

Řešení úlohy č. 5

Kuličkový počítač #1

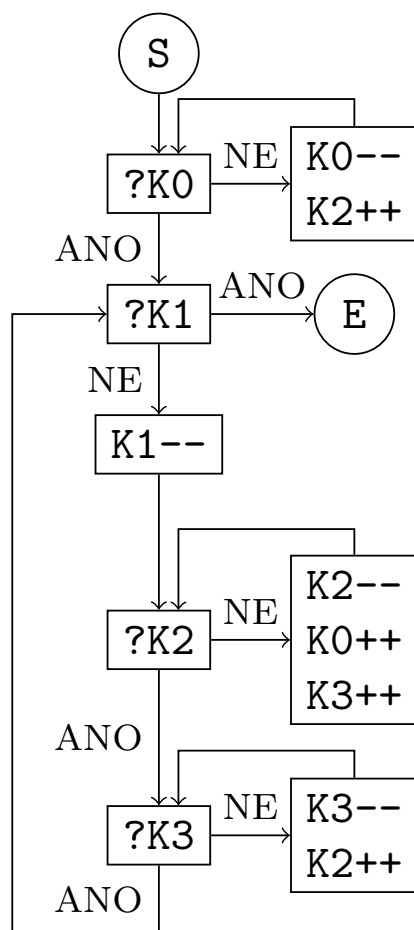
Za úkol jsme měli navrhnout tři programy pro kuličkový počítač, kde každý z nich implementovat jednu ze zadaných aritmetických operací. Řešení si tak rozebereme postupně po jednotlivých operacích.

Program pro výpočet součinu dvou čísel

Řešení této operace vyplývá jednoduše z definice součinu – máme-li vynásobit čísla a a b , jedná se vlastně o postupné přičítání čísla a celkem b -krát (a nebo naopak, bez újmy na obecnosti však uvažujeme tuto posloupnost přičítání). Můžeme tak rovnou přistoupit k návrhu programu pro kuličkový počítač, kde ovšem musíme vyřešit několik situací specifických pro model kuličkového počítače.

Jako první je zapotřebí si uvědomit, že lze jednoduše přesunout obsah kyblíčku K_i do libovolného kyblíčku K_j , kde $i \neq j$. Jednoduše se stačí opakovaně ptát na prázdnotu kyblíčku K_i a provádět operace odečtení z K_i a přičtení k K_j . Tato operace však pro řešení našeho problému nestačí, neboť pokud bychom přesunuli obsah kyblíčku reprezentující číslo a do výsledkového kyblíčku, ztratili bychom informaci o hodnotě a , neboť obsah výsledkového kyblíčku nemůžeme měnit. Řešení je však nasnadě – při přesouvání obsahu kyblíčku K_i do kyblíčku K_j stačí navíc přidávat kuličky i do kyblíčku K_k , $i \neq j \neq k$. Tímto způsobem jsme schopni obsah kyblíčku K_i rozkopírovat do dvou jiných kyblíčků. Pokud poté budeme chtít opakovaně přesouvat obsah kyblíčku K_i do kyblíčku K_j (což je přesně náš případ u násobení, viz výše), stačí při přesouvání provádět zároveň duplikaci do kyblíčku K_k a po přesunutí všech kuliček z K_i jednoduše vrátit obsah kyblíčku K_k do kyblíčku K_i .

Jelikož jsme si popsali způsob opakovaného přesouvání kuliček, je již jednoduché sestavit řešení v jazyce kuličkového počítače. Na začátku máme $K_0 = a$ a $K_1 = b$. Uvolníme kyblíček K_0 pro výsledek a jeho obsah přesuneme do kyblíčku K_2 . Nyní stačí provádět následující posloupnost operací: dokud není kyblíček K_1 prázdný, postupně kopírujeme obsah kyblíčku K_2 do kyblíčku K_0 za využití pomocného kyblíčku K_3 a za každé takové překopírování odstraníme z kyblíčku K_1 jednu kuličku. Dle obecného popisu výše tedy zvolíme $i = 2$, $j = 0$ a $k = 3$. Výsledná podoba programu pro součin dvou čísel je následující:



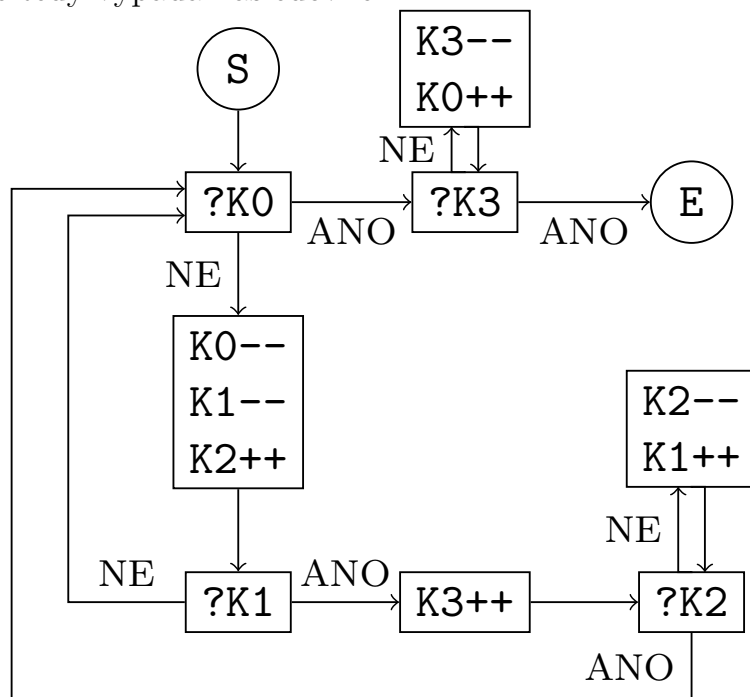
Program pro výpočet celočíselného podílu dvou čísel

Pro návrh této operace nahlédneme na definici z níž vychází podíl – pro nezáporná celá čísla a, b platí, že $a = kb + l$, kde $0 \leq l < b$. Číslo k je celočíselným podílem čísel a a b a neformálně řečeno udává kolikrát se číslo b "vleze" do čísla a . A přesně podle tohoto neformálního popisu budeme operaci implementovat, tj. postupně budeme odečítat číslo b od čísla a a poznačíme si, kolikrát se nám to povedlo. Opět však musíme vyřešit některé vzniklé kuličkopočítačové situace.

Na začátek je zapotřebí si uvědomit, že nelze dělit nulou, a tedy pro $b = 0$ nedává náš program vůbec smysl. To můžeme v kuličkovém počítači ošetřit podmíněným ukončením programu, ale pro jednoduchost popisu uvažujme, že na začátku vykonávání programu platí $b > 0$. Další situace, která vyvstává, je prakticky opačná vůči situaci u sčítání. Budeme totiž postupně odečítat kuličky z kyblíčků K_i a K_j (K_i bude odpovídat číslu a a K_j číslu b), po dobu dokud se kyblíček K_i nevyprázdní tak, že K_j bude opakovaně nastavováno na výchozí počet kuliček. Pokud je v kyblíčku K_i více kuliček než v kyblíčku K_j , jistě budeme odečítat několikrát, a musíme si tak ukládat informaci o tom, kolik bylo v kyblíčku K_j kuliček, abychom odečítání mohli zopakovat. To provedeme tak, že si počet kuliček v K_j během odečítání budeme zároveň počítat v kyblíčku K_k , kde opět $i \neq j \neq k$. Pro nastavení kyblíčku K_j do výchozího stavu poté stačí překopírovat obsah kyblíčku K_k zpět do kyblíčku K_j .

Dle popisu výše pro postupné odečítání zvolíme např. $i = 0, j = 1$ a $k = 2$. Pro počáteční konfiguraci $K_0 = a$ a $K_1 = b$ tedy vskutku budeme postupně odečítat b od a . Zároveň si do jiného kyblíčku K_3 budeme značit počet provedených odečítání, tj. počet toho, kolikrát jsme odečetli celý obsah kyblíčku K_1 od K_0 a vrátili jsme jej na hodnotu b . Pokud by kdykoliv během odečítání (a to i před jeho započítáním) došlo k tomu, že číslo v kyblíčku K_0 je nulové, budeme mít v kyblíčku K_3 výsledný celočíselný podíl a program můžeme ukončit. Nesmíme

však ještě zapomenout překopírovat výsledek z kyblíčku K_3 do kyblíčku K_0 . Program v jazyce kuličkového počítače tedy vypadá následovně:



Program pro výpočet zbytku po celočíselném dělení dvou čísel

Pro návrh tohoto programu si stačí opět rozepsat vztah pro dělení dvou čísel, kde $a = kb + l$. U programu pro dělení jsme záměrně přeskočili číslo l , které je zbytkem po celočíselném dělení, a značí tedy výsledek, jehož chceme tímto programem dosáhnout. Stačí si však uvědomit, že vzhledem k předchozímu návrhu programu je toto číslo vlastně počet odečtených kuliček z kyblíčku K_j (značícího číslo b) za situace, kdy je počet kuliček v kyblíčku K_i (značící průběžný stav čísla a) nulový. Tento počet jsme si však ukládali do kyblíčku K_k ! Zároveň je program pro dělení navržen tak, že dělí-li číslo b číslo a beze zbytku, (tj. je-li někdy během běhu programu počet kuliček v kyblíčku K_k roven b a K_i je prázdný), dojde ještě před ukončením programu k přidání kuličky do výsledkového kyblíčku a k zpětnému přesunu obsahu kyblíčku K_k do kyblíčku K_j . Kyblíček K_k tak bude i v tomto případě obsahovat korektní výsledek, a to nula kuliček.

Pro volbu $i = 0, j = 1$ a $k = 2$ tedy stačí použít pro tuto operaci naprosto identický program jako pro celočíselné dělení, pouze se před ukončením programu zkopíruje do kyblíčku K_0 obsah kyblíčku K_2 (namísto kyblíčku K_3).