1. Rozhodněte, zda je třída *rekurzivních* jazyků uzavřena vůči inverznímu morfismu. Popište ideu důkazu, podobně jako je tomu u věty 8.4 na přednáškách (můžete použít vícepáskový Turingův stroj).

   Ano, třída rekurzivních jazyků je uzavřena vůči inverznímu morfismu.

   Idea důkazu:

   - Aby byla třída rekurzivních jazyků uzavřená vůči inverznímu morfismu, musí platit, že pro každý rekurzivní jazyk $L_1 \subseteq \Sigma_1^*$ a pro každý morfismus $h : \Sigma_2^* \to \Sigma_1^*$ je $L_2 = h^{-1}(L_1) \subseteq \Sigma_2^*$ také rekurzivní jazyk.

   - Jazyk $L \subseteq \Sigma^*$ je rekurzivní, jestliže $L = L(M)$ pro nějaký úplný Turingův stroj M.

   - Z předchozích dvou bodů plyne, že třída rekurzivních jazyků je uzavřena vůči inverznímu morfismu, pokud je možné pro každý rekurzivní jazyk $L_1 \subseteq \Sigma_1^*$ a pro každý morfismus $h : \Sigma_2^* \to \Sigma_1^*$ sestavit úplný Turingův stroj M_2 takový, že $L(M_2) = L_2 = h^{-1}(L_1) \subseteq \Sigma_2^*$.

   - Z definic morfismu jazyka a inverzního morfismu jazyka ve studijní opoře plyne, že musí také platit:
     $w \in L_2 \Leftrightarrow h(w) \in L_1$.

   - M_2 lze sestrojit takto:

     – Pro morfismus $h : \Sigma_2^* \to \Sigma_1^*$ můžeme definovat funkci $h_0 : \Sigma_2 \to \Sigma_1^*$ takovou, že $h_0(a) = h(a)$ a $h(w) = h_0(a_1)h_0(a_2)...h_0(a_n)$.

     – Protože množina Σ_2 je konečná, tak h_0 může být zakódována v řízeni M_2.

     – Z výše uvedené ekvivalence plyne, že dvoupáskový M_2 může pracovat tak, že pro vstupní řetězec $w \in \Sigma_2^*$ na první pásce uloží na druhou pásku řetězec $h(w) \in \Sigma_1^*$ (postupně prochází vstup na první pásce a pro každý znak a uloží na korespondující pozici na druhé pásce $h_0(a)$). Poté na druhé pásce simuluje běh stroje M_1. Pokud ten přijme, přijme i M_2.

   - Je zřejmé, že pokud je M_1 úplný, M_2 musí také být úplný. Třída rekurzivních jazyků tudíž je uzavřena vůči inverznímu morfismu.

2. Uvažujte jazyk L_{LOA} kódů lineárně omezených Turingových strojů (t.j., jazyk kódů Turingových strojů takových, že v žádném jejich výpočtu neopustí vstupní hlava oblast pásky, na které bylo zapsáno vstupní slovo). Dokažte redukcí, že L_{LOA} není RE. Může vám pomoci následující nápověda.

   (a) Co víte o problému prázdnosti jazyka Turingova stroje?

   (b) Pro libovolný TS T můžeme sestrojit TS T', který na vstupu očekává vstupní slovo w stroje T následované vyznačeným úsekem použitelné pásky. Bude simulovat výpočet T na w a vhodně zareaguje, když simulovaný výpočet T skončí a také když simulace T vede k opuštění vyznačeného úseku pásky.

   Popis stroje T' je v nápovědě záměrně neformální a nejednoznačný. V důkazu svou verzi stroje T' definujte přesněji, nemusíte detailně popisovat redukci, ale musí být jasné, co T' dělá a že jde opravdu o redukci.

   - Použijeme redukci z problému prázdnosti jazyka Turingova stroje, který je charakterizován jazykem $L_{EMP} = \{\langle M\rangle \mid M \text{ je TS takový, že } L(M) = \emptyset\}$. Tento problém není ani částečně rozhodnutelný (viz. kapitola 6.4.2 ve studijní opoře) a tudíž jemu odpovídající jazyk L_{EMP} není ani rekurzivně vyčíslitelný.
   - $L_{LOA} = \{\langle M\rangle \mid M \text{ je lineárně omezený TS}\}$.
   - Sestrojíme redukci $\delta : \{0,1\}^* \to \{0,1\}^*$ z jazyka L_{EMP} na jazyk L_{LOA}. Musí platit: $\forall w \in \{0,1\}^* : w \in L_{EMP} \Leftrightarrow \delta(w) \in L_{LOA}$
   - Úplný TS M_δ implementující redukci δ přiřadí každému řetězci $x \in \{0,1\}^*$ řetězec $\langle M_x\rangle$, kde M_x je TS, který na vstupu $w \in \{0,1\}^*$ pracuje takto:
     – Pokud x není platný kód TS, M_x bude posouvat svojí hlavu směrem doprava až na první výskyt symbolu Δ a poté přijme.
     – Jinak M_x projde svůj vstup w a zkontroluje, zda má strukturu $v\omega^k\#$, kde v je platný kód vstupu TS s kódem x, ω a $\#$ jsou speciální zásobníkové symboly nepatřící do zásobníkové abecedy TS s kódem x a $k \geq 0$ je celé číslo. Pokud ne, M_x odmítne.
     (tento krok by neměl způsobit, aby se čtecí hlava M_x dostala mimo oblast na pásce, kde je zapsaný vstup, jinak důkaz nebude fungovat, ale pokud na vstupu např. je pouze v bez speciálních symbolů, tak M_x při kontrole nutně vyjede ze vstupu, tudíž ani nemůže být lineárně omezený, ale nevím, co s tím)
     – Jinak M_x s využitím univerzálního TS M_u, který je jeho komponentou, simuluje běh TS s kódem x na vstupu w, přičemž k speciálním symbolům ω se chová, jako by to byly symboly Δ, a pokud se jeho hlava dostane nad symbol $\#$, M_u zastaví a odmítne.
     – Pokud M_u odmítne, odmítne i M_x.
     – Pokud M_u přijme, M_x bude posouvat svojí hlavu směrem doprava až na první výskyt symbolu Δ a poté přijme.
   - δ lze evidentně implementovat s pomocí úplného TS M_δ. Stačí, aby vypsal kód TS zahrnujícího test členství x v regulárním jazyce dobře zformovaných kódů TS, test členství w v regulárním jazyce platných kódů vstupu TS s kódem x konkatenovaných s $\omega^*\#$, univerzální TS M_u a jeho aplikaci na w.
   - Pro TS M_x platí:
     – $\langle M_x\rangle \in L_{LOA} \Leftrightarrow x$ je platný kód TS a zároveň pro všechna možná w platí, že se hlava TS M_u simulujícího běh TS s kódem x na vstupu w dostane nad symbol $\#$ nebo M_u odmítne nebo cyklí, aniž by se jeho hlava dostala nad symbol $\#$. Tato varianta je zřejmě ekvivalentní tomu, že TS s kódem x při jakémkoli vstupu buď odmítne nebo cyklí.

  - $\langle M_x \rangle \notin L_{LOA} \Leftrightarrow x$ není platný kód TS nebo existuje nějaké w takové, že TS M_u simulující běh TS s kódem x na vstupu w přijme. Tato varianta je zřejmě ekvivalentní tomu, že TS s kódem x přijímá alespoň jedno slovo.

- Ukážeme, že δ zachovává členství v jazyce dle definice redukce:
  $\forall x \in \{0,1\}^* : x \in L_{EMP} \Leftrightarrow$ TS s kódem x nepřijímá žádné slovo $\Leftrightarrow \langle M_x \rangle \in L_{LOA}$
- Tedy L_{LOA} není ani rekurzivně vyčíslitelný.

3. Uvažujte následujíci dva důkazy diagonalizací. Oba jsou chybné (ani jedno z dokazovaných tvrzení neplatí). Vysvětlete, který krok je chybný a zdůvodněte proč. Může vám pomoci důkladně si prostudovat kapitolu o diagonalizaci ve studijní opoře. Nehledejte komplikovaná řešení, ke zdůvodnění by vám měla stačit jedna až dvě věty.

   (a) Důkaz, že množina $A \subseteq 2^{\{0\}^*}$ všech *konečných* jazyků nad abecedou $\{0\}$ má jinou mohutnost než množina $B \subseteq 2^{\{0,1\}^*}$ všech *konečných* jazyků nad abecedou $\{0, 1\}$.

      i. Předpokládejme, že A i B mají stejnou mohutnost. Pak existuje bijekce $f : A \to B$.
      ii. Prvky množiny A je možné očíslovat přirozenými čísly a seřadit do posloupnosti $L_1, L_2, L_3, ...$
      iii. Slova z množiny $\{0, 1\}^*$ je taktéž možno uspořádat do nějaké posloupnosti $w_1, w_2, w_3, ...$ (například lexikograficky).
      iv. f potom můžeme zobrazit nekonečnou maticí m, kde m_{ij} je 1, pokud $w_j \in f(L_i)$, a 0 jinak:

$$
\begin{array}{ccccc}
 & w_1 & w_2 & w_3 & ... \\
f(L_1) & 1 & 0 & 1 & ... \\
f(L_2) & 0 & 0 & 1 & ... \\
f(L_3) & 1 & 1 & 1 & ... \\
... & ... & ... & ... & ...
\end{array}
$$

      v. Uvažujme jazyk $L \subseteq \{0, 1\}^*$, který vznikne komplementací diagonály, tedy takový, že pro každé $i > 0, w_i \in L$, právě když $w_i \notin f(L_i)$ (tedy právě když $m_{ii} = 0$).
      vi. Jazyk L se zřejmě liší od každého jazyka $f(L_i), i > 0$ (alespoň slovem w_i).
      vii. Zároveň je jazyk L v B.
      viii. To ale znamená, že f není surjektivní, a tedy nemůže být bijekcí. Spor.

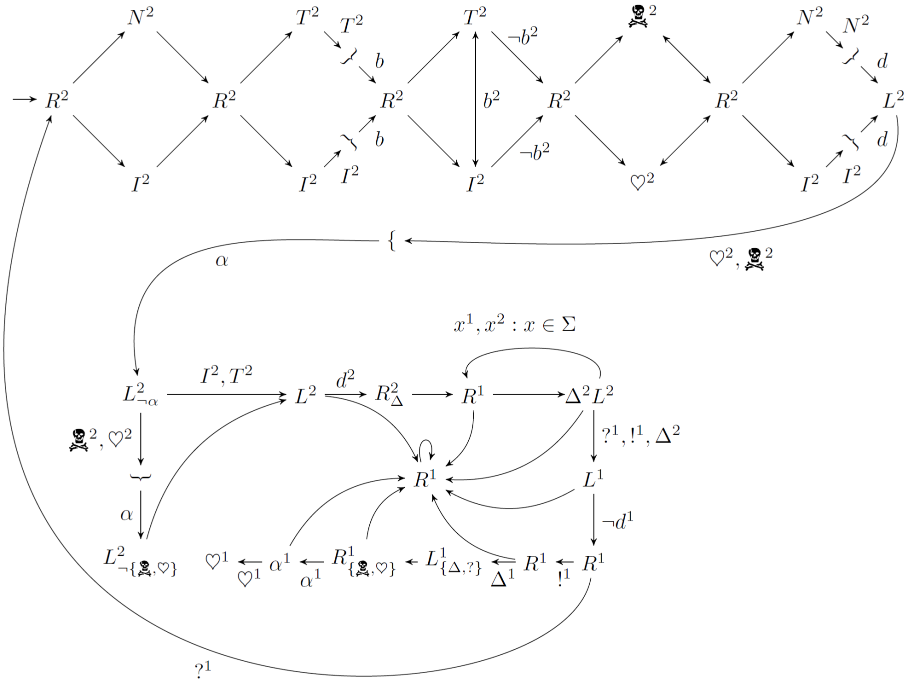
      Krok vii. je chybný! Jelikož je matice nekonečná, i L je nekonečný, tudíž nemůže patřit do množiny konečných jazyků B!

   (b) Stejné tvrzení a stejný důkaz jako v bodě 3a, pouze s tím rozdílem, že nyní $A = 2^{\{0\}^*}$ a $B = 2^{\{0,1\}^*}$, t.j., A i B teď obsahují všechny jazyky nad danými abecedami, včetně *nekonečných*.

      Krok ii. je chybný! A je nespočetná, viz. Lemma 6.1.1. ve studijní opoře, tudíž její prvky nelze očíslovat přirozenými čísly a seřadit do posloupnosti $L_1, L_2, L_3, ...$

4. Na Obrázku 1 je zobrazen dvoupáskový NTS *Happy End*, který přijímá jednoduchý regulární jazyk nad abecedou $\Sigma = \{N, T, I, \heartsuit, \text{☠}, ?, !\}$. Jaký?

   Demonstrujte běh *Happy End* na nějakém slově z L(*Happy End*) (stačí, když uvedete konfigurace druhé pásky na vstupu každého uzlu z obrázku 1, kterým běh stroje prochází).



Obrázek 1: NTS *Happy End*.

NTS *Happy End* přijímá regulární jazyk ekvivalentní regulárnímu výrazu: $(I(\heartsuit+\text{☠})^+TIN?)^*I\heartsuit^+TIN!$

Posloupnost konfigurací druhé pásky NTS *Happy End* na vstupu každé jeho komponenty, kterou prochází jeho běh, který přijme slovo $I\heartsuit TIN!$:

1. $[\underline{\Delta}\Delta\ldots]$
2. $[\Delta\underline{\Delta}\Delta\ldots]$
3. $[\Delta\underline{N}\Delta\ldots]$
4. $[\Delta N\underline{\Delta}\ldots]$
5. $[\Delta N\underline{I}\Delta\ldots]$
6. $[\Delta NI\underline{\Delta}\ldots]$
7. $[\Delta NI\underline{T}\Delta\ldots]$
8. $[\Delta NIT\underline{\Delta}\ldots]$
9. $[\Delta NIT\underline{\heartsuit}\Delta\ldots]$
10. $[\Delta NIT\heartsuit\underline{\Delta}\ldots]$
11. $[\Delta NIT\heartsuit\underline{I}\Delta\ldots]$
12. $[\Delta NIT\underline{\heartsuit}I\Delta\ldots]$
13. $[\Delta NI\underline{T}\heartsuit I\Delta\ldots]$
14. $[\Delta N\underline{I}T\heartsuit I\Delta\ldots]$
15. $[\Delta NIT\heartsuit I\underline{\Delta}\ldots]$
16. $[\Delta NIT\heartsuit I\underline{\Delta}\ldots]$
17. $[\Delta NIT\heartsuit I\underline{\Delta}\ldots]$
18. $[\Delta NIT\heartsuit\underline{I}\Delta\ldots]$
19. $[\Delta NIT\heartsuit\underline{I}\Delta\ldots]$
20. $[\Delta NIT\heartsuit\underline{\Delta}\Delta\ldots]$
21. $[\Delta NIT\underline{\heartsuit}\Delta\Delta\ldots]$
22. $[\Delta NIT\underline{\heartsuit}\Delta\Delta\ldots]$
23. $[\Delta NIT\underline{\Delta}\Delta\Delta\ldots]$
24. $[\Delta NI\underline{T}\Delta\Delta\Delta\ldots]$
25. $[\Delta NI\underline{T}\Delta\Delta\Delta\ldots]$
26. $[\Delta NI\underline{\Delta}\Delta\Delta\Delta\ldots]$
27. $[\Delta N\underline{I}\Delta\Delta\Delta\Delta\ldots]$
28. $[\Delta N\underline{I}\Delta\Delta\Delta\Delta\ldots]$
29. $[\Delta N\underline{\Delta}\Delta\Delta\Delta\Delta\ldots]$
30. $[\Delta\underline{N}\Delta\Delta\Delta\Delta\Delta\ldots]$
31. $[\Delta\underline{N}\Delta\Delta\Delta\Delta\Delta\ldots]$
32. $[\Delta\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\ldots]$
33. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
34. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
35. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
36. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
37. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
38. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$
39. $[\underline{\Delta}\Delta\Delta\Delta\Delta\Delta\Delta\ldots]$