Nedeterministický a deterministický konečný automat

Thursday, May 30, 2013 8:38 AM

Deterministický konečný automat

Je uspořádaná pětice (S,Σ,P,s,F), kde:

- S je konečná množina stavů.
- Σ je konečná množina vstupních symbolů nazývaná abeceda.
- P je tzv. přechodová funkce (též přechodová tabulka), popisující pravidla přechodů mezi stavy.
- s je počáteční stav (s naleží S)
- F je množina koncových stavů (F je podmnožinou S)

Nedeterministický konečný automat

Nedeterministickým konečným automatem (NKA) *bez výstupu* nazýváme každou pětici

A = (Q, Σ, δ, S, F), kde:

- Q je konečná, neprázdná, množina stavů
- Σ je konečná neprázdná množina vstupních symbolů (vstupní abeceda)
- δ (přechodová funkce) je zobrazení δ: QxΣ → P(Q). Kde P(Q) je potenční množina (množina všech podmnožin množiny Q včetně prázné množiny *e*)
- S je množina počátečních stavů (S náleží Q) - není jednoznačně určen počáteční stav
- F je množina koncových stavů (F náleží Q)

Oborem hodnot přechodové funkce jsou všechny podmnožiny množiny stavů

Formálně je definován podobně jako DKA, ale obsahuje prvky nedeterminismu:

1. nejednoznačně určený počáteční stav (může jich být více)
2. nejednoznačné přechody (při přijetí stejného vstupu lze přejít do více stavů)
3. e - přechody (přechod do stavu bez přijetí vstupního symbolu)

- chování NKA lze popsat sekvencí pozic (množina stavů, ve kterých se automat může nacházet), z nichž každá jednoznačně definuje, zda je zpracovaný řetězec akceptován či zamítnut
- pozic je konečný počet
- přechody mezi pozicemi jsou jednoznačné
- Nejdůležitější rozdíl mezi DKA a NKA je v tom, že výsledkem přechodové funkce není pouze jeden stav, ale množina stavů, která může být i prázdná
- to vše jsou vlastnosti DKA a proto **ke každému NKA existuje ekvivalentní DKA**

V případě nedeterministického konečného automatu (NKA) je vstupní slovo akceptováno (rozpoznáno,) pokud toto slovo může automat převést do některého z koncových stavů (množina F) z některého z počátečních stavů (množina S).

Převod NKA na DKA

1. Lineární gramatiku nejprve převedeme na regulární tvar (postup viz otázka [*Ekvivalence konečných automatů a regulárních gramatik*].

2.Pak zkonstruujeme nedeterministický konečný automat a z něho nakonec deterministický (jak viz dále).

A) Hlavní myšlenka je taková, že **každý stav vytvořeného DFA odpovídá množině stavů NFA.**

B) Nebo: Z nedeterministického automatu se vytváří **strom**, který již popisuje deterministický automat, popisující tentýž problém.

Postup převodu

Samotný převod stojí na myšlence, že pokud lze ze vstupního uzlu

S
 přejít jdo uzlu

A
 a do uzlu

B
, tak vytvoříme nový uzel, říkejme mu

$[A, B]$
. Tento uzel bude mít stejné vstupy a výstupy jako sjednocení uzlů

A
 a

B
. Nyní tabulka převedeného automatu obsahuje dva uzly

$\{S, [A, B]\}$
. Postup opakujeme pro uzel

$[A, B]$
. Takto postupně projdeme všechny stavy nově vytvářeného deterministického automatu.

Koncovými uzly převedeného deterministického automatu budou takové uzly, které jsou nadmnožinou koncových uzlů původního automatu (měl-li původně automat výstupní uzel

A
, tak uzel

$[A, X]$
, který vznikl jako sjednocení uzlu

A
 a uzlu

X
, bude také výstupní).

Tento postup zároveň eliminuje všechny stavy, do kterých se deterministická verze automatu nemůže vůbec dostat. Zároveň ale mohou vzniknout uzly, které mají totožné vlastnosti (vstupní a výstupní uzly, konečnost, vlastnost *být počátečním uzlem*). Tyto uzly můžeme po doběhnutí algoritmu ztotožnit.

Příklad

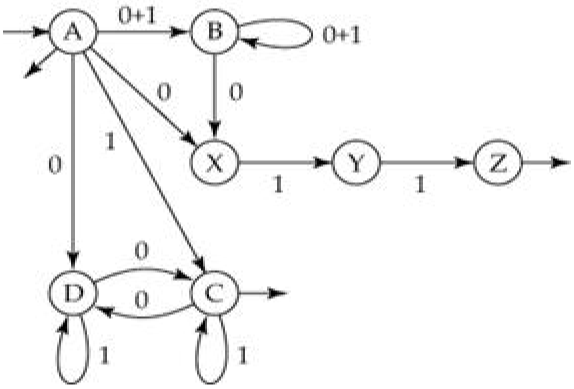
Zadaná pravá lineární gramatika:

```
A --> B | C
B --> 0B | 1B | 011
C --> 0D | 1C | e
D --> 0C | 1D
```

Pravá regulární gramatika:

```
A --> 0B | 1B | 0X | 0D | 1C | e
B --> 0B | 1B | 0X
X --> 1Y
Y --> 1Z
Z --> e
C --> 0D | 1C | e
D --> 0C | 1D
```

Nedeterministický konečný automat:



Deterministický konečný automat:

Přechodovou tabulku deterministického konečného automatu vytvoříme z přechodového diagramu nedeterministického kon. automatu takto:

- Do prvního řádku tabulky zapíšeme počáteční stav automatu a postupně zjistíme, do jakých množin stavů se nedeterministický automat může dostat z tohoto stavu přijmutím jednotlivých symbolů jeho vstupní abecedy.
- Z nalezených množin s více než jedním stavem vytvoříme tzv. *kompozitní* stavy det. automatu. Ty pak použijeme do přechodové tabulky det. automatu jako výstupy přechodové funkce pro počáteční stav a odpovídající vstupní symboly.
- Vzniklé kompozitní stavy (a případně i normální stavy) také využijeme v dalších řádcích přechodové tabulky a případně doplňujeme nové kompozitní stavy, do kterých se můžeme dostat z množin původních stavů každého kompozitního stavu přes vstupní symboly.
- Takto postupně vytvoříme celou přechodovou tabulku ekvivalentního deterministického automatu.
- Kompozitní stavy, zahrnující původní koncové stavy, můžeme označit také jako koncové.

	stav	0	1
<-->	A	BXD	BC
	BXD	BXC	BYD
<--	BC	BXD	BC
<--	BXC	BXD	BYC
	BYD	BXC	BZD
<--	BYC	BXD	BZC
<--	BZD	BXC	BD
<--	BZC	BXD	BC
	BD	BXC	BD

Nové stavy jsou A,BXD, BC, BXC, atd. Přechody 0,1.