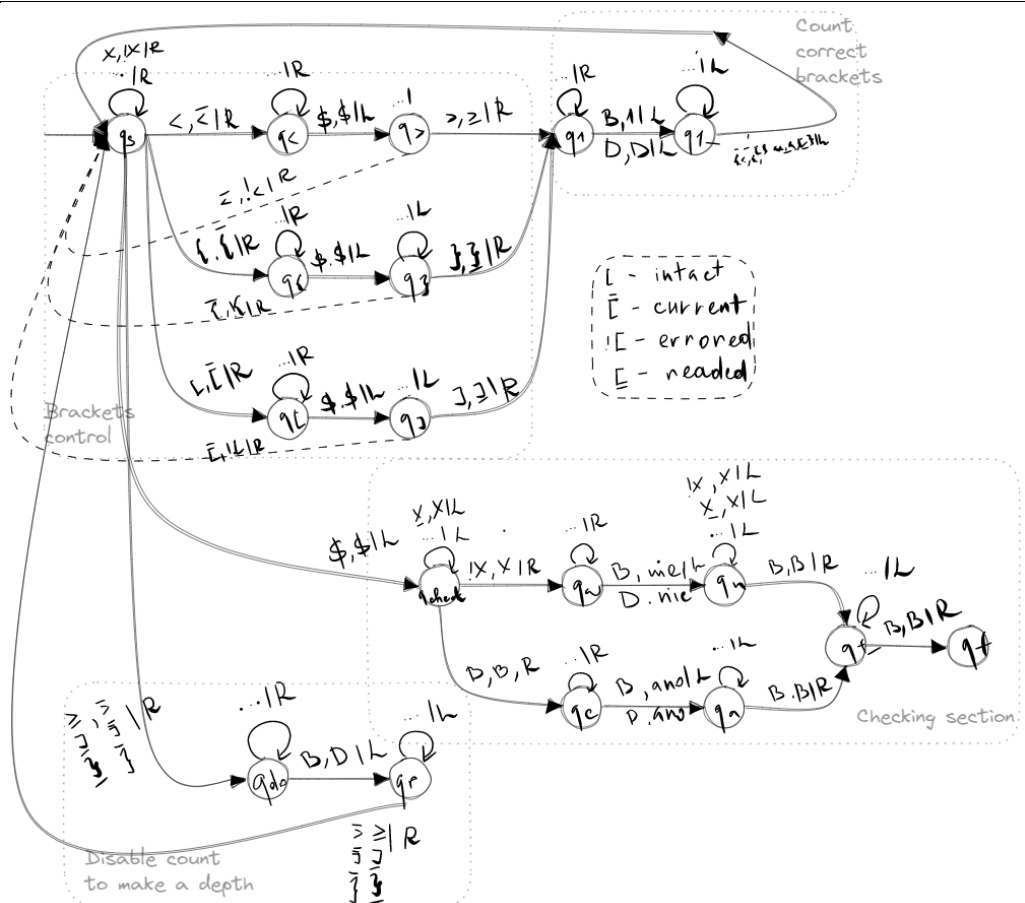


Meno:	Artur Kozubov	Hodnotenie projektu: (max 10(TS)/5(RAM) bodov)
Cvičenie:	Pondelok 18:00, J. Petřík	
Dátum:	24. 4. 2023	

Projekt TZIV LS2022/23 – TS

Zadanie:	Na vstupe je aritmetický výraz `(+, -, *, /, (,), [,], {, },)` , ohraničený oddeľovačmi `\$`. Navrhnete Turingov stroj - riešenie, ktoré vypíše v unárnej sústave hĺbku najviac vnoreného výrazu (počítajú sa vnorenia prostredníctvom zátvoriek `()` a `[]` a `{}`) a reťazec "ano" v prípade, že počet, poradie a typ zátvoriek sú korektné alebo "nie" ak nie sú. V prípade, že počet, poradie a typ zátvoriek nie sú korektné, hĺbku vnorenia počítajte zľava.
Vstup:	Zmenené zátvorky kvôli nemožnosti pracovať s nimi v SimStudio Akceptované vstupy (čísla a matematické znaky - zámerne vynechané) : <ul style="list-style-type: none"> - $10+5*<10*[9+8]> \\$ - $<>[]\langle\rangle \\$ - $<<>> \\$ - $<[[]]\langle\rangle> \\$ Neakceptované vstupy (čísla a matematické znaky - zámerne vynechané) : <ul style="list-style-type: none"> - $10+5*<10*[9+8]>> \\$ - $>>[]\rangle\rangle \\$ - $<]> \\$ - $<[><]> \\$

Neformálne riešenie:



* Graf, ktorý minimálne reprezentuje štruktúru a logiku môjho automatu. Kvôli pohodliu je tiež rozdelený do sekcií.

Najprv sa vyhladá prvá otvorená zátvorka na ľavej strane qS a potom najbližšia uzavretá zátvorka rovnakého typu z pravej strany $qI, qI, q>$. Tá sa potom počíta ako jedna hĺbková vrstva $qI, qI_$.

Ak stroj pri hľadaní "dvojíc" dostane neplatnú hodnotu, bude ju považovať za chybnú *Bodkované šípky*, aby ju na konci mohol kontrolór $qCheck$ zohľadniť pri konečnom vyhodnotení.

Je tiež špeciálne navrhnutý tak, aby zohľadňoval viacero párov zátvoriek v jednej zátvorke, a to práve preto, aby sa hodnota hĺbky nezvyšovala qDO (*Disable output*).

Zložitosť riešenia:

Problémom pri riešení bola kontrola a nájdenie identickej dvojice zátvoriek. Celková časová zložitosť riešenia rastie vzhľadom na počet zátvoriek vo výraze.

Simulátor:

SimStudio

```

ignored = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, a, n, o, i, e}

Q0 = {>, ], >>}
!Q0 = {!>, !], !>>}
f(q0, Q0) = (q0, !Q0, R)
f(q0, ignored) = (q0, ignored, R)

f(q0, <) = (q<, @<, R)
f(q0, [) = (q[, @[, R)
f(q0, «) = (q«, @«, R)

f(q0, $) = (qCheck, $, L)

QDO = {>_, ]_, >>_}
@QDO = {@>, @], @>>}
f(q0, QDO) = (qDO, @QDO, R)

brackets = {<, >, >_, !>, [, ], ]_, «, », >>}
wrong = {<, [, «}
!wrong = {!<, ![, !«}
ignoredBrackets = {>_, ]_, >>_, !<, !>, ![, !], !«, !>>}

f(q<, brackets) = (q<, brackets, R)
f(q<, ignored) = (q<, ignored, R)
f(q<, $) = (q>, $, L)

Q> = {], >>}
!Q> = {!], !>>}
f(q>, @<) = (q0, !<, R)
f(q>, wrong) = (q>, wrong, L)
f(q>, Q>) = (q>, Q>, L)
f(q>, >) = (q1, >_, R)
f(q>, ignoredBrackets) = (q>, ignoredBrackets, L)

f(q[, brackets) = (q[, brackets, R)
f(q[, ignored) = (q[, ignored, R)
f(q[, $) = (q], $, L)

Q] = {>, >>}
!Q] = {!>, !>>}
f(q], @[] = (q0, ![, R)
f(q], wrong) = (q], wrong, L)
f(q], Q]) = (q], Q], L)
f(q], ]) = (q1, ]_, R)
f(q], ignoredBrackets) = (q], ignoredBrackets, L)

f(q«, brackets) = (q«, brackets, R)
f(q«, ignored) = (q«, ignored, R)
f(q«, $) = (q», $, L)

Q» = {], >}
!Q» = {!], !>}
f(q», @«) = (q0, !«, R)
f(q», wrong) = (q», wrong, L)
f(q», Q») = (q», Q», L)
f(q», >>) = (q1, >>_, R)
f(q», ignoredBrackets) = (q», ignoredBrackets, L)

```

```

Q1 = {<, >, [, ], «, », !<, !>, ![, !], !«, !», <_, >_, [_], ]_, «_, »_, $}
f(q1, ignored) = (q1, ignored, R)
f(q1, Q1) = (q1, Q1, R)
f(q1, Blank) = (q1_, 1, L)
f(q1, D) = (q1_, D, L)

@Q1_ = {@<, @[, @«}
Q1_ = {<_, [_], «_}
f(q1_, Q1) = (q1_, Q1, L)
f(q1_, ignored) = (q1_, ignored, L)
f(q1_, @Q1_) = (q0, Q1_, R)

QCheck = {<, >, [, ], «, »}
QCheck_ = {<_, >_, [_], ]_, «_, »_}
!QCheck = {!<, !>, ![, !], !«, !»}
f(qCheck, QCheck_) = (qCheck, QCheck, L)
f(qCheck, ignored) = (qCheck, ignored, L)
f(qCheck, !QCheck) = (qN, !QCheck, R)
f(qCheck, Blank) = (qA, Blank, R)

f(qDO, ignored) = (qDO, ignored, R)
f(qDO, Q1) = (qDO, Q1, R)
f(qDO, Blank) = (qR, D, L)
f(qDO, D) = (qR, D, L)
f(qR, Q1) = (qR, Q1, L)
f(qR, ignored) = (qR, ignored, L)
f(qR, @QDO) = (q0, QDO, R)

f(qN, QCheck) = (qN, QCheck, R)
f(qN, ignored) = (qN, ignored, R)
f(qN, $) = (qN, $, R)
f(qN, Blank) = (qi, n, R)
f(qN, D) = (qi, n, R)
f(qi, Blank) = (qe, i, R)
f(qe, Blank) = (qf_, e, L)

f(qA, QCheck) = (qA, QCheck, R)
f(qA, ignored) = (qA, ignored, R)
f(qA, $) = (qA, $, R)
f(qA, Blank) = (qn, a, R)
f(qA, D) = (qn, a, R)
f(qn, Blank) = (qo, n, R)
f(qo, Blank) = (qf_, o, L)

Qf = {<, >, [, ], «, », <, >, [, ], «, », <, >, [, ], «, », $}
f(qf_, ignored) = (qf_, ignored, L)
f(qf_, Q1) = (qf_, Qf, L)
f(qf_, Blank) = (qf, Blank, R)

```