

Exercise 5 - Solutions

1. Iterate through the character string and apply a bitwise OR with bit mask $0x20$ to each character.

In Python, for example:

```
S = "MyString"
out = ""
for c in S:
    out += chr(ord(c) | 0x20)

print(out)
```

2. 8-bit characters effectively use 7 bit freely, i.e. there are $2^7 = 128$ character codes.
16-bit " " " " " " " " " " " ", i.e. " " " " $2^{14} = 16384$ " " "

⇒ The total is $16384 + 128 = 16512$ character codes.

3.
 - ASCII (quite common legacy encoding) is a valid subset of UTF-8.
 - Substantially more code points than UTF-16. (But currently, there is still a lot of room even in UTF-16.)
 - More efficient on average for text with common markup (e.g. HTML) which is predominantly ASCII. Thus, Western languages are coded much more efficiently and even East Asian languages with lots of surrogate characters benefit in practical contexts such as full web pages.
 - The raw byte stream does not contain $0x00$ -bytes, so can be handled safely by (legacy) C-style string processing routines.
4.
 - Iterate through the bytes of the byte string
 - Check the first bit of every byte
 - * if set (on at least one byte), it's not ASCII
 - * if not set on any byte, it's valid ASCII

5. (a) All strings over the alphabet $\Sigma = \{a, b\}$ that

- start with 0 or more letters "a",
- continue with 1 or more letters "b",
- followed by exactly 1 "a".

Remark: Regex is a^*b+a (classical syntax)
 $^a^*b+a\$$ (python regex)

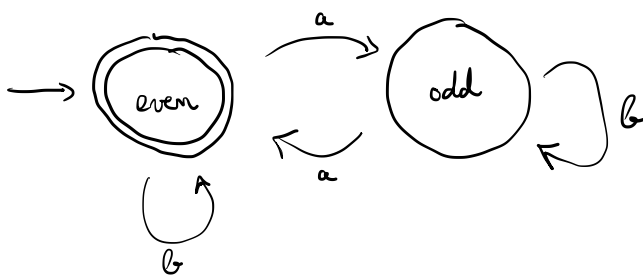
(b) Any binary string that ends in "101".

Remark: Regex is $[01]^*101$ (classical syntax)
 $^[01]^*101\$$ (python regex)

Why?

- Reading "101" from any state puts the FSA into the accepting state S_3
- There is no other way to reach the accepting state S_3
- The FSA cannot "get stuck"

6.

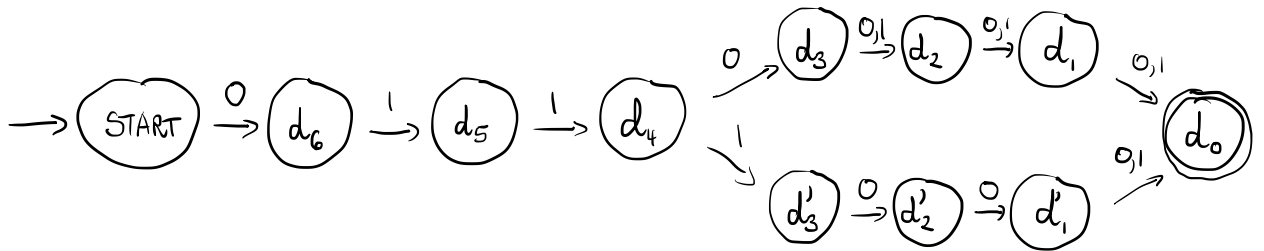


7. Number digits in ASCII are $0x30$ to $0x39$, in 7-bit binary

$011\ 0000$ to $011\ 1001$, so:

- start with 011
- if next bit is 0, it's a number digit for sure
- if next bit is 1, it's a number digit only iff followed by two 0.

FSA:



(We take the convention that if the machine get "stuck", we reject the input string.)