Redukcie

Peter Kostolányi

19. marca 2024

Doposiaľ sme s pojmom redukcie pracovali v čisto intuitívnej rovine a pod redukciou rozhodovacieho problému P na rozhodovací problém R sme v princípe rozumeli dôkaz, že rozhodnuteľnosť – prípadne rekurzívna vyčísliteľnosť – problému R implikuje rozhodnuteľnosť – alebo rekurzívnu vyčísliteľnosť – problému P. Táto predstava je skutočne postačujúca, ak je jediným cieľom dokázať nerozhodnuteľnosť problému P. Akonáhle sa však stredobodom pozornosti stanú samotné nerozhodnuteľné problémy a ich jemnejšia klasifikácia, vzniká potreba definovať pojem redukcie o niečo opatrnejšie.

Nedostatkom uvedenej naivnej predstavy o redukciách je predovšetkým jej benevolentnosť pokiaľ ide o vlastnosti, ktoré od redukcií vyžadujeme. Uvažujme napríklad ľubovoľný nerozhodnuteľný problém R. Pri ponímaní opísanom vyššie by potom nutne existovala redukcia ľubovoľného rozhodovacieho problému P na problém R, keďže v implikácii „ak R je rozhodnuteľný, tak aj P je rozhodnuteľný“ je ľavá strana nepravdivá, a teda implikácia ako celok je pravdivá. Tento poznatok však neposkytuje vôbec žiaden návod, ako s využitím existencie Turingovho stroja rozhodujúceho problém R skonštruovať Turingov stroj rozhodujúci problém P. Ukazuje sa dokonca, že existujú aj také problémy P, pre ktoré je takáto konštrukcia principiálne nemožná.[1]

Tento nedostatok je pomerne vážny. Pri všetkých redukciách rozhodovacích problémov, s ktorými sme sa doposiaľ stretli, sme totiž kládli dôraz na skutočnosť, že v prípade rozhodnuteľnosti problému R vieme *skonštruovať* stroj rozhodujúci problém P, ktorý môže využívať (hypotetický) stroj pre problém R ako „podprogram“. V nasledujúcom preto zavedieme pojem tzv. *turingovskej redukcie* zachycujúci práve túto vlastnosť danej dvojice problémov. Ak budeme neskôr hovoriť o „redukcii“ bez ďalšieho prívlastku, vždy budeme mať na mysli turingovskú redukciu.

Takto definovaný pojem turingovskej redukcie nám tiež umožní *klasifikovať* nerozhodnuteľné problémy podľa ich obtiažnosti. Pre turingovskú redukciu už totiž neplatí, že na každý nerozhodnuteľný problém R možno redukovať všetky rozhodovacie problémy P. Práve naopak – ak možno rozhodovací problém P redukovať na rozhodovací problém R, je to známkou toho, že problém P je *najviac taký ťažký* ako problém R.

Neskôr na niekoľkých riešených úlohách preskúmame vzájomnú (turingovskú) redukovateľnosť niektorých variantov Postovho korešpondenčného problému. V samom závere týchto poznámok zavedieme dva ďalšie druhy redukcií prísnejších, než turingovská redukcia. Na tomto predmete s nimi síce ďalej pracovať nebudeme, ale v teórii vypočítateľnosti sú užitočným nástrojom na klasifikáciu neriešiteľných problémov (jemnejšiu, než je možné dosiahnuť s použitím turingovských redukcií).

1 Turingovská redukcia a turingovská ekvivalencia

Pojem *turingovskej redukcie* problému P na problém R je formalizáciou vyššie opísanej intuitívnej predstavy o tom, že v prípade možnosti využívať „podprogram“ rozhodujúci problém R – kde tento predpoklad je, samozrejme, čisto hypotetický – by bolo možné rozhodovať aj problém P. Na formálnu definíciu turingovskej redukcie síce budeme potrebovať pojem deterministického Turingovho stroja s orákulom, v ňom však čitateľ istotne bez väčších problémov nájde paralelu s práve vysvetleným intuitívnym ponímaním. Pri aplikáciách pojmu turingovskej redukcie už budeme pracovať výlučne v neformálnej rovine.

Nech $L \subseteq \Sigma^*$ je ľubovoľný jazyk. *Deterministický Turingov stroj s orákulom* L je deterministický Turingov stroj so „zázračnou“ znalosťou príslušnosti slov do jazyka L. Ide teda o deterministický Turingov stroj, ktorý má navyše k dispozícii jednu špeciálnu dopytovaciu pásku (s ktorou pracuje ako s bežnou pracovnou páskou) a tri špeciálne stavy $q_?$, q_{yes} a q_{no}. Ak sa stroj dostane do stavu $q_?$, pričom obsahom dopytovacej pásky je v takejto konfigurácii slovo $x \in \Sigma^*$, v ďalšom kroku stroj

[1] Formalizácia tohto tvrdenia využíva pojem turingovskej redukcie definovaný nižšie. Jeho dôkaz ale presahuje rámec tohto predmetu.

„automaticky“ prejde do stavu q_{yes} (ak $x \in L$) resp. q_{no} (ak $x \notin L$). Obsah dopytovacej pásky sa v takomto kroku môže zmazať, prípadne môže ostať nezmenený (ide o technický detail, ktorý nijako nemení silu výsledného modelu). Formálnu definíciu deterministického Turingovho stroja s orákulom ako usporiadanej k-tice, ako aj definíciu konfigurácie, kroku výpočtu a akceptovaného jazyka, prenechávame čitateľovi ako jednoduché (avšak silno odporúčané) cvičenie.

Takto definovaný Turingov stroj s orákulom umožňuje zaviesť pojem turingovskej redukcie tak, ako bol neformálne opísaný vyššie.

Definícia 1. Nech $P, R \subseteq \Sigma^*$ sú rozhodovacie problémy reprezentované ako jazyky nad abecedou Σ. Hovoríme, že problém P je *turingovsky redukovateľný* na problém R, ak existuje na každom vstupe zastavujúci deterministický Turingov stroj A s orákulom R taký, že $L(A) = P$. V takom prípade píšeme $P \leq_T R$.

Zápis $P \leq_T R$ možno čítať aj ako „problém P je ľahší alebo rovnako ťažký ako problém R“ (vzhľadom na turingovskú redukciu).

Veta 1. *Nech P, R sú rozhodovacie problémy také, že $P \leq_T R$. Ak je problém R rozhodnuteľný, tak je rozhodnuteľný aj problém P.*

Dôkaz. Ľahko možno nahliadnuť, že v prípade rozhodnuteľnosti problému R možno každý deterministický Turingov stroj s orákulom R prerobiť na ekvivalentný bežný deterministický Turingov stroj bez orákula. Stačí každé „volanie“ orákula nahradiť „volaním“ deterministického Turingovho stroja rozhodujúceho R. □

Pojem turingovskej redukcie možno okrem iného využiť na klasifikáciu nerozhodnuteľných problémov podľa obtiažnosti. Základom na vybudovanie takejto teórie[2] je zavedenie pojmu turingovskej ekvivalencie.

Definícia 2. Nech $P, R \subseteq \Sigma^*$ sú rozhodovacie problémy reprezentované ako jazyky nad abecedou Σ. Hovoríme, že problém P je (vzhľadom na turingovskú redukciu) *rovnako ťažký* ako problém R, ak $P \leq_T R$ a zároveň $R \leq_T P$. V takom prípade píšeme $P \equiv_T R$ a hovoríme tiež, že problém P je *turingovsky ekvivalentný* problému R.

V nasledujúcom zhrnieme niekoľko elementárnych vlastností turingovskej redukcie:

- Ak $L_1, L_2 \in \mathscr{L}_{rec}$, tak $L_1 \equiv_T L_2$. Ak totiž $L_1 \in \mathscr{L}_{rec}$, existuje deterministický Turingov stroj A_1, ktorý sa na každom vstupe zastaví a pre ktorý je $L(A_1) = L_1$. Ľahko vidieť, že stroj A_1 možno prerobiť na ekvivalentný deterministický Turingov stroj s orákulom L_2 – ten bude jednoducho simulovať stroj A_1 a orákulum nikdy nezavolá. Preto $L_1 \leq_T L_2$ a na dôkaz $L_2 \leq_T L_1$ možno použiť symetrickú argumentáciu.

- Ak $L_1 \in \mathscr{L}_{rec}$ a $L_2 \notin \mathscr{L}_{rec}$, tak vždy $L_1 \leq_T L_2$, ale nikdy nemôže byť $L_2 \leq_T L_1$. Dôkaz prvého tvrdenia je rovnaký ako v predchádzajúcom prípade. Na dôkaz druhého tvrdenia si stačí uvedomiť, že keby bolo $L_2 \leq_T L_1$, musel by byť problém L_2 podľa vety 1 tiež rozhodnuteľný.

- Ak $L_1, L_2 \in \mathscr{L}_{RE} - \mathscr{L}_{rec}$, tak vo všeobecnosti *nemusí byť* $L_1 \equiv_T L_2$. Dá sa dokonca dokázať, že existuje dvojica jazykov $L_1, L_2 \in \mathscr{L}_{RE} - \mathscr{L}_{rec}$, pre ktoré nie je ani $L_1 \leq_T L_2$, ani $L_2 \leq_T L_1$. Tento výsledok je známy aj ako Friedbergova-Mučnikova veta a jeho dôkaz výrazne presahuje rámec tohto predmetu. Napriek tomu je dobré mať na zreteli skutočnosť, že rekurzívna vyčísliteľnosť dvoch nerozhodnuteľných problémov ešte neimplikuje, že obidva problémy sú (vzhľadom na turingovskú redukciu) rovnako ťažké.

- Pre všetky jazyky L je $L \equiv_T L^C$. Na rozhodovanie problému L totiž stačí pre každý vstup w „zavolať“ orákulum L^C a zistiť tak, či $w \in L^C$ – potom už iba stačí výstup orákula znegovať. N. B.: ak $L \in \mathscr{L}_{RE} - \mathscr{L}_{rec}$, tak $L^C \notin \mathscr{L}_{RE}$; napriek tomu sú ale tieto dva problémy (vzhľadom na turingovskú redukciu) rovnako ťažké.

[2] Ktorá však z väčšej časti presahuje rámec tohto predmetu.

- Relácia $\leq_T$ je očividne reflexívna a tranzitívna, no nie je antisymetrická – preto *nie je čiastočným usporiadaním*, ale *je predusporiadaním.*

- Relácia $\equiv_T$ *je reláciou ekvivalencie* na triede všetkých rozhodovacích problémov (jazykov). Triedy ekvivalencie relácie $\equiv_T$ sa nazývajú *Turingove triedy.*

- Reláciu $\leq_T$ možno prirodzene rozšíriť na Turingove triedy. V takom prípade už *je čiastočným usporiadaním.*

Poznámka 1. Aj keď je koncept Turingovho stroja s orákulom nutný na formálnu definíciu turingovskej redukcie, v nasledujúcom ho nahradíme intuitívnejším konceptom algoritmov, ktoré majú možnosť „zavolať podprogram“ zodpovedajúci orákulu. V prípade, že prijmeme Turingovu tézu, sú oba tieto koncepty navzájom ekvivalentné. To znamená, že pre dva rozhodovacie problémy P, R je $P \leq_T R$ práve vtedy, keď existuje algoritmus rozhodujúci problém P za predpokladu, že má možnosť zavolať ako „podprogram“ (hypotetický) algoritmus rozhodujúci problém R. To súhlasí s intuitívnou predstavou o redukciách, ako sme ich na tomto predmete používali doposiaľ.

Príklad 1. Dokážeme, že modifikovaný Postov korešpondenčný problém je (vzhľadom na turingovskú redukciu) rovnako ťažký ako Postov korešpondenčný problém. Skutočnosť MPKP $\leq_T$ PKP sme už v podstate dokázali v poznámkach k Postovmu korešpondenčnému problému – stačí nahliadnuť, že redukcia, ktorú sme tam použili, je turingovská. To je však pomerne zrejmé, keďže stačí algoritmicky transformovať prípad MPKP na k nemu prislúchajúci prípad PKP a naň jedenkrát zavolať „podprogram“ (resp. orákulum) pre PKP.

Aby sme dokázali PKP $\leq_T$ MPKP, musíme opísať algoritmus, ktorý rozhoduje PKP za predpokladu, že má možnosť pristupovať k „podprogramu“ rozhodujúcemu MPKP. Ten ale môže pracovať tak, že postupne zavolá MPKP pre všetky možné voľby prvej dlaždice v danom prípade PKP. Ak je pre aspoň jeden z týchto prípadov MPKP výstupom „áno“, má daný prípad PKP riešenie; v opačnom prípade zjavne riešenie nemá.

2 Riešené úlohy

V nasledujúcom vyriešime niekoľko ukážkových úloh, v ktorých je cieľom zistiť, či sú dané varianty Postovho korešpondenčného problému (vzhľadom na turingovskú redukciu) rovnako ťažké ako bežný PKP. Buď teda treba dokázať, že pre daný variant PKP (označme ho PKP$'$) je PKP$' \leq_T$ PKP a zároveň PKP $\leq_T$ PKP$'$, alebo treba ukázať, že niektorým smerom redukciu urobiť nemožno. Jedinou metódou, ktorú pritom máme k dispozícii na dôkaz *nemožnosti* turingovskej redukovateľnosti je pozorovanie, že jeden z problémov je rozhodnuteľný, kým druhý nie je. Treba však mať na pamäti, že existujú aj iné dvojice problémov, ktoré nie sú rovnako ťažké – hoci dôkaz tejto skutočnosti by vyžadoval použitie pomerne pokročilých techník.

V riešeniach nasledujúcich úloh nebudeme venovať zvláštnu pozornosť skutočnosti, že použité redukcie sú turingovské. Namiesto toho sa vrátime k intuitívnemu ponímaniu redukcií, pričom ich „turingovskosť“ bude zväčša očividná.

Úloha 1. Uvažujme rozhodovací problém PKP$'$, kde na vstupe je prípad PKP a treba rozhodnúť, či preň existuje riešenie dĺžky aspoň dva. Zistite, či je tento problém (vzhľadom na turingovskú redukciu) rovnako ťažký ako PKP a svoje tvrdenie dokážte.

Riešenie. Problémy *sú rovnako ťažké* – je teda PKP$' \equiv_T$ PKP. Obidve redukcie sú dokonca triviálne, keďže prípad PKP má riešenie práve vtedy, keď má riešenie dĺžky dva – pre ľubovoľné riešenie $x \in \Delta^+$ je totiž riešením aj x^2; ak pritom x pozostáva z jedinej dlaždice, pozostáva x^2 z dvoch dlaždíc. To znamená, že na rozhodnutie prípadu PKP$'$ stačí na nezmenenom vstupe zavolať (hypotetický) algoritmus rozhodujúci PKP a na rozhodnutie prípadu PKP stačí na nezmenenom vstupe zavolať (hypotetický) algoritmus rozhodujúci PKP$'$. □

Úloha 2. Uvažujme rozhodovací problém PKP$'$, kde na vstupe je prípad PKP s jednou označenou dlaždicou a treba rozhodnúť, či má daný prípad PKP riešenie dĺžky aspoň dva také, že predposledná dlaždica v tomto riešení je označená. Zistite, či je tento problém (vzhľadom na turingovskú redukciu) rovnako ťažký ako PKP a svoje tvrdenie dokážte.

Riešenie. Problémy *sú rovnako ťažké* – je teda PKP$' \equiv_T$ PKP. Je vcelku očividné, že keby sme vedeli rozhodovať PKP$'$, vedeli by sme rozhodovať aj PKP. Stačilo by totiž pre daný prípad PKP vyskúšať všetky možné voľby označenej dlaždice a pre každú zavolať algoritmus rozhodujúci PKP$'$. Ak je pre nejakú voľbu označenej dlaždice výstupom „áno“, riešenie daného prípadu PKP$'$ je súčasne aj riešením pôvodného prípadu PKP. Ak má naopak prípad PKP riešenie dĺžky aspoň dva, stačí označiť dlaždicu, ktorá je v ňom predposledná a výsledný prípad PKP$'$ má riešenie. Stačí sa teda odvolať na pozorovanie z riešenia predchádzajúcej úlohy: ak má prípad PKP riešenie, tak má aj riešenie dĺžky aspoň dva. Dokázali sme teda, že PKP $\leq_T$ PKP$'$.

Zostáva dokázať, že keby sme vedeli rozhodovať PKP, vedeli by sme rozhodovať aj PKP$'$. Majme daný prípad PKP$'$ nad abecedou Σ takou, že $\Sigma \cap \{\#, \$\} = \emptyset$. Ten upravíme na prípad PKP pomocou „omriežkovania“, podobne ako pri štandardnej redukcii MPKP na PKP. Výsledná sada bude obsahovať jednu počiatočnú dlaždicu

$$\begin{array}{|c|}\hline \#\# \\ \hline \# \\ \hline\end{array},$$

pričom neskôr bude zrejmé, že každé riešenie konštruovaného prípadu sa bude musieť začínať touto dlaždicou. Pre každú dlaždicu

$$\begin{array}{|c|}\hline u \\ \hline v \\ \hline\end{array}$$

pôvodného prípadu PKP$'$ bude ďalej konštruovaný prípad PKP obsahovať dlaždicu

$$\begin{array}{|c|}\hline \overline{u}\# \\ \hline \#\overline{v} \\ \hline\end{array},$$

kde $\overline{a_1 a_2 \ldots a_n} = a_1 \# a_2 \# \ldots \# a_n$. Ak je nakoniec

$$\begin{array}{|c|}\hline u_{ozn} \\ \hline v_{ozn} \\ \hline\end{array}$$

označená dlaždica v prípade PKP$'$, tak pre každú dlaždicu

$$\begin{array}{|c|}\hline u \\ \hline v \\ \hline\end{array}$$

tohto prípadu PKP$'$ bude konštruovaný prípad PKP obsahovať „zdvojenú“ dlaždicu

$$\begin{array}{|c|}\hline \overline{u_{ozn}}\,\overline{u}\$ \\ \hline \#\overline{v_{ozn}}\,\overline{v}\$ \\ \hline\end{array}.$$

Každé riešenie takto skonštruovaného prípadu PKP sa zjavne musí začínať počiatočnou dlaždicou a končiť „zdvojenou“ dlaždicou – v minimálnom riešení sa navyše takéto „špeciálne“ dlaždice nikde inde nevyskytujú. „Zdvojená“ dlaždica na konci riešenia zjavne zabezpečí, že v zodpovedajúcom riešení PKP$'$ je na predposlednej pozícii označená dlaždica. Skutočne teda PKP$' \leq_T$ PKP. ☐

Úloha 3. Uvažujme rozhodovací problém PKP$'$, kde na vstupe je prípad PKP *pozostávajúci z jedinej dlaždice* a treba rozhodnúť, či má daný prípad PKP riešenie. Zistite, či je tento problém rovnako ťažký ako PKP (vzhľadom na turingovskú redukciu) a svoje tvrdenie dokážte.

Riešenie. Problémy *nie sú rovnako ťažké.* Problém PKP$'$ je totiž zjavne rozhodnuteľný (jeho prípad má riešenie práve vtedy, keď daná dlaždica obsahuje na oboch „poschodiach“ rovnaké slovo), a teda pre nerozhodnuteľný PKP nemôže byť PKP $\leq_T$ PKP$'$. ☐

3 Ďalšie druhy redukcií

V teórii vypočítateľnosti sa okrem turingovskej redukcie študujú aj ďalšie, prísnejšie druhy redukcií. Dôležitým príkladom takýchto redukcií sú tzv. *„many-one“ redukcie*. Podobne ako pri turingovskej redukcii je problém P „many-one“-redukovateľný na problém R, ak za predpokladu možnosti pristupovať k „podprogramu“ rozhodujúcemu problém R existuje algoritmus rozhodujúci problém P. Tento „podprogram“ však možno „zavolať“ iba raz a jeho výstup musí byť aj výstupom algoritmu rozhodujúceho P (nemožno teda napríklad vymeniť výstupy „áno“ a „nie“).

Definícia 3. Nech $P, R \subseteq \Sigma^*$ sú rozhodovacie problémy reprezentované ako jazyky nad abecedou Σ. Hovoríme, že problém P je *„many-one“-redukovateľný* na R, ak existuje vypočítateľná funkcia[3] $f\colon \Sigma^* \to \Sigma^*$ taká, že pre všetky $w \in \Sigma^*$ je $w \in P$ práve vtedy, keď $f(w) \in R$. V takom prípade píšeme $P \leq_m R$.

Napríklad štandardná redukcia MPKP na PKP je „many-one“ redukciou – funkcia f v tomto prípade realizuje štandardnú transformáciu prípadu MPKP na k nemu prislúchajúci prípad PKP. Naopak redukcia PKP na MPKP z príkladu 1 nie je „many-one“ redukciou, keďže k „podprogramu“ pre MPKP sa v nej pristupuje niekoľkokrát.

Pomenovanie „many-one“ redukcia vychádza zo skutočnosti, že funkcia f v jej definícii môže zobraziť aj viacero vstupov problému P na jeden rovnaký vstup problému R. V prípade, že je táto funkcia f injektívna, hovoríme o *„one-one“ redukcii*.

Definícia 4. Nech $P, R \subseteq \Sigma^*$ sú rozhodovacie problémy reprezentované ako jazyky nad abecedou Σ. Hovoríme, že problém P je *„one-one“-redukovateľný* na R, ak existuje vypočítateľná injektívna funkcia $f\colon \Sigma^* \to \Sigma^*$ taká, že pre všetky $w \in \Sigma^*$ je $w \in P$ práve vtedy, keď $f(w) \in R$. V takom prípade píšeme $P \leq_1 R$.

Je zrejmé, že ak pre dvojicu rozhodovacích problémov P, R je $P \leq_1 R$, tak nutne aj $P \leq_m R$ a ak $P \leq_m R$, tak nutne aj $P \leq_T R$. Tieto implikácie však nemožno obrátiť. Z toho dôvodu sú „many-one“ a „one-one“ redukcie užitočnými nástrojmi na klasifikáciu nerozhodnuteľných problémov podľa ich obtiažnosti, keďže takáto klasifikácia je jemnejšia, než je tomu pri klasifikácii na základe turingovskej redukovateľnosti.

[3] Funkcia $f\colon \Sigma^* \to \Sigma^*$ je vypočítateľná, ak existuje deterministický Turingov stroj so vstupnou a výstupnou páskou taký, že pre všetky obsahy vstupnej pásky $w \in \Sigma^*$ sa stroj v konečnom čase zastaví v konfigurácii, kde obsahom výstupnej pásky je slovo $f(w)$.