
Szakértői rendszerek

HOLNAPY DEZSŐ – BIRI SALAH

A számítógépek megjelenése forradalmasította a számítás (kiszámítás) technikáját. Amit addig szabályok felhasználásával gépiesen kellett elvégeznünk, azt gépekre bízhattuk. A kiszámítási szabályok szekvenciáit, amelyek egyértelműen előírták a számítás végrehajtásának menetét, algoritmusoknak neveztük, és ez képezte a programkészítés alapját.

Mi a szakértői rendszer?

Amire algoritmus szerkeszthető, az gépesíthető. Tudásunk, ismereteink hierarchikusak. A tudás felett áll az a tudás (ismeret), amely a tudásra vonatkozik (metatudás), s ezen ismeret-hierarchia kibontása intellektuális igénye a kutatóknak. A számítógépek megjelenése után megindult az algoritmusok szélesebb körének kidolgozása, s e munka rövidesen átlépte a numerikus problémák körét.

Mintegy két és félezer éve, *Arisztotelész* óta tudjuk, hogy a következtetési folyamatok bizonyos részei algoritmikusan és formálisan (szintaxis szerint), vagyis a tartalom (szemantika) ismerete nélkül is végrehajthatók (arisztotelészi szillogizmusok). Mégis csupán most döbbenünk rá, hogy az algoritmusokkal működtethető automatáink, amelyeket eddig számításokra használtunk, következtetések gépesítésére is alkalmasak.

A szakértői rendszerek (egyesek szakértő rendszereknek nevezik) az arisztotelészi formális logika (pontosabban fogalmazva az elsőrendű predikátum kalkulus) felhasználásával következtetések automatizált végrehajtását végzik el egy diszciplína állításaiból kiindulva, a diszciplínához tartozó feltételes állítások (szabályok, implikációk) felhasználásával.

A fejlődés útja

A termelés fejlődése a numerikus kérdések megoldását kívánta, s így a számítás, a valós számtest alkalmazása, a műveletek algoritmizálása vált elsődlegessé, ma pedig az ismeretek dzsungelje kényszerít bennünket arra, hogy újra felfedezzük az Arisztotelész idejében már algoritmizált következtetési mechanizmusokat.

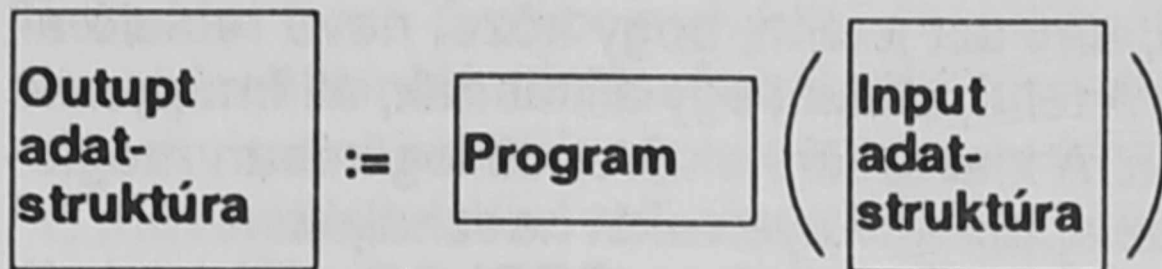
Gépek, algoritmusok, nyelvek és imperatív nyelvek

A Neumann-féle programvezérelt digitálautomaták, a számítógépek úgy működnek, mint egy egyszerű összeadógép „általánosításai”. Megmondjuk mit kell összeadni (operandusok), és leírjuk, hogy hogyan kell azt végrehajtani (operátor). Gondoljunk $5 + 6$ -ra, ami egyenlő 11 -gyel.

Bonyolultabb esetben, amit nyilván a számítógépekre bízunk, van egy input-adatsorozat, ez tartalmazza az operandusokat, van egy program, ami az operátor funkcióját tölti be, s az eredmény egy output-adatsorozat lesz. (1. ábra)

Lényegében tehát semmi sem változik az összeadáshoz képest, csak több a szám, és bonyolultabb a művelet algoritmus, de ez mit sem számít, mert azt úgymint gépek végzik.

A programot, az algoritmusnak egy egyezményes jelekkel kódolt formáját azonban valakinek el kell készítenie, le kell írnia. Ehhez nyelv kell, pontos nyelvtani szabályokkal (szintaxis). A Chomsky-féle nyelvtani leírás (1) alkalmas volt a mesterséges nyelvek, a



1. ábra
A program fogalma

a a jobb oldalon szereplő kifejezés értékével. Az algoritmust leggyakrabban ilyen imperatív nyelven írják le.

Numerikus problémamegoldó algoritmusainkat ilyen típusú nyelveken (FORTRAN, Algol, Pascal, C, stb.) szokták rögzíteni, és közölni a számítógépekkel.

Intelligencia, mesterséges intelligencia-nyelvek és applikatív nyelvek

Mint már említettük, a numerikus algoritmus fölött lévő nemnumerikus tudásunkat, ismeretünket is szeretnénk gépesíteni, ha lehet. Először is soroljuk hierarchiába a problémamegoldás szintjeit (2), és nevezzük el azokat *Morris* szemiotikai rendszerében használatos nevekkel (3).

A legalacsonyabb szintű problémakezelés a *szintaktikai* (a formai, a „helyesírási” szabályok mértékéig rögzített) problémakezelés. Ekkor tudnunk kell, hogy MIT akarunk megoldani és HOGYAN. Ez algoritmikus szintű problémamegoldást jelent. Minthogy ez esetben az utolsó vesszőig mindent a feladatmegoldónak kell elvégeznie, ezért szintaktikai szintű problémamegoldásnak is szokás nevezni az ilyen megoldásokat.

A következő a *szemantikai* (a tartalom, az értelem, a jelentés mértékéig rögzített) szint, amely a mesterséges intelligencia szintű problémakezeléseknél fordul elő. Ekkor tudnunk kell azt, hogy MIT akarunk, azonban nem kell algoritmust adni arra, hogy az eredményt HOGYAN kell elérni. Ez a nemalgoritmikus problémamegoldás, vagy más néven a logikai szintű problémamegoldás szintje. Minthogy ez esetben csak az elérendő célt kell megadni ahhoz, hogy a feladat – egy intelligens rendszer segítségével – megoldódjék, szemantikai szintű megoldásnak is szokták nevezni az ilyen problémamegoldásokat.

A legfelső szint a *pragmatikai* (a jelek befogadjának, értelmezőjének képességeit is figyelembe vevő) szint. Ebben az esetben már azt sem kell pontosan tudnunk, hogy MIT akarunk. Ott már természetesen azt sem kell meghatározni, HOGYAN érhetjük el a megoldást. Ez az ún. rosszul definiált problémamegoldás szintje. Ez esetben a cél sem ismert, az is egy intelligens rendszerrel történő párbeszéd során alakul ki, s ezért szokták pragmatikai szintű problémamegoldásnak is nevezni az ilyen megoldásokat.

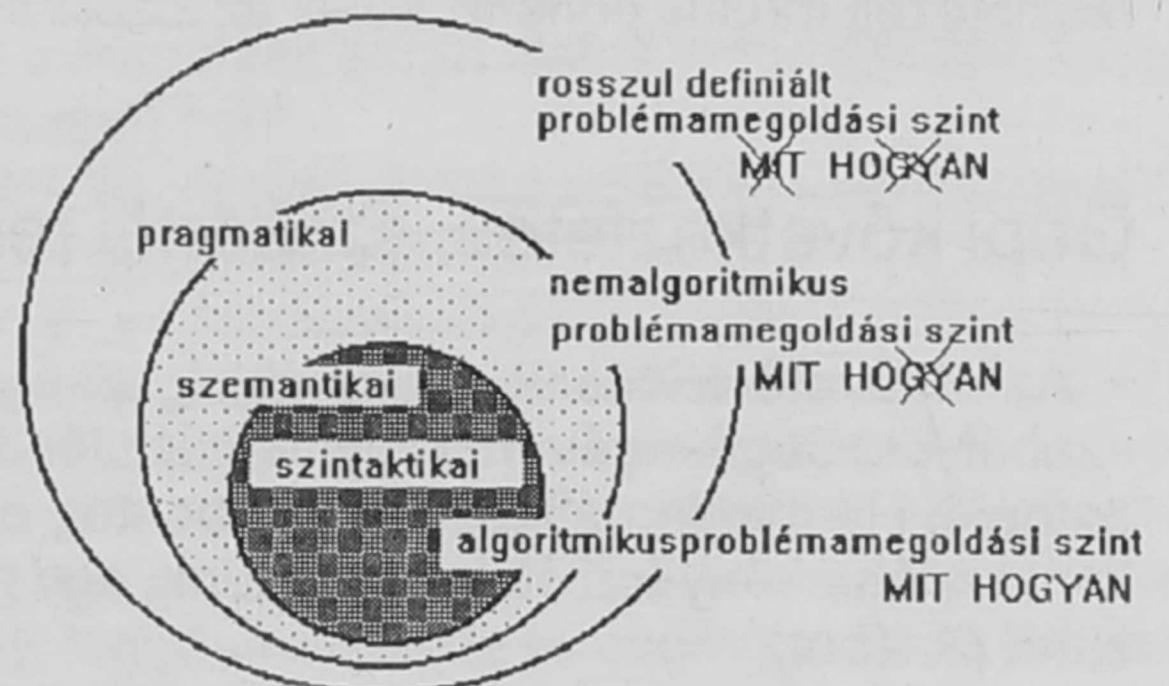
Amíg algoritmikus problémákat kezeltünk, a 2. ábra legbelsőbb körében jártunk. Az ún. mesterséges intelligencia programok a második szintbe, a szemantikai szintbe tartoznak.

Az intelligencia fogalmának használatával azonban vigyáznunk kell (4). Többnyire ui. másra gondolunk, amikor humán intelligenciáról beszélünk, és másra amikor gépi intelligenciáról van szó. Az utóbbiba az intuíciót is igénylő következtetési láncok természetesen nem tartoznak bele.

programozási nyelvek definiálására, s ez a matematikai nyelvészet fejlődését is elősegítette visszahatásként.

Az ún. *imperatív programozási nyelvek* a programozás, az operációt végrehajtó algoritmusok könnyű leírását szolgálták. Az imperatív szó arra utal, hogy e nyelvek egy mondata pl. a következő:

$a := 3 \cdot b + 1$,
ami azt jelenti, hogy *legyen egyenlő*



2. ábra
A problémamegoldás szintjei

Az ún. *applikatív programozási nyelvek* állapotok leírására szolgálnak. Az applikatív nyelvek jellegzetes utasítása reláció formájú. Pl.:

kőzet*gránit, mélységi, kvarc, káliföldpát), ami azt jelenti, hogy kőzet nevű reláció áll fenn a gránit, mélységi stb. fogalmak között. A fenti relációt egy állításnak, állítmánynak, idegen szóval predikátumnak is nevezhetjük. A klasszikus elsőrendű logikában megfogalmazható igazságok leírására többnyire az applikatív nyelveket használják.

A nemnumerikus problémák leírását a fenti típusú nyelveken (PROLOG stb.) szokták rögzíteni, és közölni a számítógéppel.

A formális logika

A formális logikában igazságértékkel rendelkező premisszákból (állításokból) úgy jutunk egy konklúzióra, hogy közben az állítások tartalmát nem vizsgáljuk. A leggyakrabban alkalmazott következtetési szabály az ún. *modus ponens* (tételező mód, leválasztási szabály), amely a következőket mondja ki:

$$\{ p, q, p \} \rightarrow q,$$

ahol p és q állítások, a $\{ \}$ pedig a premisszák halmazának a jelölésére szolgál. Jelentése pedig a következő: ha p akkor q feltételes állítás (implikáció) fennállásából és p , ún. előtag igaz voltából következik, hogy az implikáció utótagja, a q állítás is igaz. (5)

Közérthető példával:

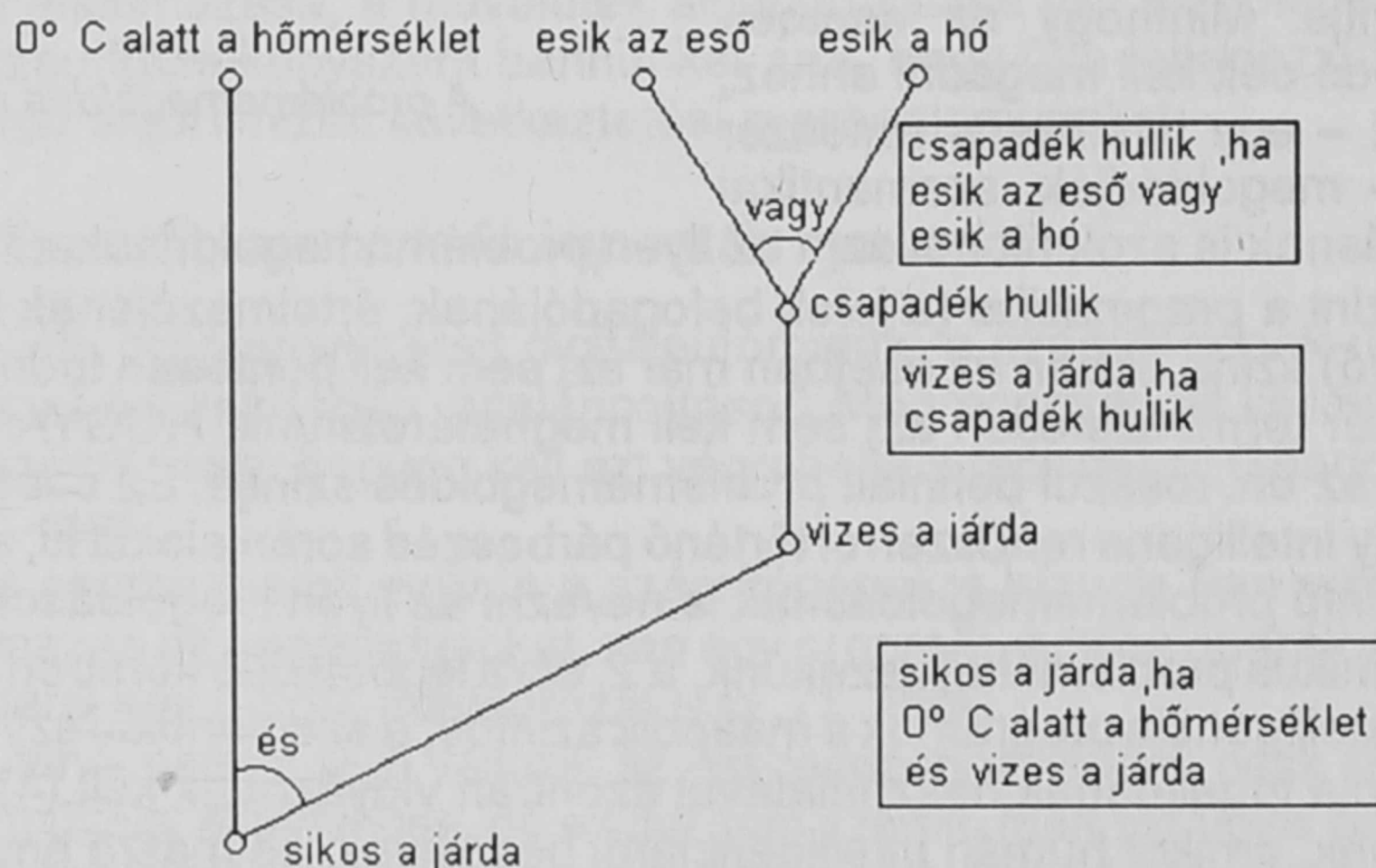
{ha „esik az eső” akkor „vizes a járda”, „esik az eső”}

„vizes a járda”.

A formális következtetések leggyakrabban alkalmazott következtetési szabálya a fentebb részletezett *modus ponens*, és ez az alapja a szakértői rendszerek működésének is.

Gépi következtetés, szakértői rendszer

Az alapvető állítások (axiómák) igazságértékeiből, a feltételes állítások (levezetési szabályok) segítségével az így definiált tér összes állítása, ill. annak igazságértéke levezethető. Ha minden állításnak egy pontot, a szabályok alapján pedig a premisszákból a konklúzióba irányított éleket fektetünk, egy gráfot kapunk, amelyet keresési térnek nevezünk (3. ábra).



3. ábra
Keresési tér

A keresési tér az előbbieken részletezett módon előre menő következtetéssel felépíthető, teljes egészében explicitté tehető. Legtöbbször azonban nem dolgozzuk ki a teljes keresési teret. Amikor egy állítás igazságát bizonyítani akarjuk, keresünk egy szabályt, amelynek konklúzió részében a bizonyítandó állítás van (lehet több ilyen szabály is), és ezzel visszavezettük a problémát a premisszák igazságértékének a meghatározására. E művelet ismétlésével, explicitté téve a keresési tér egy részét, eljutunk az axiómáig (szakértői rendszerek esetében ezek igazságértékére a rendszer rákérdez), és azok igazságértékeinek ismeretében előre haladó következtetéssel már minden közbenső állításhoz igazságértéket rendelhetünk. Ily módon kiinduló állításunk is igazolódik, vagy megállapítható róla, hogy a keresési tér alapján az állítás igazsága nem bizonyítható.

A fent említett eljárás az automatikus tételbizonyítás végrehajtásmódja. A szakértői rendszerek következtető gépe (következtető mechanizmusa) is az automatikus tételbizonyítás elvén működik.

Keretrendszerek és példák

A softverfejlesztők a szakértői rendszer kifejlesztéséhez is segítséget akartak nyújtani. A szakértői rendszerek különböző típusaihoz olyan felhasználóbarát, ún. keretrendszereket készítettek, amelyek sem a matematikai logika mély ismeretét, sem az applikatív programozási nyelvek ismeretét nem igénylik, mégis lehetővé teszik szakértői rendszerek egyszerű eszközökkel történő létrehozását.

A szakértői rendszerek leggyakrabban alkalmazott típusára, a „szabályalapú szakértői keretrendszerek” alkalmazására, a következőkben mutatunk be néhány példát.

Esik az eső

Az algoritmus szintű problémamegoldásoknál a gép a feladat megoldására vonatkozó tervet (algoritmust, programot) készen kapja. A nemalgoritmikus szintű problémamegoldás esetén a megoldás terve, algoritmus a megoldás során alakul ki (6). Kérdés, hogy mi módon gépesíthetjük az utóbbi feladatmegoldást?

Vegyük szemügyre a 3. ábra szerinti példát, és vizsgáljuk meg az ábrán vázolt keresési tér segítségével, hogy igaz-e az az állítás, hogy „síkos a járda”.

A keresési teret az axiómák, az állítások és a szabályok alkotják. Az axiómák, a keresési gráf azon alaptudást reprezentáló csomópontjai, amelyekbe nincs befutó él. Igazságértékei külső, a matematikai apparátustól független döntések eredményei. A keresési tér többi csomópontjának igazságértéke a fentiekből levezethető. Az említett csomópontok állításokat képviselnek. A szabályok, feltételes utasítások, ill. implikációk a befutó élek által képviselt premisszák igazságértékeitől függően a kérdéses csomópontához tartozó állítás igazságértékét adják meg.

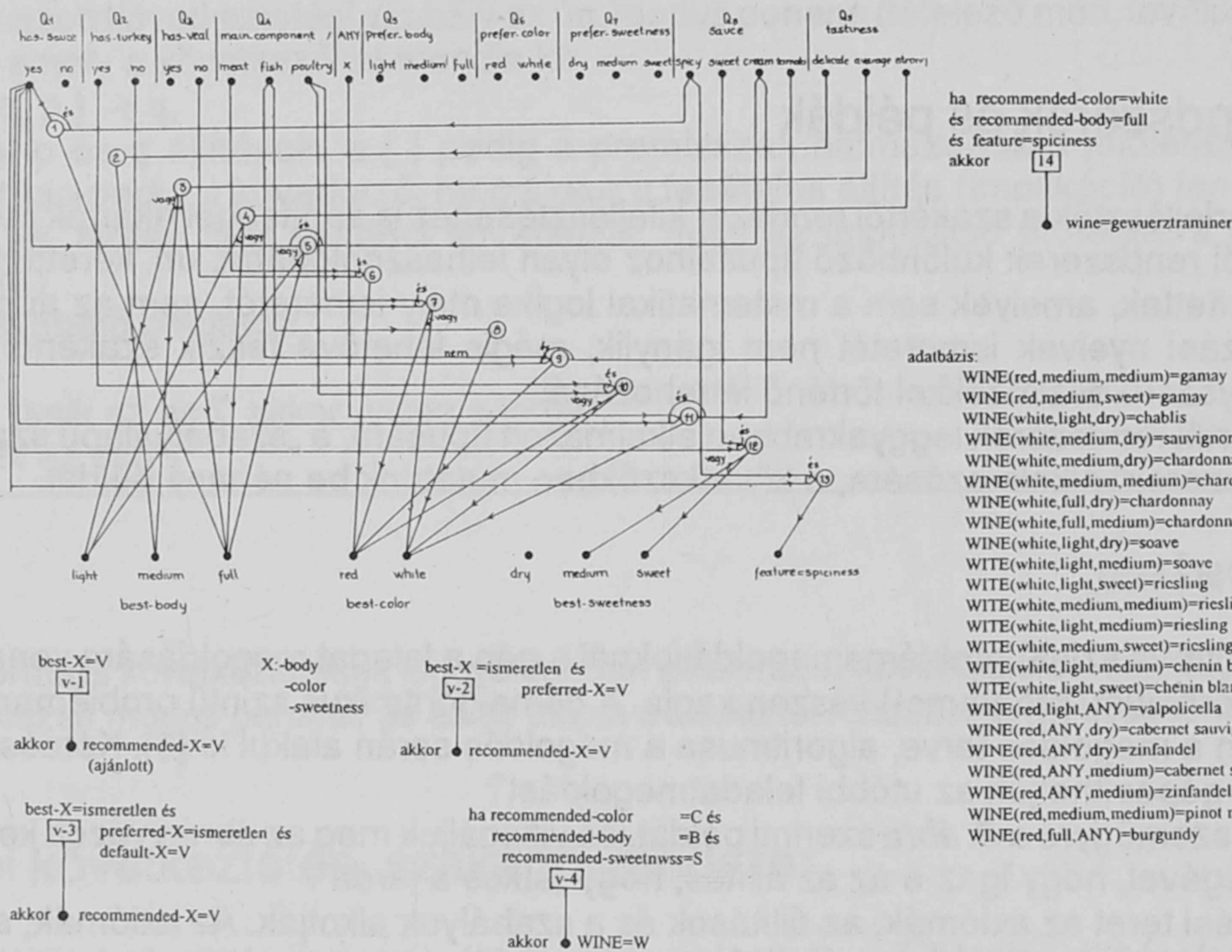
A „síkos a járda” kijelentés igazságértékének eldöntéséhez meg kell vizsgálnunk, hogy a „0°C alatt a hőmérséklet” igaz-e? Ez utóbbi, tény kérdés, ki kell nézni az ablakon és megadni az állítás igazságértékét. A döntéshez szükséges még a „vizes a járda” állítás igazságértéke is. A döntés azonban a „csapadék hull” igazságértéke, az pedig az „esik az eső”, ill. az „esik a hó” igazságértéke alapján, szabályok segítségével határozható meg. Az utóbbiak igazságértékét a tények határozzák meg. A döntéssorozat végrehajtása ezután meghozza a megoldást.

Hús van ebédre, mit igyunk?

A címbeli feladat esetében egy nemnumerikus problémamegoldással állunk szemben. Tudjuk, hogy hús van ebédre, de nem tudjuk, milyen italt illik hozzá felszolgálni. Amennyiben a megoldást egy intelligens rendszertől várjuk, az étel-ital egymáshoz rendelésre vonatkozó tudásunkat a 3. ábrához hasonlóan egy keresési gráfba kell rendeznünk. E művelet az előbbi példánál bonyolultabb, ugyanis általános érvényű reduktorok is szerepelnek a térben, amit a szokásos gráffal bemutatni nem lehet.

A 4. ábra szerinti keresési térben a döntés a tényállítások bekérése után például a következő módon jöhet létre:

- mi a főétel?
- hús?
- hal?
- szárnyas? válasz: hús.
- a hús borjúhús? válasz: igen.
- az ételhez kell szósz? válasz: igen.
- milyen szósz akar hozzá?
- fűszereset?
- krémeset?
- édeset?
- paradicsomosat? válasz:...



4. ábra
Az italrendelés keresési tere

A válaszok alapján az intelligens rendszer eldönti, hogy milyen legyen a bor színe, testessége, édessége stb. A döntést, mint az előző példából is lehetett látni, szabályok alapján hozza meg a rendszer. A döntéshez felhasznált szabályok például a következők lehetnek:

- HA az ételhez kell szósz = igen,
- ÉS milyen szószt akar hozzá = fűszereset,
- AKKOR a testesség = telt.
- HA mi a főétel = hús,
- ÉS a hús borjúhús = igen, AKKOR a szín = vörös.
- HA az ételhez kell szósz = igen,
- ÉS milyen szószt akar hozzá = édeset, AKKOR az édesség = édes,
- VAGY az édesség = közepes.

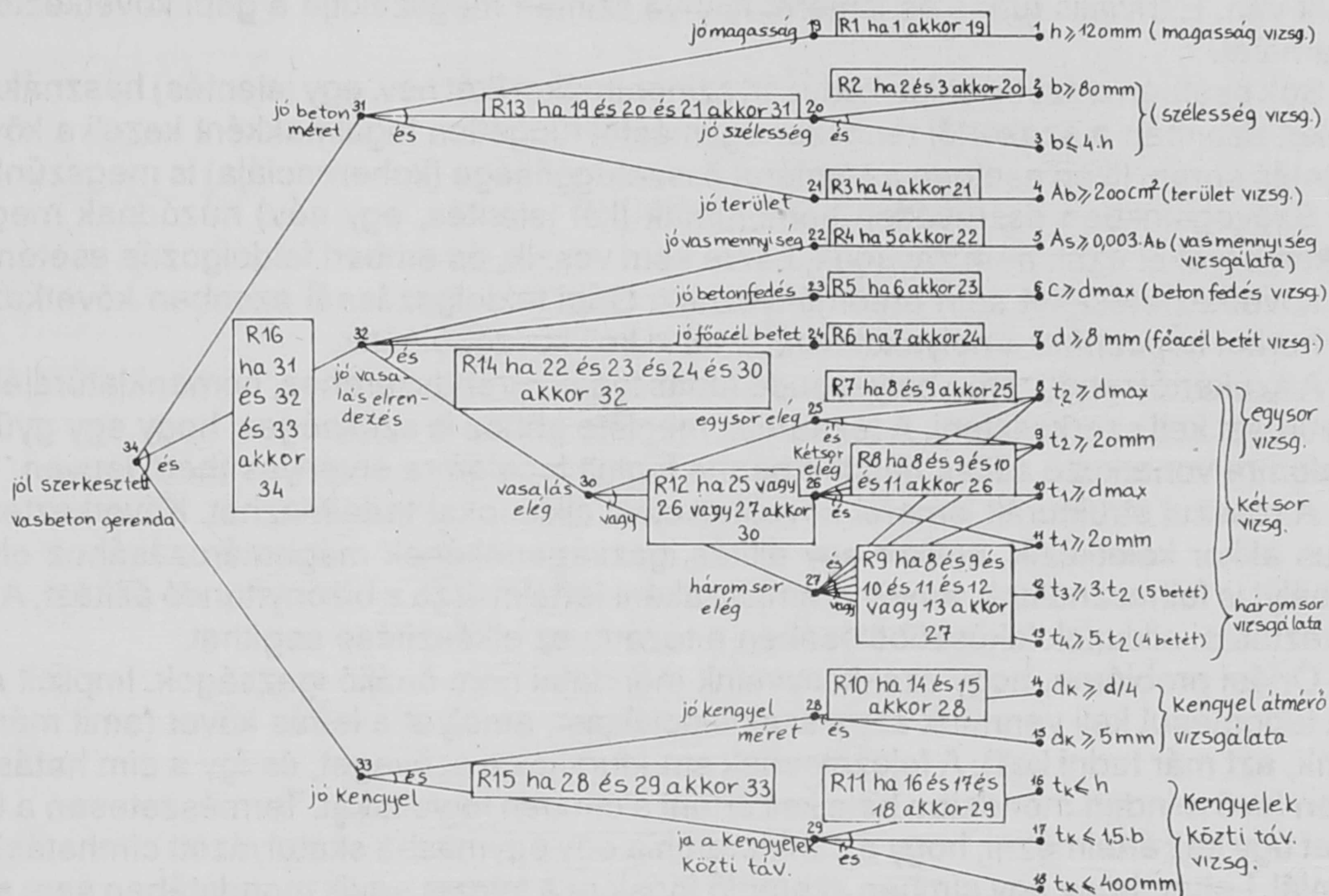
Nyilvánvalóan alkothatók olyan szabályok, hogy ha valaki a vöröset szereti, akkor mindenképpen vöröset rendel az ételekhez, vagy elsősorban ezt rendeli hozzá. Az M1 jelű szakértői keretrendszerben megírt 4. ábra szerinti szakértői rendszer valóban a fenti prioritásokat is figyelembe veszi a szabálybázisában. A hozzárendeléseket minden lehetséges módon kiolvashatjuk a 4. ábra keresési terének gráfjaiból.

A fent bemutatott példából azt is láthatjuk, hogy egy igen egyszerű probléma esetén, egy igen világos koncepció szerint kialakított szakértői rendszer mennyire áttekinthet-

lenné, és gép nélkül mennyire kezelhetetlenné tud válni. E rendszer tudásbázisában pedig csak mintegy 40 állítás és kevesebb mint 20 szabály van feldolgozva, mégis alig követhető!

Nemnumerikus problémák megoldása a műszaki életben

Szakértői rendszereket a műszaki életben is használunk (vagy ha szerényebbek vagyunk, azt kell mondanunk, hogy szeretnénk használni). Kísérleteinkből egyet példaként itt is bemutatunk. E feladatban a bizonyítandó, hogy a „jól szerkesztett vasbeton gerenda” állítás igaz. Vegyük szemügyre az 5. ábrát, amelynek keresési teret jelentő grájából a következőket olvashatjuk ki (az ábrán az állítások sorszámozva vannak, a szabályok sorszáma pedig egy R betűhöz kapcsolódik):



5. ábra
Gerendavasalás szakértői rendszere

- HA jó betonméret = igaz,
- ÉS jó vaselrendezés = igaz,
- ÉS jó kengyel = igaz,
- AKKOR jól szerkesztett vasbeton gerenda = igaz.
- HA jó magasság = igaz,
- ÉS jó szélesség = igaz,
- ÉS jó terület = igaz,
- AKKOR jó betonméret = igaz.
- HA $h \geq 120$ mm,
- AKKOR jó magasság = igaz.
- HA $b \geq 80$ mm,
- ÉS $b \leq 4 \cdot h$,
- AKKOR jó szélesség = igaz.
- HA $A_b \geq 200$ cm²,
- AKKOR jó terület = igaz.

Ugyanígy járunk el az első szabály többi premisszájának az igazolásával is (5. ábra). Ha mindhárom premisszaként felsorolt állítás igaznak bizonyul, akkor a „jól szerkesztett vasbeton gerenda” állítás igazságértéke igaz lesz.

Szakértői rendszerekkel kapcsolatos problémák

Mai ismereteink, amelyeket szavakkal írunk le, előkészítetlenek arra, hogy azokat automatizált problémamegoldás céljára felhasználjuk. A következőkben néhány nehézséget mutatunk be a figyelem felkeltésének szándékával (tudásunk, ismeretünk csak az alantiek megszívlelése esetén lehet alkalmas egy szakértői tudásbázis kialakítására):

– Szakkönyveink általában hézagos tudást tartalmaznak, aminek „megértéséhez” speciális szaktudás szükséges. Ez természetesnek látszik, azonban a szakértői rendszerek számára ezt az ún. szaktudást explicit módon szükséges megfogalmazni. E nélkül ui. megszakad a következtetési lánc, nem teljes az ismeret, tehát nem automatizálható a következtetés.

– Vannak triviális ismereteink a világról, amelyeket emberi döntéseink során nem fogalmazunk meg explicit állítások formájában, döntési következtetéseinkben mégis szerepük van. E triviális tudás és ismeret hiánya szintén megszakítja a gépi következtetés folyamatát.

– Sok esetben a szebb stílus kedvéért szinonímákat (két név, egy jelentés) használunk. Ezeket azonban a szakértői rendszer egymástól független fogalmakként kezeli a következtetés során. Ilyen esetben az ismeret összefüggősége (koherenciája) is megszűnhet.

– Szövegeinkben észrevétlen homonímák (két jelentés, egy név) húzódnak meg. A szakembereket ezek nem zavarják, észre sem veszik, és emberi feldolgozás esetén téves következtetéseket sem eredményeznek. Gépi feldolgozásnál azonban következtetési hurkot képeznek, amelyeket feltétlenül ki kell küszöbölnünk.

– A szakértői rendszerbe helyezendő tudás fogalomrendszeréhez, nómenklatúrájához tezauruszt kell szerkeszteni. A tezaurusz megléte ahhoz is szükséges, hogy egy gyűjtőfogalomra vonatkozó állítás minden benne foglalt fogalomra érvényesíthető legyen.

– A rosszul strukturált ismeret következtetési ciklusokat tartalmazhat. Következtetési ciklus akkor keletkezik, amikor egy állítás igazságértékének meghatározásához olyan szabályt is felhasználunk, amely premisszaként tartalmazza a bizonyítandó állítást. A következtetési ciklusok kiküszöbölésében a tezaurusz elkészítése segíthet.

– Óriási probléma, hogy szakkönyveink mondatai nem önálló igazságok. Implicit módon tudomásul kell vennünk azt a szekvencialitást, amelyet a leírás követ (amit már leírtunk, azt már tudni kell). A fejezetcímek strukturálják a szöveget, és így a cím hatáskörében lévő minden mondatba bele kell érteni a címben foglaltakat. Természetesen a fentieket úgy kell értelmezni, hogy a címhierarchia egy egymásba skatulyázott címhatáskört definiál. Lehet, hogy egy címben szereplő fogalom a fejezet egyik mondatában sem szerepel, mégis minden mondatához hozzá kell érteni.

– Zavar származhat abból, is hogy a fogalmak jelentése függ a környezettől. Emberi következtetések esetén ilyenkor sincs probléma, mert hallgatólagos igazságok iktatódnak be a premisszák mellé, és a következtetés helyes lesz.

Perspektívák: az 5. számítógép-generáció

Túlságosan hozzászoktunk ahhoz, hogy a számítógépeket számolások elvégzésére használjuk. A programvezérelt digitálautomaták, amelyekett természetesen számításra is lehet használni, nemnumerikus információk kezelésére is képesek.

A nemnumerikus információkezelés keretein belül – mint dolgozatunkból is kitűnhetett – számítógépeinkkel a klasszikus elsőrendű (arisztotelészi) logika formális alkalmazásával automatikus tételbizonyítást is végeztethetünk. Az automatikus tételbizonyítás pedig a szakértői rendszerek alkalmazásának az alapja.

Japánban napjainkban készül az 5. generációs projekt, amely olyan automaták előállítását is tervezi, amelyeknek a vezérlési mechanizmusa rekurzív. Ezek az új automaták (számítógépek) óriási kapacitással rendelkeznek, gyors működésűek, és egyik „legegyszerűbb nyelvük” egy logikai alapú, állapotleíró, ún. applikatív nyelv (valószínűleg a PROLOG) lesz. Addigra, amikorra ezek a berendezések megjelennek a piacon, formalizálható döntéseinket meg kell egzaktul fogalmaznunk, mert a technikai lehetőségeket csak így módon tudjuk majd kiaknázni.

IRODALOM

- (1) Chomsky, N.: Generatív grammatika. Európa Könyvkiadó, Budapest, 1985.
- (2) Holnapy D.: Problémamegoldás és intelligens rendszerek a műszaki tervezésben. Informatika 2. kötet. Budapesti Műszaki Egyetem Építőmérnöki Kar Informatikai Laboratórium kiadványa, Budapest, 1991.
- (3) Józsa P.: Lévi-Strauss, strukturalizmus, szemiotika. Akadémiai Kiadó, Budapest, 1980.
- (4) Ladrirre, J.: A mesterséges intelligencia. In: Mérleg 28 (1992), 272-283. pp.
- (5) Pólos L. – Ruzsa I.: A logika elemei. Tankönyvkiadó, Budapest, 1987.
- (6) Szóts M.: Feladatmegoldás és számítógép. A műszaki tervezés mint a feladatmegoldás speciális esete. In: Építési kutatás, fejlesztés 13 (1980), 71-74. pp.