Deskriptive Pattern und Ketten von Patternsprachen

Dominik D. Freydenberger

Goethe-Universität, Frankfurt am Main

Zusammenfassung Die in diesem Vortrag vorgestellte Forschung setzt die Untersuchungen zur (Nicht-)Existenz deskriptiver Pattern fort, die von Freydenberger und Reidenbach begonnen wurden (*Existence and nonexistence of descriptive patterns*, Theor. Comput. Sci. 411 (2010)). Als technischer Hauptbeitrag können *Kettensysteme* betrachtet werden, ein neuer Mechanismus der die Arbeit mit Ketten von terminalfreien E-Patternsprachen erleichtert. Diese wiederum sind ein wichtiges Werkzeug in Beweisen zu deskriptiven Pattern. Anhand von Kettensystemen lässt such der Hauptbeweis aus Freydenberger und Reidenbach 2010 vereinfachen und verallgemeinern, zudem führen sie zu neuen Erkenntnisse über die Topologie der Klasse der terminalfreien E-Patternsprachen und zur Frage welche Sprachen sich nicht durch solche Patternsprachen approximieren lassen.

1 Ein kurzer Überblick

Ein *Pattern* ist ein Wort $\alpha \in (X \cup \Sigma)^+$, wobei X ein *Variablenalphabet* und Σ ein *Terminalalphabet* ist. Pattern können als kompakte und natürliche Sprachenerzeuger benutzt werden; ein Wort $w \in \Sigma^*$ gehört zur Sprache $L_{\mathrm{E}}(\alpha)$ eines Patterns α wenn es durch Ersetzen aller Variablen in α durch Terminalwörter erzeugt werden kann (wobei gleiche Variablen gleich ersetzt werden). Formaler gesprochen ist

$$L_{\mathrm{E}}(\alpha) := \{\sigma(\alpha) \mid \sigma \text{ ist eine Substitution}\},$$

wobei eine *Substitution* ein Homomorphismus $\sigma : (X \cup \Sigma)^* \to \Sigma^*$ ist, so dass $\sigma(a) = a$ für alle $a \in \Sigma$ gilt.

Zum Beispiel erzeugt das Pattern $\alpha := xx\,\mathtt{a}\,\mathtt{b}\,yy$ (mit $x, y \in X$ und $\mathtt{a}, \mathtt{b} \in \Sigma$) die Sprache $L_{\mathrm{E}}(\alpha) = \{uu\,\mathtt{a}\,\mathtt{b}\,vv \mid u, v \in \Sigma^*\}$. Gewöhnlich unterscheidet man in der Literatur zwischen *E-Patternsprachen* (bei denen das Ersetzen von Variablen durch das leere Wort *E*rlaubt ist) und *NE-Patternsprachen* (bei denen das Löschen von Variablen *N*icht *E*rlaubt ist). Im Gegensatz zu ihren ähnlichen Definitionen haben die Klasse der E- und die Klasse der NE-Patternsprachen oft sehr unterschiedliche Eigenschaften. Dieser Vortrag befasst sich ausschließlich mit E-Patternsprachen.

Patternsprachen wurden zuerst von Angluin [1] in der Lerntheorie eingeführt und, ausgehend von Jiang et al. [6], auch in der Sprachtheorie ausgiebig untersucht. Eine vergleichsweise aktuelle Übersicht zu Patternsprachen in diesen beiden Gebieten finden sich in Ng und Shinohara [7] und Salomaa [8].

Wir nennen ein Pattern δ *deskriptiv* für eine Sprache $L \subseteq \Sigma^*$, wenn

1. $L_{\mathrm{E}}(\delta) \supseteq L$ gilt, und außerdem
2. kein Pattern γ existiert, für das $L_{\mathrm{E}}(\delta) \supset L_{\mathrm{E}}(\gamma) \supseteq L$ gilt.

Ein Pattern, das für eine Sprache L deskriptiv ist, kann also als eine der besten Abschätzungen von L verstanden werden (im Sinne von inklusionsminimalen Generalisierungen), die innerhalb der Klasse der E-Patternsprachen möglich sind. Aus diesem Grunde spielen deskriptive Pattern eine wichtige Rolle in der Lerntheorie (bereits seit [1]). Dieser Ansatz wurde, zusammen mit Resultaten und Techniken aus Freydenberger und Reidenbach [4] zur Existenz deskriptiver Pattern, in Freydenberger und Reidenbach [5] weiterentwickelt zur *deskriptiven Generalisierung*, einem mächtigen Verfahren zum approximativen Lernen mittels deskriptiver Pattern.

Dieses Lernverfahren wiederum wurde kürzlich durch Freydenberger und Kötzing [3] zum Lernen von eingeschränkten regulären Ausdrücken verwendet, die von großer Bedeutung in der XML-Schemainferenz sind. Diese Entwicklung kann durchaus als überraschend betrachtet werden, da die Klasse der regulären Sprachen und die Klasse der Patternsprachen unvergleichbar sind und eine vollkommen unterschiedliche Topologie besitzen. Dennoch ließen sich viele der Techniken zu deskriptiven Pattern aus [5] (und damit indirekt auch [4]) auf die in [4] betrachteten regulären Ausdrücke übertragen.

Aus Sicht des Autors zeigt dies, dass die Untersuchung deskriptiver Pattern sowie der deskriptiven Generalisierung nicht nur im Kontext von Patternsprachen von Bedeutung ist, sondern auch darüber hinaus zu überraschenden Anwendungen in anderen Gebieten führen kann.

Im Gegensatz zur einfachen Definition von Patternsprachen sind viele kanonische Fragen zur Patternsprachen überraschend schwer. Dies gilt auch für Fragen zu deskriptiven Pattern. Insbesondere wurde die Frage, ob zu jeder Sprache ein deskriptives Pattern existiert, bereits von Jiang et al. [6] gestellt, aber erst von Freydenberger and Reidenbach [4] negativ beantwortet.

Damit zu einer Sprache L *kein* Pattern deskriptiv ist, muss zwischen jedem Pattern α mit $L_{\mathrm{E}}(\alpha) \supseteq L$ und L selbst eine unendliche absteigende *Kette* von Patternsprachen existieren. Dies ist leicht zu sehen: Falls kein Pattern deskriptiv für L ist, muss zu jedem Pattern α_i mit $L_{\mathrm{E}}(\alpha_i) \supset L$ ein weiteres Pattern α_{i+1} existieren, für das $L_{\mathrm{E}}(\alpha_i) \supset L_{\mathrm{E}}(\alpha_{i+1}) \supset L$ gilt. Dieser Prozess lässt sich unendlich fortsetzen, so dass die erwähnte Kette entsteht.

Eine solche Kette ist daher auch essentieller Bestandteil des Beweis in [4] zur Existenz einer Sprache, für die kein Pattern deskriptiv ist. Dieser Beweis gibt außerdem Grund zu der Vermutung, dass die Struktur jeder solchen Sprache stark mit der Struktur der entsprechenden Kette (oder auch Ketten) zusammenhängt. Nach eingehender Beschäftigung mit diesen Zusammenhängen könnte man zu dem Schluss kommen, dass sich auf diese Art eine Charakterisierung erstellen lässt, die die Sprachen beschreibt, für die kein Pattern deskriptiv ist.

Dies ist allerdings ein Trugschluss: Anhand der in diesem Vortrag vorgestellten Resultate lässt sich zeigen, dass eine solche Charakterisierung wahrscheinlich recht technisch sein dürfte, so sie überhaupt gefunden werden kann. Um dies zu

beweisen wird ein neues Spracherzeugungsmodell eingeführt, die sogenannten *Kettensysteme*.

Ein Kettensystem besteht aus einem Startpattern α_0 und einer unendlichen Folge von Homomorphismen $(\phi_i)_{i\in\mathbb{N}}$. Ein Kettensytem $C = (\alpha_0, (\phi_i)_{i\in\mathbb{N}})$ erzeugt durch $\alpha_{i+1} := \phi_i(\alpha_i)$ eine folge von Pattern, und mittels $L(C) := \bigcap_{i\in\mathbb{N}} L_{\mathrm{E}}(\alpha_i)$ eine Sprache. Hauptbeitrag der in diesem Vortrag vorgestellten Arbeit ist ein Sortiment von Resultaten zu Kettensystemen, die das Arbeiten mit diesem Modell erleichtern (insbesondere auch auf die Schwierigkeiten, die im Umgang mit einem Modell entstehen, dass über eine unendliche Folge von Homomorphismen definiert wird).

Anhand dieser Resultate wird der Hauptbeweis aus [4] vereinfacht, und es können mehrere schwere Gegenbeispiele definiert werden, die die Vermutung einer einfachen Charakterisierung zwar nicht endgültig widerlegen, jedoch sehr unplausibel erscheinen lassen. Obwohl die in diesen Beispielen verwendeten Sprachen auf den ersten Blick sehr kompliziert wirken mögen, wird gezeigt, dass sie allesamt EDT0L-Sprachen sind.

Die meisten der im Vortrag vorgestellten Resultate erschienen zuvor in [2], der Dissertation des Autors (dieser ist übrigens gerne bereit, Interessenten auf Nachfrage ein persönliches gedrucktes Exemplar zu überlassen).

Literaturverzeichnis

[1] D. Angluin. Finding patterns common to a set of strings. *J. Comput. Syst. Sci.*, 21:46–62, 1980.

[2] D. D. Freydenberger. *Inclusion of pattern languages and related problems.* PhD thesis, Institut für Informatik, Goethe-Universität Frankfurt am Main, 2011. Logos Verlag, Berlin.

[3] D. D. Freydenberger and T. Kötzing. Fast learning of restricted regular expressions and DTDs. In *Proc. ICDT 2013*, pages 45–56, 2013.

[4] D. D. Freydenberger and D. Reidenbach. Existence and nonexistence of descriptive patterns. *Theor. Comput. Sci.*, 411(34-36):3274–3286, 2010.

[5] D. D. Freydenberger and D. Reidenbach. Inferring descriptive generalisations of formal languages. *J. Comput. Syst. Sci.*, 79(5):622–639, 2013.

[6] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, and S. Yu. Pattern languages with and without erasing. *Int. J. Comput. Math.*, 50:147–163, 1994.

[7] Y. K. Ng and T. Shinohara. Developments from enquiries into the learnability of the pattern languages from positive data. *Theor. Comput. Sci.*, 397:150–165, 2008.

[8] K. Salomaa. Patterns. In C. Martin-Vide, V. Mitrana, and G. Păun, editors, *Formal Languages and Applications*, number 148 in Studies in Fuzziness and Soft Computing, pages 367–379. Springer, 2004.