Solutions to selected exercises

Note: These solutions are provided in order to make sure that you don't get stuck, and consequently unable to make progress with the course. Teachers as well as students know that this page exists. Many colleges and universities have specific rules against plagiarism, and provision for taking disciplinary action against students that commit **plagiarism,** i.e. including words or ideas from someone else in your work without proper attribution, so as to intentionally (mis)lead the reader to believe that the words or ideas are your own. But don't be paranoid.

Exercise 2.3. The solution is given on p. 22.

Exercise 2.4. The solution is given on p. 21.

Exercise 2.6. The sampling rate of `joe.wav` is 16000 samples per second, whereas `cosine.dat` was at 8000 samples per second.

Exercise 2.7. $2^{15} = 32768$.

Exercise 2.8. The effect of the change is to print the sequence of samples out as a list of numbers on the computer screen, rather than generating a signal file.

Exercise 3.1.

No. of samples *m* in the moving average	Low pass filter cut–off frequency (Hz) (for an input signal with 16000 samples/s)
2	8000
3	5333
4	4000
5	3200
6	2666
7	2286
8	2000
10	1600
12	1333
16	1000
20	800
24	667
30	533
40	400
50	320
80	200
120	133
160	100

Exercise 3.2. 80 samples.

Exercise 3.3. `meansof80.c`.

Exercise 3.4. A 3–sample window will be acceptable. (A window of 2.909 samples is impossible to set up, note.)

Exercise 4.4. Get `lpcana2.c` from here.

Exercise 4.5. Here are my estimates. If yours are a little different, that doesn't mean they're wrong. Note that `joe.dat` has a sampling rate of 16000 samples/s, not the 8000 samples/s that `lpc_spectrum` expects. So, try applying `lpc_spectrum` to `joe8k.dat` instead (and halve the sample numbers in columns 5 and 6 below). A blank worksheet is available here.

						Measured from `joe8k.dat` using `lpc_spectrum`						Olive et al. (1993) (American male)						Cruttend (Britis	
		Sample numbers				¼ vowel			¾ vowel			¼ vowel			¾ vowel			First component	
Vowel		Vowel start	Vowel end	¼ vowel	¾ vowel	F_1 (Hz)	F_2 (Hz)	F_3 (Hz)	F_1 (Hz)	F_2 (Hz)	F_3 (Hz)	F_1 (Hz)	F_2 (Hz)	F_3 (Hz)	F_1 (Hz)	F_2 (Hz)	F_3 (Hz)	F_1 (Hz)	F_2 (Hz)
(J)**oe**	[ə ʊ]	3450	4950	3825	4575	296	1281	2125	200	1156	2226	550	1000		400	800		537	1266
(t)**oo**(k)	[ʊ]	5430	5830	5523	5708	?	1312	2101	?	1125	2187	450	950	2400	450	950	2400	414	1051
(f)**a**	[ɑ]	7020	8390	7363	8048	?	?	2406	375	984	2671	750	1100	2600	750	1100	2600	687	1077
(th)**er**(s)	[ə]	8750	9550	8950	9350	218	1234	2562	148?	1515	2609	500	1500	2500	500	1500	2500		
(sh)**oe**	[ju]	10850	11730	11070	11510	195	2046	3218? (F_4?)	156	1359	2093	300	1800		300	850	2400		
(b)**e**(nch)	[ɛ]	12090	12910	12295	12705	320	1375	2078	?	1640	2328	550	1700	2550	550	1700	2550	494	1650
ou(t).	[æʊ]	13840	14975	14124	14692	398	1406	2281	468	1195	2296	800	1350		550	900		780	1368
(Sh)**e**	[i]	18605	19140	18739	19007	150?	2468	3218? (F_4?)	?	2171	3375? (F_4?)	280	2250	2750	280	2250	2750	280	2249
(w)**a**(s)	[ə]	19570	20165	19719	20017	210	1242	2421	?	1492	2843	500	1500	2500	500	1500	2500		
(w)**ai**	[eɪ]	21665?	22175	21793	22048	?	2218? (F_3?)	3343? (F_4?)	?	2390	?	500	1750		>250	2250		587	1945
(t)**i**(ng)	[ɪ]	22900	23475	23044	23332	?	2437	3812? (F_4?)	?	2140	2593?	400	1900	2600	400	1900	2600	367	1757
a(t)	[ə]	25193	25678	25314	25557	539	1437	2296	?	1375	2343	500	1500	2500	500	1500	2500		
(m)**y**	[ɑ ɪ]	27165	27990	27372	27784	484	1171?	2343	367	1390	2390	800	1300		>250	1950		734	1117
(l)**aw**(n).	[ɔ]	28830	30350	29210	29970	171?	?	2539	600?	?	2367	650	900	2500	650	900	2500	415	828

Exercise 5.2.

```
?- accept([s,p,r,e,'N',s,t]).
Yes

?- accept([s,p,l,'O',n,d]).
Yes

?- accept([s,t,r,0,l,k,t]).
Yes

?- accept([t,r,'V',l,t,'T']).
Yes

?- accept([b,l,e,m]).
Yes

?- accept([t,l,'U',i,m,v]).
No

?- accept([s,g,r,'I',n,t]).
No

?- accept([s,p,r,'I',m,'T']).
```

```
No
```

Exercise 5.3.

```
?- accept([s,p,l,O,n,d]).
```

gives the answer

```
O = 'I'
```

Pressing the semicolon key gives another answer:

```
O = e
```

Pressing it 11 more time gives:

```
O = & ;

O = 'V' ;

O = 0 ;

O = 'U' ;

O = 3 ;

O = 'A' ;

O = 'O' ;

O = u ;

O = i ;

O = @ ;

No
```

In other words, Prolog finds all the single vowels that can occur between **spl** and **nd**, according the grammar implicit in NFSA1.

Exercise 5.4.

```
?- accept(X).
```

gives the result:

```
X = [s, p, r, 'I', @]
```

Exercise 5.5.

Add other currency symbols to the set {£}, to make it e.g. {£, $, ¢, ¥, €}. The problem with searching for prices in pence is that the abbreviation for pence, p, is an ordinary letter of the alphabet, and thus occurs in almost any text.

Figure 6.1 is a sentence of Tashlhiyt Berber. It says [ini za $g^{w}bi\hbar$ yat tklit adni]

Exercise 6.1.

d:

1	5	5	4	4	4	5	5
2	4	4	3	5	5	4	4
3	3	3	2	6	6	3	3
4	2	2	1	7	7	2	2
5	1	1	0	8	8	1	1
5	1	1	0	8	8	1	1
1	5	5	4	4	4	5	5
1	5	5	4	4	4	5	5
	6	6	5	−3	−3	6	6

accdist:

26	26	20	20	22	27	31
21	21	16	18	22	26	30
17	17	13	17	24	27	29
14	14	11	17	24	26	26
12	12	10	18	26	24	24
11	11	10	18	26	23	24
10	10	14	18	22	27	32
5	10	14	18	22	27	32

move:

1	2	1	2	2	2	2
1	2	1	2	2	3	3
1	2	1	2	2	2	2
1	2	1	2	3	3	2
1	2	1	2	2	1	2
1	2	2	3	2	2	3
1	2	2	2	2	2	2
0	3	3	3	3	3	3

Exercise 7.1.

The probability of picking 'lama' from the British National Corpus is 132/100000000 = 0.00000132. The probability of picking "llama" is 0.00000033.

Exercise 7.2.

P('lama' | 'Tibetan')
= *C*('Tibetan lama')/*C*('Tibetan')
= 1/217
≃ 0.007874

The independent probability *P*('lama') is 0.00000132 (see exercise 7.1). The probability of 'lama' is greatly increased when following the word 'Tibetan'.

Exercise 7.3.

P('lama' | 'I saw a')
= C('I saw a lama')/C('I saw a')
= 0/408
= 0

The problem is that C('I saw a lama') = 0, so the probability of 'lama' following 'I saw a' is 0, which means it should never ever occur. Yet it can occur.

Exercise 7.4.

P('I') = 869460/10000000 = 0.0086946

P('saw' | 'I') = C('I saw')/C('I') = 4391/869460 ≃ 0.00505

P('a' | 'saw') = C('saw a')/C('saw') = 1841/26737 ≃ 0.06885589

P('Tibetan' | 'a') = C('a Tibetan')/C('a') = 25/2150885 ≃ 0.000011623

P('lama' | 'Tibetan') ≃ 0.007874

P('I saw a Tibetan lama')
= 0.0086946 × 0.00505 × 0.06885589 × 0.000011623 × 0.007874
≃ 0.00000000000027669

which, though a very small number, is greater than zero (which is an improvement on the problem identified in Exercise 7.3).

Exercise 7.5. A first-order model.

Exercise 7.6.

P('such' | 'to create') = C('to create such')/C('to create') = 16/4632 ≃ 0.00345423

Exercise 9.1.

(1') S –> NP kicked Det Adj bucket

To capture the fact that the Adj is optional, we could either a) keep both the original rule 1 and the new rule 1' in the grammar, or b) add a rule for empty adjectives, i.e. Adj –> ''.