Teoretická informatika (TIN) – 2024/2025
Úkol 2
(max. zisk 8 bodů – 10 bodů níže odpovídá 1 bodu v hodnocení předmětu)

1. Uvažujme abecedu Σ, t.ž., symbol $R \notin \Sigma$, a následující kódování deterministického konečného automatu do Turingova stroje: pro $A = (Q, \Sigma, \delta, q_0, F)$ sestrojíme TS $M_{sim}(A) = (Q \cup \{q_0^M, q_f^M\}, \Sigma, \Sigma \cup \{\Delta\}, \delta_M, q_0^M, q_f^M)$, kde $Q \cap \{q_0^M, q_f^M\} = \emptyset$ a δ_M je definována následovně:

   - $\delta_M(q_0^M, \Delta) = (q_0, R)$
   - $\forall f \in F : \delta_M(f, \Delta) = (q_f^M, R)$
   - $\delta_M(q, a) = (p, R) \Leftrightarrow \delta(q, a) = p$

   Množina kódů turingových strojů vzniklých transfomaci DKA $KA = \{\langle M\rangle \mid M = M_{sim}(A)$ pro nějaký DKA $A\}$ je rozhodnutelná, protože lze jednoduše ověřit tvar přechodové funkce.

   Rozhodněte a dokažte, zda následující jazyky jsou (resp. nejsou) rekurzivní (resp. rekursivně vyčíslitelné):

   - $L_1 = \{\langle M\rangle\#\langle w\rangle \mid w \notin L(M) \wedge \langle M\rangle \in KA\}$
   - $L_2 = \{\langle M\rangle\#\langle w\rangle \mid w \notin L(M) \wedge \langle M\rangle \notin KA\}$

   20 bodů

2. Jan a Eliška si vymysleli novou hru. Mají barevné křídy o b ($b \geq 2$) barvách. Na chodník si nakreslili křídou kolečka a některá z nich propojili čárami. Teď by rádi do každého kolečka namalovali x ($x \geq 2$) barevných značek (barvy značek v jednom kolečku se mohou opakovat) tak, aby kolečka propojená čárou nebyla označená stejně. Pořadí značek v kolečku nehraje roli. Otázka zní, jestli je takového označení možné.

   Formálně definujme hru Jana a Elišky jako n-tici $H = (K, C, b, x)$, kde

   - K je konečná množina koleček,
   - $C \subseteq \{\{a, b\} \mid a, b \in K\}$ je množina čar,
   - $b \geq 2$ je počet barvev,
   - $x \geq 2$ požadovaný počet značek.

   Hra H má řešení, pokud existuje zobrazení $O : K \times \langle 1, b\rangle \to \mathbb{N}$ takové, že

   - $\forall a \in K : \sum_{i=1}^{b} O(a, i) = x$ (počet značek v jednom kolečku je roven x)
   - $\forall \{a, b\} \in C\ \exists i : O(a, i) \neq O(b, i)$ (označení dvojice míst spojených čárou se liší)

   Dokažte, že problém existence řešení pro hru H je NP-úplný.

   (Pozn: Pomůže Vám NP-úplnost některého z problémů uvedených zde:
   `https://en.wikipedia.org/wiki/NP-completeness#NP-complete_problems`
   v odstavci „*NP-complete problems*“.)

   15 bodů

3. Uvažujeme funkci get_next, která má na vstupu řetězec nad abecedou $\Sigma = \{A, ..., Z\}$ a jeho délku l. Funkce vrácí následující řetězec vzhledem k lexikografickému uspořádání. Funkce $next\_char$ vrací následující znak latinské abecedy. Analyzujte a zdůvodněte amortizovanou časovou složitost libovolné posloupnosti n operací $str := get\_next(str, l)$. Na začátku je zafixována konstanta $l > 0$ a počáteční hodnota $str = A^l$ (řetězec obsahující l symbolů A).

   Předpokládejme uniformní cenové kriterium, kde každý řádek má cenu 1 (vyjímkou je řádek 7 s cenou 0).

```
1  Function get_next(char [] str, int l)
2      fin := false;
3      while ¬(fin) ∧ l > 0 do
4          l := l − 1;
5          if str[l] = Z then
6              str[l] := A;
7          else
8              next_char(str[l]);
9              fin := true;
10     return str;
```

   15 bodů

4. Uvažujme funkci *find_suffix*, která má na vstupu pole čísel $array$ o velikosti $size$ (chybné vstupy neuvažujte) a která se snaží nalést v rámci pole suffix takový, že součet čísel v tomto suffixu je roven hodnotě $final$.

   - Analyzujte časovou složitost funkce *find_suffix* v nejlepším případě
   - Analyzujte časovou složitost funkce *find_suffix* v nejhorším případě.
   - Navrhněte funkci *find_opt*, která bude dávat stejný výsledek jako funkce *find_suffix*, ale bude mít lepší asymptotickou složitost v nejhorším případě.
   - Analyzujte časovou složitost funkce *find_opt* v nejhorším případě.

   Předpokládejme uniformní cenové kriterium, kde každý řádek má cenu 1.

```
Function find_suffix(int * array, int size, int final)
    int i, j;
    i := 0;
    while i < size do
        j := i;
        int tmp := 0;
        while j < size do
            tmp := tmp + array[j];
            j := j + 1;
        if tmp = final then
            return ANO
        i := i + 1;
    return NE
```

15 bodů

5. Mějme teorii T se signaturou $\langle\{Kral_{/0}, Dama_{/0}, Vez_{/0}, Kun_{/0}, Pesec_{/0}\}, \{ohrozuje_{/2}, =_{/2}\}\rangle$ (= je standardní rovnost) se speciálními axiomy

$$\forall x(x = Kral \lor x = Dama \lor x = Vez \lor x = Kun \lor x = Pesec)$$
$$\forall x\ \neg ohrozuje(x, Kral)$$
$$\forall x, y(ohrozuje(x, y) \land ohrozuje(y, x) \Rightarrow x \neq Kun)$$
$$\forall x(ohrozuje(Pesec, x) \Rightarrow ohrozuje(x, Pesec) \lor x = Kun)$$

   i) Rozhodněte a stručně zdůvodněte, zda T je: a) bezesporná, b) úplná a c) rozhodnutelná (tj. množina důsledků T je rozhodnutelná).

15 bodů