)
(XR -5.0) (YR 0.0)

<<<CREATING CB>>> TIME = 24.423837
RESPOT ** (R RDB) (TX 10.0) (TY 30.0) (REYE EYE) (CB 20) (CD 20) (D
B | OB 0) (DX 59) (DY 59) (EX 60) (EY 60) (ANG 180) (X 40.0) (Y
30.0) (XR 0) (YR 0)

<<<DESTROYING CB>>> TIME = 24.423837
RESPOT ** (R RDB) (TX 10.0) (TY 30.0) (REYE EYE) (CB 20) (CD 20) (D
B | OB 0) (DX 59) (DY 59) (EX 60) (EY 60) (ANG 180) (X 40.0) (Y
30.0) (XR 0) (YR 0)
```

```
<<<CREATING CB>>> TIME = 26.4 9997
GOTO ** (P POP) (?XT 10.0) (?YT 30.0) (?XF 40.0) (?YF 30.0) (FD 30.0)
 (FXP -5.0) (FYP 0.0)

<<<DESTROYING CB>>> TIME = 32.423838
GOTO ** (P POP) (?XT 10.0) (?YT 30.0) (?XF 40.0) (?YF 30.0) (FD 30.0)
 (FXP -5.0) (FYP 0.0)

<<<CREATING CB>>> TIME = 32.423838
SEARCH ** (P POP) (?X 9.9999988) (?Y 30.0) (?PX 10.0) (?PY 30.0) (CB
60) (CD 60) (?DV 20) (?ANG 180) (ENDANG 90) (FX 9.9999988) (FY 50.0)

<<<DESTROYING CB>>> TIME = 32.423838
SEARCH ** (P POP) (?X 9.9999988) (?Y 30.0) (?PX 10.0) (?PY 30.0) (CB
60) (CD 60) (?DV 20) (?ANG 180) (ENDANG 90) (FX 9.9999988) (FY 50.0)

<<<CREATING CB>>> TIME = 32.423838
GOTO ** (P POP) (?XT 9.9999988) (?YT 50.0) (?XF 9.9999988) (?YF 30.0)
 (FD 20.0) (FXP 0.0) (FYP 5.0)

<<<DESTROYING CB>>> TIME = 36.423838
GOTO ** (P POP) (?XT 9.9999988) (?YT 50.0) (?XF 9.9999988) (?YF 30.0)
 (FD 20.0) (FXP 0.0) (FYP 5.0)

<<<CREATING CB>>> TIME = 36.423838
SEARCH ** (P POP) (?X 9.9999988) (?Y 50.0) (?PX 9.9999988) (?PY 50.0)
 (CB 60) (CD 60) (?DV 20) (?ANG 90) (ENDANG 0) (FX 50.0) (FY 50.0)

<<<DESTROYING CB>>> TIME = 36.423838
SEARCH ** (P POP) (?X 9.9999988) (?Y 50.0) (?PX 9.9999988) (?PY 50.0)
 (CB 60) (CD 60) (?DV 20) (?ANG 90) (ENDANG 0) (FX 50.0) (FY 50.0)

<<<CREATING CB>>> TIME = 36.423838
GOTO ** (P POP) (?XT 50.0) (?YT 50.0) (?XF 9.9999988) (?YF 50.0) (FD
40.000001) (FXP 5.0) (FYP 0.0)

<<<CREATING CB>>> TIME = 36.423838
MONITOR-SPOT ** (P POP) (REYE EYE) (?EX 60) (?EY 60) (OBJ OBJ3) (?DX
59) (?DY 59) (?ANG 0) (?X 9.9999988) (?Y 50.0) (?XR 5.0) (?YR 0.0) (?
B 20) (?D 20) (ENDS 30.000001)

<<<DESTROYING CB>>> TIME = 42.423838
MONITOR-SPOT ** (P POP) (REYE EYE) (?EX 60) (?EY 60) (OBJ OBJ3) (?DX
59) (?DY 59) (?ANG 0) (?X 9.9999988) (?Y 50.0) (?XR 5.0) (?YR 0.0) (?
B 20) (?D 20) (ENDS 30.000001)

<<<DESTROYING CB>>> TIME = 42.423838
GOTO ** (P POP) (?XT 50.0) (?YT 50.0) (?XF 9.9999988) (?YF 50.0) (FD
40.000001) (FXP 5.0) (FYP 0.0)

<<<CREATING CB>>> TIME = 42.423838
SPOT ** (P POP) (REYE EYE) (OBJ OBJ3) (?DX 59) (?DY 59) (?EX 60) (?EY
 60) (?ANG 0) (?X 40.000001) (?Y 50.0) (?XR 0) (?YR 0) (?B 20) (?D 20
)
```

```
<<<DESTROYING CB>>> TIME = 42.423838
SPOT ** (B BOB) (PEYE EYE) (OBJ OBJ3) (COX 59) (COY 59) (CEX 60) (CEY
 60) (CANG 0) (CX 40.000001) (CY 50.0) (CXB 0) (CYB 0) (CB B0) (CD 20
)

<<<CREATING CB>>> TIME = 42.423838
MOVETO ** (OBJ OBJ3) (B BOB) (I EYE) (CEX 60) (CEY 60) (COY 59) (COY
 59) (CX 40.000001) (CY 50.0) (CANG 0) (CRES 5) (EANG 25.346191) (EX 5
5.385049) (EY 57.287672)

<<<DESTROYING CB>>> TIME = 42.423838
MOVETO ** (OBJ OBJ3) (B BOB) (I EYE) (CEX 60) (CEY 60) (COY 59) (COY
 59) (CX 40.000001) (CY 50.0) (CANG 0) (CRES 5) (EANG 25.346191) (EX 5
5.385049) (EY 57.287672)

<<<CREATING CB>>> TIME = 42.423838
GOTO ** (B BOB) (CXT 55.385049) (CYT 57.287672) (CXF 40.000001) (CYF
 50.0) (ED 17.023803) (EXB 4.518687) (EYB 2.140436)

<<<DESTROYING CB>>> TIME = 45.223599
GOTO ** (B BOB) (CXT 55.385049) (CYT 57.287672) (CXF 40.000001) (CYF
 50.0) (ED 17.023803) (EXB 4.518687) (EYB 2.140436)

<<<CREATING CB>>> TIME = 45.223599
RESEARCH ** (B BOB) (CBX 55.385050) (CBY 57.287672) (CX 40.000001) (C
Y 50.0) (ENDANG 205.34624)

<<<DESTROYING CB>>> TIME = 45.223599
RESEARCH ** (B BOB) (CBX 55.385050) (CBY 57.287672) (CX 40.000001) (C
Y 50.0) (ENDANG 205.34624)

<<<CREATING CB>>> TIME = 45.223599
GOTO ** (B BOB) (CXT 40.000001) (CYT 50.0) (CXF 55.385050) (CYF 57.28
7672) (ED 17.023804) (EXB -4.518687) (EYB -2.140436)
```

```
****TIME****
45.828599
****EXPR****
(DESCRIBED DB.4 AUTOMOBILE)
(DESCRIBED DB.1 BICYCLE)
(OLD DB.3)
(OLD DB.2)
(OLD DB.1)
(CANSEE DB.3)
(FROM ROB 55.385050 57.287672)
(GOTO ROB 40.000001 50.0)
(LAST-ROB 9.9999985 50.0)
(FIND ROB FEMALE-ROBOT)
(NEW DB.4)
(CHAR DB.4 FEMALE-ROBOT)
(CHAR DB.3 MOTORCYCLE)
(CHAR DB.2 AUTOMOBILE)
(CHAR DB.1 BICYCLE)
(TYPE DB.4 DB.ROOT)
(TYPE DB.3 DB.ROOT)
(TYPE DB.2 DB.ROOT)
(TYPE DB.1 DB.ROOT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT DB.4 65 60)
(AT DB.3 59 59)
(AT DB.2 49 11)
(AT DB.1 15 9)
(STATE ROB SPOTTING)
(ANGLE ROB 205.3466.24)
(XRATE ROB -4.5186670)
(YRATE ROB -2.1404360)
(PATH ROB OFF)
(OLD-PT ROB 40.000001 50.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-ROB 50.0 50.0)
(PATH-ANG 0)
****SKIRS****
(AT ROB 55.385050 57.287672)
**********

<<<DESTROYING CB>>> TIME = 49.233360
GOTO ** (R ROB) (CXT 40.000001) (CYT 50.0) (CXF 55.385050) (CYF 57.28
7672) (ED 17.023204) (EXR -4.5186670) (EYR -2.1404360)

<<<CREATING CB>>> TIME = 49.233360
RESTART ** (R ROB) (CX 40.000001) (CY 50.0) (CDX 40.000001) (CDY 50.0
) (CXT 50.0) (CYT 50.0) (CANG 0)

<<<DESTROYING CB>>> TIME = 49.233360
RESTART ** (R ROB) (CX 40.000001) (CY 50.0) (CDX 40.000001) (CDY 50.0
) (CXT 50.0) (CYT 50.0) (CANG 0)

<<<CREATING CB>>> TIME = 49.233360
GOTO ** (R ROB) (CXT 50.0) (CYT 50.0) (CXF 40.000001) (CYF 50.0) (ED
9.9999985) (EXR 5.0) (EYR 0.0)
```

```
<<<DESTROYING CB>>> TIME = 51.233359
GOTO ** (R RDR) (CXT 50.0) (CYT 50.0) (CXF 40.000001) (CYF 50.0) (ED
9.9999985) (EXR 5.0) (EYR 0.0)

<<<CREATING CB>>> TIME = 51.233359
MOVETO ** (OBJ OBJ2) (R RDR) (I EYE) (CEX 60) (CEY 60) (CDX 59) (CDY
59) (CX 49.999999) (CY 50.0) (CANG 0) (CRES 5) (EANG 44.999996) (EX 5
6.171576) (EY 56.171577)

<<<DESTROYING CB>>> TIME = 51.233359
MOVETO ** (OBJ OBJ3) (R RDR) (I EYE) (CEX 60) (CEY 60) (CDX 59) (CDY
59) (CX 49.999999) (CY 50.0) (CANG 0) (CRES 5) (EANG 44.999996) (EX 5
6.171576) (EY 56.171577)

<<<CREATING CB>>> TIME = 51.233359
GOTO ** (R RDR) (CXT 56.171576) (CYT 56.171577) (CXF 49.999999) (CYF
50.0) (ED 8.7279988) (EXR 3.5355340) (EYR 3.5355337)

<<<DESTROYING CB>>> TIME = 52.978945
GOTO ** (R RDR) (CXT 56.171576) (CYT 56.171577) (CXF 49.999999) (CYF
50.0) (ED 8.7279988) (EXR 3.5355340) (EYR 3.5355337)

<<<CREATING CB>>> TIME = 52.978945
EXAMINE ** (OBJ OBJ3) (OBJ-CHAR MOTORCYCLE) (R RDR) (I EYE) (CDX 59)
(CDY 59) (CX 56.171577) (CY 56.171577) (CANG 44.999996) (CRES 5)

<<<DESTROYING CB>>> TIME = 52.978945
EXAMINE ** (OBJ OBJ3) (OBJ-CHAR MOTORCYCLE) (R RDR) (I EYE) (CDX 59)
(CDY 59) (CX 56.171577) (CY 56.171577) (CANG 44.999996) (CRES 5)

<<<CREATING CB>>> TIME = 52.978945
RESEARCH ** (R RDR) (CRX 56.171577) (CRY 56.171577) (CX 49.999999) (C
Y 50.0) (ENDANG 224.99999)

<<<DESTROYING CB>>> TIME = 52.978945
RESEARCH ** (R RDR) (CRX 56.171577) (CRY 56.171577) (CX 49.999999) (C
Y 50.0) (ENDANG 224.99999)

<<<CREATING CB>>> TIME = 52.978945
GOTO ** (R RDR) (CXT 49.999999) (CYT 50.0) (CXF 56.171577) (CYF 56.17
1577) (ED 8.7279995) (EXR -3.5355340) (EYR -3.5355337)
```

```
****TIME****
59.978945
****EXPRS****
(DESCRIBED OBJ3 MOTORCYCLE)
(DESCRIBED OBJ2 AUTOMOBILE)
(DESCRIBED OBJ1 BICYCLE)
(OLD OBJ3)
(OLD OBJ2)
(OLD OBJ1)
(FROM ROB 56.171577 56.171577)
(GOTO ROB 49.999999 50.0)
(LAST-PNS 9 9999999 50.0)
(FIND ROB FEMALE-ROBOT)
(NEW OBJ4)
(CHAR OBJ4 FEMALE-ROBOT)
(CHAR OBJ3 MOTORCYCLE)
(CHAR OBJ2 AUTOMOBILE)
(CHAR OBJ1 BICYCLE)
(TYPE OBJ4 OBJECT)
(TYPE OBJ3 OBJECT)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT OBJ4 65 60)
(AT OBJ3 59 59)
(AT OBJ2 49 11)
(AT OBJ1 15 9)
(STATE ROB NORMAL)
(ANGLE ROB 224.999999)
(XRATE ROB -3.5355334)
(YRATE ROB -3.5355337)
(PATH ROB DEF)
(OLD-PT ROB 49.999999 50.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-PNS 50.0 50.0)
(PATH-ANG 0)
****SKLRS****
(AT ROB 56.171577 56.171577)
******************

<<<DESTROYING OBJ>>> TIME = 54.724531
GOTO ** (# ROB) (CXT 49.999999) (CYT 50.0) (CXF 56.171577) (CYF 56.17
1577) (D 8.7279295) (EXR -3.5355334) (EYR -3.5355337)

COMMAND: *(DEL (PATH ROB *))

COMMAND: *(ADD (PATH ROB DNG))

COMMAND: *GO

<<<CREATING OBJ>>> TIME = 54.724531
SEARCH ** (# ROB) (CX 49.999999) (CY 50.0) (CPX 50.0) (CPY 50.0) (CB
60) (CD 60) (CPW 20) (CANG 0) (ENDANG 90) (EX 49.999999) (EY 50.0)
```

```
<<<DESTROYING CB>>> TIME = 54.724531
SEARCH ** (P BOB) (CX 49.999999) (CY 50.0) (PX 50.0) (PY 50.0) (CB
 80) (CD 80) (CDV 20) (CANG 0) (ENDANG 90) (EX 49.999999) (EY 50.0)

<<<CREATING CB>>> TIME = 54.724531
STOP-SEARCH ** (CX 49.999999) (CY 50.0) (CL> 49.999999) (CLY 50.0) (P
 BOB) (NB.1-CHAR FEMALE-ROBOT)

<<<DESTROYING CB>>> TIME = 54.724531
STOP-SEARCH ** (CX 49.999999) (CY 50.0) (CL> 49.999999) (CLY 50.0) (P
 BOB) (NB.1-CHAR FEMALE-ROBOT)

<<<CREATING CB>>> TIME = 54.724531
BOHOME ** (P BOB) (PEYE EYE) (CB 20) (CX 49.999999) (CY 50.0) (EY 10.
0) (EANG 218.65988)

<<<DESTROYING CB>>> TIME = 54.724531
BOHOME ** (P BOB) (PEYE EYE) (CB 20) (CX 49.999999) (CY 50.0) (EY 10.
0) (EANG 218.65988)

<<<CREATING CB>>> TIME = 54.724531
BOTO ** (P BOB) (CXT 0) (CYT 10.0) (CXE 49.999999) (CYE 50.0) (ED 64.
031242) (EXR -3.9043439) (EYR -3.1234752)

<<<DESTROYING CB>>> TIME = 67.530779
BOTO ** (P BOB) (CXT 0) (CYT 10.0) (CXE 49.999999) (CYE 50.0) (ED 64.
031242) (EXR -3.9043439) (EYR -3.1234752)

COMMAND: *PICTURE
```

```
****TIME****
67.530779
****EXPRS****
(MNTLOCATED FEMALE-ROBOT OBJECT WITHIN FIELD)
(DESCRIBED OBJ3 MNTOBCYCLE)
(DESCRIBED OBJ2 AUTOMOBILE)
(DESCRIBED OBJ1 BICYCLE)
(OLD OBJ3)
(OLD OBJ2)
(OLD OBJ1)
(LAST-POS 49.999999 50.0)
(NEW OBJ4)
(CHAR OBJ4 FEMALE-ROBOT)
(CHAR OBJ3 MNTOBCYCLE)
(CHAR OBJ2 AUTOMOBILE)
(CHAR OBJ1 BICYCLE)
(TYPE OBJ4 OBJECT)
(TYPE OBJ3 OBJECT)
(TYPE OBJ2 OBJECT)
(TYPE OBJ1 OBJECT)
(TYPE ROB ROBOT)
(TYPE EYE ROBOT-EYE)
(AT ROB 0.9536743E-6 10.000000)
(AT OBJ4 45 50)
(AT OBJ3 59 59)
(AT OBJ2 49 11)
(AT OBJ1 15 9)
(STATE ROB NORMAL)
(ANGLE ROB 90)
(XRATE ROB 0)
(YRATE ROB 0)
(PATH ROB ON)
(DIR-PT ROB 49.999999 50.0)
(VISION EYE 20 20)
(RESOLUTION EYE 5)
(DIMENSION FIELD 60 60)
(PATH-POS 49.999999 50.0)
(PATH-ANG 90)
****SKIRS****
******************

COMMAND: *STOP

(****TERMINATED-AT-TIME**** 67.530779)
```

APPENDIX D

Execution of the Hendrix World

```
*(RSIM)

================================
HENDRIX SIMULATING SYSTEM
================================

INPUT SCENARIO LIST: *(EVAL SLIST)
SETALARM MONITORSET AWAKE*ROBOT SOUNDALARM MONITORALARM DEFALARM SLEE
ROBOT TURNVALVE FILLBUCKET GRASP RELEASE MOVABILITY GOTO LOC

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(ADD
*(ALL-ACT RBT RBT-MU GOTO 40 50 20)
*(ALL-ACT RBT RBT-ARM RELEASE CLK)
*(ALL-ACT RBT RBT-ARM SETALARM CLK 11)
*(ALL-ACT RBT RBT-ARM GRASP CLK)
*(ALL-ACT RBT RBT-MU GOTO 50 150 20))

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM GRASP CLK) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-MU GOTO 50 150 20) FOUND

COMMAND: *(SNAPSHOT 5)

COMMAND: *(BREAK 10.5)

COMMAND: *(TRACE *)

COMMAND: *GO

<<<CREATING (R)>>> TIME = 0
GOTO ** (R RBT) (M RBT-MU) (CVT 50) (CVT 150) (CSPD 20) (CSPDL 20) (C
XF 40) (CVE 50) (FD 100.49875) (EVR 1.9900743) (EVR 19.900743)
```

```
****TIME****
5
****EVENTS****
(ALL-ACT RBT RBT-MU GOTO 50 150 20)
(ALL-ACT RBT RBT-ARM GRASP CLK)
(ALL-ACT RBT RBT-ARM SETALARM CLK 11)
(ALL-ACT RBT RBT-ARM RELEASE CLK)
(ALL-ACT RBT RBT-MU GOTO 40 50 20)
(AT CLK 50 150)
(AT BKT 100 100)
(AT VLV 150 50)
(AT TAP1 150 150)
(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MU MOBILITYUNIT)
(TYPE RBT-ARM ARM)
(MOVABLE CLK)
(MOVABLE RBT)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM OFF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 0)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0)
(TURNRATE VLV 0)
(MAXTURNRATEABS VLV 5)
(XRATE RBT 1.9900743)
(YRATE RBT 19.900743)
(SPEEDLIMIT RBT 20)
(STATE RBT AWAKE)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED CLK)
(NOTGRASPED BKT)
(NOTGRASPED VLV)
(NOTGRASPED TAP1)
****SKLPS****
(AT RBT 49.950371 149.50371)
**********

<<<DESTROYING CB>>> TIME = 5.0249378
GOTO ** (R RBT) (M RBT-MU) (CXT 50) (CYT 150) (CTPD 20) (CSPDL 20) (C
XF 40) (CYF 50) (ED 100.49875) (EXP 1.9900743) (EYP 19.900743)

<<<CREATING CB>>> TIME = 5.0249378
GRASP ** (A RBT-ARM) (B CLK) (R RBT) (CX 50.0) (CY 150.0)

<<<DESTROYING CB>>> TIME = 5.0249378
GRASP ** (A RBT-ARM) (B CLK) (R RBT) (CX 50.0) (CY 150.0
```

```
<<<CREATING CB>>> TIME = 5.0249378
SETALARM ** (R RBT) (A RBT-ARM) (K CLK) (CSTIME 11) (CX 50.0) (CY 150
.0)

<<<DESTROYING CB>>> TIME = 5.0249378
SETALARM ** (R RBT) (A RBT-ARM) (K CLK) (CSTIME 11) (CX 50.0) (CY 150
.0)

<<<CREATING CB>>> TIME = 5.0249378
MONITORALARM ** (K CLK) (CTM 11)

<<<CREATING CB>>> TIME = 5.0249378
RELEASE ** (A RBT-ARM) (B CLK) (R RBT)

<<<DESTROYING CB>>> TIME = 5.0249378
RELEASE ** (A RBT-ARM) (B CLK) (R RBT)

<<<CREATING CB>>> TIME = 5.0249378
GOTO ** (R RBT) (M RBT-MD) (CXT 40) (CYT 50) (CSPD 20) (CTPL 20) (CX
F 50.0) (CYF 150.0) (FD 100.49875) (EXR -1.9900743) (EYR -19.900743)

<<<DESTROYING CB>>> TIME = 10.049875
GOTO ** (R RBT) (M RBT-MD) (CXT 40) (CYT 50) (CSPD 20) (CSPN 20) (CX
F 50.0) (CYF 150.0) (FD 100.49875) (EXR -1.9900743) (EYR -19.900743)

============================
>>>BREAK AT 10.5
============================

COMMAND: *(APP
*(ALL-ACT RBT RBT-MD GOTO 150 150 20)
*(ALL-ACT RBT RBT-ARM GRASP BKT)
*(ALL-ACT RBT RBT-MD GOTO 100 100 20)
*(ALL-ACT RBT RBT-ARM RELEASE CLK)
*(ALL-ACT RBT <LEFPROBOT>)

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-MD GOTO 100 100 20) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM RELEASE CLK) FOUND

COMMAND: *PICTURE
```

```
****TIME****
10.5
****EXPRS****
(ALL-ACT RBT <LEFFBRDBDT>
(ALL-ACT RBT RBT-ARM RELEASE CLK>
(ALL-ACT RBT RBT-MU GOTO 100 100 20>
(ALL-ACT RBT RBT-ARM GRASP PKT>
(ALL-ACT RBT RBT-MU GOTO 150 150 20>
(AT RBT 40.0 50.0>
(AT CLK 50 150>
(AT BKT 100 100>
(AT VLV 160 50>
(AT TAP1 150 150>
(TYPE CLK CLOCK>
(TYPE RBT ROBOT>
(TYPE BKT BUCKET>
(TYPE VLV VALVE>
(TYPE TAP1 TAPY
(TYPE RBT-MU MOBILITYUNIT>
(TYPE RBT-ARM ARM>
(MOVABLE CLK>
(MOVABLE RBT>
(MOVABLE BKT>
(IMMOVABLE VLV>
(IMMOVABLE TAP1>
(ALARM SET CLK 11>
(ORIENTATION BKT UP>
(CONTENT BKT 0>
(CAPACITY BKT 100>
(CONTROL VLV TAP1>
(MAXRATE VLV 10>
(RATE VLV 0>
(TURNRATE VLV 0>
(MAXTURNRATEABS VLV 5>
(XRATE RBT 0>
(YRATE RBT 0>
(SPEEDLIMIT RBT 20>
(STATE RBT AWAKE>
(GRASPABLE CLK>
(GRASPABLE BKT>
(GRASPABLE VLV>
(NOTGRASPED CLK>
(NOTGRASPED BKT>
(NOTGRASPED VLV>
(NOTGRASPED TAP1>
****SKIES****
******************

COMMAND: *(SNAPSHOT 12 20>

COMMAND: *(EVAL (SETQ EPSILON 1.0E-4>>

1.0E-4

COMMAND: *GO

==============================
```

```
<<<CREATING CB>>> TIME = 10.5
SLEEPROBOT ** (R RBT)

<<<DESTROYING CB>>> TIME = 10.5
SLEEPROBOT ** (R RBT)

<<<DESTROYING CB>>> TIME = 11
MONITORALARM ** (K CLK) (CTM 11)

<<<CREATING CB>>> TIME = 11
SOUNDALARM ** (K CLK) (CSTIME 11)

<<<DESTROYING CB>>> TIME = 11
SOUNDALARM ** (K CLK) (CSTIME 11)

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM OFFALARM CLK) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM GRASP CLK) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-MU GOTO 50 150 20) FOUND

<<<CREATING CB>>> TIME = 11
AWAKENROBOT ** (R RBT) (K CLK) (CX 50) (CY 150)

<<<DESTROYING CB>>> TIME = 11
AWAKENROBOT ** (R RBT) (K CLK) (CX 50) (CY 150)

<<<CREATING CB>>> TIME = 11
GOTO ** (R RBT) (M RBT-MLD) (CXT 50) (CYT 150) (CSPD 20) (CSPDL 20) (C
XF 40.0) (CYF 50.0) (FD 100.498275) (FXR 1.9900743) (FYR 19.900743)
```

```
****TIME****
12
****EXPRS****
(ALL-ACT RBT RBT-MU GOTO 50 150 20)
(ALL-ACT RBT RBT-ARM GRASP CLK)
(ALL-ACT RBT RBT-ARM OFFALARM CLK)
(ALL-ACT RBT RBT-ARM RELEASE CLK)
(ALL-ACT RBT RBT-MU GOTO 100 100 20)
(ALL-ACT RBT RBT-ARM GRASP BKT)
(ALL-ACT RBT RBT-MU GOTO 150 150 20)
(AT CLK 50 150)
(AT BKT 100 100)
(AT VLV 140 50)
(AT TAP1 150 150)
(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MU MOBILITYUNIT)
(TYPE RBT-ARM ARM)
(MOVABLE CLK)
(MOVABLE RBT)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM SOUNDING CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 0)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0)
(TURNRATE VLV 0)
(MAXTURNRATEABS VLV 5)
(VRATE RBT 1.9900743)
(VRATE RBT 19.900743)
(SPEEDLIMIT RBT 20)
(STATE RBT AWAKE)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED CLK)
(NOTGRASPED BKT)
(NOTGRASPED VLV)
(NOTGRASPED TAP1)
****SKIPS****
(AT RBT 41.900074 69.90074)
**********

<<<DESTROYING CP>>> TIME = 16.024937
GOTO ** (R RBT) (M RBT-MU) (CXT 50) (CYT 150) (CSPD 20) (CSPDL 20) (C
XF 40.0) (CYF 50.0) (FD 100.49875) (FXR 1.9900743) (FYR 19.900743)

<<<CREATING CP>>> TIME = 16.024937
GRASP ** (A RBT-ARM) (B CLK) (R RBT) (CX 50.0) (CY 150.00000)
```

```
<<<DESTROYING CB>>> TIME = 16.024937
GRASP ** (A PRT-ARM) (B CLK) (B PRT) (CX 50.0) (CY 150.000000)

<<<CREATING CB>>> TIME = 16.024937
DEFALARM ** (B PRT) (A PRT-ARM) (K CLK) (CX 50) (CY 150)

<<<DESTROYING CB>>> TIME = 16.024937
DEFALARM ** (B PRT) (A PRT-ARM) (K CLK) (CX 50) (CY 150)

<<<CREATING CB>>> TIME = 16.024937
RELEASE ** (A PRT-ARM) (B CLK) (B PRT)

<<<DESTROYING CB>>> TIME = 16.024937
RELEASE ** (A PRT-ARM) (B CLK) (B PRT)

<<<CREATING CB>>> TIME = 16.024937
GOTO ** (B PRT) (M PRT-MUD) (CXT 100) (CYT 100) (CSPD 20) (CSPDL 20) (
CXF 50.0) (CYF 150.000000) (ET 70.710680) (EXP 14.142135) (EYP -14.142
135)

<<<DESTROYING CB>>> TIME = 19.560471
GOTO ** (B PRT) (M PRT-MUD) (CXT 100) (CYT 100) (CSPD 20) (CSPDL 20) (
CXF 50.0) (CYF 150.000000) (ET 70.710680) (EXP 14.142135) (EYP -14.142
135)

<<<CREATING CB>>> TIME = 19.560471
GRASP ** (A PRT-ARM) (B BKT) (B PRT) (CX 99.999999) (CY 100.000000)

<<<DESTROYING CB>>> TIME = 19.560471
GRASP ** (A PRT-ARM) (B BKT) (B PRT) (CX 99.999999) (CY 100.000000)

<<<CREATING CB>>> TIME = 19.560471
GOTO ** (B PRT) (M PRT-MUD) (CXT 150) (CYT 150) (CSPD 20) (CSPDL 20) (
CXF 99.999999) (CYF 100.000000) (ET 70.710678) (EXP 14.142135) (EYP 14
.142135)

<<<CREATING CB>>> TIME = 19.560471
LOC ** (B PRT) (A PRT-ARM) (B BKT) (CXF 100) (CYF 100) (CXP 14.142135
) (CYP 14.142135)
```

```
****TIME****
20
****EXPRS****
(GRASPING PRT PRT-ARM BKT)
(ALL-ACT PRT PRT-MU GOTO 150 150 20)
(AT CLK 50 150)
(AT VLV 160 50)
(AT TAP1 150 150)
(TYPE CLK CLOCK)
(TYPE PRT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE PRT-MU MOBILITYUNIT)
(TYPE PRT-ARM ARM)
(MOVABLE CLK)
(MOVABLE PRT)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM DEF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 0)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0)
(TURNRATE VLV 0)
(MAXTURNRATEARS VLV 5)
(XRATE PRT 14.142135)
(YRATE PRT 14.142135)
(SPEEDLIMIT PRT 20)
(STATE PRT AWAKE)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED CLK)
(NOTGRASPED VLV)
(NOTGRASPED TAP1)
****SELFS****
(AT BKT 106.215886 106.215886)
(AT PRT 106.215886 106.215886)
**********

<<<DESTROYING CR>>> TIME = 23.096005
GOTO ** (P PRT) (M PRT-MU) (XT 150) (YT 150) (SPD 20) (SPDL 20) (
DXF 99.999999) (DYF 100.00000) (ED 70.710678) (EXR 14.142135) (EYR 14
.142135)

<<<DESTROYING CR>>> TIME = 23.096005
LOC ** (P PRT) (A PRT-ARM) (P BKT) (DXF 100) (DYF 100) (EXR 14.142135
) (EYR 14.142135)


COMMAND: *(ADD
*(ALL-ACT PRT PRT-ARM TURNVALVE VLV 0.5 3)
*(ALL-ACT PRT PRT-ARM GRASP VLV)
*(ALL-ACT PRT PRT-MU GOTO 160 50 20)
*(ALL-ACT PRT PRT-ARM RELEASE BKT))
```

```
////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-MU SETD 160 50 20) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM RELEASE BKT) FOUND

COMMAND: *PICTURE

****TIME****
23,096005
****EXPRS****
(GRASPING RBT RBT-ARM BKT)
(ALL-ACT RBT RBT-ARM RELEASE BKT)
(ALL-ACT RBT RBT-MU SETD 160 50 20)
(ALL-ACT RBT RBT-ARM GRASP VLV)
(ALL-ACT RBT RBT-ARM TURNVALVE VLV 0.5 3)
(AT BKT 150.00000 149.99999)
(AT RBT 150.0 150.0)
(AT CLK 50 150)
(AT VLV 160 50)
(AT TAP1 150 150)
(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MU MOBILITYUNIT)
(TYPE RBT-ARM ARM)
(MOVABLE CLK)
(MOVABLE RBT)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM OFF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 0)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0)
(TURNRATE VLV 0)
(MAXTURNRATEPRS VLV 5)
(XRATE RBT 0)
(YRATE RBT 0)
(SPEEDLIMIT RBT 20)
(STATE RBT AWAKE)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED CLK)
(NOTGRASPED VLV)
(NOTGRASPED TAP1)
****SKLRS****
****************

COMMAND: *GO
```

```
<<<CREATING CB>>> TIME = 23.000005
RELEASE ** (A RBT-ARM) (B BKT) (P RBT)

<<<DESTROYING CB>>> TIME = 23.096005
RELEASE ** (A RBT-ARM) (B BKT) (P RBT)

<<<CREATING CB>>> TIME = 23.096005
GOTO ** (P RBT) (M RBT-MU) (CXT 160) (CYT 50) (CSPD 20) (CSPDL 20) (C
XF 150.0) (CYF 150.0) (ED 100.498750) (EXP 1.9900743) (EYP -19.900743)


<<<DESTROYING CB>>> TIME = 28.120943
GOTO ** (P RBT) (M RBT-MU) (CXT 160) (CYT 50) (CSPD 20) (CSPDL 20) (C
XF 150.0) (CYF 150.0) (ED 100.498750) (EXP 1.9900743) (EYP -19.900743)


<<<CREATING CB>>> TIME = 28.120943
GRASP ** (A RBT-ARM) (B VLV) (P RBT) (CX 160.0) (CY 49.999999)

<<<DESTROYING CB>>> TIME = 28.120943
GRASP ** (A RBT-ARM) (B VLV) (P RBT) (CX 160.0) (CY 49.999999)

<<<CREATING CB>>> TIME = 28.120943
TURNVALVE ** (P RBT) (A RBT-ARM) (V VLV) (CTURNRATE 0.5) (CDESIREDFLO
WRATE 3) (CINITIALFLOWRATE 0) (CMAXFLOWRATE 10) (CMAXTURNRATE 5) (CX
160.0) (CY 49.999999)

<<<CREATING CB>>> TIME = 28.120943
MOVABILITY ** (P RBT) (A RBT-ARM) (B VLV)

<<<DESTROYING CB>>> TIME = 34.120943
TURNVALVE ** (P RBT) (A RBT-ARM) (V VLV) (CTURNRATE 0.5) (CDESIREDFLO
WRATE 3) (CINITIALFLOWRATE 0) (CMAXFLOWRATE 10) (CMAXTURNRATE 5) (CX
160.0) (CY 49.999999)

<<<CREATING CB>>> TIME = 34.120943
FILLBUCKET ** (V VLV) (T TAP1) (B BKT) (CINITIALFLOWRATE 3.0) (CTURNR
ATE 0) (CCAPACITY 100) (CINITIALCONTENT 0) (CX 150) (CY 150)

<<<DESTROYING CB>>> TIME = 47.454277
FILLBUCKET ** (V VLV) (T TAP1) (B BKT) (CINITIALFLOWRATE 3.0) (CTURNR
ATE 0) (CCAPACITY 100) (CINITIALCONTENT 0) (CX 150) (CY 150)

COMMAND: *(SNAPSHOT 68)

COMMAND: *(ADD (ALL-ACT RBT RBT-ARM TURNVALVE V V -2 0))

COMMAND: *GO

<<<CREATING CB>>> TIME = 47.454277
TURNVALVE ** (P RBT) (A RBT-ARM) (V VLV) (CTURNRATE -2) (CDESIREDFLOW
RATE 0) (CINITIALFLOWRATE 3.0) (CMAXFLOWRATE 10) (CMAXTURNRATE 5) (CX
160.0) (CY 49.999999)
```

```
****TIME****
68
****EXPRS****
(GRASPING RBT RBT-ARM VLV)
(ALL-ACT RBT RBT-ARM TURNVALVE VLV -2 0)
(AT RBT 160 0 49.999999)
(AT BKT 150.00000 149.99999)
(AT CLK 50 150)
(AT VLV 160 50)
(AT TAP1 150 150)
(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MU MOBILITYUNIT)
(TYPE RBT-ARM ARM)
(MOVABLE CLK)
(MOVABLE BKT)
(IMMOVABLE RBT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM DEF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 100.000000)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(TURNRATE VLV -2)
(MAXTURNRATEABS VLV 5)
(XRATE RBT 0)
(YRATE RBT 0)
(SPEEDLIMIT RBT 20)
(STATE RBT AWAKE)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED BKT)
(NOTGRASPED CLK)
(NOTGRASPED TAP1)
****SKIPS****
(RATE VLV 1.9985554)
**********

<<<DESTROYING CB>>> TIME = 68.954277
TURNVALVE ** (R RBT) (A RBT-ARM) (V VLV) (TURNRATE -2) (DESIREDFLOW
RATE 0) (INITIALFLOWRATE 9.0) (MAXFLOWRATE 10) (MAXTURNRATE 5) (X
 160.0) (Y 49.999999)

COMMAND: *(ADD
*(ALL-ACT RBT RBT-MU GOTO 100 100 5)
*(ALL-ACT RBT RBT-ARM GRASP BKT)
*(ALL-ACT RBT RBT-MU GOTO 150 150 20)
*(ALL-ACT RBT RBT-ARM RELEASE VLV))
```

```
////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-MV GOTO 150 150 20) FOUND

////WARNING////
EXPR LIKE (ALL-ACT RBT RBT-ARM RELEASE VLV) FOUND

COMMAND: *GO

<<<DESTROYING CB>>> TIME = 68.954277
MOVABILITY ** (P RBT) (A RBT-ARM) (B VLV)

<<<CREATING CB>>> TIME = 68.954277
RELEASE ** (A RBT-ARM) (B VLV) (P RBT)

<<<DESTROYING CB>>> TIME = 68.954277
RELEASE ** (A RBT-ARM) (B VLV) (P RBT)

<<<CREATING CB>>> TIME = 68.954277
GOTO ** (P RBT) (M RBT-MV) (CXT 150) (CYT 150) (CSPD 20) (CSPDL 20) (
CXF 160.0) (CYF 49.999999) (ED 100.49975) (EXR -1.99007431) (EYR 19.90
0743)

<<<DESTROYING CB>>> TIME = 78.979214
GOTO ** (P RBT) (M RBT-MV) (CXT 150) (CYT 150) (CSPD 20) (CSPDL 20) (
CXF 160.0) (CYF 49.999999) (ED 100.49975) (EXR -1.99007431) (EYR 19.90
0743)

<<<CREATING CB>>> TIME = 78.979214
GRASP ** (A RBT-ARM) (B BKT) (P RBT) (CX 150.0) (CY 149.99999)

<<<DESTROYING CB>>> TIME = 78.979214
GRASP ** (A RBT-ARM) (B BKT) (P RBT) (CX 150.0) (CY 149.99999)

<<<CREATING CB>>> TIME = 78.979214
GOTO ** (P RBT) (M RBT-MV) (CXT 100) (CYT 100) (CSPD 5) (CSPDL 20) (C
XF 150.0) (CYF 149.99999) (ED 70.710678) (EXR -3.53553391) (EYR -3.535
53391)

<<<CREATING CB>>> TIME = 78.979214
LOC ** (P RBT) (A RBT-ARM) (B BKT) (CXF 150.00000) (CYF 149.99999) (C
XR -3.53553391) (CYR -3.53553391)

<<<DESTROYING CB>>> TIME = 88.121350
GOTO ** (P RBT) (M RBT-MV) (CXT 100) (CYT 100) (CSPD 5) (CSPDL 20) (C
XF 150.0) (CYF 149.99999) (ED 70.710678) (EXR -3.53553391) (EYR -3.535
53391)

<<<DESTROYING CB>>> TIME = 88.121350
LOC ** (P RBT) (A RBT-ARM) (B BKT) (CXF 150.00000) (CYF 149.99999) (C
XR -3.53553391) (CYR -3.53553391)

COMMAND: *(ADD
*(ALL-ACT RBT RBT-MV GOTO 40 50 30)
*(ALL-ACT RBT RBT-ARM RELEASE BKT))

COMMAND: *GO
```

```
<<<CREATING CB>>> TIME = 88.121350
RELEASE ** (A RBT-ARM) (B RBT) (R RBT)

<<<DESTROYING CB>>> TIME = 88.121350
RELEASE ** (R RBT-ARM) (B RBT) (R RBT)

<<<CREATING CB>>> TIME = 88.121350
GOTO ** (R RBT) (M RBT-MID) (CXT 40) (CYT 50) (CSPD 20) (CSPDL 20) (CX
F 99.999999) (CYF 99.999999) (ED 78.105486) (EXR -15.364425) (EYR -12
.803687)

<<<DESTROYING CB>>> TIME = 98.026474
GOTO ** (R RBT) (M RBT-MID) (CXT 40) (CYT 50) (CSPD 20) (CSPDL 20) (CX
F 99.999999) (CYF 99.999999) (ED 78.105486) (EXR -15.364425) (EYR -12
.803687)

COMMAND: *(ADD (ALL-ART RBT SLEEPROBOT))

COMMAND: *GO

<<<CREATING CB>>> TIME = 98.026474
SLEEPROBOT ** (R RBT)

<<<DESTROYING CB>>> TIME = 98.026474
SLEEPROBOT ** (R RBT)

COMMAND: *PICTURE
```

```
****TIME****
90.026474
****EVRS****
(AT RBT 40.000001 50.000001)
(AT BKT 100.00000 100.00000)
(AT CLK 50.150)
(AT VLV 150 50)
(AT TAP1.150 150)
(TYPE CLK CLOCK)
(TYPE RBT ROBOT)
(TYPE BKT BUCKET)
(TYPE VLV VALVE)
(TYPE TAP1 TAP)
(TYPE RBT-MH MOBILITYUNIT)
(TYPE RBT-ARM ARM)
(MOVABLE RBT)
(MOVABLE CLK)
(MOVABLE BKT)
(IMMOVABLE VLV)
(IMMOVABLE TAP1)
(ALARM OFF CLK)
(ORIENTATION BKT UP)
(CONTENT BKT 100.00000)
(CAPACITY BKT 100)
(CONTROL VLV TAP1)
(MAXRATE VLV 10)
(RATE VLV 0.0)
(TURNRATE VLV 0)
(MAXTURNRATEABS VLV 5)
(XRATE RBT 0)
(YRATE RBT 0)
(SPEEDLIMIT RBT 20)
(STATE RBT ASLEEP)
(GRASPABLE CLK)
(GRASPABLE BKT)
(GRASPABLE VLV)
(NOTGRASPED BKT)
(NOTGRASPED VLV)
(NOTGRASPED CLK)
(NOTGRASPED TAP1)
****SKLRS****
******************

COMMAND: *STOP

(****TERMINATED-AT-TIME**** 90.026474)
```

Execution of the Billiards World

```
*(RSIM)

==============================
HENDRIX SIMULATING SYSTEM
==============================

INPUT SCENARIO LIST: *(EVAL SLIST)
HIT NOHIT OFF-THE-TABLE OFF-THE-WALL BOUNCE,STOP-ROLL SHOOT MAYHIT

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(ADD (SHOOT CUE-BALL 50 0))

COMMAND: *PICTURE

****TIME****
0
****EXPRS****
(SHOOT CUE-BALL 50 0)
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 75 75)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
****SKLRS****
++++++++++++++++++

COMMAND: *(TRACE *)

COMMAND: *AUTOSNAP

COMMAND: *GO

<<<CREATING CB>>> TIME = 0
SHOOT ** (B CUE-BALL) (DSPD 50) (DANG 0) (DX 75) (DY 75) (TAB TABLE)
(DRAT 0) (DIAM 6.0325000) (EPAT 50) (EX 304.79999) (EY 75.0) (EDIS 22
6.78374)

****TIME****
0
****EXPRS****
(FROM CUE-BALL 75 75 50)
(ROLLING CUE-BALL 304.79999 75.0 0)
(SHOOT CUE-BALL 50 0)
(STATE CUE-BALL MOVING)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SKLRS****
(AT CUE-BALL 75 75)
(RATE CUE-BALL 50)
++++++++++++++++++
```

```
<<<DESTROYING CB>>> TIME = 9
SHOOT ** (B CUE-BALL) (SPD 50) (ANG 0) (X 75) (Y 75) (TAB TABLE)
(RAT 0) (DIAM 6.0325000) (ERAT 50) (EX 304.79999) (EY 75.0) (EDIS 22
6.78374)

<<<CREATING CB>>> TIME = 9
STOP-ROLL ** (B CUE-BALL) (RAT 0)

<<<DESTROYING CB>>> TIME = 9
STOP-ROLL ** (B CUE-BALL) (RAT 0)

COMMAND: *PICTURE

****TIME****
9
****EXPRS****
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 172.90048 75.0)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
****SKLRS****
******************

COMMAND: *(ADD (SHOOT CUE-BALL 200 147))

COMMAND: *GO

<<<CREATING CB>>> TIME = 9
SHOOT ** (B CUE-BALL) (SPD 200) (ANG 147) (X 172.90048) (Y 75.0)
(TAB TABLE) (RAT 0) (DIAM 6.0325000) (ERAT 200) (EX 53.716105) (EY 1
52.39999) (EDIS 139.09529)

****TIME****
9
****EXPRS****
(FROM CUE-BALL 172.90048 75.0 200)
(ROLLING CUE-BALL 53.716105 152.39999 147)
(SHOOT CUE-BALL 200 147)
(STATE CUE-BALL MOVING)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SKLRS****
(AT CUE-BALL 172.90048 75.0)
(RATE CUE-BALL 200)
******************
```

```
<<<DESTROYING CB>>> TIME = 9.6954761
SHOOT ** (B CUE-BALL) (SPD 100) (ANG 147) (X 172.900048) (Y 75.0)
(TAB TABLE) (RAT 0) (DAM 6.0325000) (FRT 200) (FX 53.716105) (FY 1
52.39999) (FDIS 199.09523)

<<<CREATING CB>>> TIME = 9.6954761
BOUNCE ** (B CUE-BALL) (X 53.716105) (Y 152.39999) (ANG 147) (BX
56.246393) (BY 150.75679) (DAM 6.0325000) (FANG 213)

<<<DESTROYING CB>>> TIME = 9.6954761
BOUNCE ** (B CUE-BALL) (X 53.716105) (Y 152.39999) (ANG 147) (BX
56.246393) (BY 150.75679) (DAM 6.0325000) (FANG 213)

<<<CREATING CB>>> TIME = 9.6954761
OFF-THE-TABLE ** (B CUE-BALL) (ANG 213) (SPD 100) (DAM 6.0325000)
(BX 56.246393) (BY 150.75679) (RAT 144.36190)

<<<DESTROYING CB>>> TIME = 9.6954761
OFF-THE-TABLE ** (B CUE-BALL) (ANG 213) (SPD 100) (DAM 6.0325000)
(BX 56.246393) (BY 150.75679) (RAT 144.36190)

COMMAND: *PICTURE

****TIME****
9.6954761
****EXPRS****
(OFF CUE-BALL TABLE)
(STATE CUE-BALL TIMER)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(TYPE TABLE BILLIARDS)
****SKIPS****
******************

COMMAND: *(DEL (OFF CUE-BALL TABLE) (STATE CUE-BALL TIMER))

COMMAND: *(ADD (AT CUE-BALL 75 75) (ON CUE-BALL TABLE) (STATE CUE-BALL
 STOPPED) (RATE CUE-BALL 0))

COMMAND: *(ADD (SHOOT CUE-BALL 100 57))

COMMAND: *GO

<<<CREATING CB>>> TIME = 9.6954761
SHOOT ** (B CUE-BALL) (SPD 100) (ANG 57) (X 75) (Y 75) (TAB TABLE
) (RAT 0) (DAM 6.0325000) (FRT 100) (FX 125.26419) (FY 152.39999)
(FDIS 89.272696)

<<<DESTROYING CB>>> TIME = 10.850021
SHOOT ** (B CUE-BALL) (SPD 100) (ANG 57) (X 75) (Y 75) (TAB TABLE
) (RAT 0) (DAM 6.0325000) (FRT 100) (FX 125.26419) (FY 152.39999)
(FDIS 89.272696)

<<<CREATING CB>>> TIME = 10.850021
BOUNCE ** (B CUE-BALL) (X 125.26419) (Y 152.39999) (ANG 57) (BX 1
28.62145) (BY 149.87040) (DAM 6.0325000) (FANG 303)
```

```
<<<DESTROYING CB>>> TIME = 10.850021
BOUNCE ** (B CUE-BALL) (CX 125.26419) (CY 152.99999) (CANG 57) (CBX 1
23.62145) (CBY 149.87040) (CDAM 6.0325000) (FANG 303)

<<<CREATING CB>>> TIME = 10.850021
OFF-THE-WALL ** (B CUE-BALL) (CA 303) (CR 56.290921) (CS 100)

<<<DESTROYING CB>>> TIME = 10.850021
OFF-THE-WALL ** (B CUE-BALL) (CA 303) (CR 56.290921) (CS 100)

<<<CREATING CB>>> TIME = 10.850021
SHOOT ** (B CUE-BALL) (CSPD 0) (CANG 303) (CX 123.62145) (CY 149.8704
0) (TAB TABLE) (CRAT 56.290921) (CDAM 6.0325000) (FRAT 56.290921) (FX
 220.94804) (FY 0) (FDIS 175.68950)

<<<DESTROYING CB>>> TIME = 19.850021
SHOOT ** (B CUE-BALL) (CSPD 0) (CANG 303) (CX 123.62145) (CY 149.8704
0) (TAB TABLE) (CRAT 56.290921) (CDAM 6.0325000) (FRAT 56.290921) (FX
 220.94804) (FY 0) (FDIS 175.68950)

<<<CREATING CB>>> TIME = 19.850021
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 19.850021
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

COMMAND: *PICTURE

****TIME****
19.850021
****EXPRS****
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 183.65065 57.434462)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
****SKIPS****
******************

COMMAND: *(ADD (SHOOT CUE-BALL 120 315))

COMMAND: *GO

<<<CREATING CB>>> TIME = 19.850021
SHOOT ** (B CUE-BALL) (CSPD 120) (CANG 315) (CX 183.65065) (CY 57.434
462) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (FRAT 120) (FX 241.08481)
(FY 0) (FDIS 78.2081.34)

<<<DESTROYING CB>>> TIME = 20.501755
SHOOT ** (B CUE-BALL) (CSPD 120) (CANG 315) (CX 183.65065) (CY 57.434
462) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (FRAT 120) (FX 241.08481)
(FY 0) (FDIS 78.2081.34)
```

```
<<<CREATING CB>>> TIME = 20.501755
BOUNCE ** (B CUE-BALL) (CX 241.08481) (CY 0) (CANG 315) (CBX 238.9521
9) (CBY 2.1332135) (CDAM 6.0325000) (EANG 45)

<<<DESTROYING CB>>> TIME = 20.501755
BOUNCE ** (B CUE-BALL) (CX 241.08481) (CY 0) (CANG 315) (CBX 238.9521
9) (CBY 2.1332135) (CDAM 6.0325000) (EANG 45)

<<<CREATING CB>>> TIME = 20.501755
OFF-THE-WALL ** (B CUE-BALL) (CA 45) (CR 88.716752) (CS 100)

<<<DESTROYING CB>>> TIME = 20.501755
OFF-THE-WALL ** (B CUE-BALL) (CA 45) (CR 88.716752) (CS 100)

<<<CREATING CB>>> TIME = 20.501755
SHOOT ** (B CUE-BALL) (CSPD 0) (CANG 45) (CY 238.95219) (CY 2.1332135
) (TAB TABLE) (CRAT 88.716752) (CDAM 6.0325000) (ERAT 88.716752) (EX
304.79999) (EY 67.981019) (EDIS 90.106610)

****TIME****
20.501755
****EXPRS****
(FROM CUE-BALL 238.95219 2.1332135 88.716752)
(ROLLING CUE-BALL 304.79999 67.981019 45)
(SHOOT CUE-BALL 0 45)
(STATE CUE-BALL MOVING)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SLPS****
(AT CUE-BALL 238.95219 2.1332135)
(RATE CUE-BALL 88.716752)
**********

<<<DESTROYING CB>>> TIME = 21.861199
SHOOT ** (B CUE-BALL) (CSPD 0) (CANG 45) (CX 238.95219) (CY 2.1332135
) (TAB TABLE) (CRAT 88.716752) (CDAM 6.0325000) (ERAT 88.716752) (EX
304.79999) (EY 67.981019) (EDIS 90.106610)

<<<CREATING CB>>> TIME = 21.861199
BOUNCE ** (B CUE-BALL) (CX 304.79999) (CY 67.981019) (CANG 45) (CBX 3
02.66723) (CBY 65.848251) (CDAM 6.0325000) (EANG 135)

<<<DESTROYING CB>>> TIME = 21.861199
BOUNCE ** (B CUE-BALL) (CX 304.79999) (CY 67.981019) (CANG 45) (CBX 3
02.66723) (CBY 65.848251) (CDAM 6.0325000) (EANG 135)

<<<CREATING CB>>> TIME = 21.861199
OFF-THE-WALL ** (B CUE-BALL) (CA 135) (CR 45.576769) (CS 100)

<<<DESTROYING CB>>> TIME = 21.861199
OFF-THE-WALL ** (B CUE-BALL) (CA 135) (CR 45.576769) (CS 100)
```

```
<<<CREATING CB>>> TIME = 21.81199
SHOOT ** (B CUE-BALL) (SPD 0) (ANG 135) (X 302.66729) (Y 65.84825
1) (TAB TABLE) (RAT 45.576769) (DAM 6.0325000) (ERAT 45.576769) (EX
 216.11593) (EY 152.39999) (EDIS 119.38608)

<<<DESTROYING CB>>> TIME = 30.861199
SHOOT ** (B CUE-BALL) (SPD 0) (ANG 135) (X 302.66729) (Y 65.84825
1) (TAB TABLE) (RAT 45.576769) (DAM 6.0325000) (ERAT 45.576769) (EX
 216.11593) (EY 152.39999) (EDIS 119.38608)

<<<CREATING CB>>> TIME = 30.861199
STOP-ROLL ** (B CUE-BALL) (RAT 0)

<<<DESTROYING CB>>> TIME = 30.861199
STOP-ROLL ** (B CUE-BALL) (RAT 0)

COMMAND: *PICTURE

****TIME****
30.861199
****EXPRS****
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 299.56548 128.95032)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
****SHAPES****
******************

COMMAND: *(DEL (AT * * *))

COMMAND: *(ADD (AT CUE-BALL 75 75) (AT BALL1 225 75) (STATE BALL1 STOP
PED) (ON BALL1 TABLE) (NOTBEHIND BALL1) (RATE BALL 0))

COMMAND: *(SNAPSHOT 31)

COMMAND: *(DEL (RATE BALL 0))

COMMAND: *(ADD (RATE BALL1 0))

COMMAND: *GO

COMMAND: *(ADD (SHOOT CUE-BALL 175 0))

COMMAND: *GO

<<<CREATING CB>>> TIME = 30.861199
SHOOT ** (B CUE-BALL) (SPD 175) (ANG 0) (X 75) (Y 75) (TAB TABLE)
 (RAT 0) (DAM 6.0325000) (ERAT 175) (EX 304.79999) (EY 75.0) (EDIS
226.78374)
```

```
<<<CREATING CB>>> TIME = 30.861199
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 75.0) (CANG 0) (B2 BALL1)
(C2X 225) (C2Y 75) (CDAM 6.0325000) (CEX 75) (CEY 75) (CRAT 175) (CDS
 146.98374) (CDEF 0.0) (CANG1 0.0) (CANG2 0.0)

****TIME****
30.861199
****EXPRS****
(FROM CUE-BALL 75 75 175)
(ROLLING CUE-BALL 304.79999 75.0 0)
(SHOOT CUE-BALL 175 0)
(STATE CUE-BALL MOVING)
(STATE BALL1 STOPPED)
(NOTBEHIND BALL1)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL1 225 75)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL1 0)
****SKIPS****
(AT CUE-BALL 75 75)
(RATE CUE-BALL 175)
**********

****TIME****
31
****EXPRS****
(FROM CUE-BALL 75 75 175)
(ROLLING CUE-BALL 304.79999 75.0 0)
(SHOOT CUE-BALL 175 0)
(STATE CUE-BALL MOVING)
(STATE BALL1 STOPPED)
(NOTBEHIND BALL1)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL1 225 75)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL1 0)
****SKIPS****
(AT CUE-BALL 98.815852 75.0)
(RATE CUE-BALL 165.98394)
**********

<<<DESTROYING CB>>> TIME = 31.927711
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 75.0) (CANG 0) (B2 BALL1)
(C2X 225) (C2Y 75) (CDAM 6.0325000) (CEX 75) (CEY 75) (CRAT 175) (CDS
 146.98374) (CDEF 0.0) (CANG1 0.0) (CANG2 0.0)
```

```
<<<DESTROYING CB>>> TIME = 31.927711
SHOOT ** (B CUE-BALL) (SPD 175) (ANG 0) (X 75) (Y 75) (TAB TABLE)
 (CRAT 0) (DIAM 6.0325000) (ERAT 175) (EX 304.79999) (EY 75.0) (EDIS
226.78374)

<<<CREATING CB>>> TIME = 31.927711
HIT ** (B CUE-BALL) (DIAM 6.0325000) (B2 BALL1) (X 225) (Y 75) (DX
 221.98374) (DY 75.0) (DRAT 102.20650) (A1 0.0) (A2 0.0) (DIFF 0.0
) (ERAT1 10.220650) (ERAT2 91.985851)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CB>>> TIME = 31.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 31.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<CREATING CB>>> TIME = 31.927711
SHOOT ** (B BALL1) (SPD 91.985851) (ANG 0.0) (X 225) (Y 75) (TAB
TABLE) (CRAT 0) (DIAM 6.0325000) (ERAT 91.985851) (EX 304.79999) (EY
75.0) (EDIS 78.787749)

<<<CREATING CB>>> TIME = 31.927711
SHOOT ** (B CUE-BALL) (SPD 10.220650) (ANG 0.0) (DX 221.98374) (DY
75.0) (TAB TABLE) (CRAT 0) (DIAM 6.0325000) (ERAT 10.220650) (EX 304.

<<<DESTROYING CB>>> TIME = 21.861199
OFF-THE-WALL ** (B CUE-BALL) (DA 135) (DR 45.576769) (DS 100)

<<<CREATING CB>>> TIME = 21.861199
SHOOT ** (B CUE-BALL) (SPD 0) (ANG 135) (DX 302.66729) (DY 65.84825
1) (TAB TABLE) (CRAT 45.576769) (DIAM 6.0325000) (ERAT 45.576769) (EX
 216.11593) (EY 152.39999) (EDIS 119.38608)

<<<DESTROYING CB>>> TIME = 30.861199
SHOOT ** (B CUE-BALL) (SPD 0) (ANG 135) (DX 302.66729) (DY 65.84825
1) (TAB TABLE) (CRAT 45.576769) (DIAM 6.0325000) (ERAT 45.576769) (EX
 216.11593) (EY 152.39999) (EDIS 119.38608)

<<<CREATING CB>>> TIME = 30.861199
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 30.861199
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

COMMAND: *PICTURE

****TIME****
30.861199
****EXPRS****
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 299.56548 128.95032)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARD)
(RATE CUE-BALL 0)
****SKIPS****
****************
```

```
COMMAND: *(DEL (AT * * *))

COMMAND: *(ADD (AT CUE-BALL 75 75) (AT BALL1 225 75) (STATE BALL1 STOP
PED) (ON BALL1 TABLE) (NOTBEHIND BALL1) (RATE BALL 0))

COMMAND: *(SNAPSHOT 31)

COMMAND: *(DEL (RATE BALL 0))

COMMAND: *(ADD (RATE BALL1 0))

COMMAND: *GO

COMMAND: *(ADD (SHOOT CUE-BALL 175 0))

COMMAND: *GO

<<<CREATING CB>>> TIME = 30.861199
SHOOT ** (B CUE-BALL) (SPD 175) (ANG 0) (X 75) (Y 75) (TAB TABLE)
 (CRAT 0) (CDAM 6.0325000) (ERAT 175) (EX 304.79999) (EY 75.0) (EDIS
226.78974)

<<<CREATING CB>>> TIME = 30.861199
MAYHIT ** (B1 CUE-BALL) (X 304.79999) (Y 75.0) (ANG 0) (B2 BALL1)
(2X 225) (2Y 75) (DAM 6.0325000) (EX 75) (EY 75) (ERAT 175) (EDS
 146.99374) (EDEF 0.0) (EANG1 0.0) (EANG2 0.0)

****TIME****
30.861199
****EXPRS****
(FROM CUE-BALL 75 75 175)
(ROLLING CUE-BALL 304.79999 75.0 0)
(SHOOT CUE-BALL 175 0)
(STATE CUE-BALL MOVING)
(STATE BALL1 STOPPED)
(NOTBEHIND BALL1)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL1 225 75)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL1 0)
****SKIPS****
(AT CUE-BALL 75 75)
(RATE CUE-BALL 175)
**********
```

```
****TIME****
31
****EXPRS****
(FROM CUE-BALL 75 75 175)
(ROLLING CUE-BALL 304.79999 75.0 0)
(SHOOT CUE-BALL 175 0)
(STATE CUE-BALL MOVING)
(STATE BALL1 STOPPED)
(NOTBEHIND BALL1)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL1 225 75)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL1 0)
****SKIPS****
(AT CUE-BALL 98.615852 75.0)
(RATE CUE-BALL 165.28394)
****************

<<<DESTROYING CR>>> TIME = 31.927711
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 75.0) (CANG 0) (B2 BALL1)
(C2X 225) (C2Y 75) (CDAM 6.0325000) (CEX 75) (CEY 75) (CRAT 175) (EDS
 146.98374) (EDFF 0.0) (EANG1 0.0) (EANG2 0.0)

<<<DESTROYING CR>>> TIME = 31.927711
SHOOT ** (B CUE-BALL) (CSPD 175) (CANG 0) (CX 75) (CY 75) (TAB TABLE)
 (CRAT 0) (CDAM 6.0325000) (ERAT 175) (EX 304.79999) (EY 75.0) (EDTS
226.78374)

<<<CREATING CR>>> TIME = 31.927711
HIT ** (B CUE-BALL) (CDAM 6.0325000) (B2 BALL1) (CX 225) (CY 75) (CBX
 221.98374) (CBY 75.0) (CRAT 102.20650) (CA1 0.0) (CA2 0.0) (CDFF 0.0
) (ERAT1 10.220650) (ERAT2 91.9858S1)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CR>>> TIME = 31.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CR>>> TIME = 31.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<CREATING CR>>> TIME = 31.927711
SHOOT ** (B BALL1) (CSPD 91.9858S1) (CANG 0.0) (CX 225) (CY 75) (TAB
TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 91.9858S1) (EX 304.79999) (EY
75.0) (EDTS 76.783749)

<<<CREATING CR>>> TIME = 31.927711
SHOOT ** (B CUE-BALL) (CSPD 10.220650) (CANG 0.0) (CX 221.98374) (CY
75.0) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 10.220650) (EX 304.
79999) (EY 75.0) (EDTS 79.800000)
```

```
****TIME****
91.927711
****EXPRS****
(BEHIND CUE-BALL)
(HASHIT CUE-BALL BALL1 225 75)
(FROM CUE-BALL 221.98374 75.0 10.220650)
(FROM BALL1 225 75 91.985851)
(ROLLING CUE-BALL 304.79999 75.0 0.0)
(ROLLING BALL1 304.79999 75.0 0.0)
(SHOOT BALL1 91.985851 0.0)
(SHOOT CUE-BALL 10.220650 0.0)
(STATE CUE-BALL MOVING)
(STATE BALL1 MOVING)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.99999)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SKLRS****
(AT CUE-BALL 221.98374 75.0)
(AT BALL1 225 75)
(RATE CUE-BALL 10.220650)
(RATE BALL1 91.985851)
****************

<<<DESTROYING CB>>> TIME = 32.985601
SHOOT ** (B BALL1) (SPD 91.985851) (ANG 0.0) (X 225) (Y 75) (TAB
TABLE) (RAT 0) (DAM 6.0325000) (ERAT 91.985851) (EX 304.79999) (EY
75.0) (DIS 76.783749)

<<<DESTROYING CB>>> TIME = 32.985601
HIT ** (B CUE-BALL) (DAM 6.0325000) (BB BALL1) (X 225) (Y 75) (BX
221.98374) (BY 75.0) (RAT 102.20650) (A1 0.0) (A2 0.0) (COEF 0.0
) (ERAT1 10.220650) (ERAT2 91.985851)

<<<CREATING CB>>> TIME = 32.985601
BOUNCE ** (B BALL1) (X 304.79999) (Y 75.0) (ANG 0.0) (BX 301.7837
4) (BY 75.0) (DAM 6.0325000) (EANG 180.0)

<<<DESTROYING CB>>> TIME = 32.985601
BOUNCE ** (B BALL1) (X 304.79999) (Y 75.0) (ANG 0.0) (BX 301.7837
4) (BY 75.0) (DAM 6.0325000) (EANG 180.0)

<<<CREATING CB>>> TIME = 32.985601
OFF-THE-WALL ** (B BALL1) (A 180.0) (R 53.913487) (S 100)

<<<DESTROYING CB>>> TIME = 32.985601
OFF-THE-WALL ** (B BALL1) (A 180.0) (R 53.913487) (S 100)

<<<CREATING CB>>> TIME = 32.985601
SHOOT ** (B BALL1) (SPD 0) (ANG 180.0) (X 301.78374) (Y 75.0) (TA
B TABLE) (RAT 53.913487) (DAM 6.0325000) (ERAT 53.913487) (EX 0) (E
Y 75.0) (DIS 298.76749)

<<<DESTROYING CB>>> TIME = 37.927711
SHOOT ** (B CUE-BALL) (SPD 10.220650) (ANG 0.0) (X 221.98374) (Y
75.0) (TAB TABLE) (RAT 0) (DAM 6.0325000) (ERAT 10.220650) (EX 304.
79999) (EY 75.0) (DIS 79.800000)
```

```
<<<CREATING CB>>> TIME = 37.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 37.927711
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<CREATING CB>>> TIME = 37.927711
MAYHIT ** (B1 BALL1) (CX 0) (CY 75.0) (CANG 180.0) (B2 CUE-BALL) (C2X
 240.43815) (C2Y 75.0) (CDIAM 6.0325000) (CEX 301.78974) (CEY 75.0) (C
RAT 53.913487) (EDIS 58.329342) (EOFF 0.0) (EANG1 180.0) (EANG2 180.0)


<<<DESTROYING CB>>> TIME = 39.397554
MAYHIT ** (B1 BALL1) (CX 0) (CY 75.0) (CANG 180.0) (B2 CUE-BALL) (C2X
 240.43815) (C2Y 75.0) (CDIAM 6.0325000) (CEX 301.78974) (CEY 75.0) (C
RAT 53.913487) (EDIS 58.329342) (EOFF 0.0) (EANG1 180.0) (EANG2 180.0)


<<<DESTROYING CB>>> TIME = 39.397554
SHOOT ** (B BALL1) (CSPD 0) (CANG 180.0) (CX 301.78974) (CY 75.0) (TA
B TABLE) (CRAT 53.913487) (CDIAM 6.0325000) (ERAT 53.913487) (EX 0) (E
Y 75.0) (EDIS 298.76749)

<<<CREATING CB>>> TIME = 39.397554
HIT ** (B BALL1) (CDIAM 6.0325000) (B2 CUE-BALL) (CX 240.43815) (CY 75
.0) (CBX 240.44026) (CBY 75.0) (CRAT 2.1000986) (CA1 180.0) (CA2 180.
0) (COFF 0.0) (ERAT1 0.21000986) (ERAT2 1.8900887)
////ERROR////
(ROLLING BALL1 * * *) NOT FOUND

<<<CREATING CB>>> TIME = 39.397554
STOP-ROLL ** (B BALL1) (CRAT 0)

<<<DESTROYING CB>>> TIME = 39.397554
STOP-ROLL ** (B BALL1) (CRAT 0)

<<<CREATING CB>>> TIME = 39.397554
SHOOT ** (B CUE-BALL) (CSPD 1.8900887) (CANG 180.0) (CX 240.43815) (C
Y 75.0) (TAB TABLE) (CRAT 0) (CDIAM 6.0325000) (ERAT 1.8900887) (EX 0)
 (EY 75.0) (EDIS 937.42190)

<<<CREATING CB>>> TIME = 39.397554
SHOOT ** (B BALL1) (CSPD 0.21000986) (CANG 180.0) (CX 240.44026) (CY
75.0) (TAB TABLE) (CRAT 0) (CDIAM 6.0325000) (ERAT 0.21000986) (EX 0)
(EY 75.0) (EDIS 240.44401)
```

```
****TIME****
39.397554
****EXPRS****
(BEHIND BALL 1)
(HASHIT BALL 1 CUE-BALL 240.43815 75.0)
(FROM BALL 1 243.46026 75.0 0.21008986)
(FROM CUE-BALL 240.43815 75.0 1.8908087)
(ROLLING BALL 1 0 75.0 180.0)
(ROLLING CUE-BALL 0 75.0 180.0)
(SHOOT CUE-BALL 1.8908087 180.0)
(SHOOT BALL 1 0.21008986 180.0)
(STATE BALL 1 MOVING)
(STATE CUE-BALL MOVING)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(ON BALL 1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SKIPS****
(AT BALL 1 243.46026 75.0)
(AT CUE-BALL 240.43815 75.0)
(RATE BALL 1 0)
(RATE CUE-BALL 1.8908087)
****************

<<<DESTROYING CB>>> TIME = 40.397554
SHOOT ** (B BALL 1) (SPD 0.21008986) (ANG 180.0) (CX 243.46026) (CY
75.0) (TAB TABLE) (CRAT 0) (DAM 6.0325000) (ERAT 0.21008986) (EX 0)
(EY 75.0) (DIS 240.44401)

<<<CREATING CB>>> TIME = 40.397554
STOP-ROLL ** (B BALL 1) (CRAT 0)

<<<DESTROYING CB>>> TIME = 40.397554
STOP-ROLL ** (B BALL 1) (CRAT 0)

<<<DESTROYING CB>>> TIME = 42.397554
SHOOT ** (B CUE-BALL) (SPD 1.8908087) (ANG 180.0) (CX 240.43815) (C
Y 75.0) (TAB TABLE) (CRAT 0) (DAM 6.0325000) (ERAT 1.8908087) (EX 0)
 (EY 75.0) (DIS 237.42190)

<<<DESTROYING CB>>> TIME = 42.397554
HIT ** (B BALL 1) (DAM 6.0325000) (B2 CUE-BALL) (CX 240.43815) (CY 75
.0) (CBX 243.46026) (CBY 75.0) (CRAT 2.1008986) (A1 180.0) (A2 180.
0) (COEF 0.0) (ERAT1 0.21008986) (ERAT2 1.8908087)

<<<CREATING CB>>> TIME = 42.397554
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 42.397554
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

COMMAND: *PICTURE
```

```
****TIME****
42.397554
****EXPR****
(STATE CUE-BALL STOPPED)
(STATE BALL1 STOPPED)
(NOTBEHIND BALL1)
(NOTBEHIND CUE-BALL)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 238.35847 75.0)
(AT BALL1 243.46026 75.0)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
(RATE BALL1 0)
****RULES****
******************

COMMAND: *(ADD (SHOOT CUE-BALL 50 1))

COMMAND: *GO

<<<CREATING CB>>> TIME = 42.397554
SHOOT ** (B CUE-BALL) (SPD 50) (ANG 1) (X 238.35847) (Y 75.0) (TA
B TABLE) (RAT 0) (DAM 6.0325000) (RAT 50) (X 304.79999) (Y 76.15
9829) (DIS 63.439297)

<<<CREATING CB>>> TIME = 42.397554
MAYHIT ** (B1 CUE-BALL) (X 304.79999) (Y 76.159829) (ANG 1) (B2 BA
LL1) (X 243.46026) (Y 75.0) (DAM 6.0325000) (X 238.35847) (Y
 75.0) (RAT 50) (DIS 2.0855413) (EOFF 0.89058876E-1) (ANG1 3.657372
1) (ANG2 273.65737)

<<<DESTROYING CB>>> TIME = 42.439265
MAYHIT ** (B1 CUE-BALL) (X 304.79999) (Y 76.159829) (ANG 1) (B2 BA
LL1) (X 243.46026) (Y 75.0) (DAM 6.0325000) (X 238.35847) (Y
 75.0) (RAT 50) (DIS 2.0855413) (EOFF 0.89058876E-1) (ANG1 3.657372
1) (ANG2 273.65737)

<<<DESTROYING CB>>> TIME = 42.439265
SHOOT ** (B CUE-BALL) (SPD 50) (ANG 1) (X 238.35847) (Y 75.0) (TA
B TABLE) (RAT 0) (DAM 6.0325000) (RAT 50) (X 304.79999) (Y 76.15
9829) (DIS 63.439297)

<<<CREATING CB>>> TIME = 42.439265
HIT ** (B CUE-BALL) (DAM 6.0325000) (B2 BALL1) (X 243.46026) (Y 75
.0) (BX 240.44969) (BY 75.036400) (RAT 49.165782) (A1 3.6573721)
(A2 273.65737) (OFF 0.89058876E-1) (RAT1 6.2230963) (RAT2 42.9426
86)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CB>>> TIME = 42.439265
STOP-ROLL ** (B CUE-BALL) (RAT 0)

<<<DESTROYING CB>>> TIME = 42.439265
STOP-ROLL ** (B CUE-BALL) (RAT 0)
```

```
<<<CREATING CB>>> TIME = 42.439265
SHOOT ** (B BALL1) (CSPD 42.942686) (CANG 273.65737) (CX 243.46026) (
CY 75.0) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 42.942686) (EX 2
48.25416) (EY 0) (EDIS 72.136803)

<<<CREATING CB>>> TIME = 42.439265
SHOOT ** (B CUE-BALL) (CSPD 6.2230963) (CANG 3.6573721) (CX 240.44369
) (CY 75.036400) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 6.223096
3) (EX 304.79999) (EY 79.150138) (EDIS 61.471394)

****TIME****
42.439265
****EXPRS****
(BEHIND CUE-BALL)
(HASHIT CUE-BALL BALL1 243.46026 75.0)
(FROM CUE-BALL 240.44369 75.036400 6.2230963)
(FROM BALL1 243.46026 75.0 42.942686)
(ROLLING CUE-BALL 304.79999 79.150138 3.6573721)
(ROLLING BALL1 248.25416 0 273.65737)
(SHOOT BALL1 42.942686 273.65737)
(SHOOT CUE-BALL 6.2230963 3.6573721)
(STATE CUE-BALL MOVING)
(STATE BALL1 MOVING)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
****SKLRS****
(AT CUE-BALL 240.44369 75.036400)
(AT BALL1 243.46026 75.0)
(RATE CUE-BALL 6.2230963)
(RATE BALL1 42.942686)
******************

<<<DESTROYING CB>>> TIME = 45.957039
SHOOT ** (B BALL1) (CSPD 42.942686) (CANG 273.65737) (CX 243.46026) (
CY 75.0) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 42.942686) (EX 2
48.25416) (EY 0) (EDIS 72.136803)

<<<DESTROYING CB>>> TIME = 45.957039
HIT ** (B CUE-BALL) (CDAM 6.0325000) (B2 BALL1) (CX 243.46026) (CY 75
.0) (CBX 240.44369) (CBY 75.036400) (CRAT 49.165782) (CA1 3.6573721)
(CA2 273.65737) (CDEF 0.89058876E-1) (ERAT1 6.2230963) (ERAT2 42.9426
86)

<<<CREATING CB>>> TIME = 45.957039
BOUNCE ** (B BALL1) (CX 248.25416) (CY 0) (CANG 273.65737) (CBX 248.0
6195) (CBY 3.0155715) (CDAM 6.0325000) (EANG 86.342628)

<<<DESTROYING CB>>> TIME = 45.957039
BOUNCE ** (B BALL1) (CX 248.25416) (CY 0) (CANG 273.65737) (CBX 248.0
6195) (CBY 3.0155715) (CDAM 6.0325000) (EANG 86.342628)

<<<CREATING CB>>> TIME = 45.957039
OFF-THE-WALL ** (B BALL1) (CA 86.342628) (CR 7.3545516) (CS 100)

<<<DESTROYING CB>>> TIME = 45.957039
OFF-THE-WALL ** (B BALL1) (CA 86.342628) (CR 7.3545516) (CS 100)
```

```
<<<CREATING CR>>> TIME = 45.95 039
SHOOT ** (B BALL1) (SPD 0) (ANG 86.9426828) (X 248.06195) (Y 3.015
5715) (TAB TABLE) (RAT 7.3545 16) (DAM 6.0325000) (ERAT 7.3545516)
(EX 257.61079) (EY 152.39999) (EDIS 146.67305)

<<<DESTROYING CR>>> TIME = 47.439265
SHOOT ** (B CUE-BALL) (SPD 6.2230963) (ANG 3.6573721) (X 240.44969
) (Y 75.084000) (TAB TABLE) (RAT 0) (DAM 6.0325000) (ERAT 6.223096
3) (EX 304.79999) (EY 79.150139) (EDIS 61.471994)

<<<CREATING CR>>> TIME = 47.439265
DP-ROLL ** (B CUE-BALL) (RAT 0)

<<<DESTROYING CR>>> TIME = 47.439265
STOP-ROLL ** (B CUE-BALL) (RAT 0)

<<<DESTROYING CR>>> TIME = 50.957039
SHOOT ** (B BALL1) (SPD 0) (ANG 86.9426828) (X 248.06195) (Y 3.015
5715) (TAB TABLE) (RAT 7.3545516) (DAM 6.0325000) (ERAT 7.3545516)
(EX 257.61079) (EY 152.39999) (EDIS 146.67305)

<<<CREATING CR>>> TIME = 50.957039
STOP-ROLL ** (B BALL1) (RAT 0)

<<<DESTROYING CR>>> TIME = 50.957039
STOP-ROLL ** (B BALL1) (RAT 0)

COMMAND: *PICTURE

****TIME****
50.957039
****EXPRS****
(STATE BALL1 STOPPED)
(STATE CUE-BALL STOPPED)
(NOTBEHIND CUE-BALL)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL1 248.64825 15.316696)
(AT CUE-BALL 250.85296 75.701736)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL1 0)
(RATE CUE-BALL 0)
****SKIPS****
****************

COMMAND: *(DEL (AT CUE-BALL * *) (AT BALL1 * *))

COMMAND: *(ADD (AT CUE-BALL 75 75) (AT BALL1 225 75))

COMMAND: *(ADD (AT BALL2 225 81) (STATE BALL2 STOPPED) (NOTBEHIND BALL
2) (ON BALL2 TABLE) (RATE BALL2 0))

COMMAND: *(ADD (SHOOT CUE-BALL 156 0.69999999))

COMMAND: *GO
```

```
<<<CREATING CB>>> TIME = 50.95 039
SHOOT ** (B CUE-BALL) (CSPD 150) (CANG 0.69999999) (CX 75) (CY 75) (T
AB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 150) (EX 304.79999) (EY 77.-
807983) (EDIS 226.80090

<<<CREATING CB>>> TIME = 50.95 039
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 77.807983) (CANG 0.6999999
9) (B2 BALL1) (C2X 225) (C2Y 75) (CDAM 6.0325000) (CEX 75) (CEY 75) (
CRAT 150) (EDS 146.98374) (EDFF 1.8328876) (EANG1 55.390389) (EANG2 3
25.39038)

<<<DESTROYING CB>>> TIME = 52.256858
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 77.807983) (CANG 0.6999999
9) (B2 BALL1) (C2X 225) (C2Y 75) (CDAM 6.0325000) (CEX 75) (CEY 75) (
CRAT 150) (EDS 146.98374) (EDFF 1.8328876) (EANG1 55.390389) (EANG2 3
25.39038)

<<<DESTROYING CB>>> TIME = 52.256858
SHOOT ** (B CUE-BALL) (CSPD 150) (CANG 0.69999999) (CX 75) (CY 75) (T
AB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 150) (EX 304.79999) (EY 77.
807983) (EDIS 226.80090

<<<CREATING CB>>> TIME = 52.256858
HIT ** (B CUE-BALL) (CDAM 6.0325000) (B2 BALL1) (CX 225) (CY 75) (CBX
 221.97278) (CBY 76.795897) (CRAT 79.206500) (CA1 55.390389) (CA2 325
.39038) (CDFF 1.8328876) (ERAT1 51.238993) (ERAT2 27.967506)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CB>>> TIME = 52.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 52.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<CREATING CB>>> TIME = 52.256858
SHOOT ** (B BALL1) (CSPD 27.967506) (CANG 325.39038) (CX 225) (CY 75)
 (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 27.967506) (EX 304.79999
) (EY 19.9294419) (EDIS 93.941517)

<<<CREATING CB>>> TIME = 52.256858
SHOOT ** (B CUE-BALL) (CSPD 51.238993) (CANG 55.390389) (CX 221.97278
) (CY 76.795897) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 51.23899
3) (EX 274.14731) (EY 152.39999) (EDIS 88.843223)

<<<CREATING CB>>> TIME = 52.256858
MAYHIT ** (B1 CUE-BALL) (CX 274.14731) (CY 152.39999) (CANG 55.390389
) (B2 BALL2) (C2X 225) (C2Y 81) (CDAM 6.0325000) (CEX 221.97278) (CEY
 76.795897) (CRAT 51.238993) (EDS 2.1643424) (EDFF 0.18252372) (EANG1
 60.836600) (EANG2 330.83660)

<<<DESTROYING CB>>> TIME = 52.299098
MAYHIT ** (B1 CUE-BALL) (CX 274.14731) (CY 152.39999) (CANG 55.390389
) (B2 BALL2) (C2X 225) (C2Y 81) (CDAM 6.0325000) (CEX 221.97278) (CEY
 76.795897) (CRAT 51.238993) (EDS 2.1643424) (EDFF 0.18252372) (EANG1
 60.836600) (EANG2 330.83660)

<<<DESTROYING CB>>> TIME = 52.299098
SHOOT ** (B CUE-BALL) (CSPD 51.238993) (CANG 55.390389) (CX 221.97278
) (CY 76.795897) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 51.23899
3) (EX 274.14731) (EY 152.39999) (EDIS 88.843223)
```

```
****TIME****
52.299098
****EXPRS****
(BEHIND CUE-BALL)
(HASHIT CUE-BALL BALL1 225 75)
(OFFCENTER CUE-BALL BALL2 60.836600 330.83660 0.18252372)
(MAYHIT CUE-BALL BALL2 225 81)
(FROM BALL1 225 75 27.967506)
(ROLLING BALL1 304.79999 19.929419 325.39038)
(SHOOT BALL1 27.967506 325.39038)
(STATE CUE-BALL (TMRD)
(STATE BALL1 MOVING)
(STATE BALL2 STOPPED)
(NOTBEHIND BALL2)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 223.20208 78.577240)
(AT BALL2 225 81)
(ON BALL2 TABLE)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 50.373256)
(RATE BALL2 0)
****SKIRS****
(AT BALL1 225.96402 74.338484)
(RATE BALL1 27.494965)
****************

<<<DESTROYING CR>>> TIME = 60.256858
SHOOT ** (B BALL1) (SPD 27.967506) (ANG 325.39038) (X 225) (Y 75)
 (TAB TABLE) (CRAT 0) (DIAM 6.0325000) (ERAT 27.967506) (EX 304.79999
) (EY 19.929419) (EDIS 93.941517)

<<<DESTROYING CR>>> TIME = 60.256858
HIT ** (B CUE-BALL) (DIAM 6.0325000) (B2 BALL1) (CX 225) (CY 75) (CBX
 221.97278) (CBY 76.795897) (CRAT 79.206500) (CA1 55.390399) (CA2 325
.39038) (COFF 1.8328878) (ERAT1 51.238999) (ERAT2 27.967506)

<<<CREATING CR>>> TIME = 60.256858
HIT ** (B CUE-BALL) (DIAM 6.0325000) (B2 BALL2) (CY 225) (CY 81) (CBX
 223.20208) (CBY 78.577240) (CRAT 50.373256) (CA1 60.836600) (CA2 330
.83660) (COFF 0.18252372) (ERAT1 7.7807599) (ERAT2 42.592496)

<<<CREATING CR>>> TIME = 60.256858
STOP-ROLL ** (B BALL1) (CRAT 0)

<<<DESTROYING CR>>> TIME = 60.256858
STOP-ROLL ** (B BALL1) (CRAT 0)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CR>>> TIME = 60.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CR>>> TIME = 60.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)
```

```
<<<CREATING CR>>> TIME = 60.256858
SHOOT ** (B BALL2) (CSPD 42.592496) (CANG 330.83660) (CX 225) (CY 81)
 (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 42.592496) (EX 304.79999
) (EY 36.467600) (EDIS 88.368511)

<<<CREATING CR>>> TIME = 60.256858
SHOOT ** (B CUE-BALL) (CSPD 7.7807598) (CANG 60.836600) (CX 223.20208
) (CY 78.577240) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 7.780759
8) (EX 264.39842) (EY 152.39999) (EDIS 81.523312)

<<<DESTROYING CR>>> TIME = 66.256858
SHOOT ** (B CUE-BALL) (CSPD 7.7807598) (CANG 60.836600) (CX 223.20208
) (CY 78.577240) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 7.780759
8) (EX 264.39842) (EY 152.39999) (EDIS 81.523312)

<<<CREATING CR>>> TIME = 66.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CR>>> TIME = 66.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CR>>> TIME = 69.256858
SHOOT ** (B BALL2) (CSPD 42.592496) (CANG 330.83660) (CX 225) (CY 81)
 (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (ERAT 42.592496) (EX 304.79999
) (EY 36.467600) (EDIS 88.368511)

<<<DESTROYING CR>>> TIME = 69.256858
HIT ** (B CUE-BALL) (CDAM 6.0325000) (B2 BALL2) (CX 225) (CY 81) (CBX
 223.20208) (CBY 78.577240) (CRAT 50.373256) (CA1 60.836600) (CA2 330
.83660) (COEF 0.18252372) (ERAT1 7.7807598) (ERAT2 42.592496)

<<<CREATING CR>>> TIME = 69.256858
STOP-ROLL ** (B BALL2) (CRAT 0)

<<<DESTROYING CR>>> TIME = 69.256858
STOP-ROLL ** (B BALL2) (CRAT 0)

COMMAND: *PICTURE

****TIME****
69.256858
****EXPRS****
(STATE BALL2 STOPPED)
(STATE CUE-BALL STOPPED)
(STATE BALL1 STOPPED)
(NOTBEHIND CUE-BALL)
(NOTBEHIND BALL2)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT BALL2 297.82465 40.360870)
(AT CUE-BALL 230.04817 90.845250)
(AT BALL1 269.42591 44.341742)
(ON BALL2 TABLE)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE BALL2 0)
(RATE CUE-BALL 0)
(RATE BALL1 0)
****SKLRS****
*****************
```

```
COMMAND: *(EVAL (SLANG 230.04817 90.845250 269.42591 44.341742))
310.25691

COMMAND: *(ADD (SHOOT CUE-BALL 5 310.256900))

COMMAND: *GO

<<<CREATING CB>>> TIME = 69.256858
SHOOT ** (B CUE-BALL) (CSPD 5) (CANG 310.25690) (CX 230.04817) (CY 90
.845250) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (CRAT 5) (CX 304.79999
) (CY 2.5660228) (CDIS 112.66026)

<<<CREATING CB>>> TIME = 69.256858
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 2.5660228) (CANG 310.25690
) (B2 BALL1) (C2X 269.42591) (C2Y 44.341742) (CDAM 6.0325000) (CRX 23
0.04817) (CRY 90.845250) (CRAT 5) (CDS 57.919836) (CDFF -0.18167495E-
3) (CANG1 310.25148) (CANG2 40.251487)

<<<DESTROYING CB>>> TIME = 74.256858
MAYHIT ** (B1 CUE-BALL) (CX 304.79999) (CY 2.5660228) (CANG 310.25690
) (B2 BALL1) (C2X 269.42591) (C2Y 44.341742) (CDAM 6.0325000) (CRX 23
0.04817) (CRY 90.845250) (CRAT 5) (CDS 57.919836) (CDFF -0.18167495E-
3) (CANG1 310.25148) (CANG2 40.251487)

<<<DESTROYING CB>>> TIME = 74.256858
SHOOT ** (B CUE-BALL) (CSPD 5) (CANG 310.25690) (CX 230.04817) (CY 90
.845250) (TAB TABLE) (CRAT 0) (CDAM 6.0325000) (CRAT 5) (CX 304.79999
) (CY 2.5660228) (CDIS 112.66026)

<<<CREATING CB>>> TIME = 74.256858
NOHIT ** (B CUE-BALL) (STAT LIMBO) (CDAM 6.0325000) (B2 BALL1) (CX 26
9.42591) (CY 44.341742) (CRX 235.46347) (CRY 84.450650)

<<<DESTROYING CB>>> TIME = 74.256858
NOHIT ** (B CUE-BALL) (STAT LIMBO) (CDAM 6.0325000) (B2 BALL1) (CX 26
9.42591) (CY 44.341742) (CRX 235.46347) (CRY 84.450650)
////ERROR////
(ROLLING CUE-BALL * * *) NOT FOUND

<<<CREATING CB>>> TIME = 74.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

<<<DESTROYING CB>>> TIME = 74.256858
STOP-ROLL ** (B CUE-BALL) (CRAT 0)

COMMAND: *PICTURE
```

```
****TIME****
74.256858
****EXPRS****
(STATE CUE-BALL STOPPED)
(STATE BALL2 STOPPED)
(STATE BALL1 STOPPED)
(NOTBEHIND CUE-BALL)
(NOTBEHIND BALL2)
(NOTBEHIND BALL1)
(DIAMETER BALL 6.0325000)
(SPEED BALL 100)
(DIMENSION TABLE 304.79999 152.39999)
(AT CUE-BALL 235.46347 84.450050)
(AT BALL2 297.82465 40.960870)
(AT BALL1 269.42591 44.941742)
(ON BALL2 TABLE)
(ON BALL1 TABLE)
(ON CUE-BALL TABLE)
(TYPE TABLE BILLIARDS)
(RATE CUE-BALL 0)
(RATE BALL2 0)
(RATE BALL1 0)
****SKLRS****
******************

COMMAND: *STOP

(****TERMINATED-AT-TIME**** 74.256858)
```

APPENDIX F

Execution of the Robot Arm World

```
*(HSIM)

==============================
HENDRIX SIMULATING SYSTEM
==============================

INPUT SCENARIO LIST: *(EVAL SLIST)
SHOULDER ELBOW CLAMP SERVO-TURN BAR UPARM DOWNARM ROTATE LOCR TURNTO
GOTO LOCG CLAMPTO GRASP RELEASE

INPUT SWM RELATION LIST: *(EVAL SWM)

COMMAND: *(DEL (AT SPHERE * * *))

COMMAND: *(ADD (AT SPHERE 50 50 5))

COMMAND: *(TRACE *)

COMMAND: *AUTOSNAP

COMMAND: *(ADD (GRASP RBT SPHERE))

COMMAND: *GO

<<<CREATING CB>>> TIME = 0
CLAMPTO ** (R RBT) (OBJ SPHERE) (KLMP CLAMP) (CX 50) (CY 50) (CRX 12.
5) (CRY 10) (EANG 316.84759) (EX 44.870462) (EY 44.529499)

<<<CREATING CB>>> TIME = 0
ROTD ** (R RBT) (CX 44.870462) (CY 44.529499) (CXF 12.5) (CYF 10) (ED
 47.329315) (EXR 3.4197053) (EYR 3.6476863)

<<<CREATING CB>>> TIME = 0
LOCG ** (R RBT) (P SHOULDER) (CXR 3.4197053) (CYR 3.6476863) (CX 10
(CY 10) (CZ 10)

<<<CREATING CB>>> TIME = 0
LOCG ** (R RBT) (P CLAMP) (CXR 3.4197053) (CYR 3.6476863) (CX 5) (CY
10) (CZ 5)

<<<CREATING CB>>> TIME = 0
LOCG ** (R RBT) (P ELBOW) (CXR 3.4197053) (CYR 3.6476863) (CX 5) (CY
10) (CZ 10)
```

```
****TIME****
0
****EXPRS****
(GOTO RBT 44,8704&P 44,528499)
(GRASP RBT SPHERE)
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE ARM DOWN)
(STATE GEAR 0)
(STATE WINCH 0)
(STATE WHEEL 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE BAR 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(RATE SHOULDER 0 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH BAR 90)
(LENGTH CABLE 50)
(LENGTH SHOULDER 5)
(ANGLE RBT 90)
(ANGLE FOREARM 270)
(ANGLE SHOULDER 180)
(GOAL CLAMP 0)
(APART CLAMP 5)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 3.41970&3)
(YRATE RBT 9.647&8&3)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT SPHERE 50 50 5)
****SKLRS****
(AT ELBOW 5.0 10.0 10)
(AT CLAMP 5.0 10.0 5)
(AT SHOULDER 10.0 10.0 10)
(AT RBT 12.5 10.0)
**********
```

```
<<<DESTROYING CB>>> TIME = 9.4658629
GOTO ** (R RBT) (CX 44.870462) (CY 44.528499) (CXF 12.5) (CYF 10) (ED
 47.3299315) (EXR 3.4197053) (EYR 3.6476863)

<<<DESTROYING CB>>> TIME = 9.4658629
LOCG ** (R RBT) (P ELBOW) (CXR 3.4197053) (CYR 3.6476863) (CX 5) (CY
10) (CZ 10)

<<<DESTROYING CB>>> TIME = 9.4658629
LOCG ** (R RBT) (P CLAMP) (CXR 3.4197053) (CYR 3.6476863) (CX 5) (CY
10) (CZ 5)

<<<DESTROYING CB>>> TIME = 9.4658629
LOCG ** (R RBT) (P SHOULDER) (CXR 3.4197053) (CYR 3.6476863) (CX 10)
(CY 10) (CZ 10)

<<<DESTROYING CB>>> TIME = 9.4658629
CLAMPTO ** (R RBT) (OBJ SPHERE) (KLMP CLAMP) (CX 50) (CY 50) (CRX 12.
5) (CRY 10) (FANG 316.84759) (EX 44.870462) (EY 44.528499)

<<<CREATING CB>>> TIME = 9.4658629
TURNTO ** (R RBT) (FANG 316.84759) (COLD 90) (FANG -133.15240)

<<<DESTROYING CB>>> TIME = 9.4658629
TURNTO ** (R RBT) (FANG 316.84759) (COLD 90) (FANG -133.15240)

<<<CREATING CB>>> TIME = 9.4658629
ROTATE ** (R RBT) (CANG -133.15240) (COLD 90) (CR 20) (CRAT -20)

<<<CREATING CB>>> TIME = 9.4658629
LOCR ** (R RBT) (CA 90) (CRAT -20) (P SHOULDER) (CRX 44.870461) (CRY
44.528499) (CX 42.370461) (CY 44.528499) (CZ 10) (EDIS 2.5)

<<<CREATING CB>>> TIME = 9.4658629
LOCR ** (R RBT) (CA 90) (CRAT -20) (P CLAMP) (CRX 44.870461) (CRY 44.
528499) (CX 37.370461) (CY 44.528499) (CZ 5) (EDIS 7.5)

<<<CREATING CB>>> TIME = 9.4658629
LOCR ** (R RBT) (CA 90) (CRAT -20) (P ELBOW) (CRX 44.870461) (CRY 44.
528499) (CX 37.370461) (CY 44.528499) (CZ 10) (EDIS 7.5)
```

```
****TIME****
9.4658629
****EXPRS****
(ROTATING FROM RBT 90 -20)
(GRASP RBT SPHERE)
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE ARM DOWN)
(STATE GEAR 0)
(STATE WINCH 0)
(STATE WHEEL 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE BAR 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(RATE SHOULDER 0 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH BAR 90)
(LENGTH CABLE 50)
(LENGTH SHOULDER 5)
(ANGLE FOREARM 270)
(ANGLE SHOULDER 180)
(GOAL CLAMP 0)
(APART CLAMP 5)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
****SKIES****
(ANGLE RBT 90.0)
(AT ELBOW 37.371215 44.528499 10)
(AT CLAMP 37.371215 44.528499 5)
(AT SHOULDER 42.370713 44.528499 10)
****************
```

```
<<<DESTROYING CR>>> TIME = 16.123488
ROTATE ** (R RBT) (ANG -133.15240) (OLD 90) (R 20) (RAT -20)

<<<DESTROYING CR>>> TIME = 16.123489
LOCR ** (R RBT) (A 90) (RAT -20) (R ELBOW) (RX 44.870461) (RY 44.
528499) (X 37.870461) (Y 44.528499) (Z 10) (DIS 7.5)

<<<DESTROYING CR>>> TIME = 16.123489
LOCR ** (R RBT) (A 90) (RAT -20) (R CLAMP) (RY 44.870461) (RY 44.
528499) (X 37.870461) (Y 44.528499) (Z 5) (DIS 7.5)

<<<DESTROYING CR>>> TIME = 16.123489
LOCR ** (R RBT) (A 90) (RAT -20) (R SHOULDER) (RX 44.870461) (RY
44.528499) (X 42.370461) (Y 44.528499) (Z 10) (DIS 2.5)

<<<CREATING CR>>> TIME = 16.123489
GRASP ** (R RBT) (OBJ SPHERE) (KLMP CLAMP) (DIS 5) (CAB CABLE) (W WI
NCH) (DAM 3) (X 50) (Y 50) (Z 5) (KX 50.000025) (KY 50.000030)
(KZ 5)

<<<CREATING CR>>> TIME = 16.123489
SERVO-TURN ** (S WINCH) (DEV SERVO-WINCH) (TAT -1) (SPD 5) (AFD CAB
LE) (LEN 50) (HI 100) (LO 0) (STP 0) (RAT -5)

<<<CREATING CR>>> TIME = 16.123489
CLAMP ** (S CABLE) (RAT -5) (K CLAMP) (DIS 5) (HI 5) (LO 0) (G 3
) (SPD 0.5) (RAT -0.5) (STP 3)
```

```
****TIME****
16.193493
****EXPRS****
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE WINCH -1)
(STATE ARM DOWN)
(STATE GEAR 0)
(STATE WHEEL 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE CLAMP -0.5)
(RATE CABLE -5)
(RATE BAR 0)
(RATE SHOULDER 0 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH BAR 90)
(LENGTH SHOULDER 5)
(ANGLE RBT 316.84759)
(ANGLE FOREARM 270)
(ANGLE SHOULDER 180)
(GOAL CLAMP 3)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT SHOULDER 46.580316 46.352342 10)
(AT CLAMP 50.000025 50.000030 5)
(AT ELBOW 50.000025 50.000030 10)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
****SKLRS****
(APART CLAMP 5.0)
(LENGTH CABLE 50.0)
**********
```

```
<<<DESTROYING CB>>> TIME = 20.123483
CLAMP ** (S CABLE) (CRAT -5) (K CLAMP) (CDIS 5) (CHI 5) (CLO 0) (CG 3
) (CSPD 0.5) (ERAT -0.5) (ESTP 3)

<<<DESTROYING CB>>> TIME = 20.123483
SERVO-TURN ** (S WINCH) (DEV SERVO-WINCH) (CTAT -1) (CSPD 5) (AFD CAB
LE) (CLEN 500) (CHI 1000) (CLO 0) (ESTP 0) (ERAT -5)

<<<DESTROYING CB>>> TIME = 20.123483
GRASP ** (P ART) (OBJ SPHERE) (KLMP CLAMP) (CDIS 5) (CAB CABLE) (W WI
NCH) (CDAM 3) (CX 500) (CY 500) (CZ 5) (CKX 50.000025) (CKY 50.000030)
(CKZ 5)

COMMAND: *(ADD (RELEASE SPHERE))

COMMAND: *PICTURE

****TIME****
20.123483
****EXPRS****
(RELEASE SPHERE)
(GRASPING CLAMP SPHERE)
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE WINCH 0)
(STATE ARM DOWN)
(STATE GEAR 0)
(STATE WHEEL 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE CLAMP 0)
(RATE CABLE 0)
(RATE BAR 0)
(RATE SHOULDER 0 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH CABLE 30.0)
(LENGTH BAR 90)
(LENGTH SHOULDER 5)
```

```
(ANGLE RBT 316.84759)
(ANGLE FOREARM 270)
(ANGLE SHOULDER 180)
(APART CLAMP 3.0)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT P0)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT SHOULDER 46.580316 46.358948 10)
(AT CLAMP 50.000025 50.000030 5)
(AT ELBOW 50.000025 50.000030 10)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
****SKLRS****
******************

COMMAND: *GO
////ERROR////
(GOAL CLAMP *) NOT FOUND

<<<CREATING CB>>> TIME = 20.123483
RELEASE ** (OBJ SPHERE) (KLMP CLAMP) (CAJR CABLE) (W WINCH)

<<<DESTROYING CB>>> TIME = 20.123483
RELEASE ** (OBJ SPHERE) (KLMP CLAMP) (CAR CABLE) (W WINCH)

<<<CREATING CB>>> TIME = 20.123483
SERVO-TURN ** (S WINCH) (DEV SERVO-WINCH) (CTAT 1) (CSPD 5) (AFD CABL
E) (CLEN 30.0) (CHI 100) (CLO 0) (ESTP 100) (ERAT 5)

<<<CREATING CB>>> TIME = 20.123483
CLAMP ** (S CABLE) (CRAT 5) (K CLAMP) (CDIS 3.0) (CHI 5) (CLO 0) (CG
0) (CSPD 0.5) (ERAT 0.5) (ESTP 5)

<<<DESTROYING CB>>> TIME = 24.123483
CLAMP ** (S CABLE) (CRAT 5) (K CLAMP) (CDIS 3.0) (CHI 5) (CLO 0) (CG
0) (CSPD 0.5) (ERAT 0.5) (ESTP 5)

<<<DESTROYING CB>>> TIME = 24.123483
SERVO-TURN ** (S WINCH) (DEV SERVO-WINCH) (CTAT 1) (CSPD 5) (AFD CABL
E) (CLEN 30.0) (CHI 100) (CLO 0) (ESTP 100) (ERAT 5)

COMMAND: *(ADD (UPARM RBT))

COMMAND: *GO

<<<CREATING CB>>> TIME = 24.123483
UPARM ** (R RBT) (G GEAR) (W WHEEL) (B BAR) (S ELBOW)

<<<CREATING CB>>> TIME = 24.123483
SHOULDER ** (S SHOULDER) (W WHEEL) (CTAT -1) (CSPD 3) (CANG 180) (CHI
 180) (CLO 150) (ERAT -3) (ESTP 150)

<<<CREATING CB>>> TIME = 24.123483
ELBOW ** (S SHOULDER) (W WHEEL) (CRAT -3) (CANG 180) (LBO ELBOW) (CX
50.000025) (CY 50.000030) (CZ 10) (CLEN 5) (CSZ 10)
```

```
****TIME****
84.129483
****EXPRS****
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE WHEEL -1)
(STATE WINCH 0)
(STATE ARM DOWN)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE SHOULDER -3 180)
(RATE CLAMP 0)
(RATE CABLE 0)
(RATE BAR 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH CABLE 50.0)
(LENGTH BAR 90)
(LENGTH SHOULDER 5)
(ANGLE RBT 316.84759)
(ANGLE FOREARM 270)
(GRAL CLAMP 0)
(APART CLAMP 5.0)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT SHOULDER 46.598316 46.352342 10)
(AT CLAMP 50.000025 50.000030 5)
(AT RBT 44.870461 44.598499)
(AT SPHERE 50 50 5)
****SKLRS****
(ANGLE SHOULDER -180.0)
(AT ELBOW 50.000025 50.000030 5.0005022)
******************
```

```
<<<DESTROYING CB>>> TIME = 34.123483
SHOULDER ** (S SHOULDER) (W WHEEL) (STAT -1) (SPD 3) (ANG 180) (HI
 180) (LO 150) (RAT -3) (STP 150)

<<<DESTROYING CB>>> TIME = 34.123483
UPARM ** (B BAR) (G GEAR) (W WHEEL) (B BAR) (S ELBOW)

<<<DESTROYING CB>>> TIME = 34.123483
ELBOW ** (S SHOULDER) (W WHEEL) (RAT -3) (ANG 180) (LBO ELBOW) (X
50.000025) (Y 50.000030) (Z 10) (LEN 5) (SZ 10)

<<<CREATING CB>>> TIME = 34.123483
SERVO-TURN ** (S GEAR) (DEV SERVO-GEAR) (STAT -1) (SPD 10) (AFD BAR)
 (LEN 90) (HI 90) (LO 0) (STP 0) (RAT -10)

<<<CREATING CB>>> TIME = 34.123483
BAR ** (B BAR) (RAT -10) (F FOREARM) (HI 270) (LO 240) (ANG 270)
(STP 240)

****TIME****
34.123483
****EXPRS****
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE WHEEL 0)
(STATE ARM UP)
(STATE GEAR -1)
(STATE WINCH 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE BAR -10)
(RATE SHOULDER 0 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150
(LENGTH CABLE 50.0)
(LENGTH SHOULDER 5)
```

```
(ANGLE SHOULDER 150.0)
(ANGLE RBT 316.84759)
(GOAL CLAMP 0)
(APART CLAMP 5.0)
(DIAMETER SPHERE 9)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(YRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT ELBOW 50.000025 50.000030 5.6699181)
(AT SHOULDER 46.520316 46.352342 10)
(AT CLAMP 50.000025 50.000030 5)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
****SKIRS****
(LENGTH BAR 90.0)
(ANGLE FOREARM 270.0)
******************

<<<DESTROYING CB>>> TIME = 43.123483
BAR ** (B BAR) (PRT -10) (F FOREARM) (CHI 270) (CLO 240) (ANG 270)
(STP 240)

<<<DESTROYING CB>>> TIME = 43.123483
SERVO-TURN ** (S GEAR) (DEV SERVO-GEAR) (STAT -1) (SPD 10) (AFD BAR)
 (CLEN 90) (CHI 90) (CLO 0) (STP 0) (PRT -10)

COMMAND: *(ADD (DOWNARM RBT))

COMMAND: *GO

<<<CREATING CB>>> TIME = 43.123483
DOWNARM ** (R RBT) (G GEAR) (W WHEEL) (B BAR) (S ELBOW)

<<<CREATING CB>>> TIME = 43.123483
SERVO-TURN ** (S GEAR) (DEV SERVO-GEAR) (STAT 1) (SPD 10) (AFD BAR)
(CLEN 0.0) (CHI 90) (CLO 0) (STP 90) (PRT 10)

<<<CREATING CB>>> TIME = 43.123483
BAR ** (B BAR) (PRT 10) (F FOREARM) (CHI 270) (CLO 240) (ANG 240.0)
 (STP 270)
```

```
****TIME****
43.123483
****EXPRS****
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE GEAR 1)
(STATE ARM UP)
(STATE WINCH 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE BAR 10)
(RATE SHOULDER 0 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH CABLE 50.0)
(LENGTH SHOULDER 5)
(ANGLE SHOULDER 150.0)
(ANGLE RBT 316.84759)
(GOAL CLAMP 0)
(APART CLAMP 5.0)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT ELBOW 50.000025 50.000030 5.6899181)
(AT SHOULDER 46.580316 46.352342 10)
(AT CLAMP 50.000025 50.000030 5)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
****SKLRS****
(LENGTH BAR 0.0)
(ANGLE FOREARM 240.0)
**********
```

```
<<<DESTROYING CB>>> TIME = 52.123483
BAR ** (B BAR) (CRAT 10) (F FOREARM) (CHI 270) (CLO 240) (CANG 240.0)
 (ESTP 270)

<<<DESTROYING CB>>> TIME = 52.123483
SERVO-TURN ** (S GEAR) (DEV SERVO-GEAR) (CTAT 1) (CSPD 10) (AFD BAR)
(CLEN 0.0) (CHI 90) (CLO 0) (ESTP 90) (ERAT 10)

<<<DESTROYING CB>>> TIME = 52.123483
DOWNARM ** (B BAT) (G GEAR) (W WHEEL) (B BAR) (S ELBOW)

<<<CREATING CB>>> TIME = 52.123483
SHOULDER ** (S SHOULDER) (W WHEEL) (CTAT 1) (CSPD 3) (CANG 150.0) (CH
I 180) (CLO 150) (ERAT 3) (ESTP 180)

<<<CREATING CB>>> TIME = 52.123483
ELBOW ** (S SHOULDER) (W WHEEL) (CRAT 3) (CANG 150.0) (LBO ELBOW) (CX
 50.000025) (CY 50.000030) (CZ 5.6699181) (CLEN 5) (CSZ 10)

****TIME****
52.123483
****EXPRS****
(TYPE GEAR SERVO-GEAR)
(TYPE BAR EXTENSION-BAR)
(TYPE WINCH SERVO-WINCH)
(TYPE CLAMP CLAMP)
(TYPE SHOULDER WHEEL)
(TYPE WHEEL SERVO-MOTOR)
(CONTROLS GEAR BAR)
(CONTROLS BAR FOREARM)
(CONTROLS CABLE CLAMP)
(CONTROLS WINCH CABLE)
(STATE GEAR 0)
(STATE WHEEL 1)
(STATE ARM DOWN)
(STATE WINCH 0)
(SPEED GEAR 10)
(SPEED CLAMP 0.5)
(SPEED WINCH 5)
(SPEED WHEEL 3)
(RATE SHOULDER 3 150.0)
(RATE BAR 0)
(RATE CLAMP 0)
(RATE CABLE 0)
(EXTENT BAR 90 0)
(EXTENT FOREARM 270 240)
(EXTENT CLAMP 5 0)
(EXTENT CABLE 100 0)
(EXTENT SHOULDER 180 150)
(LENGTH BAR 90.0)
(LENGTH CABLE 50.0)
(LENGTH SHOULDER 5)
```

```
(ANGLE FOREARM 270.0)
(ANGLE RBT 316.847594)
(GOAL CLAMP 0)
(APART CLAMP 5.0)
(DIAMETER SPHERE 3)
(MOVES WHEEL ELBOW)
(ROTATION RBT 20)
(XRATE RBT 0)
(YRATE RBT 0)
(NOTGRASPING CLAMP)
(NOTROTATING RBT)
(HASASPART RBT SHOULDER)
(HASASPART RBT CLAMP)
(HASASPART RBT ELBOW)
(AT SHOULDER 46.580316 46.352342 10)
(AT CLAMP 50.000025 50.000030 5)
(AT RBT 44.870461 44.528499)
(AT SPHERE 50 50 5)
++++SKLRS++++
(ANGLE SHOULDER 150.0)
(AT ELBOW 50.000025 50.000030 5.6699181)
++++++++++++++++++

<<<DESTROYING (B)>>> TIME = 62.123483
SHOULDER ** (S SHOULDER) (W WHEEL) (CTAT 1) (CSPD 3) (CANG 150.0) (CH
T 180) (CLD 150) (ERAT 3) (ESTP 180)

<<<DESTROYING (B)>>> TIME = 62.123483
ELBOW ** (S SHOULDER) (W WHEEL) (CRAT 3) (CANG 150.0) (LBD ELBOW) (CX
 50.000025) (CY 50.000030) (CZ 5.6699181) (CLEN 5) (CSZ 10)

COMMAND: *STOP

(++++TERMINATED-AT-TIME++++ 62.123483)
```

APPENDIX G

AUXFUN - Auxillary Functions

INDEX

(BUCKET A) - Storage function setting global variable BUCKET to A.
(CONVERT A) - Converts A into an angle (in degrees) less than or equal to 190°.
(COS A) - Returns the trigonometric cosine of angle A°.
(DEG R) - Tunrs radian R into degrees.
(DIS X1 Y1 X2 Y2) - Returns the distance between points (X1, Y1) and (X2, Y2).
(DUB I F) - Is used by E2X to raise E(2.71828) to integer I (using POWER) and multiplies this result by E raised to the fraction (using E X).
E - Is a global variable set to 2.71828.
(EEQAL ITM LIZ) - Is an EXPR used by EQALL.
(ENTN F N L) - Receives the function F (GT or LT) and recurs on L eventually returning the smallest or largest number.
(EPROGE BOD) - Is an EXPR used by PROGE to call EPROG3.
(EPROG3 VAR BOD) - Saves the value held in global variable VAR, evaluates BOD and then reassigns VAR its value.
(EQALL FEXPR) - Returns T if the first element in the FEXPR list is equal to any other element; otherwise NIL.
(EQF FEXPR) - Performs EQ with unevaluated arguments.
(E2X X) - Splits X up into integer and fractional parts and sends these to DUM.
(E X X) - Returns E raised to X if $X \leq 1$; otherwise returns (E2X X).
(FAC NUM) - Returns the factorial of NUM.
(FRAC X) - Returns the decimal portion of real number X.
(GE A B) - Returns T if $A \geq B$; otherwise NIL.
(GT N1 N2) - Retunrs T if $N1 > N2$; otherwise NIL.
(GTN FEXPR) - Returns the greatest parameter in the FEXPR list given to GTN.
(INT X) - Returns the integer portion of number X.
(INV N) - Returns the inverse of N.
(INVCOS A) - Returns the inverse cosine of A.
(INVSIN A) - Returns the inverse sine of A.
(INVTAN A) - Returns the inverse tangent of A.
(INV2 A) - Returns the inverse tangent of A if $A^2 > 1$ (called from INVTAN).
(KILL) - Wipes out all SWM relations held in the data structure of the Hendrix System.
(LE A B) - Returns T if $A \leq B$; otherwise NIL.
(LT N1 N2) - Returns T if $N1 < N2$; otherwise NIL.
(LTN FEXPR) - Returns the smallest numerical parameter in the FEXPR list given to LTN.
(NE A B) - Uses EQUAL to return T is $A \neq B$; otherwise NIL.
(NEG N) - Returns the value of N negated.
(ONEP N) - Uses EQUAL to return T if $N = 1$. otherwise NIL.
PI - Global variable set to 3.14159.
(POWER NUM RAISE) - Raises NUM to power POWER.
(PRGO FEXPR) - FEXPR identical to PROG used in place of it for TRACE.
(PROGE FEXPR) - Saves the current value of EPSILON, executes the body

of the parameter through PROG (which may reset EPSILON), and before exiting restores EPSILON to its original value.
(PROG1 A B) - Evaluates A and B returning A.
(PROG3 FEXPR) - Like PROG2 this performs a call to PROG but saves the value of any global variable (quoted as the first argument in the PROG body) and then resets it before exit.
(QUAD A B C) - Is the quadratic formula returning the roots of the equation $Ax^2 + Bx + C$.
(RAC L) - Returns the last element in list L.
(RAD ANG) - Returns the radian equivalent of ANG.
(RDAC L) - Returns the (RAC(RDC L)).
(RDC L) - Returns the list L with the (RAC L) removed.
(RDDC L) - Returns (RDC(RDC L)).
(SIN A) - Returns the trigonometric sine of angle A.
(SITQ L) - Sets the evaluated CAR of L to the CADR of L and returns L (used by :=). This provides global assignments to E and Y variables.
(SLANG X1 Y1 X2 Y2) - Returns the angle of the line between (X1,Y1) and (X2,Y2) based on its slope.
(SLOPE L) - Returns the slope of angle L if L is a number. Otherwise L should be a list containing 4 numbers representing 2 cartesian coordinates. These are then used to determine the slope between these points.
(SNUM N) - Returns the sign (-1,1, or $\emptyset$) of N.
(TAN A) - Returns the trigonometric tangent of angle A.
(THIN N B) - Takes angle N and recursively subtracts (or adds if negative) until $\emptyset \leq N \leq B$. Used for thinning angles like ANG where $\emptyset > ANG > 36\emptyset$.
(UPSILON) - Raises the value of EPSILON times 10.
(WITHIN N1 N2 N3) - Returns T if $N1 \leq N2 \leq N3$ or $N1 \geq N2 \geq N3$; otherwise NIL.
(XDIS X A D) - Gives the X-coordinate found distance D away from X at angle A.
(XYDIS X Y A D) - Returns the list of calls from XDIS and YDIS.
(YDIS Y A D) - Gives the Y-coordinate found distance D away from Y at angle A.
(:= FEXPR) - Is a Hendrix System function redefined to set a global variable assignment through SITQ.

```
;@**********BOOLEAN**********
(DF GT (N1 N2) (*GREAT N1 N2))
(DF LT (N1 N2) (*GREAT N2 N1))
(DF GE (N1 N2) (OR (GT N1 N2) (EQUAL N1 N2)))
(DF LE (N1 N2) (OR (LT N1 N2) (EQUAL N1 N2)))
(DF WITHIN (N1 N2 N3) (OR (AND (GE N2 N1) (GE N3 N2))
                          (AND (GE N1 N2) (GE N2 N3))))
(DF ONEP (N) (EQUAL N 1))
(DF NE (A B) (NOT (EQUAL A B)))
;@**********EXPRS**********
(DF EDF (EDE) (EQ (CAR EDE) (CADR EDE)))
(DF := (X:=) (SITQ (LIST (CAR X:=) (EVAL (CADR X:=)))))
(DF SITQ (L) (PROGP (SET (CAR L) (CADR L)) L))
(SETQ BUCKET NIL)
(DF BUCKET (A) (SETQ BUCKET A))
(DF GTN (NTG) (ENTN 3GT (CAR NTG) (CDR NTG)))
(DF LTN (NTL) (ENTN 3LT (CAR NTL) (CDR NTL)))
(DF ENTN (E N L) (COND
    ((NULL L) N)
    ((EVAL (LIST E (CAR L) N)) (ENTN E (CAR L) (CDR L)))
    (T (ENTN E N (CDR L)))))
(DF EQALL (L LADE) (EEQAL (CAR LLADE) (CDR LLADE)))
(DF EEQAL (ITM LIZ) (COND
    ((NULL LIZ) NIL)
    ((EVAL (LIST 3EQUAL ITM (CAR LIZ))) T)
    (T (EEQAL ITM (CDR LIZ)))))
(DF PROG (OGRP) (EVAL (CONS 3PROG OGRP)))
(DF PROG1 (A B) A)
(DF PROGE (EGDEP) (EPROGE EGDEP))
(DF EPROGE (BOD) (EPROG3 3EPSILON BOD))
(DF PROG3 (GDEP3) (EPROG3 (CAR GDEP3) (EVAL (CADR GDEP3))))
(DF EPROG3 (VAR BOD) (PROG (VAV)
    (SETQ VAV (EVAL VAR)) (RETURN
    (PROG1 (EVAL (CONS 3PROG BOD)) (SET VAR VAV)))))
;@**********NUMERICAL**********
(EVAL (DECIMAL))
(DF COS (A) (PROGE (I P C S R)
    (SETQ R (THIN A 360))
    (RETURN (COND
    ((ZERO R) 1)
    ((EQUAL R 270 90) 0)
    ((AND (SETQ R (RAD (CONVERT R)))
          (SETQ P -P) (SETQ I R)
          (SETQ S -1) (SETQ C 0) NIL))
    ((ERRSET
        (REPEAT UNTIL (EQUAL I C)
            (SETQ I C) (SETQ S (NEG S)) (SETQ P (*PLUS P P))
            (SETQ C (*PLUS I (*TIMES S (QUO (POWER R P) (FAC P))))))
        NIL) C)
    (T (UPSILON) (COS A)))))
(DF SIN (A) (COS (*DIF A 90)))
(DF UPSILON () (SETQ EPSILON (*TIMES EPSILON 10)))
(DF CONVERT (A) (COND
    ((LE A 180) A)
    (T (*DIF 360 A))))
(DF RAD (ANG) (QUO (*TIMES ANG PI) 180))
(DF DEG (R) (*TIMES (QUO 180 PI) R))
```

```
(DE QUAD(A B C)(LIST
     (QUO(+PLUS(NEG B)(SQRT(+DIF(SQ B)
          (TIMES 4 A C))))(+TIMES 2 A))
     (QUO(+DIF(NEG B)(SQRT(+DIF(SQ B)
          (TIMES 4 A C))))(+TIMES 2 A))]
(SETQ PI 3.14159)
(SETQ E 2.71828)
(DE POWER(NUM RAISE)(COND
     ((ZEROP RAISE)1)
     ((LT RAISE 0)(QUO 1 (POWER NUM (NEG RAISE))))
     (T(+TIMES NUM(POWER NUM(SUB1 RAISE))))))
(DE FAC (NUM)(COND
     ((ZEROP NUM)1)
     ((LT NUM 0)(PRINT '*ERROR-NEGATIVE-FACTORIAL*))
     (T(+TIMES NUM(FAC(SUB1 NUM))))))
(DE SNUM(N)(COND
     ((EQUAL N 0)0)
     ((LT N 0) -1)
     (T 1))]
(DE NEG(N)(+TIMES -1 N))
(DE INV(N)(QUO 1 N))
(DE SLANG(FX FY TX TY)(PROG(SL X)
     (SETQ SL(SLOPE(LIST FX FY TX TY)))
     (SETQ X(DEP FX FY TX TY))
     (COND((LT SL -1)(SETQ X(NEG X))))
     (RETURN(COND((EQUAL 360(SETQ X(THIN(+PLUS
     (COND((NULL SL)(SETQ X(NEG X))-90)(T(INVTAN SL)))
     (COND((LT X 0)360)(T 180)))360)))0)(T X)))]
(DE DEP(FX FY TX TY)(COND
     ((EQUAL FX TX)(+DIF FY TY))
     (T(+DIF FX TX))]
(DE SLOPE(L)(COND
     ((ATOM L)(COND((EQUAL (THIN L 180)90)NIL)(T(TAN (THIN L 360)))))
     ((ZEROP(+DIF (CAR L)(CADR L)))NIL)
     (T(QUO(+DIF(CADR L)(CAR L))
          (+DIF(CAR L)(CADR L))))]
(DE DIS(X1 Y1 X2 Y2)(SQRT
     (+PLUS(SQ(+DIF X1 X2))(SQ(+DIF Y1 Y2)))))]
(DE XDIS(X A D)(+PLUS(+TIMES(COS A)D)X)]
(DE YDIS(Y A D)(+PLUS(+TIMES(SIN A)D)Y)]
(DE XYDIS(X Y A D)(LIST(XDIS X A D)(YDIS Y A D))]
(DE THIN(N B)(COND
     ((WITHIN B N 0)N)
     ((LT N 0)(THIN (+PLUS N B)B))
     (T(THIN(+DIF N B)B)))]
(DE INVCOS(A)(INVTAN(QUO(SQRT(+DIF 1(SQ A)))A)))
(DE INVSIN(A)(INVTAN(QUO A(SQRT(+DIF 1(SQ A))))))]
(DE TAN(A)(PROGE(NA)
     (SETQ NA(THIN A 180))
     (RETURN(COND
          ((EQUAL NA 90)0)
          ((ERRSET(SETQ NA(QUO(SIN NA)(COS NA)))NIL)NA)
          (T(UPSTLON)(TAN A)))))]
```

```
(DE INVTAN(A)(PROG(L P C S)(RETURN(COND
    ((GT(SQ A)1)(INV2 A))
    ((EQUAL(SQ A)1)(*TIMES A 45))
    ((AND(SETQ C A)(SETQ S 1)(SETQ P 1)(SETQ L 0)NIL))
    ((ERRSET
        (REPEAT UNTIL (EQUAL C L)
            (SETQ L C)(SETQ S(NEG S))(SETQ P(*PLUS P 2))
            (SETQ C(*PLUS(TIMES(POWER A P)(INV P)S)L)))
        NIL)(NEG C))
    (T(UPSTILON)(INVTAN A))))]
(DE INV2(A)(PROG(L P C S)
    (SETQ C(QUOT PI 2))(SETQ S 1)(SETQ P -1)
    (RETURN(COND((ERRSET
        (REPEAT UNTIL (EQUAL L C)
            (SETQ L C)(SETQ S(NEG S))(SETQ P(*PLUS P 2))
            (SETQ C(*PLUS(*TIMES(INV(*TIMES(POWER A P)P))S)L)))
        NIL)(NEG C))
    (T(UPSTILON)(INV2 A))))]
(DE E^X(X)(PROG(E D N)
    ((COND((GT(ABS X)1)(RETURN(E2X X))))
    (SETQ N 1)(SETQ E 0)
    (REPEAT UNTIL (EQUAL D N)
        (SETQ D N)(SETQ E (ADD1 E))
        (SETQ N (*PLUS D (QUOT (POWER X E)(FAC E)))))
    (RETURN N)))]
(DE E2X(X)(DUB(INT X)(FRAC X)))]
(DE DUB(I F)(*TIMES(POWER E I)(E^X F)))]
(DE INT(X)(COND
    ((LT X 0)(NEG(INT(NEG X))))
    ((GF X 0)(FIX X))
    ((EQUAL X(FIX X))X)
    (T(SUB1(FIX X)))))]
(DE FRAC(X)(COND
    ((LT X 0)(NEG(FRAC(NEG X))))
    ((EQUAL X 0.0)
    ((LT X 1)X)
    (T(FRAC(SUB1 X)))))]
$**********LIST**********
(DE SNOC(A L)(APPEND L(LIST A)))
(DE RDOC(L)(DOC(RDC L)))]
(DE FAC(L)(CAR(LAST L)))]
(DE RDC(L)(COND
    ((NULL L)(PRINT *NIL-RDC*))
    ((NULL(CDR L))NIL)
    (T(CONS(CAR L)(RDC(CDR L))))))]
(DE RDAC(L)(FAC(RDC L)))]
```

REFERENCES

1. Hendrix, Gary g., "Modeling Simultaneous Actions and Continuous Processes," Artificial Intelligence An International Journal,vol. 4, no. 3, 4, North-Holland Publishing Company, Amsterdam, Winter 1973.

2. Lowrance, John D.; Friedman, Daniel P., "The Hendrix Model of Simultaneous Actions and Continuous Processes: An Introduction and Implementation Description," Technical Report No. 33, Computer Science Department, Indiana University, Bloomington, Indiana, June 1975.

3. McCarthy, J. ; Abrahams, P. W.; Edwards, D. J.; Hart, T. P.; and Levin, M. I., LISP 1.5 Programmer's Manual, MIT PRESS, Cambridge, Mass., 1962.

4. DECsystem-10 Users Handbook, Digital Equipment Corporation, Maynard, Mass.

5. Quam, Lynn H., Stanford LISP 1.6 Manual Stanford Artificial Intelligence Project, Stanford University, September, 1969.

6. Griswold, R. E.; Poage, J. F.; and Polonsky, I. P., The SNOBOL4 Programming Language, Prentice-Hall, Inc., Englewood, New Jersey, 1971.

7. Baumgart, Bruce G., Micro Planner Alternate Reference Manual Stanford Artificial Intelligence Laboratory, Operating Note Number 67, April, 1972.

8. Sussman, Gerald Jay; Winograd, Terry; Charniak, Eugene, Micro-Planner Reference Manual, Memo No. 203A, Massachusetts Institute of Technology, December, 1971.

9. Albus, James S.; Evans, John M. Jr., "Robot Systems", Scientific American Vol. 243, No. 2, pp. 77-86b(inc), February, 1976.