HW 2: Nod what ice Ed

Re-submit Assignment

Due Feb 9 by 11:59pm **Points** 100 **Submitting** a file upload
File Types pdf and pl

In this assignment you will submit four files:

1. a PDF file with the answer to question 1 and pictures of FSTs for questions 2, 3 and 4 (hw2.pdf).

2. a prolog file called hw2q2.pl

3. a prolog file called hw2q3.pl

4. a prolog file called hw2q4.pl

The contents of each file are specified below.

(On the Pages section of canvas there is an example of what the submission should look like.)

Part 1: Buy Pie

As we saw in HW 1, sequences of phonemes can correspond to many different sequences of words. Although this does not bother us at all as we listen to speech, the problem faced by machines is even worse than it seems, because some phonemes sound alike. Consider the following pairs:

- [p] and [b]
- [k] and [g]
- [s] and [z]

QUESTION 1: How does the first item in each pair above differ form the second?

(Hint: if you are unsure, read about *phone properties* in the Ojeda textbook on page 57, and consult pages 335-337. Or look closely at the IPA chart: **https://upload.wikimedia.org/wikipedia/commons/8/8e/IPA_chart_2018.pdf (https://upload.wikimedia.org/wikipedia/commons/8/8e/IPA_chart_2018.pdf)** .)

Draw an FST that changes to and from [p] and [b], [k] and [g], and [s] and [z] in CMU pronunciation dictionary transcriptions. The input alphabet is the set of phonemes listed under phoneme set in the **CMU pronunciation dictionary page (http://www.speech.cs.cmu.edu/cgi-bin/cmudict)** (e.g. AA, AE, AH, ...). The output alphabet is exactly the same as the input alphabet. Each string in the input language should be mapped to one or more similar string(s) in the output language, except that each B can remain a B or become a P, P can remain a P or become a B, and so on with K/G and S/Z.

(Hint 1: this type of FST is sometimes referred to as the *flower* FST!)

For example, here is the entry for the word *buy*:

```
BUY   B AY
```

The input string

```
B AY
```

corresponds to the output strings

```
B AY
```

and

```
P AY
```

where each results from a different path in the non-deterministic FST.

QUESTION 2:

a. Submit your FST (specified above) in a prolog file (.pl) using the prolog format used in HW 1. Name the file hw2q2.pl.

b. Add a picture of your FST diagram (specified above) to your PDF submission.

(Hint 2: In the prolog format, we simply list each transition as follows:

```
transition(State1, InputSymbol, State2, OutputSymbol).
```

For example, to go from state 15 to state 20, with input symbol a and output symbol x, we write:

```
transition(15, a, 20, x).
```

What does the following transition do?

```
transition(20, Sym, 26, Sym).
```

where 20 and 26 could be any pair of states. Notice that Sym is capitalized, which means it is a variable.

And what about:

```
transition(20, Sym, 26, Sym) :- Sym \= eps.
```

Recall the discussion about prolog rules. This says that the transition exists if Sym is not eps (but it can be anything else!).

You don't need to comment on this in your submission, but think about what this transition does and why. This type of transition is not strictly necessary in your solution, but will save you some time!)

Consider now a similar FST, but one that allows only at most one substitution at a time. For example, with the input:

```
B AE G
```

Your FST from question 2 can produce

```
B AE G
```

```
P AE G
```

```
B AE K
```

```
P AE K
```

But your new FST can only produce

```
B AE G
```

```
P AE G
```

```
B AE K
```

(In prolog, we will use lowercase characters, as in HW 1.)

In other words, you can have zero substitutions or one substitution, but no more.

QUESTION 3:

a. Submit your FST in a prolog file (.pl) using the prolog format used in HW 1. Name the file hw2q3.pl.

b. Add a picture of your FST diagram to your PDF submission.

(Hint: this will not be a flower FST. Your machine will now have to keep track of whether a substation has occurred. How do we encode that it is ok to have no substitutions, and it is ok to have exactly one substitution?)

Part 2: Aspire

You may have noticed that the t sound is a little different in the words *top* and *stop*, and the p sound is also a little different in *pit* and *spit*. This is because of aspiration. In top and pit, there is a stronger burst of breath that comes out with the first consonant sound.

A (simplistic) phonological rule for aspiration in English goes as follows:

Voiceless plosives are aspirated when they occur immediately before a stressed vowel, and there is no [s] immediately preceding the voiceless plosive.

For example, in pot [p^{h}ɑt] we get aspiration, but not in spot [spɑt].

Let's now consider a version of the CMU pronunciation dictionary that has stress marked in the transcriptions. For example, the word potato looks like

```
P AH T EY1 T OW
```

where the 1 appended to EY means that the vowel is stressed. Every vowel can be unstressed, e.g. AA, EY, OW, or stressed, e.g. AA1, EY1, OW1 (this is a slight simplification of how stress is represented in the standard CMU pronunciation dictionary.)

Create an FST that adds aspiration to transcriptions according to the rule above. The input alphabet is the set of symbols in the ARPABET plus the stressed vowels ending in 1. The output alphabet is the same, plus additional symbols as follows: each symbol that corresponds to a voiceless plosive gets a duplicate with _h appended to it. For example, the symbol T is in the output alphabet, and in addition, we also have the new symbol T_h in the output alphabet.

Step 1: Find the CMU dictionary ARPABET symbols that correspond to voiceless plosives (consult the Ojeda book if needed, or the IPA chart).

Step 2: Consider now the expanded output alphabet based on step 1.

Step 3: Create a transducer that implements the phonological rule above and produces output with the new symbols that reflect aspiration when appropriate. The input is the transcription as produced by a transducer like cmudict.pl from HW 1, but with the additional symbols to denote stress (e.g. ay1, ow1, ey1), as shown above. The output should be the transcription with aspiration.

To create the transducer in prolog format, we make all of the symbols lowercase. For example, if our input is

```
[p, ah, t, ey1, t, ow]
```

the output should be

```
[p, ah, t_h, ey1, t, ow]
```

where the t_h is the aspirated allophone of t.

Notice that the first t is aspirated, but the p and second t are not.

(It is fine if your transducer maps the input string to itself, in addition to the aspirated version. However, it is not fine if your transducer creates aspirated plosives where it is not appropriate according to the rule above.)

QUESTION 4:

a. Submit your FST as a file in prolog format. Call it hw2q4.pl.

b. Add a picture of your FST diagram to your PDF submission.