Computational Linguistics
LIN 567/467 – Spring 2018
Project Report

Author : Saleem Ahmed
Computer Science Department
University of Buffalo
Person Number # 50247637

File Structure :

- main.pl : The main file from which the chat predicate loop runs
- lemmaRules.pl : Contains all the lemmas
- lexicons.pl : contains lexical for the lemmas
- phrasalRules.pl : contains the phrasal rules
- parsers.pl : contains the code for the shift right parser
- modelCheckers.pl: consists of two files - models.pl , satisfiers.pl
- models.pl - Contains the meaning representation model for real world knowledge
- satisfiers.pl - the bridge predicates for checking against model truth
- evaluations.pl : evaluates model responses from sat to further responses
- responses.pl - writes output to chat depending on evaluations

Outputs :

Lex :

```
?- lex(X,box).
X = n(_G2706^box(_G2706)) ;
false.

?- lex(X,boxes).
X = n(_G2706^box(_G2706)) ;
false.

?- lex(X,contained).
X = tv(_G2707^_G2710^contain(_G2707, _G2710), []) ;
false.

?- lex(X,contains).
X = tv(_G2707^_G2710^contain(_G2707, _G2710), []) ;
false.

?- lex(X,contain).
X = tv(_G2707^_G2710^contain(_G2707, _G2710), []) ;
false.

?- lex(X,containing).
X = tv(_G2707^_G2710^contain(_G2707, _G2710), []) ;
false.
```

Parser :

```
?- parse([a,blue,box,contains,some,ham],X).
X = s(exists(_G2745, and(and(box(_G2745), blue(_G2745)), exists(_G2832, and(ham(_G2832),
contain(_G2745, _G2832))))), []) ;
false.

?- parse([a,blue,box,contains,ham],X).
X = s(exists(_G2739, and(and(box(_G2739), blue(_G2739)), exists(_G2826, and(ham(_G2826),
contain(_G2739, _G2826))))), []) ;
false.

?- parse([is,there,an,egg,inside,the,blue,box],X).
X = ynq(exists(_G2772, and(egg(_G2772), the(_G2823, and(and(box(_G2823), blue(_G2823)),
inside(_G2772, _G2823))))) ;
X = ynq(exists(_G2772, and(and(egg(_G2772), the(_G2811, and(and(box(_G2811), blue(_G2811)),
inside(_G2772, _G2811)))), sunflower(sunflower)))) ;
```

false.

?- parse([are,there,two,eggs,inside,the,blue,box],X).
X = ynq(two(_G2772, and(egg(_G2772), the(_G2823, and(and(box(_G2823), blue(_G2823)), inside(_G2772, _G2823))))) ;
X = ynq(two(_G2772, and(and(egg(_G2772), the(_G2811, and(and(box(_G2811), blue(_G2811)), inside(_G2772, _G2811)))), sunflower(sunflower)))) ;
false.

?- parse([what,does,the,green,box,contain],X).
X = q(_G2745, and(thing(_G2745), the(_G2767, and(and(box(_G2767), green(_G2767)), contain(_G2767, _G2745))))) ;
X = q(_G2745, and(thing(_G2745), the(_G2767, and(and(box(_G2767), green(_G2767)), contain(_G2767, _G2745))))) ;
false.

?- parse([who,put,every,yellow,box,on,the,white,bowl],X).
X = rc(_G2764^forall(_G2767, imp(and(box(_G2767), yellow(_G2767)), and(put(_G2764, _G2767, _G2770), the(_G2883, and(and(bowl(_G2883), white(_G2883)), on(_G2770, _G2883))))), []) ;
X = rc(_G2764^forall(_G2767, imp(and(box(_G2767), yellow(_G2767)), and(put(_G2764, _G2767, _G2770), the(_G2883, and(and(bowl(_G2883), white(_G2883)), on(_G2770, _G2883))))), []) ;
X = rc(_G2764^forall(_G2767, imp(and(box(_G2767), yellow(_G2767)), the(_G2770, and(and(bowl(_G2770), white(_G2770)), put(_G2764, _G2767, _G2770))))), []) ;
X = rc(_G2764^forall(_G2767, imp(and(box(_G2767), yellow(_G2767)), the(_G2770, and(and(bowl(_G2770), white(_G2770)), put(_G2764, _G2767, _G2770))))), []) ;
X = q(_G2763, and(person(_G2763), forall(_G2783, imp(and(box(_G2783), yellow(_G2783)), and(put(_G2763, _G2783, _G2786), the(_G2899, and(and(bowl(_G2899), white(_G2899)), on(_G2786, _G2899))))))) ;
X = q(_G2763, and(person(_G2763), forall(_G2783, imp(and(box(_G2783), yellow(_G2783)), and(put(_G2763, _G2783, _G2786), the(_G2899, and(and(bowl(_G2899), white(_G2899)), on(_G2786, _G2899))))))) ;
X = q(_G2763, and(person(_G2763), forall(_G2783, imp(and(box(_G2783), yellow(_G2783)), the(_G2786, and(and(bowl(_G2786), white(_G2786)), put(_G2763, _G2783, _G2786)))))) ;
X = q(_G2763, and(person(_G2763), forall(_G2783, imp(and(box(_G2783), yellow(_G2783)), the(_G2786, and(and(bowl(_G2786), white(_G2786)), put(_G2763, _G2783, _G2786)))))) ;
false.

Chat :

|: Every blue container on the top shelf contains a sandwich that has no meat
That is not correct, negative, out of the question,
Explanation : no sandwich defined in model.

|: What does the yellow bowl on the middle shelf contain
[[_G6910059,ba1]]
Explanation : ba1 is defined as banana contained in yellow bowl on middle shelf, To Do : reverse search using f(Symbol, Value) and return symbol banana instead of ba1

|: Are there two watermelons in the fridge
no siree, not on your life, thumbs down
Explanation : I have no explaination

|: is there milk
Yes

Who drank the almond milk
That is not correct, negative, out of the question
Explanation : Model is not handling who questions

|: Is there a sandwich that does not contain meat
no siree, not on your life, thumbs down
Explanation : There is no such sandwhich

|: Is there an empty box of popsicles in the freezer
no siree, not on your life, thumbs down
Explanation : no such worldy knowledge is known by model.

|: is there an egg inside the blue box
yes

|: are there two eggs inside the blue box
[_G401,b1]
yes

?-modelchecker(s(forall(_G387,if(table(_G387),exists(_G395,and(and(box(_G395),black(_G395)),
contain(_G395,_G387))))),[]), Result).
Result = [not_true_in_the_model] ;
false.

modelchecker(q(_G234,and(thing(_G234),exists(_G235,and(and(box(_G235),green(_G235)),
contain(_G235,_G234))))), Result).
Result = []

Custom :
|: is there a watermelon
yes

|: are there two watermelon
[_G447,w1]
yes

|: is there a green box
yes

|: is there a red box
no siree, not on your life, thumbs down

|: Are there two eggs
[_G204,e1]
yes

|: are there ten eggs inside the blue box
no siree, not on your life, thumbs down

|: who put the yellow box on the yellow bowl

That is not correct, negative, out of the question,