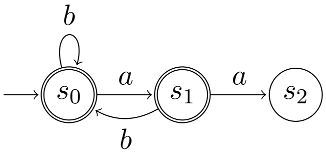
Probleme cu automate finite și expresii regulate

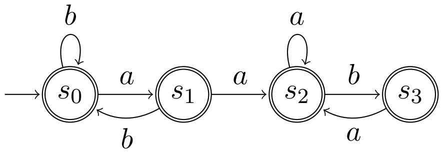
Exercițiu Construiți un automat determinist care acceptă toate șirurile de a și b în care dacă apare bb, până atunci nu a apărut aa.

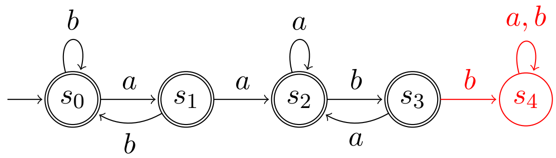
Raționăm astfel: orice șir care *nu* conține aa e bun, dar dacă apare aa, mai târziu nu poate să mai apară bb, altfel nu respectă condiția cerută. Deci construim întâi un automat care recunoaște șirurile în care *nu* apare aa: orice șir care se termină înainte de a fi apărut aa e acceptat.



După orice a, în șir trebuie să apară b pentru a putea continua. Sau, altfel exprimat, numărăm câți a consecutivi am avut imediat înainte de starea curentă: 0 în starea s_0, unul în starea s_1. Deci cu orice b se revine în s_0. Din s_2 însă (doi a consecutivi), automatul nu mai acceptă nimic.

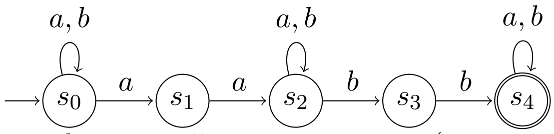
Completăm acum acest automat ca odată ce a văzut aa (din s_2), să nu mai permită bb. Rolurile se inversează acum, după orice b, în șir trebuie să urmeze un a ca să respecte în continuare condițiile:





Toate stările din automat sunt acceptoare: oriunde s-ar opri șirul, el satisface condițiile din enunț. Nu există tranziție pe b din s_3: implicit, un astfel de șir nu e acceptat. Pentru a respecta riguros condiția ca orice stare să aibă tranziție pe orice simbol, putem introduce explicit starea neacceptoare s_4 în care odată ajuns, automatul rămâne indiferent de intrare.

O altă soluție e să definim întâi un automat pentru șirurile care *nu* respectă condiția din enunț: putem construi simplu un automat *nedeterminist* care acceptă dacă întâi vede aa, și mai târziu, bb:



La început, între aa și bb, precum și la sfârșit, șirul poate conține orice combinații de a și b.

Apoi determinizăm acest automat (care acceptă toate șirurile *nedorite*) și obținem complementul lui (șirurile dorite), schimbând fiecare stare din acceptoare în neacceptoare și reciproc. Verificați că obțineți un automat determinist echivalent celui construit direct mai sus.

Erori frecvente

Un automat recunoaște (acceptă) șiruri de un anumit fel. Dacă nu marcăm stări ca fiind acceptoare, nu va accepta nimic. Nu există stare acceptoare implicită (de exemplu "ultima" – nu există "ultima" într-un automat cu cicluri). Nici nu trebuie să fie unică – definiția are *o mulțime* de stări acceptoare.

Un automat consumă pe fiecare tranziție *un singur* simbol. Deci o tranziție nu poate fi etichetată cu ab, sunt necesare două tranziții, cu o stare intermediară.

Un automat cu o stare care pentru același simbol are două tranziții nu e determinist, deci nu e bun dacă se cere un automat determinist. (Dacă a fost însă gândit bine, poate fi determinizat).

Un automat cu o tranziție în buclă revenind în aceeași stare pentru toate simbolurile alfabetului (ca s_0 în ultima figură) poate consuma *orice șir* rămânând în acea stare. Într-un automat *determinist*, având deja tranziții pe toate simbolurile, ea nu poate avea nici tranziții spre alte stări. Deci nu are sens decât eventual ca ultimă stare acceptoare sau de eroare.

Discuție E util să distingem câteva clase tipice de șiruri "bune" (acceptate). Putem avea cazurile:
– odată ce un șir a devenit "bun", el rămâne acceptat, de exemplu, șirurile care *conțin* un anumit tipar: odată tiparul găsit, poate urma orice
– odată ce un șir *nu* e bun, nu mai poate fi acceptat, cum ar fi șirurile care *nu conțin* un anumit tipar: odată ce tiparul apare, poate urma orice, dar șirul nu mai poate fi acceptat
– putem avea o alternanță de stări acceptoare și neacceptoare, cum ar fi șiruri cu un număr par de 1: fiecare 1 citit trece automatul dintr-o stare acceptoare într-una neacceptoare și invers.

Odată ce am scris un automat, e util să îl parcurgem pentru a vedea câteva șiruri acceptate, cu mare atenție la cicluri: combinând cicluri putem obține situații care poate ne-au scăpat inițial.

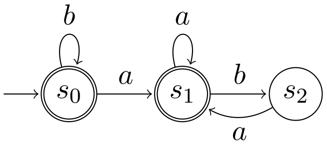
Exercițiu Scrieți o expresie regulată pentru șirurile de a și b în care orice ab e urmat imediat de a.

Notăm cu | alternativa. Precedența cea mai mare o are *, urmată de concatenare și alternativă.

Putem gândi soluția în mai multe feluri, dar oricum trebuie să exprimăm cum arată șirurile odată ce a apărut ab. Orice ab trebuie urmat de a. În particular, ar putea fi urmat de încă un ab (urmat la rândul său de a), dar putem avea și oricâți a între doi ab consecutivi. Ajungem la expresia regulată $(ab|a)^*$. Dacă se termină cu ab, ea trebuie urmată de a; e valabil însă și șirul vid: $\epsilon|(ab|a)^*a$.
Înainte de porțiunea unde se poate repeta ab, putem avea oricâți b urmați de oricâți a; după această porțiune (deci când nu mai apare b) putem încheia cu oricâți a. Repetiția de a e însă deja inclusă în expresia găsită. E suficient să completăm deci: $b^*(\epsilon|(ab|a)^*a)$.

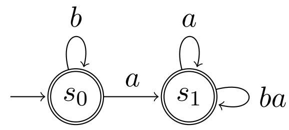
Putem gândi și așa: odată ce a apărut primul a, nu mai putem avea bb, ci doar ba. Ajungem la $(a|ba)^*$. Din nou, aceasta include oricâți a dorim, la început și sfârșit. E suficient deci să adăugăm oricâți b inițial: $b^*(a|ba)^*$.

O altă variantă, poate mai laborioasă dar mai sigură e să construim întâi un automat.

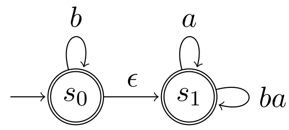


Inițial, avem un șir de b care nu introduce constrângeri. Apoi, orice b din s_1 înseamnă că a fost precedat de un a, deci trebuie urmat de un a, revenind în aceeași stare.

Pentru a obține expresia regulată, vedem ușor că putem elimina starea intermediară neacceptoare s_2. Obținem:



Cum ambele stări rămase sunt acceptoare, limbajul devine $b^*(\epsilon|a(a|ba)^*)$. Remarcând că a din fața parantezei interioare poate fi generat și din repetiție, iar apoi ϵ la fel, ajungem la aceeași formă simplificată, $b^*(a|ba)^*$. Putem vedea mai bine simplificarea remarcând că înlocuind a cu ϵ pe tranziția $s_0 \to s_1$ obținem același efect, și ϵ e element neutru (nu contează) la concatenare.



Erori frecvente

Expresia regulată $(a|b)^*$ poate genera orice șir de a și b. Ca și la automate, dacă vrem să generăm șiruri care *nu* conțin un tipar, nu putem avea $(a|b)^*$ ca subexpresie, pentru că poate genera orice, inclusiv tiparul nedorit.

Deși expresiile regulate sunt echivalente cu automatele, dacă problema cere o expresie regulată, aceasta e ceea ce trebuie să dăm ca soluție.

Discuție

Cele două reprezentări din dreapta *nu* sunt automate (nici deterministe nici nedeterministe) după definiția dată, pentru că acestea nu pot avea șiruri arbitrare pe tranziții, ci doar câte *singur* simbol. Pot fi însă reprezentări informale utile pentru transformările prin care obținem o expresie regulată.