

ComS 342  
Recitation 2, 10:00 Tuesday  
Homework 9

Sean Gordon

December 4, 2019

---

1)  $Z = [1, 4, 6, 3, 6]$ .

---

2a)

fib(0, 0).  
fib(1, 1).

fib(N, Res) :-  
  N > 1,  
  N1 is N-1,  
  N2 is N-2,

  fib(N1, R1),  
  fib(N2, R2),

  Res is R1+R2.

2b)

rev(L, Res) :-  
  revHelper(L, [], Res).

  revHelper([], Accum, Accum).  
  revHelper([H|T], Accum, Res) :-  
    number(H),  
    revHelper(T, [H|Accum], Res);

  revHelper(H, [], Temp),  
  revHelper(T, [Temp|Accum], Res).

3a)

```
sentence([ ]).  
sentence(S) :- s(S, [ ]).
```

```
s(L1, L2) :- f(L1, L2).  
s(L1, L4) :- t(L1, L2), n(L2, L3), t(L3, L4).
```

```
f(L1, L5) :- if(L1, L2), b(L2, L3), then(L3, L4), s(L4, L5).  
f(L1, L7) :- if(L1, L2), b(L2, L3), then(L3, L4), s(L4, L5), else(L5, L6), s(L6, L7).
```

```
b(L1, L4) :- t(L1, L2), e(L2, L3), t(L3, L4).
```

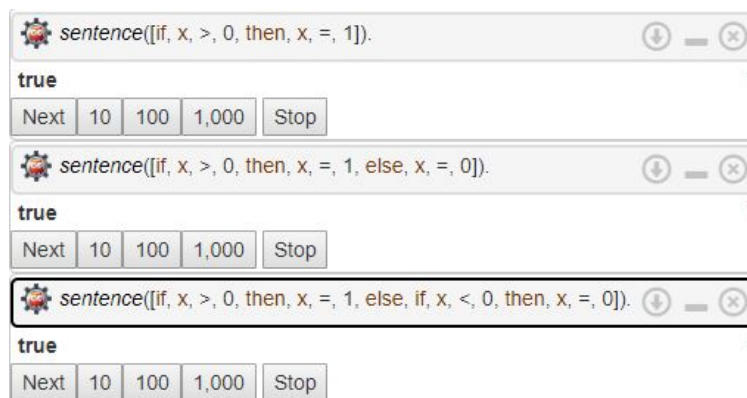
```
if([if|Tail], Tail).  
then([then|Tail], Tail).  
else([else|Tail], Tail).
```

```
t([x|Tail], Tail).  
t([y|Tail], Tail).  
t([z|Tail], Tail).  
t([1|Tail], Tail).  
t([0|Tail], Tail).
```

```
e([<|Tail], Tail).  
e([>|Tail], Tail).
```

```
n([+|Tail], Tail).  
n([-|Tail], Tail).  
n([=|Tail], Tail).
```

3b) sentence(X)? Exceeds memory due to  $S \leftrightarrow F$  recursion. I'm confused.



3c) Yes, sub-goal order matters as they are executed sequentially.

---

```

4)
% House: [Color, Man, Pet, Drink, Cig]
next(A, B, L):-nextto(A, B, L) ; nextto(B, A, L).
write_list(L):-forall(member(Mem, L), (write(Mem), nl)).

start(Houses):- length(Houses, 5),
member([red, english, -, -, -], Houses),
member([-, spaniard, dog, -, -], Houses),
member([green, -, -, coffee, -], Houses),
member([-, ukrainian, -, tea, -], Houses),

next([ivory, -, -, -, -], [green, -, -, -, -], Houses),

member([-, -, snails, -, oldGold], Houses),
member([yellow, -, -, -, kools], Houses),

Houses= [-, -, [-, -, -, milk, -], -, -],
Houses= [[-, norwegian, -, -, -], -, -, -, -],

next([-, -, -, -, camel], [-, -, fox, -, -], Houses),
next([-, -, -, -, kools], [-, -, horse, -, -], Houses),

member([-, -, -, orangeJuice, luckyStrike], Houses),
member([-, japanese, -, -, parliaments], Houses),

next([-, norwegian, -, -, -], [blue, -, -, -, -], Houses),

member([-, -, -, water, -], Houses),
member([-, -, zebra, -, -], Houses),

write_list(Houses).

```

After running “start(S),fail.”, the below list is printed:

```

[yellow, norwegian, fox, water, kools]
[blue, ukrainian, horse, tea, camel]
[red, english, snails, milk, oldGold]
[ivory, spaniard, dog, orangeJuice, luckyStrike]
[green, japanese, zebra, coffee, parliaments]

```

Therefore, the norwegian drinks water, and the japanese owns the zebra.