Finite State Machines

Example: How to count words in a stream or string of text?

- ~~# spaces - 1?~~ (might have consecutive spaces) (could also have leading/trailing spaces)

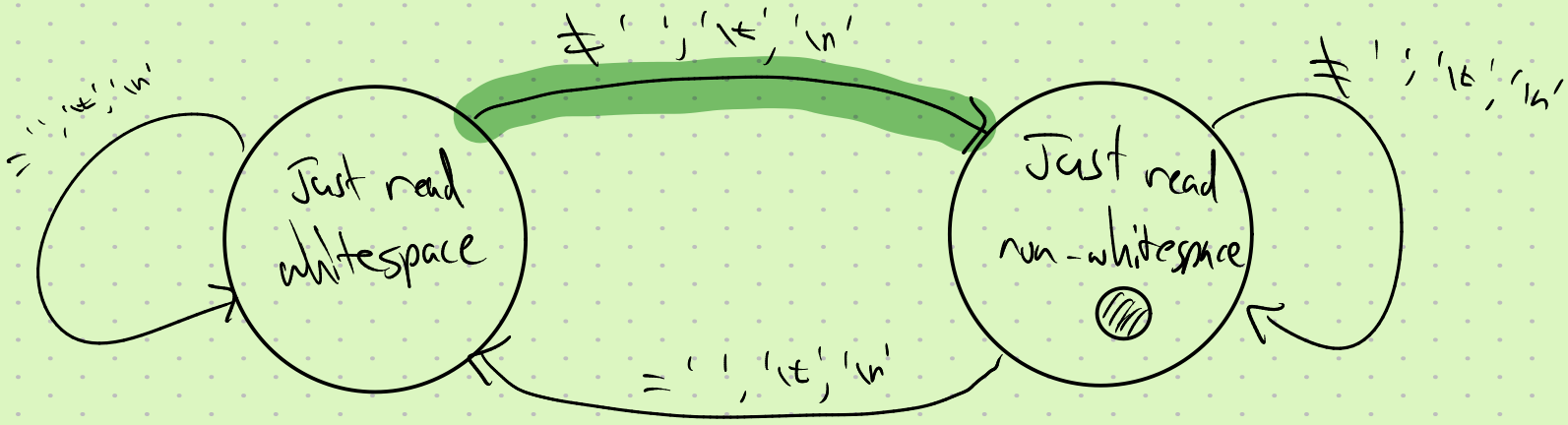
Better: look for word boundaries: transitions from "whitespace" char → non-whitespace char

whitespace: ' ', '\t', '\n'

Solution via state machines:

"The␣ grass␣␣␣is␣␣always␣greener..."

space



How to program diagram?

Need to know where the rock is. Only 2 possibilities,

So we could use a boolean:

```
bool justreadws = true;
```

Also need # of words:

```
size_t wc = 0;
```

Now the diagram simulation.

Need to handle {states} x {arrows from that state}

(all (state, arrow) pairs)

```
char c;
while(scanf("%c", &c) == 1) { // ≈ cin >> c

    if (justreadws) {  // rock is left

        if ( c != ' ' && c != '\t' && c != '\n') { // green arrow

            wc++;
            justreadws = false;

        }
```

```
        else {          //self-arrow on left
        }
    } else { // rock is on right
        if (c==' ' || c=='\t' || c=='\n') {
            // bottom arrow; move rock left:
            justrocks = true;
        } else {...} // self arrow on right...

    }

}
cout << wc << "\n";
```

---

other State Machine Applications:

- parsing file formats
- used in writing assemblers & compilers...
- network protocols (TCP, SMTP, TLS...)

Example: detect/remove comments from C code.

```
// .... \n
/* .... */
```

String S = "// not a comment";

String S2 = "Not the \" end of the string";

Approximation of diagram: