Substituce a morfismy jednoduše

Petr Zemek*

31. července 2010†

Abstrakt

Tento text si dává za cíl srozumitelně a formou příkladů osvětlit problematiku substitucí a morfismů v rozsahu předmětu Teoretická informatika (TIN) v magisterských studijních oborech na Fakultě informačních technologií v Brně (FIT). Jsou prezentovány dvě verze výkladu: neformální a formální.

1 Úvod

Během mého studia na FITu, ať již při absolvování předmětu TIN či při přípravě na státní závěrečnou zkoušku jsem nabyl dojmu, že problematika substitucí, morfismů a uzavřeností jazykových tříd vzhledem k těmto operacím nebývá dostatečně pochopena. Tímto textem bych to chtěl napravit, protože si myslím, že po vhodném vysvětlení na tom není nic složitého ;).

Výklad a použité konvence budou založeny na studijní opoře do TINu [5], konkrétně sekci 5.5.2 (Uzávěrové vlastnosti bezkontextových jazyků). Budu prezentovat dvě verze. Nejdříve vše popíši neformálně a ilustruji to na příkladech a poté to zformalizuji. Formální verze je založena na knihách [2] (strany 49–51) a [4] (strany 71 a 423–424), jejichž výklad se mi zdá vhodnější, než ten, který je prezentován v opoře k TINu. K pochopení této problematiky bude ovšem stačit porozumět mému neformálnímu výkladu.

2 Substituce a morfismy neformálně

Abych neformální část co nejvíce zjednodušil, tak se omezím pouze na bezkontextové gramatiky a bezkontextové jazyky – z pohledu TINu na to stejně byl kladen největší důraz. Je třeba si ale uvědomit to, že tyto koncepty jsou univerzální, čili se vztahují i na ostatní jazykové třídy.

2.1 Substituce

Pojem *substituce* lze chápat jako *nahrazení*. V našem případě nahradíme každý symbol nějakou množinou řetězců, tedy jazykem. Vezměme si např. řetězec ab a substituci σ (sigma), která je definována následovně:

$$\sigma(a) = \{0, 00\} \text{ a } \sigma(b) = \{1, 111, 1110\}. \tag{2.1}$$

Znamená to, že každé a se nahradí jazykem $\{0, 00\}$ a každé b se nahradí jazykem $\{1, 111, 1110\}$. Při aplikaci substituce na řetězec se tato substituce aplikuje na každý symbol zvlášť, čili v případě řetězce ab dostaneme

$$\sigma(ab) = \sigma(a)\sigma(b) = \{0, 00\}\{1, 111, 1110\},$$

*Fakulta informačních technologií, Vysoké učení technické v Brně, e-mail: izemek@fit.vutbr.cz

†Poslední revize: 4. listopadu 2015

což nám po provedení konkatenace dá

$$\sigma(ab) = \{01, 0111, 01110, 001, 00111, 001110\}.$$

Jelikož jsou oba jazyky v (2.1) konečné, tak se jedná s *konečnou substituci*. Obecně ale nemusí být substituce konečná. Vezměme si např. substituci definovanou takto:

$$\sigma(a) = \{0^n 1^n \mid n \geq 0\} \text{ a } \sigma(b) = \{0\}^*. \tag{2.2}$$

Tato substituce je *nekonečná*. Po jejím aplikování na ab dostaneme

$$\sigma(ab) = \sigma(a)\sigma(b) = \{0^n 1^n \mid n \geq 0\}\{0\}^*,$$

což nám ve výsledku dá jazyk

$$\sigma(ab) = \{0^n 1^n 0^m \mid m, n \geq 0\}.$$

V obou případech jsme substituci aplikovali na jediný řetězec. Substituci lze ale aplikovat i na jazyk (ten může být i nekonečný), a to tak, že se aplikuje na každý řetězec z tohoto jazyka a výsledky se sjednotí. Vezměme si jazyk $\sigma(a)$ z (2.2) a označme jej L, čili

$$L = \{a^n b^n \mid n \geq 0\}, \tag{2.3}$$

a substituci definovanou následovně:

$$\sigma(a) = L_a \text{ a } \sigma(b) = L_b,$$

kde $L_a = \{0, 1\}$ a $L_b = \{dd, ee, fff, ggg\}$. Pro jednoduchost jsem zvolil konečnou substituci (L_a i L_b jsou konečné jazyky). Po aplikaci σ na L dostaneme jazyk

$$\sigma(L) = \bigcup_{n \geq 0} L_a^n L_b^n = \bigcup_{n \geq 0} \{0, 1\}^n \{dd, ee, fff, ggg\}^n. \tag{2.4}$$

Do jazyka (2.4) tedy patří řetězce ε, $00ddee$, $1fff$, $010gggdddd$, $1111ddddgggee$ apod.

Pokud bych to měl shrnout, tak substituce nám definuje nahrazení každého symbolu nějakým jazykem. Substituce aplikovaná na řetězec je definovaná tak, že se aplikuje na každý symbol v tomto řetězci. A konečně, pokud aplikujeme substituci na jazyk, pak se substituce aplikuje na každý řetězec z tohoto jazyka a výsledky se sjednotí.

2.2 Morfismy

Morfismus je speciální případ substituce, kde každý jazyk obsahuje pouze jediný řetězec. Pro odlišení budu morfismy označovat φ („fí“). Můžeme si tedy zjednodušit zápis, a to tak, že místo

$$\varphi(a) = \{w\}$$

budeme psát jen

$$\varphi(a) = w,$$

kde w je řetězec. Pojďme si ukázat příklad. Mějme morfismus definovaný následovně:

$$\varphi(a) = 00, \varphi(b) = 01 \text{ a } \varphi(c) = 10. \tag{2.5}$$

Pak např.

$$\varphi(abac) = \varphi(a)\varphi(b)\varphi(a)\varphi(c) = 00010010.$$

Vidíme, že na rozdíl od substituce je výsledek aplikování morfismu vždy jednoznačný (vždy dostaneme jediný řetězec). Dá se říct, že morfismus nám řetězec pouze nějak překóduje. Pěkným příkladem morfismu je tzv. *Morseův kód*[1], který je používán v telegrafii. Každé písmeno latinské abecedy se kóduje do posloupnosti znaků · (tečka) a – (pomlčka). Pro názornost uvedu jen kódování pár symbolů:

$$\begin{aligned}\varphi(a) &= \cdot-\\ \varphi(b) &= -\cdots\\ \varphi(c) &= -\cdot-\cdot\\ &\cdots\end{aligned}$$

Řetězec aab se pak zakóduje do

$$\varphi(aab) = \varphi(a)\varphi(a)\varphi(b) = \cdot-\cdot--\cdots .$$

Pomocí morfismu mohu některé znaky vymazat, čili nahradit prázdným řetězcem. Dejme tomu, že bych chtěl z libovolného řetězce znaků 0 a 1 odstranit všechny znaky 1. Lze toho docílit pomocí morfismu φ definovaným jako

$$\varphi(0) = 0 \text{ a } \varphi(1) = \varepsilon.$$

Skutečně,

$$\varphi(0110001) = \varphi(0)\varphi(1)\varphi(1)\varphi(0)\varphi(0)\varphi(0)\varphi(1) = 0000.$$

Dalším využitím morfismů je případ, kdy chci některé znaky přeznačit. Dejme tomu, budu mít množinu neterminálů $N = \{A, B, C\}$ a množinu terminálů $\Sigma = \{a, b, c\}$ a já chci každý neterminál nahradit jeho čárkovanou verzí, tj. místo A budu mít A' atd. Právě toto dělá následující morfismus:

$$\varphi(A) = A', \varphi(B) = B', \varphi(C) = C',$$
$$\varphi(a) = a, \varphi(b) = b \text{ a } \varphi(c) = c.$$

Jak vidíte, tak s přibývajícím množstvím znaků nám roste složitost popisu. V podobných případech lze s výhodou využít následujícího ekvivalentního popisu: definujme morfismus φ tak, že $\varphi(X) = X'$ pro všechna $X \in N$ a $\varphi(x) = x$ pro všechna $x \in \Sigma$.

U morfismu jazyka je to obdobné jako u substituce. Jelikož ale pro každý řetězec existuje právě jeden zakódovaný řetězec, tak je to jednodušší. Pokud si vezmeme jazyk L z (2.3) a morfismus definovaný v (2.5), tak

$$\varphi(L) = \varphi\big(\{a^n b^n \mid n \geq 0\}\big) = \big\{(00)^n(01)^n \mid n \geq 0\big\}. \tag{2.6}$$

Do jazyka (2.6) patří např. řetězce ε, 0001 a 00000101.

2.3 Uzavřenost třídy bezkontextových jazyků

Nejdříve si musíme vysvětlit, co je to tzv. *bezkontextová substituce*. Ta je definována obdobně jako konečná a nekonečná substituce, čili bezkontextová substituce je taková substituce, kde je každý symbol nahrazen bezkontextovým jazykem. Všechny substituce ze sekce 2.1 jsou bezkontextové. Ovšem např. následující substituce

$$\sigma(0) = \{a^n b^n c^n \mid n \geq 0\} \text{ a } \sigma(1) = \{c, d\}$$

bezkontextová není, protože jazyk $\{a^n b^n c^n \mid n \geq 0\}$ patří mezi známé jazyky, které bezkontextové nejsou.

Třída bezkontextových jazyků je *uzavřena vůči substituci*, pokud když si vezmeme libovolný bezkontextový jazyk L a libovolnou bezkontextovou substituci σ, tak $\sigma(L)$ je bezkontextový jazyk. Teď se pokusím méně formálně přeformulovat důkazy vět 5.10 a 5.11 z [5] (dám hlavní ideu).

[1] http://cs.wikipedia.org/wiki/Morseova_abeceda

Věta 1. *Třída bezkontextových jazyků je uzavřena vůči substituci.*

Důkaz. Vezmeme si libovolný bezkontextový jazyk L a libovolnou bezkontextovou substituci σ. Teď musíme ukázat, že $\sigma(L)$ je bezkontextový jazyk. Jelikož je L bezkontextový, tak existuje bezkontextový gramatika $G = (N, \Sigma, P, S)$, která jej generuje. Důkaz je založen na tom, že jelikož je σ bezkontextová substituce, tak pro každý terminál $a \in \Sigma$ existuje bezkontextová gramatika $G_a = (N_a, \Sigma_a, P_a, S_a)$, která generuje $\sigma(a)$. Vytvoříme tedy novou gramatiku G' tak, že každý terminál a na pravé straně každého pravidla v P nahradíme startujícím neterminálem S_a.

Např. pro pravidlo $A \to aBc$ dostaneme pravidlo $A \to S_a B S_c$. Jakmile by tedy v původní gramatice došlo k vygenerování terminálu, tak místo něj se vygeneruje startující neterminál příslušné gramatiky. Z toho se pak vygeneruje řetězec z příslušného jazyka (pro terminál a to bude řetězec z jazyka $\sigma(a)$ atp.). □

Uzavřenost třídy bezkontextových jazyků vůči morfismu je definovaná obdobně. Pokud si vezmeme libovolný bezkontextový jazyk L a libovolný morfismus φ, tak $\varphi(L)$ musí být také bezkontextový jazyk. Všimněte si, že nepožaduji nic takového, aby φ byl „bezkontextový morfismus“, protože to nedává smysl.

Věta 2. *Třída bezkontextových jazyků je uzavřena vůči morfismu.*

Důkaz. Jelikož je morfismus speciálním případem substituce, tak tato věta platí ihned díky Větě 1. Pokud bychom na to chtěli jít konstrukčně, čili pro libovolný bezkontextový jazyk L a libovolný morfismus φ vytvořit bezkontextovou gramatiku, která generuje $\varphi(L)$, tak bychom postupovali stejně jako v důkaze Věty 1 s tím rozdílem, že místo nahrazení každého terminálu a startujícím neterminálem S_a ho přímo nahradíme řetězcem $\varphi(a)$. Tedy např. pokud $\varphi(a) = 000$ a $\varphi(c) = 11$, tak pravidlo $A \to aBc$ nahradíme pravidlem $A \to 000B11$. □

3 Substituce a morfismy formálně

Definice 1. Nechť Σ a Γ jsou abecedy. Funkce $\sigma\colon \Sigma^* \to 2^{\Gamma^*}$ se nazývá *substituce* právě tehdy, když

(1) $\sigma(\varepsilon) = \{\varepsilon\}$,
(2) $\sigma(xy) = \sigma(x)\sigma(y)$ pro všechny řetězce $x, y \in \Sigma^*$.

Ač to vypadá složitě, tak Σ^* označuje množinu všech řetězců nad abecedou Σ a 2^{Γ^*} označuje množinu všech jazyků nad abecedou Γ, viz strana 10 v opoře k TINu [5]. Podmínka (1) se někdy vynechává [2], protože vyplývá z podmínky (2). Podmínka (2) nám zaručuje, že substituci lze definovat tak, jak jsme si ji neformálně definovali v sekci 2.1, čili každému symbolu přiřadíme jazyk a v případě řetězce aplikujeme substituci na každý z jeho symbolů. Podle definice sice přiřazujeme jazyk každému řetězci, ale jelikož se substituce chová „bezkontextově“, tak ji stačí definovat po jednotlivých symbolech. Substituci řetězce pak dostaneme automaticky.

Definice 2. Nechť Σ a Γ jsou abecedy. Funkce $\varphi\colon \Sigma^* \to \Gamma^*$ se nazývá *morfismus* právě tehdy, když

(1) $\varphi(\varepsilon) = \varepsilon$,
(2) $\varphi(xy) = \varphi(x)\varphi(y)$ pro všechny řetězce $x, y \in \Sigma^*$.

Jelikož je morfismus[2] pouze speciální případ substituce, tak je jeho definice velice podobná. Takže to, co platí pro substituci, platí i pro morfismus.

[2]Poznámka pro zvídavé: Σ^* se nazývá *volný monoid* s operací konkatenace a neutrálním prvkem ε a onen morfismus je pouze zkrácený název pro *homomorfismus* nad monoidy, jak jej znáte z MATu. Viz např. sekce 11.1 v [1].

Rozšíření substitucí a morfismů na jazyky dává následující definice.

Definice 3. Nechť Σ a Γ jsou abecedy, $L \subseteq \Sigma^*$ je jazyk nad Σ, $\sigma\colon \Sigma^* \to 2^{\Gamma^*}$ je substituce a $\varphi\colon \Sigma^* \to \Gamma^*$ je morfismus. Pak

$$\sigma(L) = \bigcup_{w \in L} \sigma(w)$$

a

$$\varphi(L) = \{\varphi(w) \mid w \in L\}.$$

Konečně, uzavřenost jazykové třídy vůči substituci a morfismu je formálně definována následovně.

Definice 4. Nechť $\mathcal{L}$ je třída jazyků. $\mathcal{L}$ je *uzavřena vůči substituci* právě tehdy, když pro libovolný jazyk $L \in \mathcal{L}$ a pro libovolnou substituci $\sigma\colon \Sigma^* \to 2^{\Gamma^*}$ takovou, že $\sigma(a) \in \mathcal{L}$ pro všechna $a \in \Sigma$, platí, že $\sigma(L) \in \mathcal{L}$. $\mathcal{L}$ je *uzavřena vůči morfismu* právě tehdy, když pro libovolný jazyk $L \in \mathcal{L}$ a pro libovolný morfismus $\varphi\colon \Sigma^* \to \Gamma^*$ platí, že $\varphi(L) \in \mathcal{L}$.

4 Závěr

Substituce nahrazuje každý symbol jazykem, kdežto morfismus nahrazuje každý symbol řetězcem. Morfismus je speciálním případem substituce. Díky vlastnostem substituce a morfismu stačí, když je definujeme pro jednotlivé symboly, a automaticky je máme definovány pro řetězce. Rozšíření substitucí a morfismů na jazyky je přímočaré.

Pevně doufám, že se mi tyto koncepty podařilo dostatečně objasnit. Vzhledem k tomu, že jsem chtěl tento text vměstnat do max. pěti stránek, tak se sem nevlezla některá témata, např. *inverzní morfismus* a uzavřenost třídy bezkontextových jazyků vůči němu. Věřím ale, že po pochopení těchto základů pochopíte i tento koncept. Zájemcům o rozšiřující a obecnější koncepty se doporučuji podívat na *konečné převody* (*finite transduction*) a *GSM mapování* (*GSM mapping*, od *Generalized Sequential Machine*), viz např. sekce 2.4 v [3].

Reference

[1] Cohn, P. M.: *Further Algebra and Applications*. Springer, 2003, ISBN 1-85233-667-6.

[2] Meduna, A.: *Automata and Languages: Theory and Applications*. Springer, 2000, ISBN 1-85233-074-0.

[3] Rozenberg, G.; Salomaa, A. (editoři): *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*, kapitola 2. Springer, 1997, ISBN 3-540-60420-0, s. 41–110.

[4] Wood, D.: *Theory of Computation: A Primer*. Addison-Wesley Longman Publishing Co., Inc., 1987, ISBN 0-06-047208-1.

[5] Češka, M.; Smrčka, A.; Vojnar, T.: Studijní opora do předmětu Teoretická informatika [online]. Poslední aktualizace 2009-11-18. [cit. 2010-07-31]. Dostupné na URL: <http://www.fit.vutbr.cz/study/courses/TIN/public/Texty/oporaTIN.pdf>.