

# Complexity

doc. RNDr. Ondřej Čepek, Ph.D.

March 8, 2021



# Contents

<b>1</b>	<b>Repetition, NS hierarchy</b>	<b>2</b>
1.1	NS hierarchy . . . . .	4

## 1 Repetition, NS hierarchy

**Theorem 1.1 (NT and DS relation).**

$$\forall f : \mathbb{N} \rightarrow \mathbb{N} : NT(f(n)) \subseteq DS(f(n))$$

*Proof.* Construct TM with following algorithm:

1.  $k = 1$
2. do
3. forall  $y$  vetev delky  $k$
4. Simuluj  $M(x)$  dle  $y$
5. If(Acc) prijmi
6.  $k++$ ;
7. until vsechny simulace  $M(x)$  dle  $y$  ktera odmlitly // pokud vsechny vetve odmitli, neexistuje pr. vypocet
8. reject

$M$  pracuje v case  $g(n) = O(f(n))$ , vypocet v 4. skonci do  $g(n)$  kroku.

Prostor:

- 1) na ulozeni  $k$  potrebujeme  $g(n)$  prostoru,  $k \leq g(n)$ .
- 2) Na simulace  $M(x)$  dle  $y$  potrebujeme  $O(g(n))$  prostoru.

Korektnost:

Chceme aby det. TS prijimal stejny jazyk nako puvodni NTS. Pokud existuje pr. vypocet NTS, TS prijme dle nejakeho  $y$ . Opacne dojde k  $k = g(n)$  a TS odmitne.

Zacykleni nemuze nastat, dle definice cas. slozitosti.

Technical details:

Constructed TM should have 2 tapes:

- a) working tape
- b) tape to store  $y$  (vector that encode branch of NTM) □

**Note 1.2.** If we are constrained to use single tape, any  $k$  tape machine can be compressed to 1 tape machine. We need 2 steps:

1. compress  $\Sigma \rightarrow \Sigma^k$
2. How to deal with many heads of the original TM?  $\Rightarrow$  simulate 1 step of the original machine by many steps in new TM. In total, time complexity is  $\mathcal{O}(n^2)$  and space complexity is  $\mathcal{O}(n)$ .

**Theorem 1.3 (NS and DT relation).**

$$\forall f : \mathbb{N} \rightarrow \mathbb{N}, \forall L \in NS(f(n)) \Rightarrow \exists c_L : L \in DT\left(c_L^{f(n)}\right)$$

*Proof.* All accepting conf. leaves are bounded by the # of conf. However, # of all paths  $2^{c_L^{f(n)}}$ . How to avoid double exponent?  $\Rightarrow$  Perform BFH in conf. graph. # of edges could be quadratic, however

$$\left(c_L^{f(n)}\right)^2 = \left(c_L^2\right)^{f(n)}$$

$c_L^2$  is another constant. □

**Universal TM** Input:  $(x, y)$ , where  $x$  is Godel number of TM to be simulated.  $y$  is input of TM.

Alphabet:  $\Sigma = \{0, 1\}$ .

Working tapes: 3

1. Tape with transition table of  $M_x$ .
2. Tape with current state  $q$
3. Working tape

If  $S(n)$  is space used by  $M_x \Rightarrow$  we need  $\max(\lceil \log(t) \rceil, S(n), |x|)$ .

**Observation 1.4.** If initial TM has  $k$  tapes and time complexity is  $T(n)$  we can compress to 2 tapes with a cost of time complexity being  $\mathcal{O}(T(n) \log(n))$ .

// TODO write proof (moving blocks on tape)

Therefore simulating TM with multiple tapes could be reduces to 2 tapes, then simulate on Universal TM. Time complexity of Universal TM is dominated by finding the transition  $\Rightarrow \mathcal{O}(|x| \cdot T(n))$ .

**Definition 1.5.** Function  $f$  is *Space constructible*  $\iff \exists$  TM with unary alphabet which marks exactly  $|f(n)|$  cells on the working tape. E.g.  $\log, e^x$ , polynomials.

**Theorem 1.6 (Space hierarchy).** Let  $S_1, S_2$  are space constructible functions and

$$S_1 \in o(S_2) \Rightarrow DS(S_1(n)) \subsetneq DS(S_2(n))$$

*Proof.* Construction by diagonalization  $\Rightarrow$  find a language that is different from all in  $\mathcal{O}(S_1)$ . □

**Definition 1.7.** Function  $f$  is *Time constructible*  $\iff \exists$  TM that does exactly  $|f(n)|$  steps. You can think of it as alarm clock.

**Note 1.8.** Every time constructible function is space constructible.

**Theorem 1.9 (Space hierarchy).** Let  $S_1, S_2$  are time constructible functions and

$$T_1 \cdot \log(T_1(n)) \in o(T_2) \Rightarrow DT(T_1(n)) \subsetneq DT(T_2(n))$$

Note that  $\log(T_1(n))$  is required because of  $k$  to 2 tapes compression.

*Proof.* Construction by diagonalization  $\Rightarrow$  find a language that is different from all in  $\mathcal{O}(T_1)$ . □

**Theorem 1.10 (Savic).** Under some mild assumptions (space constructibility, functions bigger than  $\log(n)$ ) following statement is true:

$$NS(f(n)) \subseteq DS(f^2(n))$$

*Proof.* Find a path in configuration path. We cannot use neither BFS not DFS as the complexity is linear in edges which is exponential comparing to input. Therefore we use recursive algorithm which for all states  $K$  reachable by a path of length

$$\frac{c_L^{f(n)}}{2^i}$$

tries to find a path from  $C_{init} \rightarrow K \rightarrow C_{accept}$ .

As we divide path by 2 at every recursive call, recursion tree height is equal to  $\log_2(n)$ . Therefore time complexity is

$$\log_2(c_L^{f(n)}) = f(n) \cdot \log_2(c_L)$$

On each level of recursion we have to store  $C_{init}, K, C_{accept}$  which requires  $\mathcal{O}(f(n))$  space. Having  $\mathcal{O}(f(n))$  levels, total space complexity is  $\mathcal{O}(f^2(n))$ .  $\square$

**Note 1.11.** Time version of Savic would imply  $P = NP$ .

## 1.1 NS hierarchy

**Lemma 1.12 (Translation lemma).** *Let  $S_1(n), S_2(n), f(n)$  be space constructible functions, also*

$$S_2(n) \geq n, f(n) \geq n$$

*Then*

$$NS(S_1(n)) \subseteq NS(S_2(n)) \Rightarrow NS(S_1(f(n))) \subseteq NS(S_2(f(n)))$$

*Lemma allows to replace  $n \rightarrow f(n)$ .*

*Proof.* Let  $L_1 \in NS(S_1(f(n)))$  arbitrary,  $L_1$  is recognized by NTS  $M_1$  in space  $S_1(f(n))$ . We want to prove that  $L_1 \in NS(S_2(f(n)))$  by constructing  $L_2 \in NS(S_1(n))$  using padding. Define  $L_2 := \{x\Delta^i \mid M_1 \text{ accepts } x \text{ in space } S_1(|x| + i)\}$ . Where  $\Delta$  is a new symbol, that  $\notin$  initial  $\Sigma$ .

Algorithm of  $M_2, L(M_2) = L_2$  on input  $x\Delta^i$  is the following:

1. Mark  $S_1(|x| + i)$  cells on tape.
2. Simulate  $M_1$ .
3. if( $M_1(x)$  accept  $\wedge$  did not use more space than marked in 1) then accept

From the construction,  $M_2$  recognizes  $L_2$  using  $S_1(n)$  space. Consequently,  $L_2 \in NS(S_1(n))$ . Combining with assumption of the lemma

$$L_2 \in NS(S_2(n))$$

And there exists a TM  $M_3$  that recognizes  $L_2$  using  $S_2(n)$  space.

The last step is to construct  $M_4$  that recognizes  $L_1$  using  $S_2(f(n))$  space.  $M_4$  has 2 working tapes and has following algorithm ( $M_4$  on input  $x$ ):

1. Mark  $f(n)$  cells on 1st tape.
2. Mark  $S_2(f(n))$  cells on 2nd tape. Using 1st tape as "input".

3. foreach  $x\Delta^i, i = 0, 1, \dots$   
 simulate  $M_3$  on input  $x\Delta^i$ . If head of  $M_3$  is inside  $x$   $M_4$ 's head is in the same place.  
 Otherwise, head of  $M_3$  is inside padding, so  $M_4$  uses counters to track  $M_3$ 's position.  
 Counter has length at most  $1 + \log i$ .
4. if( $M_3(x)$  accept) then  $M_4$  accept.
5. else if( $1 + \log(i) \leq S_2(f(n))$ ) then  $++i$ ;  
 Counter did not overflow and is less than  $S_2(f(n))$ .
6. else if( $1 + \log(i) > S_2(f(n))$ ) then reject;

If  $M_4$  accepted input  $x$ ,  $M_3$  accepted  $x\Delta^i$  for some  $i \Rightarrow x\Delta^i \in L_2 \Rightarrow M_1$  accepts  $x \Rightarrow x \in L_1$ .  
 On the other hand, if  $x \in L_1 \Rightarrow x\Delta^i \in L_2$  for  $i = f(|x|) - |x|$  therefore counter  $i$  requires

$$\log(f(|x|) - |x|) \leq S_2(f(|x|))$$

space. □

**Note 1.13.** We can also prove Translation Lemma for  $DS, NT, DT$ .

**Theorem 1.14 (NS hierarchy for polynomials).** *Let  $\varepsilon > 0, r > 1$ . Then*

$$NS(n^r) \subsetneq NS(n^{r+\varepsilon})$$

*Proof.* From the density of rationals:

$$\exists s, t \in \mathbb{N} : r \leq \frac{s}{t} \leq \frac{s+1}{t} \leq r + \varepsilon$$

It is sufficient to prove that:

$$NS(n^{\frac{s}{t}}) \subsetneq NS(n^{\frac{s+1}{t}})$$

Assume by contradiction

$$NS(n^{\frac{s+1}{t}}) \subseteq NS(n^{\frac{s}{t}})$$

Now we use 1.12 for

$$S_1 = \frac{s+1}{t}, S_2 = \frac{s}{t}, f(n) = n^{(s+i)t}, i = 0, 1, \dots, t$$

We get

$$\forall i : NS((n^{(s+i)t})^{(s+1)/t}) \subseteq NS((n^{(s+i)t})^{s/t}) \Rightarrow \forall i : NS(n^{(s+i)(s+1)}) \subseteq NS(n^{(s+i)s})$$

Now we write for all  $i$ :

- $i = 0 : NS(n^{s(s+1)}) \subseteq NS(n^{s^2})$
- $i = 1 : NS(n^{(s+1)(s+1)}) \subseteq NS(n^{(s+1)s})$
- ...
- $i = s : NS(n^{(s+1)2s}) \subseteq NS(n^{(2s)s})$

From the exponents we conclude that for every inequality right side is a subset of left side.  
 So we get a chain of subsets and can use Savic theorem to get a contradiction:

$$NS(n^{(s+1)2s}) \subseteq NS(n^{s^2}) \subseteq DS(n^{2s*s}) \subsetneq DS(n^{2s*s+2s}) \subseteq NS(n^{(s+1)2s})$$

As the beginning and the end are the same sets and the chain of subsets has strict inclusion. □