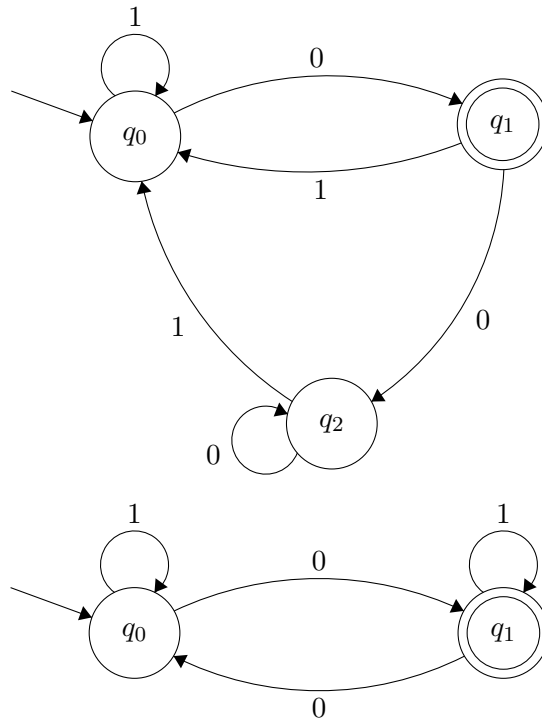


0. príklad

Na obrázku vidíte dva konečné automaty A a B. Nech $L_1 = L(A)$ a $L_2 = L(B)$.



Zostrojte automat, ktorý bude rozpoznávať jazyk $L = \{xyzw \mid x, y, z, w \in \Sigma^*, xz \in L_1 \wedge yw \in L_2\}$.

1. príklad

Zostrojte Turingov stroj počítajúci funkciu, ktorá:

- (a) na vstupe dostane slovo a^n a po výpočte na páske zostane a^{n+1}
- (b) na vstupe dostane a^n a vráti b^n ;
- (c)* na vstupe dostane $a^x \# a^y$ a vráti a^{x+y} ;
- (d) na vstupe dostane $a^x \# a^y$ a vráti a^{x-y} , pričom predpokladáme, že $x \geq y$;
- (e)* na vstupe dostane a^n a vráti a^{2n} ;

Riešenie 1. príkladu

Čo znamená slovné spojenie „Turingov stroj počítajúci funkciu”? Je to TS, ktorý na vstupe dostane nejaké slovo na páske a vráti toto slovo nejakým zmeneným. Čo znamená, že neriešime, či je slovo z jazyka, alebo nie, ale upravujeme vstup. Ideálny začiatok na zoznámenie sa s konštruovaním turingových strojov.

Pozn. *Skôr, ako začneme riešiť, odporúčam vám stiahnuť si program JF-LAP, v ktorom vám budem aj dávať riešenia. Môžete v ňom priamo svoje „turingáče” vytvárať a skúšať, takže vám to určite pôjde rýchlejšie ako na papier. Pozrite časť Dodatočné linky.*

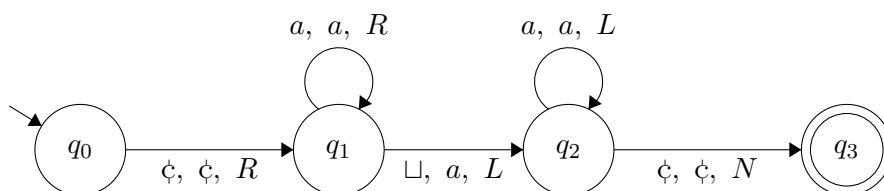
Ako začať pri konštruovaní TS? Musíme si najprv premyslieť stratégiu, čiže ako budeme postupovať, kedy sa čo stane... Je vhodné si túto stratégiu napísať v bodoch. Následne sa treba zamyslieť nad tým, či sme pokryli všetky (dakedy aj okrajové) prípady, a ak nie, tak ich doplniť. Z definície TS vieme, že jeho prechody obsahujú viac informácií ako konečné automaty – potrebujeme vedieť, čo na páske čítame, čo na pásku (na to isté políčko) zapíšeme a kam na nej posunieme čítaciu hlavu (doľava, doprava, alebo bude stáť). Preto medzi každým stavom budeme musieť myslieť na všetky tri tieto akcie. Ukážme si to na príkladoch.

Príklad (a): zo zadania je jasné, že máme počet áčok na páske zvýšiť o jedna. Ako vyzerá páska na začiatku? Máme ϵ , potom niekoľko a -čok a následne nekonečno veľa blankov \sqcup . Náš postup by preto mohol byť taký, že prečítame všetky áčka a budeme sa hýbať doprava, až kým neprídeme na prvý blank. Ten prepíšeme na a a vrátime čítaciu hlavu na začiatok pásky (čo je dobrým zvykom pri konštruovaní takýchto TS).

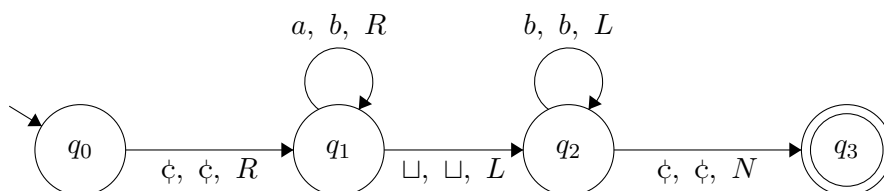
Takže, potrebujeme vstupný stav, tam prečítame cent a posunieme sa doprava na áčka¹. Tam sa budeme „točiť” – čítať áčka, posúvať sa doprava, a áčka necháme na páske (t.j. ich „prepíšeme na a ”). Keď prečítame blank, prepíšeme ho na a , hlava zostane stáť a posunieme sa do ďalšieho stavu. Tam čítaciu hlavu vrátime úplne naľavo (na začiatok pásky) a aby sme na páske nechali to, čo tam bolo, to čo čítame na ňu aj zapisujeme.

Výsledný TS je na obrázku, alebo si ho [stiahnite](#).

¹Pozor, cent pri TS NIKDY neprepisujeme, a ideálne, ani nič neprepisujeme na cent, keďže tým by sme akoby „odstihli” pásku.



Príklad (b): v tomto príklade máme áčka prepísať na béčka. To je pomerne jednoduché. Vždy, keď prečítame áčko, prepíšeme ho na béčko a posunieme sa doprava. Toto opakujeme, kým neprečítame blank. Následne sa bez prepisovania vrátíme na začiatok pásky. Výsledný TS je na obrázku, alebo si ho [stiahnite](#). Ako by sme museli zmeniť TS, ak by na vstupe bolo slovo nad abecedou $\{a, b\}$ a chceli by sme prepísať áčka na béčka? A čo by sme chceli zameniť áčka za béčka a béčka za áčka?



Príklad (c): nechávam na vás. Zamyslite sa, čo treba spraviť s mriežkou, aby ste dostali výsledok.

Príklad (d): kým v predchádzajúcom príklade sa stačilo iba jednoducho zbaviť mriežky, pri odčítavaní budeme musieť po páske prejsť viackrát. Za každé áčko napravo od mriežky budeme musieť jedno áčko naľavo „vymazať“. Ako to spravíme? Možností je viacero. Popíšem jednu: nájdeme mriežku, doprava za ňou je áčko. To vymažeme, ale aby sme vedeli, že za ním ešte môžu byť áčka, neprepíšeme ho na blank, ale na ďalšiu mriežku. Teraz sa budeme vracat doľava, kým nenarazíme na prvé áčko pred mriežkou. Toto opäť prepíšeme na mriežku – čím ho vymažeme, ale zároveň budeme vedieť, že pred ním ešte môžu byť nejaké áčka. Páska bude obsahovať slovo v tvare $a^i \#^j a^k$, čiže nejaké áčka, nejaké mriežky a nejaké zvyšné áčka, ktoré ešte musíme odčítavať. Aby sme mali lepšiu predstavu, čo sa deje, skúsme si odkrokovat, ako bude vyzerat naša páska na tomto príklade:

$$\zeta aaaaa \# aaa \sqcup \quad (1)$$

$$\zeta aaaaa \# \# aa \sqcup \quad (2)$$

$$\zeta aaaa \# \# \# aa \sqcup \quad (3)$$

$$\zeta aaaa \#\#\# a \sqcup \quad (4)$$

$$\zeta aaa \#\#\# \#\# a \sqcup \quad (5)$$

$$\zeta aaa \#\#\# \#\# \#\# \sqcup \quad (6)$$

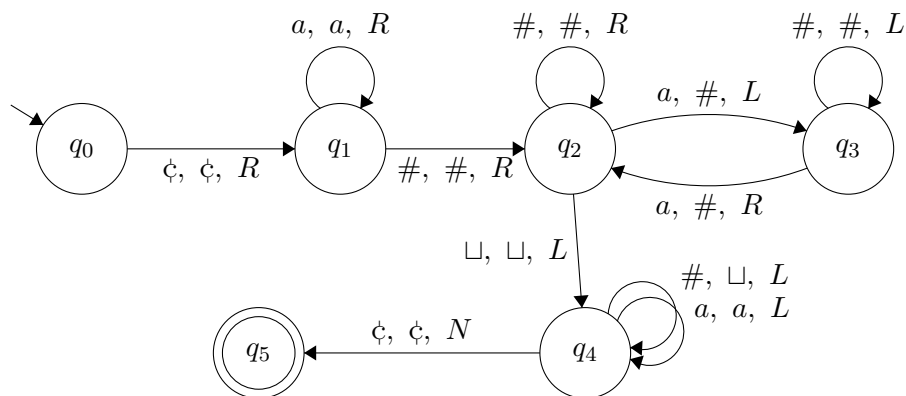
$$\zeta aa \#\#\# \#\# \#\# \#\# \sqcup \quad (7)$$

Ako budeme vedieť, že už sme všetky áčka za mriežkou odčítali? Keď budeme hľadať ďalšie a namiesto neho nájdeme blank, pretože to znamená, že už sme na konci nášho slova na páske. Počet áčok je už v poriadku, ale mali by sme sa zbaviť mriežok. Keďže aktuálne sme na prvom blanku, cestou dolava všetky mriežky prepíšeme na blanky, áčka necháme áčkami a celý výpočet skončíme, keď sa dostaneme na cent.

Čo myslíte, mohli by sme áčka prepisovať na blanky? Ak áno, prečo a ak nie, prečo? Vyskúšajte si upraviť tento TS, aby to robil. Treba ešte nejaké ďalšie úpravy? Viete potom jasne určiť, kedy už sme dokončili výpočet?

Viete vymyslieť aj iný spôsob, ako túto úlohu riešiť? Skúste pre oba spôsoby zistiť ich náročnosť – koľkokrát sa budete musieť presúvať po páske? Budú tieto prechody rovnako dlhé v oboch spôsoboch? Vyskúšajte si pre tento spôsob navrhnúť TS a odkrojujte ho na rovnakých slovách, aké skúsíte na našom príklade.

Riešenie na obrázku alebo na [stiahnite](#).



Príklad 1(e): na vstupe dostaneme n áčok a máme vrátiť dvakrát viac. Pre každé áčko na vstupe, tak potrebujeme dopísať na pásku ešte jedno. Znie to jednoducho, nájdeme áčko, nejako zaznačíme, že sme ho zarátali, ideme na koniec slova, pridáme tam a , vrátime sa a toto opakujeme. Ale ako zaručíme, že budeme dopisovať áčka iba za tie, ktoré tam boli už predtým a nie za tie, ktoré sme dopísali? Možno tak, že ich nebudeme priamo písať ako a , ale napríklad ako A .

Čiže, postup je nasledovný: prečítame prvé áčko za centom, prepíšeme ho na X . Prejdeme na prvý blank a napíšeme tam A . Vrátime sa po malých áčkach doľava, kým nenarazíme na X , z neho sa posunieme doľava na a , ktoré prepíšeme na X a posúvame sa po blank. Pozor, teraz bude na páske $\epsilon X^i a^j A^k \sqcup$, čiže ak sa chceme dostať na blank, budeme musieť prečítať aj nejaké veľké Áčka. Rovnaké platí aj pre spätné hľadanie X . Toto budeme opakovať dokedy? Pokiaľ sa z X neposunieme doprava a tam namiesto a bude A . Čo znamená, že sme už zdvojnásobili každé pôvodné áčko. Teraz potrebujeme iba všetky X a A na páske prepísať na a . Nájdem preto koniec slova (prvý blank), a odtiaľ pôjdeme doľava a budeme X aj A prepisovať na a .

Skúste si tento TS skonštruovať a porovnajte s tým mojím.

Myslíte si, že by postup fungoval aj keby sme pôvodné áčka neprepisovali na X ale na A ? Viete vymyslieť aj iný spôsob, ako toto spraviť? Skúste porovnať tieto prístupy. Ktorý bude efektívnejší?

2. príklad

Zostrojte Turingov stroj, ktorý vracia TRUE/FALSE (teda akceptuje alebo zamietá), podľa toho, či slovo na páske patrí do jazyka:

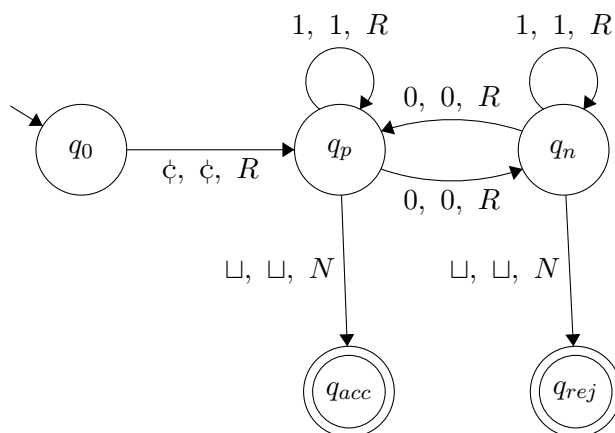
- (a) $L_1 = \{|w|_0 \bmod 2 = 0 \mid w \in \{0, 1\}^*\}$
- (b) $L_2 = \{w\#w \mid w \in \{a\}^*\}$
- (c)* $L_3 = \{w\#w \mid w \in \{a, b\}^*\}$

Riešenie 2. príkladu

Tieto TS sú už také, aké ste videli na prednáške. V niečom sa budú správať veľmi podobne ako naše konečné automaty, ale s tým, že TS sú výpočtovo silnejšie. Čím? No práve možnosťou si niečo na pásku zaznačovať. Poďme postupne.

Riešenie 2(a): jazyk L_1 je určite regulárny, čo znamená, že vieme vytvoriť DKA, ktorý ho bude rozoznávať. Ten vieme veľmi ľahko prerobiť na TS, v ktorom nepotrebujeme nič zapisovať. Zmena oproti DKA je tá, že na konci slova máme blank. A ten môže byť na konci slova z jazyka – vtedy slovo akceptujeme, takže pôjde do stavu q_{accept} , a ak slovo z jazyka nebude, skončí v stave q_{reject} .

Z obrázku vidíme, že sme museli pridať iba tieto dva stavy a vstup, kde sme prečítali cent. Vyskúšajte si riešenie.



Čo myslíte, vieme každý DKA prerobiť na TS? Vieme zostrojiť TS pre každý regulárny jazyk?

Riešenie 2(b): v tomto príklade už vidíme jazyk, ktorý nie je regulárny. Nevieme teda zostrojiť automat, ktorý by ho rozpoznával. Ukážme si, že pomocou TS to už nebude problém.

Chceme zistiť, či počet áčok pred mriežkou je rovnaký ako počet áčok za mriežkou. (Opäť, existuje veľa možných riešení, skúste potom vymyslieť nejaké iné.) Môj postup bude nasledovný – nájdem áčko tesne pred mriežkou, prepíšem ho na mriežku, a budem hľadať prvé áčko za mriežkou, ktoré tiež prepíšem na mriežku. Vrátim sa pred mriežku a pôjdem znovu. Vždy tak budem mať slovo v tvare áčka-mriežky-áčka. Toto budem opakovať dovtedy, kým sa mi áčka na jednej alebo druhej strane neminú. Keď sa minú na oboch stranách (čiže moje slovo sa bude skladať iba z mriežok), slovo akceptujem, ak na nejakej strane ešte áčko zostane (tj. buď $a^i \#^j$ alebo $\#^k a^m$), slovo zamietnem.

Vyskúšajte spustiť na riešení rôzne slová. Čo myslíte, všetky vetvy sú potrebné? Existujú tam stavy, do ktorých sa nikdy nedostanete? Odstráňte ich.

Riešenie 2(c): vedeli by sme upraviť TS z 2(b) tak, aby mohli byť slová nad abecedou $\{a, b\}$? Asi nebudeme môcť použiť rovnaký prístup, pretože napríklad pre slovo $ab\#ab$ by sme porovnávali a za mriežkou s b pred mriežkou, čo nie je dobrý prístup. Budeme preto musieť porovnávať písmená za centom s písmenami za mriežkou. Už sme spomínali, že v pekných riešeniach nič neprepisujeme na cent, preto písmená zo začiatku pásky budeme prepisovať napríklad na hviezdičku *. Budeme potrebovať aj dve vetvy – ak sme našli za centom (neskôr za hviezdičkami) a , a ak sme našli b . Podľa toho

budeme vedieť, aký znak hľadať za mriežkou (mriežkami). Zhrňme si to:

- nájdeme prvý znak za centom (neskôr hviezdičkou):
 - ak je to a alebo b , prepíšeme ho na hviezdičku a posúvame sa doprava až za mriežku:
 - * ak tam bude rovnaký znak, prepíšeme ho na mriežku a vraciame sa doľava, kým nenarazíme na hviezdičku, a opakujeme celý postup.
 - * ak tam bude iný znak alebo blank, môžeme rovno slovo zamietnuť.
 - ak je ďalší znak mriežka, znamená to, že na začiatku už žiadne písmená nemáme. Skontrolujeme, či na konci sú už iba mriežky:
 - * ak nájdeme a alebo b zamietame (za mriežkou sa nachádza dlhšie slovo ako pred ňou),
 - * ak nájdeme iba blanky, zjavne sú slová rovnaké a akceptujeme.

Vyskúšajme postup na pár slovách:

- 1) $\zeta abab\#abab\sqcup \rightarrow \zeta abab\#abab\sqcup \rightarrow \zeta * bab\#abab\sqcup \rightarrow \zeta * bab\#\#abab\sqcup$
 $\rightarrow \zeta * * ab\#\#abab\sqcup \rightarrow \zeta * * ab\#\#\#abab\sqcup \rightarrow \zeta * * * b\#\#\#abab\sqcup \rightarrow$
 $\zeta * * * b\#\#\#\#abab\sqcup \rightarrow \zeta * * * * \#\#\#\#abab\sqcup \rightarrow \zeta * * * * \#\#\#\#\#abab\sqcup \rightarrow$
 ACCEPT
- 2) $\zeta abab\#aba\sqcup \rightarrow \zeta abab\#aba\sqcup \rightarrow \zeta * bab\#aba\sqcup \rightarrow \zeta * bab\#\#ba\sqcup$
 $\rightarrow \zeta * * ab\#\#ba\sqcup \rightarrow \zeta * * ab\#\#\#a\sqcup \rightarrow \zeta * * * b\#\#\#a\sqcup \rightarrow \zeta * * * b\#\#\#\#a\sqcup$
 $\rightarrow \zeta * * * * \#\#\#\#a\sqcup \rightarrow$ REJECT
- 3) $\zeta aba\#abab\sqcup \rightarrow \zeta aba\#abab\sqcup \rightarrow \zeta * ba\#abab\sqcup \rightarrow \zeta * ba\#\#bab\sqcup$
 $\rightarrow \zeta * * a\#\#bab\sqcup \rightarrow \zeta * * a\#\#\#ab\sqcup \rightarrow \zeta * * * \#\#\#ab\sqcup \rightarrow \zeta * * * \#\#\#\#ab\sqcup$
 \rightarrow REJECT
- 3) $\zeta aba\#abb\sqcup \rightarrow \zeta aba\#abb\sqcup \rightarrow \zeta * ba\#abb\sqcup \rightarrow \zeta * ba\#\#bb\sqcup \rightarrow$
 $\zeta * * a\#\#bb\sqcup \rightarrow \zeta * * a\#\#\#b\sqcup \rightarrow \zeta * * * \#\#\#b\sqcup \rightarrow$ REJECT

V prvom slove skončíme, keď po hviezdičke vpravo bude mriežka. Prejdeme zvyšok pásky a keďže nenájdeme žiaden iný znak ako mriežku, slovo akceptujeme.

V druhom slove budeme hľadať béčko za mriežkami, ale nájdeme tam iba blank, čiže slovo pred mriežkou je dlhšie, zamietame.

V treťom slove skončíme, keď po hviezdíčke vpravo bude mriežka. Skontrolujeme zvyšok pásky a nájdeme na konci b . Slovo za mriežkou je dlhšie ako slovo pred mriežkou, zamietame.

V štvrtom slove budeme hľadať za mriežkou a , keďže to sme prepísali pred mriežkou posledné, nájdeme tam ale b . Slová nie sú rovnaké, preto zamietame.

Vyskúšajte si podľa tohto postupu zostrojiť TS, alebo si ho [stiahnite](#) a skontrolujte.

Dodatočné linky

- Motivačné/fun video - <https://www.youtube.com/watch?v=cYw2ewo06c4>
- JFLAP - program spolu s príkladmi na automaty, gramatiky (pozrite si regulárnu), pumpovaciu lému, Turingove stroje a mnoho iného. <http://www.jflap.org/modules/>
- Návrhy (vždy jedno z možných riešení) turingových strojov z príkladov v programe JFLAP si viete otvoriť, upravovať, simulovať výpočty. Pozor, ϵ je nahradený c a namiesto nášho N na to, že čítacia hlava stojí, používajú S . Tiež používajú obojsmerne nekonečnú pásku, preto ak chcete simulovať výpočet na slove, musí začínať c (centom).

Poznámky

Obyčajné Turingove stroje

- Pri konštrukcii Turingovho stroja si najprv rozmyšľajte celú ideu toho, ako bude pracovať a načrtnite ju. Postupne riešte „dobré“ slová, vysvetlite, ako bude páska vyzeráť po aplikovaní navrhnutých akcií, ako na to bude váš TS reagovať, aké máte krajné prípady a kde ich ošetríte, ako zistíte, že je slovo z jazyka (t.j. kedy sa dostane do akceptačného stavu)... Zamyslite sa, či ste vyriešili naozaj všetky možnosti. Ak niečo predpokladáte, napíšte to (napr. *na vstupe je korektné slovo* a pod.).
- TS je vlastne algoritmus, alebo program. Popíšte ho preto tak, aby si čitateľ vedel zostrojiť vlastný TS iba podľa vášho opisu.
- Ak medzi dvoma stavmi (alebo na jednej slučke) viac prechodov rovnakého typu, napr. $a, a, R, b, b, R, c, c, R$, môžete ich napísať ako jeden použitím substitúcie. Na prechod napíšete x, x, R a k nemu $x \in \{a, b, c\}$.

Pre prechody $0, \sqcup, L$ a $1, \sqcup, L$ sa dá zas použiť substitúcia y, \sqcup, L , kde $y \in \{0, 1\}$ atď.

- Vhodne si pomenujte stavy, označte si časti TS, ak to potrebujete, a popisujte ich zvlášť. Ak používate substitúciu, buďte jednotní v celom stroji a napíšte ju niekam na viditeľné miesto (sprehľadní to váš diagram).
- TS v každom stave niečo z pásy číta, niečo na ňu zapisuje a hýbe sa vpravo, vľavo alebo stojí. Determinizmus pri TS určuje to, že v prípade, že v stave q prečítame symbol e , vieme jednoznačne určiť, aký symbol zapíšeme a kam sa pohneme. Zjavne teda nevieme mať z jedného stavu zároveň takéto prechody e, e, R a e, e, L , či e, a, R .
- Ak sa raz do akceptačného stavu dostaneme, TS slovo akceptuje bez ohľadu na to, či sme ho prečítali celé alebo nie (t.j. z akceptačného stavu sa nevieme dostať preč). To isté platí aj pre zamietací stav.
- Cent nemažeme, neprepisujeme na nič iné a nič iné neprepisujeme na cent. Páska je nekonečná iba jedným smerom (doprava), ľavý koniec pásy je označený centom. Za vstupným slovom nasleduje nekonečno blankov.

Pri domácich úlohách a skúškach je dôležitá najmä jasná idea celého TS. Tj. postup, ako pracuje, čo kde sa má stať, kedy akceptuje, kedy zamietá... Takže popis algoritmu. Potom, ak sa aj pomýlite v prechodovej funkcii, je to menší problém, akoby ste napísali iba prechodovú funkciu a nenapísali myšlienku algoritmu. Opäť, nezabúdajte na vhodné opisy stavov či častí TS (na ktoré sa môžete odkazovať v popise), a nezabudnite spraviť definíciu TS (sedmica z prednášky). Prechodovú funkciu máte už priamo v diagrame, ak ho nakreslíte. Alebo ju môžete iba napísať.