X AUTOMATA AND FORMAL LANGUAGES

Tenth Lecture

27 October 2022

GOAL The equivalence problem for regular grammars is solvable.

Theorem If I, I' are irreducible. Then

$$\mathcal{L}(I) = \mathcal{L}(I') \iff I \cong I'.$$

PROCEDURE : Given regular grammars G, G'

STEP 1 Produce automata D, D'

STEP 2 Form equivalent irreducible automata I, I'

Step 2a Remove inaccessible states

Step 2b Form the quotient

Step 3 Check whether $I \cong I'$.

GOAL The equivalence problem for regular grammars is solvable

Theorem If I, I' are irreducible. Then
$$\mathcal{L}(I) = \mathcal{L}(I') \iff I \cong I'$$

PROCEDURE : Given regular grammars G, G'

STEP 1 Produce automata D, D'

ALGORITHMIC "subset construction"

STEP 2 Form equivalent irreducible automata I, I'

Step 2a Remove inaccessible states

Prove to be algorithmic in Lecture IX

Step 2b Form the quotient

Step 3 Check whether $I \cong I'$.

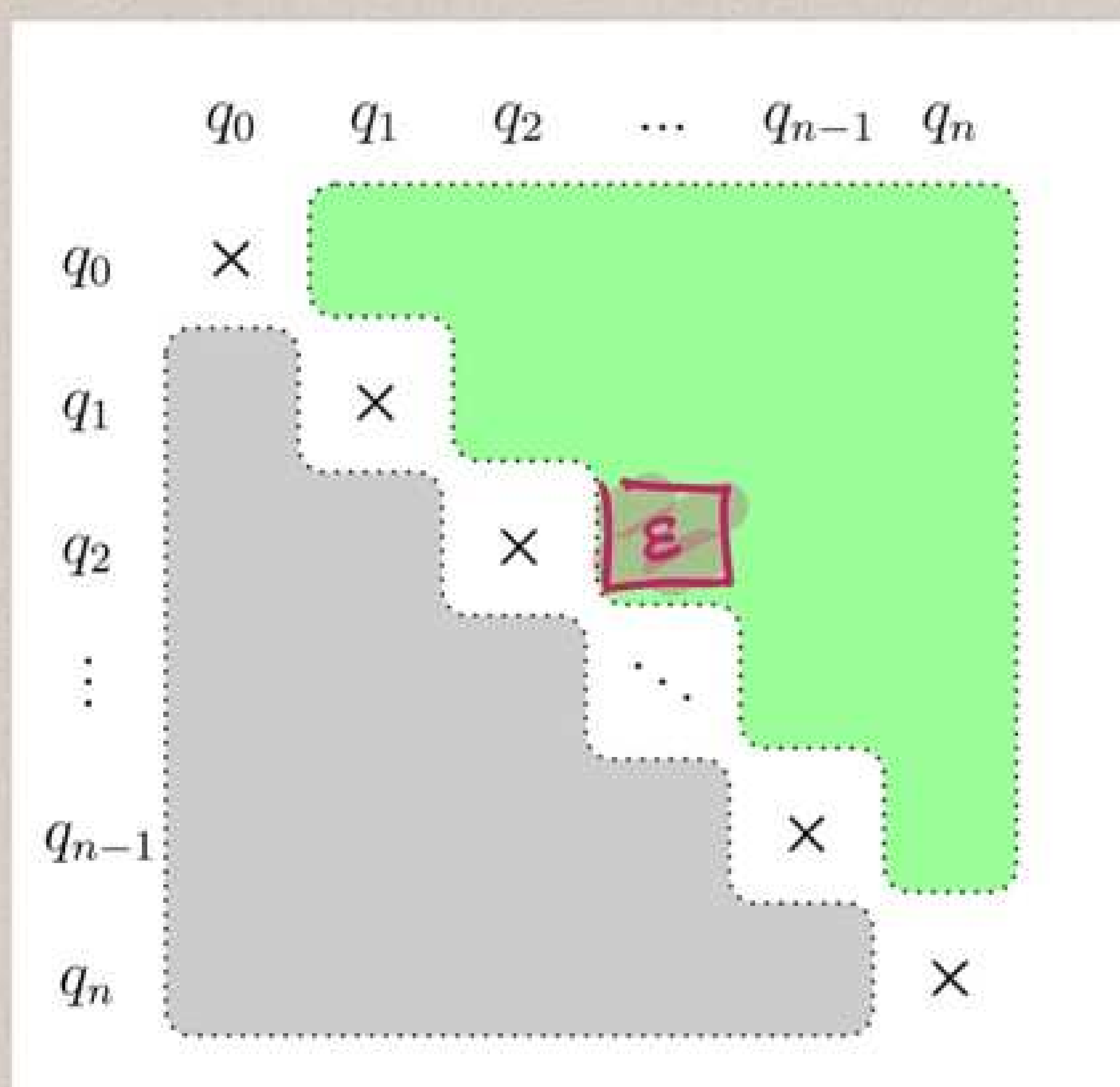
It remains to be shown that this is algorithmic.

We need:

Theorem There is an algorithm s.t. if D is an automaton, $q, q' \in Q$, then the algorithm determines whether $q \sim q'$.

The TABLE FILLING ALGORITHM:

Form matrix $Q \times Q$ and enter information about distinguishability of (q,q') in component (q,q').



By symmetry, we only need the green part!

STEP 0: If $q \in F$ & $q' \notin F$ or vice versa, then ε distinguishes q, q', so write ε in (q,q').

$\sim n^2/2$ substeps

STEP $n \longrightarrow$ STEP $n+1$: For each (q,q') and each $a \in \Sigma$, check $(\delta(q,a), \delta(q',a))$. If w is in that node, then write aw in (q,q').

$\sim n^2/2 \cdot |\Sigma|$ substeps

We say the algorithm TERMINATES if at any point, no new node is marked between STEP n & STEP $n+1$.

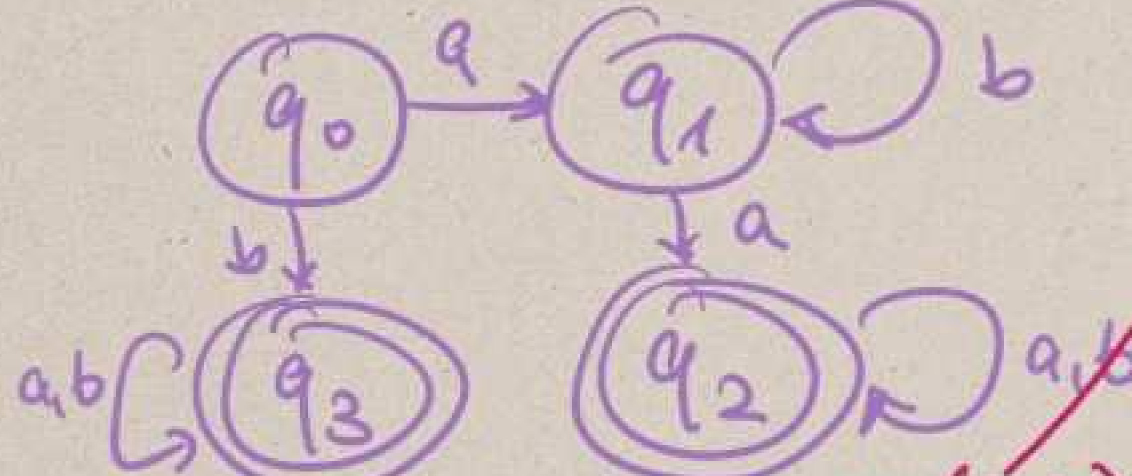
This will happen at some finite point since there are only finitely many elements of $Q \times Q$.

Claim (q, q') is marked in the algorithm

$$\Longleftrightarrow$$

$$\cdot \quad q \not\sim q'$$

EXAMPLE.



	q_0	q_1	q_2	q_3
q_0	×	a	ε	ε
q_1		×	ε	ε
q_2			×	×
q_3				×

STEP 0 Deals with (q_0, q_2), (q_0, q_3), (q_1, q_2) and (q_1, q_3)

STEP 1 Check $\delta(q_0, a) = q_1$; $\delta(q_1, a) = q_2$, (q_1, q_2) is marked ε, so (q_0, q_1) is marked $a\varepsilon = a$.

STEP 2 ~~Check that~~ Observe → $\boxed{\begin{aligned}\delta(q_3, a) &= \delta(q_3, b) = q_3 \\ \delta(q_2, a) &= \delta(q_2, b) = q_2.\end{aligned}}$

NOTHING CHANGES

→ TERMINATES

Proof of Claim

Clearly, if (q,q') is marked by w, then by construction

$$\hat{\delta}(q,w) \in F \ \& \ \hat{\delta}(q',w) \notin F$$

or vice versa,

so w distinguishes q, q'.

Suppose for contradiction that the converse doesn't hold.

So there is (q,q') unmarked, yet distinguished

$\boxed{\text{BAD PAIR}}$

Suppose bad pairs exist; pick bad pair (q,q') s.t. the distinguishing word has minimal length.

Clearly, by STEP 0, the word can't be ε.

So the word is of the form aw for some $a \in \Sigma$.

Clearly, $\delta(q,a)$ & $\delta(q',a)$ are distinguished by w. But they are unmarked [o/w STEP $n+1$ would have marked (q,q')]

Contradiction to minimality of BAD PAIR (q,q').

q.e.d.

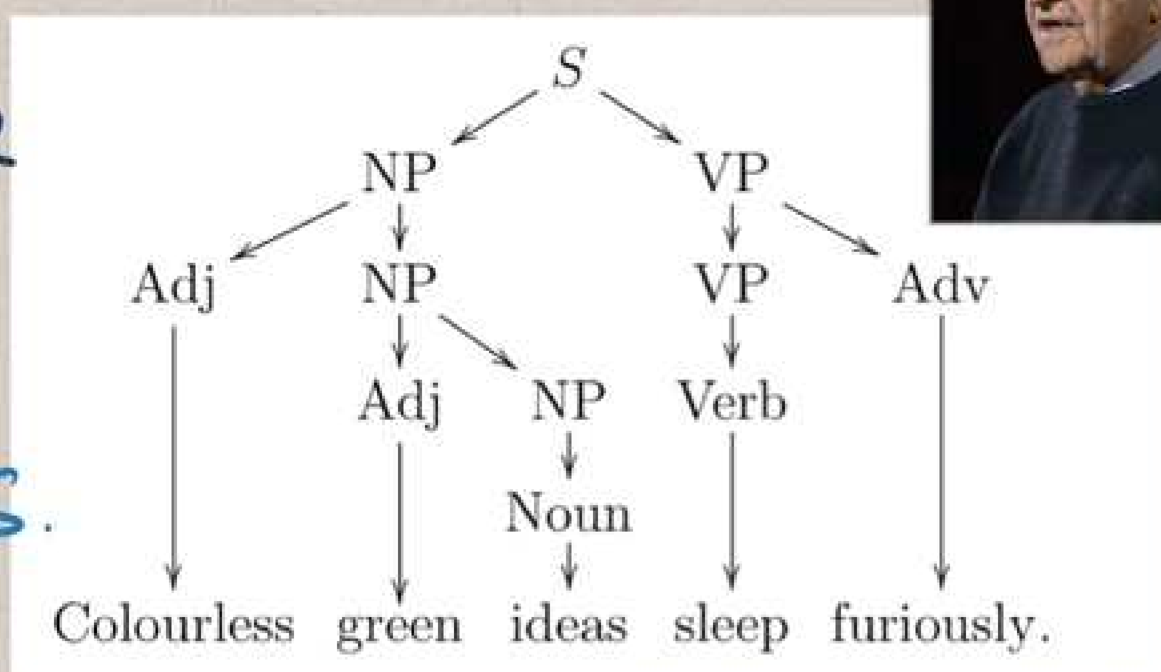
<u>Corollary</u> The equivalence problem for regular grammars is solvable.

So, for regular grammars all problems we care about have positive solutions.

Chapter 3 CONTEXT-FREE GRAMMARS

Reminder: $\{0^n 1^n ; n > 0\}$ is context-free but not regular.

The structure of context-free derivations is determined by PARSE TREES.



From Chapter 1: Chomsky's grammatical nonsensical sentence

§ 3.1 Parse trees (finitely branching)

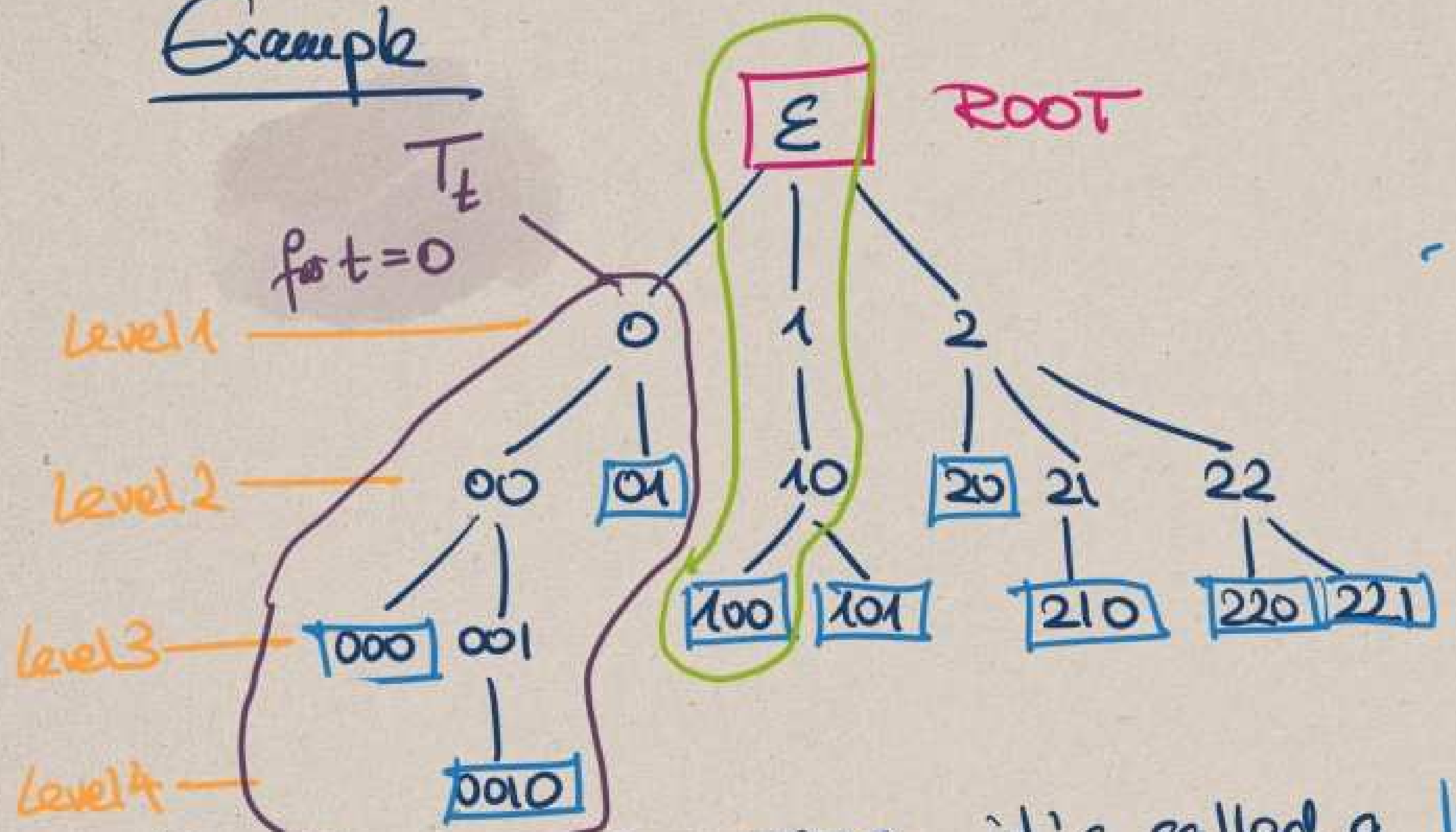
We say $T \subseteq \mathbb{N}^*$ is called a tree if

(a) it is closed under initial segments (i.e., $t \in T$, $s \subseteq t \Rightarrow s \in T$)

(b) For each $t \in T$ there is $n \in \mathbb{N}$ branching number s.t.

$$\forall k \quad tk \in T \iff k < n.$$

Example



If t has no successors, it's called a leaf.
terminal node

A node $t \in T$ has level k if $|t| = k$

If T is finite, there is a maximum level: called the height of the tree.

If t is a node, then

$$t\restriction 0,\ t\restriction 1,\ \dots,\ t\restriction |t| = t$$

is called the branch leading to t.

If $t \in T$, then $T_t := \{s \,;\, ts \in T\}$ is the subtree starting from t.

We can define a partial order on T by

$$t < s :\Longleftrightarrow t \neq s \text{ and if there is } k \text{ s.t. } t(k) \neq s(k) \text{ and } k_0 \text{ is minimal with this property, then } t(k_0) < s(k_0).$$

LEFT-TO-RIGHT ORDER

It's only partial since it does not order two distinct node that lie on the same branch.

But, it is a total order on:

(a) all nodes of a fixed level k

(b) all leaves.

MOTIVATION for PARSE TREES:

Assign elts of Ω to the nodes s.t. elts of V are assigned to non-leaves and elts of Σ are assigned to leaves.

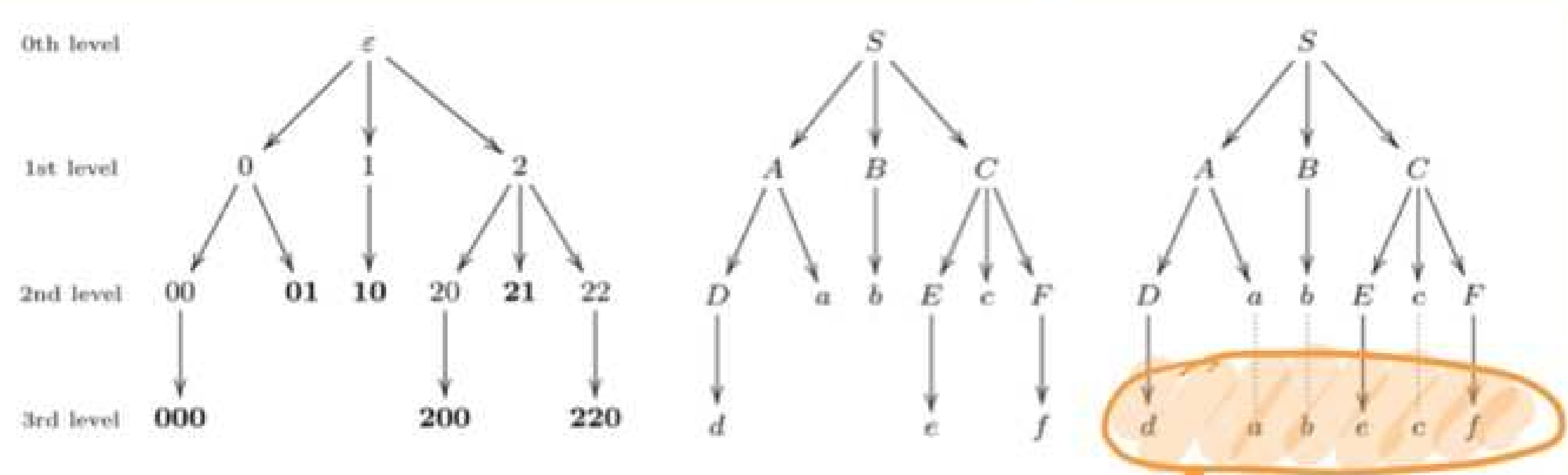


Figure 2: *Left.* A finitely branching tree; leaves are marked in boldface font. *Middle.* The same tree labelled to form a G-parse tree. *Right.* The same parse tree with the leaves extended to the final level to highlight that the parse tree parses the word $dabecf$.

We can read off the parsed word from the tree by reading the final level from left-to-right.