CS6740/IS 6300, Lecture 25: Semantic Representations; intro to AMR

.

1. So far, we've seen first-order-logic-style representations of semantics

a) "cashiers put candy in boxes" → $put'(cashiers', candy', in'(boxes'))$

[We need the primes for the board, where I can't write italics]
From the reading: importance of predicates, arguments, and roles.

2. Some subproblems with the above style[1]:

a) Do we need a separate predicate for each arity (= # of arguments) and argument type?

i. "I ate" →
ii. "I ate a sandwich" →
iii. "I ate at my desk" →

b) For inference (and from the reading), **events are important**.

"Neo-Davidson" event representation: events are objects; each argument gets separate predicate.

i. could become $\exists\, e\ eating'(e) \wedge eater'(e, speaker')$ [2]
ii. (treated independently) could become
$\exists\, e\ eating'^{(e)} \wedge eater'(e, speaker') \wedge eaten'(e, sandwich')$

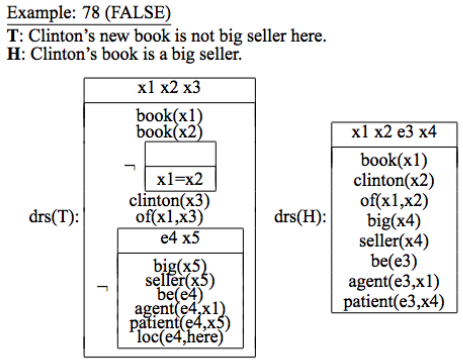
Exercise 1: How could you represent iii (treated independently of i. and ii.)?

Exercise 2A: give a representation of the combination of assertions ii and iii such that it follows that the speaker ate a sandwich at their desk.

Exercise 2B: would would you change your response so that it doesn't necessarily follow that the speaker ate a sandwich at their desk

[1] Organization follows JM 3rd edition §16.4 (stopping at 16.4.1 exclusive), examples taken, with a few simplifications, from there.
[2] Let's just take $speaker'$ for granted as a grounded item.

3. Example: Bos and Markert (2005), for first RTE challenge.
A CCG parser creates the following structures:



These are translated into first-order logic and a theorem prover is run.

4. AMR by example:

```
http://cohort.inf.ed.ac.uk/amreager.html?lang=en&sent=i+ate+a+sandwich+at+my+desk
```

```
# ::snt i ate a sandwich at my desk
# ::alignments 0-1|0.0 1-2|0 3-4|0.1 6-7|0.2
(v2 / eat-01
  :ARG0 (v1 / i)
  :ARG1 (v3 / sandwich)
  :location (v4 / desk))
```