Ієрархія Чомскі

Матеріал з Вікіпедії — вільної енциклопедії.

Ієра́рхія Чо́мскі, або **Ієра́рхія Чо́мскі-Шутценбе́ргера** (названа на честь мовознавця Ноама Чомскі та математика Марселя Шутценберґера) — поняття в теоретичній інформатиці, яким позначають ієрархію формальних граматик, які породжують формальні мови. Вперше описана Ноамом Чомскі в 1956 році. Чотири описані Чомскі типи граматик виходять від базової, необмеженої граматики (граматика типу 0), на яку послідовно накладають обмеження на правила продукції.

В залежності від типу найпростішої граматики, яка може згенерувати задану формальну мову, формальні мови ділять на відповідні категорії від типу 0 до типу 3.

Зміст

Вступ

Формальні мови відіграють в інформатиці важливу роль. Завдяки ним, інформацію можна представляти у вигляді, придатному для автоматичного аналізу та обробки на комп'ютері. Формальні мови в дечому подібні до природних. Вони складаються з множини так званих термінальних символів (їм відповідають склади в природних мовах), які можна складати в слова. Крім того, задані правила (їх називають граматикою), які вказують способи побудови виразів на основі слів (вирази аналогічні реченням природної мови). Наведені приклади не свідчать про повну ідентичність, а лише вказують на основну ідею поняття. Серед прикладів застосування формальних мов можна назвати мови програмування або мови розмітки даних, наприклад, XML або HTML. Слід звернути увагу на те, що формальні мови визначають лише спосіб представлення даних (наприклад, те, що XML-дані побудовані з вкладених тегів), але не самі дані.

Ідея формальних мов полягає в створенні точного математичного описання для подібних мов. На основі цього опису можна будувати засоби для автоматичної обробки цих мов (так звані синтаксичні аналізатори). В попередньому абзаці вже названо основні складові формальної мови. Також слід додати множину основних символів — алфавіт. Алфавіт зазвичай позначають **Σ**, і може складатись, наприклад, зі звичних літер, цифр тощо. Ще однією складовою формальної мови є граматика. Граматики задають правилами підстановки виду $\alpha \rightarrow \beta$. За цим правилом можна замінити дійсне слово позначене як α на слово позначене як β та отримати дійсне слово мови. На додачу, задають початковий символ, часто як аксіому. Початковий символ, за визначенням, є словом з мови. Також слід відрізняти термінальні та нетермінальні символи. Строго кажучи, слова та речення формальної мови можуть складатись лише з термінальних символів (наприклад, літери алфавіту). Нетермінальні символи (також називають *змінними*) відповідають, певним чином, деякому поняттю. В наступному прикладі описано мову, яка породжує математичні вирази (суми). Термінальні символи підкреслені, а нетермінальні виділено курсивом. Кожен рядок визначає одне правило.

1. *Сума* → *Цифра*
2. *Сума* → *Сума* + *Цифра*
3. *Сума* → *Сума* - *Цифра*
4. *Цифра* → 1..9

Аксіомою в цій мові є *Сума*. Виходячи з неї, застосуванням наведених вище правил, можна отримати необхідний вираз:

Сума	Аксіома
Сума − *Цифра*	Застосовано 3 правило
Сума + *Цифра* − *Цифра*	Застосовано 2 правило
Цифра + *Цифра* − *Цифра*	Застосовано 1 правило
1 + *Цифра* − *Цифра*	Застосовано 4 правило
1 + 2 − *Цифра*	Застосовано 4 правило
1 + 2 − 9	Застосовано 4 правило

Порядок застосування правил довільний. Граматика визначає лише правила, які дозволено використовувати у поточній ситуації, і не дає вказівки, які правила мають бути застосовані. Наприклад, до першого речення можна застосувати правила 1—3, та не можна застосувати 4 правило. Починаючи з 4 речення можливе застосування лише 4 правила.

Формальні граматики дозволяють описувати складніші формальні мови. Наприклад, синтаксис мови програмування Паскаль визначено у вигляді формальної граматики.[1]

Ієрархія Чомскі намагається класифікувати необмежені, в принципі, мови, символьні вирази та правила. Для цього запроваджують певні класи мов, для яких можливо встановити швидкодію та підхід до комп'ютерної обробки. Крім того, мови тим більше обмежені, чим глибше знаходяться в ієрархії. Взагалі, можна помітити, що простіші мови, з одного боку, простіше піддаються комп'ютерній обробці, а з іншого — їм бракує виразності. Наприклад, для пошуку деяких шаблонів у тексті використовують регулярні вирази. Регулярні вирази відповідають граматикам Чомскі типу 3. Просто їхньої виразності досить, аби шукати різноманітні фрагменти тексту, але їх не достатньо, для, наприклад, описання мов програмування. Для описання мов програмування використовують, зазвичай, граматики типу 2.

Огляд

Далі позначатимемо формальну граматику $G = (N, \Sigma, P, S)$. Так, N означатиме множину нетермінальних символів, Σ множину термінальних символів, P множину правил виводу та S початковий символ.

В наступній таблиці наведено огляд чотирьої типів формальних граматик, правил виводу, формальних мов та автомати, здатні її розпізнати.

Граматика	Правила	Мови	Автомати	Скорочення
Тип-0 Довільна формальна граматика	α → β α ∈ V^* *N* V^*, β ∈ V^*	рекурсивно зліченна	Машина Тюринга	KSV*
Тип-1 Контекстно-залежна граматика	*α A β* → *α'**γ**'β* *A* ∈ *N*, *α*, *β* ∈ V^*, *γ* ∈ V^+ S → *ε* дозволене, коли серед правил *P* відсутнє *α* → *β* S*γ*.	контекстно-залежна	Лінійний обмежений автомат	КЗ
Тип-2 Контекстно-вільна граматика	*A* → *γ* *A* ∈ *N*, *γ* ∈ V^*	контекстно-вільна	недетермінований автомат з магазинною пам'ятю	КВ
Тип-3 регулярна граматика	*S* → ε *A* → *aB* (праволінійна) або *A* → *Ba* (ліволінійна) *A* → *a* *A* → ε *A*, *B* ∈ *N*, *a* ∈ Σ	регулярна	Скінченний автомат (як детермінований, так і недетермінований)	А

Позначення множин

- Σ: множина термінальних символів
- *N*: множина нетермінальних символів
- *V*=Σ ∪ *N*: словник граматики (множина всіх термінальних та нетермінальних символів)

Операції

- *C* = Доповнення множин
- *K* = Конкатенація формальних мов
- *S* = Перетин множин
- *V* = Об'єднання множин
- * = Зірочка Кліні

Ієрархія

Граматика типу 0

Визначення

Граматики типу 0 називають *необмеженими.* До них належать всі формальні граматики виду $G = (\Sigma, N, S, P)$, де Σ — це скінченний алфавіт, N — множину нетермінальних символів, $S \in N$ — початковий символ, а P — множину правил виведення P: $((\Sigma \cup N)^* N(\Sigma \cup N)^*)) \times (\Sigma \cup N)^*$.

Записують $G \in Tun_0$.

Мови породжені граматиками типу 0

Кожна граматика типу 0 породжує мову, яку може розпізнати машина Тюринга, та навпаки: для кожної мови, яку може розпізнати машина Тюринга, існує граматика типу 0, яка здатна її породити. Такі мови також відомі, як рекурсивно зліченні мови.

Слід, однак, відрізняти цю множину мов від множини рекурсивних мов.

Граматики типу 1

Визначення

Граматики типу 1 ще називають *контекстно-залежними*. До них належать всі граматики типу 0, в яких правила виводу обмежено правилами вигляду $\alpha A \beta \to \alpha \gamma \beta$ або $S \to \varepsilon$, де A — нетермінальний, а α, γ, β слова з термінальних (Σ) та нетермінальних (N) символів. Слова α und β можуть бути порожніми, але γ має містити щонайменше один символ (термінальний або нетермінальний).

Правило $S \to \varepsilon$ дозволене, якщо S не зустрічається в правій частині жодного з правил. Це правило необіхдне для додавання порожнього слова ε.

Якщо граматика G контекстно-вільна, записують $G \in Tun_1$.

На відміну від контекстно-незалежних граматик, кількість символів у лівій частині правила може бути > 1.

Мови породжені граматиками типу 1

Контекстно-залежні (КЗ) граматики породжують контекстно-залежні мови; тобто, кожна КЗ-граматика породжує КЗ-мову, і навпаки — для кожної КЗ мови існує КЗ граматика, що її породжує.

Контекстно-залежні мови можна розпізнати недетермінованою лінійно-обмеженою машиною Тюринга; тобто, недетермінованою машиною Тюринга, довжина стрічки якої обмежена.

Монотонні граматики

Якщо за винятком правила $S \to \varepsilon$ в правилі $\alpha \to \beta$ довжина α не більша за довжину β, $|\alpha| \leq |\beta|$ то таку КЗ-граматику називають *монотонною*.

Монотонні граматики також породжують КЗ-мови, однак не кожна монотонна граматика контекстно-залежна.

Граматики типу 2 (контекстно-вільні)

Визначення

Граматики типу 2 також називають *контекстно-вільними* (КВ).

В кожному правилі граматики другого типу з лівої сторони знаходиться один нетермінальний символ, а з правої сторони — можливо порожня послідовність термінальних та нетермінальних символів. Тобто, правила виводу обмежені:

$$\forall (w_1 \to w_2) \in P : (w_1 \in N) \land (w_2 \in (N \cup \Sigma)^*) .$$

Граматики типу 2 мають лише правила виду $A \to \gamma$, A — нетермінальний символ, а γ складається з послідовності тремінальних та нетермінальних символів.

Записують $G \in Tun_2$.

Мови породжені граматиками типу 2

Контекстно-вільні (КВ) граматики породжують КВ-мови, і для кожної КВ-мови існує КВ-граматика, що її породжує.

Контекстно-вільні мови розпізнаються недетермінованими автоматами з магазинною пам'ятю. КВ-мови відіграють важливу роль в теорії та побудові синтаксису мов програмування.

Граматики типу 3 (право- та ліво- лінійні граматики)

Визначення

Граматики третього типу називають *регулярними*. До них належать граматики другого типу, правила виводу яких обмежено $\forall (w_1 \to w_2) \in P : (w_1 \in N) \wedge (w_2 \in \Sigma \cup (\Sigma \cdot N \dot{\vee} N \cdot \Sigma) \cup \{\varepsilon\})$. Тобто, в лівій частині правила виводу зхнаходиться один нетермінальний символ. В правій частині *праволінійні* граматики мають один термінальний за ним, можливо, один нетермінальний. Так звані *ліволінійні* граматики мають в правій частині правила один нетермінальний символ, за яким, можливо, один термінальний.

Таким чином, граматики третього типу мають лише правила, в яких ліва частина складається з одного нетермінального символу, а права, з одного термінального слід за яким, можливо, йде термінальний (або навпаки, термінальний, за яким, можливо, нетермінальний).

Для кожної ліволінійної граматики існує праволінійна граматика, та навпаки, які генерують однакові мови.

Записують $G \in Tun_3$.

Мови породжені граматиками типу 3

Регулярні граматики породжують регулярні мови, і для кожної регулярної мови існує регулярна граматика, що її породжує.

Регулярні мови можна описувати також регулярними виразами. Регулярні мови можна розпізнаюти скінченними автоматами. Їх часто використовують для пошуку фрагментів тексту, або для визначення лексичної структури мови програмування.

Ієрархія Чомскі формальних мов

Формальна мова належить до типу *i*, якщо її породжує граматика типу *i*. Формально, мова *L* належить до типу $i \in \{0, \ldots, 3\}$ якщо існує граматика $G \in Tun_i$ така, що $L = L(G)$. Тоді пишуть $\boldsymbol{L \in \mathcal{L}_i}$.

В ієрархії Чомскі, множина мов типу *i* є підмножиною мов типу $i - 1$. Кожна контекстно-залежна мова рекурсивно зліченна, але існують рекурсивно зліченні мови, які не є контекстно-залежними. Так само кожна контекстно-вільна мова є контекстно-залежною, але не навпаки, та кожна регулярна мова контекстно-вільна але не кожна контекстно-вільна мова регулярна.

Класи формальних мов мають відношення $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$.

Серед прикладів мов різних класів можна назвати:

- Проблема зупинки належить до типу 0, але не до типу 1.
- $L_1 = \{a^n b^n c^n | n \geq 1\}$ належить до типу 1, але не 2.
- $L_2 = \{a^n b^n | n \geq 1\}$ належить до типу 2, але не 3.

Природні мови

Хоча дослідження формальних граматик Чомскі збирався використати для створення математичного описання природньої мови, досі вдалось розробити формальні граматики для декількох мов, в першу чергу, штучних. Проблема полягає, зокрема, в багатозначності та слів природньої мови. Правильне значення можна отримати шляхом аналізу *розширеного контексту*, в якому знаходиться аналізоване речення, або його значення взагалі неможливо встановити.

Примітки

1. ANSI-ISO-Pascal (http://www.moorecad.com/standardpascal/iso7185.html)

Література

- *Noam Chomsky.* [PDF Three models for the description of language]. — 1956. — Т. Vol.2. — С. 113–124. — (IRE Transactions on Information Theory)
- *Noam Chomsky.* On certain formal properties of grammars. — 1959. — Т. Vol.2. — С. 137–167. — (Information and Control)
- *Noam Chomsky, Marcel P. Schützenberger.* The algebraic theory of context free languages, Computer Programming and Formal Languages / P. Braffort, D. Hirschberg. — Amsterdam, 1963. — С. 118–161.
- *Sander, Stucky, Herschel.* Automaten, Sprachen, Berechenbarkeit. — Stuttgart : Teubner, 1992. — ISBN 3-519-02937-5.

Див. також

- Математична лінгвістика
- Синтаксичний аналіз
