LECTURE

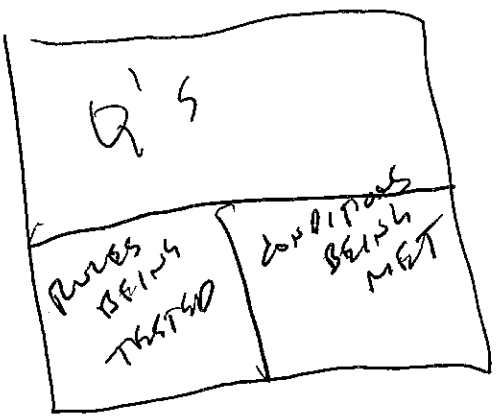
EXPERT SYSTEM CASE STUDY ~~#~~ #1

~~(i.e., this one comes before "CASE STUDY #~~)

"Advise Callers to a Doctor's Office"

CS 375 AND 434
Artificial Intelligence AND ROBOTICS

Dr. J. Wunderlich



- DOUBLE CLICK ON VPX.EXE
- PRESS "4" TO CONSULT
- HIT ENTER TO PICK "EXP1"
- PRESS "2" TO EXECUTE
- HIT ARROWS TO MOVE AROUND SELECTIONS, HIT ENTER TO TAG CERTAIN ONES

HIT END TO COMPLETE

DO "5" TO SHOW ~~[illegible]~~ RULES

TO SET TRACE, DO "6" TO SET, THEN "2" FOR (AFTER "4")

THEN "6" TO QUIT, "2" TO GO [SAVE AS TRACE.TRC USE TRACE (TEXT OR GRAPHICS)]

TO SHOW TRACE, DO "5" FOR TRACE, THEN "3" FOR GRAPHICS

TABLE OF CONTENTS

I. INTRODUCTION

This expert system has been developed to provide advice to a caller to a doctors office. The user of the system is the person who answers the doctor's telephone. The user is assumed to have limited medical knowledge and should not be making medical decisions which could jeopardize the health of a caller. Therefore, the expert system provides the user with questions to ask the caller which, when answered, should allow the expert system to provide advice for the caller. The primary decision of this expert system is to decide which of the following advice to give to the caller:

#1) Go to hospital immediately
#2) Not worry, because he/she is not sick; just take a walk
#3) Take two aspirin and call back tomorrow
#4) Come in and see the doctor

It is important to note that (#3) and (#4) are based on the same set of apparent symptoms; they only differ in how they rely on the doctor being in the office; (#3) if doctor is in, or (#4) if not. These four types of advice are the goals of the state space.

This expert system was developed using "VP-Expert"; a low-level development system for expert systems.

II. DESIGN METHODOLOGY

The process of deciding on advice to give to a caller can be represented as a state space containing the four advice goals previously listed. Since goal #1 is associated with a high-priority emergency condition, it has been made the conclusion of the first rule (rule #1) of the expert system [See Appendix A]. The premise of rule #1, (condition = emergency), then becomes a subgoal in working memory which is then pattern matched with the conclusion of rule #5. The premise of rule #5 then results in new subgoals. The state space is therefore searched by backward chaining in a depth-first manner until a subgoal matches a fact in working memory. For goal #1, these facts (primitives) are obtained when the user selects the emergency_symptoms from the monitor (i.e., chest_pain, shortness_of_breath, dizzyness, numbness_in_limbs, and/or excessive_bleeding). If the required combination of these inputs is not selected by the user, rule #1 fails, the quest to support goal #1 is abandoned, and goal #2 becomes the new focus of the search. If the required combinations of primitives does exist, (i.e., was input by user), the search halts, advice is given, and no further questions are asked.

The gathering of information only when it is needed has been accomplished by the proper ordering of rules. This was required to prevent the following situations from occurring:

(i) Time being wasted if an emergency exists
(ii) Trying to establish if the caller is sick, when it is already apparent from previous answers.
(iii) Trying to establish if the doctor is in when the caller should just go to the hospital regardless.
(iv) Trying to establish if the doctor is in when the caller is obviously not sick.
(v) Trying to establish if the doctor is in from the present time of day when it is obvious that he will not be in on that day. (i.e., Saturday or Sunday.)

A major concern in developing this system was the consequence of determining that the caller is not sick. If goal #2 (just take a walk) is concluded, the caller could possibly claim that the doctor misdiagnosed him because his symptoms were not accepted as valid by the expert system. If the symptom is not a choice for the user, the expert system unfortunately will always assume that there is nothing wrong with the caller. For this reason the selection of non_emergency_symptoms should be made relatively large and somewhat vague. A selection of eighteen non_emergency_symptoms is presented to the user in this expert system. Also, since the primitives required in supporting goal #1 involve "and" expressions, these "and" expressions will fail if only one of the two required emergency_symptoms is selected by the user. However, the emergency_symptom that was selected must be remembered and considered when determing whether or not the caller is sick (for goals #2,3, and 4). This is accomplished in rule #9.

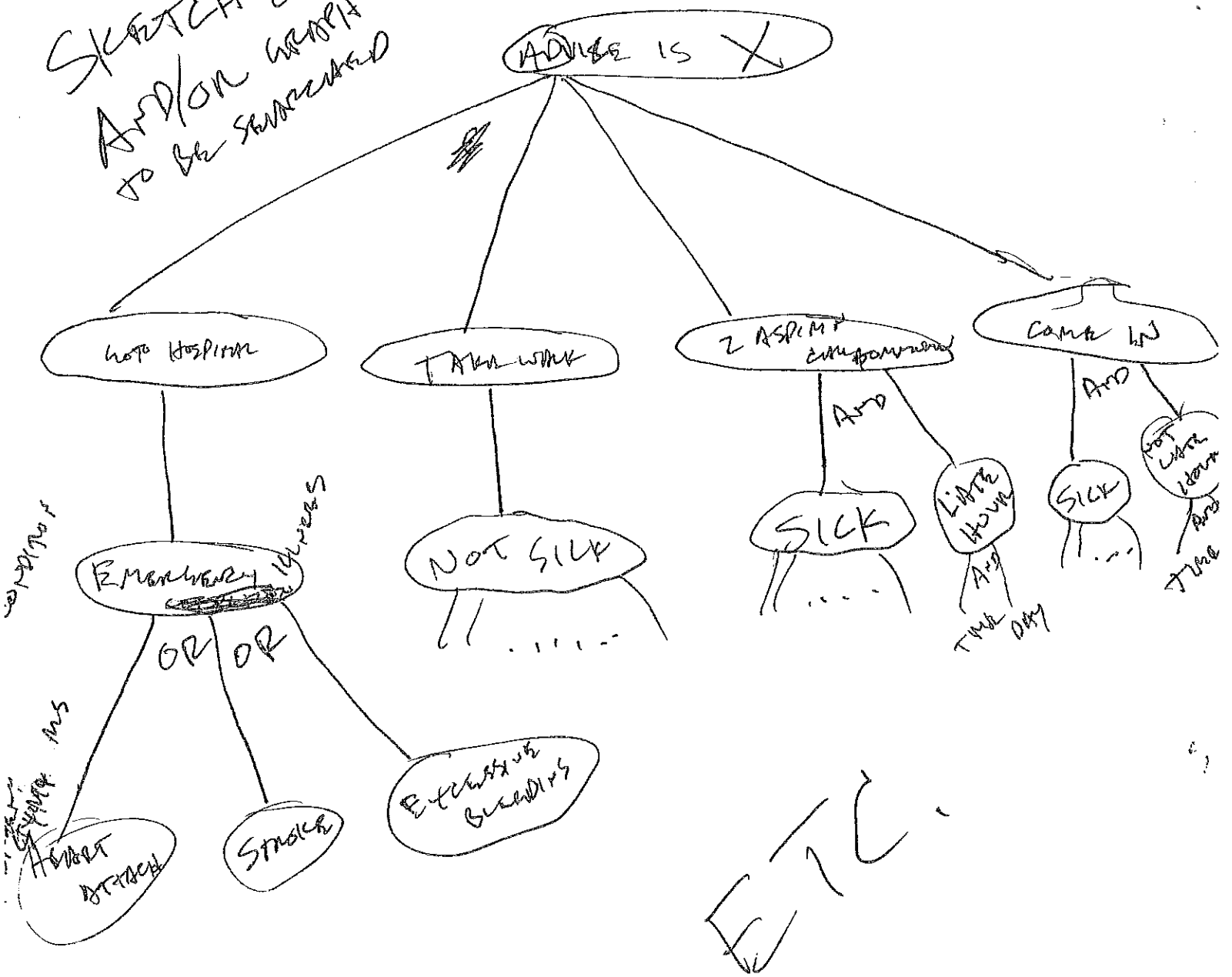
CONCLUSIONS

The choice of a backward-chaining (goal-driven) representation for this state space is a natural selection because of the nature of the problem, (i.e., four possible goals to be supported or not).

The testing of the expert system proved that the ordering of the rules established priorities and provided the caller with advice as quickly as possible and with the minimum number of questions asked for each situation.

Before this type of expert system were to be used in a doctor's office, it would need to be significantly expanded. The need for the system to consider all possible symptoms that constitute an "emergency" or the caller being "sick" must be carefully considered if the user is expected to rely solely on the system to give advice.

APPENDIX A


SKETCH OF AND/OR GRAPH TO BE SEARCHED
ADVICE IS X
GOTO HOSPITAL
TAKE WORK
2 ASPIRIN
CALL LJ
AND
AND
EMERGENCY
NOT SICK
SICK
LITTLE HOUR
SICK
OR
OR
AND
AND
ETC.

APPENDIX B

RULES * SHOW TREE IN MAIN MENU
→ SELECT 3 CHOICES

MAKE TREE FOR HW

```
ACTIONS
FIND advice;

RULE 1
IF condition = emergency
THEN advice = go to hospital
DISPLAY "advise caller to go to hospital"
DISPLAY "notify hospital of arriving patient with possible {emergency_illness}
        and claiming {emergency_symptom} symptoms";

RULE 2
IF condition = not_sick
THEN advice = take walk
DISPLAY "advise caller to take a nice refreshing walk";

RULE 3
IF condition = sick AND
hour = late
THEN advice = two aspirin call back
DISPLAY "advise caller to take two aspirin and call back in the morning";

RULE 4
IF condition = sick AND
hour = not_late
THEN advice = come in
Display "advise caller to come into office";


RULE 5
IF emergency_illness = heart_attack OR
emergency_illness = stroke OR
emergency_illness = extensive_bleeding
THEN condition = emergency;

RULE 6
IF emergency_symptom = chest_pain AND
emergency_symptom = shortness_of_breath
THEN emergency_illness = heart_attack;

RULE 7
IF emergency_symptom = dizzyness AND
emergency_symptom = numbness_in_limbs
THEN emergency_illness = stroke;

RULE 8
IF emergency_symptom = extensive_bleeding
THEN emergency_illness = extensive_bleeding;


RULE 9
IF emergency_symptom = chest_pain OR
emergency_symptom = shortness_of_breath OR
emergency_symptom = dizzyness OR
emergency_symptom = numbness_in_limbs OR
non_emergency_symptom = minor_bleeding OR
non_emergency_symptom = fever OR
```

```
non_emergency_symptom = headache OR
non_emergency_symptom = soar_throat OR
non_emergency_symptom = coughing OR
non_emergency_symptom = sneezing OR
non_emergency_symptom = stomach_pain OR
non_emergency_symptom = vomitting OR
non_emergency_symptom = itching OR
non_emergency_symptom = joint_pain OR
non_emergency_symptom = muscle_pain OR
non_emergency_symptom = loss_of_vision OR
non_emergency_symptom = loss_of_hearing OR
non_emergency_symptom = loss_of_balance OR
non_emergency_symptom = gynocological_problem OR
non_emergency_symptom = bowel_trouble OR
non_emergency_symptom = hypertension OR
non_emergency_symptom = insomnia
THEN condition = sick
ELSE condition = not_sick;

RULE 10
IF day = saturday_or_sunday
THEN hour = late;

RULE 11
IF time = nine_am_to_two_pm OR
time = two_pm_to_four_pm AND
day = monday_through_thursday
THEN hour = not_late;

RULE 12
IF day = monday_through_thursday AND
time = four_pm_to_nine_pm
THEN hour = late;

RULE 13
IF day = friday AND
time = nine_am_to_two_pm
THEN hour = not_late;

RULE 14
IF day = friday AND
time = two_pm_to_four_pm OR
time = four_pm_to_nine_pm
THEN hour = late;
```

Q5.

```
ASK emergency_symptom : "select which of the following symptoms caller has";
CHOICES emergency_symptom : chest_pain, shortness_of_breath, dizzyness,
                            numbness_in_limbs, extensive_bleeding, none_of_these;

ASK non_emergency_symptom : "select which of the following symptoms caller has";
CHOICES non_emergency_symptom : minor_bleeding, fever, headache, soar_throat,
coughing, sneezing, stomach_pain, vomitting, itching, joint_pain, muscle_pain,
loss_of_vision, loss_of_hearing, loss_of_balance, gynocological_problem,
bowel_trouble, hypertension, insomnia, none_of_these;

ASK day : "select day of week";
CHOICES day : monday_through_thursday, friday, saturday_or_sunday;

ASK time : "select present time period";
CHOICES time : nine_am_to_two_pm, two_pm_to_four_pm, four_pm_to_nine_pm;


PLURAL: emergency_symptom, non_emergency_symptom;
```

TESTING OF EXPERT SYSTEM

The expert system was tested under four scenarios (one for each goal). Appendix B contains the test traces for each of these tests.

Test #1) INPUT: Emergency_symptom = chest_pain, shortness_of_breath
OUTPUT: Advice = go_to_hospital

Test #2) INPUT: Emergency_symptom = none_of_these
INPUT: Non_emergency_symptoms = none_of_these
OUTPUT: Advice = take_walk

Test #3) INPUT: Emergency_symptom = none_of_these
INPUT: Non_emergency_symptoms = fever, headache, soar_throat
INPUT: Day = Saturday_or_Sunday
OUTPUT: Advice = two_aspirin_call_back

Test #4) INPUT: Emergency_symptom = none_of_these
INPUT: Non_emergency_symptoms = fever, headache, soar_throat
INPUT: Day = Monday_through_Thursday
INPUT: Time = nine_A.M._to_two_P.M.
OUTPUT: Advice = come_in

TRACE #3

```
Testing b:\EXP1.kbs
(= yes CNF 0 )
!   advice
!   !   Testing 1
!   !   !   condition
!   !   !   !   Testing 5
!   !   !   !   !   emergency_illness
!   !   !   !   !   !   Testing 6
!   !   !   !   !   !   !   emergency_symptom
!   !   !   !   !   !   !   !   (= none_of_these CNF 100 )
!   !   !   !   !   !   Testing 7
!   !   !   !   !   !   Testing 8
!   !   !   !   Testing 9
!   !   !   !   !   non_emergency_symptom
!   !   !   !   !   !   (= fever CNF 100 )
!   !   !   !   !   !   (= headache CNF 100 )
!   !   !   !   !   !   (= soar_throat CNF 100 )
!   !   !   !   (= sick CNF 100 )
!   !   Testing 2
!   !   Testing 3
!   !   !   hour
!   !   !   !   Testing 10
!   !   !   !   !   day
!   !   !   !   !   !   (= saturday_or_sunday CNF 100 )
!   !   !   !   (= late CNF 100 )
!   !   (= two_aspirin_call_back CNF 100 )
```

TRACE #1

```
Testing b:\EXP1.kbs
(= yes CNF 0 )
!   advice
!   !   Testing 1
!   !   !   condition
!   !   !   !   Testing 5
!   !   !   !   !   emergency_illness
!   !   !   !   !   !   Testing 6
!   !   !   !   !   !   !   emergency_symptom
!   !   !   !   !   !   !   !   (= chest_pain CNF 100 )
!   !   !   !   !   !   !   !   (= shortness_of_breath CNF 100 )
!   !   !   !   !   !   !   (= heart_attack CNF 100 )
!   !   !   !   (= emergency CNF 100 )
!   !   (= go_to_hospital CNF 100 )
```

DEFAULT CERTAINTY = 100%

BACKWARD CHAIN OF PROOFS

TRACE #2

```
Testing b:\EXP1.kbs
(= yes CNF 0 )
!   advice
!   !   Testing 1
!   !   !   condition
!   !   !   !   Testing 5
!   !   !   !   !   emergency_illness
!   !   !   !   !   !   Testing 6
!   !   !   !   !   !   !   emergency_symptom
!   !   !   !   !   !   !   !   (= none_of_these CNF 100 )
!   !   !   !   !   !   Testing 7
!   !   !   !   !   !   Testing 8
!   !   !   !   Testing 9
!   !   !   !   !   non_emergency_symptom
!   !   !   !   !   !   (= none_of_these CNF 100 )
!   !   !   !   (= not_sick CNF 100 )
!   !   Testing 2
!   !   (= take_walk CNF 100 )
```

TRACE #4

```
Testing b:\EXP1.kbs
(= yes CNF 0 )
!   advice
!   !   Testing 1
!   !   !   condition
!   !   !   !   Testing 5
!   !   !   !   !   emergency_illness
!   !   !   !   !   !   Testing 6
!   !   !   !   !   !   !   emergency_symptom
!   !   !   !   !   !   !   !   (= none_of_these CNF 100 )
!   !   !   !   !   !   Testing 7
!   !   !   !   !   !   Testing 8
!   !   !   !   Testing 9
!   !   !   !   !   non_emergency_symptom
!   !   !   !   !   !   (= fever CNF 100 )
!   !   !   !   !   !   (= headache CNF 100 )
!   !   !   !   !   !   (= soar_throat CNF 100 )
!   !   !   !   (= sick CNF 100 )
!   !   Testing 2
!   !   Testing 3
!   !   !   hour
!   !   !   !   Testing 10
!   !   !   !   !   day
!   !   !   !   !   !   (= monday_through_thurs CNF 100 )
!   !   !   !   Testing 11
!   !   !   !   !   time
!   !   !   !   !   !   (= nine_am_to_two_pm CNF 100 )
!   !   !   !   (= not_late CNF 100 )
!   !   Testing 4
!   !   (= come_in CNF 100 )
```