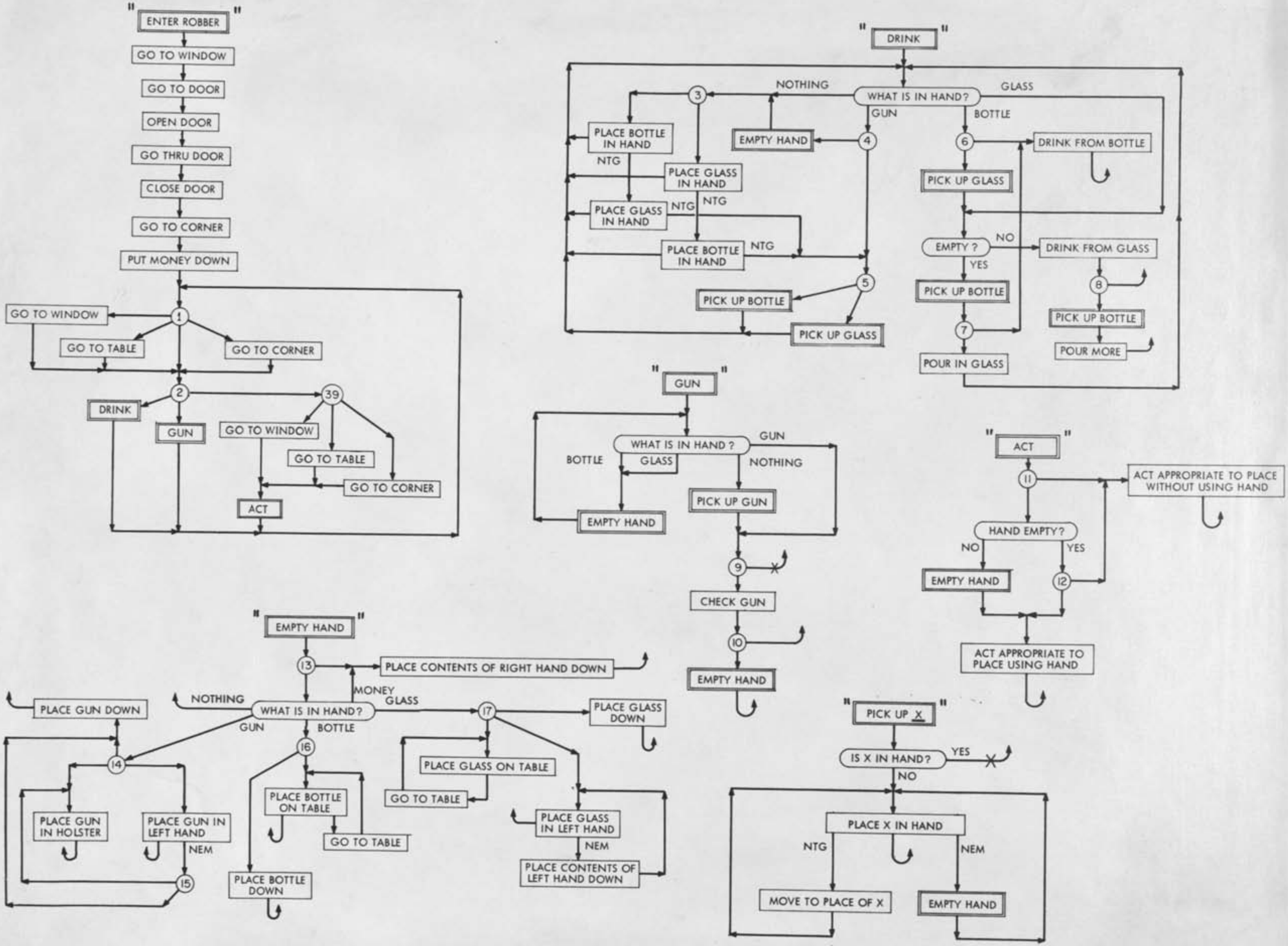
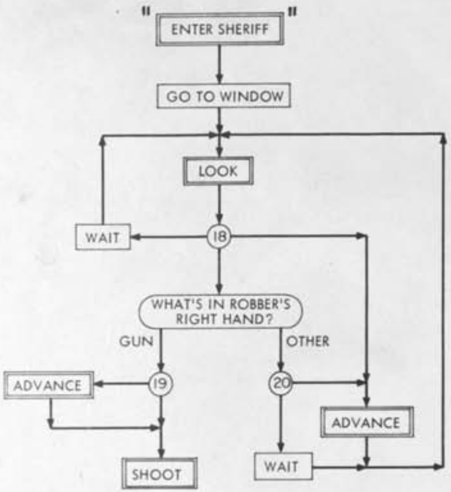
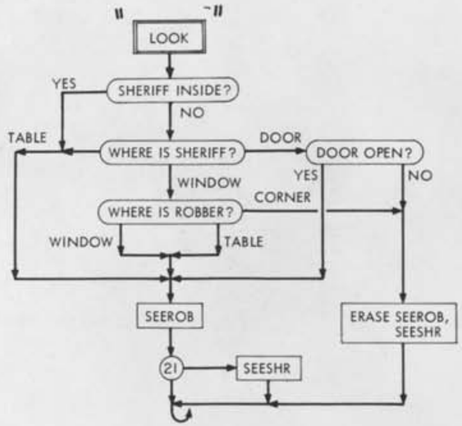
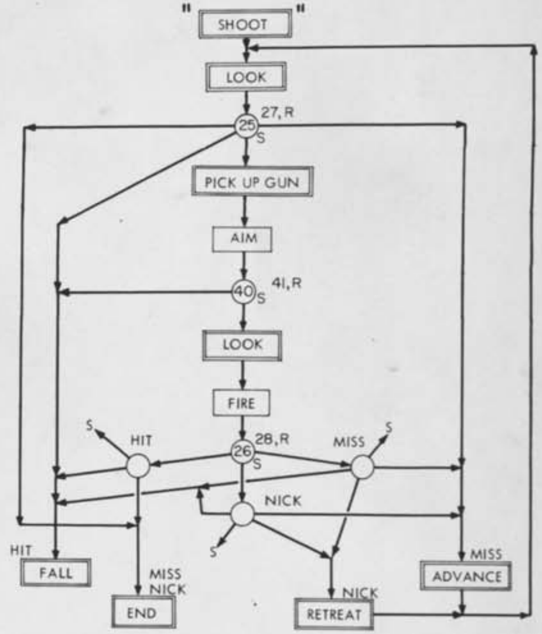
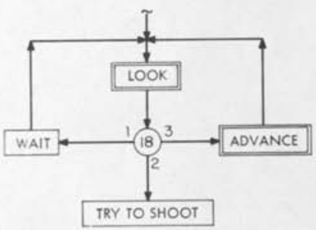


"ENTER ROBBER"
GO TO WINDOW
GO TO DOOR
OPEN DOOR
GO THRU DOOR
CLOSE DOOR
GO TO CORNER
PUT MONEY DOWN
GO TO WINDOW
GO TO TABLE
GO TO CORNER
DRINK
GUN
GO TO WINDOW
GO TO TABLE
GO TO CORNER
ACT
"DRINK"
WHAT IS IN HAND?
NOTHING
GLASS
GUN
BOTTLE
PLACE BOTTLE IN HAND
EMPTY HAND
DRINK FROM BOTTLE
NTG
PLACE GLASS IN HAND
PICK UP GLASS
PLACE GLASS IN HAND
NTG
NTG
PLACE BOTTLE IN HAND
NTG
EMPTY ?
NO
DRINK FROM GLASS
YES
PICK UP BOTTLE
PICK UP BOTTLE
PICK UP BOTTLE
PICK UP GLASS
POUR MORE
POUR IN GLASS
"GUN"
WHAT IS IN HAND ?
GUN
BOTTLE
GLASS
NOTHING
PICK UP GUN
EMPTY HAND
CHECK GUN
EMPTY HAND
"ACT"
ACT APPROPRIATE TO PLACE WITHOUT USING HAND
HAND EMPTY?
NO
YES
EMPTY HAND
ACT APPROPRIATE TO PLACE USING HAND
"EMPTY HAND"
PLACE CONTENTS OF RIGHT HAND DOWN
MONEY
NOTHING
GLASS
PLACE GUN DOWN
WHAT IS IN HAND?
PLACE GLASS DOWN
GUN
BOTTLE
PLACE GLASS ON TABLE
GO TO TABLE
PLACE BOTTLE ON TABLE
PLACE GUN IN HOLSTER
PLACE GUN IN LEFT HAND
PLACE GLASS IN LEFT HAND
NEM
GO TO TABLE
NEM
PLACE CONTENTS OF LEFT HAND DOWN
PLACE BOTTLE DOWN
"PICK UP X"
IS X IN HAND?
YES
NO
PLACE X IN HAND
NTG
NEM
MOVE TO PLACE OF X
EMPTY HAND

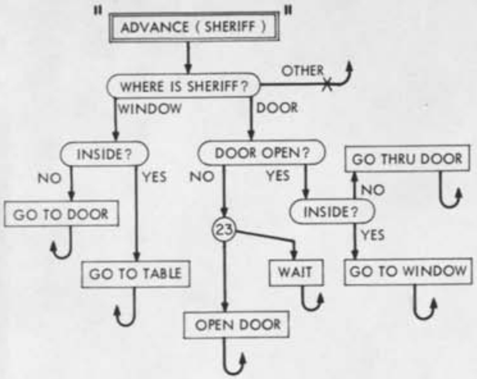


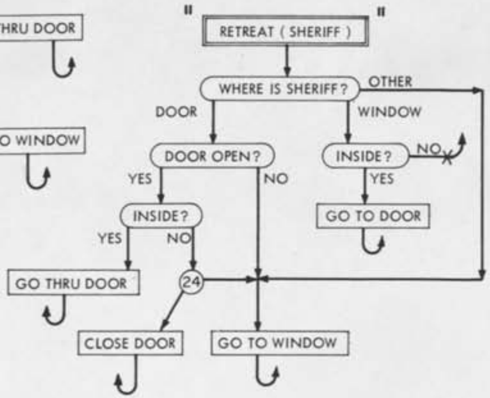


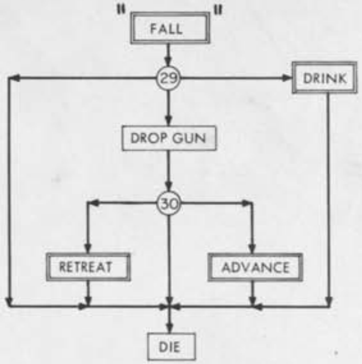


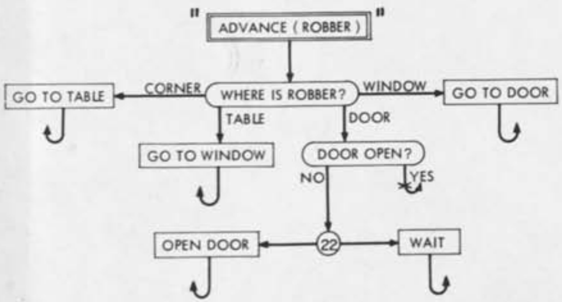


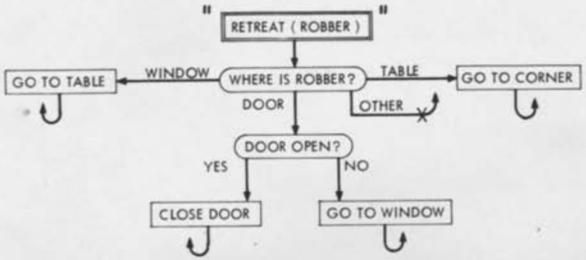
BRANCH		WEIGHT FORMULA	R = -10 S = -10	R = +10 S = -10	R = +10 S = +10
1	WAIT	10 - SEEROB	20	0	0
2	TRY	60 + 5 SEEROB	10	110	110
3	ADV	110 + SEEROB + 10 SEESHR	0	20	220

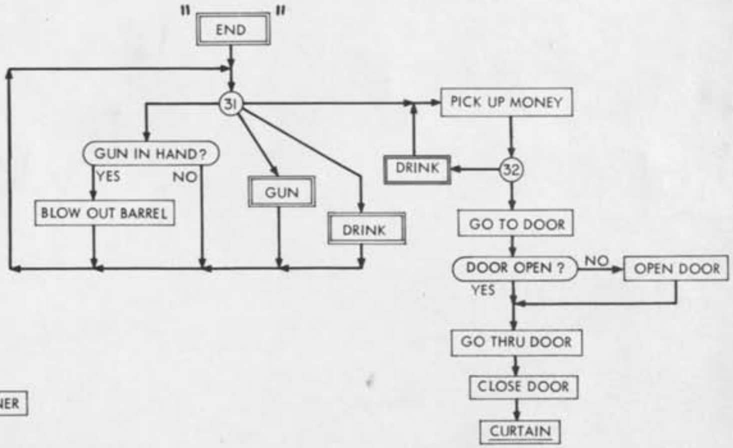

"ADVANCE (SHERIFF)"
WHERE IS SHERIFF?
OTHER
WINDOW
DOOR
INSIDE?
NO
YES
GO TO DOOR
GO TO TABLE
DOOR OPEN?
NO
YES
23
WAIT
OPEN DOOR
INSIDE?
NO
YES
GO THRU DOOR
GO TO WINDOW


"RETREAT (SHERIFF)"
WHERE IS SHERIFF?
OTHER
DOOR
WINDOW
DOOR OPEN?
YES
NO
INSIDE?
YES
NO
GO THRU DOOR
24
CLOSE DOOR
GO TO WINDOW
INSIDE?
NO
YES
GO TO DOOR


"FALL"
29
DRINK
DROP GUN
30
RETREAT
ADVANCE
DIE


"ADVANCE (ROBBER)"
GO TO TABLE
CORNER
WHERE IS ROBBER?
WINDOW
GO TO DOOR
TABLE
DOOR
GO TO WINDOW
DOOR OPEN?
NO
YES
OPEN DOOR
22
WAIT


"RETREAT (ROBBER)"
GO TO TABLE
WINDOW
WHERE IS ROBBER?
TABLE
GO TO CORNER
DOOR
OTHER
DOOR OPEN?
YES
NO
CLOSE DOOR
GO TO WINDOW


"END"
31
PICK UP MONEY
GUN IN HAND?
YES
NO
BLOW OUT BARREL
GUN
DRINK
DRINK
32
GO TO DOOR
DOOR OPEN ?
NO
OPEN DOOR
YES
GO THRU DOOR
CLOSE DOOR
CURTAIN

Cunningham & Walsh Inc. Public Relations Department • 260 Madison Ave., New York 16, MUrray Hill 3-4900

From: Joe Quinn

Released for: American Machine & Foundry Company
261 Madison Avenue
New York 16, N. Y.

FOR IMMEDIATE RELEASE

CAN MACHINES REALLY THINK?

WHAT DO SCIENTISTS WHO MAKE

THE MACHINES THINK ABOUT THIS?

"The Thinking Machine", an intriguing and informative hour of television, will be seen from 10 to 11 P.M., Wednesday, October 26, over the CBS Television network.

It will be presented by American Machine & Foundry Company.

"The Thinking Machine" will delve into the workings of man's brain. It will show how man thinks and illustrate what machines do that looks like "thinking".

The show was produced by CBS News in association with the Massachusetts Institute of Technology, which soon will mark its hundredth anniversary.

MIT scientists who will appear in "The Thinking Machine" are agreed that there is "no black magic" in computers that "think." But they split into divergent camps on the question:

"Can Machines Really Think?"

Dr. Claude Shannon, Donner Professor of Science at MIT and a world authority on information theory, opens "The Thinking Machine" by expressing confident expectation that "within a matter of ten or 15 years something will emerge from the laboratory which is not too far from the robot of science fiction fame."

Then Dr. Jerome B. Wiesner, director of the Research Laboratory of Electronics at MIT, spends the better part of an hour illustrating to David Wayne, Broadway and Hollywood star, the things machines can do that look like "thinking."

Throughout the program, Wayne is impressed and keeps pressing Dr. Wiesner for an answer: "Do Machines really think?"

Dr. Wiesner leaves part of the answer to his colleagues.

Dr. Patrick Wall, associate professor of biology and executive officer of the Department of Biology at MIT, says flatly:

"I'm not at all convinced yet that machines can think. I spend my life trying to analyze real brains and, of course, I'm very impressed with people who are building artificial brains...But can these machines produce anything new?

"I don't yet see in these machines any possibility of them stepping over man-made rules which have been built into them. I believe in fact that before we begin to ask this question -- can machines think? -- we have to find out a great deal more about what we mean by our own thinking."

On the other hand, Professor Oliver Selfridge of the Lincoln Laboratory at MIT, says emphatically during "The Thinking Machine":

"I am convinced that machines can and will think. I'm not worried about the behavior of machines. I don't believe that we'll ever find it hard to distinguish between a man and a robot. And I'm not worried about my daughter marrying a computer....

(more)

"Now machines can't write good poetry or produce deathless music -- yet -- but I don't see any stumbling block in the line of progress which will enable them to in the long run. I'm convinced that machines can and will think in our lifetime."

Toward the close of the program, Dr. Shannon sums up the remarkable accomplishments of machines that translate from one language to another to a certain extent, machines that prove mathematical theorems and even play checkers, in some cases, better than the men who designed the machines.

"These, however, were projects aimed at specific mental tasks," Dr. Shannon says. "What we would like to see in the future is a more general computing system capable of learning by experience and forming inductive and deductive thoughts. This would probably consist in three main parts:

"1. Sense organs akin to the human eye or ear whereby the machine can take cognizance of events in its environment.

"2. A large, general-purpose flexible computer programmed to learn from experience, to form concepts and capable of doing logic.

"3. Output devices in the nature of the human hand, capable of allowing the machine to make use of the thoughts that it had, of the cognizant processes in order to actually affect the environment.

"Work is going on in all of these fronts simultaneously and rapid progress is being made. I confidently expect that within a matter of 10 or 15 years, something will emerge from the laboratory which is not too far short from the robot of science fiction fame. In any case, this is certainly one of the most exciting and challenging branches of modern science."

"The Thinking Machine" promises to be one of the most intriguing and informative shows of the television season.

# # #

Cunningham & Walsh Inc. Public Relations Department • 260 Madison Ave., New York 16, MUrray Hill 3-4900

From: Joe Quinn

Released for: American Machine & Foundry Company
261 Madison Avenue
New York 16, N. Y.

FOR IMMEDIATE RELEASE

TX-O DIGITAL COMPUTER DEVELOPED AT MIT

WRITES WESTERN DRAMA TO BE SEEN ON CBS-TV

FROM 10 TO 11 P.M. (EDT) WEDNESDAY, OCT. 26

How does a computer write a Western drama for television?

The answer will be seen in an intriguing and informative hour of television from 10 to 11 P.M. (EDT) Wednesday, October 26, on the CBS TV Network. It will be presented by American Machine & Foundry Company.

But in the meantime, Douglas T. Ross, computer scientist at the Massachusetts Institute of Technology, gives us an insight into the creative life of a computer-playwright.

Ross and several MIT colleagues "coached" the computer to write a TV drama for "The Thinking Machine." It obliged by writing not one, but two Westerns.

And in the doing, the computer -- MIT's TX-O digital Computer -- injected a note of originality. In the first computer-written Western, the robber dies in accepted Western tradition.

But in another version, the computer permits the robber to kill the sheriff -- hardly the triumph of justice, but it is a new twist to an old tale.

Ross, 30-year-old head of the Computer Applications Group in the Electronics Systems Laboratory at MIT, said the computer, like its human counterpart, builds a Western drama according to a certain set of rules -- thus providing a demonstration of what scientists call artificial intelligence.

"But it is not a demonstration that authors are being pushed into oblivion," Ross said. "The chances of ever creating an electronic Euripedes or a transistorized Tolstoy are infinitely negligible.

"But interestingly enough, many of the new techniques we developed in the program will prove very useful in more serious programming problems."

Ross said the MIT scientists designed a program for the computer's debut in playwriting after CBS producer-author Tom Wolf inquired if the computer could write a Western.

Ross' immediate answer was "yes." He said there were three principal and serious problems encountered in what he termed an "interesting study".

"We wanted to demonstrate in a simple way that, first, intelligent behavior is rule-obeying behavior," Ross said. "If appropriate general rules are not obeyed, unintelligent behavior will result.

"Secondly, we wanted to show that a computer can be made to perform supposedly creative tasks.

"And thirdly, we wanted to emphasize that there is no black magic about computers. Given a problem that a layman can follow, he can get a reasonable understanding of computer programming."

To achieve these objectives, the computer's "coaches" had to consider every basic aspect of the problem and/or plot.

The scientists gave the computer a group of things, telling it what properties they had, and gave the computer suggested rules for ways in which they could be put together. This essentially is what a human author does, Ross said.

(more)

The scientists also gave the computer's "play" a beginning (the robber enters and puts down the money) and an ending (the survivor, if any, walks out with the money and closes the door). The survivor might be either the sheriff or the robber.

Ross emphasized that the computer had to be told how to be intelligent. He said "we could have forced the computer to be intelligent by forcing the weight balance to yield particular choices."

"But we didn't want to," Ross said. "Everybody has the prerogative of not acting intelligently even when it knows better -- and we wanted the computer to have it, too.

"We also wanted to experiment with a situation that might lead the computer to break the rules of intelligent behavior, but break them in a natural way. This led to the introduction of the inebriation factor.

"Every time the sheriff or the robber takes a drink -- and we programmed opportunities for doing this -- the inebriation factor is increased, making intelligent actions a little less probable and unintelligent actions a little more probable."

After a few drinks, a man takes longer to do things. After the computer's robber has had several drinks, instead of aiming only once to fire, he often aims three times before firing, Ross said.

Ross said the probability of the robber's not completing an action also increased with the more he drank. He said that in experiments the MIT scientists several times send the robber off into an electronic stupor of muddled over-and-over-again action.

The young MIT scientist said his group spent a month in planning and about two weeks in cleaning up the program before they began to get the results they wanted -- and which the viewer will see on his television screen October 26.

"Once everything was running smoothly," Ross said, "the computer produced a three or four-page Western in about two minutes. Moreover, it turned out dozens of plays, and although these varied, justice usually triumphed.

"This resulted because probability weights which specified the marksmanship of the robber and the sheriff were identical except for one thing -- the inebriation factor of the robber.

"The wages of 'sin' were that the robber lost three-to-one to the sober sheriff."

The sheriff didn't imbibe until <u>after</u> the battle, Ross concluded.

# # #

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Electronic Systems Laboratory
Department of Electrical Engineering
Cambridge 39, Massachusetts

MEMORANDUM 8436-M-29

TO: D.T. Ross

FROM: H.R. Morse

SUBJECT: Preliminary Operating Notes for SAGA II

DATE: October 19, 1960

This memorandum presents, in condensed form, the features of the control and operating program for SAGA II, the TV script-writing program. The control routine permits modification, printing and punching of "switch" values; tracing the operation of SAGA at several levels of detail; and examination and modification of certain desired quantities, all under control of the on-line Flexowriter. No program details are given in this brief memorandum. A complete description of the design and programming of SAGA II will be issued later as a Technical Memorandum.

Starting Procedure:

Upon read-in, control is automatically transferred to the control routine.

Typing the character r followed by a carriage return (↩) will reset and start SAGA.

The run number typed is the contents of the random number generator at the start of the play, and may be examined and modified by typing r=. Starting SAGA with the same switch settings, and the same number in the random number generator will reproduce a play.

The timing number initially controls the length of time until the sheriff arrives. If it is positive the sheriff will arrive immediately after the robber; the more negative the timing number, the longer until the sheriff arrives. This timing number may be examined and modified by typing t=, and is initially set to -1000.

Switches

A "switch" is a probability branching device which controls the sequencing of action in SAGA. The state of the action is represented by 16 "state variables" which are multiplied by weights (a-values) to determine the probability of taking any given branch of a switch. Considering switch n, the value for branch k is

$$P_{n,k} = a_{n,k,0} + \sum_{j=1}^{16} (a_{n,k,j})\cdot(b_j)$$

where the b's are the state variables

The P's are scaled, by a simple procedure, to be fractions representing the probabilities for the respective branches.

Probabilities are calculated each time a switch is entered, since the state variable can change at any time, after which a random number is generated, and used to choose a branch. A list of state variables is given in Appendix 1.

A flow chart which explains the use of each switch can be found in the TX-0 computer room.

The "character" of the program is controlled by the switches, so by changing the a-values of a given switch, one may make the robber an excellent shot, or a lousy one; make the sheriff a fast actor and the robber a slow one; make the robber a good shot when drunk; or make the sheriff a brave man; to give a few examples. Examples of changing switches will be given later.

Switch Examination and Modification

- sN↵ selects switch N.
- sNr will reset switch N, and ask for the number of branches. Switch N may then be defined or redefined.
- sNp selects switch N and prints the a-values for each branch, with the exception of most zero a-values.

After a switch has been selected, an a-value may be examined by typing aI,J=, where I is the branch number, and J the number of the associated state variable. The a-value may then be modified by typing the new value followed by a carriage return (↵), or left as is by just typing a carriage return. After modifying an a-value in a branch, other a-values in the same branch may be referred to by typing aJ= where J refers to the associated state variable. Switch numbers, branch numbers, and state variable numbers are in decimal, while the a-values are taken as octal, and should be typed as such. The value of A_o for any brancy may be full range

$$-2^{17} \quad A_o \quad +2^{17},$$

while all other a-values are taken modulo 2^5.

Error Indications Possible When Modifying A-values

nbr	There exists no such branch
ado	The switch cannot accomodate another a-value at present. Redefining switch N by sNr will eliminate the problem
nrm	Means the a-table is full. Uncorrectable.

If a modified a-table produces good plays, it may be punched out by typing <u>ap</u>, and read into SAGA when desired. Switch 42 has been reserved for identification purposes; $a_{1,0}$ containing the octal date, and $a_{2,0}$ containing the number of the a-table punched on that date (first, second, etc.).

<u>An Example of Modifying A Switch</u>

SWITCH 26

br1	a0 = 10	a12 = 30	
br2	a0 = 60	a5 = 1	
br3	a0 = 110	a1 = -10	a5 = 3

Switch 26 controls whether the sheriff hits (branch 1), nicks (branch 2) or misses (branch 3) the robber when he shoots.

State variable 1 is +10 if the sheriff sees the robber
-10 if he does not
" " 5 is 0 if the sheriff has not been nicked
+10 if he has
" " 12 is +10 if sheriff is using his last bullet
0 if he is not.

So if the sheriff is not using his last bullet, sees the robber, and has not been nicked, then the probabilities are

p1 = 10 (hit)
p2 = 60 (nick) (1)
p3 = 10 (miss

i.e. he is 6 times more likely to nick the robber then miss or hit him.

The sheriff could be made a good shot by setting $a_{1,0}$ = 60 and $a_{2,0}$ = 10. Then the probabilities for the above case become

p1 = 60 (hit)
p2 = 10 (nick)
p3 = 10 (miss)

Or he could be made a lousy shot by setting $a_{3,0} = 60$ and $a_{2,0} = 10$. Then we have

p1 = 10 (hit)

p2 = 10 (nick)

p3 = 60 (miss)

Also, it can easily be seen how changes in the state of the program would affect the probabilities on the branches, i.e. - if the sheriff were nicked (b5 = 10) then (1) would become

p1 = 10 (hit)

p2 = 70 (nick)

p3 = 40 (miss)

The a-values may be minus, but if a final branch-probability is minus, then the probability is set to zero. Notice also that there is no state variable zero. A_o is used purely to balance the branch, and is treated as if state variable zero were always 1.

Program Interruptions and Tracing

The remaining part of the control program will be mentioned only briefly here, as it was designed mainly for debugging purposes.

The program may be interrupted at any time by changing the position of TAC_o, which should be 0 upon read-in. SAGA will break at the next switch, and transfer to the control program. Any of the control operations may be performed, and the play continued by typing a period (.).

Tracing features:

Typing p — will cause each switch number to be printed as it is used, followed by the probabilities calculated for each branch, and the play continued.

n — will cause just the switch numbers to be printed, and the play continued.

b — will cause the program to wait for a command from the flexo at each switch, and may be used in conjunction with p and n.

e will cause p,b,n to be erased.

. will always continue the interrupted play after a break.

sNb may be preceeded by p or n, and followed by b, all having the same effect as above when switch N is used. As many as 8 switch break points may be active at once.

sb erases all switch break points.

STATE VARIABLES

No.	Condition	Value Yes	Value No
1	shf sees rob	+10	-10
2	rob sees shf	+10	-10
3	sheriff is coming	+10	0
4	rob is aiming	+10	0
5	sheriff nicked	+10	0
6	robber nicked	+10	0
7	sheriff hit	+10	0
8	robber hit	+10	0
9	inebriation factor	starts at zero, stepped by +10 for every drink from glass, +20 for every drink from bottle that rob takes.	
10	sequence	Stepped in varying units of +10 if rob acting, by -10 if shf acting.	
11	robber's last bullet	+10	0
12	sheriff's last bullet	+10	0
13	door open	+10	0
14	sheriff inside	+10	0
15	robber inside	+10	0
16	drink full	+10	0

SUMMARY OF COMMANDS

Typed Command	Effect
r↵	reset and restart SAGA
r=	examine contents of random number generator
t=	examine initial setting of timing number
sN↵	select switch N
sNr	reset switch N
sNp	select and print switch N
sp	print all switches
aI,J=	examine aJ for branch I of switch N
aJ=	examine aJ for last branch examined of switch N
ap	punch entire a-table
b	break at each switch entered
.	continue after break
p	print switch numbers and branch probabilities as used and continue play (unless command followed by b)
n	as p, except print just switch numbers
sNb	break only at switch N. (this may be preceeded by p or n, or followed by b, with the expected results). As many as 8 breaks may be used at once.
sb	erase all switch break points
e	erase all of the break commands
TAC_o	changing TAC_o will cause SAGA to break at the next switch used. This is a "one shot" break. As usual (.) will continue the play.

NEW YORK TIMES
Sunday, October 2, 1960

NEWS OF TV AND RADIO: GADGETRY

No Hands! Computer Writes a TV Western For C.B.S.—Items

By VAL ADAMS

A MACHINE-written Western playlet, in which a sheriff and a bandit have a gun duel, will be televised this month by the Columbia Broadcasting System. The machine, an electronic computer called TX-O, was developed by the Massachusetts Institute of Technology. Now that C. B. S. is in the act, TX-O may become the Zane Grey of computers and enjoy many happy residuals.

The two-minute playlet will be presented on "Tomorrow," a new science series. It begins on Oct. 26 at 10 P. M. with a one-hour program titled "The Thinking Machine." David Wayne, the actor, will be seen visiting M. I. T. and talking to scientists about machines that seem capable of reasoning. After Mr. Wayne watches TX-O type out an "original" Western, the script, which is without dialogue, is performed by two actors.

The idea of asking M. I. T. for a machine-written script originated with Thomas H. Wolf, executive producer of "Tomorrow." TX-O was provided with a dramatic situation in which a robber with newly-stolen money enters a hideout shack and is overtaken there by the sheriff. The machine, which was "told" that objects in the shack included money, a table, a glass and a bottle of whiskey, then typed out the chain of action and arrived at its own dénouement.

In a series of test runs, TX-O was programed with the "shoot-it-out" situation fifty times but never did it type out the identical script twice. The good man killed the bad man about three times out of four. In some cases TX-O went on a real six-gun toot and let each kill the other.

C. B. S. will show one version where the sheriff wins and another where the robber is victorious. Then just to demonstrate that TX-O's reasoning can go off the beam, another script will show the sheriff putting his gun in the robber's holster, pouring whiskey into a glass but drinking from the bottle.

The following is the last part of one of the Western dramas as it was typed out by TX-O:

SHERIFF: The sheriff is at the window. Go to door. Wait. Open door. Sees robber. Sees sheriff. Go through door.

ROBBER: Take gun from holster with right hand. Aim.

SHERIFF: Aim. Fire. Robber hit. Blow out barrel. Put gun in holster.

ROBBER: Fire. Missed. Drop gun. Go to table. Robber dies.

SHERIFF: Go to corner. Pick up money with right hand. Go to door. Go through door. Close door. Curtain.

C. B. S. has not provided equal time for real-life writers to appear on the program in rebuttal of TX-0. But Mr. Wayne warns an M. I. T. scientist:

"If the computer ever learns to act, I'll break its back."

WED OCT 26 10 PM
CHANNEL 7

run 112263

the gun is in the right hand

the money is in the left hand

the glass is on the table

the bottle is on the table

the holster is on the robber

the sheriffs gun is in the sheriffs right hand

the sheriffs holster is on the sheriff

Good
9-3

ROBBER:

the robber is at the window

go to door
open door
go thru door
close door

go to corner

put money down at corner
check gun

put gun in holster

go to table

pick up bottle with right hand

put bottle on table

pick up glass with right hand
empty

take glass from right hand with left hand

pick up bottle with right hand
pour

put bottle on table

take glass from left hand with right hand
take a drink from glass

take glass from right hand with left hand

pick up bottle with right hand
pour

put bottle on table
sit and fidgit

take gun from holster with right hand

go to corner

put gun in holster
count money

take gun from holster with right hand
check gun

go to table

put gun on table

take glass from left hand with right hand
take a drink from glass

take glass from right hand with left hand

pick up bottle with right hand
pour

SHERIFF:

the sheriff is at the window
sees robber
sees sheriff

go to door

ROBBER:

put bottle on table

pick up gun with right hand
check gun

put gun in holster

go to window

take gun from holster with right hand
check gun

put gun down at window

SHERIFF:

wait
wait
wait
open door
sees robber
sees sheriff

ROBBER:

pick up gun with right hand
aim
fire
MISSED

go to door

SHERIFF:

go thru door
aim

ROBBER:

aim
fire
SHERIFF HIT

SHERIFF:

fire
MISSED

put gun in holster

take glass from robbers left hand with right hand
take a drink from glass

go to table

put glass on table

pick up bottle with right hand
pour

ROBBER:

put gun in holster

go to corner

pick up money with right hand

SHERIFF:

sheriff dies

ROBBER:

go to door
go thru door
close door
curtain

r=675530 714234
r=714234
r
run 714234

the gun is in the holster

the money is in the left hand

the glass is on the table

the bottle is on the table

the holster is on the robber

the sheriffs gun is in the sheriffs holster

the sheriffs holster is on the sheriff

ROBBER:

the robber is at the window

go to door

open door

go thru door

close door

go to corner

put money down at corner

go to table

pick up bottle with right hand

put bottle on table

pick up glass with right hand

empty

take glass from right hand with left hand

pick up bottle with right hand

pour

put bottle on table

take glass from left hand with right hand

take a drink from the glass

go to corner

go to corner

go to table

put glass on table

sit and fidgit

sit and fidgit

go to window

go to table

pick up bottle with right hand

put bottle on table

pick up glass with right hand

empty

take glass from right hand with left hand

pick up bottle with right hand

pour

put bottle on table

take glass from left hand with right hand

take a drink from the glass

go to corner

put glass down at corner

go to table

pick up bottle with right hand

take a drink from the bottle

SHERIFF:

the sheriff is at the window

sheriff sees robber

robber sees sheriff

wait

go to door

wait

open door

sheriff sees robber

robber sees sheriff

ROBBER:

put bottle on table

take gun from holster with right hand

aim

fire

SHERIFF HIT

blow out barrel

put gun on table

go to corner

pick up money with right hand

SHERIFF:

go thru door

go to window

drop gun

go to table
sheriff dies

ROBBER:

go to door

go thru door

close door

CURTAIN

2

Your number is 01
Rating 00

First move by player

Your move is
4x
r
run 0

the gun is in the holster

the money is in the left hand

the glass is on the table

the bottle is on the table

the holster is on the robber

the sheriffs gun is in the sheriffs holster

the sheriffs holster is on the sheriff

ROBBER:

the robber is at the window

go to door

open door

go thru door

close door

go to corner

put money down at corner

gloat over money

gloat over money

go to table

sit at table

go to corner

count money

go to table

pick up bottle with right hand

put bottle on table

pick up glass with right hand

empty

put glass on table

pick up bottle with right hand

pour

put bottle on table

pick up glass with right hand

take a drink from the glass

go to corner

take glass from right hand with left hand

take gun from holster with right hand

check gun

put gun in holster

go to table

sit at table

go to corner

gloat over money

SHERIFF:

the sheriff is at the window

wait

wait

ROBBER:

take glass from left hand with right hand

empty

put glass down at corner

go to table

pick up bottle with right hand

SHERIFF:

go to door

open door

sheriff sees robber

robber sees sheriff

wait

ROBBER:

put bottle on table

take gun from holster with right hand

aim

fire

MISSED

SHERIFF:

go thru door

take gun from holster with right hand

ROBBER:

go to window

go to door

SHERIFF:

aim

fire

ROBBER NICKED

ROBBER:

aim

aim

fire

SHERIFF NICKED

SHERIFF:

go to window

aim

fire

ROBBER NICKED

ROBBER:

ROBBER:

aim

fire

SHERIFF HIT

blow out barrel

put gun down at door

SHERIFF:

drop gun

go to door

go to window

go to table

ROBBER:

go to corner

pick up glass with right hand

empty

put glass down at corner

go to table

pick up bottle with right hand

SHERIFF:

go to window

pick up gun with right hand

aim
sheriff dies

ROBBER:

go to corner

pour

take a drink from the bottle

take a drink from the bottle

put bottle down at corner

pick up glass with right hand

take a drink from the glass

put glass down at corner

put glass down at corner

pick up bottle with right hand

pour

put bottle down at corner

go to door

pick up gun with right hand

put gun in holster

go to corner

pick up bottle with right hand

put bottle down at corner

pick up glass with right hand

take a drink from the glass

put glass down at corner

pick up money with right hand

put money down at corner

pick up glass with right hand

empty

go to table

put glass on table

go to corner

pick up bottle with right hand

take a drink from the bottle

put bottle down at corner

pick up money with right hand

put money down at corner

go to table

pick up glass with right hand

empty

put glass on table

go to corner

pick up bottle with right hand

take a drink from the bottle

put bottle down at corner

pick up money with right hand

put money down at corner

pick up bottle with right hand

put bottle down at corner

go to table

pick up glass with right hand

empty

put glass on table

go to corner

pick up bottle with right hand

take a drink from the bottle

go to table

put bottle on table

go to corner

pick up money with right hand

put money down at corner

go to table

pick up bottle with right hand

take a drink from the bottle

put bottle on table

go to corner

pick up money with right hand

put money down at corner

go to table

pick up glass with right hand

empty

put glass on table

pick up bottle with right hand

take a drink from the bottle

put bottle on table

go to corner

pick up money with right hand

go to door

go thru door

go thru door

close door

CURTAIN