PCAD
Programmazione Concorrente
Algoritmi Distribuiti

Arnaud Sangnier
arnaud.sangnier@unige.it

Verifica di sistemi 2

Proprietà regolari

- **Problem:**
  - Come descrivere in un modo conciso e manipolabile le proprietà temporali lineari che sono in pratica sotto-insiemi di $(2^{PA})^\omega$
- **Solution:**

  1) Con degli automi (pratici da un punto di vista algoritmico, ma meno per descrive specifiche)
     - Un automa permette infatti di descrive in un modo finiti degli insiemi infiniti di sequenze
     - Con gli **automi di Büchi** si possono rappresentare insiemi infiniti di sequenze infinite !
  2) Con delle formule logiche (pratiche per descrivere specifiche ma meno pratiche da un punto di vista algoritmico)

Automi di Büchi

- Definizione: Un **automa di Büchi (BA)** A è un tuple $(Q, \Sigma, \delta, Q_{in}, F)$ dove:
  - Q è un insieme finite di stato
  - Σ è l'alfabeto
  - $\delta: Q \times \Sigma \mapsto 2^Q$ è la funzione di transizione
  - $Q_{in} \subseteq Q$ è l'insieme dei stati iniziali
  - $F \subseteq Q$ è l'insieme dei stati accettanti

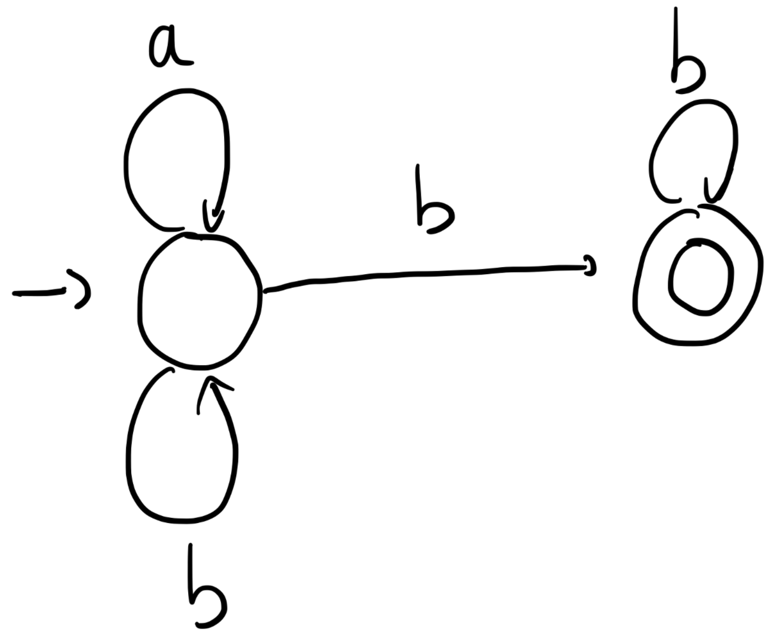
- **Notazione:** Scriveremo q -a→ q' if $q' \in \delta(q,a)$

Linguaggio

- Consideriamo un automa di Büchi $A=(Q, \Sigma, \delta, Q_{in}, F)$
- Un esecuzione di A su una sequenza (parola) infinita $w=w_0\ w_1\ w_2 \dots$ in Σ^ω è una sequenza infinita di stati $q_0\ q_1\ q_2 \dots$ tale che $q_0 \in Q_{in}$ e q_i -wi→ q_{i+1} per tutti i
- L'esecuzione è detta accettante se esiste un numero infinito di i tale che $q_i \in F$
- I.e. in una esecuzione accettante dei stati di F sono visitati infinitamente spesso
- Denotiamo $L_\omega(A)=\{w$ in Σ^ω | esiste una esecuzione accettante di A su w$\}$
- $L_\omega(A)$ è il linguaggio di A
- Abbiamo $L_\omega(A) \subseteq \Sigma^\omega$

Esempio

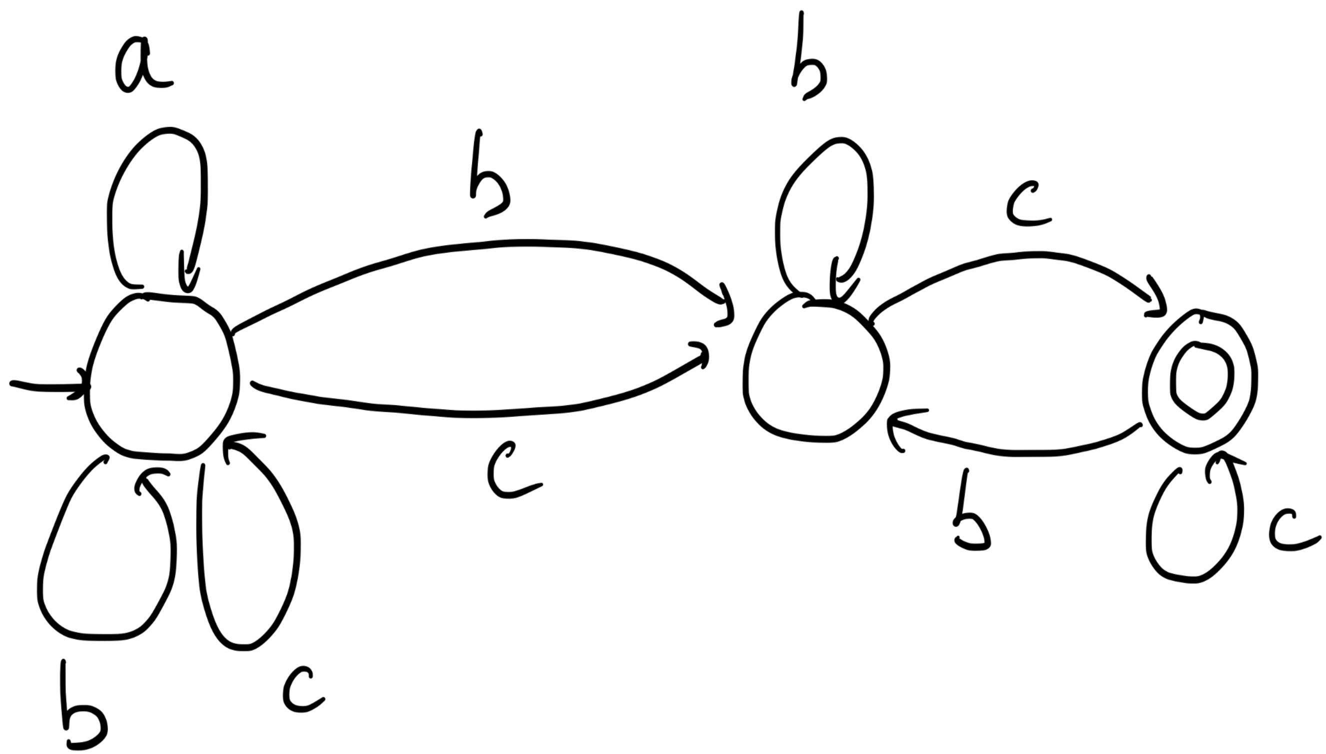
- Questo automa di Büchi su Σ={a,b} riconosce tutte le parole infinite su l'alfabeto che hanno un numero finito di a



→ ○ : stato iniziale   ◎ stato accettante

Esempio

- Questo automa di Büchi su Σ={a,b,c} riconosce tutte le parole infinite su l'alfabeto che hanno un numero finito di a e un numero infinito di c



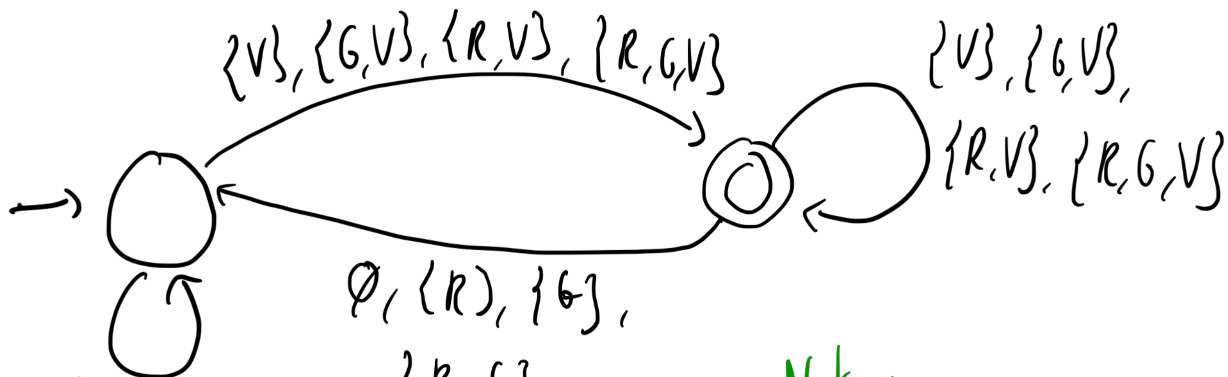
Uso degli automi di Büchi

- Usiamo gli automi di Büchi per rapprentare delle proprietà temporali lineari
- L'alfabeto di questi automi sarà dunque $\Sigma = 2^{PA}$, i.e. ogni transizione verrà etichettata con un sotto-insieme di PA
- Una proprietà temporale lineare $P \subseteq (2^{PA})^{\omega}$ è detta **regolare** se esiste un automa di Büchi A tale che $P = L_{\omega}(A)$
- **Importante**: non tutte le proprietà temporale lineare sono regolare
- **Importante**: l'insieme delle proprietà temporale lineare regolare è chiuso per l'operazione di unione, intersezione e complemento (teoria dei linguaggi ω-regolari)
  - Se P e P' sono proprietà temporali lineari regolari allora $P \cup P'$, $P \cap P'$ e $(2^{PA})^{\omega} \setminus P$ sono proprietà temporali lineari regolari

Esempio

- Prendiamo PA={R,G,V} e consideriamo la proprietà temporale lineare P_V che dice che si vede V infinitamente spesso
- P_V è regolare

$$2^{PA} = \{\emptyset, \{R\}, \{G\}, \{V\}, \{R,G\}, \{G,V\}, \{R,V\}, \{R,G,V\}\}$$

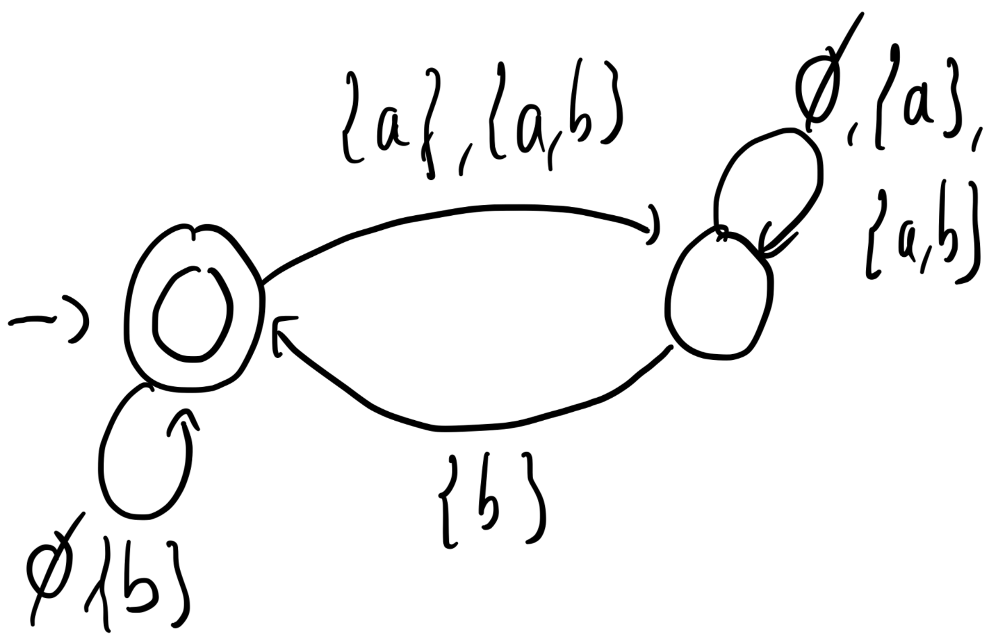


Notazione:

Esempio

- Prendiamo PA={a,b} e consideriamo la proprietà temporale lineare P_{ab} che dice che ogni volta che vediamo a allora strettamente dopo nel futuro si vede un b
- P_{ab} è regolare

$$2^{PA} = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$$



Testare se il linguaggio è vuoto

- Consideriamo un automa di Büchi $A=(Q, \Sigma, \delta, Q_{in}, F)$
- Le due seguente proposte sono equivalente:
  1) $L_\omega(A) \neq \emptyset$
  2) esiste uno stato $q \in F$ tale che q è raggiungibile in A da un stato in Q_{in} e q appartiene ad un ciclo (non vuoto) in A
- Per testare se $L_\omega(A) \neq \emptyset$, basta cercare un stato q che verifica la proprietà 2)

Model-checking di proprietà regolari

- Sia P una proprietà temporale lineare **regolare** e KS una struttura di Kripke
- Vogliamo verificare se Trace(KS) $\subseteq$ P
- Ragioneremo su $\overline{P} = (2^{PA})^{\omega} \setminus P$
- Ricordiamo che $\overline{P}$ è anche lei regolare
- Abbiamo Trace(KS) $\subseteq$ P sse Trace(KS) $\cap\ \overline{P} = \emptyset$
- Come $\overline{P}$ è regolare allora esiste un automa de Büchi $\overline{A}$ tale che $L(\overline{A})=\overline{P}$
- Quindi infine abbiamo:

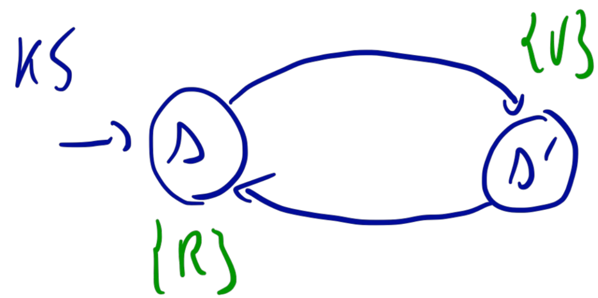
$$\mathbf{KS \models P \text{ sse } Trace(KS) \cap L_{\omega}(\overline{A}) = \emptyset}$$

Verificare se Trace(KS) $\cap$ $L_\omega(\overline{A}) = \emptyset$

- Costruiamo un prodotto KS $\otimes$ $\overline{A}$
- Sia KS=(S,$\rightarrow$, s_{in}, PA, L) e $\overline{A}$=(Q, Σ, δ, Q_{in}, F), definiamo una struttura (un grafo con vertici iniziali) tale KS $\otimes$ $\overline{A}$=(S',$\Rightarrow$,I) tale che:
  - S'=S x Q
  - $\Rightarrow$ $\subseteq$ S' x S' verifica (s,q) (t,p) sse s$\rightarrow$t (in KS) e q-L(t)$\rightarrow$p (in $\overline{A}$)
  - I={(s_{in},q) | esiste $q_{in} \in Q_{in}$ tale che q_{in}-L(s_{in})$\rightarrow$q
- Guardiamo se esiste in KS $\otimes$ $\overline{A}$ un camino da un vertice in I verso un vertice (s,f) con f in F e (s,f) appartiene ad un ciclo in KS $\otimes$ $\overline{A}$
- Se non esiste un tal cammino, abbiamo Trace(KS) $\cap$ $L_\omega(\overline{A}) = \emptyset$ e quindi KS $\models$ P, altrimenti KS $\not\models$ P

Esempio

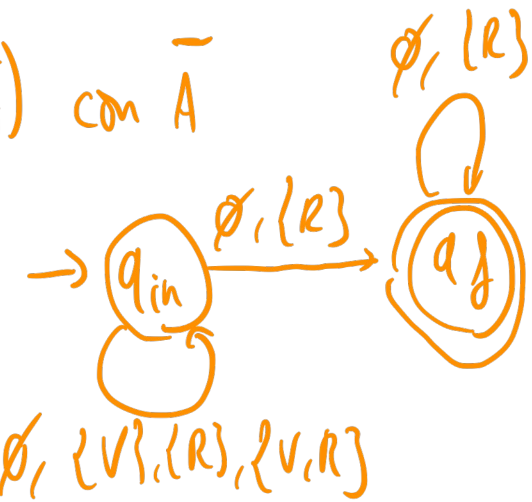
PA = {V, R}



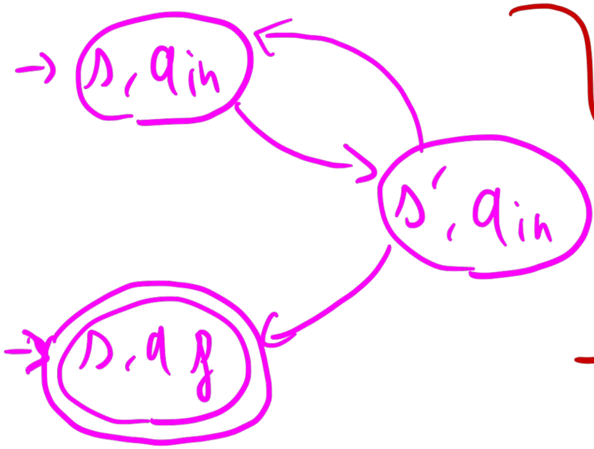
P: si vede V infinitamente spesso

$\bar{P}$: si vede V un numero finito di volte

$\bar{P} = \mathcal{L}_\omega(\bar{A})$ con $\bar{A}$



KS ⊗ $\bar{A}$



(s, q_f) è raggiungibile da un stato iniziale ma non appartiene ad un ciclo! $KS \models P$