

DÁTOVÉ ŠTRUKTÚRY A ALGORITMY

ZADANIE č.4 : „Primitívna umelá inteligencia“

Program ai.cpp

Abstraktný dátový typ binary tree (celkom 10 bodov)

V programovacom jazyku C++ naprogramujte a otestujte primitívnu umelú inteligencia. Pomocou svojej implementácie demonštrujte pochopenie využívania tried v jazyku C++ a abstraktného dátového typu binary tree zrealizovaného prostredníctvom zreťazeného zoznamu.

POPIS ZADANIA

Program, ktorý sa „učí“ kladením otázok používateľovi, predstavuje základný problém vytvárania umelej inteligencie. Ako svoju hlavnú dátovú štruktúru program využíva binárny rozhodovací strom (binary decision tree). Uzly (nodes) stromu majúce potomkov obsahujú vo svojej dátovej časti „áno/nie“ otázky. Používateľ si myslí nejaký objekt a na základe jeho charakteristík odpovedá na otázky programu.

Program na začiatku činnosti položí otázku nachádzajúcu sa v dátovej časti koreňového uzla stromu. Ak používateľ odpovie nie/áno, program sa posunie na ľavého/pravého potomka koreňového uzla a pokladá otázku v jeho dátovej časti.

Tento proces sa opakuje a program sa pohybuje v rozhodovacom strome smerom dole, až sa dopracuje k „listovému“ uzlu (leaf node). „Listové“ uzly nemajú potomkov, v dátovej časti nesú názov objektu zodpovedajúceho sekvencii odpovedí používateľa.

Keď program dosiahne „listový“ uzol, nepokladá otázku, ale vypíše výsledný objekt. Ak sa objekt zhoduje s mysleným objektom používateľa, program „preukázal svoju inteligenciu“. Ak sa objekty líšia, program rozvíja svoju vedomostnú základňu rozširovaním svojho rozhodovacieho stromu. Požaduje od používateľa zadanie mysleného objektu a „áno/nie“ otázky, ktorá umožňuje odlíšenie objektu používateľa od objektu programu. Program následne túto otázku umiestni na vhodné miesto svojho rozhodovacieho stromu.

S narastajúcim počtom „hier“ narastá aj schopnosť programu rozpoznávať rôzne objekty.

FUNKCIONALITA

V rámci interakcie s používateľom zahrňte nasledujúce príkazy:

- a** – vytvorenie „základného“ rozhodovacieho stromu pozn. otázku koreňového uzla a dve možné odpovede môžete implementovať priamo v špecifikácii ADT binárneho stromu
- b** – načítanie rozhodovacieho stromu zo súboru
- c** – spustenie procesu učenia programu
- d** – export aktuálneho rozhodovacieho stromu do súboru
- x** – ukončenie činnosti programu

Pri spustení program vypíše používateľské menu.

Po vytvorení základného/načítaní rozhodovacieho stromu program umožní spustiť proces učenia. **Proces učenia a exportu rozhodovacieho stromu sa dá spustiť až po vytvorení**

„základného“/načítaní rozhodovacieho stromu zo súboru. Proces učenia začína kontrolným výpisom: *myslite na nejaký objekt. odpovedajte na moje otázky.*

Následne program pokladá otázky z dátovej časti uzlov rozhodovacieho stromu: *otazka_z_uzla? (Y/N)*

Popísaný proces sa opakuje až kým program vo svojom rozhodovacom strome na základe odpovedí používateľa nenarazí na „listový“ uzol. V tomto prípade vypíše: *mysleli ste na objekt objekt_z_listoveho_uzla? (Y/N)*

Ak používateľ odpovie Y, program vypíše kontrolné hlásenie *SOM INTELLIGENTNY* a čaká na zadanie používateľskej voľby z interaktívneho menu.

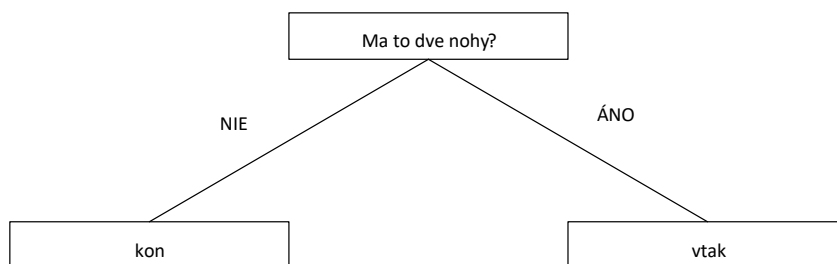
Ak používateľ odpovie N, program vyžaduje zadanie dodatočných informácií: *na aký objekt ste mysleli? prosím zadajte otázku, na ktorú odpoviete ano pre váš objekt a nie pre môj objekt*

DÁTOVÉ ŠTRUKTÚRY

Vyžaduje sa dôsledné používanie tried v C++ a hlavičkových súborov pri spracovaní vstupov z klávesnice, resp. pre vstup/výstup do textového súboru. Hlavnou dátovou štruktúrou programu bude ADT binárny strom (binary tree) implementovaný prostredníctvom zreťazeného zoznamu, ako bol špecifikovaný na prednáške/cvičení. Jeho špecifikácia sa bude nachádzať v hlavičkovom súbore **bintree.h** a jeho implementácia v súbore **bintree.cpp**. Dátová (*private*) časť binárneho stromu bude implementovaná použitím smerníkov a dynamickej pamäte. Pri práci s binárnym stromom sa musia používať výhradne operácie binárneho stromu špecifikované v súbore **bintree.h**. Program musí byť funkčný, a zároveň riadne otestovaný v rámci svojho main-programu t.j. testovacieho driveru **ai.cpp**.

PRÍKLAD INTERAKCIE

V textovom súbore je uložený nasledovný rozhodovací strom(napr. v „serializovanej“ podobe):



Používateľ myslí na rybu.

a – vytvorenie „základného“ rozhodovacieho stromu

b – načítanie rozhodovacieho stromu zo súboru

c – spustenie procesu učenia programu

d – export rozhodovacieho stromu do súboru

x – ukončenie činnosti programu

USER: b

AI: *rozhodovací strom nacistany*

USER: c

AI: *myslite na nejaky objekt. odpovedajte na moje otazky.*

AI: *ma to dve nohy?(y/n)*

USER: n

AI: *mysleli ste na objekt kon?*

USER: n

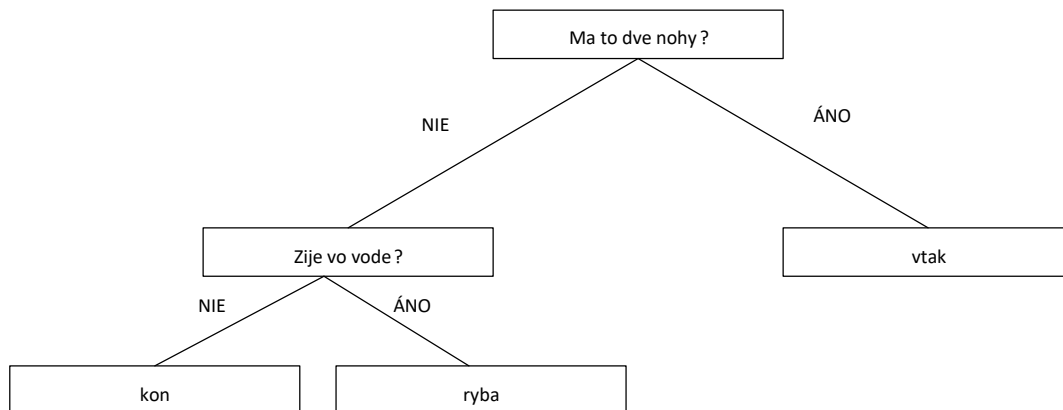
AI: *na aky objekt ste mysleli?*

USER: ryba

AI: *prosim zadajte otazku, na ktoru odpoviete ano pre vas objekt a nie pre moj objekt*

USER: *zije vo vode?*

(program tu zodpovedajúcim spôsobom rozšíri svoj rozhodovací strom)



AI: *ucenie dokoncene*

USER: c

AI: *ma to dve nohy?(y/n)*

USER: n

AI: *zije vo vode?(y/n)* USER:

y

AI: *mysleli ste na objekt ryba?*

USER: y

AI: *SOM INTELLIGENTNY.*

USER:d

AI: *rozhodovací strom exportovany*

ODOVZDÁVANIE

- termín odovzdania: **24.4.2024**

Odobzdajte odladené zdrojové kódy (*bintree.h*, *bintree.cpp*, *ai.cpp*) s komentármi.

KONTAKT

Hotové zadania a prípadné otázky posielajte na: **frank.schindler@paneurouni.com**