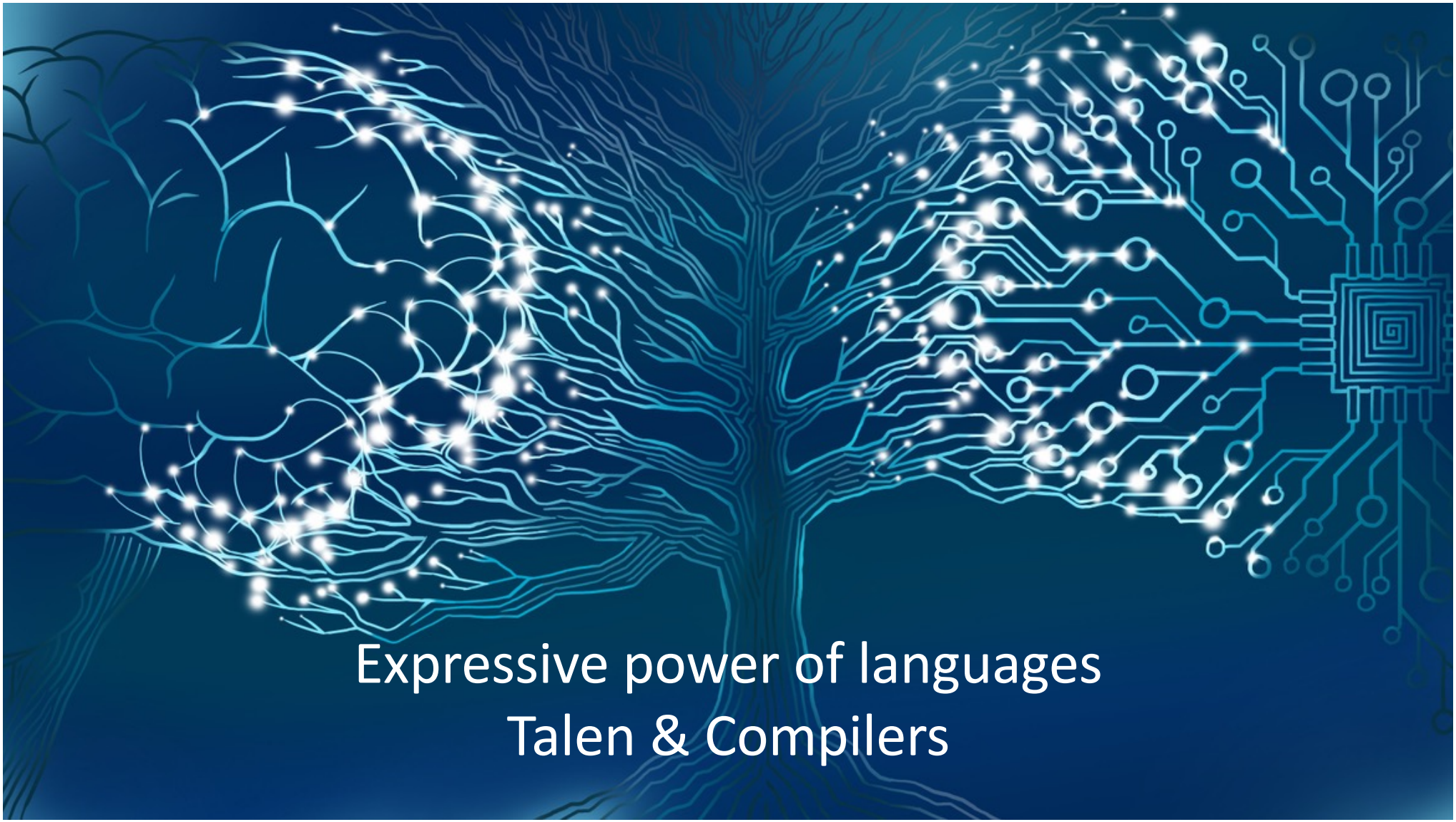

Expressive power of languages
Talen & Compilers

# Expressive power of languages

Lecture notes: 8

Utrecht University

# Learning goals

- prove that a language is not regular
- prove that a language is not context-free
- identify languages and grammars as regular, context-free or none of these
- give examples of languages that are not regular, and/or not context-free
- explain the Chomsky hierarchy

Utrecht University

# Quality and efficiency aspects of language[1]

Sapir - Whorf: the words we use influence the way we perceive the world

The sounds we make depend on our climate (warm: long distance communication: vowels; cold: short distance communication: consonants)

Many adult second language learners -> less complex language

Difficult to find out through experiments

[1]Steven Mithen (2024). The language puzzle - Piecing Together the Six-Million-Year Story of How Words Evolved. Basic Books

Utrecht University

# Some common questions…

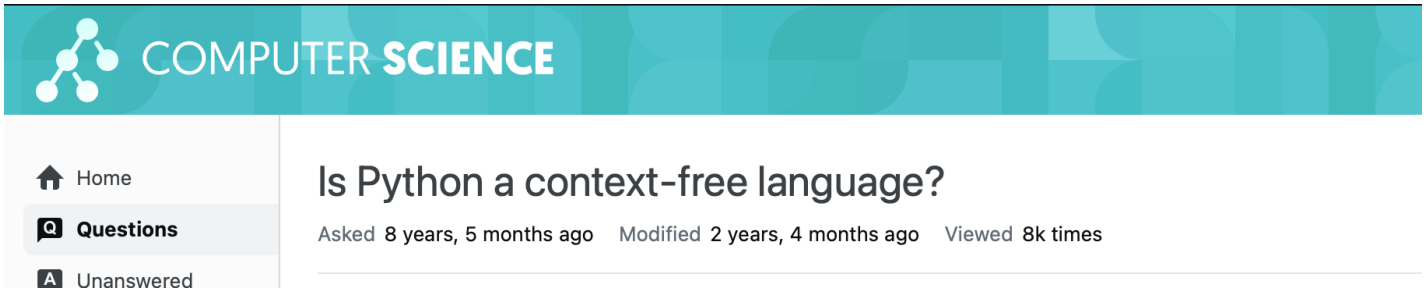
r/AskProgramming • 3y ago
[deleted]

## ELI5- Why can't regex parse HTML?

I saw in a thread recently about how you can't use RegEx searches to parse HTML due to Chomsky grammar hierarchy differences… while I'm willing to accept this in an abstract sense I'm having a hard time understanding what this means practically. What about HTML makes it… of a higher order, and what about regex makes it insufficient to parse HTML?

Sorry if this is an inelegant way of phrasing these questions. I'd just really like to know more about the topic because it was fascinating to me.

36 · 46



Utrecht University

# Grammar types

Chomsky's hierarchy (1956)

Type 3: regular grammars

Type 2: context-free grammars

Strictly more powerful than regular grammars

Type 1: context-sensitive grammars:

Rewrite rules of the form $\phi A \psi \rightarrow \phi \delta \psi$

Strictly more powerful than context-free grammars

Type 0:

Rewrite rules of the form $\phi \rightarrow \psi$

Strictly more powerful than context-sensitive grammars

Utrecht University

# How do you prove a language is not regular?

To show that a language is regular: give a regular grammar, FSA, regexp, …

To show that a language is **not** regular: show for all regular grammars that they don't describe the language.

Utrecht University

# How do you prove a language is not regular?

Expose a limitation in the formalism (in this case, in the concept of finite state automata)

From this limitation, derive a property that all languages in the class (in this case, regular languages) satisfy

If a language does not have that property, it cannot be in the class

Utrecht University

# Loops in DFAs

Suppose we have a DFA, and we use it to accept a string

How many states do we visit…

- If the string has length 0?

  One (the start state)

- If the string has length 1?

  Two or one. If one we visit a state twice and go through a loop.

- If the string has length 2?

  Three or less. If less than 3 we go through a loop.

Utrecht University

# Finite state automata are finite

Any DFA has a finite number of states

Suppose we have a DFA with n states

How many states do we visit if we read a string that is accepted and has length n?

n+1 or less. If less, we go through a loop

But there are only n states! So we **have to** go through a loop.

Utrecht University

# A property satisfied by all regular languages

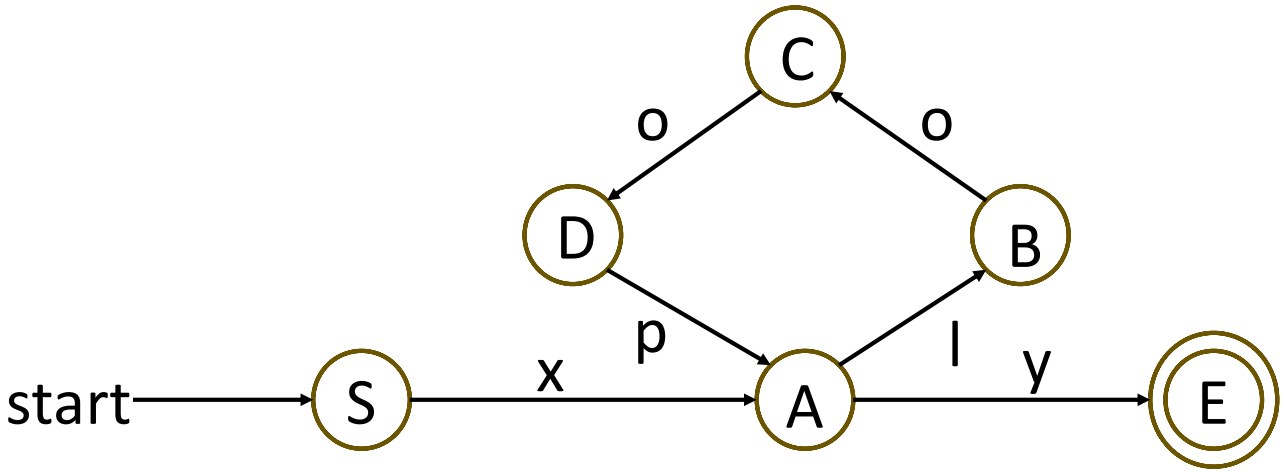
What can we do if a DFA has a loop?

We can go through it as often as we want

If we have a word that is accepted and goes through the loop once, then the words that follow the same path and go through the loop any number of times are also accepted

Utrecht University

# An example




Accepts: xy

xloopy

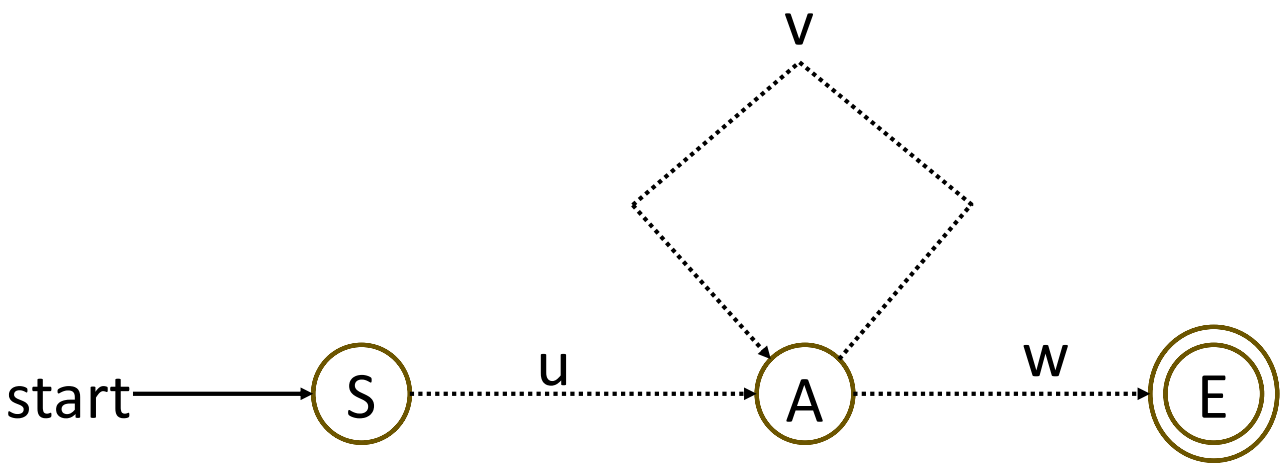
xlooploopy

xlooplooploopy

…

Utrecht University

# The general case




This is a simplified situation: there may be more nodes and edges

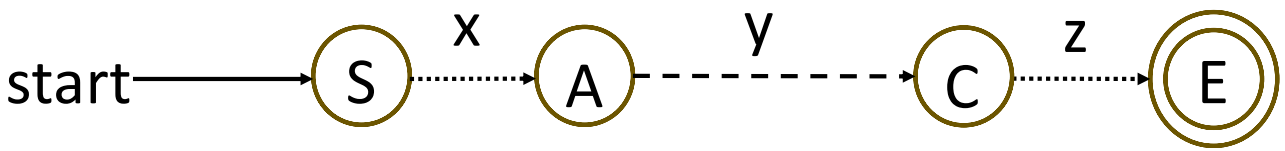
u and w may be empty, v not

All words of the form $uv^iw$ for $i \in N$ are accepted

Utrecht University

# Generalised

A loop occurs in any subword of at least length n

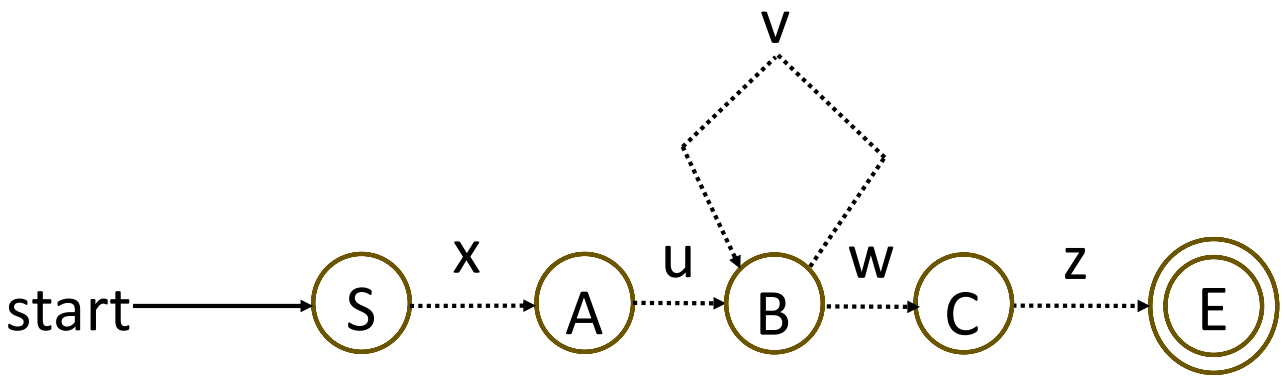



Suppose xyz is accepted, and length y is at least n

Utrecht University

# Generalised

A loop occurs in any subword of at least length n




Suppose xyz is accepted, and length y is at least n

Then y is of the form uvw, with v not empty, accepted by a loop

All words of the form $xuv^iwz$ for $i \in N$ are accepted

Utrecht University

# A property satisfied by all regular languages

**Pumping lemma for regular languages**
(Rabin and Scott, 1959; gave them a Turing award (well…))

For every regular language L there exists an $n \in N$

such that for every word $w = xyz$ in L with $|y| \geq n$

we can split y into three parts, $y = uvw$, with $|v| > 0$

such that for every $i \in N$, we have $xuv^iwz \in L$

Extremely informally: a DFA cannot recognize a language when it needs to maintain a counter that can become arbitrarily large

Utrecht University

# How do you prove a language is not regular?

Expose a limitation in the formalism (in this case, in the concept of finite state automata)

From this limitation, derive a property that all languages in the class (in this case, regular languages) satisfy

If a language does not have that property, it cannot be in the class

Utrecht University

# How do you prove a language is not regular?

**Expose a limitation in the formalism (in this case, in the concept of finite state automata)**

**From this limitation, derive a property that all languages in the class (in this case, regular languages) satisfy**

If a language does not have that property, it cannot be in the class

Utrecht University

# Using the pumping lemma

To show that a language is not regular, we show that it does not have the pumping lemma property as follows:

Assume that the language is regular

Use the pumping lemma to derive words that must be in the language, but are not:

find a word xyz in L with $|y| \geq n$,
from the pumping lemma there must be a loop in y,
but repeating this loop takes us outside of the language

The contradiction means that the language cannot be regular

Utrecht University

# Using the pumping lemma - strategy

For **every** natural number n

find a word xyz in L with $|y| \geq n$ (you **choose** the word)

such that for **every** splitting y= uvw with $|v| > 0$,

there exists a number i (you **choose** the number)

such that $xuv^iwz \notin L$ (you have to **show** it)

Utrecht University

# Example

The language $L = \{a^m b^m \mid m \in N\}$ is not regular

Proof:

Assume L is regular

Then there exists a DFA accepting L
Assume this DFA has n states

Utrecht University

# Example

$a^n b^n$ is a word in L

Let $x = \varepsilon$, $y = a^n$, $z = b^n$, then $xyz = a^n b^n \in L$ and $|y| \geq n$

From the pumping lemma, we know there must be a loop in y

Let $y = uvw$ where $|v| > 0$ and $xuv^i wz \in L$ for all $i \in N$

Let $u = a^p$, $v = a^q$, $w = a^r$ where $p + q + r = n$

If $i = 2$, then $xuv^2wz = a^p a^{2q} a^r b^n = a^{n+q} b^n$

But $n + q \neq n$ because $q > 0$! So $xuv^2wz \notin L$

Utrecht University

# Example

The language $L = \{a^m b^m \mid m \in N\}$ is not regular

Proof:

Assume L is regular

[Previous slide: L does not satisfy the property of regular languages]

The contradiction means our assumption is wrong: L is not regular

Utrecht University

1 Go to **wooclap.com**

2 Enter the event code in the top banner

Event code
**PEGNKG**

Enable answers by SMS


# Q1

Utrecht University

# How about context-free languages?

If you want to prove that a certain language is not context-free, apply the same strategy as for regular languages:

Expose a limitation in the formalism (in this case, in the concept of **context-free grammars**)

From this limitation, derive a property that all languages in the class (in this case, **context-free languages**) satisfy

If a language does not have that property, it cannot be in the class

Utrecht University

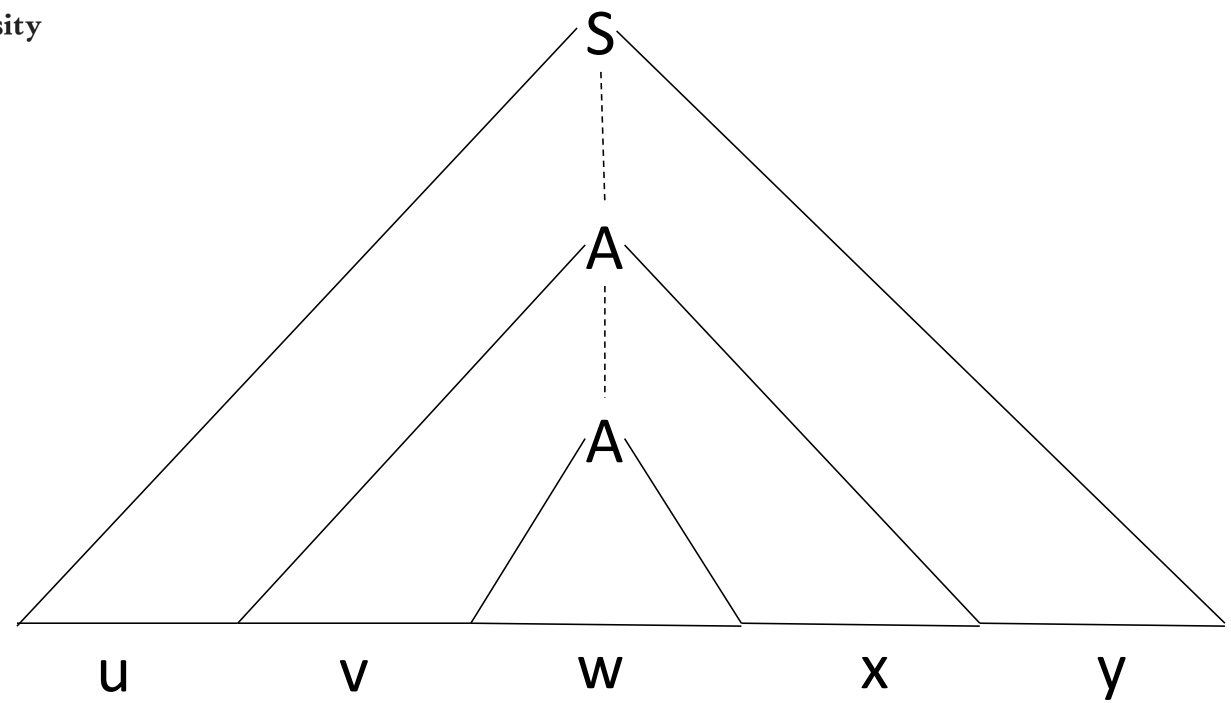
# Grammars and parse trees

Every sentence in a context-free language has a parse tree

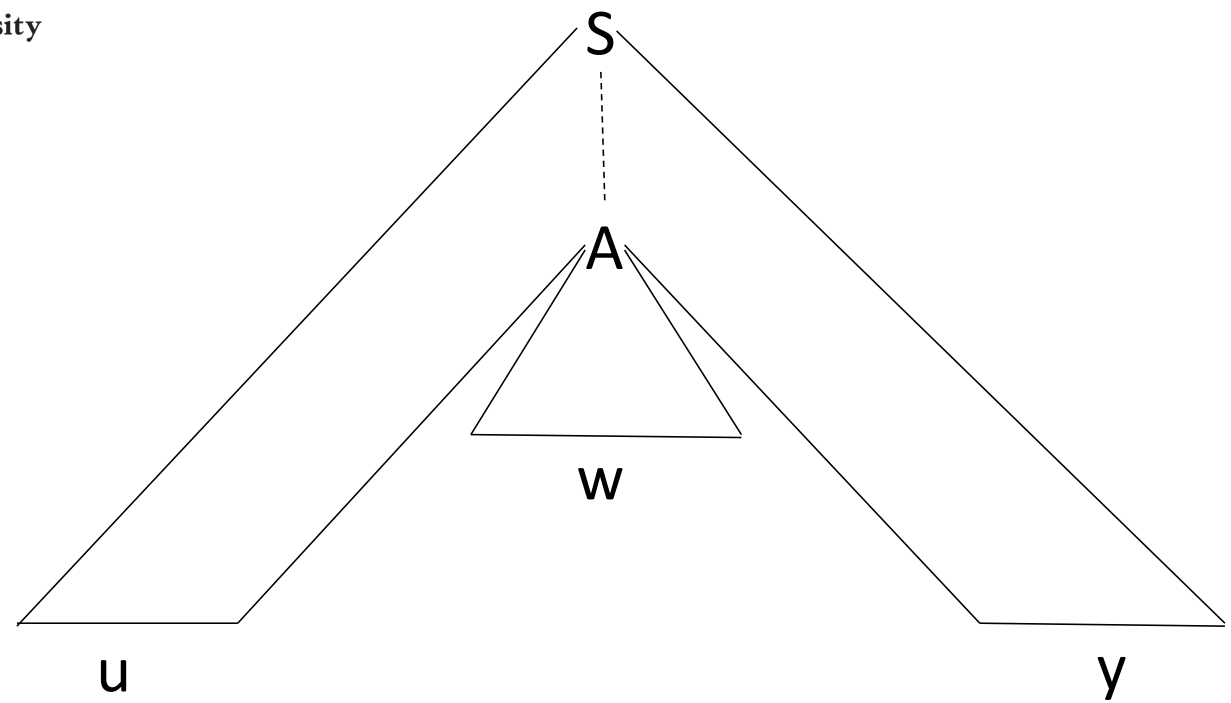
We can produce parse trees of arbitrary depth if we find sentences in the language that are long enough, because the number of children per node is bounded by the maximum length of a right-hand side of a production

Once a path from a leaf to the root has more than n internal nodes, where n is the number of nonterminals in the grammar, one nonterminal has to occur twice on such a path

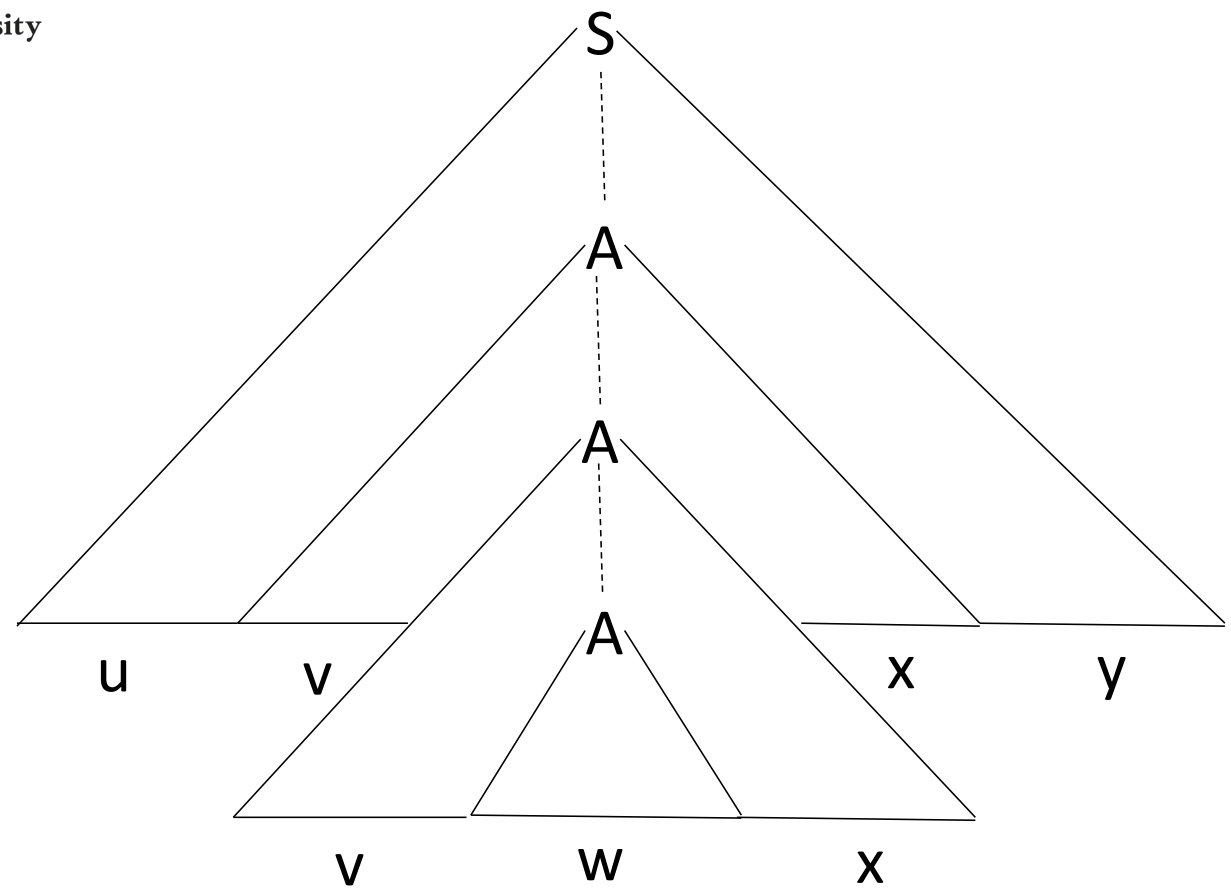
Utrecht University



Utrecht University



Utrecht University



Utrecht University

# The situation

If a sentence is long enough, we have a derivation of the form

$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvwxy$
where $|vx| > 0$

Because the grammar is context-free:
$A \Rightarrow^* vAx$
$A \Rightarrow^* w$

It follows we can derive
$S \Rightarrow^* uAy \Rightarrow^* uv^iwx^iy$
for any i in N

Utrecht University

# A property satisfied by all context-free languages

**Pumping lemma for context-free languages**
(Bar-Hillel, 1961)

For every context-free language L there exist c, d ∈ N

such that for every word z in L with $|z| \geq c$

we can split z into five parts, y = uvwxy, with $|vx| > 0$ and $|vwx| \leq d$

such that for every i ∈ N, we have $uv^iwx^iy \in L$

Extremely informally: a CFG cannot recognize a language when it needs to maintain **two** counters that can become arbitrarily large

Utrecht University

# Using the pumping lemma - strategy

For **every** pair of numbers c and d

find a word z in L with $|z| \geq c$ (you **choose** the word)

such that for **every** splitting z = uvwxy with $|vx| > 0$ and $|vwx| \leq d$

there exists a number i (you **choose** the number)

such that $uv^iwx^iy \notin L$ (you have to **show** it)

Utrecht University

# Example

The language $L = \{a^m b^m c^m \mid m \in N\}$ is not context-free

Proof:

Assume L is context-free

Next slide: L does not satisfy the property of context-free languages

The contradiction means our assumption is wrong: L is not context-free

Utrecht University

# Example

$L = \{a^m b^m c^m \mid m \in N\}$

Let r = max c d

Take $z = a^r b^r c^r$

Pump z such that the part that gets pumped is at most d, with $d \leq r$

The pumped part can thus not contain a's, b's **and** c's, but is not empty either, leading to a contradiction


# Q2

Utrecht University

# Summary

Different kinds of classes (types) of grammars vary in expressive power

Pumping lemmas describe a property that all languages of a particular type satisfy

We can use pumping lemmas to show that a language is not regular or not context-free