Azzolini Riccardo 2020-12-11

Gerarchia di Chomsky

1 Definizione generale di grammatica

Le grammatiche context-free sono un caso particolare della definizione generale di grammatica.

In generale, una **grammatica** è una quadrupla $G = \langle V, T, P, S \rangle$ in cui:

- V è l'insieme finito dei *simboli non-terminali*;
- T è l'insieme finito dei *simboli terminali*;
- P è l'insieme finito delle *regole di produzione*, che sono coppie $\alpha \to \beta$ dove:
  - $\alpha \in (T \cup V)^+$, la testa della produzione, è una sequenza non vuota di simboli terminali e non-terminali;
  - $\beta \in (T \cup V)^*$, il corpo della produzione, è una sequenza (possibilmente vuota) di simboli terminali e non-terminali;
- $S \in V$ è il *simbolo iniziale* della grammatica.

L'unica cosa che cambia rispetto alla definizione di CFG è la forma delle regole di produzione: mentre nelle CFG la testa doveva essere un simbolo non-terminale, qui può essere un'arbitraria stringa non vuota sull'insieme dei terminali e non-terminali della grammatica.

1.1 Derivazioni e linguaggio generato

Data una grammatica $G = \langle V, T, P, S \rangle$,

- Si definisce la *relazione di derivazione* $\Rightarrow_G$ nel modo seguente: date $\alpha, \beta, \gamma, \delta \in (T \cup V)^*$, con $\alpha \neq \epsilon$, si ha $\gamma\alpha\delta \Rightarrow_G \gamma\beta\delta$ se $\alpha \to \beta$ è una produzione in P.

  In pratica, un passo di derivazione consiste nella sostituzione di una stringa con un'altra secondo una regola di produzione. Questa è una generalizzazione del passo di derivazione per le CFG, nel quale si sostituiva un singolo simbolo non-terminale con una stringa.

- La relazione di *derivazione in zero o più passi*, $\overset{*}{\Rightarrow}_G$, è la chiusura riflessiva e transitiva di $\Rightarrow_G$ (esattamente come nel caso delle CFG).

- Il linguaggio generato da G è l'insieme di stringhe terminali derivabili in zero o più passi dal simboli iniziale di G,

$$L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow}_G w\}$$

(esattamente come nel caso delle CFG).

1.2 Esempio

Un esempio di grammatica secondo la definizione generale è

$$G = \langle \{S, A, B, C\}, \{0, 1, 2\}, P, S \rangle$$

dove P contiene le seguenti regole di produzione:

(1)	$S \to 0SBC$	(4)	$0B \to 01$
(2)	$S \to 0BC$	(5)	$1B \to 11$
(3)	$CB \to BC$	(6)	$1C \to 12$
		(7)	$2C \to 22$

Siccome le regole (3)–(7) hanno come testa delle stringhe di più simboli (terminali e non-terminali), questa grammatica non è context-free.

Un esempio di derivazione su G è:

$\boldsymbol{S} \Rightarrow \mathbf{0}\boldsymbol{SBC}$	Regola (1)	$S \to 0SBC$
$0\boldsymbol{S}BC \Rightarrow 0\mathbf{0}\boldsymbol{BC}BC$	(2)	$S \to 0BC$
$00B\boldsymbol{CB}C \Rightarrow 00B\boldsymbol{BC}C$	(3)	$CB \to BC$
$0\mathbf{0}\boldsymbol{B}BCC \Rightarrow 0\mathbf{01}BCC$	(4)	$0B \to 01$
$00\mathbf{1}\boldsymbol{B}CC \Rightarrow 00\mathbf{11}CC$	(5)	$1B \to 11$
$001\mathbf{1}\boldsymbol{C}C \Rightarrow 001\mathbf{12}C$	(6)	$1C \to 12$
$0011\mathbf{2}\boldsymbol{C} \Rightarrow 0011\mathbf{22}$	(7)	$2C \to 22$

(qui il risultato di ogni passo è riportato nuovamente a sinistra del passo successivo, in modo da poter evidenziare, in grassetto, sia la stringa a cui è applicata ciascuna regola che la stringa generata da tale applicazione).

Si può dimostrare che il linguaggio generato da questa grammatica G è

$$L(G) = \{0^n 1^n 2^n \mid n \geq 1\}$$

che, come dimostrato in precedenza, non è context-free: esso rientra invece nella classe dei *linguaggi context-sensitive*, che — come si vedrà in seguito — sono generati dalle *grammatiche context-sensitive*.

Come suggeriscono i nomi, la differenza tra le grammatiche context-free e quelle context-sensitive sta nei contesti in cui è possibile applicare le regole di produzione:

- Una regole di produzione di una CFG ha come testa un singolo simbolo non-terminale, dunque è applicabile *indipendentemente dal contesto* in cui tale simbolo compare.
- Nelle grammatiche context-sensitive, invece, l'applicazione di una regola può essere vincolata ai casi in cui un simbolo compare in un determinato contesto. Ad esempio, la grammatica G ha 3 regole per il non-terminale B,

$$(3) \quad CB \to BC$$
$$(4) \quad 0B \to 01$$
$$(5) \quad 1B \to 11$$

  ma queste sono applicabili solo in contesti diversi (rispettivamente quando il simbolo B è preceduto da C, da 0 o da 1).

2 Gerarchia di Chomsky

La **gerarchia di Chomsky** è una gerarchia di tipi di grammatiche, che a partire dalla definizione generale impone vincoli via via sempre più forti sulla forma delle regole di produzione, riducendo così la classe dei linguaggi generati. I livelli della gerarchia sono numerati da 0 a 3: numeri più grandi corrispondono a grammatiche più ristrette.

- Le **grammatiche di tipo 0** sono le grammatiche nella loro definizione generale, che generano i **linguaggi ricorsivamente enumerabili**, riconosciuti dalle **macchine di Turing**.
- Le **grammatiche di tipo 1**, dette **dipendenti dal contesto** o **context-sensitive**, hanno produzioni $\alpha \to \beta$ (con $\alpha \in (T \cup V)^+$ e $\beta \in (T \cup V)^*$, come nella definizione generale) tali che $|\alpha| \leq |\beta|$ (intuitivamente, ciò significa che l'applicazione di una regola non accorcia mai la stringa che si sta derivando). Esse generano i **linguaggi context-sensitive**, riconosciuti dalle **macchine di Turing linearmente limitate** (cioè macchine di Turing con un vincolo sulla quantità di memoria utilizzata).
- Le **grammatiche di tipo 2** sono le CFG, cioè hanno produzione del tipo $A \to \beta$ con $A \in V$ (e ancora $\beta \in (T \cup V)^*$). Esse generano i **linguaggi context-free**, riconosciuti dagli **automi a pila**.
- Le **grammatiche di tipo 3**, dette **regolari**, sono quelle in cui ciascuna regola di produzione ha la forma $A \to aB$ oppure $A \to a$, con $A, B \in V$ e $a \in T$: la testa è ancora un simbolo non-terminale, come nelle CFG, ma il corpo può essere solo un simbolo terminale opzionalmente seguito da un non-terminale. Queste grammatiche generano la classe dei **linguaggi regolari**, per i quali costituiscono dunque un generatore alternativo alle espressioni regolari. Come già visto, i riconoscitori per i linguaggi regolari sono i vari tipi di **automi a stati finiti**: DFA, NFA e ϵ-NFA.

Siccome a ogni livello vengono solo aggiunte restrizioni in più, le grammatiche di tipo 3 sono un caso particolare di quelle di tipo 2, che sono a loro volta un caso particolare di quelle di tipo 1, e infine queste sono un caso particolare delle grammatiche di tipo 0. Di conseguenza, la classe dei linguaggi regolari è un sottoinsieme proprio di quella dei linguaggi context-free, e così via per le classi associate ai livelli inferiori.