CS/ENGRI 172, Fall 2002

11/22/02: Homework Six

You taught me language; and my profit on't
Is, I know how to curse.
– Caliban, *The Tempest*

Due at the beginning of class on Friday, December 6. Hand in the parts separately, with your name on each part. Answers will be graded on both correctness and clarity. In general, you should always include an explanation of your answers when appropriate (among other things, this helps in assigning partial credit).

Part A

1. Consider the following PDA P (self-check: we guarantee that P accepts the sentence abc):

States: $\text{cyc1}, \text{cyc2}, \text{count-c}$
Initial state: cyc1
Accept state: count-c
Input symbols: a, b, c, $\dashv$
Stack symbols: $\pm$, B
Moves:
Rule 1. $(\text{cyc1}, a, \pm) \longrightarrow (\text{cyc2}, \pm)$
Rule 2. $(\text{cyc1}, a, B) \longrightarrow (\text{cyc2}, B)$
Rule 3. $(\text{cyc1}, c, B) \longrightarrow (\text{count-c}, \text{pop})$
Rule 4. $(\text{cyc2}, b, \pm) \longrightarrow (\text{cyc1}, B\pm)$
Rule 5. $(\text{cyc2}, b, B) \longrightarrow (\text{cyc1}, BB)$
Rule 6. $(\text{count-c}, c, B) \longrightarrow (\text{count-c}, \text{pop})$
Rule 7. $(\text{count-c}, \dashv, \pm) \longrightarrow (\text{count-c}, \text{pop})$

(a) Does P accept the sentence $abcabc$? Explain; include a drawing of the sequence of configurations P enters on this input.

(b) Give a precise characterization of the exact set of sentences that P accepts. Explain your answer; include a description of the roles played by the states and rules (e.g., "cyc1 stores double the number of the sentence's a's on the stack, as shown by Rules 1 and 2").

(c) Here is a similar PDA P' (formed by essentially collapsing cyc1 and cyc2 in P):

States: $\text{cyc}', \text{count-c}'$
Initial state: cyc'
Accept state: $\text{count-c}'$
Input symbols: a, b, c, $\dashv$
Stack symbols: $\pm$, B
Moves:
Rule 1′. $(\text{cyc}', a, \pm) \longrightarrow (\text{cyc}', \pm)$
Rule 2′. $(\text{cyc}', a, B) \longrightarrow (\text{cyc}', B)$
Rule 3′. $(\text{cyc}', c, B) \longrightarrow (\text{count-c}', \text{pop})$
Rule 4′. $(\text{cyc}', b, \pm) \longrightarrow (\text{cyc}', B\pm)$
Rule 5′. $(\text{cyc}', b, B) \longrightarrow (\text{cyc}', BB)$
Rule 6′. $(\text{count-c}', c, B) \longrightarrow (\text{count-c}', \text{pop})$
Rule 7′. $(\text{count-c}', \dashv, \pm) \longrightarrow (\text{count-c}', \text{pop})$

Does P' accept the exact same set of sentences as P (i.e., was P's state cyc2 necessary)? Explain.

2. Mosteller and Wallace, in attempting to determine the authorship of the disputed Federalist papers, used the following log-odds ratio in their considerations:

$$LOR(d) = \sum_{i=1}^{n} \log \frac{P_H(v_i)}{P_M(v_i)}$$

(recall that the v_i's are indicator words occurring in document d). A seemingly similar alternative would have been to use the following *odds ratio*:

$$OR(d) = \sum_{i=1}^{n} \frac{P_H(v_i)}{P_M(v_i)}$$

One way to weigh the relative merits of these two functions is to consider the question of how to actually make an authorship determination based on them. Give a plausible decision rule for LOR and explain why it is reasonable. (An implausible decision rule would be, "if $LOR(d)$ is an integer, choose Madison, otherwise, choose Hamilton".) Can you give an analogous decision rule for OR? Justify your answer.

Part B

3. In this question, we look at the TANGO segmentation algorithm more carefully. To facilitate our investigation, we simplify the algorithm and also transpose the setting to English.

Pretend that all the spaces and punctuation have been removed from all English text (and all upper-case letters converted to lower-case, to make the situation more analogous to Japanese), and you have been asked to apply the TANGO algorithm to restore the word boundaries. However, you are to use a simple version of the algorithm: the threshold has been set to $t = 0.25$, you are only to use bigram comparisons, and you are only to use the threshold condition (so a local maximum does *not* automatically trigger a word boundary).

Suddenly, it occurs to you that there might be a problem. In your unsegmented training data, every time a "q" occurs, it is *always* followed by a "u". However, in the data that you are applying the algorithm to, you see the following subsequence:

...twentyyearsagoiraqunderwentamajor...

Must the simplified TANGO algorithm fail to place a boundary between the "q" and the "u"? If so, explain why. If not, give a *concrete example* of plausible letter bigram counts from the unsegmented training data such that the algorithm would place a boundary between the "q" and the "u", making sure to explain why in your example the algorithm would determine that a space should be inserted and why your counts could plausibly arise in English text. By "concrete example", we mean you should say something like, "If "qu" occurs 50 times ..." and similarly for other relevant bigrams.

Note: first convince yourself that the letter bigram "aq" *cannot* be more frequent in your unsegmented training data than the letter bigram "qu", according to the conditions given in this problem.

4. Recall the "Miller's monkey" model of word generation. Suppose we alter the setting to the following. This time, the typewriter has just two keys: the "a" key, which the monkey strikes with probability p (where p is some constant such that $0 < p < 1$), and the return key, which the monkey strikes with probability $1 - p$. As before, the monkey's keystrokes are independent of each other.

For this simplified case, explicitly compute as a function of p and i what the probability of the ith ranked word is, showing and explaining your work. (You should probably follow the steps from class.) Does a power law relationship between rank and probability arise from this two-key setting? Explain your answer.

Part C

5. This question considers some details regarding the Grosz and Sidner discourse model. In particular, according to the original Grosz and Sidner (1986) paper, the stack has what we might call *transparency*: "information in lower spaces is usually accessible from higher ones (but less so than the information in the higher spaces)" (pg. 180). So, the focus stack provides an *ordering* on salient entities, but entities in focus spaces below the top focus space can be referred to.[1] As you know, this is *not* the case for the stacks our PDAs use.

In the following (apocryphal) dialog, we have a travel agent (A) and a caller (C). We have also indicated a possible analysis of the linguistic structure of the discourse: discourse segments are indicated by brackets and bear italic labels (e.g., *DS1*).

> *DS1*
>
> C : I want to go from Chicago to Hippopotamus, New York.
>
>> *DS2*
>>
>>> *DS3*
>>>
>>> A : Are you sure that's the name of the town?
>>>
>>>> *DS4*
>>>>
>>>> C : Yes,
>>>>
>>>> C : what flights do you have?
>>
>>> *DS5*
>>>
>>> A : I'm sorry,
>>>
>>> A : but I've looked up every airport code in the country
>>>
>>> A : and I can't find a Hippopotamus anywhere.
>>
>>> *DS6*
>>>
>>> C : Oh don't be silly.
>>>
>>> C : Everyone knows where it is.
>>>
>>> C : Check your map!
>>
>>> *DS7*
>>>
>>> A : You don't mean Buffalo, do you?
>>>
>>>> *DS8*
>>>>
>>>> C : That's it!
>>>>
>>>> C : I knew it was a big animal!

(a) Is it possible that DSP1 is, "(on the part of the caller) Secretly determine whether the travel agent is a human or a computer"? Explain your answer.

(b) Either give the intentional structure of this dialog that corresponds to the linguistic segmentation provided above (there may be multiple possible answers) — and if you do so, explain your analysis — or explain why no such intentional structure exists.

[1] Note that this provides an explanation for why, in the Kasparov dialog, we knew that the second appearance of the phrase "psychological pressure" was a repetition.

(c) In accordance with the linguistic segmentation above (and the intentional structure you provided, if you determined that one existed), draw what the focus stack looks like just after the utterance of the line, “That’s it!”, explaining why your stack is correct according to the Grosz and Sidner theory. What should that “it” refer to, according to the focus stack you have drawn? Do you need to assume “stack transparency” in order for “it” refer to what you predict? Explain your answers.

6. The statistical machine-translation algorithm we discussed *bootstraps*[2] itself purely from data — it has no prior knowledge of language whatsoever. However, it might be handy to be able to incorporate into the algorithm some language information: this could help prevent it from learning poor translations, and could reduce the number of iterations it undergoes.

Suppose you know that it is impossible for source-language word s to be translated as target-language word t. Does there exist a simple change that can be made *just to the initialization step* of the algorithm that will guarantee that once the algorithm has converged, the translation weight $\mathrm{tr}(s \rightarrow t)$ will be zero? If so, describe the change and explain why it works. If not, explain why changing the initialization step does not suffice.

[2] The term is derived from the expression “Pull oneself up by one’s bootstraps”.