Linguaggi Formali e Traduttori

4.1 Parsing top-down e grammatiche LL(1)

- Sommario
- Strategia per il parsing top-down
- Stringhe annullabili (NULL)
- Esempi di stringhe annullabili
- Inizi di una stringa (FIRST)
- Come calcolare FIRST
- Esempi di calcolo di FIRST
- Seguiti di una variabile (FOLLOW)
- Come calcolare FOLLOW
- Esempi di calcolo di FOLLOW
- Insiemi guida
- Grammatiche LL(1)
- Esempio: espressioni aritmetiche
- Esercizi

Sommario

Problema

- Data una grammatica $G = (V, T, P, S)$ e una stringa $w \in T^*$, determinare se

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow w$$

  o, equivalentemente, se esiste un albero sintattico di G con radice S e prodotto w.

- La costruzione dell'automa corrispondente a G produce un PDA non deterministico.
- Per alcune G sappiamo che non è possibile trovare un DPDA.

In questa lezione

- Identifichiamo una famiglia di grammatiche libere per le quali è possibile costruire riconoscitori (parser) deterministici, cioè che non fanno uso di backtracking.
- Questi parser sono detti **top-down** perché costruiscono l'albero sintattico di w dalla radice (top) verso le foglie (down) o, equivalentemente, cercano una derivazione sinistra per w.

Strategia per il parsing top-down

Data una grammatica $G = (V, T, P, S)$ e una stringa $w \in T^*$, il parser cerca di ottenere una derivazione a sinistra $S \Rightarrow_{lm}^* w$ in cui, al passo i, il parser sa che

$$S \Rightarrow_{lm}^* uA\beta$$

e deve stabilire se

$$uA\beta \Rightarrow_{lm}^* w$$

Ci sono due casi da considerare:

- Se u **non è** prefisso di w, allora il parser **rifiuta** w.
- Se $w = uav$, allora il parser deve **scegliere** una produzione per riscrivere A

$$A \to \alpha_1 \mid \cdots \mid \alpha_n$$

  e per farlo può usare a come "guida", a patto che tale simbolo identifichi univocamente l'α_i tale che $\alpha_i\beta \Rightarrow_{lm}^* av$.

Per ogni produzione $A \to \alpha_i$ occorre saper calcolare gli insiemi di simboli terminali che possono **iniziare** le stringhe derivate da $\alpha_i\beta$ e richiedere che tali insiemi siano disgiunti.

Stringhe annullabili (NULL)

Definizione

Data una grammatica $G = (V, T, P, S)$, diciamo che $\alpha \in (V \cup T)^*$ è **annullabile**, e scriviamo $\text{NULL}(\alpha)$, se e solo se $\alpha \Rightarrow_G^* \varepsilon$, ovvero se α può essere riscritta nella stringa vuota.

Come determinare se una stringa è annullabile

(1) Se $\text{NULL}(X_1), \dots, \text{NULL}(X_n)$, allora $\text{NULL}(X_1 \cdots X_n)$.
(2) Se esiste una produzione $A \to \alpha \in P$ e $\text{NULL}(\alpha)$, allora $\text{NULL}(A)$.

Note

- Come caso particolare di (1) quando $n = 0$ abbiamo $\text{NULL}(\varepsilon)$.
- Combinando (1) e (2) abbiamo che $A \to \varepsilon \in P$ implica $\text{NULL}(A)$.
- Una stringa che contiene simboli terminali <u>non è mai annullabile</u>.

Esempi di stringhe annullabili

$$
\begin{aligned}
A &\rightarrow a \mid Bc \\
B &\rightarrow \varepsilon \mid bB \\
C &\rightarrow d \mid Cc \mid BB
\end{aligned}
$$

Esempi di stringhe annullabili

$$\begin{array}{lcl} A & \rightarrow & a \mid Bc \\ B & \rightarrow & \varepsilon \mid bB \\ C & \rightarrow & d \mid Cc \mid BB \end{array}$$

- Da $\textsc{null}(\varepsilon)$ e dalla produzione $B \rightarrow \varepsilon$ deduciamo $\textsc{null}(B)$.
- Da $\textsc{null}(B)$ e dalla produzione $C \rightarrow BB$ deduciamo $\textsc{null}(C)$.
- Da $\textsc{null}(B)$ e $\textsc{null}(C)$ deduciamo $\textsc{null}(BC)$.
- Da $\neg\textsc{null}(a)$ e $\neg\textsc{null}(Bc)$ deduciamo $\neg\textsc{null}(A)$.

Inizi di una stringa (FIRST)

Definizione

Data una grammatica $G = (V, T, P, S)$ e una stringa $\alpha \in (V \cup T)^*$, indichiamo con $\text{FIRST}(\alpha)$ gli **inizi** di α, ovvero l'insieme dei simboli terminali che possono trovarsi all'<u>inizio</u> delle stringhe derivate da α. Formalmente:

$$\text{FIRST}(\alpha) \overset{\text{def}}{=} \{a \in T \mid \alpha \Rightarrow^*_G a\beta\}$$

Attenzione

Il libro di testo usa un'unica funzione FIRST_{libro} che <u>combina</u> NULL e FIRST così:

$$\text{FIRST}_{libro}(\alpha) = \begin{cases} \text{FIRST}(\alpha) \cup \{\varepsilon\} & \text{se } \text{NULL}(\alpha) \\ \text{FIRST}(\alpha) & \text{altrimenti} \end{cases}$$

In pratica, l'approccio seguito dal libro ammette il simbolo speciale ε tra gli inizi di α per indicare il fatto che α è annullabile. Noi abbiamo definito un predicato $\text{NULL}(\alpha)$ apposito mentre $\text{FIRST}(\alpha)$ contiene solo simboli terminali.

Come calcolare FIRST

È possibile calcolare $\textsc{first}(\alpha)$ per induzione su α, usando le seguenti regole:

$$
\begin{array}{rcl}
\textsc{first}(\varepsilon) & = & \emptyset \\
\textsc{first}(a) & = & \{a\} \\
\textsc{first}(A) & = & \bigcup_{A \to \alpha} \textsc{first}(\alpha) \\
\textsc{first}(X\alpha) & = & \begin{cases} \textsc{first}(X) \cup \textsc{first}(\alpha) & \textbf{se } \textsc{null}(X) \\ \textsc{first}(X) & \textbf{altrimenti} \end{cases}
\end{array}
$$

Attenzione

Applicando le regole qui sopra, può capitare di arrivare a equazioni della forma

$$\textsc{first}(A) = \textsc{first}(A) \cup \mathcal{S}$$

dove $\mathcal{S}$ è un insieme di terminali. Questa equazione si può semplificare a

$$\textsc{first}(A) = \mathcal{S}$$

in quanto siamo interessati a ottenere il più piccolo insieme di terminali con la proprietà descritta nella slide precedente.

Esempi di calcolo di FIRST

$$
\begin{aligned}
S &\rightarrow Ac \mid Ba \\
A &\rightarrow \varepsilon \mid a \\
B &\rightarrow b \\
C &\rightarrow a \mid Cb \\
D &\rightarrow \varepsilon \mid d \mid Db
\end{aligned}
$$

Esempi di calcolo di FIRST

$$\begin{array}{rcl} S & \rightarrow & Ac \mid Ba \\ A & \rightarrow & \varepsilon \mid a \\ B & \rightarrow & b \\ C & \rightarrow & a \mid Cb \\ D & \rightarrow & \varepsilon \mid d \mid Db \end{array}$$

Variabili annullabili

- $\text{NULL}(A)$
- $\text{NULL}(D)$

Calcolo di FIRST di tutte le variabili

- $\text{FIRST}(B) = \text{FIRST}(b) = \{b\}$
- $\text{FIRST}(A) = \text{FIRST}(\varepsilon) \cup \text{FIRST}(a) = \{a\}$
- $\text{FIRST}(S) = \text{FIRST}(Ac) \cup \text{FIRST}(Ba) = \text{FIRST}(A) \cup \text{FIRST}(c) \cup \text{FIRST}(B) = \{a, b, c\}$
- $\text{FIRST}(C) = \text{FIRST}(a) \cup \text{FIRST}(Cb) = \{a\} \cup \text{FIRST}(C) = \{a\}$
- $\text{FIRST}(D) = \text{FIRST}(\varepsilon) \cup \text{FIRST}(d) \cup \text{FIRST}(Db) = \{d\} \cup \text{FIRST}(D) \cup \text{FIRST}(b) = \{b, d\}$

Seguiti di una variabile (FOLLOW)

Definizione

Data una grammatica $G = (V, T, P, S)$ e una variabile $A \in V$, indichiamo con $\textsc{follow}(A)$ i **seguiti** di A, ovvero l'insieme dei simboli terminali che possono <u>seguire</u> A in una forma sentenziale. Formalmente:

$$\textsc{follow}(A) \stackrel{\text{def}}{=} \{a \in T \mid S \Rightarrow_G^* \alpha A a \beta\}$$

Attenzione

- Per convenzione aggiungeremo una sentinella **$** ai seguiti del simbolo iniziale S.
- In questo modo il parser può capire quando è arrivato alla fine della stringa da riconoscere.

Come calcolare FOLLOW

Il calcolo di $\textsc{follow}$ si effettua in due fasi.

Fase 1

In questa fase si annotano relazioni di appartenenza ed inclusione insiemistica secondo il seguente algoritmo:

- Annotare $\$ \in \textsc{follow}(S)$.
- Ripetere i passi seguenti <u>per ogni produzione</u> e <u>per ogni variabile</u> nel corpo di queste:
    1. Se $A \to \alpha B\beta$, allora annotare $\textsc{first}(\beta) \subseteq \textsc{follow}(B)$.
    2. Se $A \to \alpha B\beta$ e $\textsc{null}(\beta)$, allora annotare $\textsc{follow}(A) \subseteq \textsc{follow}(B)$.

<u>Caso particolare di (2)</u>: se $A \to \alpha B$, allora annotare $\textsc{follow}(A) \subseteq \textsc{follow}(B)$.

Fase 2

Si determinano i seguiti propagando i simboli terminali (e $\$$) rispettando l'ordine delle inclusioni insiemistiche $\subseteq$ che sono state annotate.

Per grammatiche complesse può essere utile fare una tabella con due colonne, l'elenco di tutte le variabili nella prima ed i seguiti corrispondenti alle variabili nella seconda.

Esempi di calcolo di FOLLOW

$$
\begin{array}{lcl}
S & \rightarrow & Ac \mid Ba \\
A & \rightarrow & \varepsilon \mid a \\
B & \rightarrow & b \\
C & \rightarrow & a \mid Cb \\
D & \rightarrow & \varepsilon \mid d \mid Db
\end{array}
$$

Esempi di calcolo di FOLLOW

$$\begin{array}{lcl} S & \to & Ac \mid Ba \\ A & \to & \varepsilon \mid a \\ B & \to & b \\ C & \to & a \mid Cb \\ D & \to & \varepsilon \mid d \mid Db \end{array}$$

Fase 1

- $\$ \in \textsc{follow}(S)$
- $\textsc{first}(c) \subseteq \textsc{follow}(A)$
- $\textsc{first}(a) \subseteq \textsc{follow}(B)$
- $\textsc{first}(b) \subseteq \textsc{follow}(C)$
- $\textsc{first}(b) \subseteq \textsc{follow}(D)$

Fase 2

X	$\textsc{follow}(X)$
S	$\{\$\}$
A	$\{c\}$
B	$\{a\}$
C	$\{b\}$
D	$\{b\}$

Insiemi guida

Definizione

Data una grammatica $G = (V, T, P, S)$ e una produzione $A \to \alpha$, indichiamo con $\textsc{guida}(A \to \alpha)$ l'**insieme guida** di $A \to \alpha$, ovvero l'insieme

$$\textsc{guida}(A \to \alpha) \stackrel{\text{def}}{=} \begin{cases} \text{FIRST}(\alpha) \cup \text{FOLLOW}(A) & \text{se } \text{NULL}(\alpha) \\ \text{FIRST}(\alpha) & \text{altrimenti} \end{cases}$$

Intuizione

Un parser predittivo che sceglie di riscrivere la variabile A usando la produzione $A \to \alpha$ si aspetta di leggere nella stringa di input uno dei simboli nell'insieme guida di $A \to \alpha$.

Sono due i casi da considerare:

1. Il simbolo è uno degli inizi di α, oppure
2. α è annullabile ed il simbolo è uno dei seguiti di A.

Grammatiche LL(1)

Definizione

Diciamo che una grammatica $G = (V, T, P, S)$ è **LL(1)** se, per ogni coppia di produzioni distinte $A \to \alpha$ e $A \to \beta$ in P, abbiamo che

$$\textsc{guida}(A \to \alpha) \cap \textsc{guida}(A \to \beta) = \emptyset$$

Intuizione

Noto il simbolo da riscrivere A, note le produzioni $A \to \beta_1 \mid \cdots \mid \beta_n$ e noto il prossimo simbolo terminale a nella stringa da riconoscere, in una grammatica LL(1) esiste al massimo una produzione "giusta" tale che $a \in \textsc{guida}(A \to \beta_i)$ dunque il parser predittivo <u>identifica univocamente</u> la produzione $A \to \beta_i$ a partire da a.

Cosa c'è nel nome LL(1)

- L $\longrightarrow$ la stringa in input viene analizzata <u>da sinistra (left) a destra</u>;
- L $\longrightarrow$ il parser cerca di costruire una <u>derivazione canonica sinistra</u> (leftmost);
- 1 $\longrightarrow$ il parser usa <u>un solo simbolo terminale</u> della stringa per scegliere la produzione.

Esempio: espressioni aritmetiche

$$
\begin{array}{rcll}
E & \to & TE' & \\
E' & \to & +TE' \mid \varepsilon & \textsc{null}(E') \\
T & \to & FT' & \\
T' & \to & *FT' \mid \varepsilon & \textsc{null}(T') \\
F & \to & (E) \mid \mathtt{id} &
\end{array}
$$

- $\$ \in \textsc{follow}(E)$
- $\{+\} = \textsc{first}(E') \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(E')$
- $\textsc{follow}(E') \subseteq \textsc{follow}(T)$
- $\{*\} = \textsc{first}(T') \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(T')$
- $\textsc{follow}(T') \subseteq \textsc{follow}(F)$
- $\{)\} = \textsc{first}()) \subseteq \textsc{follow}(E)$

Esempio: espressioni aritmetiche

$$
\begin{array}{rcll}
E & \to & TE' & \\
E' & \to & \texttt{+}TE' \mid \varepsilon & \textsc{null}(E') \\
T & \to & FT' & \\
T' & \to & \texttt{*}FT' \mid \varepsilon & \textsc{null}(T') \\
F & \to & \texttt{(}E\texttt{)} \mid \texttt{id} &
\end{array}
$$

$$
\begin{array}{l}
\textsc{first}(E) = \textsc{first}(T) = \{\texttt{(}, \texttt{id}\} \\
\textsc{first}(E') = \{\texttt{+}\} \\
\textsc{first}(T) = \textsc{first}(F) = \{\texttt{(}, \texttt{id}\} \\
\textsc{first}(T') = \{\texttt{*}\} \\
\textsc{first}(F) = \{\texttt{(}, \texttt{id}\}
\end{array}
$$

- $\texttt{\$} \in \textsc{follow}(E)$
- $\{\texttt{+}\} = \textsc{first}(E') \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(E')$
- $\textsc{follow}(E') \subseteq \textsc{follow}(T)$
- $\{\texttt{*}\} = \textsc{first}(T') \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(T')$
- $\textsc{follow}(T') \subseteq \textsc{follow}(F)$
- $\{\texttt{)}\} = \textsc{first}(\texttt{)}) \subseteq \textsc{follow}(E)$

Esempio: espressioni aritmetiche

$$
\begin{array}{rcll}
E & \to & TE' & \\
E' & \to & +TE' \mid \varepsilon & \textsc{null}(E') \\
T & \to & FT' & \\
T' & \to & *FT' \mid \varepsilon & \textsc{null}(T') \\
F & \to & (E) \mid \mathtt{id} &
\end{array}
$$

$$
\begin{aligned}
\textsc{first}(E) &= \textsc{first}(T) = \{(, \mathtt{id}\} \\
\textsc{first}(E') &= \{+\} \\
\textsc{first}(T) &= \textsc{first}(F) = \{(, \mathtt{id}\} \\
\textsc{first}(T') &= \{*\} \\
\textsc{first}(F) &= \{(, \mathtt{id}\}
\end{aligned}
$$

- $\$ \in \textsc{follow}(E)$
- $\{+\} = \textsc{first}(E') \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(T)$
- $\textsc{follow}(E) \subseteq \textsc{follow}(E')$
- $\textsc{follow}(E') \subseteq \textsc{follow}(T)$
- $\{*\} = \textsc{first}(T') \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(F)$
- $\textsc{follow}(T) \subseteq \textsc{follow}(T')$
- $\textsc{follow}(T') \subseteq \textsc{follow}(F)$
- $\{)\} = \textsc{first}()) \subseteq \textsc{follow}(E)$

X	$\textsc{follow}(X)$
E	$,)
E'	$,)
T	$,), +
T'	$,), +
F	$,), +, *

Esercizi

1. Calcolare gli insiemi guida della grammatica nella slide 14. La grammatica è LL(1)?

2. Calcolare gli insiemi guida della seguente grammatica e determinare se è LL(1).

$$\begin{array}{rcl} A & \rightarrow & BC \mid D \\ B & \rightarrow & \varepsilon \mid a \\ C & \rightarrow & b \mid cCc \\ D & \rightarrow & \varepsilon \mid CD \end{array}$$

3. Ripetere l'esercizio precedente per la grammatica

$$\begin{array}{rcl} S & \rightarrow & \texttt{if}\ E\ \texttt{then}\ SS'\ \texttt{fi} \mid \texttt{skip} \\ S' & \rightarrow & \texttt{else}\ S \mid \varepsilon \\ E & \rightarrow & \texttt{true} \mid \texttt{false} \end{array}$$

   in cui S, S' ed E sono variabili e **if**, **then**, ... sono terminali.

4. Ripetere l'esercizio precedente dopo aver rimosso il terminale **fi** dalla grammatica.