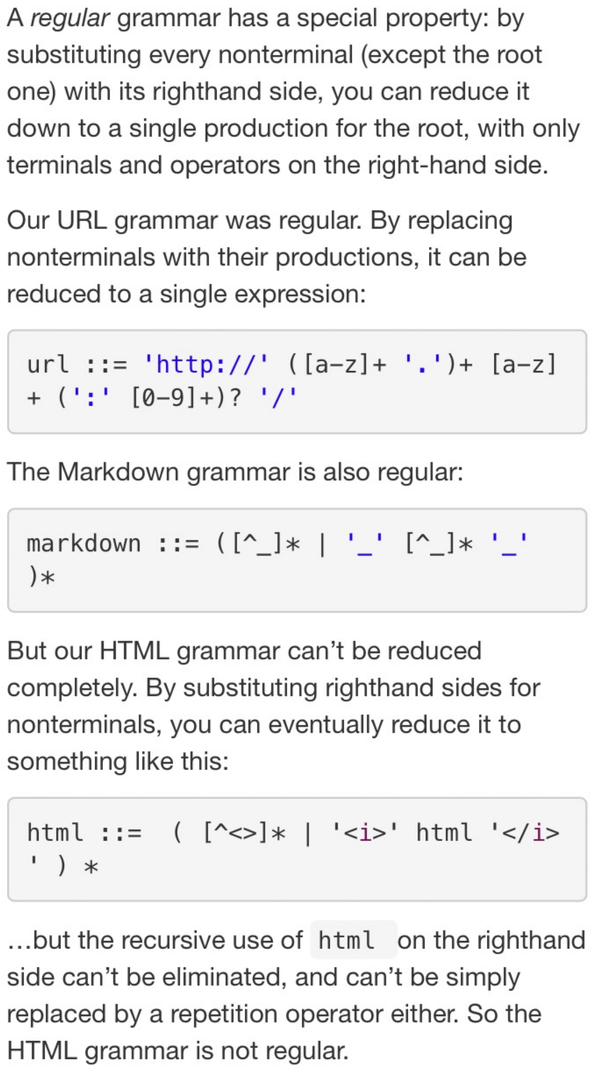
Ch 17

Grammar

- used to describe a seq of symbols
  - ↳ operations
    - ↳ concat
    - ↳ repetition
    - ↳ union
    - ↳ 1+ repetition
    - ↳ character classes

Regular Grammar



A *regular* grammar has a special property: by substituting every nonterminal (except the root one) with its righthand side, you can reduce it down to a single production for the root, with only terminals and operators on the right-hand side.

Our URL grammar was regular. By replacing nonterminals with their productions, it can be reduced to a single expression:

```
url ::= 'http://' ([a-z]+ '.')+ [a-z]
+ (':' [0-9]+)? '/'
```

The Markdown grammar is also regular:

```
markdown ::= ([^_]* | '_' [^_]* '_'
)*
```

But our HTML grammar can't be reduced completely. By substituting righthand sides for nonterminals, you can eventually reduce it to something like this:

```
html ::=  ( [^<>]* | '<i>' html '</i>
' ) *
```

…but the recursive use of `html` on the righthand side can't be eliminated, and can't be simply replaced by a repetition operator either. So the HTML grammar is not regular.

Context-Free Grammars

- a language that can be expressed using the prev. defined system of grammar

- context-free $\not\Leftrightarrow$ regular

- nested languages are context free but not regular