Mesterséges intelligencia 2.

gyakorlat

5. feladat

(alkalmas reprezentáció a konstruktív szemantikai táblához)

Készítette:

Szathmáry László
III. PTM.

5. feladat:

Egy alkalmas reprezentáció kidolgozása a konstruktív szemantikai táblához.

A feladatnak nem volt része az implementálás, de szeretnék megadni egy struktúrát, amelyet implementálás esetén alkalmaznék:

```
struct smuryan {
   char    azon;              //azonosító: 1,2,3, ... ,255
   char    jel;               //t,f
   struct  fa *mutat;
   char    *halott_form;      //amik már nem használhatók
   struct  smuryan *elozo;    //a lev. fában visszamutat
};
```

A levezetési fában ilyen rekordok fognak megjelenni.

Mivel a fa ágazhat, ezért az egyes ágak végét, "alját" is számon kell tartani. Ezt egy láncolt listában tehetnénk meg a legegyszerűbben, amelyben mutatók vannak, méghozzá olyan mutatók, amelyek egy ilyen rekordra mutatnak (az ág utolsó elemére).

```
struct agak {
   struct  smuryan *ag;
   struct  agak *kov;
} *aglista_mutat = NULL;      //inicializálás, ez lesz a fej
```

A levezetési fában az egyes ágakat meg kellene jelölni valamilyen módon. Erre való a **count** változó:

```
unsigned char count = 0x01;
```

Vagyis mindenághoz egy számot fogunk rendelni 1 és 255 között. (Ezt a változót majd mindig növeljük 1-gyel, s így lépkedünk az ASCII táblában sorra. Gondolva egy esetleges C++ implementációra a 0x00 kezdőértéket elhagyjuk, ui. majd ezeket a karaktereket sztringbe is fel kell majd fűzni, s ott ez "sztring-vége" jelként gondot okozna).

Az algoritmus működése szövegesen a következő lenne:

1. lépés: *inicializálás*

a bizonyítandó formulának tárterületet foglalunk, ez lesz majd a levezetési fa "csúcsán".
Azonosító := count; //ennek kezdetben 1 az értéke
Jel := f; //jelölt formulát csinálunk belőle
halott_form := NULL; //ez üres, még nincs "halott" ág
Az ágakból álló láncolt lista kap egy elemet: ennek a rekordnak a címét.

2. lépés:

Van-e olyan ág, amely még nem zárt, ahol folytatni tudjuk a keresést? (Vagyis: az ágakból álló láncolt lista nem üres-e).

- Nincs $\Rightarrow$ minden ág zárt, a kezdeti formula tehát logikai törvény
- Van $\Rightarrow$ itt egy cím van, ami a fa egy elemére mutat. Erre az elemre ráállunk a fában.

3. lépés:

Az előző lépés ráállított egy faelemre, ami egy ág végén van. Ezt megvizsgáljuk, hogy azzal tudunk-e valamit kezdeni, vagyis van-e rá alkalmazható szabály:

- Igen $\Rightarrow$ alkalmazzuk ezt a szabályt. Minden szabály MAGA DEFINIÁLJA, hogy milyen műveletsort kell elvégezni.
  Ennek használatát mutatja majd a konkrét példa a leírás végén.

- Nincs ott az ág végén alkalmazható művelet $\Rightarrow$ ekkor használjuk fel a visszafelé mutatókat arra, hogy lépkedünk visszafelé az ágon, s mindegyik esetben megnézzük, hogy ott van-e valamilyen alkalmazható szabály.
  Az ág minden elemének van egy **`char *halott_form`** mutatója, ami egy "sztringre mutat". Itt azok az azonosítók vannak felsorolva, amely faelemek NEM HASZNÁLHATÓK abban a faelemben. Ennek a módosítása az egyes szabályokra van bízva.
  Pl.: **f¬A** alkalmazása esetén mi történik?

  a **tA** -nak tárterületet foglal, azonosítója := ++count, jel:=t, visszamutatót ráállítja **f¬A** -ra, a "halott" formulák listáját örökli a "szülőtől", **f¬A** -tól, s mivel "vágás" történik (-f), ezért elindul az ágon visszafelé, s minden olyan **f** szimbólumú elem azonosítóját felfűzi magának, amelyek azonosítója még nem szerepelt nála. Így tehát felfűzi **f¬A** azonosítóját is.
  Mindezen műveletsor az **f¬** dolga!!!

  Tehát ott jártunk, hogy az ág végén nem volt alkalmazható művelet, s elindultunk visszafelé olyan elemet keresni, ami még nincs "törölve" (nincs "halott" formulaként nyilvántartva).

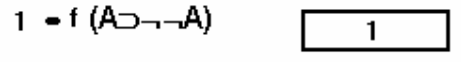
  Ha találunk olyan elemet, ahol van alkalmazható művelet, akkor elvégezzük (ő definiálja, hogy mik ennek a lépései).
  Ha eljutunk a fa tetejéig, s nem találtunk semmi alkalmazható műveletet, akkor nem tudunk továbblépni, NEM log. törv. a kezdeti formula, hibaüzenet, kilépünk.

Ide akkor jutunk már csak el, ha volt alkalmazható művelet. Ha nem volt ágazás: az ágak "aljáról" nyilvántartott információt frissítjük, vagyis a láncolt listánkban azt a címet, ahonnan idejöttünk, felülírjuk. Ha ágazott: az előzőeket elvégzi az egyik ágra, s a másik új ág "aljának" címét IS felfűzi, vagyis ekkor a láncolt lista bővülni fog!!!

A láncolt lista frissítését követi az ellenőrzés, hogy lett-e zárt águnk. A listában lévő mutatók az ágak végére mutatnak, ezek alapján felfelé haladva bejárhatók az egyes ágak, s figyelhető, hogy van-e egyezés. (Persze a "halott" formulákkal lehet egyezés, de az nem számítható be!). Ha valamelyik ág zárt: azt a listaelemet töröljük, vagyis a lista mérete csökkenni fog. Ha minden ág zárt: a lista nulla elemű lesz.
Vissza a 2. pontra.

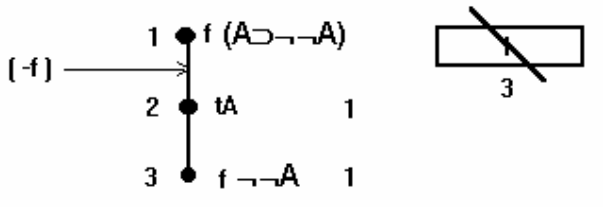
Példa:

Nézzük meg az (**A**⊃¬¬**A**) formula logikai törvény voltát.



Inicializáljuk: **f** -et elérakjuk (jelölt formula lesz), adunk neki egy azonosítót (1), a "halott" formulák listája üres, s ennek a címét berakjuk az ágakat nyilvántartó láncolt listába (ezt a téglalap szemléltetné: egy listaelem, aminek tartalma az 1-es formula címe).

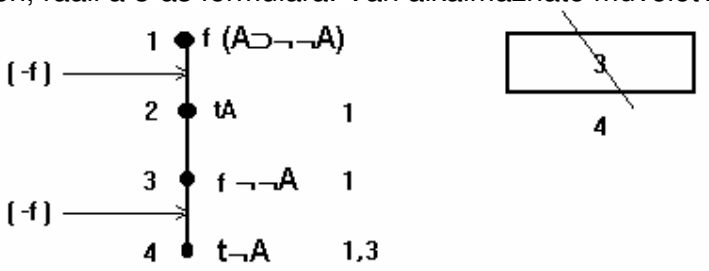
Van-e nyílt ág? Igen, ráállunk a cím segítségével. Van-e alkalmazható szabály? Igen, az **f**¬ . Alkalmazzuk:



Ez a szabály a következőt teszi: létrehozza a **tA** -t (tárterületet foglal neki), azonosítót ad neki (2), megkapja a "halott" formulák listáját (ami üres most), s a vágás miatt ha elindulnánk felfelé, akkor az **f** szimbólumú elemek azonosítóját (most egy ilyen van, az 1-es azonosítójú).

f ¬¬**A** -t hasonlóképp kezeli le. A visszamutatókat szépen beállítja. Majd a láncolt listában elhelyezi az új címet, a 3-as azonosítójú formula címét.

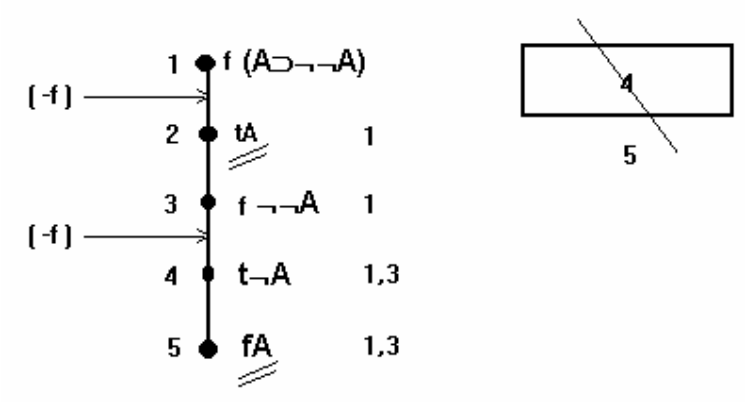
Van nyílt ág? Igen, rááll a 3-as formulára. Van alkalmazható művelet? Igen, **f**¬ :



Új faelem: **t¬A** . Azonosító (4), "halott" formulák listáját MINDIG a szülőtől örökli, majd ha vágás van, akkor megy visszafelé. Így felfűzi a 3-ast, ill. az 1-est már nem, mert az már szerepel.

Láncolt listát frissíti, majd ezután MINDIG ellenőriz. Nincs egyezés, megy tovább.

Rááll a 4-esre, van alkalmazható művelet:



Lesz **fA** , azonosító, "halott" formulák listáját örökli, de mivel **t¬A** marad, ezért az ő azonosítóját nem vesszük fel.
Lista frissítése: 5-össel felülírjuk a 4-es címét, majd ellenőrzés: **fA**, **tA** szerepel, **tA** nem "halott" **fA** számára $\Rightarrow$ logikai törvénnyel van dolgunk.

****

Ami nem triviális: egy ág esetén hogyan választom ki azt az elemet, amelyre alkalmazni kellene a megfelelő szabályt. Erre valamilyen heurisztika kellene.