yxu66

CS 662
Artificial Intelligence Programming Homework #4
First Order Logic, Probabilities and NLP

1.

a. ∀x mortal(x)^ holding(x, Ring) ==> tempted(x)
b. hobbit(Frodo)
c. ∀x hobbit(x) ==> mortal(x)
d. ∀x tempted(x) ==> putOn(x, Ring)
e. ~ holding(Frodo, Ring) ==> holding(Gandalf, Ring)
f. ~ holding(Gandalf, Ring)

2.

--> use: {x/Frodo} && b. hobbit(x) && c. ∀x hobbit(x) ==> mortal(x)
get: g. mortal(Frodo)
--> use: e.~ holding(Frodo, Ring) ==> holding(Gandalf, Ring)
get: h. ~ holding(Gandalf, Ring) ==> holding(Frodo, Ring)
--> use: f.~ holding(Gandalf, Ring) &&
h. ~ holding(Gandalf, Ring) ==> holding(Frodo, Ring)
get: i. holding(Frodo, Ring)
--> use: {x/Frodo} && g. mortal(Frodo) && i. holding(Frodo, Ring)
&& a. ∀x mortal (x) ^ holding(x, Ring) ==> tempted(x)
get: j. mortal(Frodo) ^ holding(Frodo, Ring) ==> tempted(Frodo)
--> use: {x/Frodo} && j. tempted(Frodo) && d. ∀x tempted(x) ==> putOn(x, Ring)
get: k. tempted(Frodo) ==> putOn(Frodo, Ring)

3.

--> To prove: 1. putOn(Frodo, Ring)
--> 1 unifiers with d. To prove:
2. tempted(Frodo)
--> 2 unifiers with a. To prove:
3. mortal (Frodo) , 4. holding(Frodo, Ring)
--> 3 unifiers with c. To prove:
5. hobbit(Frodo), 4. holding(Frodo, Ring)
--> 5 unifiers with b. To prove:
4. holding(Frodo, Ring)
--> 4 unifiers with e. To prove:
6. ~holding(Gandalf, Ring)
--> 6 unifiers with f

The list of goals is empty, so we are done.

4.

1. ~mortal(x) v ~holding(x, Ring) v tempted(x)
2. hobbit(Frodo)
3. ~hobbit(x) v mortal(x)
4. ~tempted(x) v putOn(x, Ring)
5. holding(Frodo, Ring) v holding(Gandalf, Ring)
6. ~holding(Gandalf, Ring)
7. ~putOn(Frodo, Ring)

--> Resolve 4 && 7. Add:
8. ~tempted(Frodo)
--> Resolve 1 && 8. Add:
9. ~mortal(x) v ~holding(x, Ring)
--> Resolve 2 && 3. Add:
10. mortal(Frodo)
--> Resolve 5 && 6. Add:
11. holding(Frodo, Ring)
--> Resolve 9 && 10. Add:
12. ~holding(Frodo, Ring)
--> Resolve 11 && 12. Contradiction!

So putOn(Frodo, Ring) is true.

5.

a. P(toothache) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2
b. P(Cavity)= 0.108 + 0.012 + 0.072 + 0.008 = 0.2
c. P(Toothache|cavity) = P(Toothache,cavity) / P(cavity)
= (0.108 + 0.012) / 0.2 = 0.6
d. P(Cavity|toothache ∨ catch) = P(Cavity ∧ (toothache ∨ catch)) / P(toothache ∨ catch)
Since:
P(Cavity ∧ (toothache ∨ catch)) = 0.108 + 0.012 + 0.072 = 0.192
P(toothache ∨ catch) = 0.108 + 0.016 + 0.012 + 0.064 + 0.072 + 0.144 = 0.416

So P(Cavity|toothache ∨ catch) = 0.192 / 0.416 = 0.462

6.

We know:
P(Forget | Complete) = 0.01
P(Forget | ~Complete) = 0.5
P(Complete) = 0.9

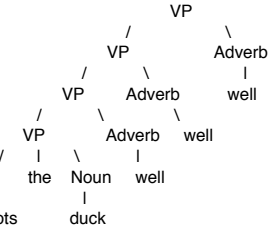
We want to know: P(Complete | Forget)
P(Complete | Forget) = (P(Forget | Complete) * P(Complete)) / P(Forget)

From P(Forget | Complete) = 0.01
From P(Forget | ~Complete) = 0.5
And P(Forget) = P(Forget | Complete) * P(Complete) + P(Forget | ~Complete) * P(~Complete)
= 0.01 * 0.9 + 0.5 * 0.1 = 0.059
So P(Complete | Forget) = (0.01 * 0.9) / 0.059 = 0.15

7.

0.1:VP →Verb
0.2 : VP → Copula Adjective
0.5: VP → Verb the Noun
0.2 : VP → VP Adverb
0.5 : Verb → is
0.5 : Verb → shoots
0.8 : Copula → is
0.2 : Copula → seems
0.5 : Adjective → unwell
0.5 : Adjective → well
0.5 : Adverb → well
0.5 : Adverb → badly
0.6 : Noun → duck
0.4 : Noun → well

a. (i) shoots the duck well well well :



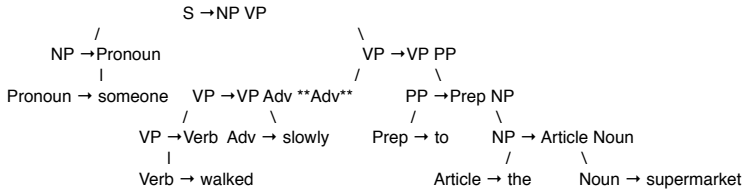
So it's nonzero probability.
(ii) seems the well well:
Copula → seems, so the only VP is VP → Copula Adjective, cannot followed by "the".
So cannot parse.
(iii) shoots the unwell well badly:
VP → Verb the Noun is the only VP that within "the", but Adjective → unwell, not Noun.
So cannot parse.

b. 1. P("is well well") = P(VP → VP Adverb) * P(VP → VP Adverb) * P(VP →Verb) * P(Verb → is) *
P(Adverb → well) * P(Adverb → well)
= 0.2 * 0.2 * 0.1 * 0.5 * 0.5 * 0.5 = 0.0005
2. P("is well well") = P(VP → VP Adverb) * P(VP → Copula Adjective) * P(Copula → is) *
P(Adjective → well) * P(Adverb → well)
= 0.2 * 0.2 * 0.8 * 0.5 * 0.5 = 0.008
So P("is well well") = 0.008 + 0.0005 = 0.00085

c. It's syntactic. Since we have two parse tree for the phrase, so it's syntactic ambiguity.
d.Yes, it's possible. We can find all the trees that with 10-word sentences and then add up the probability.

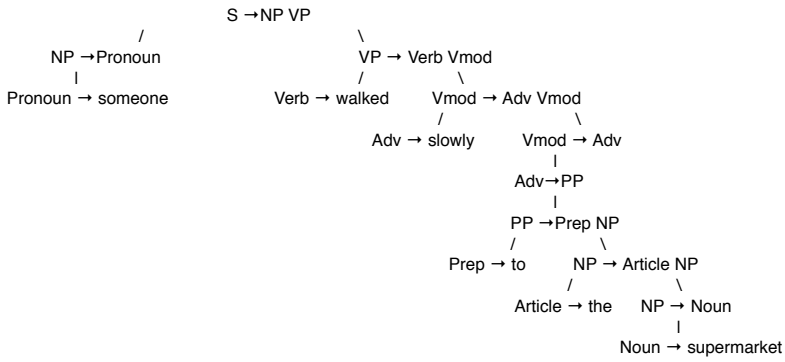
8.
Pronoun → someone
Adv → slowly
Article → the
Verb → walked
Prep → to
Noun → supermarket

(A):
S →NP VP
NP →Pronoun
NP → Article Noun
VP →VP PP
VP →VP Adv Adv
VP →Verb
PP →Prep NP
NP →Noun



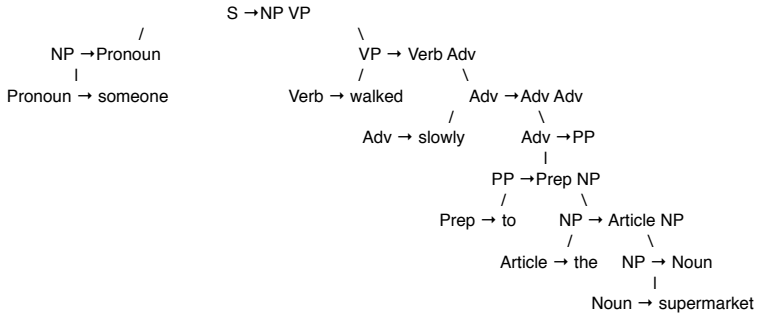
In the place where is marked with '**', there is one more Adv. So it cannot generates the given sentence.

(B):
S→NP VP
NP → Pronoun
NP → Noun
NP → Article NP
VP → Verb Vmod
Vmod → Adv Vmod
Vmod → Adv
Adv→PP
PP →Prep NP



So it can generates the given sentence.

(C):
S →NP VP
NP →Pronoun
NP → Article NP
VP → Verb Adv
Adv →Adv Adv
Adv →PP
PP →Prep NP
NP →Noun



So it can generates the given sentence.

(d)Writting sentence:

Pronoun → someone
Adv → happily, early, very, well
Article → the
Verb → played, goes, flied, barks
Prep → in, to, into, on
Noun → dog, yard, morning, bird, sky, beach, work

(A)English:

The dog played very happily in the yard.
Someone goes to the beach very early.
The bird flied into the sky.

Non-English:
The dog goes happliy early very well in the beach.
The bird flied to morning into work.
Someone played in the yard very happily well early to the morning.

(B)English:
The dog goes to the yard early.
Someone goes to work very happily.
The bird flied very early into the yard.

Non-English:
The someone goes early happily well very.
The the the morning played happily.
The the the the work flied well very happily early.

(C)English:
The bird played very happily on the beach.
Someone goes very early to the beach in the morning.
The dog barks early in the morning.

Non-English:
The sky flied well early very happily in the work.
The work goes very well early happily in the yard on the beach into the sky.
The Someone barks to the dog into the work on the bird in the sky.

Suggestion:
All of these three grammers are easy goes into endless loop. For example:
(A) VP →VP PP and VP →VP Adv Adv
(B) VP → Verb Vmod with Vmod → Adv Vmod
(C) VP → Verb Adv with Adv →Adv Adv and VP → Verb Adv with Adv →Adv Adv, Adv →PP
Also, in (B) and (C), there is easy to have NP -> Atricle Pronoun, which is wrong grammer.

So we can do:
(A) Change VP →VP PP to VP1 →Verb PP
Change VP →VP Adv Adv to VP2 →Verb Adv Adv
Add: VP →VP1
VP →VP2
(B) Change: NP → Pronoun to NP1 → Pronoun
Change: NP → Noun to NP2 → Noun
Change: NP → Article NP to NP → Article NP2
Add: NP → NP1

Change: Vmod → Adv Vmod to Vmod → Adv Vmod1
Change: Vmod → Adv to Vmod1 → Adv

(C) Change: NP → Pronoun to NP1 → Pronoun
Change: NP → Noun to NP2 → Noun
Change: NP → Article NP to NP → Article NP2
Add: NP → NP1

Change: Adv →Adv Adv to Adv → Adv1 Adv1
Change: Adv →PP to Adv1 →PP
Add: Adv -> Adverb