

1 Краткое содержание

- Введение. Алфавит, слово, язык.
- Способы задания языка: порождение, генерация, грамматика, уравнения, логика над позициями.
- Грамматика, вывод, дерево вывода, однозначные, неоднозначные языки и грамматики
- Конъюнктивные языки, грамматики. Булевы языки, грамматики.
- нормальная форма Хомского
- СΥΚ
- Синтаксический анализ графов
- Перемножение матриц, алгоритм Уоршалла
- LL, FIRST, FOLLOW, устранение левой рекурсии, левая факторизация, не LL языки
- GLL, поддержка EBNF, поддержка нелинейного входа
- Tree languages, SPPF — regular tree language
- Почему работает синтаксический анализ графов
- PEG
- комбинаторы

2 Лекция 1. Введение

2.1 Алфавит, слово, язык

2.2 Способы задания языка

3 Лекция 2. Грамматика, вывод

3.1 !!!

3.2 Способы задания языка

4 Лекция k. Tree languages

Вспомним определение языка. Вообще говоря, мы просто описываем правила построения некоторых множеств. Давайте конструировать множества деревьев.

Давайте взглянем на СΥК. На тот СΥК, который умеет строить деревья. Или просто на правила построения деревьев вывода. Вообще говоря, если не разделять шаги, то мы можем строить множества деревьев по каким-то правилам.

Эти множества могут обладать некоторыми интересными свойствами и их можно изучать как обобщённые структуры, а не страдать с каждым деревом отдельно. Вы наверняка знаете про бинарные деревья, вообще n -арные, сбалансированные и т.д.

Definition 1 (Tree grammar). Tree grammar $G = (S, N, F, P)$

- S — аксиома. Сартовый нетерминал.
- N — нетерминальные символы
- F — терминальные символы
- P — продукции вида $\alpha \rightarrow \beta$, α, β — деревья.

Грамматика над деревьями называется регулярной, если все нетерминальные символы имеют арность 0, и все продукции имеют вид $A \rightarrow \beta$, $A \in N$, $\beta \in T(F \cup N)$

Пример — списки натуральных чисел.

Theorem 1. 1. G — CF grammar, тогда множество всех деревьев разбора $L(G)$ — регулярный язык деревьев.

2. L — регулярный язык деревьев, тогда $Yield(L)$ — контекстно свободный язык.

3. Существуют регулярные языки деревьев, не являющиеся множеством деревьев вывода никакого плоского КС.

Пункт 1. $G = (S, N, \Sigma, P)$ — КС грамматика. $G' = (S, N, F, P')$ — регулярная грамматика деревьев.

- $F = \Sigma \cup \{\varepsilon\} \cup \{A_n | A \in N, \exists A \rightarrow \alpha \in P, |\alpha| = n\}$
- Если $(A \rightarrow \varepsilon) \in P$, тогда $A \rightarrow A_0(\varepsilon) \in P'$
- Если $(A \rightarrow a_1 \dots a_p) \in P$, тогда $A \rightarrow A_p(a_1, \dots, a_p) \in P'$

Интересен пункт 3.

$G = (S, N, F, P), F = (s(,); g(); a; b), N = (S; G'; G''), P = (S \rightarrow s(G', G''); G' \rightarrow g(a); G'' \rightarrow g(b))$ В $L(G)$ есть единственное дерево $s(g(a), g(b))$.

Если попробуем построить КС-грамматику, то получим что-то вроде $S \rightarrow GG, G \rightarrow a, G \rightarrow b$. В таком языке есть "aa" $S(G(a), G(a))$

Список литературы

- [1] Dowell R. D., Eddy S. R. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction //BMC bioinformatics. — 2004. — Т. 5. — №. 1. — С. 1.
- [2] Eddy S. R. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure //BMC bioinformatics. — 2002. — Т. 3. — №. 1. — С. 1.
- [3] Yogeve S., Milo N., Ziv-Ukelson M. StemSearch: RNA search tool based on stem identification and indexing //Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on. — IEEE, 2013. — С. 145-152.
- [4] Eddy S. R. Homology searches for structural RNAs: from proof of principle to practical use //RNA. — 2015. — Т. 21. — №. 4. — С. 605-607.
- [5] Anderson J. W. J. et al. Evolving stochastic context-free grammars for RNA secondary structure prediction //BMC bioinformatics. — 2012. — Т. 13. — №. 1. — С. 78.