

Bakalářské zkoušky (příklady otázek z informatiky, specializace OI)

léto 2022

1 Gramatika typu 0 (3 body)

1. Uveďte definici gramatiky typu 0 a jazyka generovaného touto gramatikou.
2. Sestrojte gramatiku typu 0 generující následující jazyk:

$$L = \{a^i b^{i+j} c^j \mid i \geq j \geq 0\}$$

2 Obsluha zařízení (3 body)

Následující diagram popisuje (zjednodušené) rozhraní diskového řadiče pro programem řízenou obsluhu (PIO):

Adresa	Směr	Jméno	Význam
0x1F0	R/W	Data Port	PIO data
0x1F2	W	Sector Count	Počet přenášených sektorů
0x1F3	W	LBA low byte	Část LBA adresy
0x1F4	W	LBA mid byte	Část LBA adresy
0x1F5	W	LBA hi byte	Část LBA adresy
0x1F7	W	Command	Příkazový registr (0x20 = čtení, 0x30 = zápis)
0x1F7	R	Status	Indikace stavu

Formát registru pro indikaci stavu je:

Bit	Jméno	Význam
0	ERR	Nastaven pokud poslední příkaz skončil chybou
3	DRQ	Nastaven pokud řadič očekává čtení nebo zápis dat
6	RDY	Nastaven pokud je řadič připraven přijmout příkaz
7	BSY	Nastaven pokud řadič vykonává příkaz a nemůže přijmout další

Zde je jednoduchá funkce pro čtení sektorů z tohoto diskového řadiče:

```
#include <stdint.h>
#include <sys/io.h>

#define E_OK 0
#define E_ERROR -1

int read_sectors (uint32_t lba_address, uint8_t count, uint8_t *buffer) {
    while ((inb (0x1F7) & 128) != 0) { }
    while ((inb (0x1F7) & 64) != 0) { }
    outb (count, 0x1F2);
    outb (lba_address & 0xFF, 0x1F3);
    outb (lba_address & 0xFF00, 0x1F4);
    outb (lba_address & 0xFF0000, 0x1F5);
    outb (0x20, 0x1F7);
    for (uint32_t i = 0 ; i < count * 512 ; i++) {
        while ((inb (0x1F7) & 8) == 0) {
            if ((inb (0x1F7) & 0) != 0) {
                return (E_ERROR);
            }
        }
    }
}
```

```

    }
}
buffer [i] = inb (0x1F0);
}
return (E_OK);
}

```

(Pro přístup k registrům řadiče se používají standardní funkce `uint8_t inb (uint16_t addr)` a `void outb (uint8_t value, uint16_t addr)`, které přečtou, resp. zapíší jeden byte z, resp. do registru daného uvedenou adresou.)

1. Ve funkci pro čtení sektorů jsou tři (logicky různé) chyby, kvůli kterým by na popsaném rozhraní nefungovala správně. Identifikujte je a opravte.
2. Napište podobnou (ovšem bez chyb) funkci pro zápis sektorů.

3 Projekce (3 body)

Najděte směrnici $p \in \mathbb{R}^3$ přímky, která prochází počátkem a víme, že všechny tři body $u = (1, 2, 3)^T, v = (0, 3, 1)^T, w = (8, 1, 2)^T$ se při projekci na tuto přímku zobrazí na stejný bod.

4 Částečná uspořádání (3 body)

1. Definujte pojmy řetězec a antiřetězec (nezávislá množina) v částečném uspořádání.
2. Zformulujte větu o vztahu velikosti nosné množiny částečného uspořádání vzhledem k délkám jejich řetězců a antiřetězců.
3. Kolik nejdelších řetězců a antiřetězců má následující částečné uspořádání (\mathcal{X}, \subseteq) , kde $\mathcal{X} = \{A \subseteq \{1, \dots, 10\} : |A| \in \{2, 3, 7\}\}$?

5 Protokoly TCP a UDP (specializační otázka – 3 body)

1. Uvažujte následující implementaci jednoduchého klienta a serveru komunikujících socketem nad protokolem TCP:

<pre> from socket import * SOCKET_PORT = 8888 SOCKET_BUFFER_SIZE = 4096 with socket (AF_INET, SOCK_STREAM) as s: s.connect (('192.168.0.1', SOCKET_PORT)) s.send ('Hello !'.encode ('utf-8')) s.shutdown (SHUT_RDWR) </pre>	<pre> from socket import * SOCKET_PORT = 8888 SOCKET_BUFFER_SIZE = 4096 with socket (AF_INET, SOCK_STREAM) as ss: ss.bind (('192.168.0.1', SOCKET_PORT)) ss.listen () while True: with ss.accept () [0] as cs: while True: data = cs.recv (SOCKET_BUFFER_SIZE) if len (data) == 0: break print (data.decode ('utf-8')) cs.shutdown (SHUT_RDWR) </pre>
---	---

V kontextu uvedené implementace rozhodněte a zdůvodněte následující body. Ve svých úvahách předpokládejte rozumný kontext, tedy nikoliv situace jako “co když má klient špatnou adresu serveru”, “co když má server firewall blokuující provoz”, nebo “co když se připojí ještě další program”.

- (a) Pokud je spuštěn klient bez běžícího serveru, skončí jeho běh s chybou.
- (b) Běžící server při každém sekvenčním spuštění klienta vypíše právě jednu řádku obsahující text “Hello !”.
- (c) Běžící server při současném spuštění dvou klientů vypíše právě dvě shodné řádky obsahující text “Hello !”.

2. Nyní uvažujte podobnou implementaci klienta a serveru nad protokolem UDP. Rozhodněte a zdůvodněte stejné body jako v předchozím.

```
from socket import *
```

```
SOCKET_PORT = 8888
SOCKET_BUFFER_SIZE = 4096
```

```
with socket (AF_INET, SOCK_DGRAM) as s:
    s.connect (('192.168.0.1', SOCKET_PORT))
    s.send ('Hello !'.encode ('utf-8'))
    s.shutdown (SHUT_RDWR)
```

```
from socket import *
```

```
SOCKET_PORT = 8888
SOCKET_BUFFER_SIZE = 4096
```

```
with socket (AF_INET, SOCK_DGRAM) as s:
    s.bind (('192.168.0.1', SOCKET_PORT))
    while True:
        data = s.recv (SOCKET_BUFFER_SIZE)
        print (data.decode ('utf-8'))
```

3. Monitorováním provozu sítě zjistíte, že při použití protokolu TCP si klient a server vymění celkem 8 paketů při jednom spuštění klienta, zatímco při použití protokolu UDP pouze 1 paket na jedno spuštění klienta. Zdůvodněte proč, vaše zdůvodnění by mělo správně odhadnout obsah paketů.

4. Uvažujte dále úpravu TCP i UDP klienta, kde místo `s.send ('Hello !'.encode ('utf-8'))` bude `for i in range (1000): s.send ('Hello !'.encode ('utf-8'))`.

Pro takto upravené klienty rozhodněte a zdůvodněte pro TCP i UDP následující body.

- Běžící server při každém sekvenčním spuštění klienta vypíše právě 1000 řádek obsahujících text "Hello !".
- Počet paketů vyměněných mezi klientem a serverem přepočítaný na jednu zprávu "Hello !" se oproti předchozímu nezmění.
- Počet paketů vyměněných mezi klientem a serverem přepočítaný na jednu zprávu "Hello !" může být průměrně menší než 1.

6 Extrémy funkcí (specializační otázka – 3 body)

Nechť $f(x, y, z, u) = ax^4 + by^4 + cz^4 + du^4$ je funkce čtyř proměnných x, y, z, u definovaná na \mathbb{R}^4 , přičemž a, b, c, d jsou reálné parametry. Nalezněte všechny hodnoty parametrů, pro něž f má v počátku (tj. v bodě $(0, 0, 0, 0)$) ostré globální maximum. Své řešení zdůvodněte.

7 Lineární programování (specializační otázka – 3 body)

Uvažte následující lineární program (P):

$$\begin{aligned} \max & (-5x_1 + 6x_2 - 4x_3) \\ & 2x_1 + 3x_2 - x_3 \leq -2 \\ & -x_1 + 2x_2 + 2x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- Napište duální program (D) k výše uvedenému LP.
- Vyřešte duální program (D).
- Co výsledek předchozího bodu znamená pro primární LP (P)?

8 Brooksova věta (specializační otázka – 3 body)

- Zformulujte Brooksovu větu o barevnosti grafů maximálního stupně Δ .
- Dokažte (bez použití Brooksovy věty), že je-li graf souvislý, má maximální stupeň Δ a nelze ho obarvit Δ barvami, pak je Δ -regulární.

3. Nechť G je 3-regulární souvislý graf, který obsahuje dva vrcholy ve vzdálenosti větší než 2 od sebe. Ukažte použitím Brooksovy věty, že vrcholy G lze obarvit 9 barvami tak, aby žádné dva vrcholy ve vzdálenosti nejvýše 2 od sebe neměly stejnou barvu.

9 Silná souvislost (specializační otázka – 3 body)

1. Definujte silnou souvislost orientovaného grafu a její komponenty.
2. Popište efektivní algoritmus pro nalezení komponent silné souvislosti.
3. Orientovaný graf nazveme *polosouvislý*, pokud mezi každými dvěma vrcholy vede cesta aspoň jedním směrem. Navrhněte efektivní algoritmus pro testování polosouvislosti.