Đoán nhận ngôn ngữ

Đoán nhận ngôn ngữ

- Automat hữu hạn có thể đoán nhận được những tập ngôn ngữ nào?
  - Đây là một bài toán cực kỳ khó
  - Bài toán này lần đầu tiên được nhà toán học Mỹ Stephen Kleene giải quyết vào năm 1956
    - Các ***tập chính quy (regular set)*** có thể được đoán nhận
      - ✧ Tập được xây dựng từ tập rỗng, xâu rỗng và các xâu chỉ chứa một ký hiệu bằng cách ghép, lấy hợp và lấy bao đóng Kleene theo một thứ tự tuỳ ý
        - ✸ Văn phạm chính quy
      - ✧ Một tập là chính quy nếu và chỉ nếu nó được sinh bởi một văn phạm chính quy

Nhắc lại: Cho A là một tập con của V^*. Khi đó ***bao đóng Kleene*** của A – được ký hiệu là A^* – tập gồm các phép ghép một số tuỳ ý các xâu thuộc A. Điều này có nghĩa là $A^* = \cup_{k=0}^{\infty} A^k$

Đoán nhận ngôn ngữ

- Có những tập không thể được đoán nhận bởi automat hữu hạn
  - Có các mô hình tính toán mạnh hơn, như các ***automat đẩy xuống (pushdown)*** và các ***máy Turing***

Tập chính quy

Biểu thức chính quy (regular expression) trên tập I được định nghĩa một cách đệ quy như sau:

ký hiệu $\emptyset$ là một biểu thức chính quy

ký hiệu λ là một biểu thức chính quy

ký hiệu x là một biểu thức chính quy với mọi $x \in I$

các ký hiệu $(\boldsymbol{AB})$, $(\boldsymbol{A} \cup \boldsymbol{B})$ và $\boldsymbol{A}^*$ là các biểu thức chính quy với mọi $\boldsymbol{A}$ và $\boldsymbol{B}$ là các biểu thức chính quy.

- Tập chính quy: biểu diễn bằng biểu thức chính quy

Tập chính quy

- Ví dụ

Xác định các xâu trong tập chính quy được mô tả bởi các biểu thức chính quy sau: 10^*, $(10)^*$, $0(0 \cup 1)^*$, $(0^*1)^*$

- Lời giải
  - 10^*: một số 1 theo sau bởi một số bất kỳ số 0 (kể cả không)
  - $(10)^*$: một số bất kỳ các cặp 10 (kể cả xâu rỗng)
  - $0(0 \cup 1)^*$: xâu bất kỳ bắt đầu bằng 0
  - $(0^*1)^*$: xâu bất kỳ kết thúc bằng 1

Định lý Kleene

Một tập là chính quy khi và chỉ khi nó được chấp nhận bởi một automat hữu hạn

- Chứng minh: giáo trình

Tập chính quy và văn phạm chính quy

- Văn phạm chính quy (văn phạm loại 3)
  - Các luật sinh có dạng $S \to \lambda$, $A \to a$, hoặc $A \to aB$

Định lý: một tập sinh bởi một văn phạm chính quy khi và chỉ khi nó là một tập chính quy.

  - Chứng minh: giáo trình

Tập không được chấp nhận bởi automat hữu hạn

- Có những tập không phải là chính quy, ví dụ:
  - Tập $\{0^n 1^n \mid n = 0,1,2,\ldots\}$ không phải là chính quy
    - Chứng minh: giáo trình

Các loại máy trạng thái mạnh hơn

- Automat hữu hạn không thể thực hiện nhiều tính toán
  - Hạn chế chủ yếu của các máy này là lượng nhớ hữu hạn
  - Automat đẩy xuống (pushdown automaton), automat bị giới hạn tuyến tính (linear bounded automaton), máy Turing là các lựa chọn thay thế mạnh hơn.

Automat đẩy xuống

- Automat đẩy xuống (pushdown automat)
  - Là một automat hữu hạn kèm theo một ngăn xếp, cung cấp một bộ nhớ không giới hạn.
  - Đoán nhận các ngôn ngữ sinh bởi văn phạm phi ngữ cảnh.
  - Một tập được chấp nhận bởi một trong hai cách
    - Tập gồm tất cả các xâu sinh ra một ngăn xếp rỗng khi các xâu đó được dùng làm đầu vào.
    - Tập gồm tất cả các xâu dẫn tới một trạng thái kết thúc.

Automat bị giới hạn tuyến tính

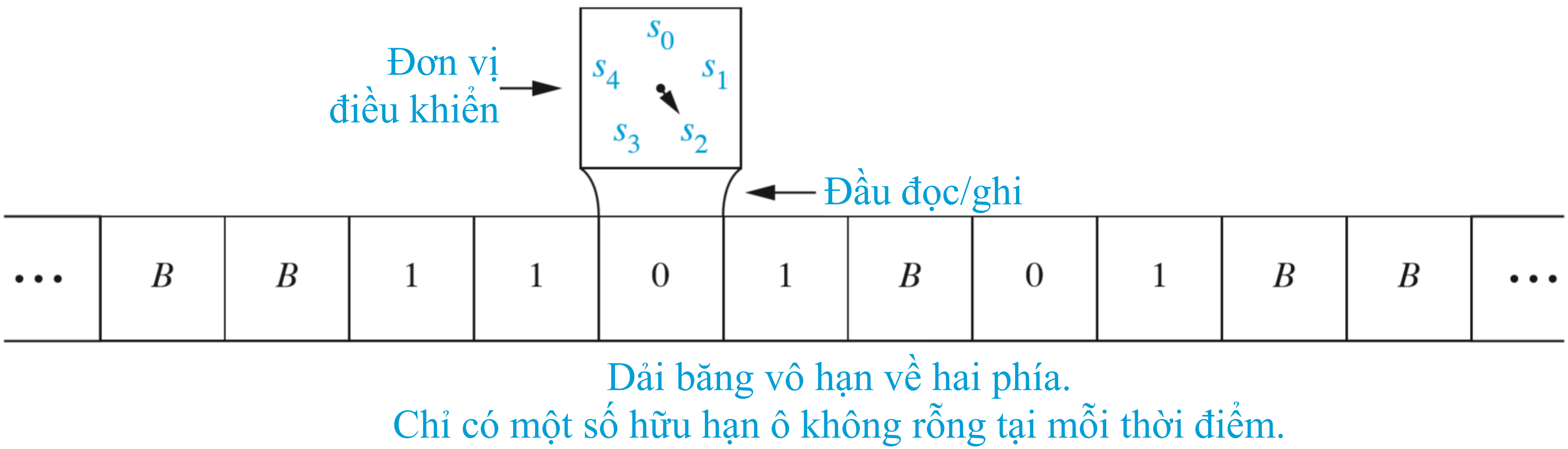
- Mạnh hơn automat đẩy xuống, có thể đoán nhận được các tập ví dụ như $\{0^n1^n2^n \mid n = 0,1,2,\ldots\}$
- Đặc biệt, automat bị giới hạn tuyến tính có thể đoán nhận được các ngôn ngữ cảm ngữ cảnh.
- Tuy nhiên, không thể đoán nhận được tất cả các văn phạm cấu trúc câu.
  - Máy Turing là automat mạnh hơn

Máy Turing

Máy Turing

- Mô hình tính toán tổng quát nhất
  - Đặc biệt, chúng có thể mô hình tất cả các tính toán có thể thực hiện trên một máy tính
    - Đặt tên theo nhà toán học nổi tiếng Alan Turing; được phát minh vào thập niên 1930
- Là một máy hữu hạn trạng thái với một dải băng từ vô hạn
  - Băng từ được chia thành nhiều ô (nhớ)
    - Các máy Turing đọc và viết cá ký hiệu trên băng khi đơn vị điều khiển chạy tới chạy lui dọc theo băng, làm thay đổi các trạng thái tuỳ thuộc vào ký hiệu được đọc trên băng.

Máy Turing



HÌNH 1. BIỂU DIỄN MỘT MÁY TURING

Máy Turing $T = (S, I, f, s_0)$ gồm một tập hữu hạn S các trạng thái, bộ chữ cái I chứa ký hiệu khoảng trống B, một hàm bộ phận f từ $S \times I$ đến $S \times I \times \{R, L\}$ và trạng thái xuất phát s_0

Máy Turing

- Quy tắc chuyển trạng thái của máy
  - Các bộ năm tương ứng với hàm bộ phận trong định nghĩa của máy Turing
  - Nếu hàm bộ phận f không xác định đối với cặp (s, x), thì máy Turing sẽ ***dừng lại***.

- Ở mỗi một bước, đơn vị điều khiển đọc ký hiệu x hiện thời trên băng. Nếu đơn vị điều khiển ở trạng thái s và nếu hàm bộ phận xác định đối với cặp (s, x) với $f(s, x) = (s', x', d)$ thì đơn vị điều khiển:
  1. chuyển vào trạng thái s'
  2. viết ký hiệu x' vào ô hiện thời sau khi đã xoá x, rồi
  3. chuyển sang phải một ô nếu $d = R$ hoặc chuyển sang trái một ô nếu $d = L$

Máy Turing để đoán nhận các tập

- ***Trạng thái kết thúc*** của máy Turing T
  - Là trạng thái không phải là trạng thái đầu tiên trong bất kỳ bộ năm phần tử nào trong đặc tả của T

Cho V là tập con của bộ chữ cái I. Một máy Turing $T = (S, I, f, s_0)$ ***chấp nhận*** xâu x trong V^* khi và chỉ khi T xuất phát từ vị trí ban đầu khi x đã được viết trên băng sẽ dừng lại ở một trạng thái kết thúc. T được nói là chấp nhận một tập con A của V^* khi và chỉ khi với mọi x thuộc A, x được chấp nhận bởi T.

- x không được chấp nhận nếu T không dừng lại hoặc dừng lại không phải ở trạng thái kết thúc khi máy hoạt động trên băng có chứa các ký hiệu của x trong các ô liên tiếp và xuất phát từ vị trí bắt đầu.

Tính các hàm bằng máy Turing

- Với một máy Turing T
  - Khi đã cho xâu đầu vào x sẽ dừng lại với xâu y trên băng của nó, ta định nghĩa $T(x) = y$
  - Việc xem xét một máy Turing như một máy tính được các giá trị của một hàm trên các xâu là rất hữu ích
    - Làm thế nào dùng máy Turing để tính các hàm được xác định trên các số nguyên, trên cặp số nguyên, trên bộ ba số nguyên, v.v?
      - Cần phương pháp biểu diễn bộ k số nguyên trên băng của máy

Tính các hàm bằng máy Turing

- Máy Turing hoạt động như một máy tính với các hàm của số nguyên
  - Dùng biểu diễn nhất phân của các số nguyên
    - Một số nguyên không âm n biểu diễn bằng xâu của $n+1$ số 1
      - Ví dụ: 0 biểu diễn là xâu 1, 5 biểu diễn là xâu 111111
    - Bộ k số nguyên $(n_1, n_2, \ldots, n_k)$ biểu diễn bằng một xâu gồm $n_1 + 1$ số 1, tiếp sau là một dấu sao $(*)$, tiếp sau gồm $n_2 + 1$ số 1, tiếp sau lại là một dấu sao, v.v, và kết thúc bằng $n_k + 1$ số 1

Các loại máy Turing khác

- Các biến thể của định nghĩa máy Turing
  - Mỗi bước dịch chuyển sang trái, sang phải hoặc không dịch chuyển hoàn toàn
  - Dùng nhiều dải băng đồng thời
    - Sử dụng bộ $(2 + 3n)$ để biểu diễn máy Turing khi sử dụng n dải băng
  - Dải băng có hai chiều
    - Dịch chuyển lên, xuống, trái hoặc phải
  - Có nhiều đầu đọc, đọc đồng thời nhiều ô khác nhau
  - Máy Turing không tất định
  - Hạn chế băng vô hạn chỉ theo một hướng hoặc hạn chế bộ chữ cái của băng chỉ có hai ký hiệu

Luận đề Church-Turing

- Máy Turing tương đối đơn giản nhưng lại cực kỳ mạnh
  - Để cộng và nhân các số
  - Để tính toán các hàm đặc biệt nào đó có thể tính được bằng một thuật toán
    - Có thể khá phức tạp để xây dựng một máy Turing nhưng hoàn toàn khả thi

Luận đề Church-Turing

- "Đối với một bài toán đã cho bất kỳ có thể giải được bằng ***một thuật toán hiệu quả***, thì sẽ tồn tại một máy Turing có thể giải được bài toán đó."
  - Khái niệm về khả năng giải quyết bằng một thuật toán hiệu quả
    - Là khai niệm không chính thức và không chính xác
    - Sử dụng lượng bộ nhớ không hạn chế
      - Đối lập với khái niệm thuật toán hiệu quả chạy trên máy tính trong thực tế, chỉ được sử dụng một lượng bộ nhớ hạn chế

Độ phức tạp tính toán, khả năng tính toán và khả năng giải quyết

- Tới nay, độ phức tạp được định nghĩa bằng số lượng phép toán cơ bản được sử dụng bởi thuật toán hiệu quả nhất
  - Các phép toán trên bit, so sánh các số nguyên, phép toán số học, v.v.
  - Không chính xác
    - Bởi vì các loại phép toán dùng để đo độ phức tạp thay đổi rất mạnh
- Nếu luận đề Church-Turing là đúng
  - Dùng máy Turing để chính xác hoá một số quan niệm liên quan đến độ phức tạp tính toán

Độ phức tạp tính toán

Một ***bài toán quyết định (decision problem)*** đặt ra câu hỏi một mệnh đề từ một lớp mệnh đề cụ thể là đúng hay không. Bài toán quyết định còn được biết đến là bài toán có-không (yes-or-no).

- Có thuật toán hay không?
  - Có thể xác định xem các mệnh đề từ lớp mệnh đề của nó là đúng hay không
- Ví dụ: “n có phải số nguyên tố?”

Độ phức tạp tính toán

- Giải quyết bài toán có-không
  - Giống như đoán nhận ngôn ngữ bao gồm tất cả các xâu bit biểu diễn giá trị đầu vào cho bài toán dẫn đến câu trả lời là “có”.
  - Do đó, giống như đoán nhận ngôn ngữ tương ứng với các giá trị đầu vào mà câu trả lời cho bài toán là “có”.

Khả năng giải quyết

- Bài toán ***có thể giải được (solvable)*** hoặc ***có thể quyết định được (decidable)***
  - Có một thuật toán hiệu quả quyết định xem các trường hợp của bài toán có đúng hay không
  - Ví dụ: bài toán xác định liệu một số nguyên dương là số nguyên tố
- Bài toán ***không giải được*** hoặc ***không quyết định được***
  - Không tồn tại ***thuật toán hiệu quả*** để giải quyết bài toán

Khả năng giải quyết

Định nghĩa: Bài toán dừng là bài toán quyết định đặt ra câu hỏi liệu một máy Turing T cuối cùng có thể dừng lại khi đầu vào là xâu x.

Định lý: Bài toán dừng là một bài toán quyết định không giải được. Nghĩa là, không tồn tại máy Turing T nào có thể xác định xem T cuối cùng có dừng lại khi bắt đầu với xâu đầu vào x được ghi trên băng của nó.

Khả năng giải quyết

- Ví dụ khác của bài toán không giải được
  - Liệu hai văn phạm phi ngữ cảnh có sinh ra tập các xâu giống nhau?
  - Liệu với một bộ gạch được dùng lặp lại có thể bao phủ toàn bộ một bề mặt mà không chồng chéo lên nhau?
  - Bài toán Hilbert thứ mười
    - Liệu có thể tìm được một thuật toán giúp ta xác định sau một số bước hữu hạn bước, một phương trình đa thức bất kỳ có nghiệm nguyên hay không? (Bài toán thứ mười trong số 23 bài toán Hilbert đặt ra vào năm 1900. Câu trả lời đã được nhà toán học Matiyasevich đưa ra năm 1970 là không có thuật toán như vậy)

Khả năng tính toán

- Hàm tính toán
  - Có thể tính toán được bằng máy Turing
- Dễ dàng để chứng minh rằng có các hàm số học không toán được
  - Không đơn giản để tạo ra một hàm như vậy

Khả năng tính toán

- Mỗi bài toán quyết định có thể được định nghĩa lại là một bài toán về tính toán một hàm
  - Giá trị là 1 khi câu trả lời cho bài toán là “có” và 0 cho trường hợp ngược lại
  - Bài toán quyết định là bài toán giải được khi và chỉ khi hàm tương ứng được xây dựng theo cách này tính toán được.

Khả năng tính toán

- Một máy Turing không tất định *T* được cho là chấp nhận một xâu *x* nếu và chỉ nếu có một chuỗi chuyển tiếp của *T* dừng ở trạng thái kết thúc.
- Các máy Turing không tất định được cho nào để giải bài toán quyết định
  - Nếu nó chấp nhận tập bao gồm tất cả các giá trị đầu vào mà câu trả lời cho bài toán là có.

Khả năng tính toán

Một bài toán quyết định thuộc lớp P nếu nó có thể giải bằng một máy Turing tất định trong thời gian đa thức. Nghĩa là, bài toán quyết định thuộc lớp P nếu có một máy Turing tất định T giải bài toán, và một đa thức $p(n)$ sao cho với tất cả các số nguyên n, T dừng ở trạng thái kết thúc sau ít hơn hoặc bằng $p(n)$ phép chuyển bất cứ khi nào đầu vào cho T là một chuỗi chiều dài n.

Một bài toán quyết định thuộc lớp NP nếu nó có thể giải bằng một máy Turing không tất định trong thời gian đa. Nghĩa là, một bài toán quyết định thuộc lớp NP nếu có một máy Turing không tất định T giải bài toán, và một đa thức $p(n)$ sao cho với tất cả các số nguyên n, T dừng với tất cả sự lựa chọn các phép chuyển sau ít hơn hoặc bằng $p(n)$ phép chuyển bất cứ khi nào đầu vào cho T là một chuỗi chiều dài n.

FILM ENDED & G’LUCK