Automi e Linguaggi Formali

Linguaggio universale, riduzioni, e teorema di Rice

A.A. 2014-2015
Alessandro Sperduti
sperduti@math.unipd.it

Università degli Studi di Padova

Il linguaggio universale

- Il linguaggio universale L_u è l'insieme delle stringhe binarie che codificano una coppia (M, w) dove $w \in L(M)$.
- Esiste una TM U, detta TM universale, tale che $L_u = L(U)$.
- U ha tre nastri: uno per il codice di M e w, uno per il nastro di M, e uno per la codifica dello stato di M. Così U simula M su w, e accetta (M, w) se e solo se M accetta w.
- Dato che alcune TM M corrispondono a linguaggi non ricorsivi, ma ricorsivamente enumerabili, non si fermano quando la stringa w in input non è del linguaggio. Anche U avrà lo stesso comportamento su (M, w). Quindi L_u è RE e non ricorsivo.

Il problema dell'arresto

- Data una TM M, definiamo $H(M)$ l'insieme delle stringhe w tali che M si arresta con input w.
- Consideriamo il linguaggio che contiene le coppie (M, w) tali che $w \in H(M)$.
- Anche per questo linguaggio esiste una TM che prende in input le coppie (M, w) tali che $w \in H(M)$, e simula il comportamento di M su w. Quindi l'insieme di tali coppie è un linguaggio RE ma non ricorsivo.
- Quindi, non esiste nessun algoritmo che possa dire se un programma termina o no. Esiste però un algoritmo che, se il programma in input termina, si ferma, altrimenti non si arresta.

Proprietà di chiusura dei linguaggi ricorsivi

I linguaggi ricorsivi sono chiusi sotto le seguenti operazioni:

- unione
- intersezione
- concatenazione
- chiusura di Kleene

Fare le prove come esercizio

Proprietà di chiusura dei linguaggi RE

I linguaggi RE sono chiusi sotto le seguenti operazioni:

- unione
- intersezione
- concatenazione
- chiusura di Kleene

Fare le prove come esercizio

RE e ricorsivo

Dato un linguaggio ricorsivo e uno RE, consideriamo la loro

- unione: RE
- intersezione: RE
- concatenazione: RE
- Se L_1 ricorsivo e L_2 RE, $L_2 - L_1$ è *RE* e $L_1 - L_2$ non è RE

Fare le prove come esercizio

Riduzione

- Dato un problema P_1 indecidibile, vogliamo vedere se un altro problema P_2 è indecidibile o no.
- Basta mostrare che, se potessimo risolvere P_2, allora protremmo usare la sua soluzione per risolvere P_1. Questa tecnica è detta riduzione di P_1 a P_2.
- Ridurre P_1 a P_2 significa convertire qualunque stringa di P_1 in una stringa di P_2, e qualunque stringa non in P_1 in una stringa non in P_2.
- Supponiamo che esista un algoritmo che risolve P_2. Data una stringa w per P_1, la convertiamo in un'altra stringa x per P_2. Poi usiamo l'algoritmo di soluzione per P_2 per decidere se x è in P_2 o no. Qualunque sia la risposta, è valida anche per w in P_1. Perciò, a partire dall'algoritmo che risolve P_2, abbiamo costruito un algoritmo che risolve P_1.

Riduzioni

- Se P_1 si riduce a P_2, allora P_2 è difficile almeno quanto P_1.
- Se P_1 non è ricorsivo, allora neanche P_2 è ricorsivo.
- Se P_1 non è RE, neanche P_2 è RE.

Theorem

Se esiste una riduzione da P_1 a P_2, allora:

- *se P_1 è indecidibile, lo è anche P_2*
- *se P_1 non è RE, non è RE neanche P_2*

Due linguaggi

Consideriamo due linguaggi:

- $L_e = \{M | L(M) = \emptyset\}$
- $L_{ne} = \{M | L(M) \neq \emptyset\}$

Vogliamo capire se sono ricorsivi, o RE, o neanche RE.

L_{ne} è RE

- $L_{ne} = \{M | L(M) \neq \emptyset\}$
- Basta mostrare che esiste una TM che lo accetta
- Basta prendere una M non-deterministica che, data M_i in input, non-deterministicamente simula M_i su tutte le possibili w. Se c'è una w che è accettata da M_i, allora M accetta M_i.

L_{ne} non è ricorsivo

- $L_{ne} = \{M | L(M) \neq \emptyset\}$
- Lo dimostriamo dando una riduzione da L_u a L_{ne}. Visto che L_u non è ricorsivo, non può esserlo neanche L_{ne}.
- Per la riduzione, dobbiamo convertire ogni coppia (M, w) tale che $w \in L(M)$ in una TM M' tale che $w \in L(M)$ se e solo se $L(M') \neq \emptyset$.
- M' è una TM che (ignora il suo input e) simula M su w. Se M accetta w, allora M' accetta il suo input. Quindi $L(M')$ non è vuoto. Se M non accetta w, allora M' non accetta nessun input. Quindi $L(M')$ è vuoto.

L_e non è RE

- $L_e = \{M | L(M) = \emptyset\}$
- Dato che L_{ne} non è ricorsivo, e che L_e è il complemento di L_{ne}, L_e non può essere ricorsivamente enumerabile.
- Se lo fosse, allora sarebbero entrambi ricorsivi.

Proprietà dei linguaggi

- Tutte le proprietà non banali di una TM (cioè dei linguaggi RE) sono indecibili.

Esempio

Essere un linguaggio libero dal contesto.
Controllare se una TM accetta un linguaggio libero dal contesto è indecidibile.

- Proprietà dei linguaggi come insieme (di linguaggi RE).
- Proprietà banale: se viene soddisfatta da tutti i linguaggi o da nessuno.
- Se P è una proprietà di linguaggi, il linguaggio L_P è linsieme dei codici delle TM M_i tali che $L(M_i)$ è un linguaggio in P. P è decidibile se L_P è decidibile.

Esempio

Consideriamo come proprietà P l'insieme dei linguaggi RE che contengono la stringa "PIPPO".

Linguaggi "PIPPO"

Il linguaggio

$$L_P = \{M_i | L(M_i) \text{ contiene la stringa "PIPPO"}\}$$

è decidibile?

provarlo...

Teorema di Rice

Theorem

Ogni proprietà non banale dei linguaggi RE è indecidibile

Proof.

- Sia P una proprietà non banale dei linguaggi RE.
- Allora deve esistere un linguaggio RE L che abbia la proprietà P. Sia M_L una TM che accetta L.
- Ridurremo L_u a L_P, dimostrando che L_P è indecidibile, dato che L_u è indecidibile.
- Input alla riduzione: coppia (M, w). Output: una TM M'. M' è tale che $L(M') = \emptyset$ se M non accetta w, e $L(M') = L$ se M accetta w.
- M' simula M su w. Se M non accetta w, M' non accetta nessuna stringa x in input. Se M accetta w, M' simula M_L su x, perciò accetterà esattamente L. Dato che L ha la proprietà P, M' è in L_P.

Esempi di problemi indecidibili

Tutti i problemi sulle TM che riguardano i linguaggi accettati sono indecidibili.

Esempi:

- Il linguaggio accettato da una TM è vuoto?
- Il linguaggio accettato da una TM è finito?
- Il linguaggio accettato da una TM è regolare?
- Il linguaggio accettato da una TM è libero da contesto?
- Il linguaggio accettato da una TM contiene la stringa ab?
- Il linguaggio accettato da una TM contiene tutti i numeri pari?

Non tutto è indecidibile

Problemi che riguardano gli stati di una TM, e non il linguaggio accettato, possono essere decidibili.

Esempi:

- Se una TM ha cinque stati.
- Se esiste un input tale che una TM faccia almeno cinque passi prima di fermarsi.
- Se una TM esegue una certa transizione.
- Se una TM passa per lo stato p.
- Se una TM non passa mai per lo stato p.
- Se una TM passa dallo stato p allo stato q in al più tre passi.

Esempio

Data una TM M_i è decidibile se la testina di tale macchina si muoverà mai a sinistra prima di arrestarsi avendo come ingresso (a destra della testina) la stringa "PIPPO" ?

provarlo...