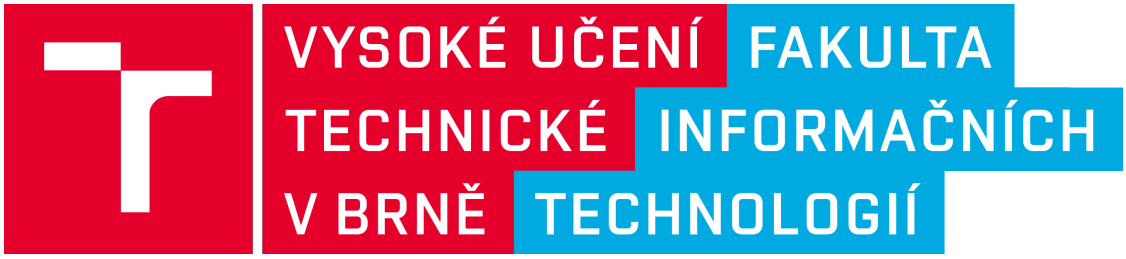

Teoretická informatika 2023/2024

Úkol 3

David Chocholatý (xchoch09)

Brno, 22. prosince 2023

1 Příklad 1

Sestavení zasedacího pořádku na Lenčině svatbě komplikují požadavky hostů na to, kdo s kým chce a nechce sedět. Dokažte, že Lenčin problém sestavení zasedacího pořádku (LP), definovaný následujícím způsobem, je **NP**-úplný: Mějme množinu hostů H, dvě ireflexivní binární relace *musí sedět s*, M, a *nechce sedět s*, N, na H, počet $S \in \mathbb{N}$ stolů, a velikostí $V \in \mathbb{N}$ stolů (počet míst u stolu). Je možné usadit všechny hosty k S stolům tak, aby u každého sedělo maximálně V hostů, aby každý seděl s těmi, s nimiž sedět musí, a zároveň neseděl s těmi, s nimiž sedět nechce?

(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“*.)

1.1 Řešení

Lemma 1.1. Nechť LP značí problém specifikovaný dle zadání. Platí, že $LP \in \mathbf{NP}$.

Důkaz. Nechť NTS značí *nedeterministický Turingův stroj*. Dále bude ukázáno, že $LP \in \mathbf{NP}$.

Bude sestrojen NTS M pracující v polynomiálním čase takový, že $L(M) = LP$. NTS M nedeterministicky zvolí rozesazení H hostů u nejvýše $S \in \mathbb{N}$ stolů na maximálně $V \in \mathbb{N}$ místech u stolu. Následně M vyhodnotí správnost rozesazení vůči ireflexivní binární relaci M a poté i N, přičemž v obou případech se jedná o lineární průchod seznamem (je uvažován průchod přes všechny dvojice relace jako přes uspořádaný seznam). Při uchování informací o počtu doposud obsazených stolů a míst u stolu, v průběhu zpracování seznamů, na dalších páskách M, je po projití uvedených seznamů možné jednoduchým porovnáním ověřit, zda byly dodrženy maximální příslušné počty.

Pokud se jedná o správné rozesazení hostů, tak M akceptuje. V opačném případě zamítá. Složitost M je tedy v $\mathcal{O}(n)$. Bylo tedy ukázáno, že pro LP existuje polynomiální verifikátor. □

Lemma 1.2. Nechť LP značí problém specifikovaný dle zadání. Problém LP je **NP**-těžký.

Důkaz. Důkaz bude proveden *polynomiální redukcí* ze známého grafového problému pojmenovaného jako *k-coloring*[1]. Notace daného problému bude dále uváděna jako k-$COLORING$, jenž je definován jako

$$k\text{-}COLORING = \{\langle G\rangle\#\langle k\rangle \mid \text{neorientovaný graf } G \text{ je obarvitelný nejvýše } k \text{ barvami}\},$$

kde graf $G = (V, E)$ má *chromatické číslo* $c \leq k$, pokud existuje funkce obarvení $f : V \to \{1, ..., c\}$ taková, že pro každé dva sousední uzly platí, že *nejsou* obarveny stejnou barvou, tj.

$$\forall\{v_1, v_2\} \in E : f(v_1) \neq f(v_2).$$

Bude ukázáno, že

$$k\text{-}COLORING \leq_P LP\,.$$

Zkonstruujeme polynomiálně vyčíslitelnou funkci f, pro kterou platí

[1] https://en.wikipedia.org/wiki/Graph_coloring#Vertex_coloring

$$A \in k\text{-}COLORING \iff f(A) \in LP\,.$$

Funkce f vytvoří z grafu $G = (V, E)$ a přirozeného čísla k instanci problému LP, a to takovou, že neorientovaný graf G je obarvitelný nejvýše k barvami tehdy a jen tehdy, když pro instanci problému LP existuje rozesazení takové, že u jednoho stolu spolu nebudou sedět ti hosté, kteří spolu *nechtějí sedět*, přičemž nejvýše mohou být daní hosté rozesazeni u $S \in \mathbb{N}$ stolů. *Vrcholy* grafu G z konečné množiny vrcholů V odpovídají *hostům* z množiny hostů H. *Hrany* grafu G z množiny hran E odpovídají jednotlivým dvojicím ireflexivní binární relace *nechce sedět s*, N, na H. Přirozené číslo k odpovídá počtu stolů S.

- $\Rightarrow$:

  Předpokládejme, že graf $G = (V, E)$ je obarvitelný nejvýše k barvami. Potom pro graf G existuje alespoň jedno obarvení vrcholů nejvýše k barvami takové, že žádné dva sousední vrcholy nejsou obarveny stejnou barvou. Zároveň existuje alespoň jedno správné rozesazení takové, že každé dva hosty, kteří spolu nechtějí sedět, je možné rozesadit k nejvýše S stolům tak, že u jednoho stolu spolu nebudou sedět hosté, kteří spolu nechtějí sedět.

- $\Leftarrow$:

  Předpokládejme, že hosté, kteří spolu nechtějí sedět, nesedí u stejného stolu, přičemž jsou rozesazeni nejvýše u S stolů. Jelikož každá dvojice hostů patřící do relace N, která spolu nechce sedět, sedí u jiného stolu (každý host z dané dvojice sedí u jiného stolu), musí být vrcholy odpovídající dvojice sousedních vrcholů grafu G obarveny rozdílnou barvou (přičemž jednotlivé barvy odpovídají příslušným stolům), a to nejvýše k barvami.

□

Teorém 1.3. Nechť LP značí problém specifikovaný dle zadání. Platí, že problém LP je **NP**-úplný.

Důkaz. Na základě lemmatu 1.1 bylo dokázáno, že $LP \in \mathbf{NP}$. Současně s lemmatem 1.2, které dokazuje, že problém LP je **NP**-těžký, vyplývá, že problém LP je **NP**-úplný. □

2 Příklad 2

Uvažujme sekvenci n operací vlož_a_uřež(k), $k \in \mathbb{N}$, provedenou nad oboustranně vázaném lineárním seznamem čísel, který je iniciálně prázdný.

Operace vlož_a_uřež(k) vloží nový prvek s hodnotou k do seznamu na takovou pozici, před kterou jsou pouze prvky s menší hodnotou než k, a za kterou nejsou prvky s menší hodnotou než k. Následně vypíše a smaže všechny prvky seznamu za vloženým prvkem.

Navrhněte, jak implementovat vlož_a_uřež tak, aby složitost sekvence n operací běžela v čase $\mathcal{O}(n)$. Dokažte, že v sekvenci operací je amortizovaná složitost jedné Vaší operace vlož_a_uřež konstantní.

(Pozn: Operaci vlož_a_uřež nepopisujte detailně na úrovni aktualizace ukazatelů, snažte se o jednoduše pochopitelný a jasný popis algoritmu s důrazem na hlavní myšlenku, bez rozebírání technikálií se zřejmým řešením. Předpokládejte, že máte funkce s konstantní časovou složitostí, které vrátí ukazatel na začátek/konec seznamu,

pro daný ukazatel na prvek seznamu vloží nový prvek s danou hodnotou před/za tento prvek, smažou a vypíší předchozí/následující prvek, posunou ukazatel na následující/předchozí prvek, atd. Porovnání dvou čísel považujte za operaci s konstantní časovou složitostí. Řešení je jednoduché, není třeba vymýšlet nic kompikovaného, používat další datové struktury apod., je třeba jen chytře využít invariant, který seznam splňuje.)

2.1 Řešení

Nejprve bude představen algoritmus zadané operace vlož_a_uřež(k), a to algoritmus 2.1. Navržený algoritmus využívá funkcí z následujícího výčtu, přičemž se přepokládá, že každá uvedená operace má konstantní časovou složitost:

- $get\_back()$

  Funkce vrátí ukazatel na konec oboustranně vázaného lineárního seznamu. Pokud je seznam prázdný, navrací se *nonce* (obdoba $NULL$ například z jazyka C).

- $get\_item()$

  Funkce získá hodnotu prvku seznamu, na který ukazuje daný ukazatel.

- $get\_prev()$

  Funkce vrací ukazatel na předchozí prvek daného ukazatele.

- $delete\_after()$

  Funkce provede výpis hodnoty prvku, na který ukazuje daný ukazatel, a následně je tento prvek ze seznamu odstraněn.

- $insert\_after(k)$

  Funkce vloží prvek za prvek seznamu, na který ukazuje daný ukazatel. Pokud je seznam prádný, je prvek do seznamu vložen standardním způsobem.

Dále se předpokládá, že porovnání dvou ukazatelů a dvou celých čísel má opět konstantní časovou složitost.

Algorithm 1 Operace vlož_a_uřež(k)

```
1: Input: Oboustranně vázaný lineární seznam list, prvek s hodnotou k, k ∈ ℕ
2: Output: nonce
3: procedure VLOŽ_A_UŘEŽ(k)
4:     pos ← list.get_back()
5:     while pos ≠ nonce and pos.get_item() ≥ k do
6:         pos.get_prev()
7:         pos.delete_after()
8:     pos.insert_after(k)      ▷ Při prázdném seznamu prvek vložen standardním způsobem
9:     return nonce                                  ▷ Operace nevrací žádnou hodnotu
```

Teorém 2.1. Nechť operace vlož_a_uřež je definovaná dle algoritmu 2.1. Potom časová složitost sekvence n operací je $\mathcal{O}(n)$ a amortizovaná složitost jedné operace vlož_a_uřež je *konstantní*.

Operace	Cena	Kredit
$list.get\_back()$	1	4
$pos \neq nonce$	1	0
$pos.get\_item()$	1	0
$\_item\_ \geq k$	1	0
$pos.get\_prev()$	1	0
$pos.delete\_after()$	1	0
$pos.insert\_after(k)$	1	6

Tabulka 1: Cena a kredity jednotlivých dílčích operací operace vlož_a_uřež.

Důkaz. Důkaz bude proveden *metodou účtů* (anglicky *accounting method*).

Na základě dané metody bude jednotlivým dílčím operacím operace vlož_a_uřež přiřazena jejich cena a kredit. Konkrétní rozdělení zachycuje tabulka 1.

Pro každou dílčí operaci se uvažuje konstantní časová složitost. Zároveň některé dílčí operace operace vlož_a_uřež nemají přiřazeny žádný kredit, respektive nulový kredit. To jsou takové operace, které jsou vykonávány při provádění jediného cyklu operace.

První dílčí operace, $get\_back()$, s nenulovým kreditem má přiřazen kredit s hodnotou 4, jelikož 1 kredit je nutný pro vykonání samotné operace a zbylé 3 kredity pro následné vyhodnocení podmínky cyklu, která bude vyhodnocena při každém volání operace vlož_a_uřež (ať již celá nebo jenom její první část).

Dále operace pro vložení prvku, $insert\_after(k)$, dostane kredit s hodnotou 6, a to z toho důvodu, že 1 kredit je nutný pro vložení prvku, další 2 kredity jsou předplaceny pro budoucí získání prvku, jeho výpis a odstranění pomocí dvou funkcí v těle cyklu a zbylé 3 kredity pro následné vyhodnocení podmínky v další iteraci cyklu po budoucím odstranění konkrétního prvku.

Platí, že složitost těchto operací ≤ 10 a tedy amortizovaná složitost jedné operace vlož_a_uřež je *konstantní*.

Vyplývá, že na základě uvedeného rozložení kreditů platí, že počet kreditů na účtě je vždy nezáporný. Celková cena n operací je $\leq 10n$ a tedy časová složitost sekvence n operací je $\mathcal{O}(n)$. □

3 Příklad 3

Formalizujte Tseytinovu transformaci a dokažte, že vrátí formuli lineární velikosti k velikosti vstupní formule. Konkrétněji:

- Předpokládejte na vstupu pouze formule generované gramatikou $\varphi \to X \mid \neg\varphi \mid (\varphi \vee \varphi)$, kde X je z množiny výrokových proměnných $\{A, B, C, ...\}$. Stejně jako na cvičení, velikost $|\varphi|$ formule φ je chápána jako délka slova nad abecedou $\{(,), \neg, \vee\} \cup X$.
- Pro formalizaci Tseytinovy transformace definujte rekurzivní funkci `Tse`(φ, i), která pracuje induktivně ke struktuře formule, a pro formuli φ a $i \in \mathbb{N}$ na vstupu vrátí konjunkci klauzulí definujících „pojmenování“ podformulí φ, kde samotná φ je pojmenována proměnnou X_i, a její vlastní podformule jsou pojmenovány proměnnými $X_{i+1}, X_{i+2}, ...$

Funkce `Tse` bude využívat funkci `CNF`, která převede formuli tvaru $X \leftrightarrow \neg Y$ nebo $X \leftrightarrow (Y \vee Z)$ do CNF klasickým způsobem.

- Lineární velikost výstupní formule dokažte indukcí ke struktuře vstupní formule.

3.1 Řešení

Definice 3.1. (Tseytinova transformace). Nechť formule φ je definována gramatikou $\varphi \rightarrow X \mid \neg\varphi \mid (\varphi \vee \varphi)$, kde X je z množiny výrokových proměnných $\{A, B, C, ...\}$. Dále nechť $X_i, X_{i+1}, X_{i+2}, ...$, kde $i \in \mathbb{N}$, značí nově definované proměnné podformulí φ a `CNF` je funkce, která převede formuli tvaru $X \leftrightarrow \neg Y$ nebo $X \leftrightarrow (Y \vee Z)$ do CNF klasickým způsobem. *Tseytinova transformace* je pak induktivně definována jako rekurzivní funkce následovně:

- $\texttt{Tse}(p, i) = p$,
- $\texttt{Tse}(\neg\varphi, i) = \begin{cases} \texttt{CNF}(X_i \leftrightarrow \neg\varphi), & \text{jestliže } \varphi \in X, \\ \texttt{CNF}(X_i \leftrightarrow \neg X_{i+1}) \wedge \texttt{Tse}(\varphi, i+1), & \text{jestliže } \varphi \notin X, \end{cases}$
- $\texttt{Tse}((\varphi_1 \vee \varphi_2), i) = \begin{cases} \texttt{CNF}(X_i \leftrightarrow (\varphi_1 \vee \varphi_2)), & \text{jestliže } \varphi_1, \varphi_2 \in X, \\ \texttt{CNF}(X_i \leftrightarrow (\varphi_1 \vee X_{i+1})) \wedge \texttt{Tse}(\varphi_2, i+1), & \text{jestliže } \varphi_1 \in X, \varphi_2 \notin X, \\ \texttt{CNF}(X_i \leftrightarrow (X_{i+1} \vee \varphi_2)) \wedge \texttt{Tse}(\varphi_1, i+1), & \text{jestliže } \varphi_1 \notin X, \varphi_2 \in X, \\ \texttt{CNF}(X_i \leftrightarrow (X_{i+1} \vee X_{n+1})) & \\ \quad \wedge \texttt{Tse}(\varphi_1, i+1) \wedge \texttt{Tse}(\varphi_2, n+1), & \text{jestliže } \varphi_1, \varphi_2 \notin X, \end{cases}$

pro $p \in X$ a $n = max_{j \in \mathbb{N}}(j)$, kde $X_{i+1}, X_{i+2}, ..., X_j, ..., X_n$ jsou nově definované proměnné všech podformulí φ_1 v $\texttt{Tse}(\varphi_1, i+1)$. Zároveň po dokončení provedení zpracování všech rekurzivních částí je na začátek výsledné formule φ_{res}, která je výsledkem funkce $\texttt{Tse}(\varphi_{in}, 1)$ s původní vstupní formulí φ_{in}, pomocí konjunkce konkatenována samotná proměnná X_1 (výsledkem je tedy $X_1 \wedge \varphi_{res}$) tehdy a jen tehdy, když $\varphi_{res} \notin X$ (výsledná formule není pouze literál). Dále nechť je Tseytinova transformace značena jako $\texttt{Tse}(\varphi, i)$.

Teorém 3.1. Tseytinova transformace $\texttt{Tse}(\varphi, i)$ formule φ s počátečním $i = 1$ roste lineárně vzhledem k $|\varphi|$. Tudíž platí, že $|\texttt{Tse}(\varphi, 1)| = \mathcal{O}(|\varphi|)$.

Důkaz. Potřebujeme ukázat, že

$$|\texttt{Tse}(\varphi, i)| \leq C\,|\varphi|$$

pro φ v dané podobě (velikost $|\varphi|$ formule φ je chápána jako délka slova nad abecedou $\{(,), \neg, \vee\} \cup X$) a pro kladnou konstantu $C \in \mathbb{Z}^+$. Důkaz bude proveden indukcí podle počtu symbolů z $\{(,), \neg, \vee\}$ vzhledem ke struktuře vstupní formule. Dále nechť je uvažováno $C = 12$.

Bázový případ. Pro daný případ, kde $p \in X$, platí pro $i \in \mathbb{N}$, že:

$$|\texttt{Tse}(p, i)| = |p| \leq 12\,|p|.$$

Bázový případ tedy platí.

Indukční předpoklad. Předpokládejme, že dané tvrzení platí pro všechny formule φ o maximálně $n = k$, $k \geq 0$, počtu symbolů z $\{(,), \neg, \vee\}$.

Indukční krok. Ukážeme, že tvrzení platí také pro formule φ s $n = k + 1$ počtem symbolů z $\{(,), \neg, \vee\}$. Protože $n = k + 1$, potom $n \geq 1$.

Nechť $p, p_1, p_2 \in X$ a φ_1, φ_2 jsou formule takové, že $\varphi_1, \varphi_2 \notin X$, přičemž samy obsahují alespoň jeden symbol z $\{(,), \neg, \vee\}$. Dále nechť $i \in \mathbb{N}$. Nastávají následující případy:

- $\texttt{Tse}(\neg p, i)$

$$\begin{aligned}\texttt{Tse}(\neg p, i) &\iff \texttt{CNF}(X_i \leftrightarrow \neg p) \\ &\iff (\neg X_i \vee \neg p) \wedge (p \vee X_i).\end{aligned}$$

  Potom platí, že

$$\begin{aligned}|\texttt{Tse}(\neg p, i)| &= |(\neg X_i \vee \neg p) \wedge (p \vee X_i)| \\ &= |(\neg X_i \vee \neg p)| + |(p \vee X_i)| = 12 \leq 12|\neg p| = 24.\end{aligned}$$

- $\texttt{Tse}(\neg \varphi, i)$

  Platí, že

$$\begin{aligned}\texttt{Tse}(\neg \varphi, i) &\iff \texttt{CNF}(X_i \leftrightarrow \neg X_{i+1}) \wedge \texttt{Tse}(\varphi, i + 1) \\ &\iff (\neg X_i \vee \neg X_{i+1}) \wedge (X_{i+1} \vee X_i) \wedge \texttt{Tse}(\varphi, i + 1).\end{aligned}$$

  Potom platí, že

$$\begin{aligned}|\texttt{Tse}(\neg \varphi, i)| &= |(\neg X_i \vee \neg X_{i+1}) \wedge (X_{i+1} \vee X_i) \wedge \texttt{Tse}(\varphi, i + 1)| \\ &= |(\neg X_i \vee \neg X_{i+1})| + |(X_{i+1} \vee X_i)| + |\texttt{Tse}(\varphi, i + 1)| \\ &= 12 + |\texttt{Tse}(\varphi, i + 1)|.\end{aligned}$$

  Dle indukčního předpokladu

$$\begin{aligned}|\texttt{Tse}(\neg \varphi, i)| &= 12 + |\texttt{Tse}(\varphi, i + 1)| \\ &= 12 + 12|\varphi| \leq 12(|\neg \varphi|) = 12(|\varphi| + 1).\end{aligned}$$

- $\texttt{Tse}((p_1 \vee p_2), i)$

$$\begin{aligned}\texttt{Tse}((p_1 \vee p_2), i) &\iff \texttt{CNF}(X_i \leftrightarrow (p_1 \vee p_2)) \\ &\iff (\neg X_i \vee p_1 \vee p_2) \wedge (\neg p_1 \vee X_i) \wedge (\neg p_2 \vee X_i).\end{aligned}$$

Potom platí, že

$$\begin{aligned}|\texttt{Tse}((p_1 \vee p_2), i)| &= |(\neg X_i \vee p_1 \vee p_2) \wedge (\neg p_1 \vee X_i) \wedge (\neg p_2 \vee X_i)| \\ &= |(\neg X_i \vee p_1 \vee p_2)| + |(\neg p_1 \vee X_i)| + |(\neg p_2 \vee X_i)| \\ &= 20 \leq 12|(p_1 \vee p_2)| = 60.\end{aligned}$$

- $\texttt{Tse}((p_1 \vee \varphi_2), i)$

Platí, že

$$\begin{aligned}\texttt{Tse}((p_1 \vee \varphi_2), i) \iff &\texttt{CNF}(X_i \leftrightarrow (p_1 \vee X_{n+1})) \wedge \texttt{Tse}(\varphi_2, n+1) \\ \iff &(\neg X_i \vee p_1 \vee X_{n+1}) \wedge (\neg p_1 \vee X_i) \wedge (\neg X_{n+1} \vee X_i) \wedge \texttt{Tse}(\varphi_2, n+1).\end{aligned}$$

Potom platí, že

$$\begin{aligned}|\texttt{Tse}((p_1 \vee \varphi_2), i)| =&|(\neg X_i \vee p_1 \vee X_{n+1}) \wedge (\neg p_1 \vee X_i) \wedge (\neg X_{n+1} \vee X_i) \wedge \texttt{Tse}(\varphi_2, n+1)| \\ =&|(\neg X_i \vee p_1 \vee X_{n+1})| + |(\neg p_1 \vee X_i)| + |(\neg X_{n+1} \vee X_i)| \\ &+ |\texttt{Tse}(\varphi_2, n+1)| \\ =&20 + |\texttt{Tse}(\varphi_2, n+1)|.\end{aligned}$$

Dle indukčního předpokladu

$$\begin{aligned}|\texttt{Tse}((p_1 \vee \varphi_2), i)| =&20 + |\texttt{Tse}(\varphi_2, n+1)| \\ =&20 + 12|\varphi_2| \leq 12|(p_1 \vee \varphi_2)| = 12(|\varphi_2| + 4).\end{aligned}$$

- $\texttt{Tse}((\varphi_1 \vee p_2), i)$

Platí, že

$$\begin{aligned}\texttt{Tse}((\varphi_1 \vee p_2) &\iff \texttt{CNF}(X_i \leftrightarrow (X_{i+1} \vee p_2)) \wedge \texttt{Tse}(\varphi_1, i+1)\\ &\iff (\neg X_i \vee X_{i+1} \vee p_2) \wedge (\neg X_{i+1} \vee X_i) \wedge (\neg p_2 \vee X_i) \wedge \texttt{Tse}(\varphi_1, i+1).\end{aligned}$$

Potom platí, že

$$\begin{aligned}|\texttt{Tse}((\varphi_1 \vee p_2), i)| &= |(\neg X_i \vee X_{i+1} \vee p_2) \wedge (\neg X_{i+1} \vee X_i) \wedge (\neg p_2 \vee X_i) \wedge \texttt{Tse}(\varphi_1, i+1)|\\ &= |(\neg X_i \vee X_{i+1} \vee p_2)| + |(\neg X_{i+1} \vee X_i)| + |(\neg p_2 \vee X_i)|\\ &\quad + |\texttt{Tse}(\varphi_1, i+1)|\\ &= 20 + |\texttt{Tse}(\varphi_1, i+1)|.\end{aligned}$$

Dle indukčního předpokladu

$$\begin{aligned}|\texttt{Tse}((\varphi_1 \vee p_2), i)| &= 20 + |\texttt{Tse}(\varphi_1, i+1)|\\ &= 20 + 12|\varphi_1| \leq 12|(\varphi_1 \vee p_2)| = 12(|\varphi_1| + 4).\end{aligned}$$

- $\texttt{Tse}((\varphi_1 \vee \varphi_2), i)$

  Platí, že

$$\begin{aligned}\texttt{Tse}((\varphi_1 \vee \varphi_2), i) &\iff \texttt{CNF}(X_i \leftrightarrow (X_{i+1} \vee X_{n+1})) \wedge \texttt{Tse}(\varphi_1, i+1) \wedge \texttt{Tse}(\varphi_2, n+1)\\ &\iff (\neg X_i \vee X_{i+1} \vee X_{n+1}) \wedge (\neg X_{i+1} \vee X_i) \wedge (\neg X_{n+1} \vee X_i)\\ &\quad \wedge \texttt{Tse}(\varphi_1, i+1) \wedge \texttt{Tse}(\varphi_2, n+1).\end{aligned}$$

  Potom platí, že

$$\begin{aligned}|\texttt{Tse}((\varphi_1 \vee \varphi_2), i)| &= |(\neg X_i \vee X_{i+1} \vee X_{n+1}) \wedge (\neg X_{i+1} \vee X_i) \wedge (\neg X_{n+1} \vee X_i)|\\ &\quad + | \wedge \texttt{Tse}(\varphi_1, i+1) \wedge \texttt{Tse}(\varphi_2, n+1)|\\ &= |(\neg X_i \vee X_{i+1} \vee X_{n+1})| + |(\neg X_{i+1} \vee X_i)| + |(\neg X_{n+1} \vee X_i)|\\ &\quad + |\texttt{Tse}(\varphi_1, i+1)| + |\texttt{Tse}(\varphi_2, n+1)|\\ &= 20 + |\texttt{Tse}(\varphi_1, i+1)| + |\texttt{Tse}(\varphi_2, n+1)|.\end{aligned}$$

  Dle indukčního předpokladu

$$\begin{aligned}|\mathtt{Tse}((\varphi_1 \vee \varphi_2), i)| =& 20 + |\mathtt{Tse}(\varphi_1, i+1)| + |\mathtt{Tse}(\varphi_2, n+1)| \\ =& 20 + 12|\varphi_1| + 12|\varphi_2| \leq 12|(\varphi_1 \vee \varphi_2)| = 12(|\varphi_1| + |\varphi_2| + 3).\end{aligned}$$

Teorém platí také pro $k+1$. Tudíž teorém platí. □

4 Příklad 4

Mějme prvořádovou teorii T_{C} s jazykem, jehož signatura obsahuje binární funkční symbol C. T_{C} definuje význam C jako funkci $\mathbb{N}$ takovou, že pro všechna $i, j \in \mathbb{N}$,

$$\mathsf{C}(i,j) = \mathsf{to\_num}(\mathsf{to\_num}^{-1}(i).\mathsf{to\_num}^{-1}(j)),$$

kde '.' je konkatenace řetězců a $\mathsf{to\_num}$ je bijektivní funkce z řetězců nad abecedou jazyka T_{C} do $\mathbb{N}$. Můžete předpokládat, že jazyk T_{C} má číslovky, tedy každé číslo $n \in \mathbb{N}$ je vyjádřeno nějakým termem jazyka.

Aplikací Tarského věty o nevyjádřitelnosti množiny pravdivých výroků dokažte, že pro Vámi zvolené Gödelovo kódování G, množina Gödelových čísel důsledků T_{C}, tedy

$$\{G(\varphi) \mid \varphi \text{ je věta jazyka } T_{\mathsf{C}} \text{ a } T_{\mathsf{C}} \models \varphi\},$$

nemůže být vyjádřitelná v teorii T_{C}.

(Pozn: Je třeba si jen uvědomit, co je co. Instanciovat správně abstraktní formální systém, zvolit Gödelovo kódování, a ukázat, že platí předpoklady Tarského věty, je potom jednoduché.)

4.1 Řešení

Teorém 4.1. Nechť T_{C} značí zadanou prvořádovou teorii. Potom množina Gödelových čísel důsledků T_{C},

$$\{G(\varphi) \mid \varphi \text{ je věta jazyka } T_{\mathsf{C}} \text{ a } T_{\mathsf{C}} \models \varphi\},$$

nemůže být vyjádřitelná v teorii T_{C}.

Důkaz. Pro aplikaci důkazu pomocí *Tarského* věty bude nejprve nutné ověřit, zda teorie T_{C} splňuje následující dva body:

- *G1*: A^* je vyjádřitelná pro každou vyjádřitelnou A,
- *G2*: $\tilde{A}$ je vyjádřitelná pro každou vyjádřitelnou A,

kde $A = \{n \in \mathbb{N} \mid H(n) \text{ je věta jazyka } T_{\mathsf{C}} \text{ a } T_{\mathsf{C}} \models H(n)\}$, přičemž $H(n)$ značí větu, kde $H \in \mathcal{H}$ a $n \in \mathbb{N}$ pro množinu predikátů abstraktního formálního systému, $\mathcal{H}$.

Splnění *G2* je snadné, jelikož v predikátové logice lze aplikovat negaci. Následně bude ověřeno splnění *G1*. Zvolme Gödelovo kódování dle tabulky 2.

symbol	$\forall$	$\neg$	$\to$	(	)	v	$'$	$=$	$\leq$	#	0	1	2	3	4	5	6	7	8	9
kód	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Tabulka 2: Tabulka Gödelova kódování pro prvořádovou teorii T_{C}, kde $v', v'', v''', ...$ značí jména proměnných a symbol # slouží jako oddělovač formulí v důkazech.

Slovo $w = a_0...a_n \in \mathbb{N}^*$ je zápisem Gödelova čísla

$$G(w) = \mathsf{C}(a_0, \mathsf{C}(a_1, \mathsf{C}(..., \mathsf{C}(a_{n-1}, a_n)))).$$

Dále bude ukázáno, že pro vyjádřitelnou $A \in \mathbb{N}$ je $A^* = \{n \in \mathbb{N} \mid G(E_n(n)) \in A\}$ také vyjádřitelná.

Konkatenace je vyjádřitelná, jelikož $G(u.v) = \mathsf{C}(u, v)$, značeno $G(u) \circ G(v)$. Pro číslo $\overline{n}$ je kód n získán jako $G(\overline{n}) = \mathsf{C}(1, \overline{n})$. $G(E_e(n))$ je vyjádřitelné na základě e a n. Pokud E_e má volnou proměnnou v, $E_e(n)$ je ekvivalentní $\forall v(v = \overline{n} \to E_e)$. $G(E_n(n))$ lze tedy vyjádřit jako

$$G(E_e(n)) = k \circ \mathsf{C}(1, \overline{n}) \circ 2 \circ e \circ 4,$$

kde $k \in \mathbb{N}$, přičemž zastupuje část „$\forall v(v =$“.

Pro predikát F vyjadřující A, pak A^* je vyjádřitelná predikátem F^* jako

$$F^*(n) = \exists v(v = k \circ \mathsf{C}(1, \overline{n}) \circ 2 \circ e \circ 4 \wedge F(v)),$$

přičemž platí, že

$$F^*(n) \iff G(F_n(n)) \in A \iff \exists v(v = G(E_n(n)) \wedge F(v)).$$

Následně již pro dokončení důkazu stačí pouze aplikovat Tarského větu.

Nechť množina $\{G(\varphi) \mid \varphi$ je věta jazyka T_{C} a $T_{\mathsf{C}} \models \varphi\}$, je vyjádřitelná, přičemž dále bude značena jako T. Z *G2* je vyjádřitelná i množina $\tilde{T}$ a z *G1* také $\tilde{T}^*$. Nechť Z vyjadřuje $\tilde{T}^*$ a nechť $G(Z) = z$, tedy $Z = E_Z$. Následně platí, že

$$\begin{aligned} Z(z) \text{ je věta jazyka } T_{\mathsf{C}} \text{ a } T_{\mathsf{C}} \models Z(z) &\overset{\text{def. } Z}{\iff} z \in \tilde{T}^* \\ &\overset{\text{def. } *}{\iff} G(Z(z)) \in \tilde{T} \\ &\overset{\text{def. } \tilde{}}{\iff} G(Z(z)) \notin T \\ &\overset{\text{def. } T}{\iff} Z(z) \text{ není věta jazyka } T_{\mathsf{C}} \text{ nebo } T_{\mathsf{C}} \not\models Z(z). \end{aligned}$$

Tudíž dochází ke sporu a důkaz je uzavřen. □