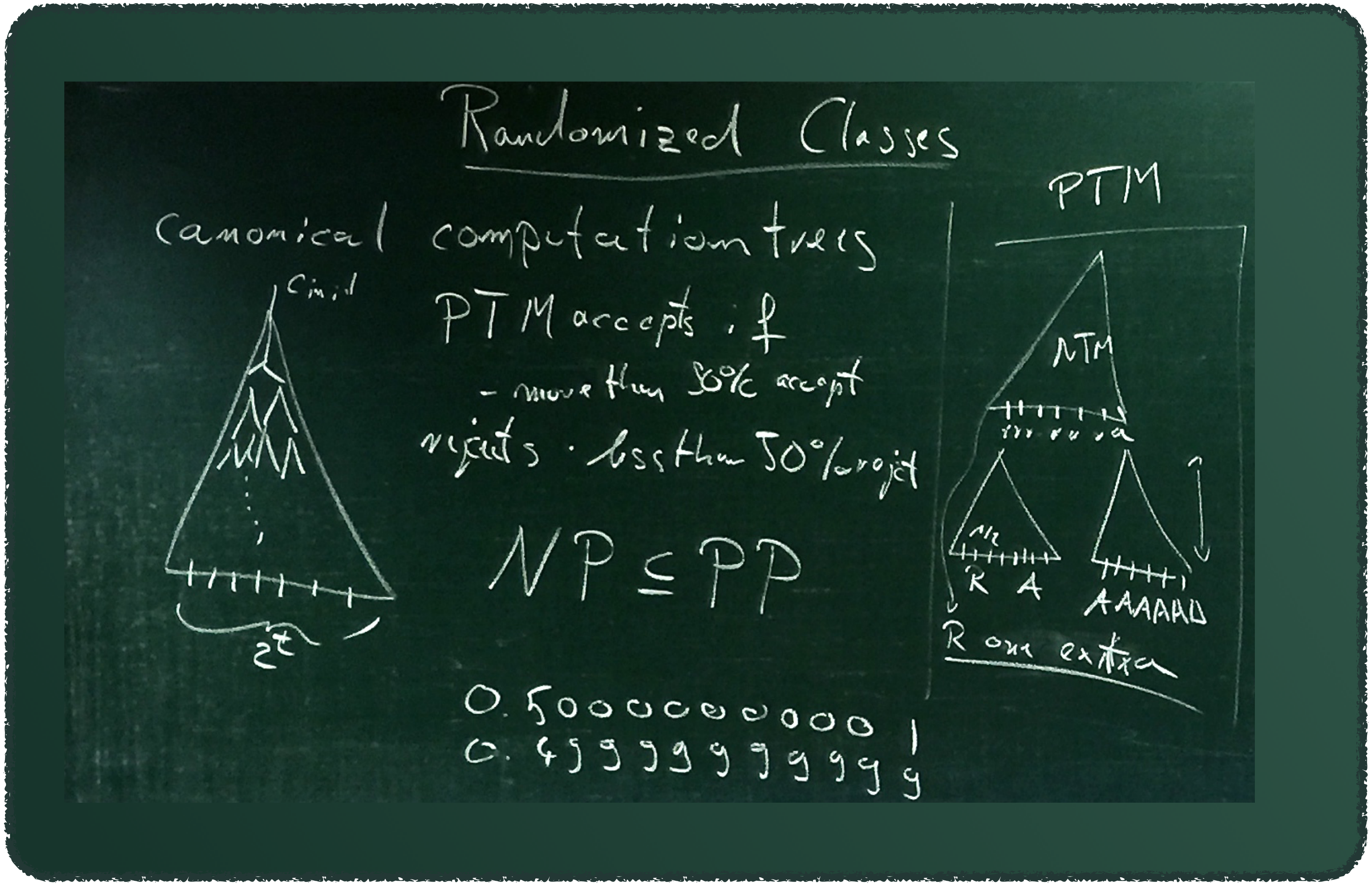
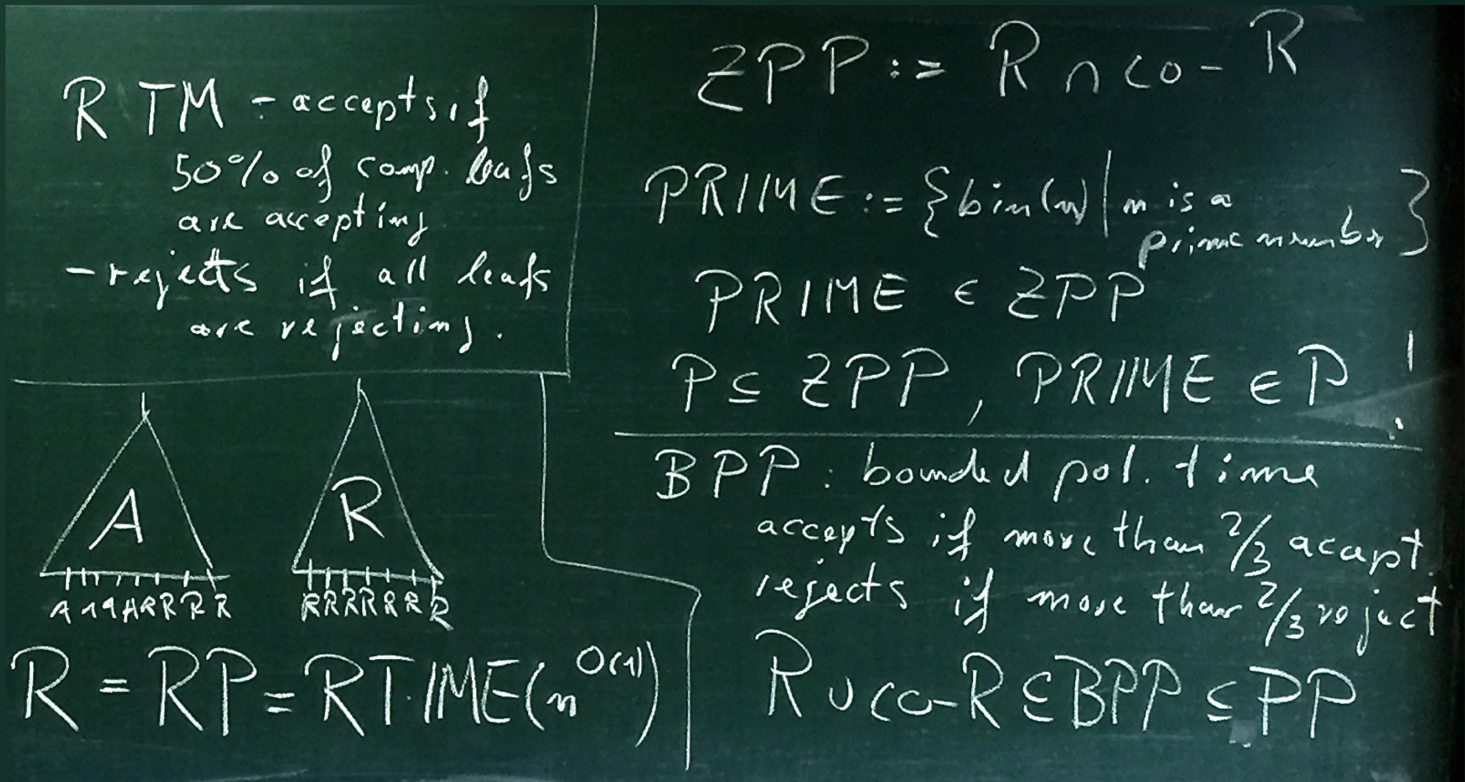
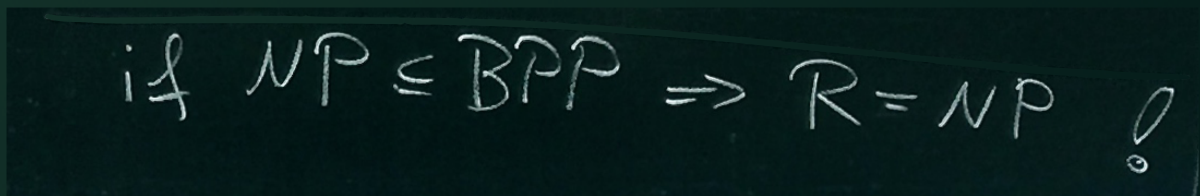
CoNe Freiburg

7 R, ZPP, BPP, Savitch and ATMs

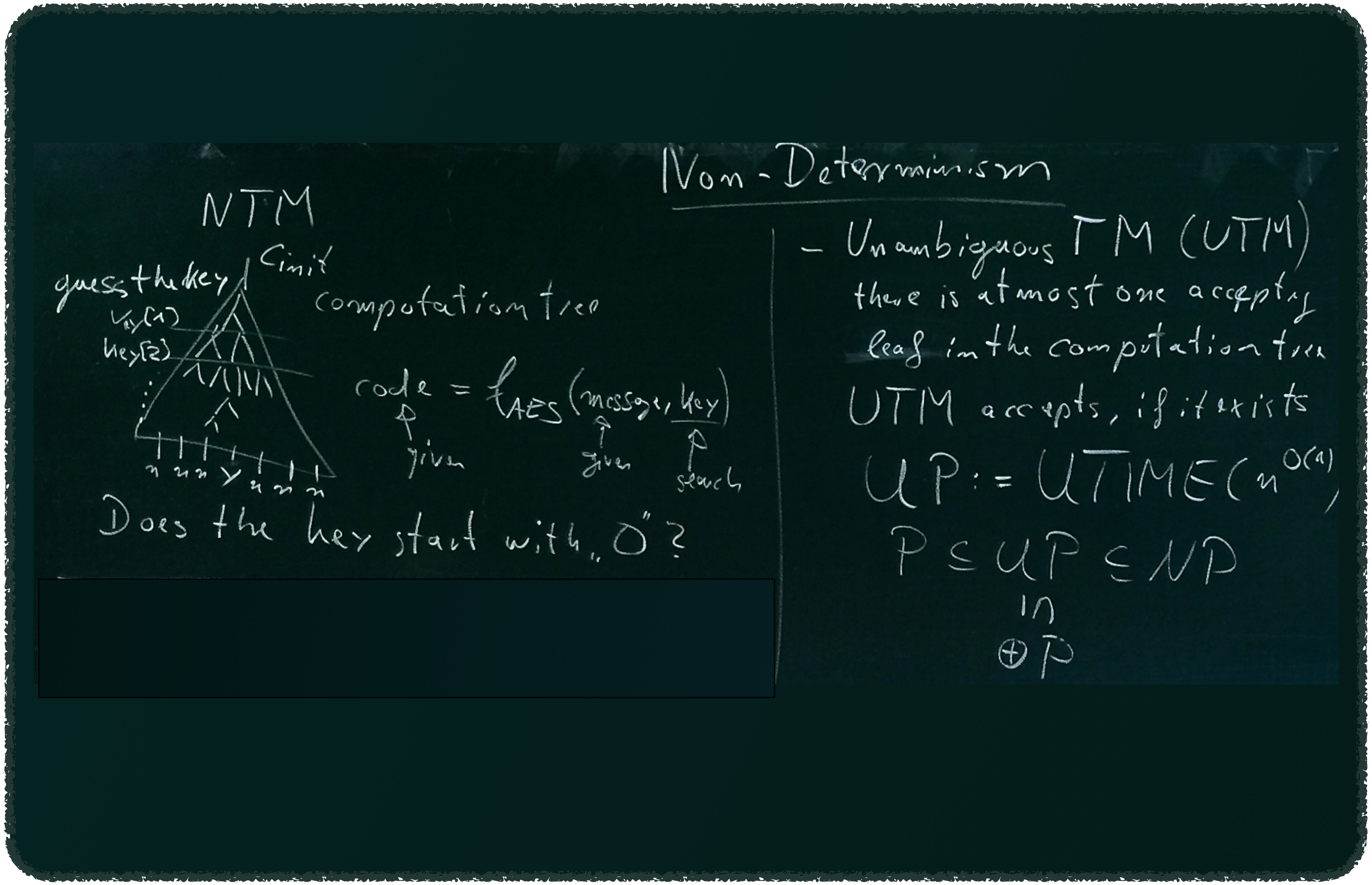


7 R, ZPP, BPP, Savitch and ATMs





7 R, ZPP, BPP, Savitch and ATMs



7 R, ZPP, BPP, Savitch and ATMs

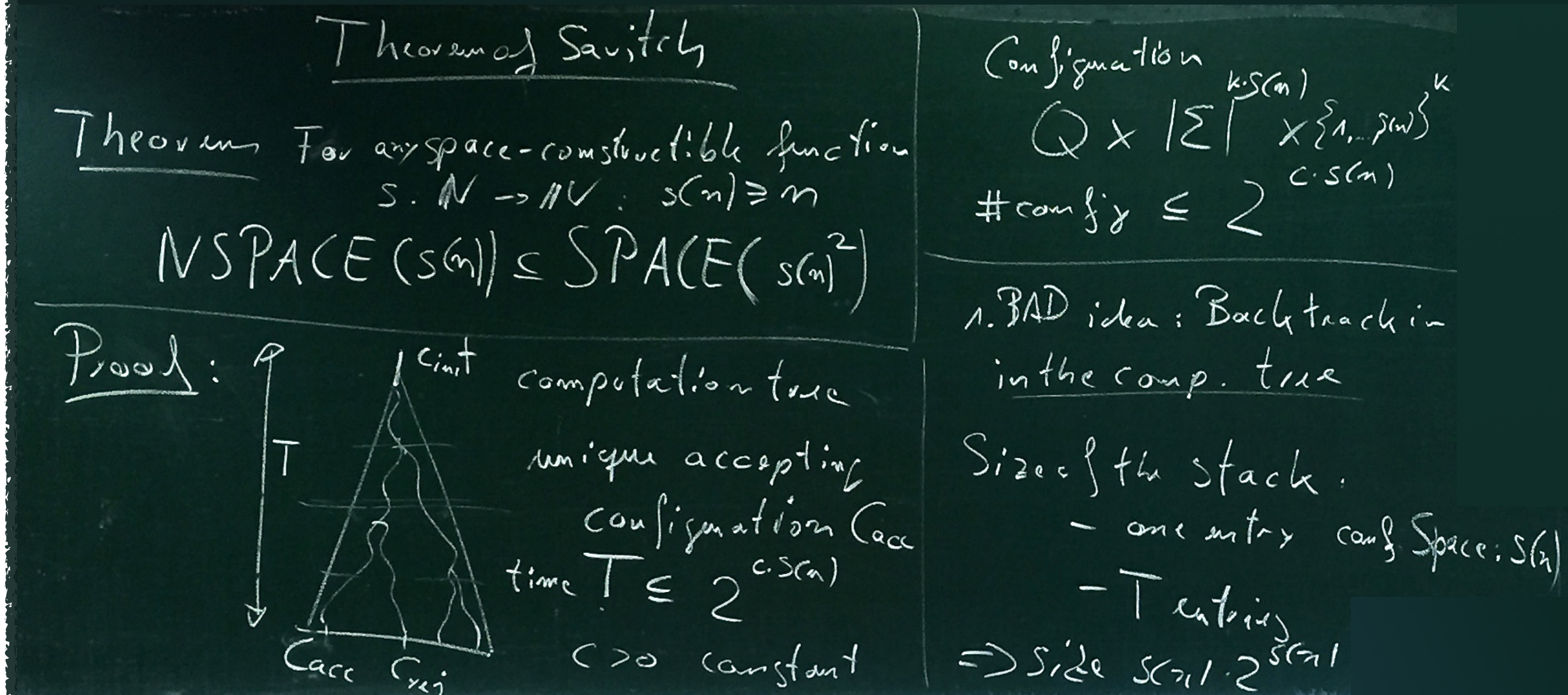
- Few-TM
  promise: the number of accepting leafs is bounded by a polynomial.

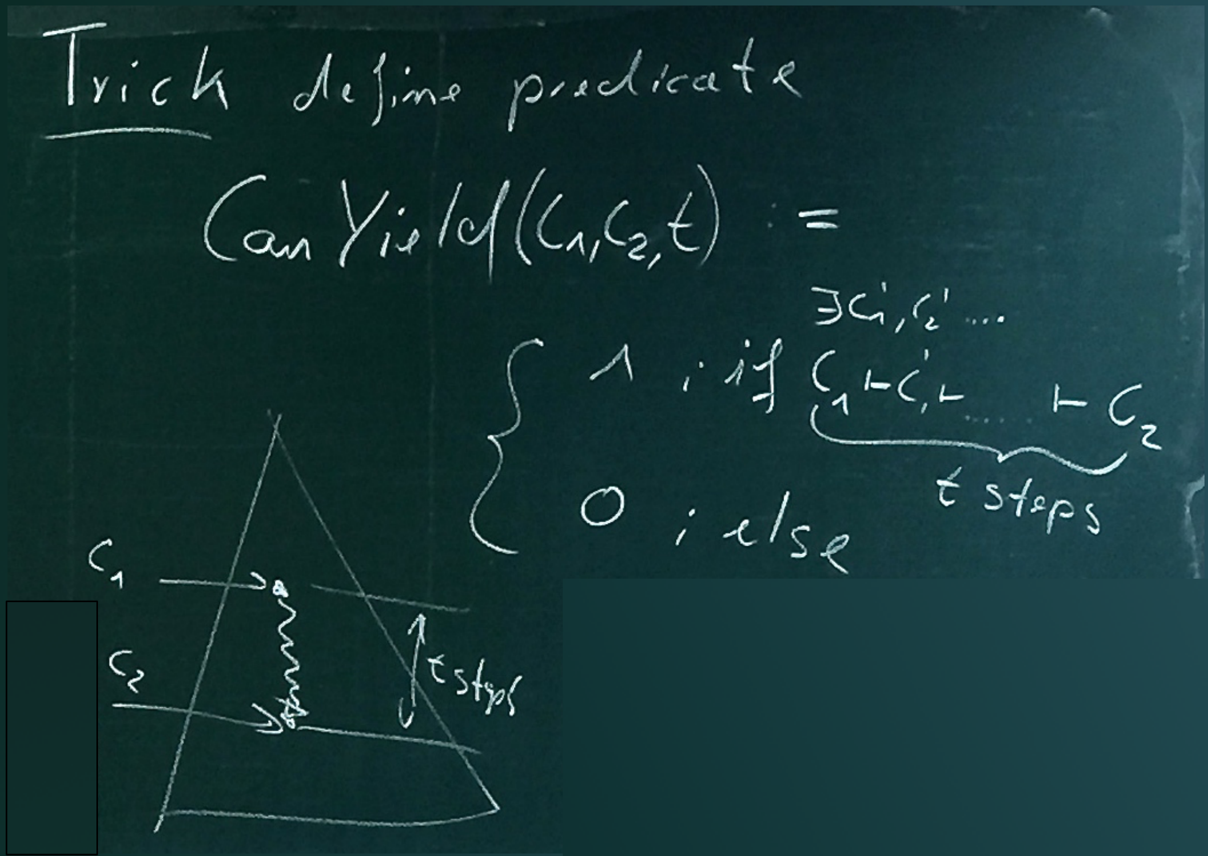
$$\mathrm{FewTIME}(n^{O(1)}) =: \mathrm{FewP}$$

$$\mathrm{UP} \subseteq \mathrm{FewP} \subseteq \mathrm{NP}$$

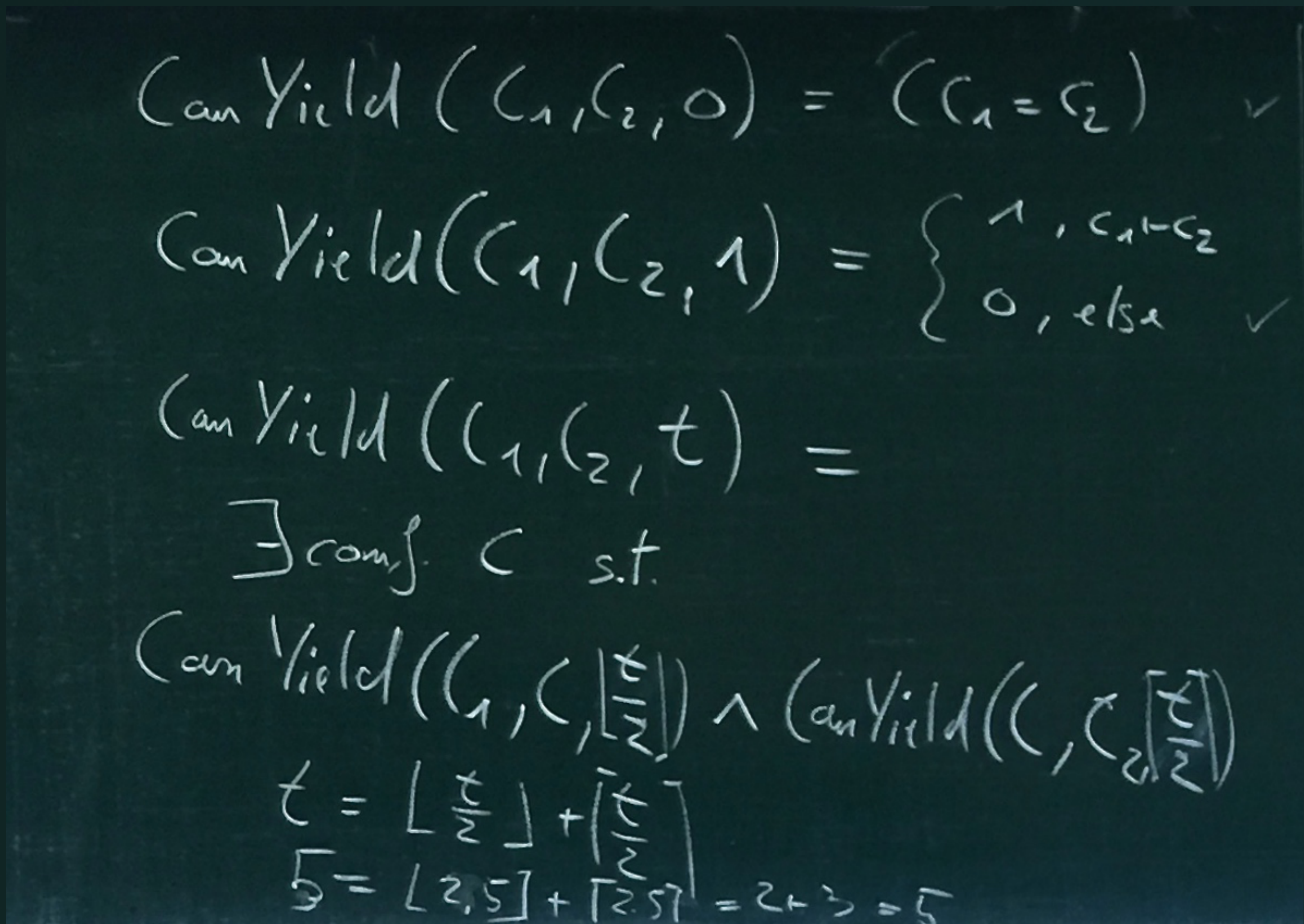
$$\mathrm{FewP} \subseteq \oplus\mathrm{P} \quad !$$

7 R, ZPP, BPP, Savitch and ATMs




Trick define predicate
$\mathrm{CanYield}(C_1, C_2, t) := \begin{cases} 1 & ; \text{ if } \exists C_1', C_2' \ldots \; \underbrace{C_1 \vdash C_1' \vdash \ldots \vdash C_2}_{t \text{ steps}} \\ 0 & ; \text{ else} \end{cases}$
C_1
C_2
t steps

7 R, ZPP, BPP, Savitch and ATMs



CoNe
Freiburg

7 R, ZPP, BPP, Savitch and ATMs

Alg CanYield(C_1, C_2, t)

```
if t=0 then if C1 = C2 then
                 RETURN true
            else RETURN false
else if t=1 then if C1 ⊢ C2 then
                 true
            else false
else forall conf C of size s(n) do
      if CanYield(C1, C, ⌊t/2⌋)
        and CanYield(C, C2, ⌈t/2⌉)
od      then RETURN TRUE
   RETURN FALSE
fi
```

space needed per recursion : $O(s(n))$

stack size $\log T + 1$

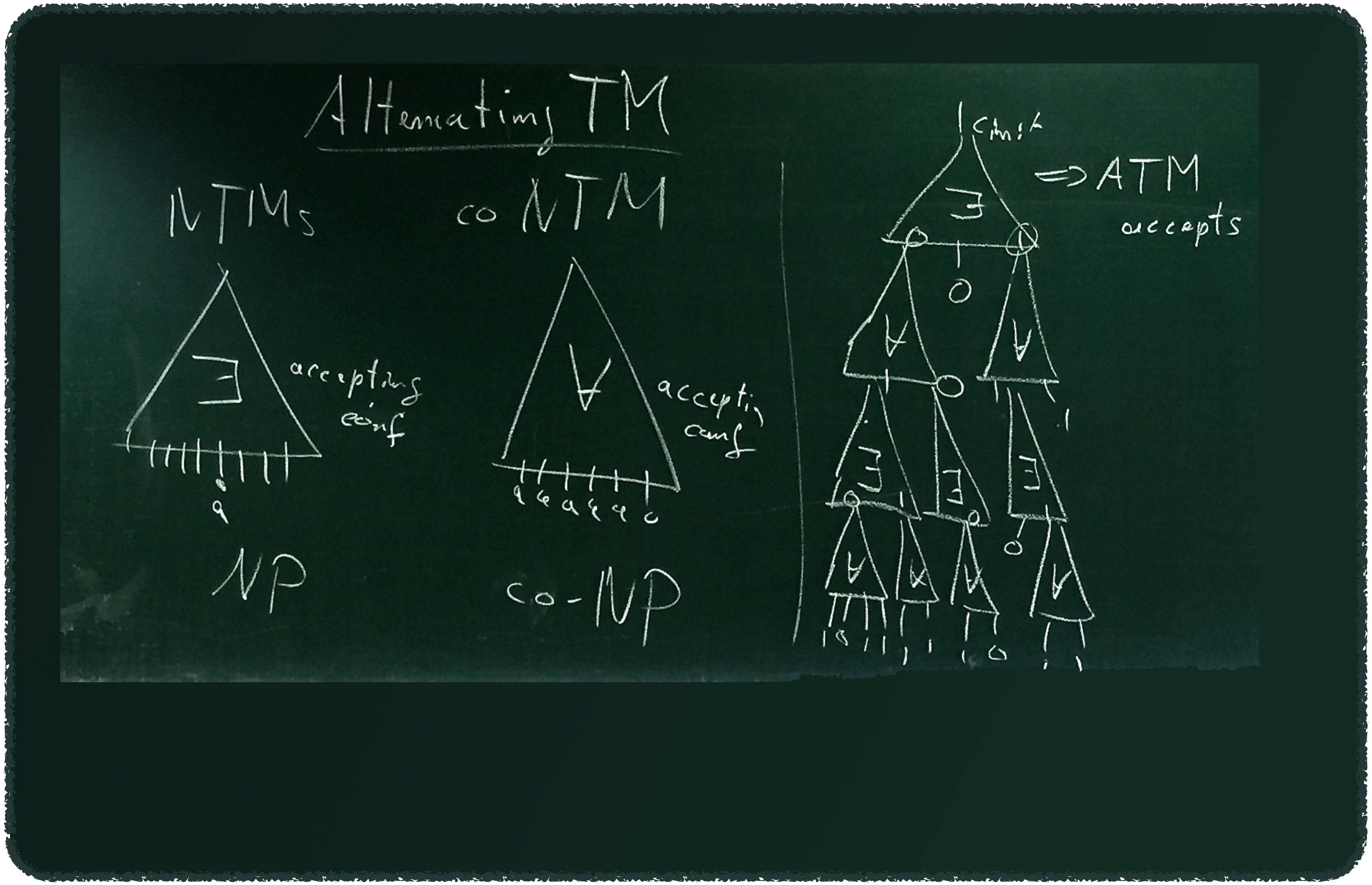
$$\underbrace{T \to T/2 \to T/4 \to \dots 1}_{\log T = O(s(n))}$$

overall space needed is $O(s(n)^2)$

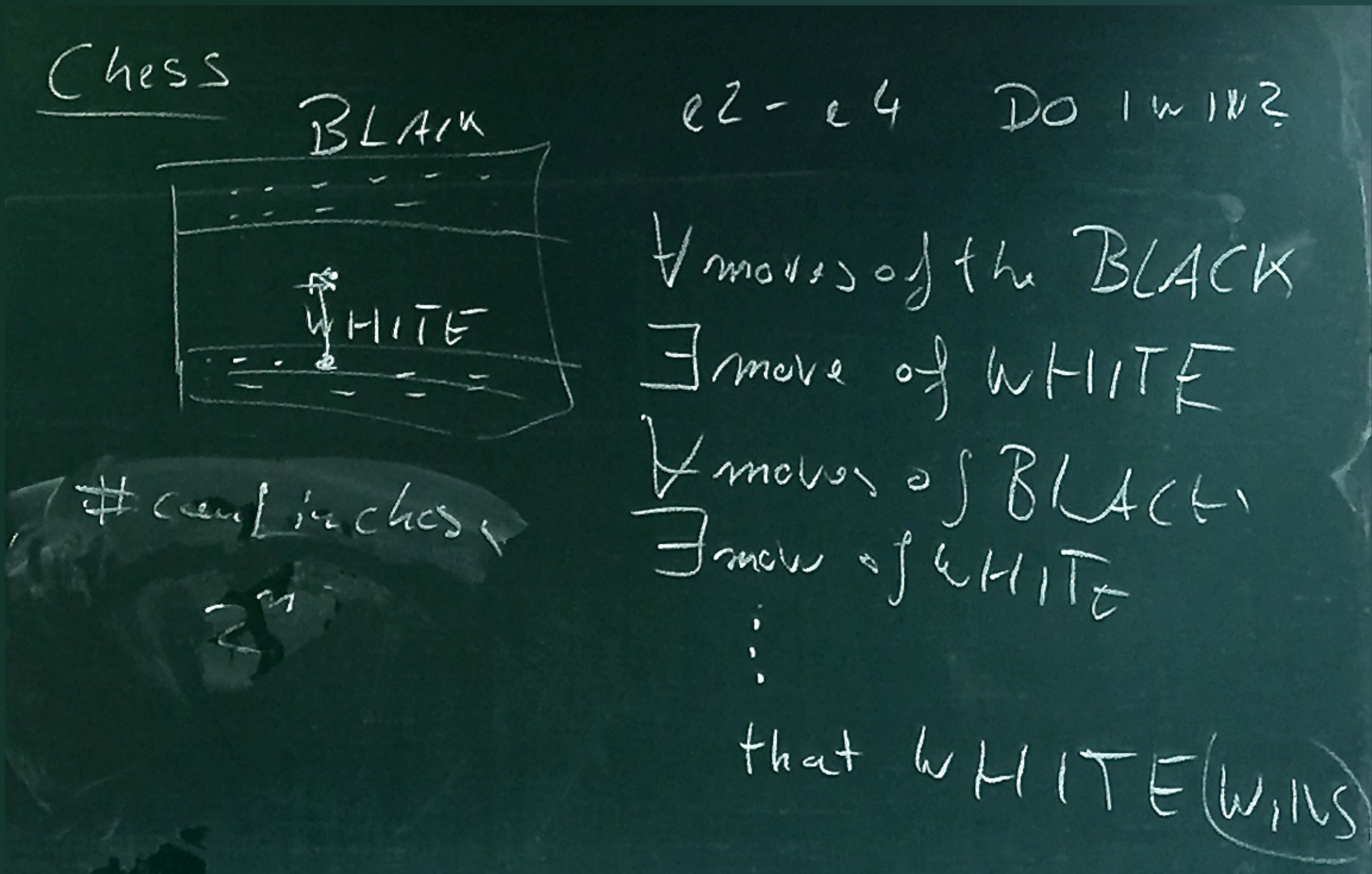
Now compute $\text{CanYield}(C_{init}, C_{acc}, 2^{c\,s(n)})$

$\Rightarrow$

PSPACE = NPSPACE


Alternating TM
NTMs
co NTM
accepting conf
NP
co-NP
start
⇒ ATM accepts

7 R, ZPP, BPP, Savitch and ATMs



7 R, ZPP, BPP, Savitch and ATMs

ATM has states Q

$$Q = \{q_{acc}\} \cup \{q_{rej}\} \cup Q_\forall \cup Q_\exists$$

ATM: M accepts in conf C

1. C is in q_{acc}
2. C is in $Q_\exists$ and
   $\exists C'.\ C \vdash C'$ and C' accepts
3. C is in $Q_\forall$ and
   $\forall C': C \vdash C' \Rightarrow C'$ accepts

$$AP = ATIME(n^{O(1)})$$

$$AP = PSPACE$$

(next time ...)