Friday 20-04-18

Intro.

-Have you signed my extension form? is it ready?

-Fasting Ramadan with my family: I want to continue doing meetings via skype, although I am not sure about the internet quality there. However, I would like to fast and most of Ramadan and clebrare Eid with my family. If you don’t mind would like you to write me a vacation letter from Monday the 28 May-Monday 25 June (4 weeks) so that I can request my annual tickets from my embassy.

Part(1): anchoring definite references and pronouns on queries

Done by allowing them to have wider scope over query; query is a qq quantifier now, not opaque.

Example that Works

```
| ?- set(showProof).
| ?- doItAll('John loves a pretty woman.',X).
name(A::{[John:NP],A},
     claim(({[tense(present)],#0}
             & ({(simple,#0),#1}
                 & ({[woman>singular,modifier(amod,pretty)],#2}
                     & ({(member,#1),B}
                         => [[love, {dobj,#2}, {subject,A}], B]))))))
| ?- listing(fact).
fact({salient,_}).
fact({[name,'John'],'#3'}).
fact({[tense(present)],'#0'}).
fact({(simple,'#0'),'#1'}).
fact({[woman>singular,modifier(amod,pretty)],'#2'}).
| ?- doItAll('John likes a woman?',X).
| ?- listing(=>).
{[name,'John'],A}=>{he,A}.
{(member,'#1'),A}=>[[love,{dobj,'#2'},{subject,'#3'}],A].
```

```
| ?- doItAll('John likes a woman?',X).
name(A::{[John:NP],A},
     query(({[tense(present)],B}
             & ({(simple,B),C}
                 & ({[woman>singular],D}
                     & ({(member,C),#4(D,C,B)}
                         => [[like, {dobj,D}, {subject,A}],
                             #4(D, C, B)]))))))
```

Part(2):Query issues:

(A)

last week we had a problem of having both LHS and RHS of a rule skolemized.

```
| ?- doItAll('every man loves a woman?',X).
query(({[man>singular],#0}
        => ({[tense(present)],#1}
             & ({(simple,#1),#2}
                 & ({[woman>singular],#3}
                     & ({(member,#2),A}
                         => [[love, {dobj,#3}, {subject,#0}], A]))))))
```

we knew that the problem is in qff, but we end up our meeting without solving it. So, after spying on qff and following the changes of the polarity mark, my attempt to solve the problem is the following:

```
qff(forall(X, P0), P2, QSTACK, -) :- %%original
    fail,
    var(X),
    !,
    qff(exists(X, P0), P2, QSTACK, +).
qff(forall(X, P0), P2, QSTACK, -) :- %% my attempt to solve the problem
    var(X),
    !,
    skolem(X, QSTACK),
    qff(P0, P2, QSTACK, -).
```

This attempt is based on the idea that the only case that a forall would have a negative polarity is if it is a part of a query- this could be wrong cause I don’t recall any other cases of negated forall. The result is:

```
query(({[man>singular],#0}
        => ({[tense(present)],A}
             & ({(simple,A),B}
                 & ({[woman>singular],C}
                     & ({(member,B),#1(C,B,A)}
                         => [[love, {dobj,C}, {subject,#0}],
                             #1(C, B, A)]))))))
```

(B)

```
| ?- doItAll('John loves animals.',X).
| ?- listing(fact).
fact({salient,_}).
fact({[name,'John'],'#3'}).
fact({[tense(present)],'#0'}).
fact({(simple,'#0'),'#1'}).
| ?- listing(=>).
{[name,'John'],A}=>{he,A}.
{(member,'#1'),A}=>{animal>plural,'#2'(A)}&[[love,{dobj,'#2'(A)},{subject,'#3'}],A].
| ?- set(showProof).
| ?- doItAll('John likes dogs?',X).
.
.
_165414Trying Horn proof of {dog>plural,#7(#1,#0)}&[[love,{dobj,#7(#1,#0)},{subject,#3}],#6(#1,#0)]
_165414No Horn proof of {dog>plural,#7(#1,#0)}&[[love,{dobj,#7(#1,#0)},{subject,#3}],#6(#1,#0)]
_165414Looking for split rule to help with proof of
{dog>plural,#7(#1,#0)}&[[love,{dobj,#7(#1,#0)},{subject,#3}],#6(#1,#0)]
_165414Can't do it using Horn clauses: think of a split rule
```

_165414No proof using split rule of
{dog>plural,#7(#1,#0)}&[[love,{dobj,#7(#1,#0)},{subject,#3}],#6(#1,#0)]

Part(3):Polarity

-in the current version of matching algorithm it is expected that:
‘**every** man loves a woman.’ -> ‘**every** human likes a woman.’ although (every) is a downward monotone on its 1st argument. (so this is wrong, we need to check polarity, **but when to do the marking?**)

-according to the standard tree polarity marking, negations like (not) reverse the polarity. Which means ‘not every man loves a woman.’ -> ‘not every human loves a woman.’ **should be true?!**

*NaturalLogicAnnotator : Marks quantifier scope and token polarity, according to natural logic semantics.
https://stanfordnlp.github.io/CoreNLP/natlog.html

Part(4):Technical

last week I had a form for ‘John is a human?’, but not anymore; no parse tree although we have ‘human’ as nroot and aroot in the dictionary. ‘John is human.’ works though!