CS161 Spr'13 HW-4 **Due**: 11:59 PM Wed. June 5

When people are given a question, they must first understand the *meaning* of the question. They use this *question concept* to search memory for a *conceptual answer*. Once they have retrieved a conceptual answer, they have to express it in a natural language (which, in our case, will be as an English sentence).

For this homework you will write two Lisp functions (& utility fcn, see last page).

- The function **ANSWER** will take a *question concept* (QCON, a frame), search some episodic story memory EP-STMEM (a list of frames), and return a concept (frame) that is a *conceptual answer* (C-ANS) to the question concept.

- The function **C-GEN** will take this conceptual answer and express it as an English sentence (as a list of atoms) via the use of *decision trees* (D-TREEs) and *English sentence patterns* (ENG-PATS).

We will assume that some conceptual lexicon (along with associated demons) already exists and that these (magical) demons have already turned each English question into its corresponding question concept **QCON**.

1. (ANSWER QCON EP-STMEM) – **ANSWER** searches an episodic story memory EP-STMEM for a CONcept that will UNIFRAME with QCON and returns a conceptual answer C-ANS.

2. (C-GEN C-ANS ENG-PATS D-TREES) – C-GEN takes a conceptual answer C-ANS (such as one returned by ANSWER), and some English patterns ENG-PATs and decision trees D-TREES which together describe how to convert frames into English text. The output is a list of atoms, readable as an English sentence.

Once written, you will test ANSWER and C-GEN on these Q/A pairs:

QA-Test-case-1:
Q2a: What did Sally tell Freddy?
A2a: Sally told Freddy that becoming friends with Hank could lead to problems for Freddy because Hank stole a car.

We will assume that (magic) demons have turned Q2a into this question concept:
Q2ACON =

(MTRANS (**AGENT** (HUMAN (**F-NAME** (SALLY))))
(**OBJECT** (V WHAT))
(**RECIP** (HUMAN (**F-NAME** (FREDDY)))))
(ANSWER Q2ACON EP-STMEM) should return the following answer concept:

A2ACON =
(MTRANS (**AGENT** (HUMAN (**F-NAME** (SALLY))))
(**RECIP** (HUMAN (**F-NAME** (FREDDY))))
(**OBJECT** (C-CAUSE
(**ANTE** (ST-CHANGE
(**TO** (FRIENDS))
(**WITH** (HUMAN (**F-NAME** (HANK))))))
(**CONSEQ**
(GOAL-FAILURE (**AGENT** (HUMAN (**F-NAME** (FREDDY))))))
(**JUSTIF** (STEAL (**AGENT** (HUMAN (**F-NAME** (HANK))))
(**OBJECT** (VEHICLE (**REF** (INDEF)))))))))

Note: ANSWER returns the MTRANS frame above (not the atom A2ACON).

(C-GEN A2ACON ENG-PATS D-TREES) should return this English answer:

(SALLY TOLD FREDDY THAT BECOMING FRIENDS WITH HANK COULD LEAD TO PROBLEMS FOR FREDDY BECAUSE HANK STOLE A CAR)

QA-Test-Case-2:
Q5a: What did Hank do at the Seven-Eleven?
A5a: Hank robbed the store owner Harold of cash using a gun.

Magical demons fire and create the question concept:
Q5ACON =
(ACT (**IS** (V WHAT))
(**AGENT** (HUMAN (**F-NAME** (HANK))))
(**LOC** (SEVN-ELEVN)))

(ANSWER Q5ACON EP-STMEM) should return the following answer concept:

A5ACON =
(ROB (**RECIP** (HUMAN (**F-NAME** (HAROLD))
(**OWN** (STORE (**REF** (DEF))))))

(**AGENT** (HUMAN (**F-NAME** (HANK))))
(**OBJECT** (CASH))
(**INSTRU** (GUN (**REF** (INDEF))))
(**LOC** (SEVN-ELEVN)))

(C-GEN A5ACON ENG-PATS D-TREES) should return this English answer:

(HANK ROBBED THE STORE OWNER HAROLD OF CASH USING A GUN)

Test-Case-3:
Q18a: Why was Harold angry?
A18a: Harold was very angry because Hank robbed the store owner Harold of cash using a gun.

Magical demons fire and create this question concept:
Q18ACON =
(CAUSE (**ANTE** (V WHAT))
(**CONSEQ** (EMOTION (**AGENT** (HUMAN (**F-NAME** (HAROLD))))
(**IS** (ANGER)))))

(ANSWER Q18ACON EP-STMEM) should return the following answer concept:
A18ACON =
(CAUSE (**ANTE** (ROB (**RECIP** (HUMAN (**F-NAME** (HAROLD))
(**OWN** (STORE (**REF** (DEF))))))
(**AGENT** (HUMAN (**F-NAME** (HANK))))
(**OBJECT** (CASH))
(**INSTRU** (GUN (**REF** (INDEF))))))
(**CONSEQ** (EMOTION (**IS** (ANGER))
(**AGENT** (HUMAN (**F-NAME** (HAROLD))))
(**VAL** (>NORM)))))

(C-GEN A18ACON ENG-PATS D-TREES) should return this English answer:

(HAROLD WAS VERY ANGRY BECAUSE HANK ROBBED THE STORE OWNER HAROLD OF CASH USING A GUN)
To further test C-GEN you will try it also on the concepts LAP-1, DRINK-2, and SMOKE-3 which are unrelated to the story, but test the selection of words like "handsome" vs. "pretty" and "drink" vs. "eat" vs. "lap up" vs. "smoke".
LAP-1 =
(MTRANS (**AGENT** (HUMAN (**REF** (DEF))
(**APPEARANCE** (>NORM))

(**GENDER** (MALE))))
(**RECIP** (HUMAN (**REF** (INDEF))
(**APPEARANCE** (<NORM))
(**GENDER** (FEMALE))))
(**OBJECT** (INGEST (**AGENT** (CANINE (**REF** (DEF))))
(**OBJECT** (LIQUID (**IS** (COKE))
(**REF** (INDEF)))))))

(C-GEN LAP-1 ENG-PATS D-TREES) should return the English:

(THE HANDSOME MALE TOLD A HOMELY FEMALE THAT THE DOG LAPPED UP A COKE)

EAT-2 =
(MTRANS (**AGENT** (HUMAN (**REF** (DEF))
(**APPEARANCE** (<NORM))
(**GENDER** (MALE))))
(**RECIP** (HUMAN (**REF** (INDEF))
(**APPEARANCE** (>NORM))
(**GENDER** (FEMALE))))
(**OBJECT** (INGEST (**AGENT** (HUMAN (**F-NAME** (SAMUEL))))
(**OBJECT** (FOOD (**IS** (DONUTS)))))))

(C-GEN EAT-2 ENG-PATS D-TREES) should return the English:

(THE UGLY MALE TOLD A PRETTY FEMALE THAT SAMUEL ATE DONUTS)

SMOKE-3 =
(INGEST (**AGENT** (HUMAN (**F-NAME** (SAMUEL))))
(**OBJECT** (GAS))
(**DESTIN** (LUNGS))
(**LOC** (SCHOOL)))

(C-GEN SMOKE-3 ENG-PATS D-TREES) should return the English:

(SAMUEL SMOKED AT SCHOOL)

(Note: to keep life simple, we have assumed that every sentence will be generated in past tense.)

First, let us first look more closely at ANSWER. ANSWER should be pretty easy to implement, with the exception of a special case when a predicate is NOT supplied in the question concept QCON (as is the case with question Q5a, above under Test-Case-2). Under the normal case, ANSWER will go through EP-STMEM (which is simply a list, see below) and attempt to UNIFRAME its QCON with one of the concepts. The first concept that unifies is the C-ANS (the conceptual answer). ANSWER will then substitute any bindings feta (due to the unification), substitute those bindings into C-ANS and then return C-ANS as its conceptual answer. You have already written UNIFRAME and SUBST-FETA so ANSWER should be relatively easy to implement. Examples of using ANSWER were given above.

To deal with the special case where the predicate is not supplied, ANSWER must first check to see if QCON has ACT as a predicate and also if there is a slot named IS with a variable as its filler. In that case, ANSWER will not call UNIFRAME directly. Instead, ANSWER will use UNIFRAME to attempt to unify all remaining slots (in its QCON) with the slots of any concepts in EP-STMEM. The first concept found whose slots unify is the C-ANS.

Here is the episodic memory you will use to test ANSWER (It only has 3 concepts in it, which are the conceptual answers in our test cases):

EP-STMEM = (A2ACON A5ACON A18ACON)

Now let us examine C-GEN more closely. C-GEN uses English patterns-of-expression. These patterns are associated with frame predicates.
Each pattern consists of a list of zero or more slots, phrases, and/or phrase-selection decisions. The phrase-selection decisions are made by accessing a list of decision-trees (D-TREES).

As C-GEN goes through a frame, it accesses a pattern for each predicate (from ENG-PATS). It then calls upon itself recursively to generate each element in that pattern, as follows:

- If the element is a SLOT, then C-GEN recursively generates the filler of that slot.
- If the element is (**DECIDE** X) then it searches in D-TREES and uses the decision-tree whose predicate = X to decide what phrase to generate for X.
- If the element is a (**PHR** X), then C-GEN simply appends the phrase X to the sentence it has generated so far.

- If C-GEN cannot find a pattern for a pred, then it appends (LIST pred) to the sentence that it is building up.

Note: C-GEN will APPEND each phase to the end of the sentence it has so far. So if any recursive call to C-GEN returns NIL, APPENDing that NIL to the sentence built up so-far will cause no change (which is what we want for the case of NIL). That is (APPEND '(A B C) NIL) returns: (A B C)

Let's examine ENG-PATS. This is a list of frame predicates and associated English-expression patterns. In each pattern, each atom is assumed to be the name of a slot unless it is marked as a **PHR**ase or decision (**DECIDE**). ENG-PATS has the form
((pred (element)*))
where element is either

- An atom (interpreted as a slot name)
- A list of the form (**PHR** (atom)+) (the literal symbol PHR followed by a list of atoms, interpreted as words to append to the sentence)
- A list of the form (**DECIDE** PRED), in which case the utility function DECIDE-PHR will be dispatched to use the entry for PRED in D-TREES to decide which words to add (see specs for DECIDE-PHR)

ENG-PATS =
(
(MTRANS (AGENT (**PHR** (TOLD)) RECIP (**PHR** (THAT) OBJECT))

(ROB (AGENT (**PHR** (robbed) RECIP (**PHR** (OF)) OBJECT (**PHR** (USING)) INSTRU))

(STEAL (AGENT (**PHR** (STOLE)) OBJECT))

(C-CAUSE (ANTE (**PHR** (COULD LEAD TO)) CONSEQ (**PHR** (BECAUSE)) JUSTIF))

(ST-CHANGE ((**PHR** (BECOMING)) TO (**PHR** (WITH)) WITH))

(GOAL-FAILURE ((**PHR** (PROBLEMS FOR)) AGENT))

(CAUSE (CONSEQ (**PHR** (BECAUSE)) ANTE))

(EMOTION (AGENT (**PHR** (WAS)) VAL IS)

(ANGER ((**PHR** (ANGRY))))

(>NORM ((**PHR** (VERY))))

(HUMAN (REF (**DECIDE** HUMAN) GENDER OWN F-NAME))

(STORE (REF (**PHR** (STORE OWNER))))

(CANINE (REF (**PHR** (DOG))))

(GUN (REF (**PHR** (GUN))))

(INGEST (AGENT (**DECIDE** INGEST) OBJECT LOC))

(COKE (REF (**PHR** (COKE))))

(CANINE (REF (**PHR** (DOG))))

(FOOD (IS))

(SCHOOL ((**PHR** (AT SCHOOL))))

(VEHICLE (REF (**PHR** (CAR))))

(INDEF ((**PHR** (A))))

(DEF ((**PHR** (THE))))

(LIQUID (REF IS))

(GAS ())
)

Consider the first pattern in ENG-PATS. The pattern is:
(AGENT (**PHR** (TOLD)) RECIP (**PHR** (THAT)) OBJECT))
and is associated with the frame predicate MTRANS. The pattern states that, to express MTRANS as a sentence in English, you first C-GEN the AGENT slot, then append (TOLD) to the end of the sentence, then recursively C-GEN the RECIP slot, then append (THAT), and, finally, you C-GEN the OBJECT slot.

Decision Trees: We will be covering decision trees (D-Trees) in chapter 18.3 (starts at p. 697 of your textbook), along with the learning of D-trees from

examples. However, in this homework we will not learn D-trees but simply use them in order to select English words/phrases.

If you look at your textbook you will see that D-trees consist of attributes (on the nodes) with their values being the links. A D-tree returns a leaf as its decision.

The decisions (leaves) or our D-trees will be English phrases used to express some part of a concept. D-TREES will be a list containing 2 d-trees and will encode the following knowledge of English expression:

1. In English, good-looking males are termed “handsome” while good-looking females are not termed “handsome” but are referred to as “pretty”, etc. Unattractive males are called “ugly” while unattractive females are termed “homely”.

2. In English, we use the verb “drink” for ingesting liquids and “eat” for ingesting foods, but if it is a dog doing the ingesting of a liquid then “lap up” is used and if gas is ingested by a human into the lungs then the term “smoking” is used.

Each decision-tree in D-TREES is associated with a predicate. Thus, D-TREES has the following form:

((pred-1 (***node*** (val-1-1 (leaf or ***node***…))
… …
(val-1-n (leaf or ***node***…))))
(pred-2 (***node*** (val-2-1 (leaf or ***node***...)) …)

Note: The nodes of each D-tree are marked below using ***bold italics***.
Note: Although each D-tree may have several levels, the tests that are made at each level are all testing slots (in a concept) which are at the top level.

D-TREES =
((HUMAN
(***GENDER***
(MALE (***APPEARANCE***
(>NORM (PHR (HANDSOME)))
(<NORM (PHR (UGLY)))))
(FEMALE (***APPEARANCE***
(>NORM (PHR (PRETTY)))
(<NORM (PHR (HOMELY)))))))

(INGEST (***OBJECT***
(LIQUID (***AGENT*** (CANINE (PHR (LAPPED UP)))
(HUMAN (PHR (DRANK)))))
(FOOD (***AGENT*** (HUMAN (PHR (ATE)))))
(GAS (***AGENT*** (HUMAN
(***DESTIN*** (LUNGS (PHR (SMOKED)))))))))

)

You should write a utility function, DECIDE-PHR, that will be used by C-GEN whenever a pattern-of-expression contains a (DECIDE PRED) element.

3. (DECIDE-PHR PRED FRAME D-TREES) first finds a D-tree in D-TREES whose predicate matches **PRED**. DECIDE-PHR then accesses the first **node** in the D-tree and looks for a top-level **slot** in **FRAME** where **slot** equals **node**. DECIDE-PHR then accesses the **pred** of the filler of **slot** and looks for a **value** of that **node** in the D-tree where (CAR **value**) equals the slot's predicate. If this value is found and is of the form (PHR x) then DECIDE-PHR returns x. If it is not (PHR x) then it must be a sub D-tree (with its own nodes and values, recursively). In this case, DECIDE-PHR operates recursively, until either a PHR is found, or else NIL is returned. NOTE: Although the D-tree can have nodes with sub-nodes etc. the slots that are being examined are always at the *top-level* of the FRAME.

Test your utility function on the following:

HM-1 =
(HUMAN (**REF** (DEF))
(**APPEARANCE** (>NORM))
(**GENDER** (MALE)))

HM-2 = (HUMAN (**REF** (INDEF))
(**APPEARANCE** (<NORM))
(**GENDER** (FEMALE))))

(DECIDE-PHR 'HUMAN HM-1 D-TREES) returns: (HANDSOME)
(DECIDE-PHR 'HUMAN HM-2 D-TREES) returns: (HOMELY)

SAMUEL-1 =
(INGEST (**AGENT** (HUMAN (**F-NAME** (SAMUEL))))

(**OBJECT** (FOOD (**IS** (DONUTS)))))

SMOKE-3 =
(INGEST (**AGENT** (HUMAN (**F-NAME** (SAMUEL))))
(**OBJECT** (GAS))
(**DESTIN** (LUNGS))
(**LOC** (SCHOOL)))

(DECIDE-PHR ‘INGEST SAMUEL-1 D-TREES) returns: (ATE)

(DECIDE-PHR ‘INGEST SMOKE-3 D-TREES) returns: (SMOKED)

This is the last homework!