Models of Computation, Languages, and Recursion
Part 2

Kevin Schewior

Website: https://sites.google.com/view/kschewior/home

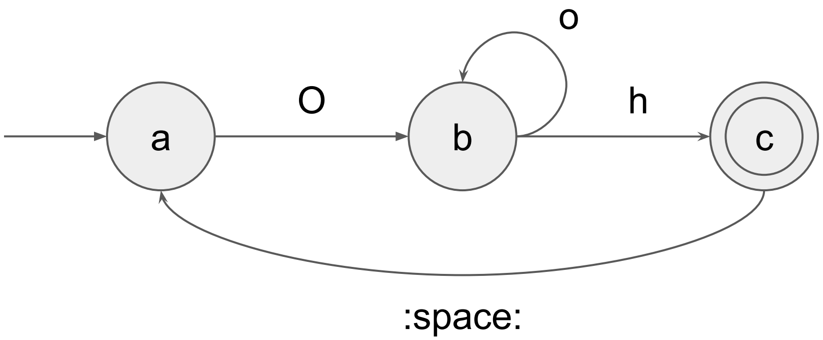
Slides largely by Fabrizio Montesi

SDU

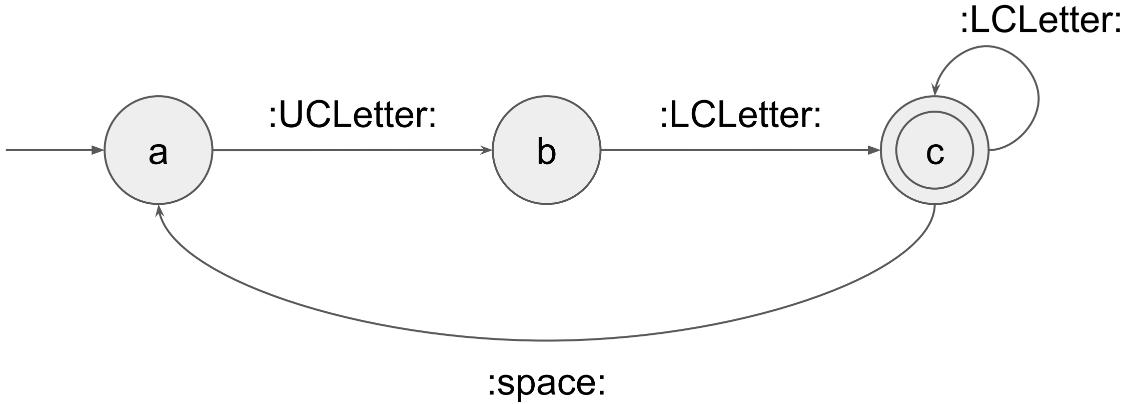
DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE

Recap: Deterministic Finite Automaton (DFA)
An informal introduction, with an example

Some states are accept states (denoted with a double circle).
Example (c is an accept state):



Recap: A DFA for Full Names, The Recursive Attempt



Recap: Context-Free Grammar (CFG)

- The capital letters that appear on the left are called non-terminal characters (non-terminals for short).
- S is the start non-terminal.
- All other characters are called terminals, which make up for the actual content of strings that the CFG can recognise.

$$
\begin{aligned}
S &\rightarrow \text{Hello } T \\
S &\rightarrow \text{Hey } T \\
T &\rightarrow \text{there}
\end{aligned}
$$

Recap: The language of a CFG

The language of a CFG is the set of all strings that can be derived by that CFG (starting from S).

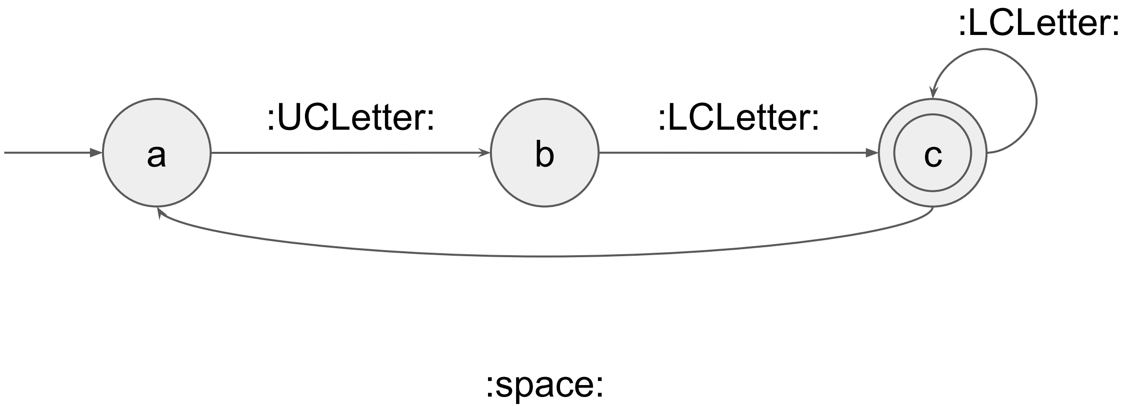
The language of this CFG

S → Hello T
S → Hey T
T → there

is {Hello there, Hey there}.

Recap: Another CFG

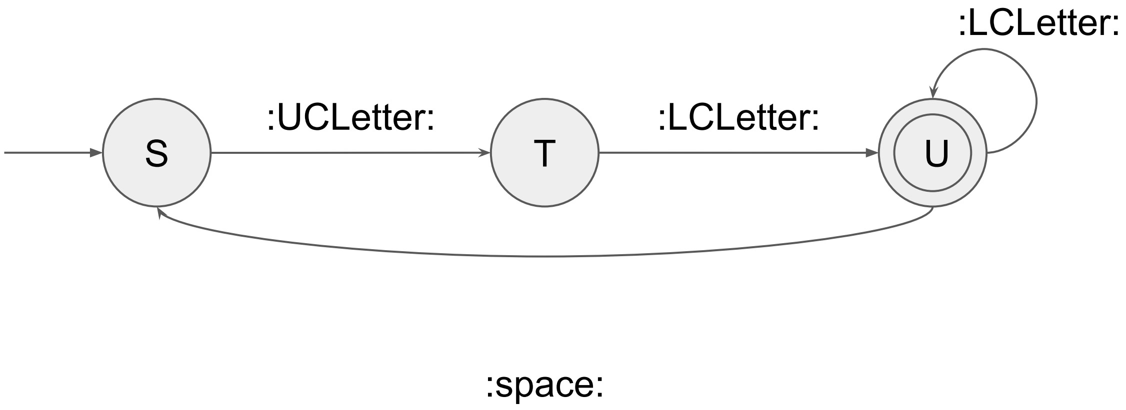
S → :UCLetter:L S
S → :UCLetter:L
L → :LCLetter:L
L → :LCLetter:



It’s exactly the same kind of recursion.

Recap: Another CFG

S → :UCLetter:T
T → :LCLetter:U
U → :LCLetter:U
U → :empty:
U → :space:S



Recap: The science of DFAs

- **Q:** What kind of strings can I recognise with a DFA?
- How do we answer?..
- We need to understand the **limitations** of DFAs.
- So another interesting question is:
  - **Q:** What are things that **cannot possibly** be recognised with a DFA?
- Which is way more fun. Breaking stuff is the best part of being a scientist.
- May ask the same questions for CFGs!

Recap: The language of balanced parentheses

- Some correct strings: (), (()), ((())), ()()(), ()(()), (())((()()())).
- Some incorrect strings: (,), ((), ()), (()(), (())),))((
- Intuitively, a string is in the language if each left parenthesis has a matching right parenthesis and the matched pairs are well nested.
- OK, let's try to come up with a DFA that recognises this.

Recap: The language of balanced parentheses

- There is no DFA for balanced parentheses.
- Why?
- We need to remember how many open parentheses we have, and this number has no bound. (We cannot predict how many there can be.)
- Since a DFA has a finite number of states, there are always cases where we do not have enough memory.

Balanced parentheses

$$S \rightarrow (S)$$
$$S \rightarrow SS$$
$$S \rightarrow ()$$

Balanced parentheses

$$S \to (S)$$
$$S \to SS$$
$$S \to ()$$

Some derivations:

- $S \to ()$
- $S \to (S) \to (())$
- $S \to SS \to ()S \to ()(S) \to ()(SS) \to ()(()S) \to ()(()())$

Balanced parentheses

$$S \to (S)$$
$$S \to SS$$
$$S \to ()$$

- Some correct strings: (), (()), ((())), ()()(), ()(()), (())((()()())).
- Some incorrect strings: (,), ((), ()), (()(), (())),))((

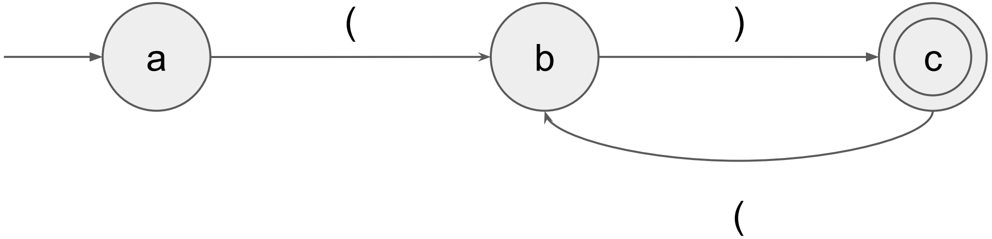
Balanced parentheses

$$
\begin{aligned}
S &\rightarrow (S) \\
S &\rightarrow SS \\
S &\rightarrow ()
\end{aligned}
$$

- Why can we do balanced parentheses with a CFG and not with a DFA?
- Because the kind of recursion that we have in CFGs is more powerful: it has a memory!
- Specifically, when you “expand” a non-terminal, we remember what to do after we are done expanding.

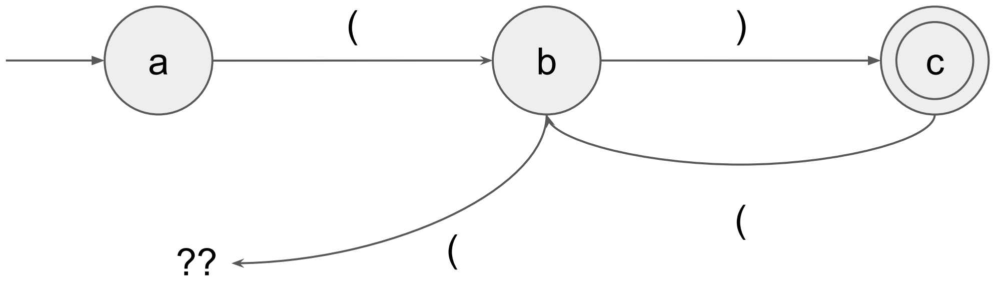
Balanced parentheses

$$S \rightarrow (S)$$
$$S \rightarrow SS$$
$$S \rightarrow ()$$



Balanced parentheses

$$S \rightarrow (S)$$
$$S \rightarrow SS$$
$$S \rightarrow ()$$

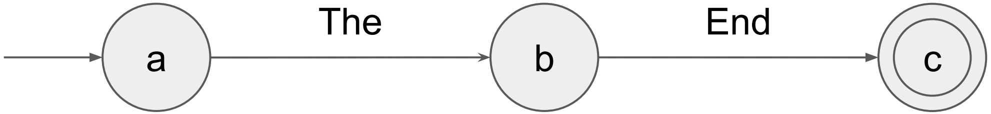


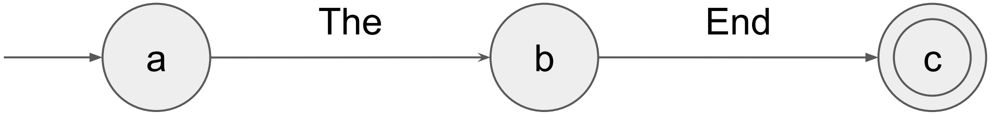
A few languages

- The set of the strings ab, abab, ababab, abababab, …
- The set of the strings ab, aabb, aaabbb, aaaabbbb, …
- The set of strings consisting of (,), [, and] such that
  - (and) are balanced taken for themselves
  - [and] are balanced taken for themselves

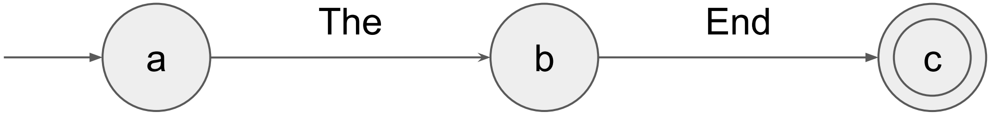
  Examples: ()[], (([)]); Non-Example: (([]

Does there exist a DFA recognizing them? A CFG?

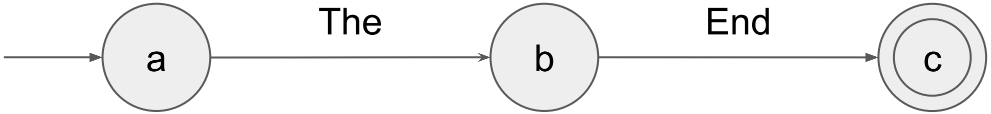

The
End
a
b
c



Wait a moment...



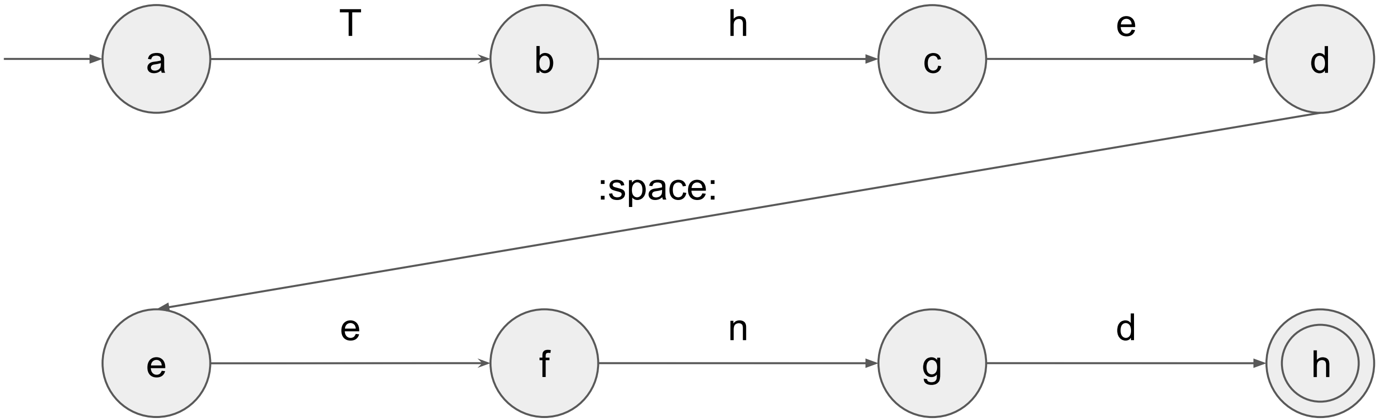
Wait a moment... is that a DFA?



Wait a moment... is that a DFA?

No, because transitions are labelled with single characters, not strings! (See many slides back.)

Don’t take these things too lightly, they can be tricky.



Questions?