God for "wird"/way

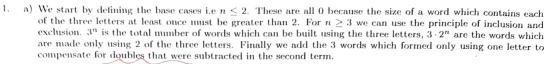
formalism => try to use

the well defined structurer

(logic, -) you abready boom!

like "thus, was we do how IX/2/X// VX'el level

Y explicately state that



- could be appaired more clearly, but it's covered-• $n = 0 \Rightarrow 0$ • $n = 1 \Rightarrow 0$
- $n \ge 3 \Rightarrow 3^n 3 \cdot 2^n + 3$
- b) We define:
 - $a_n := \#$ of words which end in 1 with length n and do not contain 11 as a subword.
 - $b_n := \#$ of words which end in 0 with length n and do not contain 11 as a subword

The solution to are task is: $sol_n := a_n + b_n$ We now derive the following two formulas:

- i. $b_n = b_{n-1} + a_{n-1} = sol_{n-1}$ because we can append anything to a word which ends in 0
- ii. $a_n = b_{n-1} = b_{n-2} + a_{n-2} = sol_{n-2}$ because words of size n-1 must end in a 0 for us to be able to append a 1

for $n \geq 3$

- $sol_0 = 0$
- $sol_1 = 2$
- $sol_2 = 3$

u yn 7/3, we have sol = sol -...

• $sol_{n\geq 3} = sol_{n-2} + sol_{n-1}$ 4 how to she recurrence ("see that it as a) We assume there is a non-empty finite language $L \neq \{\lambda\}$ satisfying $L^2 = L$ and hence it must contain a largest file.

element x according to the canonical ordering. \Rightarrow by definition of concatination there is an element $x' \in L^2$ such that $x' = x \cdot x$ x explain, how cononical ordering is related to leady

 \Rightarrow since |x'| > |x| and x was the largest element in L it follows that $x' \notin L$ \Rightarrow No non-empty finite Language L exists such that $L \neq \{\lambda\}$ and $L^2 = L$

b) We define the three languages as follows:

• $L_1 = \{0\}^*$

• $L_2 = \{0\}$ • $L_3 = \{00\}$

 $\Rightarrow (L_2 \cap L_3) = \{0\} \cap \{00\} = \emptyset$ $\Rightarrow L_1 \cdot (L_2 \cap L_3) = L_1 \cdot \emptyset \neq \emptyset$

 $\Rightarrow L_1 \cdot (L_2 \cap L_3)$ is finite Lyt= Ly at Ly = (05 = 1 You wear $L_1L_2=L_1^+$ and $L_1L_3=L_1\setminus\{\lambda,0\}$

 $\Rightarrow L_{\bullet}^{+} \cap L_{1} \setminus \{\lambda, 0\} = L_{1} \setminus \{\lambda, 0\} \qquad \Rightarrow \text{given why?}$

 $\Rightarrow L_1 \setminus \{\lambda, 0\}$ is by definition infinite because an infinite set mimnus a finite set is infinite ~ 0 3. We must prove: An infinite language L is recursive \iff there is an algorithm enumerating L

Assume L is recursive, then there exists and Algorithm A which solves the Decision Problem (Entscheidungsproblem) [7,0] \Rightarrow We iterate through each element $x \in \sum^*$ in canonical order and decide with A if we add it to our enumeration. \Rightarrow There is an algorithm enumerating L I when to terminate

Assume there is an algorithm A which enumerates L

We define an algorithm B which, when given an arbitrary $x \in \sum^*$ goes through the enumeration which is in canonical order and checks if x is equal to the current element e in the enumeration. If |x| > |e| then B outputs "x not in L".

Since $|x| < \infty$, B will terminate in a finite amount of time. If x = e then we output "x in L".

 $\downarrow \Rightarrow$ B solves the Decision Problem for L

>⇒ L is recursive

the also laster some details. They to

> this would terminate instally in most come you prob. weny 10/7/X/

So we in deple