

Navrhování rozpoznávacích automatů

Obecně platná rada pro navrhování konečného automatu: KA si prostřednictvím svého stavu musí pamatovat o dosud zpracované části řetězce vše, co je podstatné pro rozhodnutí o tom, zda řetězec akceptuje nebo zamítne.

KA se nemůže vracet zpět ke znakům, které již zpracoval.

Příklad 1.

L_1 je množina všech řetězců, které obsahují sudý počet znaků a (nulu považujeme za sudé číslo).

V našem případě si KA musí pamatovat pouze to, zda je počet znaků a v dosud zpracované části řetězce sudý nebo lichý. Na to stačí dva stavy.

Příklad 2.

L_2 je množina všech neprázdných řetězců, které obsahují sudý počet znaků a .

Počáteční stav nemůže být stavem koncovým (automat nemá akceptovat prázdný řetězec). Kromě počátečního stavu tedy budeme potřebovat další dva stavy (stejně jako v příkladu 1).

Příklad 3.

L_3 je množina všech řetězců, které obsahují právě 2 znaky a .

Automat si musí pamatovat počet znaků a v dosud zpracované části vstupního řetězce, ale počet znaků větší než 2 nemusí rozlišovat. Automat tedy bude mít stavy 0, 1, 2 a > 2 . Stav 2 bude koncový, stav > 2 bude *absorpční*, tj. automat jej již nikdy neopustí, pro a i b bude v tomto stavu smyčka. Souvisí to s tím, že podmínka v zadání je formulována tak, „že se jednou pokažený řetězec už nemůže spravit“.

Příklad 4.

L_4 je množina všech řetězců, které obsahují alespoň 2 znaky a .

Automat si musí opět pamatovat počet znaků a v dosud zpracované části vstupního řetězce, ale počet znaků větší než 1 už nemusí rozlišovat. Automat bude mít stavy 0, 1 a ≥ 2 , stav ≥ 2 bude koncový a absorpční („jakmile se v řetězci objeví dva znaky a , řetězec se už nemůže pokazit.“)

Příklad 5.

L_5 je množina všech řetězců, které začínají podřetězcem *abb*.

Automat si musí pamatovat „co z požadovaného počátečního podřetězce už našel“. V případě, že se objeví nevyhovující znak, automat přejde do nekonečného absorpčního stavu (jsou-li první tři znaky odlišné od *abb*, řetězec už nemůže mít požadovanou vlastnost, ať jsou následující znaky jakékoli). Automat tedy bude mít čtyři stavy, které budou reprezentovat tyto situace:

- q_0 = „zpracování ještě nezačalo“
- q_1 = „první znak řetězce byl *a*“
- q_2 = „první dva znaky řetězce byly *ab*“
- q_3 = „první tři znaky byly *abb* (stav je koncový a absorpční)“
- q_4 = „první tři znaky nebyly *abb* (stav není koncový, je absorpční)“

Příklad 6.

L_6 je množina všech řetězců, které končí podřetězcem *abba*.

Automat si opět musí pamatovat „co z hledaného podřetězce už našel“. Je zřejmé, že v automatu nebude absorpční stav (řetězec může pokračovat dalšími znaky, koncové čtyři znaky se změní). Stavy automatu budou reprezentovat všechny možné „začátky“ (terminologicky správně *předpony* nebo *prefixy*) hledaného podřetězce (včetně prázdného řetězce !!) . Délka hledaného podřetězce je čtyři, má tedy pět prefixů (včetně prázdného řetězce), automat tedy bude mít pět stavů, které budou reprezentovat tyto prefixy:

q_0 = *e* (poslední znak nemůže být prvním znakem podřetězce)
 q_1 = *a* (poslední znak bylo *a*)
 q_2 = *ab* (poslední dva znaky byly *ab*)
 q_3 = *abb* (poslední tři znaky byly *abb*)
 q_4 = *abba* (poslední čtyři znaky byly *abba*; výskyt pořetězce nalezen; stav je koncový)

Obecná rada: Přejchodový graf automatu, jenž má „vyhledávat podřetězec“ je vhodné začít kreslit tak, že začneme od nejkratšího řetězce, jenž splňuje požadovanou podmínku. V našem případě tak dostaneme cestu ze stavu q_0 do stavu q_4 , jejíž hrany budou postupně ohodnoceny písmeny *a, b, b, a*.

Další hrany doplníme na základě nalezení některého prefixu na konci řetězce, jenž vznikne „přiretěžením“ vstupního písmene k aktuálnímu prefixu. Prefixem může být i prázdný řetězec.

Příklad 7.

L_7 je množina všech řetězců, které obsahují podřetězec *abba*.

Od předchozího příkladu se zadání liší pouze záměnou slov „končí podřetězcem“ za „obsahují podřetězec“. Důsledkem bude drobná změna přechodového grafu spočívající v tom, že koncový stav q_4 se stane stavem absorpčním (*a*-hrana a *b*-hrana z přechodového grafu z příkladu 6 budou nahrazeny smyčkami v absorpčním koncovém stavu q_4). Důvod je zřejmý: na rozdíl od předchozího příkladu, kde byla podmínka vztažena k „plovoucímu“ konci řetězce a nemohl tedy existovat absorpční stav, se to, že se podřetězec *abba* ve vstupním řetězci již jednou vyskytl, nemůže „pokazit“.