Αυτόματα σε Άπειρες Λέξεις

Χριστίνα Φουντουκίδου

05/06/2019

Ερώτημα

Τι είναι άπειρες λέξεις; ; ;

Υπάρχουν άπειρες λέξεις;

Κίνητρο

Άπειρες λέξεις εμφανίζονται όταν εργαζόμαστε σε

- λειτουργικά συστήματα
- τραπεζικά συστήματα
- συστήματα ελέγχου εναέριας κυκλοφορίας
- συστήματα επικοινωνίας δικτύου

Το 1962, ο *Büchi* μελετώντας προβλήματα λογικής, χρησιμοποίησε αυτόματα σε άπειρες λέξεις.

Συμβολισμοί

- A^*:σύνολο όλων των πεπερασμένων λέξεων από το A
- A^ω: σύνολο όλων των απείρων λέξεων από το A

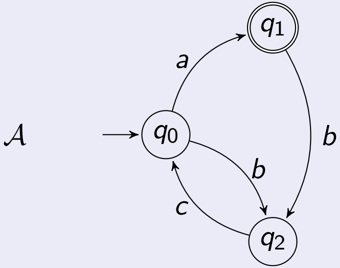
Büchi

Ένα **μη προσδιοριστό** *Büchi* αυτόματο

$$\mathcal{A} = (Q, A, I, T, F)$$

- Q: το πεπερασμένο σύνολο των καταστάσεων
- A: το αλφάβητο εισόδου
- $I \subseteq Q$: το σύνολο των αρχικών καταστάσεων
- $T \subseteq Q \times A \times Q$: το σύνολο των μεταβάσεων
- F: το σύνολο των τελικών καταστάσεων.

Παράδειγμα *Büchi* αυτομάτου

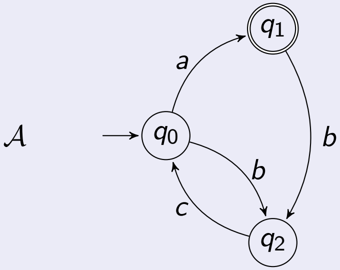


- $w_1 = abcabcabc\ldots$
- $w_2 = bcbcbc\ldots$

Büchi

- $w = a_0 a_1 \ldots \in A^\omega$
- διαδρομή του $\mathcal{A}$ στην w $\mathcal{P}_w^{(\mathcal{A})} : ((q_i, a_i, q_{i+1}))_{i \geqslant 0}$
  όπου $(q_i, a_i, q_{i+1}) \in T \ \forall i \geqslant 0$.
- $\mathcal{P}_w^{(\mathcal{A})}$: επιτυχής αν εμφανίζεται άπειρες φορές μία τουλάχιστον τελική κατάσταση
- w **αναγνωρίζεται** από το $\mathcal{A}$ αν υπάρχει $\mathcal{P}_w^{(\mathcal{A})}$ επιτυχής
- $L(\mathcal{A})$: σύνολο απείρων λέξεων που αναγνωρίζει το $\mathcal{A}$.

Παράδειγμα *Büchi* αυτομάτου



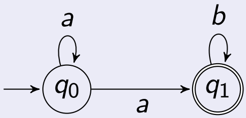
- $w_1 = abcabcabc \ldots$ αναγνωρίσιμη
- $w_2 = bcbcbc \ldots$ μη αναγνωρίσιμη

Büchi

- $L \subseteq A^\omega$: ω-αναγνωρίσιμη αν υπάρχει *Büchi* αυτόματο $\mathcal{A}$ με $L = L(\mathcal{A})$
- ω-$Rec(A)$: κλάση όλων των ω-αναγνωρισίμων γλωσσών από το A

Büchi

Παράδειγμα *Büchi* αυτομάτου



Υποθέτουμε ότι η είσοδος σε αυτό το *Büchi* αυτόματο είναι $abbbbbbbbb\ldots$, τότε η διαδρομή στο αυτόματο με αυτή την είσοδο παράγει την ακολουθία $q_0 q_1 q_1 q_1 q_1 q_1 \ldots$.

Το αυτόματο αναγνωρίζει τη συμβολοσειρά $abbbbbbbbb\ldots$ αφού η τελική κατάσταση q_1 εμφανίζεται άπειρες φορές.

ω-*Rec* γλώσσες: Ιδιότητες

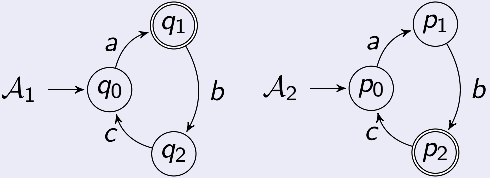
Αν $L, L_1, L_2 \in \omega\text{-}Rec(A)$, τότε:

- $L_1 \cup L_2 \in \omega\text{-}Rec(A)$
- $L_1 \cap L_2 \in \omega\text{-}Rec(A)$
- $\bar{L} \in \omega\text{-}Rec(A)$

Σκιαγράφηση της απόδειξης της ένωσης

- $\mathcal{A}_1 = (Q_1, A, I_1, T_1, F_1),\ L_1 = L(\mathcal{A}_1)$
- $\mathcal{A}_2 = (Q_2, A, I_2, T_2, F_2),\ L_2 = L(\mathcal{A}_2)$
- $Q_1 \cap Q_2 = \varnothing$
- $\mathcal{A} = (Q_1 \cup Q_2, A, I_1 \cup I_2, T_1 \cup T_2, F_1 \cup F_2)$
- $\mathcal{P}_w^{(\mathcal{A})} : ((p_i, a_i, p_{i+1}))_{i \geqslant 0}$ επιτυχής διαδρομή τότε $p_0 \in I_1 \cup I_2$.
  - $p_0 \in I_1$, $\mathcal{P}_w^{(\mathcal{A}_1)}$ επιτυχής διαδρομή στο $\mathcal{A}_1, w \in L_1$
  - $p_0 \in I_2$, $\mathcal{P}_w^{(\mathcal{A}_2)}$ επιτυχής διαδρομή στο $\mathcal{A}_2, w \in L_2$

Büchi: Για την τομή η γνωστή κατασκευή δεν δουλεύει!



(α′) αυτόματο $\mathcal{A}_1$ (β′) αυτόματο $\mathcal{A}_2$

$$\mathcal{A}_1 : \longrightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{c} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{c} q_0 \longrightarrow \cdots$$

$$\mathcal{A}_2 : \longrightarrow p_0 \xrightarrow{a} p_1 \xrightarrow{b} p_2 \xrightarrow{c} p_0 \xrightarrow{a} p_1 \xrightarrow{b} p_2 \xrightarrow{c} p_0 \longrightarrow \cdots$$

- $L_1 = \{abcabcabc...\} = L_2$
- $L_1 \cap L_2 = \{abcabcabc...\}$

Büchi

Στην κλασική κατασκευή θα είχαμε:

$$\mathcal{A}_1 \cap \mathcal{A}_2 : \longrightarrow (q_0, p_0) \xrightarrow{a} (q_1, p_1) \xrightarrow{b} (q_2, p_2) \xrightarrow{\quad c \quad}$$

$$\xrightarrow{\quad c \quad} (q_0, p_0) \xrightarrow{a} (q_1, p_1) \xrightarrow{b} (q_2, p_2) \xrightarrow{c} (q_0, p_0) \longrightarrow \cdots$$

με τελική την (q_1, p_2).

Η (q_1, p_2) δεν εμφανίζεται πουθενά!

Σκιαγράφηση της απόδειξης της τομής

Θεωρούμε σύνολο, $\{0,1,2\}$:

- $Q = Q_1 \times Q_2 \times \{0,1,2\}$
- $(q_0, p_0, 0)$ αρχική
- εμφάνιση τελικής από $\mathcal{A}_1$, το 0 γίνεται 1 και συνεχίζουμε με 1
- εμφάνιση τελικής από $\mathcal{A}_2$, το 1 γίνεται 2
- αμέσως μετά το 2 γίνεται 0
- $Q_1 \times Q_2 \times \{2\}$: τελικές καταστάσεις

Πίσω στο Παράδειγμα

$$\mathcal{A}_1 : \longrightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{c} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{c} q_0 \longrightarrow \cdots$$

$$\mathcal{A}_2 : \longrightarrow p_0 \xrightarrow{a} p_1 \xrightarrow{b} p_2 \xrightarrow{c} p_0 \xrightarrow{a} p_1 \xrightarrow{b} p_2 \xrightarrow{c} p_0 \longrightarrow \cdots$$

Εφαρμόζοντας την παραπάνω διαδικασία:

$$\mathcal{A}_1 \cap \mathcal{A}_2 : \ \longrightarrow (q_0, p_0, 0) \xrightarrow{a} (q_1, p_1, 1) \xrightarrow{b} (q_2, p_2, 2) \xrightarrow{c}$$

$$\xrightarrow{c} (q_0, p_0, 0) \xrightarrow{a} (q_1, p_1, 1) \xrightarrow{b} (q_2, p_2, 2) \xrightarrow{c} (q_0, p_0, 0) \longrightarrow \cdots$$

Büchi

Η απόδειξη για το συμπλήρωμα είναι τεχνικά δύσκολη και με μεγάλη πολυπλοκότητα!!!

Αν $L, L_1, L_2 \in \omega\text{-}Rec(A)$, τότε μπορούμε να αποφασίσουμε αν

- $L = \varnothing$ ή όχι.

- $L_1 \subseteq L_2$ ή όχι.

- $L_1 = L_2$ ή όχι.

Büchi

Παράδειγμα

$L = \{aba^2b^2a^3b^3 ...\} \notin \omega\text{-}Rec(A)$.

Αντίθετα έστω *Büchi* αυτόματο $\mathcal{A} = (Q, A, I, T, F)$ με $L = L(\mathcal{A})$. Υπάρχει κατάσταση $q \in F$ που εμφανίζεται άπειρες φορές και :

$$\rightarrow q_0 \xrightarrow{aba^2b^2 \quad a^m b^{m-k}} q \xrightarrow{b^k a^{m+1} b^{m+1} \quad a^{m+l}} q$$

με $l > 0$. Άρα η λέξη

$$aba^2b^2a^3b^3 ... a^m b^{m-k}(b^k a^{m+1} b^{m+1} ... a^{m+l} b^k a^{m+1} b^{m+1} ... a^{m+l} ...)$$

ανήκει στην L. Άτοπο

Büchi

Büchi αυτόματο $\mathcal{A} = (Q, A, I, T, F)$ **προσδιοριστό**:

- μία μόνο αρχική κατάσταση
- $\forall q \in Q, a \in A$ υπάρχει το πολύ μία $q' \in Q$ με $(q, a, q') \in T$

Büchi

Σημαντική Διαφορά

Μη προσδιοριστά *Büchi* αυτόματα $\neq$ προσδιοριστά *Büchi* αυτόματα.

Όριο

- $L \subseteq A^*$, τότε
  $\vec{L} = \{w \in A^\omega / a_0 a_1 \ldots a_n \in L$ για άπειρα σε πλήθος $n \geqslant 0\} \subseteq A^\omega$

Παράδειγμα

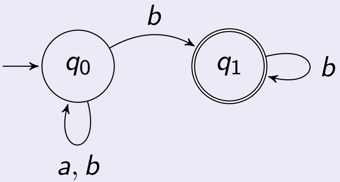
- Όταν $L = ba^*$, τότε $\vec{L} = baaa\ldots$
- Όταν $L = c^{10}b^4c^{20}b^5(ca)^*$, τότε $\vec{L} = c^{10}b^4c^{20}b^5(cacaca\ldots)$.
- Όταν $L = a^5b^{1000}c^{100}(b^4c^5a^3)^*$, τότε
  $\vec{L} = a^5b^{1000}c^{100}(b^4c^5a^3b^4c^5a^3b^4c^5a^3\ldots)$.
- Όταν $L = \{a^{10}b, c^4\}$, τότε $\vec{L} = \varnothing$.

Θεώρημα

Μια γλώσσα $L \subseteq A^\omega$ *είναι αναγνωρίσιμη από Büchi προσδιοριστό αυτόματο αν και μόνο αν υπάρχει αναγνωρίσιμη γλώσσα πεπερασμένων λέξεων* $V \subseteq A^*$ *έτσι ώστε* $L = \vec{V}$.

Παράδειγμα

$L = (a \cup b)^* bbb\ldots \in \omega\text{-}Rec(A)$ και το μη προσδιοριστό *Büchi* αυτόματο που την αναγνωρίζει



Η L **δεν αναγνωρίζεται** από προσδιοριστό *Büchi* γιατί αλλιώς θα ήταν

- υπάρχει $V \in Rec(A)$ έτσι ώστε $L = \vec{V}$.
- $bbb\ldots \in \vec{V}$, άρα υπάρχει $j_1 > 0$ με $b^{j_1} \in V$
- $b^{j_1}abbb\ldots \in \vec{V}$, άρα υπάρχει $j_2 > 0$ με $b^{j_1}ab^{j_2} \in V$.

. . .

Άτοπο

Müller

Σημαντική Διαφορά

Μη προσδιοριστά *Müller* αυτόματα $\equiv$ Προσδιοριστά *Müller* αυτόματα.

Müller

Ένα *Müller* αυτόματο

$$\mathcal{A} = (Q, A, I, T, \mathcal{F})$$

- Q: το πεπερασμένο σύνολο των καταστάσεων
- A: το αλφάβητο εισόδου
- $I \subseteq Q$: το σύνολο των αρχικών καταστάσεων
- $T \subseteq Q \times A \times Q$: το σύνολο των μεταβάσεων και
- $\mathcal{F} \subseteq \mathcal{P}(Q)$: το σύνολο των συνόλων των τελικών καταστάσεων

Müller

Παράδειγμα *Müller* αυτομάτου

$Q = \{q_0, q_1, q_2, q_4, q_5, q_7, p, p'\}$ $\mathcal{F} = \{\{p, p'\}, \{q_0, q_1, q_2\}, \{q_0, q_7\}\}$

$$\mathcal{P}_w^{(\mathcal{A})} : \longrightarrow q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} p \xrightarrow{a_3} q_4 \xrightarrow{a_4} q_5 \xrightarrow{a_5}$$

$$\xrightarrow{a_5} p' \xrightarrow{a_6} q_7 \xrightarrow{\quad|\quad} p \longrightarrow p' \longrightarrow p \longrightarrow p'$$

Müller

Θεώρημα

Μη προσδιοριστά *Büchi* ≡ Μη προσδιοριστά *Müller*

Μη προσδιοριστά *Müller* ≡ Προσδιοριστά *Müller*

Εφαρμογές

Τα *Büchi* αυτόματα χρησιμοποιούνται στην πιστοποίηση λογισμικού ($model - checking, verification$) δηλαδή στην εύρεση λογικών λαθών.

Πως γίνεται αυτό; ; ;

- Ένα πρόγραμμα για τη λύση ενός προβλήματος παριστάνεται με ένα *Büchi* αυτόματο $\mathcal{A}$.
- Οι απαιτήσεις του προβλήματος περιγράφονται με έναν τύπο από μία γλώσσα λογικής.
- Ο τύπος μεταφράζεται σε ένα *Büchi* αυτόματο $\mathcal{B}$.
- Μπορούμε να αποφασίσουμε αν $\mathcal{A} \equiv \mathcal{B}$ ή όχι, οπότε μπορούμε να αποφασίσουμε αν το πρόγραμμα ικανοποιεί τις απαιτήσεις του προβλήματος ή όχι.

Ευχαριστώ πολύ για την προσοχή σας!