

### Homework 3

#### CS161, Spring 2014

**Due:** Wednesday, May 14th by midnight. Submit via CourseWeb.

---

In this homework you will be specifying the ADD-LEX entries for reading the 9 sentences making up the Merry Widow Murderer (MWM) story. These entries will require you to define some new demon functions. But first you will define two new utility functions and modify two utility functions (from HW2).

Before we discuss the new assignment, we'll first introduce a new global variable: ISAMEM. This variable holds simple, taxonomic information in the form of ISA binary relationships between atoms. Here is its value:

ISAMEM =

```
(
  (ISA HOME LOCATION)
  (ISA HOME BUILDING)
  (ISA BUILDING INANIMATE)
  (ISA KILL VIOLENT-ACT)
  (ISA VIOLENT-ACT ACT)
  (ISA READ VISUAL-ACT)
  (ISA SEEK VISUAL-ACT)
  (ISA VISUAL-ACT ACT)
  (ISA EAT ACT)
  (ISA CO-HABITATE ACT)
  (ISA HUMAN ANIMATE)
  (ISA RING INANIMATE)
  (ISA PRINTED-MATTER PHYS-OBJ)
  (ISA ANIMATE PHYS-OBJ)
  (ISA INANIMATE PHYS-OBJ)
  (ISA RING WEAR-OBJ)
  (ISA WEAR-OJB PHYS-OBJ)
)
```

The purpose of this homework, then, will be to define enough demons that perform specific ambiguity-reducing tasks such that we can successfully "understand" the concepts implicit within the MWM sentences.

Having extracted these concepts, we will later use them to ask questions about the scenario.

**Problem 1: (ISA atm1 atm2 isam)**

This utility function returns T if atom ATM1 recursively ISA ATM2 when using the ISA memory parameter **isam**; else ISA returns NIL.

Take, for example, the ISAMEM global we listed above (replicated here for clarity):

ISAMEM =

```
(
  (ISA HOME LOCATION)
  (ISA HOME BUILDING)
  (ISA BUILDING INANIMATE)
  (ISA KILL VIOLENT-ACT)
  (ISA VIOLENT-ACT ACT)
  (ISA READ VISUAL-ACT)
  (ISA SEEK VISUAL-ACT)
  (ISA VISUAL-ACT ACT)
  (ISA EAT ACT)
  (ISA CO-HABITATE ACT)
  (ISA HUMAN ANIMATE)
  (ISA RING INANIMATE)
  (ISA PRINTED-MATTER PHYS-OBJ)
  (ISA ANIMATE PHYS-OBJ)
  (ISA INANIMATE PHYS-OBJ)
  (ISA RING WEAR-OBJ)
  (ISA WEAR-OJB PHYS-OBJ)
)
```

*Examples:*

```
(ISA 'HOME 'HOME ISAMEM) returns: T
(ISA 'HOME 'BUILDING ISAMEM) returns: T
(ISA 'HOME 'INANIMATE ISAMEM) returns: T
(ISA 'READ 'ACT ISAMEM) returns: T
(ISA 'HUMAN 'BUILDING ISAMEM) returns: NIL
(ISA 'LOCATION 'INANIMATE ISAMEM) returns: NIL
(ISA 'KILL 'ACT ISAMEM) returns: T
(ISA 'RING 'PHYS-OBJ ISAMEM) returns: T
```

*Note:*

Now that we have the utility function ISA, demons can use it. For example, DM-EXP can look for an ACT and if it finds CO-HABITATE, check if CO-HABITATE recursively isa ACT in ISAMEM. (see problems below for means of accomplishing this)

*Preemptive FAQ:*

**Q: What does it mean when it says this function "recursively" asks if atm1 IS atm2?**

Take the above example:

(ISA 'KILL 'ACT ISAMEM) returns T

Even though 'KILL does not map *directly* to 'ACT, since 'KILL *does* directly map to VIOLENT-ACT, we'll call ISA recursively and return true if VIOLENT-ACT ISA ACT.

As it turns out, it is, so we return true.

**Q: Why does (ISA 'HOME 'HOME ISAMEM) return T?**

On top of its search of the **isam** parameter for a matching ontological definition, **ISA** returns T if (equal atm1 atm2) returns T.

## Problem 2: (INSERT-SL slot filler conatm)

---

This utility function is essentially a tweaking of ADD-SF from homework 1 that performs the following with two slight additions: INSERT-SL inserts a **[slot filler]** pair at the top-level of the frame associated with **conatm**, or replaces the filler currently associated with **slot** if it already exists in the given frame. It returns the value of **conatm** (i.e., the frame it represents post-insertion) and, as a side-effect, adds **conatm** to the global variable USEDMEM. Additionally, INSERT-SL should be empowered to insert into a frame that is nested below two or more gaps: e.g., GAP1 = GAP2; GAP2 = (FRAME SLA SLOTA)...

*Example:*

USEDMEM = NIL

CON1 =

(HUMAN **F-NAME** (CHARLOTTE)  
          **L-NAME** (NEWTON)  
          **GENDER** (FEMALE))

(INSERT-SL '**AGE**' (RANGE **FROM** (13) **TO** (19) **UNIT** (YEAR)) 'CON1)

returns:

(HUMAN **F-NAME** (CHARLOTTE)  
          **L-NAME** (NEWTON)  
          **GENDER** (FEMALE)  
          **AGE** (RANGE **FROM** (13) **TO** (19) **UNIT** (YEAR)))

...and as a side-effect:

USEDMEM = (CON1)

### Problem 3: (SRCH atmlst myatm dir pred)

---

For this problem, you will modify SRCH from homework 2 to, instead of searching for a CON atom with predicate **pred**, will search for a CON atom whose predicate **ISA pred** (using ISAMEM as the ontology parameter to **ISA**), starting with **myatm** in **atmlst** in direction **dir**. This is a utility function that is used by some demons. If it finds a CON atom in atmlst whose predicate **ISA pred**, then it will return that CON atom; otherwise it returns NIL.

DIR can have one of the following 4 directions:

- *AFT* -- closest after (to the right of) **myatm**
- *IM-AFT* -- immediately after **myatm**
- *BEF* -- closest before (to the left of) **myatm**
- *IM-BEF* -- immediately before **myatm**

*Examples:*

ATMLST1 = (CON1 CON2 CON3 CON4 CON5 CON6)

where:

CON1 = (RING **VALUE** (>NORM) **MATERIAL** (GOLD))

CON2 = (INANIMATE **OBJECT** OBJ)

CON3 = (ACT **AGENT** (HUMAN H1) **DESIRE** D1)

CON4 = (UNKNOWN **WORD** (NEWTON))

CON5 = (EAT **AGENT** (HUMAN L-NAME (NEWTON)) **OBJECT** (WEINER))

CON6 = (SEEK **AGENT** (HUMAN L-NAME (NEWTON)) **OBJECT** (RING))

(SRCH ATMLST1 'CON2 'BEF 'RING) returns CON1

(SRCH ATMLST1 'CON2 'AFT 'RING) returns NIL

(SRCH ATMLST1 'CON3 'AFT 'ACT) returns CON5 (because EAT **ISA** ACT)

(SRCH ATMLST1 'CON3 'AFT 'SEEK) returns CON6

(SRCH ATMLST1 'CON3 'IM-AFT 'SEEK) returns NIL

*Note:*

After making this modification, the behavior of DM-EXP and any other demons that use SRCH will now correctly consult the ISA ontology for their search queries.

**Problem 4: (TOP-CON wkm used)**

---

Modify TOP-CON from HW2 so that it returns a list of CON atoms in **wkm** such that (a) they do not appear in **used** and (b) their value is not NIL.

*Example:*

WKM1 = (CON1 CON2 CON3 CON4)

USED1 = (CON3 CON1)

CON1 = (HUMAN)

CON2 = (EAT)

CON3 = (FOOD)

CON4 = ( )

(TOP-CON WKM1 USED1) returns: (CON2)

**Problem 5: (C-ANALYZER sent lexic)**

---

Modify **C-ANALYZER** from HW2 such that, along with returning the UNGAPPED TOP-CON of WKMEM and USEDMEM, it will also set USEDMEM = WKMEM upon returning.

*Example:*

WKMEM = (CON1 CON2 CON3)

USEDMEM = (CON1 CON3)

(C-ANALYZER '(SOME SENTENCE) LEXMEM)

...C-ANALYZER does its stuff, having returned UNGAP of CON2... and after:

WKMEM = (CON1 CON2 CON3)

USEDMEM = (CON1 CON2 CON3)

## Demon Definitions

---

The remaining functions you will define are demon functions. They will be defined below just after when they are first used in a word / phrase in the lexical memory. We will go through the words / phrases of the Merry Widow Murder (MWM) story in the order in which these words / phrases appear in the MWM story.

The definitions of the new demons you will have defined are listed next, followed by examples of how they will be used internal to the MWM story. Then, each sentence of the MWM story is broken down into its constituent phrase, frame, demon triplets, which will employ these newly defined demons and illustrate the behavior that they should elicit.

If your functions work as intended, then your tests should match the C-ANALYZER results listed at the end of every sentence analysis (also given to you for testing after the demon explanations).

### [!] REMEMBER:

All demons return (DIE) if they are successful in their task, or NIL otherwise!



### Problem 6: (DM-MODIF mycon pred dir slot filler)

Starting at **mycon**, this demon uses the SRCH function (as modified above) to look in direction **dir** for a CON atom with predicate that **ISA pred** and if *found*, uses the utility function INSERT-SL to insert pair:

[ **slot filler** ] into *found* 's frame and to add *found*'s CON atom to USEDMEM.

*Example:*

```
WKMEN = (CON1 CON2 CON3 CON4 CON5)
USEDMEM = (CON1 CON3)
CON1 = (MODIF AGE AGE7)
CON2 = (HUMAN F-NAME (CHARLOTTE))
CON3 = (UNKNOWN WORD (NEWTON))
CON4 = (AGE AMT (14) UNIT (YEAR))
CON5 = (EMOTION SENTIM (POS) SCALE (>NORM))
```

```
(DM-MODIF CON1 HUMAN AFT AGE (RANGE FROM (13)
                                         TO (19)
                                         UNIT (YEAR)))
```

...will find CON2, so as a result:

```
CON2 = (HUMAN F-NAME (CHARLOTTE)
        AGE (RANGE FROM (13)
                    TO (19)
                    UNIT (YEAR)))
```

And, as a side-effect:

```
USEDMEM = (CON1 CON3 CON2)
```

*Usage:*

```
MWM-S1 = (TEENAGER CHARLOTTE NEWTON FEELS EXCITED)
```

Phrase: (TEENAGER)

Frame: (MODIF AGE (RANGE FROM (13) TO (19) UNIT (YEAR)))

Demons:

DM-MODIF looks for HUMAN after; if found, adds the slot-filler pair:  
[AGE (RANGE FROM (13) TO (19) UNIT (YEAR))] to that HUMAN. If that HUMAN already has an AGE slot, the DM-MODIF changes whatever filler AGE has to be the new **filler** as input to DM-MODIF.

As a side-effect, the CON atom of this frame (i.e., the frame that DM-MODIF is working for, represented by **mycon**) is added to USEDMEM.

### **Problem 7: (DM-LNAME mycon myslot)**

---

This demon uses the SRCH function to look immediately after **mycon** for the predicate UNKNOWN with slot-name **WORD** and if its filler is found, it sets the gap of **myslot** to have that filler value. It also adds the CON atom of UNKNOWN to USEDMEM.

*Example:*

```
WKMED = (CON1 CON2 CON3 CON4 CON5)
USEDMEM = (CON1)
CON1 = (MODIF AGE AGE7)
CON2 = (HUMAN F-NAME (CHARLOTTE) L-NAME LNM)
CON3 = (UNKNOWN WORD (NEWTON))
CON4 = (AGE AMT (14) UNIT (YEAR))
CON5 = (EMOTION SENTIM (POS) SCALE (>NORM))
```

```
(DM-LNAME CON2 L-NAME)
```

...will find CON3, so as a result:  
LNM = (NEWTON)

And, as a side-effect:  
USEDMEM = (CON1 CON3)

*Usage:*

```
MWM-S1 = (TEENAGER CHARLOTTE NEWTON FEELS EXCITED)
```

```
Phrase: (CHARLOTTE)
Frame: (HUMAN F-NAME (CHARLOTTE)
        L-NAME LNM
        GENDER (FEMALE))
```

Demons:

DM-LNAME looks for UNKNOWN **WORD** immediately after and if found, and sets it to the gap of slot **L-NAME**

### **Problem 8: (DM-DISAMB-FEELS mycon)**

This demon attempts to disambiguate the input word. This demon uses SRCH to look immediately after MYCON for the predicate THAT and if found uses UNIQUE-GAPS (from HW2) to instantiate a frame (BELIEVE **AGENT** AG **OBJECT** OBJ) as the value of **mycon** and then uses the function SPAWN (from HW2) to spawn two DM-EXP demons (one for **AGENT**, the other for **OBJECT**) else it selects (STATE **TYPE** TYP **AGENT** AG) as the frame for the word FEELS and spawns two DM-EXP demons (one for **TYPE** and one for **AGENT**).

*Example:*

```
WKMEM = (CON1 CON2 CON3 CON4 CON5)
USEDMEM = (CON1)
CON1 = (MODIF AGE AGE7)
CON2 = (HUMAN F-NAME (CHARLOTTE))
CON3 = ( )
CON4 = (THAT)
CON5 = (EMOTION SENTIM (POS) SCALE (>NORM))
```

```
(DM-DISAMB-FEELS CON3)
```

...will find CON4, so as a result (for example, with UNIQUE-GAPS returning these illustrative gap names):

```
CON3 = (BELIEVE AGENT AG12 OBJECT OBJ13)
```

And, as a side-effect, two new demons are spawned:

```
((DM-EXP CON3 HUMAN BEF AGENT)
(DM-EXP CON3 ACT AFT OBJECT))
```

*Usage:*

MWM-S1 = (TEENAGER CHARLOTTE NEWTON **FEELS** EXCITED)

Phrase: (FEELS); *Note:* this word is ambiguous in the MWM story

Frame: ( )

Demons:

DM-DISAMB-FEELS looks for predicate THAT immediately after and if found then assigns UNIQUE-GAPS of frame:

(BELIEVE **AGENT** AG **OBJECT** OBJ) to MYCON and spawns 2 demons:

DM-EXP looking for HUMAN before

DM-EXP looking for ACT after

else DM-DISAMB-FEELS assigns UNIQUE-GAPS of frame:

(STATE **TYPE** TYP **AGENT** AG) to MYCON and spawns 2 demons:

DM-EXP looking for HUMAN before and

DM-EXP looking for EMOTION after

*Note:*

In the first example, we match the case above where predicate THAT is found immediately after **mycon**, but in the above usage example, we match the else clause where predicate THAT is NOT found immediately after **mycon**.

### **Problem 9: (DM-FNAME mycon pred dir myslot )**

Looks for **pred** in direction **dir** with a slot-name **myslot** and gets its filler (this is *found*). If found, it binds the gap of **myslot** (within the frame represented by mycon) with *found* and adds the CON atom of the frame it found to USEDMEM.

*Example:*

WKMEM = (CON1 CON2 CON3 CON4 CON5)  
USEDMEM = (CON1)  
CON1 = (MODIF **AGE** AGE7)  
CON2 = (HUMAN **F-NAME** FNM8)  
CON3 = (HUMAN **F-NAME** (EMMA))  
CON4 = (THAT)  
CON5 = (EMOTION **SENTIM** (POS) **SCALE** (>NORM))

(DM-FNAME CON2 HUMAN IM-AFT F-NAME)

...will find CON3, so as a result:  
FNM8 = (EMMA)

And, as a side-effect:  
USEDMEM = (CON1 CON3)

*Usage:*

MWM-S2 = (MYSTERIOUS UNCLE CHARLES OAKLEY WILL MOVE IN WITH **HER MOTHER** EMMA)

Phrase: (HER MOTHER)  
Frame: (HUMAN **F-NAME** FNM  
          **GENDER** (FEMALE)  
          **MOTHER-OF** MOF)

Demons:

DM-FNAME looks for HUMAN immediately after with a slot **F-NAME** and its filler (call that *found*). This demon binds gap of F-NAME (i.e. FNM) with *found* and adds the CON atom of found to USEDMEM.

DM-FINDHER looks for a HUMAN that is **GENDER** (FEMALE) before and if found, binds *found* to gap of the **MOTHER-OF** slot.

### Problem 10: (DM-FINDHER mycon pred slot filler dir myslot)

Looks for a CON atom in direction **dir** whose frame has top-level predicate **pred** with slot **slot** and filler **filler**. If found, DM-FINDHER binds *found* to the gap of **myslot**.

*Example:*

WKMEM = (CON1 CON2 CON3 CON4 CON5)

USEDMEM = (CON1)

CON1 = (MODIF **AGE** AGE7)

CON2 = (HUMAN **F-NAME** (EMMA) **GENDER** (FEMALE))

CON3 = (THAT)

CON4 = (HUMAN **MOTHER-OF** MOF)

CON5 = (EMOTION **SENTIM** (POS) **SCALE** (>NORM))

(DM-FINDHER CON4 HUMAN GENDER (FEMALE) BEF MOTHER-OF)

...will find CON2, so as a result:

MOF = CON2

And, as a side-effect (from using BIND):

USEDMEM = (CON1 CON2)

*Usage:*

MWM-S2 = (MYSTERIOUS UNCLE CHARLES OAKLEY WILL MOVE IN WITH **HER MOTHER** EMMA)

Phrase: (HER MOTHER)

Frame: (HUMAN **F-NAME** FNM

**GENDER** (FEMALE)

**MOTHER-OF** MOF)

Demons:

DM-FNAME looks for HUMAN immediately after with a slot **F-NAME** and its filler (call that *found*). This demon binds gap of F-NAME (i.e. FNM) with *found* and adds the CONatom of found to USEDMEM.

DM-FINDHER looks for a HUMAN that is **GENDER** (FEMALE) before and if found, binds *found* to gap of the **MOTHER-OF** slot.

*Note:*

The word "her" is ambiguous. It could be a person ("I was her."). It could be ownership of something ("I saw her ring."). It could be family relationship ("I saw her mother."). In this project you are NOT going to disambiguate it, so we have cheated here and (incorrectly) made "her mother" be a phrase.

### **Problem 11: (DM-MODIF2 mycon mypred dir myslot)**

Starting at **mycon**, this demon looks in direction **dir** for a CON atom with a *pred* that isa **mypred** and if *found*, uses INSERT-SL to insert **myslot** with the filler being the gap of **myslot** into the top level of the frame associated with *found*.

*Example:*

WKMEM = (CON1 CON2 CON3 CON4 CON5)

USEDMEM = (CON1)

CON1 = (MODIF **AGE** AGE7)

CON2 = (MODIF **LOC** LC)

CON3 = (THAT)

CON4 = (HUMAN **MOTHER-OF** MOF)

CON5 = (ACT **AGENT** (HUMAN **F-NAME** (CHARLOTTE)) **OBJECT** OBJ)

(DM-MODIF2 CON2 ACT AFT LOC)

...will find CON5, so as a result:

CON5 = (ACT **AGENT** (HUMAN **F-NAME** (CHARLOTTE))  
          **OBJECT** OBJ  
          **LOC** LC)

*Usage:*

MWM-S3 = (AT CHARLOTTE-S HOME CHARLES SECRETLY THROWS A NEWSPAPER  
STORY IN THE TRASH CAN BUT CHARLOTTE SEES HIM)

Phrase: (AT)

Frame: (MODIF **LOC** LC)

Demons:

DM-EXP looks for a LOCATION after and if found binds *found* to the gap of **LOC**

DM-MODIF2 looks for a pred that isa ACT after and if found, inserts into the frame of *found* the slot **LOC** with the gap of **LOC** as the filler

### **Problem 12: (DM-HIM-FNAME mycon myslot dir pred slot filler)**

---

This demon looks for a CON atom in direction **dir** that has a top-level **pred** with slot **slot** having filler value **filler**. If found, DM-HIM-FNAME looks to see if *found* has a slot **myslot**. If so, it sets the value of **myslot** in the found frame to be the value of **myslot** in the frame of that CON atom.

*Example:*

```
WKMED = (CON1 CON2 CON3 CON4 CON5)
USEDMEM = (CON1)
CON1 = (HUMAN  F-NAME (CHARLES)  GENDER (MALE))
CON2 = (HUMAN  F-NAME (EMMA)  GENDER (FEMALE))
CON3 = (THAT)
CON4 = (HUMAN  F-NAME  FNM8  GENDER (MALE))
CON5 = (EMOTION  SENTIM (POS)  SCALE (>NORM))

(DM-HIM-FNAME  CON4  F-NAME  BEF  HUMAN  GENDER  (MALE))
```

...will find CON1, so as a result:  
FNM8 = (CHARLES)

*Usage:*

```
MWM-S3 = (AT CHARLOTTE-S HOME CHARLES SECRETLY THROWS A NEWSPAPER
STORY IN THE TRASH CAN BUT CHARLOTTE SEES HIM)
```

Phrase: (HIM)  
Frame: (HUMAN F-NAME FNM GENDER (MALE))  
Demons:

DM-HIM-FNAME looks for a HUMAN with GENDER = (MALE) before and if found, accesses the filler of *found's* **F-NAME** slot and binds the gap of **F-NAME** (in the frame that this demon works for) with the **F-NAME** filler that was found.

*Note:*

The above definition sounds complicated when described abstractly but, in the demon-instance above (for the word HIM) **myslot** will be F-NAME. Thus, this demon is looking for the first name of a previously mentioned HUMAN whose **GENDER** is (MALE) and is going to assign that first name to the gap FNM above, so it is trying to figure out the first name of "him")



### **Problem 13: (DM-USED MYCON)**

---

This demon simply adds **MYCON** to USEDMEM.

*Example:*

USEDMEM = (CON1)

(DM-USED CON5)

USEDMEM = (CON1 CON5)

*Usage:*

MWM-S9 = (CHARLOTTE FEELS **THAT** CHARLES KILLED SAMANTHA)

Phrase: (THAT)

Frame: (THAT)

Demons:

DM-USED adds its CON atom (the one it works for) to USEDMEM

## Merry Widow Murderer Sentence Triplets (By Sentence)

---

### *Sentence 1*

---

MWM-S1 = (TEENAGER CHARLOTTE NEWTON FEELS EXCITED)

Phrase: (TEENAGER)

Frame: (MODIF **AGE** (RANGE **FROM** (13) **TO** (19) **UNIT** (YEAR)))

Demons:

DM-MODIF looks for HUMAN after; if found, changes the slot-filler pair:  
[**AGE** (RANGE **FROM** (13) **TO** (19) **UNIT** (YEAR))] to that HUMAN.  
As a side-effect, the CON atom of this frame is added to USEDMEM.

Phrase: (CHARLOTTE)

Frame: (HUMAN **F-NAME** (CHARLOTTE)

**L-NAME** LNM

**GENDER** (FEMALE))

Demons:

DM-LNAME looks for UNKNOWN **WORD** immediately after and if found, and sets it to the gap of slot **L-NAME**

Phrase: (NEWTON): do not add to LEXMEM. Your NEXT-PH function will return:  
(UNKNOWN **WORD** (NEWTON)) as the frame for this unknown word.

Phrase: (FEELS): *Note*: this word is ambiguous in the MWM story

Frame: ( )

Demons:

DM-DISAMB-FEELS looks for predicate immediately after and if found then assigns  
UNIQUE-GAPS of frame:

(BELIEVE **AGENT** AG **OBJECT** OBJ) to **MYCON** and spawns 2 demons:

DM-EXP looking for HUMAN before

DM-EXP looking for ACT after

else DM-DISAMB-FEELS assigns UNIQUE-GAPS of frame:

(STATE **TYPE** TYP **AGENT** AG) to **MYCON** and spawns 2 demons:

DM-EXP looking for HUMAN before and

DM-EXP looking for EMOTION after

Phrase: (EXCITED)  
Frame: (EMOTION **SENTIM** (POS) **SCALE** (>NORM))  
Demons: ( )

LEXIC-S1 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentence 1.

Example:  
(C-ANALYZER MWM-S1 LEXIC-S1)  
returns:  
((STATE **TYPE** (EMOTION **SENTIM** (POS)  
  **SCALE** (>NORM))  
  **AGENT** (HUMAN **F-NAME** (CHARLOTTE)  
                  **L-NAME** (NEWTON)  
                  **GENDER** (FEMALE)  
                  **AGE** (RANGE **FROM** (13) **TO** (19) **UNIT** (YEAR))))))

#### *Sentence 2*

---

MWM-S2 = (MYSTERIOUS UNCLE CHARLES OAKLEY WILL MOVE IN WITH HER MOTHER EMMA)

Phrase: (MYSTERIOUS)  
Frame: (MODIF **HISTORY** (UNKNOWN))  
Demons:

DM-MODIF looks for HUMAN after and if found, inserts slot-filler pair:  
[**HISTORY** (UNKNOWN)] into that HUMAN

Phrase: (UNCLE CHARLES)  
Frame: (HUMAN **F-NAME** (CHARLES)  
                  **L-NAME** LNM  
                  **GENDER** (MALE)  
                  **UNCLE-OF** UNC)

Demons:

DM-EXP looks before for a HUMAN and if found, binds *found* to the gap of the **UNCLE-OF** slot.

DM-LNAME looks immediately after for an UNKNOWN **WORD** and if found, sets its filler to the gap of the L-NAME slot.

*Note:*

To keep this project simple, we are (incorrectly) combining uncle and Charles into a single phrase. (A learning system would learn over time to consolidate words into phrases to reduce the number of demon instances needed.)

Phrase: (OAKLEY): Not added to LEXMEM, so NEXT-PH will return (UNKNOWN **WORD** (OAKLEY)) as the frame.

Phrase: (WILL)

Frame: (MODIF **TIME** (FUTURE))

Demons:

DM-MODIF looks for an ACT after and if found, inserts slot-filler pair:  
[**TIME** (FUTURE)] into the frame associated with *found*

Phrase: (MOVE IN WITH)

Frame: (CO-HABITATE **AGENT** AG **OBJECT** OBJ)

Demons:

DM-EXP looks for HUMAN before and binds to gap of **AGENT** slot DM-EXP looks for HUMAN after and binds to gap of **OBJECT** slot

Phrase: (HER MOTHER)

Frame: (HUMAN **F-NAME** FNM

**GENDER** (FEMALE)

**MOTHER-OF** MOF)

Demons:

DM-FNAME looks for HUMAN immediately after with a slot **F-NAME** and its filler (call that *found*). This demon binds gap of F-NAME (i.e. FNM) with *found* and adds the CONatom of found to USEDMEM.

DM-FINDHER looks for a HUMAN that is **GENDER** (FEMALE) before and if found, binds *found* to gap of the **MOTHER-OF** slot.

*Note:*

The word "her" is ambiguous. It could be a person ("I was her."). It could be ownership of something ("I saw her ring."). It could be family relationship ("I saw her mother."). In this project you are NOT going to disambiguate it, so we have cheated here and (incorrectly) made "her mother" be a phrase.

Phrase: (EMMA)

Frame: (HUMAN **F-NAME** (EMMA) **GENDER** (FEMALE))

Demons: ( )

*Note:*

We have (incorrectly) ignored Emma's last name (because it isn't mentioned in the story). Since some conceptual structures will have more information than others, we will later need to be able to unify structures of different sizes.

LEXIC-S1-S2 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 2.

(C-ANALYZER MWM-S2 LEXIC-S1-S2)

returns:

```
((CO-HABITATE AGENT (HUMAN
  F-NAME(CHARLES)
  L-NAME (OAKLEY)
  GENDER (MALE)
  FAMREL (UNCLE-OF
    OBJECT (HUMAN F-NAME (CHARLOTTE)
      L-NAME (NEWTON)
      GENDER (FEMALE)
      AGE (RANGE FROM (13)
        TO (19)
        UNIT (YEAR))))
    HISTORY (UNKNOWN))
  OBJECT (HUMAN F-NAME (EMMA)
    GENDER (FEMALE))
  MOTHER-OF (HUMAN F-NAME (CHARLOTTE)
    L-NAME (NEWTON)
    GENDER (FEMALE)
    AGE (RANGE FROM (13)
      TO (19)
      UNIT (YEAR)))
  TIME (FUTURE)))
```

*Sentence 3*

---

MWM-S3 = (AT CHARLOTTE-S HOME CHARLES SECRETLY THROWS A NEWSPAPER STORY IN THE TRASH CAN BUT CHARLOTTE SEES HIM)

Phrase: (AT)

Frame: (MODIF **LOC** LC)

Demons:

DM-EXP looks for a LOCATION after and if found binds *found* to the gap of **LOC**

DM-MODIF2 looks for a pred that isa ACT after and if found, inserts into the frame of *found* the slot **LOC** with the gap of **LOC** as the filler

Phrase: (CHARLOTTE-S)

Frame: ( )

Demons:

DM-MODIF looks for a PRED that isa INANIMATE after and if found, adds slot-filler: [**OWNER** (HUMAN **F-NAME** (CHARLOTTE) **GENDER** (FEMALE))] to the frame of *found*

Phrase: (HOME)

Frame: (HOME)

Demons: ( )

Phrase: (CHARLES)

Frame: (HUMAN **F-NAME** (CHARLES)

**GENDER** (MALE)

**L-NAME** LNM)

Demons:

DM-LNAME looks for UNKNOWN **WORD** immediately after and if found, sets it to the gap in slot **L-NAME** (in this case, the gap LNM)

Phrase: (SECRETLY)

Frame: (MODIF **MANNER** (SECRETLY))

Demons:

DM-MODIF looks for an ACT after and if found, inserts slot-filler: [**MANNER** (SECRETLY)] into that ACT

Phrase: (THROWS)

Frame: (THROW **AGENT** AG **OBJECT** OBJ)

Demons:

DM-EXP looks for HUMAN before and if found, binds it to the gap of the **AGENT** slot

DM-EXP looks for PHYS-OBJ after and if found, binds it to the gap of the **OBJECT** slot

Phrase: (A)

Frame: (MODIF **REF** (INDEF))

Demon:

DM-MODIF looks for a PHYS-OBJ after and if found, inserts slot-filler:  
[**REF** (INDEF)] into the frame of *found*.

Phrase: (NEWSPAPER STORY)

Frame: (PRINTED-MATTER **TYPE** (NEWSPAPER **MATERIAL** (PAPER))  
**INFO** (STORY))

Demons: ( )

Phrase: (IN THE TRASH CAN)

Frame: (MODIF **INTO** (CONTAINER **SHAPE** (CYLINDRICAL))  
**FOR** (TRASH)  
**REF** (DEF))

Demons:

DM-MODIF looks for an ACT before and if found, inserts slot-filler pair  
[**INTO** (CONTAINER **SHAPE** (CYLINDRICAL)) **FOR** (TRASH) **REF** (DEF)] into  
that ACT

*Note:*

To reduce your labor, "into the trash can" is (incorrectly) treated as a phrase.

Phrase: (BUT)

Frame: (MODIF **EXP-VIOL** EXPV)

Demons:

DM-EXP looks for a pred that isa ACT after and if found binds found to the gap of  
**EXP-VIOL**

DM-MODIF2 looks for a pred that isa ACT before and if found, inserts into the frame of found the slot **EXP-VIOL** along with its gap

Phrase: (CHARLOTTE): Already in the lexicon

Phrase: (SEES)

Frame: (SEE **AGENT** AG0 **OBJECT** OBJ0)

Demons:

DM-EXP looks for a HUMAN before and if found, binds it to gap of **AGENT**

DM-EXP looks for pred that isa PHYS-OBJ after and if found, binds *found* to the gap of **OBJECT**

Phrase: (HIM)

Frame: (HUMAN **F-NAME** FNM **GENDER** (MALE))

Demons:

DM-HIM-FNAME looks for a HUMAN with GENDER = (MALE) before and if found, accesses the filler to *found's* F-NAME slot and binds the gap of **F-NAME** with the F-NAME filler that was found.

*Note:*

The above definition sounds complicated when described abstractly but, in the demon-instance above (for the word HIM) **myslot** will be F-NAME. Thus, this demon is looking for the first name of a previously mentioned HUMAN whose **GENDER** is (MALE) and is going to assign that first name to the gap FNM above, so it is trying to figure out the first name of "him")

LEXIC-S1-S3 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 3.

(C-ANALYZE MWM-S3 LEXIC-S1-S3)

returns:

(THROW **AGENT** (HUMAN **F-NAME** (CHARLES)

**L-NAME** ( )

**GENDER** (MALE))

**OBJECT** (PRINTED-MATTER **TYPE** (NEWSPAPER **MATERIAL** (PAPER))

**INFO** (STORY)

**REF** (INDEF))

**INTO** (CONTAINER **SHAPE** (CYLINDRICAL)

**FOR** (TRASH)

**REF** (DEF))

**MANNER** (SECRETLY)



**EXP-VIOL** (SEE **AGENT** (HUMAN **F-NAME** (CHARLOTTE)  
**L-NAME** ( )  
**GENDER** (FEMALE))  
**OBJECT** (HUMAN **F-NAME** (CHARLES)  
**L-NAME** ( )  
**GENDER** (MALE)))  
**LOC** (HOME **OWNER** (HUMAN **F-NAME** (CHARLOTTE)  
**L-NAME** ( )  
**GENDER** (FEMALE))))

*Sentence 4:*

---

MWM-S4 = (CHARLOTTE READS THE NEWSPAPER STORY)

Phrase: (CHARLOTTE): Already in the lexicon

Phrase: (READS)

Frame: (READ **AGENT** AG **OBJECT** OBJ)

Demons:

DM-EXP looks for a HUMAN before and if found, binds it to gap of **AGENT**

DM-EXP looks for PRINTED-MATTER after and if found, binds it to **OBJECT**

Phrase: (THE)

Frame: (MODIF **REF** (DEF))

Demons:

DM-MODIF looks for a PHYS-OBJ after and if found, inserts slot-filler pair:  
**[REF (DEF)]** to the frame of *found*

Phrase: (NEWSPAPER STORY)

Frame: (PRINTED-MATTER **TYPE** (NEWSPAPER **MATERIAL** (PAPER))  
**INFO** (STORY))

Demons: ( )

LEXIC-S1-S4 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 4.

(C-ANALYZE MWM-S4 LEXIC-S1-S4)

returns:

(READ **AGENT** (HUMAN **F-NAME** (CHARLOTTE)

**L-NAME** ( )

**GENDER** (FEMALE))

**OBJECT** (PRINTED-MATTER **TYPE** (NEWSPAPER **MATERIAL** (PAPER))

**INFO** (STORY)

**REF** (DEF)))

*Sentence 5:*

---

MWM-S5 = (POLICE ARE LOOKING FOR THE MERRY WIDOW MURDERER)

Phrase: (POLICE)

Frame: (HUMAN **NUMBER** ( >1) **ROLE** (POLICE))

Demons: none

Phrase: (ARE LOOKING FOR)

Frame: (SEEK **AGENT** AG **OBJECT** OBJ)

Demons:

DM-EXP looks for a HUMAN before and if found, binds to **AGENT** gap

DM-EXP looks for HUMAN after and if found, binds it to **OBJECT** gap

Phrase: (THE MERRY WIDOW MURDERER)

Frame: (HUMAN **GENDER** (MALE)

**NICNAME** (M-WIDOW-M)

**ROLE** (MURDERER)

**REF** (DEF))

Demons: ( )

LEXIC-S1-S5 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 5.

(C-ANALYZE MWM-S5 LEXIC-S1-S5)

returns:

(SEEK **AGENT** (HUMAN **NUMBER** ( >1) **ROLE** (POLICE))

**OBJECT** (HUMAN **GENDER** (MALE)

**NICNAME** (M-WIDOW-M)

**ROLE** (MURDERER)

**REF** (DEF)))

*Sentence 6*

---

MWM-S6 = (THE MERRY WIDOW MURDERER KILLED WIDOW SAMANTHA VODNER FOR HER JEWELRY)

Phrase: (THE MERRY WIDOW MURDERER): Already in the lexicon

Phrase: (KILLED)

Frame: (KILL **AGENT** AG **OBJECT** OBJ)

Demons:

DM-EXP looks for a HUMAN before and if found, binds it to **AGENT**'s gap

DM-EXP looks for HUMAN after and if found, binds it to **OBJECT**'s gap

Phrase: (WIDOW)

Frame: (MODIF **ROLE** (WIDOW))

Demons:

DM-MODIF looks for a HUMAN after and if found, inserts slot-filler:  
[**ROLE** (WIDOW)] into that HUMAN

Phrase: (SAMANTHA)

Frame: (HUMAN **F-NAME** (SAMANTHA)

**L-NAME** LNM

**GENDER** (FEMALE))

Demons:

DM-LNAME looks for UNKNOWN word immed after and if found, sets the filler of  
L-NAME found to gap of slot **L-NAME**

Phrase: (VODNER): Not in lexicon so will become (UNKNOWN **WORD** (VODNER))

Phrase: (FOR HER JEWELRY)

Frame: (MODIF **MOTIVE** (JEWELRY))

Demons:

DM-MODIF looks for a pred that isa ACT before and if found, adds slot-filler:  
[**MOTIVE** (JEWELRY)] to that ACT

LEXIC-S1-S6 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 6.

(C-ANALYZE MWM-S6 LEXIC-S1-S6)

returns:

(KILL **AGENT** (HUMAN **GENDER** (MALE)  
                                  **NICNAME** (M-WIDOW-M)  
                                  **ROLE** (MURDERER))  
      **OBJECT** (HUMAN **F-NAME** (SAMANTHA)  
                                  **L-NAME** (VODNER)  
                                  **GENDER** (FEMALE)  
                                  **ROLE** (WIDOW))  
      **MOTIVE** (JEWELRY))

*Sentence 7*

---

MWM-S7 = (CHARLES GIVES CHARLOTTE AN EXPENSIVE EMERALD RING)

Phrase: (CHARLES) -- is already in LEXMEM

Phrase: (GIVES)

Frame: (GIVE **AGENT** AG0 **OBJECT** OBJ0 **TO** TO0)

Demons:

DM-EXP looks for a HUMAN before and if found, binds it to **AGENT**'s gap

DM-EXP looks for HUMAN after and if found, binds it to **TO**'s gap

DM-EXP looks for a PHYS-OBJ after and if found, binds it to **OBJECT**'s gap

Phrase: (CHARLOTTE): Already in lexicon

Phrase: (AN EXPENSIVE)

Frame: (MODIF **COST** (>NORM))

Demons:

DM-MODIF looks for a PHYS-OBJ after and if found, inserts slot-filler:  
[**COST** (>NORM)] into frame found.

Phrase: (EMERALD RING)

Frame: (RING **CONTAINS** (EMERALD))

Demons: ( )

LEXIC-S1-S7 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 7.

(C-ANALYZE MWM-S7 LEXIC-S1-S7)

returns:

(GIVE **AGENT** (HUMAN **F-NAME** (CHARLES)  
                                  **L-NAME** ( )  
                                  **GENDER** (MALE))  
  **TO** (HUMAN **F-NAME** (CHARLOTTE)  
                                  **L-NAME** ( )  
                                  **GENDER** (FEMALE))  
  **OBJECT** (RING **CONTAINS** (EMERALD)  
                                  **COST** (>NORM)))

*Sentence 8*

---

MWM-S8 = (CHARLOTTE SEES INITIALS SV ON THE RING)

Phrase: (CHARLOTTE): Already in lexicon

Phrase: (SEES)

Frame: (SEE **AGENT** AG   **OBJECT** OBJ)

Demons:

DM-EXP looks for a HUMAN before and if found, binds it to **AGENT's** gap

DM-EXP looks for pred that isa PHYS-OBJ after and if found, binds it to **OBJECT's** gap

Frame: (INITIALS SV)

Frame: (PRINTED-MATTER **TYPE** (INITIALS **F-INIT** (S) **L-INIT** (V))

Demons: ( )

Phrase: (ON THE RING)

Frame: (MODIF **ON** (RING **REF** (DEF)))

Demons:

DM-MODIF3 looks for a pred that isa PHYS-OBJ before and if found, inserts slot-filler:  
[**ON** (RING **REF** (DEF))] to frame found.

LEXIC-S1-S8 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 8.

(C-ANALYZE MWM-S8 LEXIC-S1-S8)

returns:

(SEE **AGENT** (HUMAN **F-NAME** (CHARLOTTE)

**L-NAME** ( )

**GENDER** (FEMALE))

**OBJECT** (PRINTED-MATTER **TYPE** (INITIALS **F-INIT** (S) **L-INIT** (V))

**ON** (RING **REF** (DEF))))

*Sentence 9*

---

MWM-S9 = (CHARLOTTE FEELS THAT CHARLES KILLED SAMANTHA)

Phrase: (CHARLOTTE): Already in lexicon

Phrase: (FEELS): Already in lexicon

(*Note*: this time the associated disambiguation demon will find the frame (THAT) immediately after and so will disambiguate FEELS to be the frame (BELIEVE **AGENT** AG **OBJECT** OBJ) and spawn two DM-EXP demons to fill these slots.)

Phrase: (THAT)

Frame: (THAT)

Demons:

DM-USED adds its CON atom (the **mycon** atom parameter that it works for) to USED MEM

Phrase: (CHARLES): Already in lexicon

Phrase: (KILLED): Already in lexicon

Phrase: (SAMANTHA): Already in lexicon

LEXIC-S1-S9 below refers to the content of LEXMEM after using ADD-LEX on the words/phrases in sentences 1 to 9 (i.e. entire story).

(C-ANALYZE MWM-S9 LEXIC-S1-S9)

returns:

((BELIEVE **AGENT** (HUMAN **F-NAME** (CHARLOTTE)

**L-NAME** ( )

**GENDER** (FEMALE))

**OBJECT** (KILL **AGENT** (HUMAN **F-NAME** (CHARLES)

**L-NAME** ( )

**GENDER** (MALE))

**OBJECT** (HUMAN **F-NAME** (SAMANTHA)

**L-NAME** ( )

**GENDER** (FEMALE))))))

### **What we are still missing...**

---

- Episodic memories being formed and indexed for later access during Q/A
- Numerous inferences being made as the above concepts are created.
- Questions being posed (which required that the conceptual content of questions be understood and that conceptual answers be sought in episodic memory)
- Generation of English sentences to express those retrieved conceptual structures.

Due to time constraints, you will only do small subsets of the above.



## Assignment Restrictions & Tips

---

- **Start Early!**
- **Late Policy**  
You got some dispensation with HW1, but this time the iron hammer comes down for late submissions! Submit ~0 – 24 hours late and you'll lose 30% off the bat; submit more than a day late and it's a 0.
- **Using HW1 & HW2 Solutions**  
You may use the HW1 & HW2 solutions we post on CourseWeb as the backbone for HW3, as these will be used to test your code.
- **Helper Functions Are OK!**  
Feel free to employ helper functions to be used within any of your homework solutions.
- **Make Unit Tests!**  
You should practice test-driven development for this homework, which might include a test suite that loads your function definitions and operates on some of the frames we provide.
- **Your Function Names Must Match These Exactly!**  
Failure to name your functions exactly as they appear in this homework will result in a grade of 0 for that problem!
- **Unfinished Functions**  
If you run out of time on a function, make sure you at least include that function definition and simply return 'UNIMPLEMENTED'.
- **What to Submit**  
Submit one file, titled "*your-ID-number-here.lsp*" with all of your function definitions, and any helper functions you employed. DO NOT SUBMIT UNIT TESTS.

E.g. I might submit a file named "55555555.lsp"