Графовые грамматики

Тихон Пшеницын

Математический институт им. В. А. Стеклова Российской академии наук
Московский государственный университет им. М. В. Ломоносова

29 июня 2023 года

ДИСКЛЕЙМЕР

Это обзорный доклад.

ДИСКЛЕЙМЕР

Это обзорный доклад.
Цель — разрекламировать область, показать интересные направления.

Порождающие грамматики

Определение

Порождающая грамматика — *структура вида* $Gr = \langle N, \Sigma, P, S \rangle$, *где*

- N — *конечный алфавит нетерминальных символов;*
- Σ — *конечный алфавит терминальных символов* ($N \cap \Sigma = \emptyset$);
- $S \in N$ — *стартовый символ;*
- P — *конечный набор правил вида* $\alpha \to \beta$, *где* α *и* β — *строки над алфавитом* $N \cup \Sigma$.

Порождающие грамматики

Определение

Порождающая грамматика — структура вида $Gr = \langle N, \Sigma, P, S \rangle$, где

- *N — конечный алфавит нетерминальных символов;*
- *Σ — конечный алфавит терминальных символов ($N \cap \Sigma = \emptyset$);*
- *$S \in N$ — стартовый символ;*
- *P — конечный набор правил вида $\alpha \to \beta$, где α и β — строки над алфавитом $N \cup \Sigma$.*

- Применение правила $\alpha \to \beta$ — замена α на β внутри строки: $\eta\alpha\theta \Rightarrow \eta\beta\theta$.

Порождающие грамматики

Определение

Порождающая грамматика — структура вида $Gr = \langle N, \Sigma, P, S \rangle$, где

- *N — конечный алфавит нетерминальных символов;*
- *Σ — конечный алфавит терминальных символов ($N \cap \Sigma = \emptyset$);*
- *$S \in N$ — стартовый символ;*
- *P — конечный набор правил вида $\alpha \to \beta$, где α и β — строки над алфавитом $N \cup \Sigma$.*

- Применение правила $\alpha \to \beta$ — замена α на β внутри строки: $\eta\alpha\theta \Rightarrow \eta\beta\theta$.
- Язык, задаваемый Gr, состоит из строк $w \in \Sigma^*$, таких, что $S \Rightarrow^* w$.

Пример

Пример

- *Нетерминальные символы:* A, B, S, T;
- *Терминальные символы:* b;
- S — *стартовый символ;*
- *Правила:*

1. $S \to AS$,
2. $S \to BT$,
3. $AB \to BBA$,
4. $AT \to T$,
5. $T \to \Lambda$,
6. $B \to b$

Пример вывода

$$S \Rightarrow AS \Rightarrow AAS \Rightarrow AABT \Rightarrow ABBAT \Rightarrow BBABAT \Rightarrow BBBBAAT \Rightarrow$$

$$\Rightarrow BBBBAT \Rightarrow BBBBT \Rightarrow^* bbbb$$

Иерархия Хомского

название	**вид правил**	**сложность**
неограниченные	без ограничений	Σ_1-полные
контекстно-зависимые	$\eta A\theta \to \eta\alpha\theta$	PSPACE-полные
контекстно-свободные	$A \to \alpha$	P
праволинейные	$A \to aB$, $A \to a$, $A \to \Lambda$	P

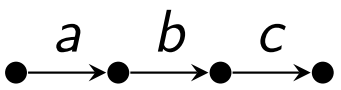
- Для лингвистических применений необходимы грамматики между контекстно-свободными и контекстно-зависимыми.
- В парсере YACC возникают контекстно-свободные грамматики.
- Синтаксис логических языков обычно задается контекстно-свободными грамматиками.
- Праволинейные грамматики = конечные автоматы.
- Неограниченные грамматики — универсальная модель вычисления.

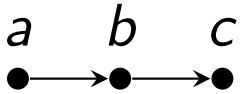
Обобщение контекстно-свободных грамматик на графы

- Правила в КСГ имеют вид $A \to \alpha$. Символ замещается строкой.

Обобщение контекстно-свободных грамматик на графы

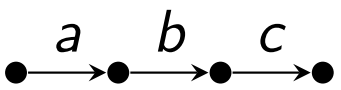
- Правила в КСГ имеют вид $A \to \alpha$. Символ замещается строкой.
- Два способа представления строки abc в виде графа:

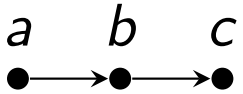




Обобщение контекстно-свободных грамматик на графы

- Правила в КСГ имеют вид $A \to \alpha$. Символ замещается строкой.
- Два способа представления строки abc в виде графа:

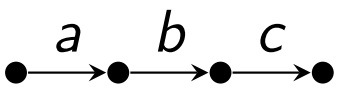


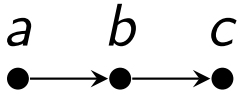


- Графовое обобщение: _____ замещается графом.

Обобщение контекстно-свободных грамматик на графы

- Правила в КСГ имеют вид $A \to \alpha$. Символ замещается строкой.
- Два способа представления строки abc в виде графа:





- Графовое обобщение: _____ замещается графом.
- Элементарный объект — либо вершина, либо ребро.

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?
- Идея: добавить к правилу “инструкцию по подключению” (connection instruction).

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?
- Идея: добавить к правилу "инструкцию по подключению" (connection instruction).
- Инструкция по подключению — пара (μ, δ), где $\mu, \delta \in N \cup \Sigma$. Смысл: если $X \to H$ применяется к вершине v, то надо соединить каждую вершину с меткой μ, смежную с v, с каждой вершиной с меткой δ из H.

Node replacement grammars

- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?
- Идея: добавить к правилу "инструкцию по подключению" (connection instruction).
- Инструкция по подключению — пара (μ, δ), где $\mu, \delta \in N \cup \Sigma$. Смысл: если $X \to H$ применяется к вершине v, то надо соединить каждую вершину с меткой μ, смежную с v, с каждой вершиной с меткой δ из H.
- NLC (Node Label Controlled) графовая грамматика — структура вида $\langle N, \Sigma, P, S, C \rangle$. Новое:

Node replacement grammars

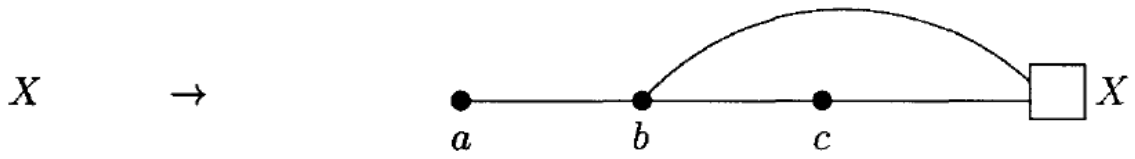
- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?
- Идея: добавить к правилу "инструкцию по подключению" (connection instruction).
- Инструкция по подключению — пара (μ, δ), где $\mu, \delta \in N \cup \Sigma$. Смысл: если $X \to H$ применяется к вершине v, то надо соединить каждую вершину с меткой μ, смежную с v, с каждой вершиной с меткой δ из H.
- NLC (Node Label Controlled) графовая грамматика — структура вида $\langle N, \Sigma, P, S, C \rangle$. Новое:
  1. S — стартовый граф;

Node replacement grammars

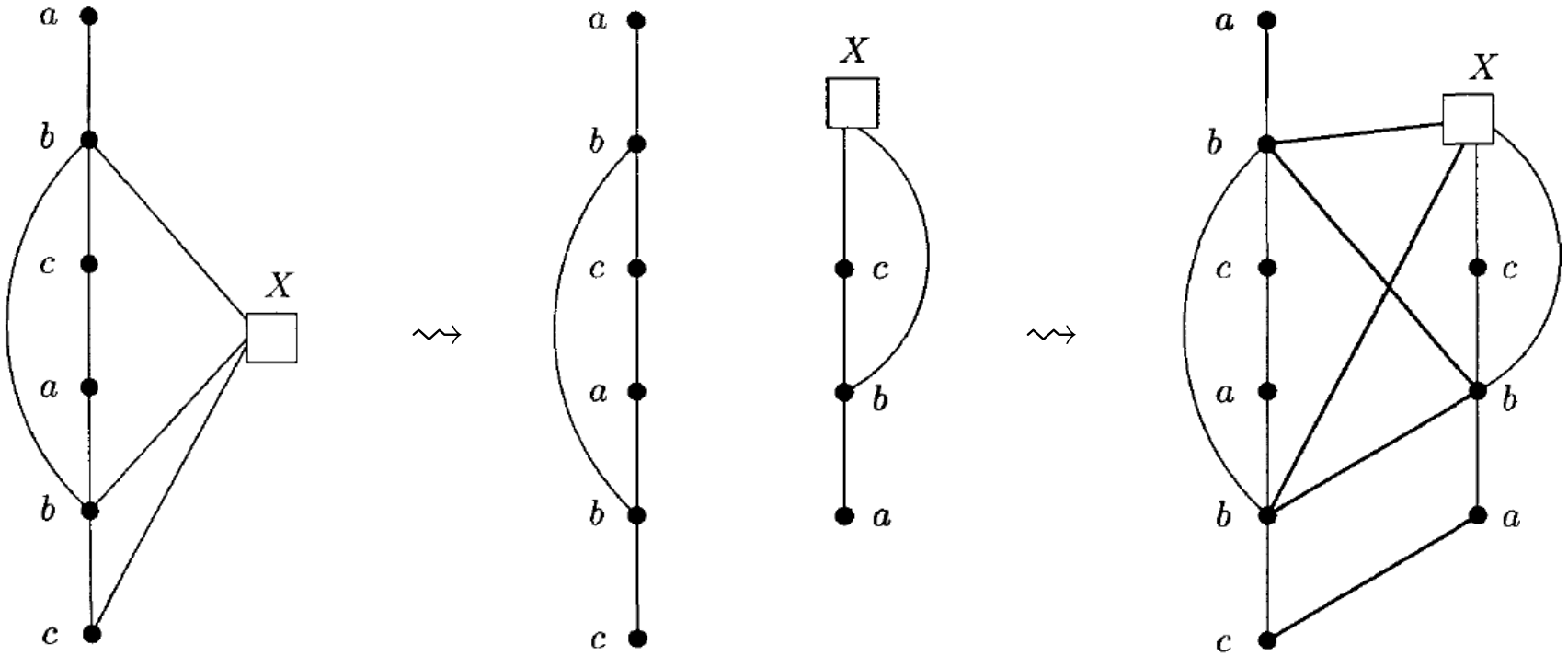
- Опять же есть алфавит нетерминальных символов N и терминальных символов Σ.
- Рассматриваются неориентированные графы с метками на вершинах.
- Правило: $X \to H$, где $X \in N$, а H — граф с метками из $N \cup \Sigma$.
- Если хотим подставить вместо вершины граф, что делать с ребрами, которые ведут в удаляемую вершину?
- Идея: добавить к правилу "инструкцию по подключению" (connection instruction).
- Инструкция по подключению — пара (μ, δ), где $\mu, \delta \in N \cup \Sigma$. Смысл: если $X \to H$ применяется к вершине v, то надо соединить каждую вершину с меткой μ, смежную с v, с каждой вершиной с меткой δ из H.
- NLC (Node Label Controlled) графовая грамматика — структура вида $\langle N, \Sigma, P, S, C \rangle$. Новое:
  1. S — стартовый граф;
  2. $C \subseteq (N \cup \Sigma)^2$ — набор инструкций по подключению.

Пример

Пусть дано правило вида:



Пусть инструкции по подключению таковы: (c, a), (b, b), (b, X). Ниже приведён пример применения правила:



(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.
- Правило: $X \to H$, где $X \in N$, H — граф с метками из $N \cup \Sigma$.

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.
- Правило: $X \to H$, где $X \in N$, H — граф с метками из $N \cup \Sigma$.
- Применение правила: если в G ребро e имеет метку X, то

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.
- Правило: $X \to H$, где $X \in N$, H — граф с метками из $N \cup \Sigma$.
- Применение правила: если в G ребро e имеет метку X, то
  1. удалить X из G;

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.
- Правило: $X \to H$, где $X \in N$, H — граф с метками из $N \cup \Sigma$.
- Применение правила: если в G ребро e имеет метку X, то
  1. удалить X из G;
  2. добавить H;

(Hyper)Edge replacement grammars

- Допустим, теперь рассматриваются графы с метками на рёбрах.
- Чтобы вместо ребра подставить граф, нужно понимать, по каким вершинам его подклеивать к вершинам, инцидентным удаляемому ребру. Также важен порядок подсоединения, чтобы применение данного правила было детерминировано.
- Рассматриваем ориентированные графы с метками на рёбрах и с двумя выделенными вершинами (называемыми внешними). Формально: $G = \langle V_G, E_G, att_G, lab_G, ext_G \rangle$, где
  - V_G — вершины, E_G — рёбра,
  - $att_G : E_G \to (V_G \times V_G)$ — функция прикрепления,
  - $lab_G : E_G \to \mathcal{T}$ — функция разметки,
  - $ext_G \in (V_G \times V_G)$ — внешние вершины.
- Правило: $X \to H$, где $X \in N$, H — граф с метками из $N \cup \Sigma$.
- Применение правила: если в G ребро e имеет метку X, то
  1. удалить X из G;
  2. добавить H;
  3. отождествить i-ю вершину, к которой было прикреплено e, с i-й внешней вершиной H.

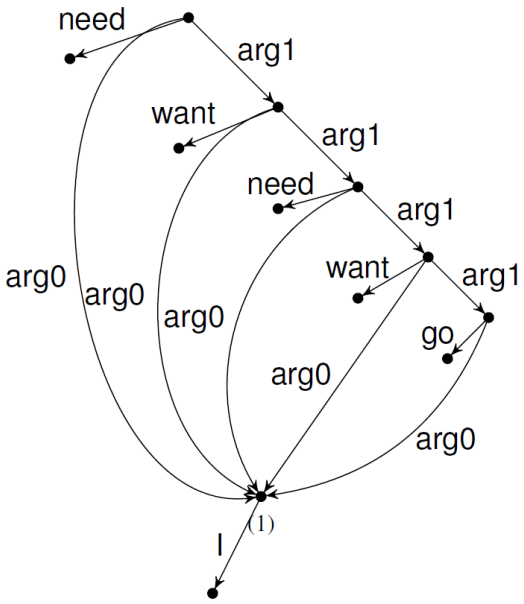
Пример

$$X \to \begin{tikzcd} & \bullet \arrow[dl, "e"'] \\ (1) \bullet \arrow[r, "c"'] & \bullet\, (2) \arrow[u, "d"'] \end{tikzcd}$$

$$(1) \bullet \xrightarrow{X} \bullet \xrightarrow{b} \bullet\, (2) \quad \Rightarrow \quad \begin{tikzcd} & \bullet \arrow[dl, "e"'] \\ (1) \bullet \arrow[r, "c"'] & \bullet \arrow[u, "d"'] \arrow[r, "b"'] & \bullet\, (2) \end{tikzcd}$$

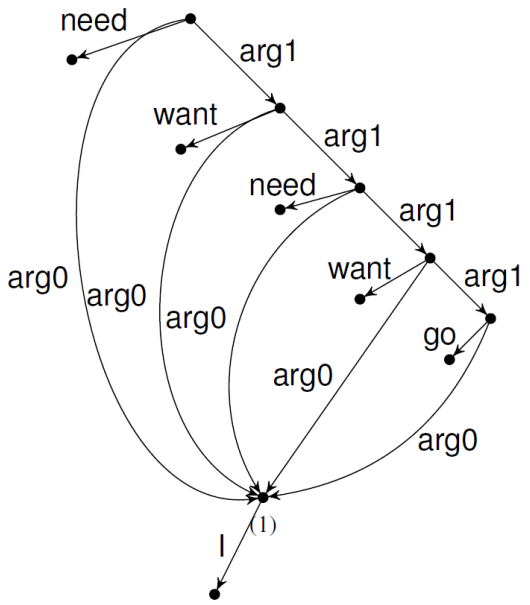
Применения в лингвистике

- S. Gilroy, A. Lopez, S. Maneth (2017). Parsing Graphs with Regular Graph Grammars. Задание abstract meaning representation с помощью грамматик замещения гиперрёбер.



Применения в лингвистике

- S. Gilroy, A. Lopez, S. Maneth (2017). Parsing Graphs with Regular Graph Grammars. Задание abstract meaning representation с помощью грамматик замещения гиперрёбер.



- D. Bauer, O. Rambow (2016). Hyperedge Replacement and Nonprojective Dependency Structures. Задание не-контекстно-свободных языковых явлений (cross-serial dependencies в голландском) с помощью грамматик замещения гиперрёбер.

Коллажевые грамматики

- Применение — порождение изображений с регулярным паттерном.

Коллажевые грамматики

- Применение — порождение изображений с регулярным паттерном.
- F. Drewes, R. Klempien-Hinrichs (2000). Picking knots from trees – The Syntactic Structure of Celtic Knotwork. Использование table-driven collage grammars для задания языков кельтских узоров.

Коллажевые грамматики

- Применение — порождение изображений с регулярным паттерном.
- F. Drewes, R. Klempien-Hinrichs (2000). Picking knots from trees – The Syntactic Structure of Celtic Knotwork. Использование table-driven collage grammars для задания языков кельтских узоров.

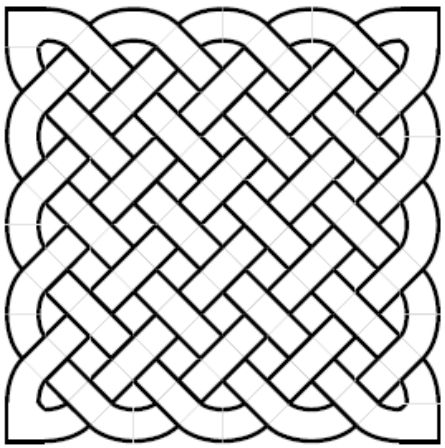


Fig. 2. A square plait

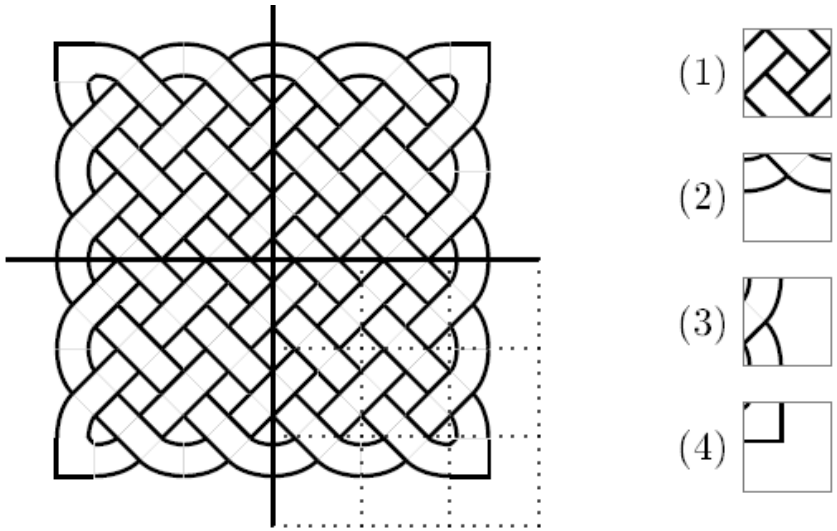
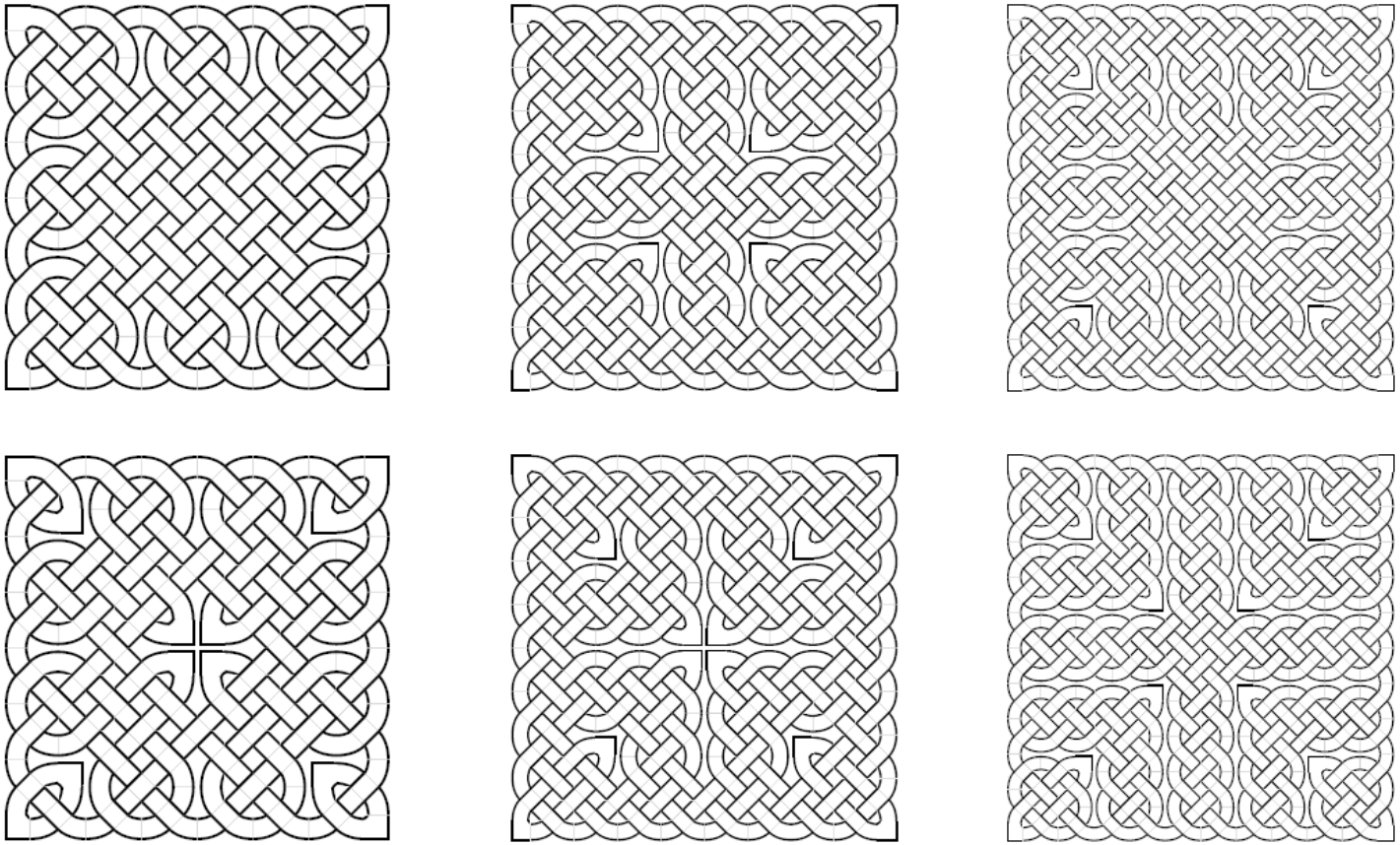


Fig. 3. Dividing the plait into tiles

Коллажевые грамматики

- Применение — порождение изображений с регулярным паттерном.
- F. Drewes, R. Klempien-Hinrichs (2000). Picking knots from trees – The Syntactic Structure of Celtic Knotwork. Использование table-driven collage grammars для задания языков кельтских узоров.



Коллажевые грамматики

- Применение — порождение изображений с регулярным паттерном.
- F. Drewes, R. Klempien-Hinrichs (2000). Picking knots from trees – The Syntactic Structure of Celtic Knotwork. Использование table-driven collage grammars для задания языков кельтских узоров.

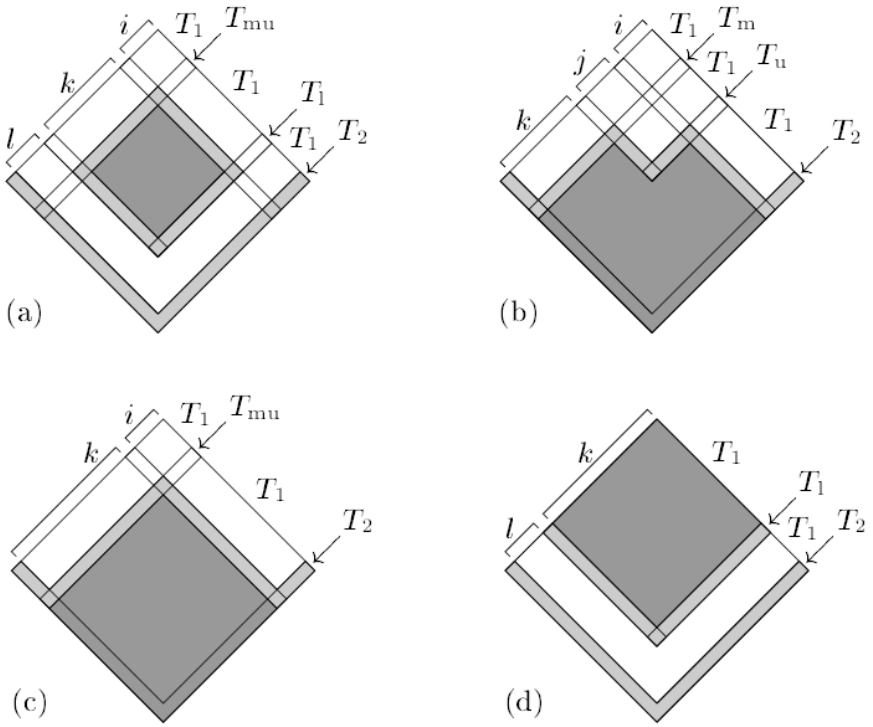


Fig. 15. Schemata of design quarters: (a) cross panel with square holes, (b) crosslet with L-'holes', (c) crosslet with square 'holes', (d) border

Графовые категориальные грамматики

- Категориальный подход: элементарным единицам (символам) поставлены в соответствие категории (формулы) некоторого логического исчисления $\mathcal{L}$; также некоторая категория S выделена. Строка $a_1 \dots a_n$ задаётся категориальной грамматикой, если каждый её символ a_i можно заменить соответствующей ему категорией T_i так, что $\mathcal{L} \vdash T_1, \dots, T_n \to S$.

Графовые категориальные грамматики

- Категориальный подход: элементарным единицам (символам) поставлены в соответствие категории (формулы) некоторого логического исчисления $\mathcal{L}$; также некоторая категория S выделена. Строка $a_1 \dots a_n$ задаётся категориальной грамматикой, если каждый её символ a_i можно заменить соответствующей ему категорией T_i так, что $\mathcal{L} \vdash T_1, \dots, T_n \to S$.
- Можно ли обобщить категориальный подход на графы?

Графовые категориальные грамматики

- Категориальный подход: элементарным единицам (символам) поставлены в соответствие категории (формулы) некоторого логического исчисления $\mathcal{L}$; также некоторая категория S выделена. Строка $a_1 \dots a_n$ задаётся категориальной грамматикой, если каждый её символ a_i можно заменить соответствующей ему категорией T_i так, что $\mathcal{L} \vdash T_1, \dots, T_n \to S$.
- Можно ли обобщить категориальный подход на графы?
- Это потребовало бы разработки секвенциального исчисления, которое выводит структуры вида $H \to S$, где S — формула, а H — граф с метками-формулами.

Графовые категориальные грамматики

- Категориальный подход: элементарным единицам (символам) поставлены в соответствие категории (формулы) некоторого логического исчисления $\mathcal{L}$; также некоторая категория S выделена. Строка $a_1 \dots a_n$ задаётся категориальной грамматикой, если каждый её символ a_i можно заменить соответствующей ему категорией T_i так, что $\mathcal{L} \vdash T_1, \dots, T_n \to S$.
- Можно ли обобщить категориальный подход на графы?
- Это потребовало бы разработки секвенциального исчисления, которое выводит структуры вида $H \to S$, где S — формула, а H — граф с метками-формулами.
- Опять же, в теории можно разработать вершинные категориальные грамматики и рёберные категориальные грамматики.

Гиперграфовые грамматики Ламбека и hyperedge replacement grammars

T. Pshenitsyn (2022). Hypergraph Lambek grammars. Рёберные гиперграфовые категориальные грамматики.

класс грамматик	**выразительная сила**	**сложность**
hypergraph Lambek grammars	язык всех графов языки с квадратичным/ экспоненциальным ростом	NP-полные
hyperedge replacement grammars	лемма-насос теорема Париха	NP-полные

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.
- Хотим обобщить неограниченные грамматики Хомского на графы. Правило вида $L \to R$ должно мочь заменить произвольный подграф вида L графа H на R.

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.
- Хотим обобщить неограниченные грамматики Хомского на графы. Правило вида $L \to R$ должно мочь заменить произвольный подграф вида L графа H на R.
- Правило — тройка: $(L \xleftarrow{\varphi_L} I \xrightarrow{\varphi_R} R)$, где φ_L, φ_R — морфизмы.

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.
- Хотим обобщить неограниченные грамматики Хомского на графы. Правило вида $L \to R$ должно мочь заменить произвольный подграф вида L графа H на R.
- Правило — тройка: $(L \xleftarrow{\varphi_L} I \xrightarrow{\varphi_R} R)$, где φ_L, φ_R — морфизмы.
- Применение правила:

$$\begin{tikzcd}
L \arrow[d, "m"] & I \arrow[l, "\varphi_L"'] \arrow[r, "\varphi_R"] \arrow[d, "\psi"] & R \arrow[d, "n"] \\
G & C \arrow[l, "\eta_L"'] \arrow[r, "\eta_R"] & H
\end{tikzcd}$$

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.
- Хотим обобщить неограниченные грамматики Хомского на графы. Правило вида $L \to R$ должно мочь заменить произвольный подграф вида L графа H на R.
- Правило — тройка: $(L \xleftarrow{\varphi_L} I \xrightarrow{\varphi_R} R)$, где φ_L, φ_R — морфизмы.
- Применение правила:

$$\begin{tikzcd}
L \arrow[d, "m"] & I \arrow[l, "\varphi_L"'] \arrow[r, "\varphi_R"] \arrow[d, "\psi"] & R \arrow[d, "n"] \\
G & C \arrow[l, "\eta_L"'] \arrow[r, "\eta_R"] & H
\end{tikzcd}$$

- Грубо говоря, G должен получаться склеиванием L и C по общей части I; аналогично, H должен получаться склеиванием R и C по общей части I.

Теоретико-категорный подход к графовым грамматикам

- Пусть дана категория Graph, где объекты — графы (у которых метки есть и на вершинах, и на рёбрах), а стрелки — морфизмы графов.
- Хотим обобщить неограниченные грамматики Хомского на графы. Правило вида $L \to R$ должно мочь заменить произвольный подграф вида L графа H на R.
- Правило — тройка: $(L \xleftarrow{\varphi_L} I \xrightarrow{\varphi_R} R)$, где φ_L, φ_R — морфизмы.
- Применение правила:

$$\begin{tikzcd}
L \arrow[d, "m"] & I \arrow[l, "\varphi_L"'] \arrow[r, "\varphi_R"] \arrow[d, "\psi"] & R \arrow[d, "n"] \\
G & C \arrow[l, "\eta_L"'] \arrow[r, "\eta_R"] & H
\end{tikzcd}$$

- Грубо говоря, G должен получаться склеиванием L и C по общей части I; аналогично, H должен получаться склеиванием R и C по общей части I.
- Почему нельзя просто написать правило в виде $L \xrightarrow{\varphi} R$ и рассмотреть одну диаграмму вместо двух?

Pushout

Если дана категория C и стрелки $b : A \to B$, $c : A \to C$, то тройка D, $g : B \to D$, $f : C \to D$ — pushout, если:

$$\begin{tikzcd}
A \arrow[r, "b"] \arrow[d, "c"'] & B \arrow[d, "g"] \arrow[dr, "g'"] & \\
C \arrow[r, "f"] \arrow[dr, "f'"'] & D \arrow[r, dotted, "h"] & D' \\
& & D'
\end{tikzcd}$$

Pushout

Если дана категория C и стрелки $b : A \to B$, $c : A \to C$, то тройка D, $g : B \to D$, $f : C \to D$ — pushout, если:

$$\begin{tikzcd}
A \arrow[r, "b"] \arrow[d, "c"'] & B \arrow[d, "g"] \arrow[dr, "g'"] & \\
C \arrow[r, "f"] \arrow[drr, "f'"'] & D \arrow[dr, dotted, "h"] & \\
& & D'
\end{tikzcd}$$

Получается, D — наименьший объект, куда “вкладываются” и B, и C.

Pushout

Если дана категория C и стрелки $b : A \to B$, $c : A \to C$, то тройка D, $g : B \to D$, $f : C \to D$ — pushout, если:

$$\begin{tikzcd}
A \arrow[r, "b"] \arrow[d, "c"'] & B \arrow[d, "g"] \arrow[dr, "g'"] & \\
C \arrow[r, "f"] \arrow[dr, "f'"'] & D \arrow[r, dotted, "h"] & D' \\
& & D'
\end{tikzcd}$$

Получается, D — наименьший объект, куда “вкладываются” и B, и C. В применении правила требуем, чтобы оба квадрата были pushout-ами.

Pushout

Если дана категория C и стрелки $b : A \to B$, $c : A \to C$, то тройка D, $g : B \to D$, $f : C \to D$ — pushout, если:

$$\begin{tikzcd}
A \arrow[r, "b"] \arrow[d, "c"'] & B \arrow[d, "g"] \arrow[dr, "g'"] & \\
C \arrow[r, "f"] \arrow[drr, "f'"'] & D \arrow[dr, dotted, "h"] & \\
& & D'
\end{tikzcd}$$

Получается, D — наименьший объект, куда “вкладываются” и B, и C. В применении правила требуем, чтобы оба квадрата были pushout-ами. Получается подход double pushout.

Pushout

Если дана категория C и стрелки $b : A \to B$, $c : A \to C$, то тройка D, $g : B \to D$, $f : C \to D$ — pushout, если:

$$\begin{tikzcd}
A \arrow[r, "b"] \arrow[d, "c"'] & B \arrow[d, "g"] \arrow[dr, "g'"] & \\
C \arrow[r, "f"] \arrow[drr, "f'"'] & D \arrow[dr, dotted, "h"] & \\
& & D'
\end{tikzcd}$$

Получается, D — наименьший объект, куда “вкладываются” и B, и C.
В применении правила требуем, чтобы оба квадрата были pushout-ами.
Получается подход double pushout.
Существует и подход single pushout, в котором применение правила использует один pushout вместо двух — но тогда следует рассматривать частичные морфизмы.

Направления исследований

- Выразительная сила грамматик.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.
- Разнообразные практические приложения.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.
- Разнообразные практические приложения.
- Fusion grammars — грамматики, вдохновлённые ДНК.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.
- Разнообразные практические приложения.
- Fusion grammars — грамматики, вдохновлённые ДНК.
- Стивен Вольфрам вместе с группой коллег разрабатывает теорию, в которой физический мир описывается как граф, законы его эволюции описываются как преобразования этого графа, а квант времени — один шаг преобразования.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.
- Разнообразные практические приложения.
- Fusion grammars — грамматики, вдохновлённые ДНК.
- Стивен Вольфрам вместе с группой коллег разрабатывает теорию, в которой физический мир описывается как граф, законы его эволюции описываются как преобразования этого графа, а квант времени — один шаг преобразования.

Направления исследований

- Выразительная сила грамматик.
  - Какие ограничения на задаваемые языки? (лемма-насос, теорема Париха)
  - Какие строковые языки могут задавать? (HRG $\leftrightarrow$ deterministic tree-walking transducers)
- Алгоритмическая сложность задачи выводимости. (по умолчанию сложнее, так как неявно проверяется изоморфизм графов)
- Свойства DPO-правил: конфлюентность (свойство Чёрча-Россера), возможность параллельного применения правил.
- Разнообразные практические приложения.
- Fusion grammars — грамматики, вдохновлённые ДНК.
- Стивен Вольфрам вместе с группой коллег разрабатывает теорию, в которой физический мир описывается как граф, законы его эволюции описываются как преобразования этого графа, а квант времени — один шаг преобразования.

Спасибо за внимание!