Announcements

- **Friday 5pm is the change of grading deadline**
  - To pass ACE
    - Get at least a C- in CS 103
    - Come to all the sections
      - You can be absent / late no questions asked for one section
      - Exceptions for extenuating circumstances (illness, emergencies, etc.), email me beforehand if possible for these absences
- If you withdraw or take an incomplete in 103, you must do the same in ACE
- If you are thinking of taking an incomplete, you MUST talk to Cynthia & Alex

Regular Expression Summary

- Describe a language by describing the patterns of strings that are in that language

- $\varnothing$ is a regular expression that represents the empty language $\varnothing$.
- ε is a regular expression that represents the language $\{\varepsilon\}$.
- For any $\texttt{a} \in \Sigma$, the symbol `a` is a regular expression for the language $\{\texttt{a}\}$.
- R^* is a regular expression for the Kleene closure of the language of R.
- For regular expressions R_1 and R_2, you can combine these regular expressions:
  - $R_1 R_2$ is a regular expression for the concatenation of the languages of R_1 and R_2.
  - $R_1 \cup R_2$ is a regular expression for the union of the languages of R_1 and R_2.
- $\texttt{R}^n$ is shorthand for n copies of `R` put together. For example, $\texttt{R}^5$ is `RRRRR`. $\texttt{R}^0$ is defined as ε.
- Σ is shorthand for "any character in Σ".
- `R?` is shorthand for $\texttt{R} \cup \varepsilon$, meaning "zero or one copies of `R`".
- `R+` is shorthand for $\texttt{RR}^*$, meaning "one or more copies of `R`".

Nonregular languages

- **Regular languages** can be represented with a DFA, NFA, or regex
- **Nonregular languages** cannot: they need “infinite memory”

Notes on “infinite”:

- All strings have finite length
- All finite languages are regular
- Some infinite languages are regular, and some infinite languages are not regular
    - What’s an example of an infinite regular language?

Distinguishability

Two strings x and y are **distinguishable** relative to a language L when

$$\exists w \in \Sigma^*.\ (xw \in L \leftrightarrow yw \notin L).$$

We write this as $x \not\equiv_L y$

> Important: we usually use the slash to mean “not”, but the slash goes through the ≡ when the strings **are** distinguishable

Intuition: they represent different states within a DFA

What would you have to show for two strings to be distinguishable?

What would it mean for two strings to **not** be distinguishable?

Distinguishing Sets

A set of strings is a **distinguishing set** for a language L when

$$\forall x \in S.\ \forall y \in S.\ (x \neq y \rightarrow x \not\equiv_L y)$$

Intuition: all possible pairs of strings from the set are distinguishable.

What would you have to show for a set to be a distinguishing set?

What would it mean for a set to **not** be a distinguishing set?

I have a mystery language L. What's the smallest distinguishing set?

Myhill-Nerode

Today's big takeaway:

To prove a language is not regular, use Myhill-Nerode.

If you can find an infinite distinguishing set for a language, that language is not regular!

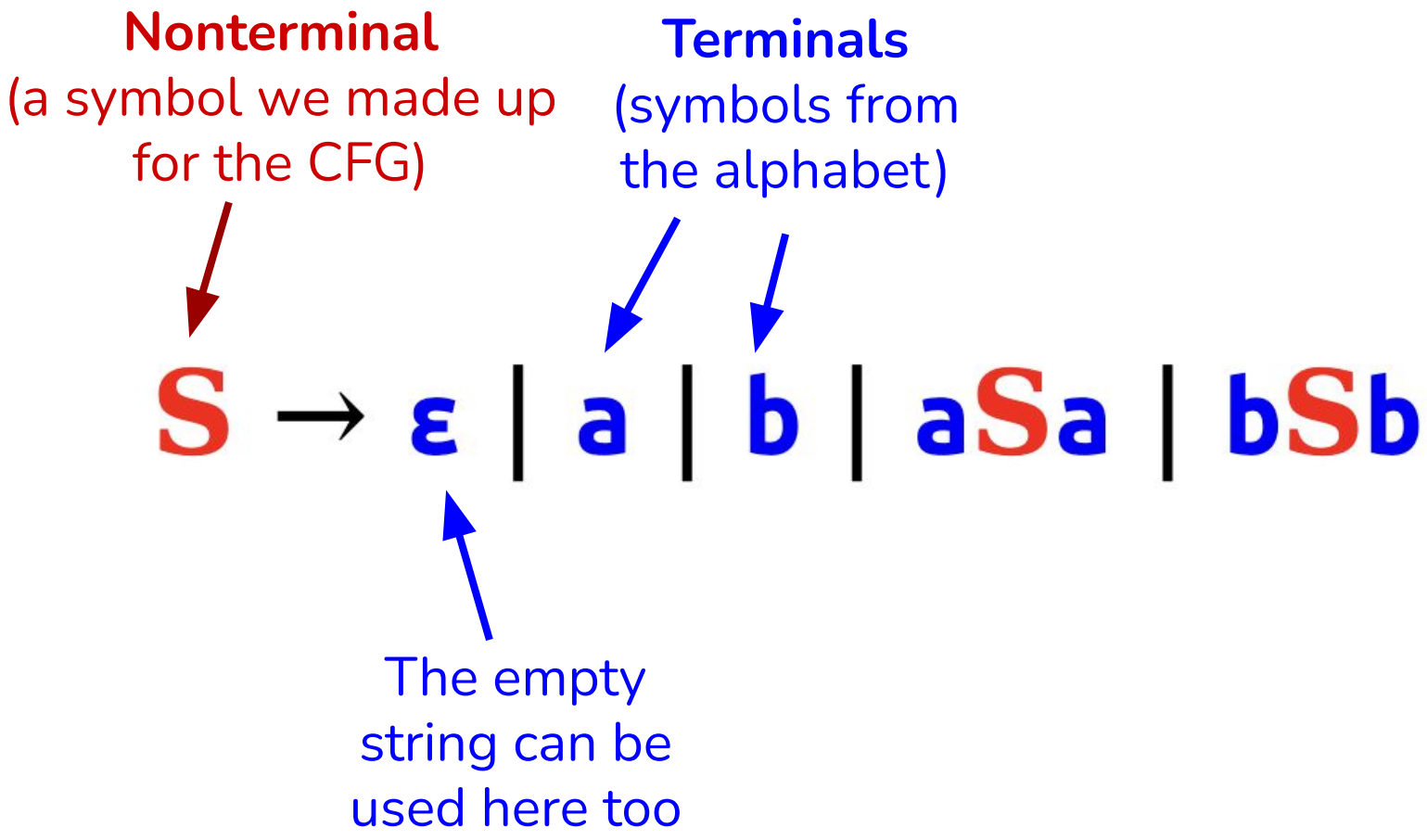
Like any existential proof, finding the set can be difficult.

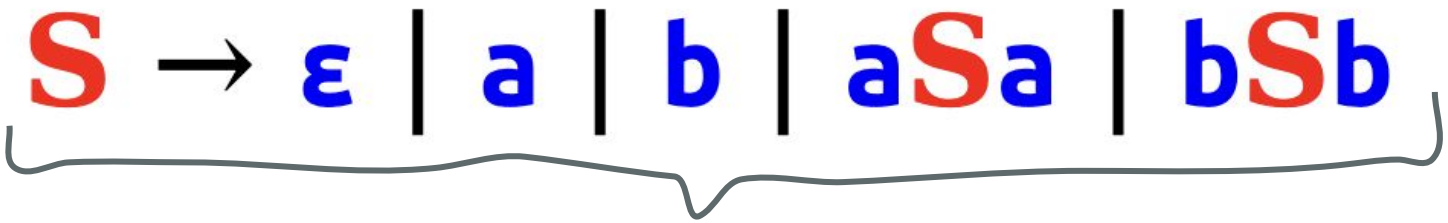
- Tip: try simple sets with a nice structure

Context-Free Grammar Summary

- Describe a language by describing recursive structure
- Intuitively, “more computing power” than DFAs/regexes

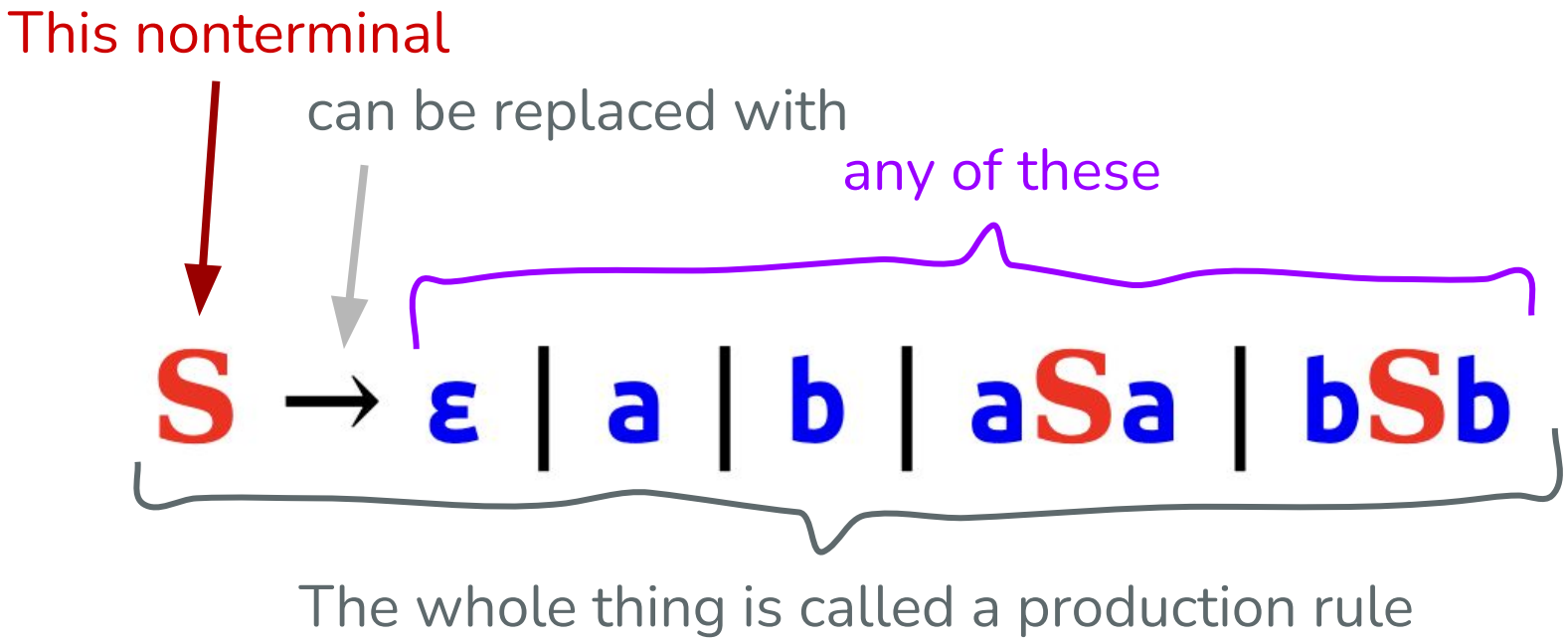
$$S \rightarrow \varepsilon \mid a \mid b \mid aSa \mid bSb$$


Nonterminal
(a symbol we made up for the CFG)
Terminals
(symbols from the alphabet)
S → ε | a | b | aSa | bSb
The empty string can be used here too

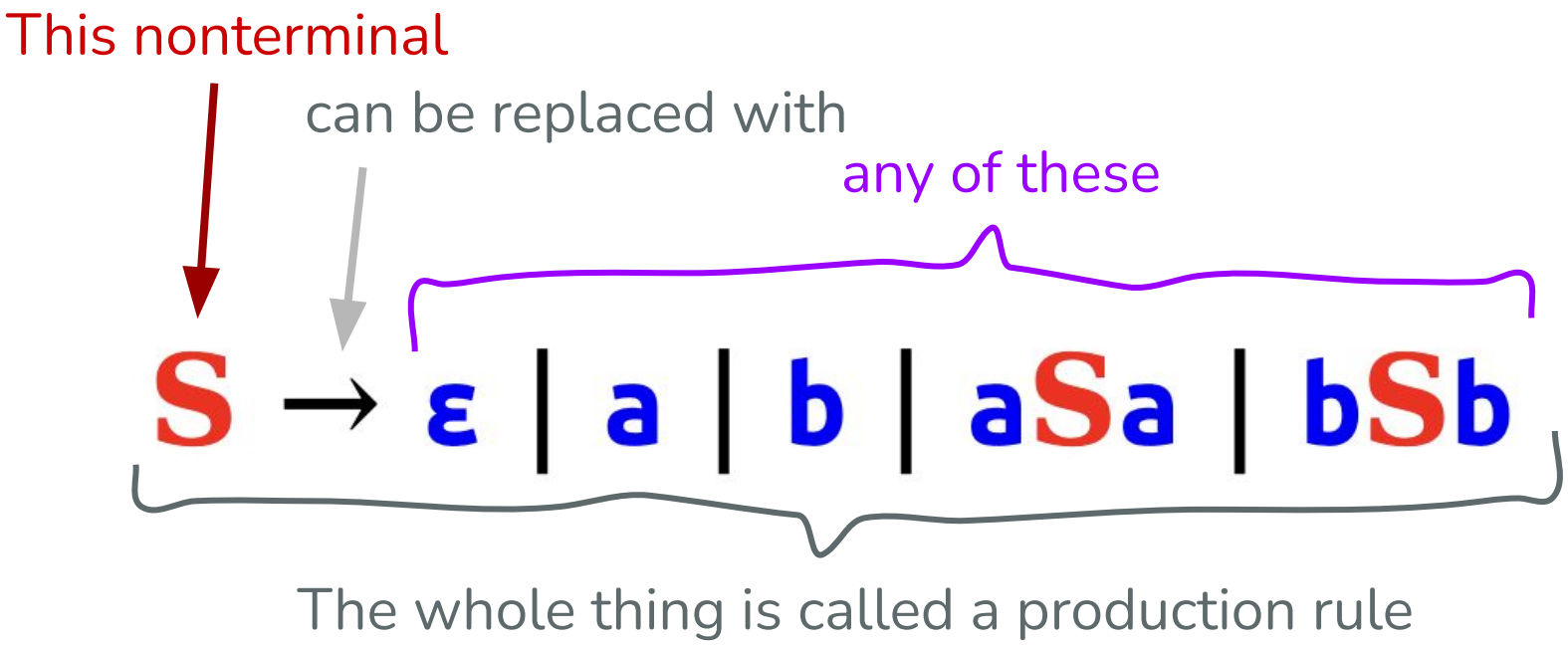


The whole thing is called a production rule

(the term is not super important)



The whole thing is called a production rule



- A CFG is a list of production rules
- A CFG's language = set of all **strings of terminals** derivable **from the start symbol** (the nonterminal from the **first** rule)