

Report 81-26
Stanford -- KSL

Scientific DataLink

BOWL: A Beginner's Program Using AGE .
AGE Example Series: Number 1.
Nelleke Aiello, H. Penny Nii,
Jun 1981

card 1 of 1

HPP-81-26

BOWL: A Beginner's Program Using AGE

AGE Example Series: Number 1

Nelleke Aiello

H. Penny Nii

Heuristic Programming Project
Computer Science Department
Stanford University
Stanford, Ca 94305

Table of Contents

1. Introduction	1
2. The Task	3
3. Problem Formulation	5
4. User Program	9
4.1. Loading the Program	9
4.2. Hypothesis Structure	10
4.3. Knowledge Sources	10
4.4. Control Settings	24
4.5. User Provided Functions	25
5. Program Run	27
6. Program Run with a Trace	31

List of Figures

Figure 3-1: Hypothesis Structure	5
Figure 3-2: Organization of the KSs	6

1. Introduction

The two AGE documents, *Joy of AGE-ing* and *AGE Reference Manual*, do not give the readers a sense of how to go about formulating a program to be built using AGE. This document is intended to fill that gap by presenting a complete program--its formulation, its construction and code, and its runs. It is a very simple program and uses only a small subset of the features in AGE. However, many problems can be formulated in similar ways, and it can be used as a template for the beginning AGE users. For most effective use of this document, it should be read in conjunction with the other two AGE documents--specific references are made when appropriate.

The example is a program that keeps track of bowling scores. It violates one of the rules of good knowledge engineering: *Use an algorithm to solve a problem when one exists*. However, the example is simple, is in a domain familiar to everyone, and has only a dozen or so rules. At the same time, the problem has been formulated to show several of the AGE features -- multiple knowledge sources, interactive input stream, multiple-hit and single-hit rule execution strategies, multiple preconditions, use of links, use of prepackaged control structure, non-standard termination, etc. These will be pointed out in the rest of the document.

2. The Task

Keep an up-to-date score for a bowler. At the end of the game, type the final score and the frames in which strikes and spares weremade.

For Those Not Familiar with Bowling.....

There are ten frames in a game of bowling. (The word "frame" is a bowling term--not to be confused with the AI term.) A player gets to throw two balls in a frame, unless he knocks down all the pins in his first throw. If it takes a player only one ball to knock down all ten pins in a frame (a *strike*), his score for that frame is 10 plus the number of pins he gets on his next two balls (in the next frame or frames). If it takes him two balls to knock down all ten pins in a frame (a *spare*), his score for that frame is 10 plus the number of pins he gets on his next ball (in the next frame). If he does not knock down ten pins in a frame (an *open* frame), his score for that frame is just the number of pins he knocked down in that frame.

3. Problem Formulation

One level data structure: the level name is FRAME. Refer to Figure 3-1. Each frame is connected to the previous frame by a link named LASTFRAME (NEXTFRAME is also specified to point to the next frame, but is never used.) Each frame has attributes FRAME#, SCORE1, CUMSCORE, STRIKE, and SPARE. Refer to Section 4.1 of Joy of AGE-ing for description on how to specify the hypothesis level and the elements. Refer to Section 4.2 for the AGE hypothesis structure.

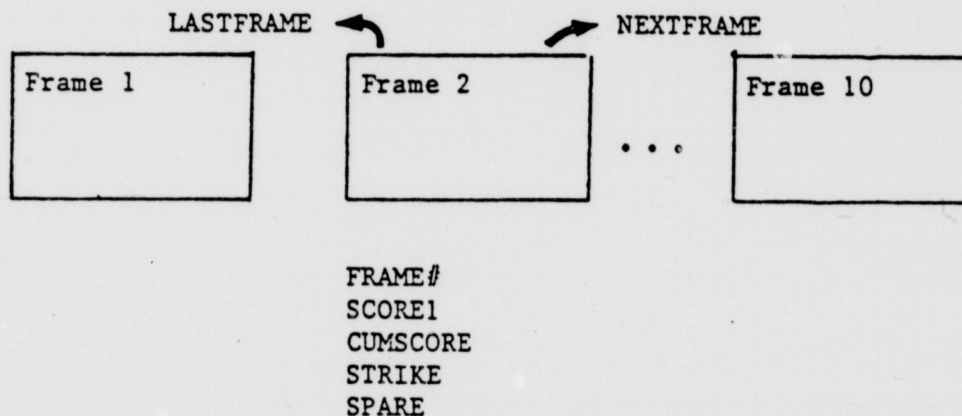


Figure 3-1: Hypothesis Structure

Ask for the number of pins struck with each throw of the ball and process the score with each ball. Use a function, PROMPTLIST, provided by AGE to ask for the number of pins struck. (Refer to Sec. 5.4 of the Reference Manual for its syntax.)

Use four knowledge sources, FIRSTFRAME, FIRST-BALL, SECOND-BALL, and FINALFRAME (see Figure 3-2). (Refer to Section 4.3 for the complete listing of the rules.)

- **FIRSTFRAME:** Process the very first ball thrown. It can either be a strike (10 pins) or a miss (less than 10 pins). The event-type from this KS is either STRIKE or FIRST-BALL.
- **FIRST-BALL:** Process the first ball of each of the rest of the frames. Compute the cumulative score for relevant previous frames. The scoring depends on whether there was a spare in the previous frame, or a strike followed by another strike in the two previous frames. Note that if there was a strike in the last frame, but not in the one before that, no scoring is done. Example: Rule 1 is a case in which the player had a miss with the current ball, but had a spare in the last frame.

This KS also generates new frames and stores their frame numbers (FRAME#), saves the numbers of pins knocked down (SCORE1), and links frames--(remember LASTFRAME is the name of a link). Refer to Sec. 2.3 of the Reference Manual for how to link nodes. The event type generated is STRIKE, FIRST-BALL, or LASTFRAME (see Rule 6) depending on the number of

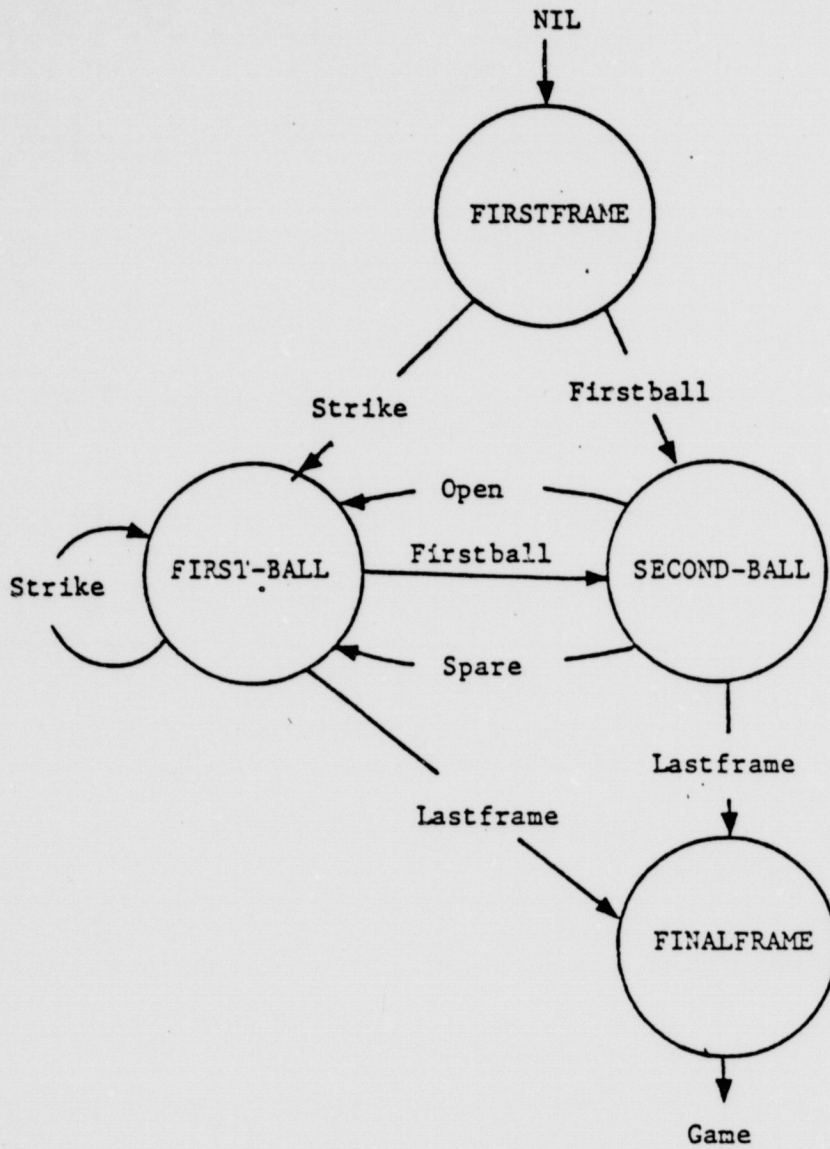


Figure 3-2: Organization of the KSs

pins knocked down and on which frame the game is in.

- **SECOND-BALL:** Process the second ball thrown. Compute the score for the previous frame, if it was a strike. For example, Rule 1 is a case in which the player had a strike in the last frame and has a miss in this frame--the scores for this frame and the last frame must be computed. The event type possible from this KS is SPARE or OPEN. In addition, an event type LASTFRAME is generated if the player is in the final frame (frame 10).

- **FINALFRAME:** There are three possible ways in which the 10th frame can be played: if the first ball is a strike, the bowler gets two other throws (Rule 2); if the first two balls end in a spare, he gets another throw (Rule 1); otherwise, he only gets a total of two balls for the frame (which have already been played). This KS is used only to determine how many extra balls the player gets--none, one or two. The only event type generated is GAME--it signals the end of the game.

Use control macro package with modifications to the Termination and Postprocessing components. Refer to Section 4.4 for the control settings. This is the simplest event-driven format, in which the latest change made to the element is the one picked for focus the next time around (SELECTION METHOD = LIFO). Note that the dummy event type, SCORING, is never the latest change. However, even if it were, since there is no precondition that matches that event type, it will be bypassed and some other event type will be chosen.

The termination function, TERMINATEWHEN, waits for the event type GAME to appear in the EVENTLIST. The postprocessing function, PROCESSANSWER, gets the information specified in the task description from the frames and prints them.

Additional features used:

- **preconditions:** FIRSTFRAME has no precondition, since it is never used after its invocation as the first KS. FIRST-BALL KS has multiple preconditions, appearance of any one as a focus event will activate the KS.
- **hit strategies:** All KSs, except SECOND-BALL, uses Single-hit rule scanning strategy. Multiple-hit is used in SECOND-BALL to perform an independent function, that of looking out for the final frame.
- **multiple actions:** Each action on the RHSs can change the values of many attributes, as long as the changes are made to the same hypothesis element. Since scoring can be delayed for one or two balls, there has to be multiple actions, one for each frame (hypothesis element). The worst cases are RULE 2 and RULE 4 in FIRST-BALL KS in which attributes in three frames are changed.
- **null action:** RULE 3 in FINALFRAME only generates an event type.
- **ordered/unordered rules:** The rules in all the KSs, except FIRSTFRAME, must be evaluated in the order presented. They can be made unordered by adding more specific conditions in the LHSs.
- **interactive program:** Most programs are set up to get input data in batch mode by specifying a priori the INPUTDATA names (see examples in the appendix of Joy of AGE-ing). This problem is set up with the data coming in one at a time. This was done both to demonstrate that an interactive program can be built very simply, and that some time-oriented data can be handled in a similar way.
- **Dynamic element generation and FOCUSEVENT:** In many problems the elements of the

hypothesis are a priori known. In such cases, the hypothesis element names can be specified during hypothesis-structure specification, and the node names can be used explicitly in the rules. In other problems, this is not possible. When elements are dynamically generated, the only way to access that element is through the use of focussing mechanisms, or by following links.

FOCUSEVENT contains the event that was selected in the kernel--in this problem, the latest event that was generated by the RHS of some rule. Note that the FOCUSEVENT node selected when FIRST-BALL KS is invoked is the previous frame; in the SECOND-BALL, it is the frame the player is currently in. The variable, PREV-PREVIOUSFRAME, in FIRST-BALL KS is set by following a link to the previous frame by using an AGE function, \$VALUE. (Refer to Sec. 3.1 for FOCUSEVENT and Sec. 5.2 for \$VALUE descriptions in the Reference Manual.)

- others: Note the use of SETQs in the LHSs of rules in FINALFRAME. This is one method of passing arguments to the RHS, although not elegant. Note the use of \$FIND in the function PROCESSANSWER. This function can also be used in the rules to find element(s) with specific attribute-value pairs (see Sec. 5.2 of the Reference Manual).

Finally, an example program run with a few trace options is included (see Section 6). There are two things that do not appear in the run sections that are useful for debugging: (1) break package that allows the program to pause at various points in the user program (not to be confused with the Interlisp Break package) and (2) single-cycle mode in which the program pauses after each "cycle". A cycle in AGE is defined as a logical step in the program execution--e.g. execution of the LHS, execution of the KHS, selection of a step, etc. (Refer to the Reference Manual for AGE debugging aids.)

4. User Program

4.1. Loading the Program

<CSD.AGE>XAGE.EXE created 16-Jul-80 14:29:40

Are you already familiar with this system? Y/N : y

What do you want to do? : get
Get the user program from a file.

What is the name of your system? : bowl
Select the files to be loaded.

Type Y/N/<name>/A/Q

-HYPOTHESIS : a

Remaining names selected:

-HYPOTHESIS -KNOWLEDGESOURCES -USERFNS -CONTROLINFO -DESIGN

FILE CREATED 4-Jun-80 09:04:34

BOWL-HYPOTHESISCOMS

(NODELEVELS reset)

(DEFINEDSLOTS reset)

(DEFINEDLINKS reset)

(LINKS&OPPOSITES reset)

FILE CREATED 1-Jul-80 17:19:07

BOWL-KNOWLEDGESOURCESCOMS

(KSLIST reset)

(#LHSEVALUATOR reset)

(#VALUEADJUSTER reset)

(RULENUMB reset)

(USERPOSTPROCESSINGFN reset)

FILE CREATED 1-Jul-80 17:36:13

BOWL-USERFNSCOMS

(USERFNS reset)

FILE CREATED 1-Jul-80 17:19:21

BOWL-CONTROLINFOCOMS

(USERINITFN reset)

(TERMINATIONCONDITION reset)

(EVENT reset)

(CONTROLRULES reset)

FILE CREATED 1-Jul-80 17:36:17

BOWL-DESIGNCOMS

End of GETSYS

4.2. Hypothesis Structure

What do you want to do? : ex
Examine parts of the user system.

What would you like to see?
Type H/K/C/U/DA/DO : h

.....
LEVEL NODE OF LEVEL 1
FRAME -
FRAME# - NIL
SCORE1 - NIL
CUMSCORE - NIL
STRIKE - NIL
SPARE - NIL

.....
.....

4.3. Knowledge Sources

What would you like to see?
Type H/K/C/U/DA/DO : k

Which knowledge source would you like printed?
Type FIN/S/FIRST-/FIRSTF/A : firstf
Bad response. Type ? for help.

Which knowledge source would you like printed?
Type FIN/S/FIRST-/FIRSTF/A : firstf

-->Knowledge Source: FIRSTFRAME

Precondition: NIL
Inference levels (from . to): (FRAME . FRAME)
Links between levels: NIL
Hit Strategy: (SINGLE)
Local variable bindings: (BALL (CAR (PROMPTLIST (* First ball) (0 1 2 3 4 5 6 7 8 9 10))))

.....
l
if

```
(LESSP BALL 10)
then
```

```
PROPOSE
```

```
ch.type - ADD
level - FRAME
attr-value - ((CUMSCORE 0)
              (SCORE1 BALL)
              (FRAME# (SETQ FN 1)))
link-node - NIL
support - FIRSTFRAME
ev.type - FIRST-BALL
comment - (The first ball less than 10, remember the no.)
```

```
2
```

```
if
(EQ BALL 10)
then
```

```
PROPOSE
```

```
ch.type - ADD
level - FRAME
attr-value - ((CUMSCORE 10)
              (STRIKE T)
              (FRAME# (SETQ FN 1)))
link-node - NIL
support - FIRSTFRAME
ev.type - STRIKE
comment - (Strike on the first ball)
```

What would you like to see?
Type H/K/C/U/DA/DO : k

Which knowledge source would you like printed?
Type FIN/S/FIRST-/FIRSTF/A : first-

-->Knowledge Source: FIRST-BALL

Precondition: (STRIKE OPEN SPARE)
Inference levels (from . to): (FRAME . FRAME)
Links between levels: (LASTFRAME . NEXTFRAME)
Hit Strategy: (SINGLE)
Local variable bindings: (BALL (CAR (PROMPTLIST (* First ball) (0 1 2 3 4 5 6 7 8 9 10))))
(PREVIOUSFRAME FOCUSEVENT:NODE)
(PREV-PREVIOUSFRAME
(\$VALUE PREVIOUSFRAME LASTFRAME LATEST))

1
if
 (LESSP BALL 10)
 (\$VALUE PREVIOUSFRAME SPARE LATEST)
then

 PROPOSE

 ch.type - SUPERSEDE
 hypo-element - PREVIOUSFRAME
 attr-value - (CUMSCORE (IPLUS BALL 10
 (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - FIRST-BALL
 ev.type - SCORING
 comment - (Score)

 PROPOSE

 ch.type - ADD
 level - FRAME
 attr-value - ((SCORE1 BALL)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - (LASTFRAME PREVIOUSFRAME)
 support - FIRST-BALL
 ev.type - FIRST-BALL
 comment - (First ball less than 10)

2

```

if
(LESSP BALL 10)
($VALUE PREVIOUSFRAME STRIKE LATEST)
PREV-PREVIOUSFRAME
($VALUE PREV-PREVIOUSFRAME STRIKE LATEST)
then

```

PROPOSE

```

ch.type - SUPERSEDE
hypo-element - PREV-PREVIOUSFRAME
attr-value - (CUMSCORE (IPLUS ($VALUE PREV-PREVIOUSFRAME
                          CUMSCORE LATEST) BALL 20))
link-node - NIL
support - FIRST-BALL
ev.type - SCORING
comment - Scoring

```

PROPOSE

```

ch.type - SUPERSEDE
hypo-element - PREVIOUSFRAME
attr-value - (CUMSCORE ($VALUE PREV-PREVIOUSFRAME CUMSCORE LATEST))
link-node - NIL
support - FIRST-BALL
ev.type - SCORING
comment - Scoring

```

PROPOSE

```

ch.type - ADD
level - FRAME
attr-value - ((SCORE1 BALL)
              (FRAME# (SETQ FN (ADD1 FN)))
              (CUMSCORE ($VALUE PREVIOUSFRAME CUMSCORE LATEST)))
link-node - (LASTFRAME PREVIOUSFRAME)
support - FIRST-BALL
ev.type - FIRST-BALL
comment - (Two strikes in a row in the previous frames)

```

3

```

if
(LESSP BALL 10)
then

```

PROPOSE

```

ch.type - ADD
level - FRAME
attr-value - ((SCORE1 BALL)

```

```

      (FRAME# (SETQ FN (ADD1 FN)))
      (CUMSCORE ($VALUE PREVIOUSFRAME CUMSCORE LATEST)))
link-node - (LASTFRAME PREVIOUSFRAME)
support -   FIRST-BALL
ev.type -   FIRST-BALL
comment -   (Open in previous frame)

```

4

```

if
(EQUAL ($VALUE FOCUSEVENT:NODE FRAME LATEST) 10)
(EQ BALL 10)
($VALUE PREVIOUSFRAME STRIKE LATEST)
PREV-PREVIOUSFRAME
($VALUE PREV-PREVIOUSFRAME STRIKE LATEST)
then

```

PROPOSE

```

ch.type -   SUPERSEDE
hypo-element - PREV-PREVIOUSFRAME
attr-value - (CUMSCORE (IPLUS 30
                      ($VALUE PREV-PREVIOUSFRAME CUMSCORE LATEST)))
link-node - NIL
support -   FIRST-BALL
ev.type -   SCORING
comment -   Scoring

```

PROPOSE

```

ch.type -   SUPERSEDE
hypo-element - PREVIOUSFRAME
attr-value - (CUMSCORE ($VALUE PREV-PREVIOUSFRAME CUMSCORE LATEST))
link-node - NIL
support -   FIRST-BALL
ev.type -   SCORING
comment -   Scoring

```

PROPOSE

```

ch.type -   ADD
level -     FRAME
attr-value - ((SCORE1 10)
              (STRIKE T)
              (FRAME# (SETQ FN (ADD1 FN)))
              (CUMSCORE ($VALUE PREVIOUSFRAME CUMSCORE LATEST)))
link-node - (LASTFRAME PREVIOUSFRAME)
support -   FIRST-BALL
ev.type -   STRIKE
comment -   (Strike with two previous frames strike)

```

PROPOSE

ch.type - NULL
 hypo-element - FOCUSEVENT:NODE
 support - FIRST-BALL
 ev.type - LASTFRAME
 comment - (STRIKE in last frame.)

5

if

(EQ BALL 10)
 (\$VALUE PREVIOUSFRAME SPARE LATEST)
 (EQUAL (\$VALUE FOCUSEVENT:NODE FRAME LATEST) 10)

then

PROPOSE

ch.type - SUPERSEDE
 hypo-element - PREVIOUSFRAME
 attr-value - (CUMSCORE (IPLUS 20
 (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - FIRST-BALL
 ev.type - SCORING
 comment - Scoring

PROPOSE

ch.type - ADD
 level - FRAME
 attr-value - ((STRIKE T)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - (LASTFRAME PREVIOUSFRAME)
 support - FIRST-BALL
 ev.type - STRIKE
 comment - (Strike with previous frame spare)

PROPOSE

ch.type - NULL
 hypo-element - FOCUSEVENT:NODE
 support - FIRST-BALL
 ev.type - LASTFRAME
 comment - (Strike in last frame.)

6

if

(EQ BALL 10)
 (EQUAL (\$VALUE FOCUSEVENT:NODE FRAME LATEST) 10)

then

PROPOSE

ch.type - ADD
 level - FRAME
 attr-value - ((STRIKE T)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - (LASTFRAME PREVIOUSFRAME)
 support - FIRST-BALL
 ev.type - STRIKE
 comment - (Open on previous frame with strike)

PROPOSE

ch.type - NULL
 hypo-element - FOCUSEVENT:NODE
 support - FIRST-BALL
 ev.type - LASTFRAME
 comment - (Strike in last frame.)

7

if

(EQ BALL 10)
 (\$VALUE PREVIOUSFRAME STRIKE LATEST)
 PREV-PREVIOUSFRAME
 (\$VALUE PREV-PREVIOUSFRAME STRIKE LATEST)

then

PROPOSE

ch.type - SUPERSEDE
 hypo-element - PREV-PREVIOUSFRAME
 attr-value - (CUMSCORE (IPLUS 30
 (\$VALUE PREV-PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - FIRST-BALL
 ev.type - SCORING
 comment - Scoring

PROPOSE

ch.type - SUPERSEDE
 hypo-element - PREVIOUSFRAME
 attr-value - (CUMSCORE (\$VALUE PREV-PREVIOUSFRAME CUMSCORE LATEST))
 link-node - NIL
 support - FIRST-BALL
 ev.type - SCORING
 comment - SCORING

PROPOSE

ch.type - ADD
 level - FRAME
 attr-value - ((SCORE1 10)
 (STRIKE T)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - (LASTFRAME PREVIOUSFRAME)
 support - FIRST-BALL
 ev.type - STRIKE
 comment - (Strike with two previous frames strike)

8

if
 (EQ BALL 10)
 (\$VALUE PREVIOUSFRAME SPARE LATEST)
 then

PROPOSE

ch.type - SUPERSEDE
 hypo-clement - PREVIOUSFRAME
 attr-value - (CUMSCORE (IPLUS 20
 (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - FIRST-BALL
 ev.type - SCORING
 comment - Scoring

PROPOSE

ch.type - ADD
 level - FRAME
 attr-value - ((STRIKE T)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - (LASTFRAME PREVIOUSFRAME)
 support - FIRST-BALL
 ev.type - STRIKE
 comment - (Strike with previous frame spare)

9

if
 (EQ BALL 10)
 then

PROPOSE

ch.type - ADD
level - FRAME
attr-value - ((STRIKE T)
 (FRAME# (SETQ FN (ADD1 FN)))
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
link-node - (LASTFRAME PREVIOUSFRAME)
support - FIRST-BALL
ev.type - STRIKE
comment - (Open on previous frame with strike)

What would you like to see?
Type H/K/C/U/DA/DO : k

Which knowledge source would you like printed?
Type FIN/S/FIRST-/FIRSTF/A : se

-->Knowledge Source: SECOND-BALL

Precondition: (FIRST-BALL)
Inference levels (from . to): (FRAME . FRAME)
Links between levels: NIL
Hit Strategy: (MULTIPLE)
Local variable bindings: (SCORE1 (\$VALUE (fetch NODE of FOCUSEVENT)
SCORE1 LATEST))
(BALL (CAR (PROMPTLIST (* Second ball) NIL NIL
(FUNCTION (LAMBDA (X)
(PINSLEFT SCORE1 X))))))
(PREVIOUSFRAME (\$VALUE (fetch NODE of FOCUSEVENT)
LASTFRAME LATEST))

1
if
(\$VALUE PREVIOUSFRAME STRIKE LATEST)
(LESSP (IPLUS SCORE1 BALL) 10)
then

PROPOSE

ch.type - SUPERSEDE
hypo-element - PREVIOUSFRAME
attr-value - (CUMSCORE (IPLUS SCORE1 BALL
(\$VALUE PREVIOUSFRAME CUMSCORE LATEST) 10))
link-node - NIL
support - SECOND-BALL
ev.type - SCORING
comment - Scoring

PROPOSE

ch.type - SUPERSEDE
hypo-element - FOCUSEVENT:NODE
attr-value - (CUMSCORE (IPLUS SCORE1 BALL
(\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
link-node - NIL
support - SECOND-BALL
ev.type - OPEN
comment - (Previous frame strike with open in this frame)

2

if
 (LESSP (IPLUS SCORE1 BALL) 10)
 then

PROPOSE

ch.type - SUPERSEDE
 hypo-element - FOCUSEVENT:NODE
 attr-value - (CUMSCORE (IPLUS SCORE1 BALL
 (OR (\$VALUE PREVIOUSFRAME CUMSCORE LATEST) 0)))
 link-node - NIL
 support - SECOND-BALL
 ev.type - OPEN
 comment - Open

3

if
 (EQ (IPLUS SCORE1 BALL) 10)
 (\$VALUE PREVIOUSFRAME STRIKE LATEST)
 then

PROPOSE

ch.type - SUPERSEDE
 hypo-element - PREVIOUSFRAME
 attr-value - (CUMSCORE (IPLUS 20
 (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - SECOND-BALL
 ev.type - SPARE
 comment - Scoring

PROPOSE

ch.type - SUPERSEDE
 hypo-element - FOCUSEVENT:NODE
 attr-value - ((SPARE T)
 (CUMSCORE (\$VALUE PREVIOUSFRAME CUMSCORE LATEST)))
 link-node - NIL
 support - SECOND-BALL
 ev.type - SPARE
 comment - (Previous frame strike, spare on this frame)

4

if
 (FQ (IPLUS SCORE1 BALL) 10)
 then

PROPOSE

ch.type - SUPERSEDE
hypo-element - FOCUSEVENT:NODE
attr-value - (SPARE T)
link-node - NIL
support - SECOND-BALL
ev.type - SPARE
comment - Spare

5

if

(EQUAL (\$VALUE (fetch NODE of FOCUSEVENT) FRAME# LATEST) 10)

then

PROPOSE

ch.type - NULL
hypo-element - FOCUSEVENT:NODE
support - SECOND-BALL
ev.type - LASTFRAME
comment - (LAST FRAME)

What would you like to see?
Type H/K/C/U/DA/DO : k

Which knowledge source would you like printed?
Type FIN/S/FIRST-/FIRSTF/A : fin

-->Knowledge Source: FINALFRAME

Precondition: (LASTFRAME)
Inference levels (from . to): (FRAME . FRAME)
Links between levels: NIL
Hit Strategy: (SINGLE)
Local variable bindings: NIL

1
if
(\$VALUE (fetch NODE of FOCUSEVENT) SPARE LATEST)
(SETQ BALL (CAR (PROMPTLIST (* Third ball of final frame.) (0 1 2 3 4 5 6 7 8 9 10))))
then

PROPOSE

ch.type - SUPERSEDE
hypo-element - FOCUSEVENT:NODE
attr-value - (CUMSCORE (IPLUS BALL (\$VALUE (fetch NODE of FOCUSEVENT)
CUMSCORE LATEST) 10))
link-node - NIL
support - FINALFRAME
ev.type - GAME
comment - (Spare in the last frame)

2
if
(\$VALUE FOCUSEVENT:NODE STRIKE LATEST)
(SETQ BALL1 (PROMPTLIST (* Second ball of final frame) (0 1 2 3 4 5 6 7 8 9 10)))
(SETQ BALL2 (CAR (PROMPTLIST (* Third ball of final frame) NIL NIL
(FUNCTION (LAMBDA (X)
(OR (AND (EQ BALL1 10)
(PINSLEFT 0 X))
(PINSLEFT BALL1 X)))))))
then

PROPOSE

ch.type - SUPERSEDE

hypo-element - FOCUSEVENT:NODE
attr-value - (CUMSCORE (IPLUS 10 BALL1 BALL2
(\$VALUE FOCUSEVENT:NODE CUMSCORE LATEST)))
link-node - NIL
support - FINALFRAME
ev.type - GAME
comment - (Strike in the last frame)

3

if
T
then

PROPOSE

ch.type - NULL
hypo-element - NIL
support - FINALFRAME
ev.type - GAME
comment - (Open in the last frame)

4.4. Control Settings

What would you like to see?

Type H/K/C/U/DA/DO : c

INITIALIZATION FUNCTION = \$DEFAULTINIT

TERMINATION FUNCTION = TERMINATEWHEN

EVENT CONTROL INFORMATION:

SELECTION METHOD = LIFO

COLLECTIONRULES = NIL

EXPECTATION CONTROL INFORMATION:

SELECTION METHOD = NIL

MATCHER = NIL

GOAL CONTROL INFORMATION:

SELECTION METHOD = NIL

MATCHER = NIL

SEEKING METHOD = NIL

POSTPROCESSING FUNCTION = (PROCESSANSWER)

What would you like to see?

Type H/K/C/U/DA/DO : do

End of EXAMINE

4.5. User Provided Functions

```
($DEFAULTINIT
 [LAMBDA NIL
  (QUOTE FIRSTFRAME))
```

```
(PINSLEFT
 [LAMBDA (B1 B2) (* edited: " 1-Jul-80 17:32")
  (AND (SMALLP B2)
   (ILEQ B2 (IDIFFERENCE 10 B1))
   (IGEQ B2 0)
   B2))
```

```
(TERMINATEWHEN
 [LAMBDA NIL (* edited: " 6-Jun-80 07:39")
  (EQUAL (QUOTE GAME)
   (fetch TYPE of (CAR EVENTLIST))
```

```
(PROCESSANSWER
 (LAMBDA NIL (* edited: " 6-Jun-80 07:41")
  (LASTFRAME -(CAR ($FIND '(FRAME# 10))))
  (PRIN1 "-----")
  (TERPRI)
  (PRIN1 "SCORE: ")
  (PRINT ($VALUE LASTFRAME CUMSCORE LATEST))
  (PRIN1 "STRIKES IN FRAMES: ")
  (PRINT ($FIND '(STRIKE T)))
  (PRIN1 "SPARES IN FRAMES: ")
  (PRINT ($FIND '(SPARE T)))
  (PRIN1 "-----")
  (TERPRI)
  (PRHYPO))))]
```


5. Program Run

What do you want to do? : run
Execute the user program.

Trace menu: Step/Focus/ADdtolist/Ks/RUlnumber/Lhs/RHs/Nodechanges/
Collection/ALI/Quit
Type ? for help

Trace mode?
Type A/R/E/Q : q

Choose your breaks thusly: A(dd) | R(emove) B(efore) | A(fter)
L(hs) | R(hs) | K(s)
< rulenumbr > | < ksname > | A(ll)

Type ? for help
Choose break>qUIT

First ball : 3

Second ball : 4

First ball : 6

Second ball : 4

First ball : 10

First ball : 10

First ball : 9

Second ball : 1

First ball : 3

Second ball : 4

First ball : 5

Second ball : 5

First ball : 3

Second ball : 7

First ball : 4

Second ball : 5

First ball : 5

Second ball : 2

SCORE: 139

STRIKES IN FRAMES: (FRAME3 FRAME4)

SPARES IN FRAMES: (FRAME2 FRAME5 FRAME7 FRAME8)

.....
LEVEL NODE OF LEVEL 1

FRAME -

FRAME# - NIL

SCORE1 - NIL

CUMSCORE - NIL

STRIKE - NIL

SPARE - NIL

INST - (FRAME10 FRAME9 FRAME8 FRAME7 FRAME6 FRAME5 FRAME4
FRAME3 FRAME2 FRAME1)

.....
THE FOLLOWING ARE THE OTHER NODES (AND THEIR PROPERTY LISTS)
ON LEVEL 1

.....
FRAME1 -

PROTO - (FRAME)

FRAME# - (1)

SCORE1 - (3)

CUMSCORE - (7)

STRIKE - NIL

SPARE - NIL

NEXTFRAME - (FRAME2)

.....
FRAME2 -

PROTO - (FRAME)

FRAME# - (2)

SCORE1 - (6)

CUMSCORE - (27)

STRIKE - NIL

SPARE - (T)

LASTFRAME - (FRAME1)

NEXTFRAME - (FRAME3)

.....
FRAME3 -

PROTO - (FRAME)

FRAME# - (3)

SCORE1 - NIL

CUMSCORE - (56)

STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME2)
NEXTFRAME - (FRAME4)

.....
FRAME4 -
PROTO - (FRAME)
FRAME# - (4)
SCORE1 - NIL
CUMSCORE - (76)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME3)
NEXTFRAME - (FRAME5)

.....
FRAME5 -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (89)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)
NEXTFRAME - (FRAME6)

.....
FRAME6 -
PROTO - (FRAME)
FRAME# - (6)
SCORE1 - (3)
CUMSCORE - (96)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME5)
NEXTFRAME - (FRAME7)

.....
FRAME7 -
PROTO - (FRAME)
FRAME# - (7)
SCORE1 - (5)
CUMSCORE - (109)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME6)
NEXTFRAME - (FRAME8)

.....
FRAME8 -
PROTO - (FRAME)
FRAME# - (8)
SCORE1 - (3)
CUMSCORE - (123)

STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME7)
NEXTFRAME - (FRAME9)

.....
FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (4)
CUMSCORE - (132)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME8)
NEXTFRAME - (FRAME10)

.....
FRAME10 -
PROTO - (FRAME)
FRAME# - (10)
SCORE1 - (5)
CUMSCORE - (139)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME9)
.....End of RUN

6. Program Run with a Trace

What do you want to do? : reset

Reset the user system.

End of RESET

What do you want to do? : run

Execute the user program.

Trace menu: Step/Focus/ADdtolist/Ks/RULnumber/Lhs/RHs/Nodechanges/
Collection/ALI/Quit

Type ? for help

Trace mode?

Type A/R/E/Q : a

Trace>fOCUSPRINTFLAG

Trace>kSPRINTFLAG

Trace>ruLENUMBERPRINTFLAG

Trace>nODECHANGESPRINTFLAG

Trace>qUIT

Trace mode?

Type A/R/E/Q : q

Choose your breaks thusly: A(dd) | R(emove) B(efore) | A(fter)

L(hs) | R(hs) | K(s)

< rulenumbe r > | < ksname > | A(ll)

Type ? for help

Choose break>qUIT

Knowledge source invoked - FIRSTFRAME

First ball : 3

Number of applicable rule: 1

New node created on level FRAME:

FRAME1 -

PROTO - (FRAME)

FRAME# - (1)

SCORE1 - (3)

CUMSCORE - (0)

STRIKE - NIL

SPARE - NIL

Focus event - (FIRST-BALL FRAME1 ((CUMSCORE 0) (SCORE1 3)

(FRAME# 1)) FIRSTFRAME EV1 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 6

Number of applicable rule: 2

Node before application of rule:

FRAME1 -
 PROTO - (FRAME)
 FRAME# - (1)
 SCORE1 - (3)
 CUMSCORE - (0)
 STRIKE - NIL
 SPARE - NIL

Node after application of rule:

FRAME1 -
 PROTO - (FRAME)
 FRAME# - (1)
 SCORE1 - (3)
 CUMSCORE - (9)
 STRIKE - NIL
 SPARE - NIL

Focus event - (OPEN FRAME1 ((CUMSCORE 9)) SECOND-BALL EV2 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 4

Number of applicable rule: 3

New node created on level FRAME:

FRAME2 -
 PROTO - (FRAME)
 FRAME# - (2)
 SCORE1 - (4)
 CUMSCORE - (9)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME1)

Focus event - (FIRST-BALL FRAME2 ((LASTFRAME FRAME1) (SCORE1 4)
 (FRAME# 2) (CUMSCORE 9)) FIRST-BALL EV3 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 6

Number of applicable rule: 4

Node before application of rule:

FRAME2 -
PROTO - (FRAME)
FRAME# - (2)
SCORE1 - (4)
CUMSCORE - (9)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME1)

Node after application of rule:

FRAME2 -
PROTO - (FRAME)
FRAME# - (2)
SCORE1 - (4)
CUMSCORE - (9)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME1)

Focus event - (SPARE FRAME2 ((SPARE T)) SECOND-BALL EV4 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 5

Number of applicable rule: 1

Node before application of rule:

FRAME2 -
PROTO - (FRAME)
FRAME# - (2)
SCORE1 - (4)
CUMSCORE - (9)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME1)

Node after application of rule:

FRAME2 -
PROTO - (FRAME)

FRAME# - (2)
 SCORE1 - (4)
 CUMSCORE - (24)
 STRIKE - NIL
 SPARE - (T)
 LASTFRAME - (FRAME1)

New node created on level FRAME:

FRAME3 -
 PROTO - (FRAME)
 FRAME# - (3)
 SCORE1 - (5)
 CUMSCORE - (24)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME2)

Focus event - (FIRST-BALL FRAME3 ((LASTFRAME FRAME2) (SCORE1 5)
 (FRAME # 3) (CUMSCORE 24)) FIRST-BALL EV6 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 4

Number of applicable rule: 2

Node before application of rule:

FRAME3 -
 PROTO - (FRAME)
 FRAME# - (3)
 SCORE1 - (5)
 CUMSCORE - (24)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME2)

Node after application of rule:

FRAME3 -
 PROTO - (FRAME)
 FRAME# - (3)
 SCORE1 - (5)
 CUMSCORE - (33)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME2)

Focus event - (OPEN FRAME3 ((CUMSCORE 33)) SECOND-BALL EV7 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 10

Number of applicable rule: 9

New node created on level FRAME:

FRAME4 -
PROTO - (FRAME)
FRAME# - (4)
SCORE1 - NIL
CUMSCORE - (33)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME3)

Focus event - (STRIKE FRAME4 ((LASTFRAME FRAME3) (STRIKE T)
(FRAME# 4) (CUMSCORE 33)) FIRST-BALL EV8 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 9

Number of applicable rule: 3

New node created on level FRAME:

FRAMES -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (33)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME4)

Focus event - (FIRST-BALL FRAMES ((LASTFRAME FRAME4) (SCORE1 9)
(FRAME# 5) (CUMSCORE 33)) FIRST-BALL EV9 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 1

Number of applicable rule: 3

Number of applicable rule: 4

Node before application of rule:

FRAME4 -
PROTO - (FRAME)
FRAME# - (4)
SCORE1 - NIL
CUMSCORE - (33)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME3)
NEXTFRAME - (FRAME5)

Node after application of rule:

FRAME4 -
PROTO - (FRAME)
FRAME# - (4)
SCORE1 - NIL
CUMSCORE - (53)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME3)
NEXTFRAME - (FRAME5)

Node before application of rule:

FRAME5 -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (33)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME4)

Node after application of rule:

FRAME5 -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (53)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)

Node before application of rule:

FRAME5 -
PROTO - (FRAME)
FRAME# - (5)

SCORE1 - (9)
CUMSCORE - (53)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)

Node after application of rule:

FRAMES -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (53)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)

Focus event - (SPARE FRAMES ((SPARE T)) SECOND-BALL EV12 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 3

Number of applicable rule: 1

Node before application of rule:

FRAMES -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (53)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)

Node after application of rule:

FRAMES -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (66)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME4)

New node created on level FRAME:

FRAME6 -

PROTO - (FRAME)
FRAME# - (6)
SCORE1 - (3)
CUMSCORE - (66)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME5)

Focus event - (FIRST-BALL FRAME6 ((LASTFRAME FRAME5) (SCORE1 3)
(FRAME# 6) (CUMSCORE 66)) FIRST-BALL EV14 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 4

Number of applicable rule: 2

Node before application of rule:

FRAME6 -
PROTO - (FRAME)
FRAME# - (6)
SCORE1 - (3)
CUMSCORE - (66)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME5)

Node after application of rule:

FRAME6 -
PROTO - (FRAME)
FRAME# - (6)
SCORE1 - (3)
CUMSCORE - (73)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME5)

Focus event - (OPEN FRAME6 ((CUMSCORE 73)) SECOND-BALL EV15 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 10

Number of applicable rule: 9

New node created on level FRAME:

FRAME7 -

PROTO - (FRAME)
 FRAME# - (7)
 SCORE1 - NIL
 CUMSCORE - (73)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME6)

Focus event - (STRIKE FRAME7 ((LASTFRAME FRAME6) (STRIKE T)
 (FRAME# 7) (CUMSCORE 73)) FIRST-BALL EV16 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 10

Number of applicable rule: 9

New node created on level FRAME:

FRAME8 -
 PROTO - (FRAME)
 FRAME# - (8)
 SCORE1 - NIL
 CUMSCORE - (73)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME7)

Focus event - (STRIKE FRAME8 ((LASTFRAME FRAME7) (STRIKE T)
 (FRAME# 8) (CUMSCORE 73)) FIRST-BALL EV17 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 9

Number of applicable rule: 2

Node before application of rule:

FRAME7 -
 PROTO - (FRAME)
 FRAME# - (7)
 SCORE1 - NIL
 CUMSCORE - (73)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME6)
 NEXTFRAME - (FRAME8)

Node after application of rule:

FRAME7 -
 PROTO - (FRAME)
 FRAME# - (7)
 SCORE1 - NIL
 CUMSCORE - (102)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME6)
 NEXTFRAME - (FRAME8)

Node before application of rule:

FRAME8 -
 PROTO - (FRAME)
 FRAME# - (8)
 SCORE1 - NIL
 CUMSCORE - (73)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME7)

Node after application of rule:

FRAME8 -
 PROTO - (FRAME)
 FRAME# - (8)
 SCORE1 - NIL
 CUMSCORE - (102)
 STRIKE - (T)
 SPARE - NIL
 LASTFRAME - (FRAME7)

New node created on level FRAME:

FRAME9 -
 PROTO - (FRAME)
 FRAME# - (9)
 SCORE1 - (9)
 CUMSCORE - (102)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME8)

Focus event - (FIRST-BALL FRAME9 ((LASTFRAME FRAME8) (SCORE1 9)
 (FRAME# 9) (CUMSCORE 102)) FIRST-BALL EV20 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 1

Number of applicable rule: 3

Number of applicable rule: 4

Node before application of rule:

FRAME8 -
PROTO - (FRAME)
FRAME# - (8)
SCORE1 - NIL
CUMSCORE - (102)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME7)
NEXTFRAME - (FRAME9)

Node after application of rule:

FRAME8 -
PROTO - (FRAME)
FRAME# - (8)
SCORE1 - NIL
CUMSCORE - (122)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME7)
NEXTFRAME - (FRAME9)

Node before application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (102)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME8)

Node after application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (122)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME8)

Node before application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (122)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME8)

Node after application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (122)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME8)

Focus event - (SPARE FRAME9 ((SPARE T)) SECOND-BALL EV23 NIL)

Knowledge source invoked - FIRST-BALL

First ball : 4

Number of applicable rule: 1

Node before application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (122)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME8)

Node after application of rule:

FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (136)
STRIKE - NIL

SPARE - (T)
LASTFRAME - (FRAME8)

New node created on level FRAME:

FRAME10 -
PROTO - (FRAME)
FRAME# - (10)
SCORE1 - (4)
CUMSCORE - (136)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME9)

Focus event - (FIRST-BALL FRAME10 ((LASTFRAME FRAME9) (SCORE1 4)
(FRAME# 10) (CUMSCORE 136)) FIRST-BALL EV25 NIL)

Knowledge source invoked - SECOND-BALL

Second ball : 5

Number of applicable rule: 2

Number of applicable rule: 5

Node before application of rule:

FRAME10 -
PROTO - (FRAME)
FRAME# - (10)
SCORE1 - (4)
CUMSCORE - (136)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME9)

Node after application of rule:

FRAME10 -
PROTO - (FRAME)
FRAME# - (10)
SCORE1 - (4)
CUMSCORE - (145)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME9)

Node before application of rule:

FRAME10 -

PROTO - (FRAME)
 FRAME# - (10)
 SCORE1 - (4)
 CUMSCORE - (145)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME9)

Node after application of rule:

FRAME10 -
 PROTO - (FRAME)
 FRAME# - (10)
 SCORE1 - (4)
 CUMSCORE - (145)
 STRIKE - NIL
 SPARE - NIL
 LASTFRAME - (FRAME9)

Focus event - (LASTFRAME FRAME10 NIL SECOND-BALL EV27 NIL)

Knowledge source invoked - FINALFRAME

Number of applicable rule: 3

Node before application of rule:

NIL -

Node after application of rule:

NIL -

 SCORE: 145
 STRIKES IN FRAMES: (FRAME4 FRAME7 FRAMES3)
 SPARES IN FRAMES: (FRAME2 FRAMES5 FRAME9)

.....
LEVEL NODE OF LEVEL 1

FRAME -
 FRAME# - NIL
 SCORE1 - NIL
 CUMSCORE - NIL
 STRIKE - NIL
 SPARE - NIL
 INST - (FRAME10 FRAME9 FRAMES3 FRAME7 FRAME6 FRAMES FRAME4 FRAME3
 FRAME2 FRAME1)

.....
THE FOLLOWING ARE THE OTHER NODES (AND THEIR PROPERTY LISTS)
ON LEVEL 1

.....
FRAME1 -
PROTO - (FRAME)
FRAME# - (1)
SCORE1 - (3)
CUMSCORE - (9)
STRIKE - NIL
SPARE - NIL
NEXTFRAME - (FRAME2)

.....
FRAME2 -
PROTO - (FRAME)
FRAME# - (2)
SCORE1 - (4)
CUMSCORE - (24)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME1)
NEXTFRAME - (FRAME3)

.....
FRAME3 -
PROTO - (FRAME)
FRAME# - (3)
SCORE1 - (5)
CUMSCORE - (33)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME2)
NEXTFRAME - (FRAME4)

.....
FRAME4 -
PROTO - (FRAME)
FRAME# - (4)
SCORE1 - NIL
CUMSCORE - (53)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME3)
NEXTFRAME - (FRAME5)

.....
FRAME5 -
PROTO - (FRAME)
FRAME# - (5)
SCORE1 - (9)
CUMSCORE - (66)
STRIKE - NIL
SPARE - (T)

LASTFRAME - (FRAME4)
NEXTFRAME - (FRAME6)

.....
FRAME6 -
PROTO - (FRAME)
FRAME# - (6)
SCORE1 - (3)
CUMSCORE - (73)
STRIKE - NIL
SPARE - NIL
LASTFRAME - (FRAME5)
NEXTFRAME - (FRAME7)

.....
FRAME7 -
PROTO - (FRAME)
FRAME# - (7)
SCORE1 - NIL
CUMSCORE - (102)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME6)
NEXTFRAME - (FRAME8)

.....
FRAME8 -
PROTO - (FRAME)
FRAME# - (8)
SCORE1 - NIL
CUMSCORE - (122)
STRIKE - (T)
SPARE - NIL
LASTFRAME - (FRAME7)
NEXTFRAME - (FRAME9)

.....
FRAME9 -
PROTO - (FRAME)
FRAME# - (9)
SCORE1 - (9)
CUMSCORE - (136)
STRIKE - NIL
SPARE - (T)
LASTFRAME - (FRAME8)
NEXTFRAME - (FRAME10)

.....
FRAME10 -
PROTO - (FRAME)
FRAME# - (10)
SCORE1 - (4)
CUMSCORE - (145)
STRIKE - NIL
SPARE - NIL

LASTFRAME - (FRAME9)

.....End of RUN

What do you want to do? : q

Exit from LISP.

**Copyright © 1985 by KSL and
Comtex Scientific Corporation**

FILMED FROM BEST AVAILABLE COPY