Questi lucidi sono basati su una traduzione in italiano dei lucidi in inglese del Prof. Jeffrey D. Ullman

http://infolab.stanford.edu/~ullman/ialc/spr10/spr10.html#LECTURE%20NOTES

http://www-db.Stanford.edu/~Ullman/ialc.html

Automi a pila (automi pushdown - PDA)

Definizione

Azioni di un PDA

Il linguaggio di un PDA

Automi a pila deterministici

Automi a pila

- Un PDA è un automa che ha lo stesso potere computazionale delle CFG.
- Solo i PDA non deterministici definiscono tutti i CFL.
- Ma la versione deterministica è il modello per i parser.
  - La maggior parte dei linguaggi di programmazione ha PDA deterministici.

Intuizione: PDA

- Bisogna pensare a un NFA con il potere aggiuntivo di poter manipolare una pila (o stack).
- Le sue azioni sono determinate da:
  1. Lo stato in cui si trova (del suo “NFA”),
  2. Il simbolo input letto (o ϵ), e
  3. Il simbolo letto sul top della sua pila.

Intuizione: PDA – (2)

Essendo non deterministico, un PDA

- Può scegliere l’azione successiva in un insieme di possibili scelte.
- Per ciascuna scelta, il PDA può:
  1. Cambiare stato, e anche
  2. Cambiare il simbolo sul top della pila con una sequenza di zero o più simboli.
     - Zero simboli = “pop.”
     - Più simboli = sequenza di “push.”

Definizione formale di PDA

- Un PDA è descritto da:
  1. Un insieme finito di *stati* (Q, generalmente).
  2. Un *alfabeto di simboli input* (Σ, generalmente).
  3. Un *alfabeto di simboli di stack* (Γ, generalmente).
  4. Una *funzione di transizione* (δ, generalmente).
  5. Uno *stato iniziale* (q_0, in Q, generalmente).
  6. Un *simbolo iniziale* (Z_0, in Γ, generalmente).
  7. Un insieme di *stati finali* ($F \subseteq Q$, in genere).

Convenzioni

- a, b, … sono simboli di input.
  - Ma a volte permettiamo che ϵ sia un possibile valore.
- …, X, Y, Z sono simboli di pila.
- …, w, x, y, z sono stringhe di simboli input.
- α, β,… sono stringhe di simboli di pila.

La Funzione di Transizione

- Dipende da tre argomenti:
  1. Uno stato, in Q.
  2. Un input, che è o un simbolo in Σ o ϵ.
  3. Un simbolo di stack in Γ.
- $\delta(q, a, Z)$ è un insieme di zero o più coppie della forma (p, α).
  - p è uno stato; α è una stringa di simboli di pila.

Azioni del PDA

- Se $\delta(q, a, Z)$ contiene (p, α) tra le sue possibili coppie, allora quello che il PDA può fare nello stato q, con input *a* e con Z sul top dello stack è:
  1. Cambiare stato, passando da q a p.
  2. ``Cancellare'' *a* dall'input (ma *a* potrebbe essere ϵ).
  3. Rimpiazzare Z sul top dello stack con α.

Esempio: PDA

- Definire un PDA che accetti

$$\{0^n 1^n \mid n \geq 1\}.$$

- Gli stati:
  - q = stato iniziale. Siamo nello stato q se abbiamo visto solo 0.
  - p = abbiamo visto almeno un 1 e ora possiamo procedere solo se i simboli input sono 1.
  - f = stato finale; accettazione (se ha letto tutto l'input).

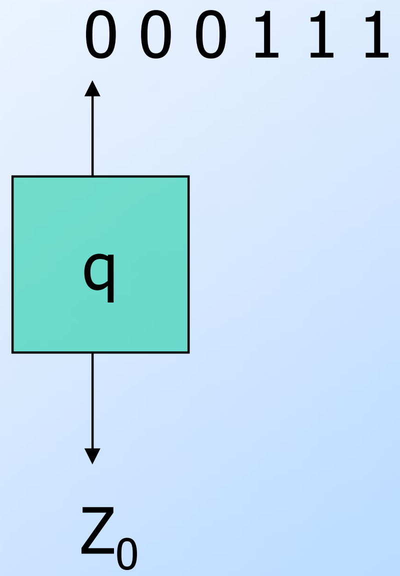
Esempio: PDA – (2)

- I simboli dello stack:
  - Z_0 = simbolo iniziale. Indica anche il fondo della pila, così sappiamo quando avremo contato lo stesso numero di 1 e di 0.
  - X = marcatore, usato per contare il numero di 0 visti nell'input.

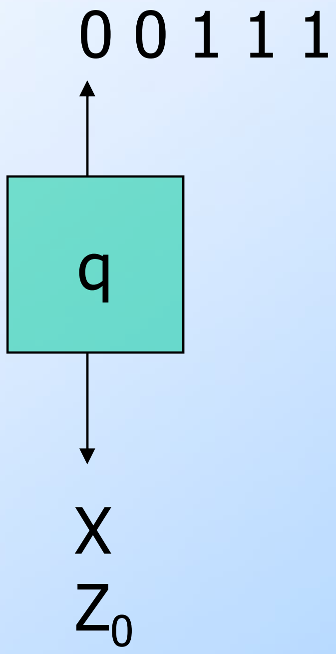
Esempio: PDA – (3)

- Le transizioni:
  - $\delta(q, 0, Z_0) = \{(q, XZ_0)\}$.
  - $\delta(q, 0, X) = \{(q, XX)\}$. Queste due regole permettono l'inserimento di una X nella pila per ogni 0 letto dall'input.
  - $\delta(q, 1, X) = \{(p, \epsilon)\}$. Quando vede un 1, il PDA va nello stato p ed elimina una X.
  - $\delta(p, 1, X) = \{(p, \epsilon)\}$. Elimina una X per ogni 1.
  - $\delta(p, \epsilon, Z_0) = \{(f, Z_0)\}$. Accetta se vede il fondo (e ha letto tutto l'input).

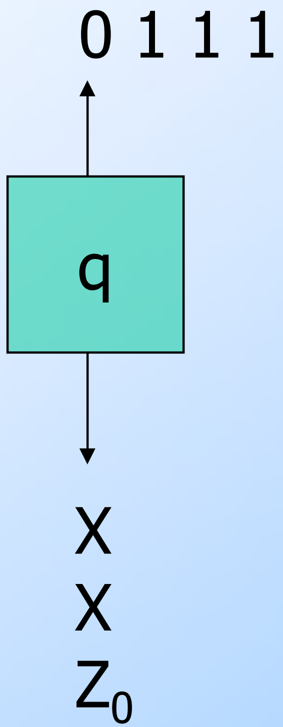
Azioni del PDA dell'Esempio



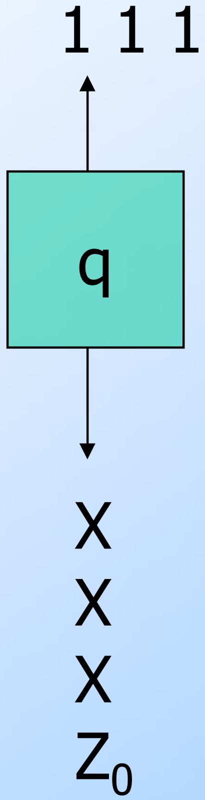
Azioni del PDA dell'Esempio



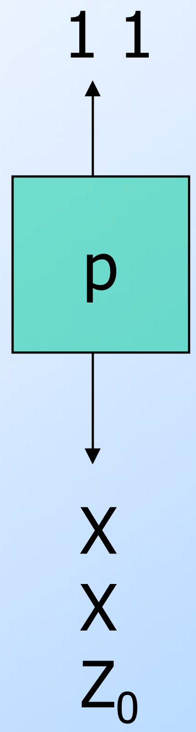
Azioni del PDA dell'Esempio



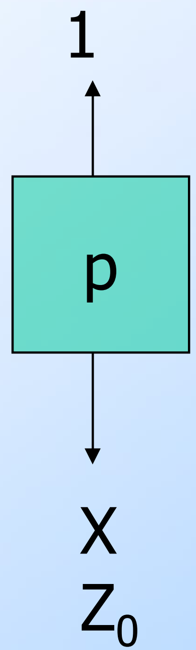
Azioni del PDA dell'Esempio



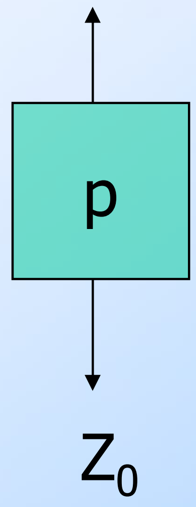
Azioni del PDA dell'Esempio



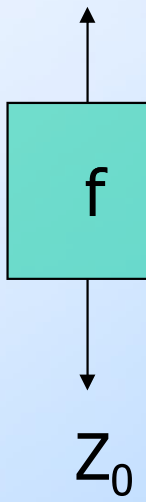
Azioni del PDA dell'Esempio



Azioni del PDA dell'Esempio



Azioni del PDA dell'Esempio



Descrizioni Istantanee

- Possiamo formalizzare le figure viste con una *descrizione istantanea* (ID).
- Una ID è una tripla (q, w, α), dove:
  1. q è lo stato (corrente).
  2. w è l'input residuo.
  3. α è il contenuto dello stack, sommità a sinistra (top → down corrisponde a left → right).

Trasformazioni di ID

- Per dire che la ID I può diventare la ID J in un'azione del PDA, scriveremo $I \vdash J$.
- Formalmente, $(q, aw, X\alpha) \vdash (p, w, \beta\alpha)$ per ogni w e α, se $\delta(q, a, X)$ contiene (p, β).
- Estendiamo $\vdash$ to $\vdash^*$, per rappresentare "zero o più mosse":
  - Base: $I \vdash^* I$.
  - Induzione: Se $I \vdash^* J$ e $J \vdash K$, allora $I \vdash^* K$.

Esempio: Trasformazioni di ID

- Usando il precedente esempio di PDA, possiamo descrivere la sequenza di mosse come segue:

$(q, 000111, Z_0) \vdash (q, 00111, XZ_0) \vdash$
$(q, 0111, XXZ_0) \vdash (q, 111, XXXZ_0) \vdash$
$(p, 11, XXZ_0) \vdash (p, 1, XZ_0) \vdash (p, \epsilon, Z_0) \vdash$
(f, ϵ, Z_0)

- Quindi, $(q, 000111, Z_0) \vdash^* (f, \epsilon, Z_0)$.
- Cosa accade sull'input 0001111?

Risposta

Legale perché un PDA può usare input ϵ anche con input residuo.

- $(q, 0001111, Z_0) \vdash (q, 001111, XZ_0) \vdash (q, 01111, XXZ_0) \vdash (q, 1111, XXXZ_0) \vdash (p, 111, XXZ_0) \vdash (p, 11, XZ_0) \vdash (p, 1, Z_0) \vdash (f, 1, Z_0)$
- Nota che non vi sono possibili mosse successive dall'ultima ID.
- 0001111 non è accettata, perché l'input non è stato interamente consumato.

Nota: Notazioni per FA e PDA

- Noi rappresentiamo le azioni di un FA mediante una δ estesa, che non menziona l'input che deve essere ancora letto.
- Avremmo potuto scegliere una notazione simile per i PDA, dove lo stato del FA è rimpiazzato da una combinazione stato-stack come mostrato nelle figure.

Notazioni per FA e PDA – (2)

- Analogamente, avremmo potuto scegliere una notazione per gli FA con ID.
  - Eliminando la componente della pila.

Linguaggio di un PDA

- Il modo comune di definire il linguaggio di un PDA è mediante gli *stati finali* *("accettazione per stati finali")*.
- Se P è un PDA, allora L(P) è l'insieme delle stringhe w tali che $(q_0, w, Z_0) \vdash^* (f, \epsilon, \alpha)$ con f stato finale e per ogni α.

Linguaggio di un PDA – (2)

- Un altro linguaggio definito dallo stesso PDA è quello delle stringhe che portano a *svuotare lo stack (accettazione per pila vuota)*.
- Se P è un PDA, allora N(P) è l'insieme delle stringhe w tali che $(q_0, w, Z_0) \vdash^* (q, \epsilon, \epsilon)$, dove q è uno stato qualsiasi.

Equivalenza delle Definizioni di Linguaggio di un PDA

1. Se L = L(P), allora c'è un altro PDA P' tale che L = N(P').
2. Se L = N(P), allora c'è un altro PDA P'' tale che L = L(P'').

Prova: L(P) -> N(P’) Intuizione

- P’ simula P.
- Se P accetta, P’ svuota la sua pila.
- P’ usa uno speciale indicatore di fondo della pila per evitare di svuotarla quando P svuota il suo stack senza accettare.

Prova: L(P) -> N(P')

- P' ha tutti gli stati, i simboli e le mosse di P, più :
  1. Un simbolo X_0 di fondo pila, usato per difendere lo stack da accidentali svuotamenti.
  2. Un nuovo stato iniziale s e uno stato "cancella" e.
  3. $\delta(s, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$. Fa partire P.
  4. $\delta(f, \epsilon, X) = \delta(e, \epsilon, X) = \{(e, \epsilon)\}$ per ogni stato finale f di P e ogni simbolo di pila X.

Prova: N(P) -> L(P'') Intuizione

- P'' simula P.
- P'' ha uno speciale indicatore di fondo della pila per rilevare quando P svuota il suo stack.
- Quando questo accade, P'' accetta (se ha letto tutto l'input).

Prova: N(P) -> L(P”)

- P” ha tutti gli stati, i simboli e le mosse di P, più :
  1. Un simbolo di pila X_0, usato per controllare il fondo dello stack.
  2. Un nuovo stato iniziale s e uno stato finale f.
  3. $\delta(s, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$. Fa partire P.
  4. $\delta(q, \epsilon, X_0) = \{(f, \epsilon)\}$ per ogni stato q di P.

PDA Deterministici

- Per essere deterministico, ci deve essere al più una possibile mossa per ogni stato q, simbolo input *a* e simbolo di stack X.
- Inoltre, non deve essere possibile una scelta tra usare come input ϵ o un input "reale".
- Formalmente, $\delta(q, a, X)$ e $\delta(q, \epsilon, X)$ non possono essere entrambi non vuoti.