space \((|x| f)O\) uses \(f W\) steps \((|x| f + |x|) O\) halts after \(f W\)

For any \(x\):
\[(|x| f \sqcup = (x) f W\) such that \(f W\) is a nondecreasing

\textbf{Function} if the following hold:

\(\text{We say that } f : \mathbb{N} \rightarrow \mathbb{N} \text{ is a proper (complexity)}\)

\textbf{Proper (Complexity) Functions}
time \text{ and SPACE}(u) \text{ and NSPACE}(u)

to complexity classes \text{ TIME}(u) \text{ and SPACE}(u)

We shall henceforth use only proper functions in relation

recursive function \( f \) (the \textbf{gap theorem})

for some \((u) f \in \text{ TIME} = (u) f \in \text{ SPACE}\) –

\text{ for complexity classes spoil the theory building.}

Nonproper functions when serving as the time bounds

If and g are proper, then so are \( f + g, 9 f, \) and \( 2 g \).

polynomials of \( u, \sqrt{u}, \log u, \) etc.

Most "reasonable" functions are proper: \( c, \log u \), etc.

Examples of Proper Functions
So we can count steps to prevent infinite loops.

For some $c$ (p. 128),

- The space-bounded computation must repeat a
  configuration if it runs for more than $cn$ steps

- The space-bounded computation must run of length $f(u)$ first.

Run the TM associated with $f$ to produce an output:

- When the space is bounded by a proper function $f$, computations can be assumed to halt.

- TMs are not required to halt at all.

In the definition of space-bounded computations, the

Space-Bounded Computation and Proper Functions
$W$ can be deterministic or nondeterministic.

The first and the last strings.

If $W$ is a TM with input and output, we exclude $W \neq *$

$\cdot (u)b$

All of its strings are at halting of length precisely

$W$ halts after precise steps, and

$\forall$ every computation path of $W$.

For every $u \in \mathbb{N}$, for every $x$ of length $n$, and for

that there are functions and such $b$ and $f$

You're Turing Machines

Precise Turing Machines
or an "alarm clock".  

\( \text{if } M \text{ is output of length } |x| \) \( f \) will serve as a "yardstick"  

with the proper function on \( x \).  

\( f \) \( M \) on input \( x \) first simulates the TM \( M \).  

\( \text{respectively).} \)

\( ((u)f)O \times (uv)(f)O \) (or space)  

which decides \( T \) in time \( O \) (or space) \( f(u) \), where \( f \) is proper. Then there is a precise TM \( M \)  

non-deterministic \( T \) \( M \) which decides \( T \) within time (or space)  

Proposition 10: Suppose that a (deterministic or  

precise) TM are general
The total space, besides the input string, is 

\[ O \]

The simulation on \( L \) is output string.

\[ \sum \] simulates on \( L \)’s output string.

\[ \sum \]

If \( f \) is a space bound:

\[ ((x)f + |x|)O \]

The time bound is therefore string exhaustd.

The simulation stops at the moment the “clock”
advancing the cursor on the “clock” string.

The simulation of each step of \( L \) on \( x \) is matched by \( L \). If \( f \) is a time bound:

The Proof (continued)
\[
(\log u) \text{SPACE} = \text{NL}
\]
\[
(\log u) \text{TIME} = \text{EXP}
\]
\[
(\log u) \text{NSPACE} = \text{NP}
\]
\[
(\log u) \text{PSPACE} = \text{P}
\]
\[
(\log u) \text{NTIME} = \text{P}
\]

- For example, \( \text{NTIME}^{0} \cup \cdots \cup \text{NTIME}^{\log n} \) expresses the union of all complexity classes, one for each value of \( k \).

- We write expressions like \( u \) to denote the union of all \( \text{NTime}(u) \) classes.

The Most Important Complexity Classes
Graphs without a Hamiltonian path.

- HAMILTON PATH COMPLEMENT is the set of expressions.

- SAT COMPLEMENT is the set of unsatisfiable boolean languages.

Recall that the complement of $L$, denoted by $\overline{L}$, is the not the languages not in RE.

- CORRE contains the complements of languages in RE.

From p. 89, we know $R$, RE, and CORRE are distinct.

Complements of Nondeterministic Classes.
whether non-deterministic classes for time are closed.

- Whether non-deterministic classes for time are closed.

  bounded by reversing the "yes" and "no" states.

  A deterministic TM deciding \( T \) can be converted to

  \[ \{ T : T \in \mathcal{C} \} \]

  Clearly, if \( \mathcal{C} \) is a deterministic time or space complexity

  \[ \{ T : T \in \mathcal{C} \} \]

  For any complexity class \( \mathcal{C} \), \( \mathcal{C} \) denotes the class

  The \( \mathcal{C} \)-classes.
Assume the input is binary.

where $W$ is deterministic.

After at most $|x|f$ steps

$x$ accepts input $W$:

\[ fH \]

Define $u$ be proper.

The Halting Problem Quantified
seconds to verify that claim.

It takes only 10 seconds to verify that it takes only 10

— Just because a Pentium processor can finish a job in

— needs to take into account all possible Wins.

because the simulator

may not be in TIME fH

linear speedup theorem.

Use the simulator (p. 43), the universal TM, and the

|\( |x| \)? f

clock of length

For each input \( \text{Win} x \), we simulate on \( \text{Win} \) with an alarm

Lemma 11 \( \text{TIME} \in fH \)

\((u)f_{)(u)(f)O \)

The Quantified Halting Problem is in \( O \)
```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
\[
\text{TIME}(2^n) \supseteq \text{TIME}((u)2^n) \supseteq \text{EXP}.
\]

By Theorem 13,

enough

Corollary 14

Combine Lemma 11 and Lemma 12.

The Time Hierarchy Theorem

\[
\text{TIME}(2^n) \supseteq \text{TIME}((u)f(2^n) + 1) \supseteq \text{EXP}
\]

If \( n \) is proper, then
Corollary 16: If $f \subseteq \text{SPACE}$,

$((u)f (u)f \log (u)f \subseteq (u)f) \subseteq (u)f)$

Theorem 15: If $(u)f$ is proper, then

The Space Hierarchy Theorem
out degree greater than one.

The start node (representing the initial configuration)

connects two nodes if one yields the other.

all the TM configurations as its nodes and edges

The reachability method images a directed graph with

transitions between configurations.

A computation of a TM (deterministic or

The Reachability Method
it takes time \(((u)f)O\) and space can be recycled.

- Each path consumes at most \((u)fO\) space because
- Use the depth-first search as \(f\) is proper.
- Explore the computation tree of the NTM for "yes."

Proof of 2:

\[
((u)_f + u) \log \gamma (u) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq ((u)f) \subseteq \text{TIME} \subseteq \text{NSPACE} \subseteq (u) \text{ is proper.}

Then

Theorem 17 Suppose that \((u)\) is proper. Then

Relations between Complexity Classes
For some \( c \), which depends on \( \mathcal{W} \),

\[
(\gamma^u f_u^+ + \iota_0 \gamma) \mathcal{O} = (\gamma^u f_{-\gamma}^u + \iota_0 \gamma) |\mathcal{X}| \times (1 + u) \times |\mathcal{Y}|
\]

The number of configurations is therefore at most

* First cursor.

\( \ell \) is an integer between 0 and \( n \) for the position of the

where

We only care about \((b, n, \ldots, n, m, n, \ldots, n, m, n, b, n, \ldots, n, m, n, b, n, \ldots, n)\)

A configuration is a \((\gamma + 1)\)-tuple

\(((\gamma^u f_t)_{t \in 1}) \in \mathcal{E} \cap \mathcal{N}\) that decides

\( (\gamma^u f_t)_{t \in 1} \in \mathcal{E} \cap \mathcal{N} \) with input and output on input \( x \) of

\( f \) generate the configuration of a \( k \)-string


Proof of \( f \) (use the reachability method):

The Proof (continued)
\[(u)_{f+u} \log (\gamma e) = \gamma (u)_{f+u} \log (\gamma e)\]

Reachability is in \text{TIME} for some \(k\) and

\[(u)_{f+u} \log (\gamma e)\]

It is in \text{TIME} because \((u)_{f+u} \log (\gamma e)\) nodes.

The problem is therefore that of \text{Reachability} on a

Graph with \(O(\log n)\) edges.

There may be many of them.

The initial configuration to some configuration of the form

there is a path in the configuration graph from the

whether \(x \in \mathcal{T}\) becomes equivalent to deciding whether

transition function.

Add edges to the configuration graph based on the

The Proof (continued)
We suspect all four inclusions are proper, but there is no proof yet.

The chain must break somewhere between $L$ and $PSPACE$.

• By Corollary 16 (p. 124), we know $L \subseteq PSPACE$.

• It is known that $PSPACE \subseteq EXP$.

• $L \subseteq NL \subseteq NP \subseteq PSPACE \subseteq EXP$.

The grand chain of inclusions
(Savitch’s theorem).

Surprisingly, the relation is only quadratic, a polynomial

• How about $NSPACE$ and $SPACE$?

Inherent.

— There is no proof that the exponential gap is

exponential gap.

By Theorem 4 (p. 69), $\text{TIME} \subseteq \text{NTIME}$, and

$\text{Non-deterministic Space}$ and $\text{Deterministic Space}$
PATH $(x, y)$, $\log n$. There is a path from $x$ to $y$ if and only if $\log n$(u).

For $\ell \geq 0$, let $\text{PATH}(x, y)$ mean that there is a path from $x$ to $y$ of length at most $2^\ell$.

Let $G$ be a graph with $n$ nodes and $x, y$ be nodes of $G$.

Reachability $\in \text{SPACE}(\log^2 n)$.

Theorem 18 (Savitch, 1970)

Savitch's Theorem
The space requirement is proportional to the depth of path of $(\gamma, \gamma)(s)$. Like stacks in recursive calls, we keep only the current search on a tree with nodes $(\gamma, \gamma)(s)$. We compute $\text{PATH}(x', \gamma)$ with a depth-first $\delta = (\gamma, \gamma)(s)$.

For $\text{PATH}(x', \gamma)(0)$, check the input graph or if $x$ exists a $\gamma$ such that $\text{PATH}(x', \gamma)$ and $\text{PATH}(x', \gamma)$ and only if there exists a $\gamma$.

The simple idea for computing $\text{PATH}(x', \gamma)$.
- Total space is $O(\log^2 n)$.
- Each node $(x, y)$ needs space $O(\log n)$.
- Depth is only $\lfloor \log n \rfloor$. 

The PATH Tree

PATH($x(z, y) \log u - 1$)

PATH($x(z, y) \log u - 1$)
The Algorithm for PATH(x, y, z)

if PATH(x, y, z) and PATH'(x, y, z) then
  \( I = I - 1 \) and return true;
else
  return false;
endif

for \( z = 1 \) to \( Z \), do
  for \( u = 1 \) to \( U \), do
    if \( x \) or \( y \) or \( z \) in \( C \), then
      return true;
      \( i = 0 \) and return.
      endif
    endif
  endfor
endfor

return false;
nodes; hence each node takes space \( O \).

From p. 128, the configuration graph has \( O \).

When \( i = 0 \), by examining the input string:

The graph is implicit—we check for connectedness only.

The NTM on the input.

Apply Savitch's theorem to the configuration graph of

\[
((u)f)^2 \subseteq \text{SPACE}(u)
\]

Corollary 19 Let \( f \) be proper. Then

Deterministic Space Only Quadratic

The Relation between Nondeterministic Space and

\[
(u) \text{SPACE} \subseteq \text{SPACE}((u)f)^2
\]
than it is with respect to time.
Non-determinism is less powerful with respect to space

$\text{PSPACE} = \text{NSPACE}$.

Implications of Savitch’s Theorem
The concept is nontrivial only for nondeterministic complexity classes.

- There is still no hint of $\text{coNP} = \text{NP}$.
- $\text{coNEXP} = \text{NEXP}$ and $\text{coPSPACE} = \text{NPSPACE}$.
- $((u)f) = ((u)f)$

We shall prove that nondeterministic space is closed under complement.
\[(|x|)_f \leq \text{ones} \leq (u)_f \text{ at most at length } n \]

As before, the machine observes a space bound on their output.

- So all successful paths agree on their output.
- At least one computation path ends up with \( (x)_f \).
- The correct answer \( (x)_f \) or \((u)_f \) in state \textit{no}.
- On input \( x \), each computation path either outputs

\[ \text{AN NTM computes function } f \text{ if the following hold:} \]

Functions and Nondeterministic TMs
How an NTM Computes a Function
So \( |S| - (u_S) \) is the desired answer.

From \( x \) by paths of length at most \( k \).

\( S(x) \) denotes the set of nodes in \( G \) that can be reached.

Let \( n \) be the number of nodes.

The algorithm has \( \log n \) nested loops.

\( \text{space } O(\log n) \).

reachable from \( x \) in \( G \) can be computed by an \( NTM \) within

Given a graph \( G \) and a node \( x \), the number of nodes

Theorem 20 (Szélkészegyvi, 1987, Immerman, 1988)

The Immerman-Szélkészegyvi Theorem
Need \( |(1 - \gamma)S| \) but not earlier ones.

\[
\begin{align*}
\text{II:} & \quad \text{return } |(1 - \gamma)S| \\
\text{II:} & \quad \text{end for}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad j := |(\gamma)S| \\
\text{II:} & \quad \text{end for}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad \text{end if}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad i + j := j
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad \text{if } (\gamma)S \in n \text{ then}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad \text{for } \gamma \in 1, 2, \ldots \text{ do}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad 0 := j
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad \text{compute } \{ |(1 - \gamma)S| \text{ from } |(\gamma)S| \}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad \text{for } \gamma = 1, 2, \ldots \text{ do}
\end{align*}
\]

\[
\begin{align*}
\text{II:} & \quad i := |(0)S| = 1
\end{align*}
\]

The Algorithm: Top 2 Levels
The Third Loop, for $n \in S^{(Y)}$

1: $m := 0$;
2: $\text{false}$;
3: for $v = 1, 2, \ldots, u$ do
4: if $v \in (I - 1)S^{(Y)} \in S^{(Y)}$ then
5: \[ + w =: w \]
6: if $\bar{v}(n,a) \subseteq (n,a)$ then
7: $x := \text{true}$;
8: $\text{false}$;
9: $\text{false}$;
10: end for
11: if $|S - I| > m$ then
12: "no"
13: end if
14: return $\text{false}$;
The Fourth Loop; for \( a \in S \setminus \{ y \} = 1 \)
encountered and it is computed.

We accept only if no accepting configurations have been

the NTM deciding \( T \in \text{NSPACE}(f) \) on input \( x \).

Run the above algorithm on the configuration graph of

\[
\text{NSPACE}(f) = \text{coNSPACE}(f)
\]

**Corollary 21** If \( f \) is proper, then

\[
|S(S')| \leq n, m, v, s, i, t, \quad f - (n, m) \quad \text{The nondeterministic algorithm needs space } O(\log n).
\]

Wrapping it up