The Cyc Lexicon

Elizabeth Coppock
University of Texas at Austin and Cycorp, Inc.

Jozef Stefan Institute, March 19, 2010
Overview

- Natural language microtheories
- Kinds of semantic predicates
- Inflectional and derivational morphology
Themes

• How to modify / add to the Cyc lexicon
• The *generative* nature of the Cyc lexicon: use of inference rules for deriving new entries
Language Microtheories

- GeneralEnglishMt
  - AmericanEnglishMt
  - BritishEnglishMt
  - IrishEnglishMt
  - NewZealandEnglishMt
- SlovenianMt
- PolishMt
- ...

Favour-TheWord

Mt : BritishEnglishMt
infinitive : °"favour"

Mt : GeneralEnglishMt
infinitive : °"favor"

Mt : BritishEnglishMt
massNumber : °"favour"

Mt : GeneralEnglishMt
massNumber : °"favor"
Lexicon Microtheories

• GeneralLexiconMt
  – EnglishLexiconMt
  – SloveneLexiconMt

• for defining (language-specific) lexicon predicates, e.g. dual
• not for lexical mappings
Overview

• NL microtheories
• Kinds of semantic predicates
• Inflectional and derivational morphology
Two string-concept routes

- concepts
  - denotation, etc.
  - words
    - massNumber, etc.
- strings
- nameString, etc.
Kinds of semantic predicates

• **nameString**-like:
  – string ~ concept

• **denotation**-like:
  – word ~ concept

• *semTrans (verbSemTrans, etc.)*:
  – word ~ semantic template

• **genTemplate, assertTemplate**:
  – sentence template ~ concept
(nameString MarkTwain "Mark Twain")
(nameString MarkTwain "Samuel Clemens")
nameString-like predicates

familyName
lastName
middleName
firstName
fullName
nickNames
formerName
hipHopMoniker
accessFilename
atomicSymbol
countryCodeDigraph
internetCountryCode
airportHasIATACode
cwEntitled
movieTitleString
sectionTitle
Lexical Info for **ECoppock**:

**Default generated phrase:** "Elizabeth Coppock" (#$nameString)

**Mt:** **EnglishMt**
- (familyName ECoppock "Coppock")
- (givenNames ECoppock "Elizabeth")
- (M(nameString ECoppock "Elizabeth Coppock")
- (nameString ECoppock "Liz Coppock")

**Mt:** **(MtSpace CyclishMt (MtTimeDimFn (IntervalStartedByFn-Inclusive (YearFn 2009))))**
- (nameString ECoppock "Elizabeth Coppock, Ph.D.")
- (nameString ECoppock "Elizabeth Coppock, PHD")

**Mt:** **EnglishMt**
- (genStringAssertion ECoppock
  - (M(nameString ECoppock "Elizabeth Coppock")))
Assertion:
- (nameString ECoppock "Elizabeth Coppock, PHD")
in (MtSpace CyclishMt (MtTimeDimFn (IntervalStartedByFn-Inclusive (YearFn 2009))))

Arguments Supporting Assertion:
Argument: Deduction #3497335
- (implies
  (and
   (isa ?PERSON PhDLevel)
   (givenNames ?PERSON ?GIVEN-NAME)
   (familyName ?PERSON ?FAMILY-NAME)
   (concatenateStrings ?NAME-PLUS ?GIVEN-NAME " " ?FAMILY-NAME ", " "PHD"))
   (nameString ?PERSON ?NAME-PLUS)) in EnglishMt
- (familyName ECoppock "Coppock") in EnglishMt
- (givenNames ECoppock "Elizabeth") in EnglishMt
- (isa ECoppock PhDLevel)
in (MtSpace CyclistsMt (MtTimeDimFn (IntervalStartedByFn-Inclusive (YearFn 2009))))

:OPAQUE (concatenateStrings "Elizabeth Coppock, PHD" "Elizabeth" " "Coppock" ", " "PHD")
Predicate: **denotation**

- **arg1lsa**: `LexicalWord`
- **arg2lsa**: `SpeechPart`
- **arg3lsa**: `Integer`
- **arg4lsa**: `Thing`
Last query in **EverythingPSC**: (isa ?ARG1 SpeechPart)

121 answers for ?ARG1:
Predicate: **speechPartPreds**

GAF Arg: 1

Mt: UniversalVocabularyMt

**comment**: "This predicate expresses the relation between parts of speech and the corresponding syntactic predicates. The first argument is a part of speech, i.e., a collection that is a specialization of **SpeechPart**. The second argument is a **SpeechPartPredicate** (q.v.). For instance, (speechPartPreds CountNoun singular) means that the predicate **singular** only applies to words that have **CountNoun** forms."
Constant : Ring-TheWord
isa : EnglishWord

Mt : EnglishMt
singular : “ring”
infinitive : “ring”

(denotation Ring-TheWord Verb 1 EmittingSound)
(denotation Ring-TheWord CountNoun 1 AudibleSound)
(denotation Ring-TheWord CountNoun 0 RingShapedObject)
(denotation Ring-TheWord CountNoun 2 Ring-Jewelry)
Constant: Coke-TheWord
isa: EnglishWord

Mt: EnglishMt
singular: "coke"    pnSingular: "Coke"
massNumber: "coke"    pnMassNumber: "Coke"

(denotation Coke-TheWord ProperCountNoun 0 (ServingFn CocaCola))
(denotation Coke-TheWord ProperMassNoun 0 CocaCola)
(denotation Coke-TheWord MassNoun 0 Cocaine-Powder)
(denotation Coke-TheWord MassNoun 2 ColaSoftDrink)
(denotation Coke-TheWord SimpleNoun 0 (ServingFn ColaSoftDrink))
(denotation Coke-TheWord MassNoun 1 CokeFuel)
Mass nouns vs. Count nouns

• Only count nouns can be plural
  – one ring, two rings [count]
  – one sand, *two sands [mass]
  – one coke, two cokes [ambiguous]

• Only mass nouns take determiner some
  – *Can I have some ring? [count]
  – Can I have some sand? [mass]
  – Can I have some coke? [ambiguous]
Multi-word lexical entries

- The `denotation` predicate maps single words to concepts.
- What about phrases like "Clinton joke" and "joke about Clinton"?
  - `multiWordString` for head-final phrases
  - `compoundString` for head-initial phrases
Predicate: multiWordString

arg1 ISA: (ListOfTypeFn ControlCharacterFreeString)
arg2 ISA: LexicalWord
arg3 ISA: SpeechPart
arg4 ISA: Thing
Mt : AllEnglishLexicalMicrotheoryPSC
   (multiWordString (TheList "Clinton") Joke-TheWord CountNoun (JokeAboutFn BillClinton)))
Mt : AllEnglishLexicalMicrotheoryPSC
(multiWordString (TheList "Clinton") Joke-TheWord CountNoun (JokeAboutFn BillClinton))
Predicate : compoundString

arg1isa : LexicalWord
arg2isa : (ListOfTypeFn ControlCharacterFreeString)
arg3isa : SpeechPart
arg4isa : Thing
CompoundString or MultiWordString?

Example:

one joke about Clinton
two jokes about Clinton
*two joke about Clintons

→ joke is the head of the phrase
→ use CompoundString instead of MultiWordString
What if I can't remember this?

• In many cases, you can just use the Dictionary Assistant.
Dictionary Assistant

Specify the term and phrase for the lexical mapping

Complete Term: AdultMaleHuman
Phrase: guy

Clear Term Clear Phrase Reset OK
Dictionary Assistant

Checking phrase ambiguity

In addition to 'man', phrase "guy" also refers to the following:
  male person
  Guy Steele
  Sir ovdia
  Sir ovdia
  Sergeant First Class Guy Hasson
  Guyana

Is that OK?

[Yes] [No]
Dictionary Assistant

Checking whether the phrase is a proper name

Is the phrase "guy" a name?

Yes  No
Dictionary Assistant

Checking usage of 'guy'

Which of the following sounds best?

- many guys; a guy: countable noun like 'book'
- much guy; some guy: uncountable noun like 'sand'

OK
What about relational nouns like *mother* and *temperature*?

- Three options:
  - `denotesArgInReln`
  - `RelationParaphraseMt`
  - `nounSemTrans`
denotesArgInReln

(denotesArgInReln Mother-TheWord CountNoun mother 2)

This says: One denotation of the word *mother* qua *CountNoun* is the collection of things M such that:

(relationExistsInstance mother Thing M)

i.e. the set of all mothers
RelationParaphraseMt

Mt : RelationParaphraseMt

(denotation Income-TheWord CountNoun 1 income)
nounSemTrans

(WWordWithSuffixFn] [Assassinate-TheWord] Ion [AbstractNounProducing-TheSuffix] 132 GenitiveFrame
and
(isa :NOUN AssassinatingSomeone)
(victim :NOUN :POSSESSOR))) in GeneralEnglishMt
verbSemTrans

Mt : AllEnglishValidatedLexicalMicrotheoryPSC
   (verbSemTrans Scratch-TheWord 0 IntransitiveVerbFrame
      (and
         (isa :ACTION ScratchingOneself)
         (primaryObjectMoving :ACTION :SUBJECT)))
   (verbSemTrans Scratch-TheWord 0 TransitiveNPFrame
      (and
         (isa :ACTION ScratchingOneself)
         (objectOfStateChange :ACTION :OBJECT)
         (doneBy :ACTION :SUBJECT)))
verbSemTrans-Canonical

(verbSemTrans-Canonical Assassinate-TheWord 0 TransitiveNPFrame AssassinatingSomeone (TheList perpetrator victim)) in GeneralEnglishMt
Kinds of semantic predicates

- **nameString**-like:
  - string \(\sim\) concept

- **denotation**-like:
  - word \(\sim\) concept

- **semTrans** (verbSemTrans, etc.):
  - word \(\sim\) semantic template

- **genTemplate**, **assertTemplate**:
  - sentence template \(\sim\) concept
genTemplate

Mt : EnglishParaphraseMt
   (genTemplate extentCardinality
      (ConcatenatePhrasesFn
         (TermParaphraseFn-NP :ARG1)
         (BestHeadVerbForInitialSubjectFn Have-TheWord)
         (BestDetNbarFn
            (TermParaphraseFn :ARG2)
            (ConcatenatePhrasesFn
               (BestNLPredicateOfStringFn "known")
               (HeadWordOfPhraseFn
                  (BestNLWordFormOfLexemeFn-Constrained CountNoun Member-TheWord))))))
That `genTemplate` in tree form

```
S
  ARG1 Have-TheWord
  NP
    ARG2 "known"
    Member-TheWord
```
Deriving genTemplates

- Usually, genTemplate assertions are hand-written for each predicate
- But in principle, they could be derived by rule using facts about the predicate
- There are some examples of this
Assertion:
(\text{genTemplate\ biologicalGrandfathers}
 (BasicTransitiveSentenceFn
 (BestDetNbarFn
 (TermParaphraseFn-Possessive :ARG1)
 (TermParaphraseFn-Constrained\ plural-Generic\ biologicalGrandfathers))\ Include-TheWord
 (TermParaphraseFn-NP :ARG2)))\ in\ EnglishParaphraseMt

Arguments\ Supporting\ Assertion:
Argument:\ Deduction\ #2835950
\text{M}(\text{implies}
 (\text{isa} \ ?PRED\ FamilyRelationSlot)
 (\text{genTemplate} \ ?PRED
 (BasicTransitiveSentenceFn
 (BestDetNbarFn
 (TermParaphraseFn-Possessive :ARG1)
 (TermParaphraseFn-Constrained\ plural-Generic \ ?PRED))\ Include-TheWord
 (TermParaphraseFn-NP :ARG2))))\ in\ EnglishParaphraseMt
\text{ISA} (\text{isa} \ biologicalGrandfathers\ FamilyRelationSlot)\ in\ UniversalVocabularyMt
Kinds of semantic predicates

• **nameString**-like:
  – string ~ concept

• **denotation**-like:
  – word ~ concept

• **semTrans** (*verbSemTrans*, etc.):
  – word ~ semantic template

• **genTemplate**, **assertTemplate**:
  – sentence template ~ concept
Overview

• NL microtheories
• Kinds of semantic predicates
• Inflectional and derivational morphology
Two string-concept routes

- concepts
  - denotation, etc.
  - words
    - massNumber, etc.
  - nameString, etc.

- strings
Two kinds of morphology

- **Inflectional**: relates multiple forms of the same word, e.g.
  - (singular Dog-TheWord "dog")
  - (plural Dog-TheWord "dogs")

- **Derivational**: relates one word to another, e.g. Wonder-TheWord ~ Wonderfulful-TheWord
Protoypical Inflectional Morphology

- tense (past, present, future, perfect)
- aspect (progressive, etc.)
- mood (conditional, imperative)
- person (first, second, third)
- number (singular, plural)
- gender (masculine, feminine, neuter)
Inflectional Morphology in Cyc

- Inflectional suffixes are not reified individuals in the KB; their orthographic content is indicated with `regularSuffix` assertions.
- Rules attach the string to the first form to produce the second form.
Inflectional Morphology in Cyc

\[ M(\text{implies}) \]
\[ \text{and} \]
\[ (\text{infinitive} ?WU ?BSTRING) \]
\[ (\text{evaluate} ?\text{ENGLISHSUFFIXFN}) \]
\[ (\text{EnglishSuffixationFn} ?BSTRING "ed")\]) \]
\[ (\text{pastTense-Universal} ?WU ?\text{ENGLISHSUFFIXFN})\]

\[ M(\text{implies}) \]
\[ \text{and} \]
\[ (\text{infinitive} ?WU ?BSTRING) \]
\[ (\text{evaluate} ?\text{ENGLISHSUFFIXFN}) \]
\[ (\text{EnglishSuffixationFn} ?BSTRING "s")\]) \]
\[ (\text{thirdPersonSg-Present} ?WU ?\text{ENGLISHSUFFIXFN})\]

\[ M(\text{implies}) \]
\[ \text{and} \]
\[ (\text{infinitive} ?WU ?BSTRING) \]
\[ (\text{evaluate} ?\text{ENGLISHSUFFIXFN}) \]
\[ (\text{EnglishSuffixationFn} ?BSTRING "ing")\]) \]
\[ (\text{presentParticiple} ?WU ?\text{ENGLISHSUFFIXFN})\]
EnglishSuffixationFn

- Defined in code
- Not pure concatenation:
  - "wait" + "ed" = "waited"
  - "gate" + "ed" = "gated" (not "gateed")
  - "plate" + "s" = "plates"
  - "glass" + "s" = "glasses"
Mt : EnglishLexiconMt
- (regularSuffix gerund-Plural gerund "s")
- (regularSuffix gerund-Plural presentParticiple "s")
- (regularSuffix gerund-Singular presentParticiple "s")
- (regularSuffix pluralVerb-Present infinitive "s")
- (regularSuffix passiveParticiple perfect "s")
- (regularSuffix firstPersonSg-Present infinitive "s")
- (regularSuffix gerund presentParticiple "s")
- (regularSuffix perfect pastTense-Universal "s")
- (regularSuffix secondPersonSg-Present infinitive "s")
- (regularSuffix superlativeAdverb regularAdverb "est")
- (regularSuffix comparativeAdverb regularAdverb "er")
- (regularSuffix agentive-Sg infinitive "er")
- (regularSuffix pastTense-Universal infinitive "ed")
- (regularSuffix thirdPersonSg-Present infinitive "s")
- (regularSuffix pnPlural pnSingular "s")
- (regularSuffix agentive-Pl agentive-Sg "s")
- (regularSuffix plural singular "s")
- (regularSuffix presentParticiple infinitive "ing")
Prototypical Derivational Morphology

• Deverbal nouns like "embarrassment", "assassination", "growth"
• Denominal verbs like "nullify", "scrutinize"
• Denominal adjectives like "wonderful", "lovely", "boorish"
• Deadjectival nouns like "happiness"
Derivational Morphology in Cyc

• Reified suffixes, associated with
  – a string
  – POS category of base word
  – POS category of derived word

• According to a Cyc rule, attach to the **baseForm** form of the base word.
Individual: Ion AbstractNounProducing-TheSuffix

GAF Arg: 1

Mt: UniversalVocabularyMt
  isa: LexicalSuffix Individual

Mt: EnglishLexiconMt
  isa: LevelOneAffix DeverbalAbstractNounProducingSuffix EnglishDerivationalSuffix

Mt: UniversalVocabularyMt
  quotedIsa: NLIMorphologyTerm

Mt: BookkeepingMt
  quotedIsa: ResearchCycConstant-NLRelated-BreakLinkForOpenCycBuild NLIKBCconstant CycSecureFORT

Mt: EnglishLexiconMt
  comment: "The suffix "ion" ("ation", "ition") means something like *the process or state of*. It converts Verbs to Nouns, generally to MassNouns. Example: (derivedUsingSuffix #$Exploration-TheWord Ion AbstractNounProducing-TheSuffix)."
(nounSemTrans (WordWithSuffixFn Assinate-TheWord) Ion AbstractNounProducing-TheSuffix) 132 GenitiveFrame (and (isa :NOUN AssassinatingSomeone) (victim :NOUN :POSSESSOR))) in GeneralEnglishMt
Comparatives and superlatives: Inflectional or Derivational?

- The do not change the category of stem (much): "nice", "nicer" both adjectives
- Can attach to derived words, e.g. "luck+y+er"

But:
- Many adjectives lack comparative form
- Killer argument: **semantics**
Inflectional Analysis of Comparatives

tall', taller', tallest'

Tall-TheWord

"tall", "taller", "tallest"
Derivational Analysis of Comparatives

```
Tall-TheWord
"tall"
```

```
Er_Comparative-TheSuffix

WordWithSuffixFn Tall-TheWord

"taller"
```
Application: Interpreting queries

• Query: "patients heavier than 150 lbs"
• Does this mean:
  – patients who are heavy? No!
  – patients who weigh 150 pounds? No!
  – patients whose weight is greater than 150 pounds? Yes.
A rule for comparative semantics

\[(\text{implies})\]
\[(\text{and})\]
\[(\text{adjectiveOfGenericValue} \ ?\text{SLOT} \ \text{veryLowToLowAmountOf} \ ?\text{WU})\]
\[(\text{termOfUnit} \ ?\text{WORDWITHSUFFIXFN})\]
\[(\text{WordWithSuffixFn} \ ?\text{WU} \ \text{Er\_Comparative-TheSuffix}))\]
\[(\text{adjSemTrans} \ ?\text{WORDWITHSUFFIXFN} \ 27)\]
\[(\text{PPCompFrameFn} \ \text{TransitivePPFrameType} \ \text{Than-TheWord})\]
\[(\text{and})\]
\[(\text{lessThan} \ ?\text{X} : \text{OBLIQUE-OBJECT})\]
\[(\text{(?SLOT} : \text{NOUN} \ ?\text{X}))\] in GeneralEnglishMt
Assertion:

(adjSemTrans
  (WordWithSuffixFn Little-TheWord Er Comparative-TheSuffix) 27
  (PPCompFrameFn TransitivePPFrameType Than-TheWord)
  (and
    (lessThan ?X :OBLIQUE-OBJECT)
    (sizeParameterOfObject :NOUN ?X))) in GeneralEnglishMt

Arguments Supporting Assertion:
Argument: Deduction #3949626

(implies
  (and
    (adjectiveOfGenericValue ?SLOT veryLowToLowAmountOf ?WU)
    (termOfUnit ?WORDWITHSUFFIXFN
      (WordWithSuffixFn ?WU Er Comparative-TheSuffix)))
  (adjSemTrans ?WORDWITHSUFFIXFN 27
    (PPCompFrameFn TransitivePPFrameType Than-TheWord)
    (and
      (lessThan ?X :OBLIQUE-OBJECT)
      (?SLOT :NOUN ?X))) in GeneralEnglishMt

(adjectiveOfGenericValue sizeParameterOfObject veryLowToLowAmountOf Little-TheWord) in GeneralEnglishMt

(termOfUnit
  (WordWithSuffixFn Little-TheWord Er Comparative-TheSuffix)
  (WordWithSuffixFn Little-TheWord Er Comparative-TheSuffix)) in BaseKB
To do:

- Improve semantic coverage for comparative adjectives
- Derive semantics for superlatives
- Develop analogous analysis for comparative and superlative adverbs (e.g. *she ran faster / the fastest*)
Nouns with arguments

• New York Times: "...the assassination of a senior Hamas official..."
  – There was an event of assassination
  – Hamas official: the **victim** or **perpetrator**?

• General problem: linking noun phrases in PP modifiers of nouns to the appropriate role
Do nouns have argument structure like verbs?

- Subject of theoretical importance since Chomsky (1970), *Remarks on Nominalization*
- Grimshaw (1990) argued that some nouns do (process nouns) and others don't (non-process nouns)
Nouns ~ Verbs

the enemy destroyed the city
the enemy's destruction of the city
the destruction of the city by the enemy
CP complements

The physicists claimed that the earth is round.

the physicists' claim that the earth is round.
Infinitival complements

They attempted to leave

their attempt to leave
Locative PP complement

The train arrived at the station

the train's arrival at the station
Nouns: optional arguments?

The doctor examined *(the patients)*

The doctor's examination (of the patients) was successful
Result vs. Process Readings

John's examination was long
result reading, John can be exam taker

John's examination of the patients took a long time
process reading, John = agent
examination vs. exam

The examination/exam was long

result reading

The examination/*exam of the patients took a long time.

process reading
-ing: unambiguously process

They felled *(the trees)
the felling *(of the trees)

They destroyed *(the city)
the destroying *(of the city)
*frequent*: disambiguates

the expression (on her face) is good
the expression of one's feelings is good

*the frequent expression is good

the frequent expression of one's feelings is good
*constant*: also disambiguates

the assignment (on plurals) is annoying

the assignment of *unsolvable problems* is annoying

*the constant assignment is annoying

the constant assignment of *unsolvable problems* is annoying
role of possessors

The instructor's examination took a long time.

result reading, instructor = test-maker

The instructor's examination of the papers took a long time.

process reading, instructor = agent
agent-by makes of obligatory

the expression *(of aggressive feelings) by patients

the examination *(of the papers) by the instructor

the development *(of inexpensive housing) by the city

So agent-by implies process reading
author-by: of optional

An examination by a competent instructor will reveal...

The assignment by Fred was no good.

Pine Tree Hollow -- a development by Holmes Associates

So: no of => by is author, not agent
Nouns vs. Verbs

• Like verbs, certain nouns have obligatory arguments. Namely, process nouns ("complex event nominals" according to Grimshaw 1990)

• The of phrase is obligatory on process readings (Grimshaw 1990)
What nouns have process readings?

• Any noun for which there is a corresponding verb?
  – No: *to mother a child*, but: *the mother of a child* cannot refer to an event of mothering a child. Zero-derived nouns are out.

• Only morphologically complex nouns, like *examination*, *assignment*
"Elders in Marjah, the former Taliban stronghold seized this month in a NATO offensive, gave President Hamid Karzai a piece of their minds this week during his first visit to the town. Sometimes shouting, tribal leaders complained of excessive use of force by U.S. troops"
A rule for deriving lexical entries for process nominals with of PPs

\(\text{implies}\)
\(\text{and}\)
\(\text{isa} \ ?\text{SUFFIX DeverbalAbstractNounProducingSUFFIX}\)
\(\text{derivedUsingSuffix} \ ?\text{NOUN} \ ?\text{SUFFIX}\)
\(\text{morphologicallyDerivedFrom} \ ?\text{NOUN} \ ?\text{WORD}\)
\(\text{verbSemTrans-Canonical} \ ?\text{WORD} \ ?\text{NUM TransitiveNPFrame} \ ?\text{COL}\)
\(\text{TheList} \ ?\text{ROLE1} \ ?\text{ROLE2})\)
\(\text{nounSemTrans} \ ?\text{NOUN} \ 132 \text{GenitiveFrame}\)
\(\text{and}\)
\(\text{isa} :\text{NOUN} \ ?\text{COL}\)
\(?\text{ROLE2} :\text{NOUN} :\text{POSSESSOR})\) in \text{GeneralLexiconMt}
Assertion:

(nounSemTrans
  (WordWithSuffixFn Assassinate-TheWord Ion AbstractNounProducing-TheSuffix) 132 GenitiveFrame
  and
  (isa :NOUN AssassinatingSomeone)
  (victim :NOUN :POSSESSOR))) in GeneralEnglishMt

Arguments Supporting Assertion:

Argument: Deduction #3767582

(implies
  (and
    (isa ?SUFFIX DeveralAbstractNounProducingSuffix)
    (derivedUsingSuffix ?NOUN ?SUFFIX)
    (morphologicallyDerivedFrom ?NOUN ?WORD)
    (verbSemTrans-Canonical ?WORD ??NUM TransitiveNPFrame ?COL
      (TheList ?ROLE1 ?ROLE2)))
  (nounSemTrans ?NOUN 132 GenitiveFrame
    and
    (isa :NOUN ?COL)
    (?ROLE2 :NOUN :POSSESSOR)))) in GeneralLexiconMt

(verbSemTrans-Canonical Assassinate-TheWord 0 TransitiveNPFrame AssassinatingSomeone
  (TheList perpetrator victim)) in GeneralEnglishMt

(derivedUsingSuffix
  (WordWithSuffixFn Assassinate-TheWord Ion AbstractNounProducing-TheSuffix) Ion AbstractNounProducing-TheSuffix) in EnglishLexiconMt

(morphologicallyDerivedFrom
  (WordWithSuffixFn Assassinate-TheWord Ion AbstractNounProducing-TheSuffix) Assassinate-TheWord) in GeneralLexiconMt

ISA (isa Ion AbstractNounProducing-TheSuffix DeveralAbstractNounProducingSuffix) in EnglishLexiconMt
To do:

- Write rules for non-canonical verbSemTrans
- Write rules for *by* phrases, possessors, and combinations
- Capture relationship among *of* phrases, *by* phrases, and possessors
- Represent non-process readings
Conclusions

• Cyc lexicon uses cool OE stuff:
  – Microtheories, forward rules, evaulatable functions and predicates, exceptions

• Cyc lexicon is generative

• Lexicography can get interesting

• Plenty to do, even for English