COMP 261 Test 2

27 May 2021

Instructions

- Time allowed: **45 minutes** .
- Answer **all** the questions. There are 45 marks in total.
- Write your answers in the booklets and remember to include your Student ID.
- If you are a remote student, then write the answers on paper and submit a photo to the relevant submission system entry.
- If you think some question is unclear, ask for clarification.
- This test contributes 25% of your final grade
- You may use paper translation dictionaries, and non-programmable calculators.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	**Marks**
1. Parsing	[20]
2. String Search	[15]
3. Compression	[10]

1. Parsing **(20 marks)**

(a) Given the following grammar, which of the following sentences belong to the language and which ones do not?

```
FOO ::= BAR abc | BAZ abc | def
BAR ::= let  BAZ | gee BAZ | hey END
BAZ ::= END | hey END
END ::= too
```

(i) **(1 mark)**

```
let  hey too
```

(ii) **(1 mark)**

```
too  hey too
```

(iii) **(1 mark)**

```
hey hey too
```

(iv) **(1 mark)**

```
gee hey
```

(v) **(1 mark)**

```
hey too abc
```

(b) **(5 marks)** Given the following grammar (different from above!), draw a concrete parse tree for the sentence below:

```
FOO ::= BAR abc | BAZ abc | def END
BAR ::= let  BAZ | gee BAZ | hey END
BAZ ::= END | woo
END ::= too BAR
```

```
def  too  hey too  let  too  gee woo
```

(c) For the following grammars, state which ones are ambiguous and why and which ones will work with a simple top down recursive descent parser. If it is ambiguous explain which rule fails and why (e.g. left recursion or intersecting first sets etc) — show an example of a string that will produce multiple concrete parse trees.

(d) **(5 marks)**

```
FOO ::= BAR abc | BAZ abc
BAR ::= def |  BAZ
BAZ ::= fed
```

(Question 1 continued on next page)

(Question 1 continued)

(e) **(5 marks)**

```
FOO ::= BAR abc | BAZ abc | def END
BAR ::= let  BAZ | gee BAZ | hey END
BAZ ::= END | hey
END ::= too BAR
```

2. String Search **(15 marks)**

(a) **(5 marks)** Show the partial match table generated for the KMP algorithm for the following search string.

```
xyxyyyzxyxx
```

(b) **(10 marks)** Show the steps the KMP algorithm using the partial match table and search string above will take when searching through the following text until the match is found.

```
abcxysyxxxyyyyxyxyyyxyxyyyzxyxxxxz
```

3. Compression **(10 marks)**

(a) **(5 marks)** Draw a Huffman Tree generated for the following symbols given their frequencies. During the generation of the tree and codes, if the two nodes have the same frequency, then prioritise the ones that have the letter that comes *earlier* in the alphabet first. For the final tree codes, put 0 on the *left* branches and put 1 on the *right* branches. This way we can expect identical trees and codes for the whole class and will not give marks for the trees that did not follow these assumptions.

a	b	c	d	e	f	g	h	i	j	k	space
30%	20%	10%	10%	5%	5%	5%	5%	4%	3%	2%	1%

(b) **(5 marks)** Now encode the following string using your Huffman Tree generated above:

```
he fed a cafe  hi  cab
```

* * * * * * * * * * * * * * *