Синтаксический анализ данных, представленных в виде контекстно-свободной грамматики

Ковалев Дмитрий Александрович

1 Постановка задачи

Целью данной работы является разработка алгоритма синтаксического анализа данных, представленных в виде контекстно-свободной грамматики.

Данная цель приводит к необходимости решения задач, связанных с проверкой пустоты пересечения двух контекстно-свободных языков. Известно, что в общем случае данная проблема неразрешима. В связи с этим, для достижения цели задачи были сформулированы следующим образом.

- Изучить существующие подходы к анализу данных, представленных в виде КС-грамматик.
- Определить ограничения, при которых синтаксический анализ такого представления является разрешимой задачей.
- Разработать алгоритм синтаксического анализа контекстно-свободного представления данных с учетом поставленных ограничений.
- Реализовать алгоритм в рамках проекта YaccConstructor.
- Доказать завершаемость алгоритма.
- Провести тестирование и апробацию.

2 Обзор

Пусть G — произвольная КС-грамматика, M — конечный автомат. Тогда задача проверки

- включения языков ($L(M) \subseteq L(G)$) — неразрешима
- пустоты пересечения ($L(M) \cap L(G) = \emptyset$) — разрешима (т.к. в пересечении не более чем КС-язык) за полиномиальное время [5]
- регулярности языка $L(G)$ — неразрешима [4]

Если использовать представление регулярного языка $L(M)$ в виде КС-грамматики G_r, то задача проверки пустоты пересечения ($L(G_r) \cap L(G) = \emptyset$) становится немного интереснее: если G_r

- нерекурсивная — задача из PSPACE [6] (точнее результата нет (я не нашел, по крайней мере))
- лево- или праволинейная — ничего не известно (см. последний абзац заключения из [6])
- принадлежит еще более широкому классу — тем более ничего не известно

Еще немного про вложенную рекурсию и регулярность языка. Грамматика без вложенной рекурсии (NSE) порождает регулярный язык [3] (обратное тоже верно, для регулярного языка можно построить NSE грамматику, т.к. праволинейная, например, — частный случай NSE). Существует алгоритм, который позволяет проверять грамматику на наличие вложенной рекурсии за полином [2]. Однако, грамматика с вложенной рекурсией тоже может порождать регулярный язык [1], поэтому задача о проверке регулярности языка, порождаемого КС-грамматикой, остается неразрешимой.

3 Алгоритм

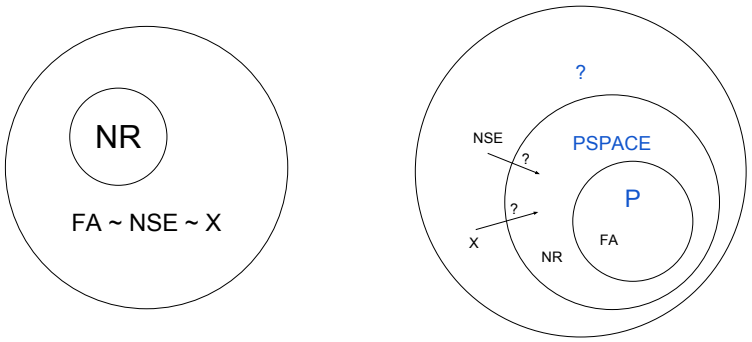
Мы пытаемся решать следующую задачу: пусть $G_1 = (N, T, S, P)$ — произвольная КС-грамматика, G_2 — NSE КС-грамматика. Алгоритм принимает на вход два рекурсивных автомата, M_1 и M_2, построенных по грамматикам G_1 и G_2 соотвественно, при этом в автомате M_2 левые/правые рекурсивные вызовы заменены на циклы, как в обычном конечном автомате.

Результатом работы алгоритма являются тройки вида (X, n_1, n_2), где $X \in N$, n_1, n_2 — номера состояний автомата M_2. Для каждой из таких троек выполняется следующее утверждение: $\exists\, \omega \in T^*$ такая, что $X \to^* \omega$ в G_1 и $\omega \in L(M')$, где M' — рекурсивный автомат, полученный из M_2 путем замены начального и конечного состояния на n_1 и n_2 соответственно.

Получая такие результаты, мы, по сути, отвечаем на вопрос о проверке пустоты пересечения КС-языка и регулярного, представленных в необычных абстракциях. Для КС-языка мы используем рекурсивный автомат, а для регулярного — нечто среднее между конечным автоматом и NSE грамматикой (это нечто все еще использует стек, но только для обработки нерекурсивных вызовов). Такое представление эквивалентно по выразительности NSE и, следовательно, FA (см. рис. 1a). Но неизвестно, к какому классу сложности относится задача (и разрешима ли вообще) о проверке пустоты пересечения регулярного языка, представленного в данной форме, с КС-языком (см. рис. 1b).

Theorem 1 *Завершаемость. Алгоритм завершает работу за конечное число шагов для произвольных входных данных*

Theorem 2 *Корректность. ???*



(a) Выразительность языка, задаваемого абстракцией

(b) Класс сложности для задачи о проверке пустоты пересечения с КС-языком

Рис. 1: Красивые круги. Здесь NR — нерекурсивная грамматика, FA — конечный автомат, NSE — грамматика без вложенной рекурсии, X — наше представление

Список литературы

[1] S. Andrei, W.-N. Chin, and S. V. Cavadini. Self-embedded context-free grammars with regular counterparts. *Acta Informatica*, 40(5):349–365, 2004.

[2] M. Anselmo, D. Giammarresi, and S. Varricchio. Finite automata and non-self-embedding grammars. In *Proceedings of the 7th International Conference on Implementation and Application of Automata*, CIAA'02, pages 47–56, Berlin, Heidelberg, 2003. Springer-Verlag.

[3] N. Chomsky. A note on phrase-structure grammars. *Information and Control.*, 2:393 – 395, 1959.

[4] S. Greibach. A note on undecidable properties of formal languages. *Mathematical systems theory*, 2(1):1–6, 1968.

[5] H. B. Hunt, III, D. J. Rosenkrantz, and T. G. Szymanski. On the equivalence, containment, and covering problems for the regular and context-free languages. *J. Comput. Syst. Sci.*, 12(2):222–268, Apr. 1976.

[6] M.-J. Nederhof and G. Satta. The language intersection problem for non-recursive context-free grammars. *Inf. Comput.*, 192(2):172–184, Aug. 2004.