Friday 23-03-18

Part(1): Proofs with multi-premises

```
?-startConversation.
?-doItAll(‘John loves Mary.’,X).
?-doItAll(‘John loves a dog.’,X).
?-listing(fact).
fact({['John':'NP'],'#2'}).
fact({['Mary':'NP'],'#3'}).
fact({[tense(present)],'#0'}).
fact({(simple,'#0'),'#1'}).
fact([[love,{dobj,'#3'},{subject,'#2'}],'#1']).
fact({[tense(present)],'#0'}).
fact({(simple,'#0'),'#1'}).
fact({[dog>singular],'#2'}).
fact([[love,{dobj,'#2'},{subject,'#2'}],'#1']).
```

Issue(1):lots of ambiguity. For example, John in both sentences is in a subject position, but when we query ‘John loves John?’ it will be answered because of the second love fact! is it because of the way Skolemn constants generated? or something else?

Issue(2): I’m making startConversation clears everything from the background knowledge, facts and rule, I’ve got a feeling that might not be right? what do you think?

Part(2):rules

Example(1)

```
?-startConversation.
?-doItAll(‘all animals are mortals.’,X).
?-listing(=>).
{[animal>plural],A}=>({[tense(present)],'#0'(B,A)}&({(simple,'#0'(B,A)),'#1'(B,A)}&[[be,{predication(xbar(v(-),n(+))),B},{subject,A}],'#1'(B,A)])=>{mortal>plural,B}).
```

Example(2)

```
?-startConversation.
?-doItAll(‘John loves peaches.’,X).
?-listing(fact).
fact({['John':'NP'],'#2'}).
?-listing(=>).
{[tense(present)],'#0'(A)}&({(simple,'#0'(A)),'#1'(A)}&[[love,{dobj,A},{subject,'#2'}],'#1'(A)])=>{peach
>plural,A}.
```

Issue(3): does these rules look right to you? and should generics have a certain scope?

Part(3):definite references

sorted quantifier stack

```
[name::{[John:NP],A},
 the::{[peach>plural],B},
 existential::{[tense(present)],C},
 existential::{(simple,C),D}]
```

QFF form

```
claim(name(A::{[John:NP],A},
           ({[tense(present)],#0}
             & ({(simple,#0),#1}
                 & [[love,
                     {dobj,ref(B,{[peach>plural],B})},
                     {subject,A}],
                    #1]))))
```

Issue(4):I’m not sure if definite references have been handled properly? And I don’t see how they could be anchored that way

Part(4):more complicated claims; XCOMP

```
?-startConversation.
?-doItAll(‘John loves eating peaches.’,X).
claim(name(A::{[John:NP],A},
           ({[tense(present)],#0}
             & ({(simple,#0),#1}
                 & [[love,
                     xcomp((({[tense(present)],#2(B)}
                              & ({(prog,#2(B)),#3(B)}
                                  & ({zero,#4(B)}
                                      & [[eat,
                                          {dobj,B},
                                          {subject,#4(B)}],
                                         #3(B)])))
                             => {peach>plural,B})),
                     {subject,A}],
                    #1])))))
?-listing(fact).
fact({['John':'NP'],'#5'}).
fact({[tense(present)],'#0'}).
fact({(simple,'#0'),'#1'}).
fact([[love,xcomp({[tense(present)],'#2'(A)}&({(prog,'#2'(A)),'#3'(A)}&({zero,'#4'(A)}&[[eat,{dobj,A},{su
bject,'#4'(A)}],'#3'(A)]))=>{peach>plural,A}),{subject,'#5'}],'#1']).
```

Issue(5):we still have problem with time def(VAR) so I temporarly make (-);existential, is the default.
Issue(6):added to KB properly?