

Navrhování rozpoznávacích automatů podruhé

Příklad 9.

L_9 je množina všech řetězců nad abecedou $\{0,1\}$, které současně splňují následující podmínky:

- řetězec obsahuje podřetězec 1101 (podmínka 1)
- řetězec neobsahuje podřetězec 001 (podmínka 2)

Je zřejmé, že na rozdíl od předchozího příkladu nelze postupovat tak, že nejprve budeme vyhledávat podřetězec z první podmínky a až po něm podřetězec z druhé podmínky (na pořadí, v němž se hledané podřetězce objeví, v tomto případě nezáleží).

Úvaha: kdybychom měli pro každou z výše uvedených podmínek zkonstruovaný jeden KA, splňoval by vstupní řetězec obě podmínky právě tehdy, když by oba dva KA převedl do koncových stavů.

Nápad = návod: vytvoříme pro každou podmínku jeden KA a poté na jejich základě zkonstruujeme KA, který si bude „současně pamatovat stavy obou výše zmíněných KA“, tj. vytvoříme finální KA jako *kartézský součin dvou automatů*.

Vytvoření kartézského součinu automatů:

KA, které „separátně“ řeší „každou podmínku zvlášť“, budeme nazývat *parciální automaty*. Výsledný automat si musí „pamatovat“ současně stavy obou parciálních automatů. Stavy výsledného automatu tedy budou uspořádané dvojice stavů obou parciálních automatů. Protože je v každém parciálním automatu pro každou dvojici (stav, vstupní symbol) jednoznačně definován následující stav, je tomu tak i u výsledného automatu. Obecně – jestliže platí $\delta_1(q_{1,i}, a) = q_{1,j}$ a $\delta_2(q_{2,k}, a) = q_{2,l}$, pak pro přechodovou funkci výsledného automatu bude platit $\delta((q_{1,i}, q_{2,k}), a) = (q_{1,j}, q_{2,l})$. Jednoznačně je definován i počáteční stav výsledného automatu jako $(q_{1,0}, q_{2,0})$, kde $q_{1,0}$ a $q_{2,0}$ jsou počáteční stavy parciálních automatů.

Výše uvedeným způsobem se vytvoří přechodová funkce výsledného automatu, množina stavů, a počáteční stav.

Vytvoření přechodové funkce prakticky:

1. Nakreslíme přechodové grafy obou dílčích automatů tak, že se snažíme všechny stavy umístit do jedné přímky (stavy prvního automatu do přímky vodorovné, druhého automatu do přímky svislé) tak, aby vzniklo místo na zakreslení stavů výsledného automatu „v průsečících souřadnic“ (viz řešení příkladu).
2. Do pozice odpovídající „průsečíku souřadnic“ počátečních stavů obou dílčích automatů zakreslíme počáteční stav.
3. Pro každé písmeno vstupní abecedy zakreslíme do „průsečíku souřadnic“ následujících stavů dílčích automatů následující stav automatu výsledného.
4. Bod 3 opakujeme tak dlouho, dokud nám v přechodovém grafu vznikají nové stavy. Je zřejmé, že tento postup někdy skončí (počet stavů výsledného automatu je konečný).

Výše uvedený postup lze jednoduše modifikovat i pro práci s tabulkovou reprezentací automatů, pak jej lze použít i na vytvoření kartézského součinu obecného počtu n parciálních automatů.

Určení množiny koncových stavů:

Je nutné si uvědomit, že množina řetězců může být ve vztahu ke dvěma podmínkám definována různým způsobem:

- řetězce musí splňovat alespoň jednu z podmínek
- řetězce musí splňovat právě jednu z podmínek
- řetězce musí splňovat obě podmínky

Do množiny koncových stavů výsledného automatu podle toho vybereme takové stavy, kde

- alespoň jeden z dvojice stavů je koncovým stavem parciálního automatu
- právě jeden z dvojice stavů je koncovým stavem parciálního automatu
- oba stavy z dvojice jsou koncovými stavy parciálních automatů

Využití konečného automatu ke vstupní konverzi konstant

Příklad 11.

Navrhněte konečněautomatový model algoritmu pro softwarovou implementaci vstupní konverze textového řetězce, reprezentujícího zápis konstanty typu INTEGER, do vnitřní reprezentace počítače (formát v pevné řádové čárce).

Ilustrace – reprezentace některých konstant ve znakové reprezentaci a ve formátu INTEGER s délkou slova 4 byty (každý řetězec je zakončen ukončovací mezerou 20_H, obsahy jednotlivých bytů jsou zapsány hexadecimálně):

Konstanta	Reprezentace ve znakovém poli							Reprezentace INTEGER			
	(kódování ASCII)										
314	33	31	34	20				00	00	01	3A
-314	2D	33	31	34	20			FF	FF	FE	C6
+27123	2B	32	37	31	32	33	20	00	00	69	F3

Úvahy

1. Rozpoznávací automat vydává o vstupním řetězci pouze odpověď akceptuji/zamítám, nicméně prostřednictvím svých stavů si o vstupním řetězci může „pamatovat“ veškerou informaci nezbytnou pro řešení dané úlohy.
2. V počítači můžeme ve formátu INTEGER zobrazit celá čísla pouze z konečného intervalu $\langle \text{MININT}, \text{MAXINT} \rangle$ (v případě zobrazení na 4 bytech je to z intervalu $\langle -2.147.483.648, 2.147.483.647 \rangle$).
3. Konečný počet stavů k řešení této úlohy stačí, otázkou je pouze to, jakým způsobem takový automat navrhnout tak, aby si v průběhu zpracování vstupního řetězce pamatoval i „hodnotu dosud zpracované části“ a jak popsat jeho přechodovou funkci.

Dílčí problém – návrh rozpoznávacího automatu, jenž bude akceptovat řetězce reprezentující konstanty typu INTEGER.

Další úvahy

1. Jednomu stavu rozpoznávacího automatu může odpovídat větší počet stavů navrhovaného konverzního automatu (jinak by si automat nedokázal „zapamatovat“ hodnotu dosud zpracované části řetězce). Jedno kolečko v přechodovém grafu tedy může představovat více stavů „srovnaných za sebou ve třetím rozměru dvourozměrné nákresny“, tedy *makrostav*.
2. Stavy tvořící jeden makrostav mezi sebou budeme rozlišovat pomocí hodnot *pomocných stavových proměnných*. Pomocné stavové proměnné jsou „součástí stavu“, proto je třeba definovat jejich počáteční hodnoty.
3. Je třeba definovat, jak se hodnoty pomocných stavových proměnných budou transformovat při přechodech.

Nápověda:

Vzhledem k „nesymetrii“ zobrazení v pevné řádové čárce je třeba oddělit zpracování řetězce, který představuje zápornou konstantu, od řetězce reprezentujícího konstantu kladnou (nelze postupovat tak, že si KA „zapamatuje“ znaménko a na konci výpočtu znaménkem vynásobí zformovanou hodnotu).

Příklad 12.

Navrhněte konečněautomatový model algoritmu pro softwarovou implementaci vstupní konverze textového řetězce, reprezentujícího zápis konstanty typu REAL, do vnitřní reprezentace v pohyblivé řádové čárce.

Příklady správných zápisů konstant:

1315. -221. +16. -0.4 .26 -.314 2.5E10 10E6 10.E6

Příklady nesprávných zápisů konstant:

314 -5 0 E20 .

Nápověda:

- Při zpracování čísla před exponentovou částí není třeba oddělovat zpracování kladného a záporného čísla, stačí si v pomocné stavové proměnné zapamatovat znaménko.
- Je třeba oddělit zpracování číslic před desetinnou tečkou a za desetinnou tečkou (každá má jiný význam, zpracovává se jinak).
- Exponentová část je celé číslo, jako část přechodového grafu použijte výsledek Příkladu 11.