



**QUILLIAN 1969: THE TEACHABLE
LANGUAGE COMPREHENDER: A
SIMULATION PROGRAM AND
THEORY OF LANGUAGE**

Bevezetés

- egy számítógépes programot ír le, ami képes írott szövegek megértésére
- BBN-LISP-ben írták
- a memória struktúrájának bemutatásán keresztül a szerző leírja a szövegértéssel kapcsolatos elméletét
- azt gondolja, hogy az agyban is kb. az játszódik le, mint ami a program működése közben

Működési elve

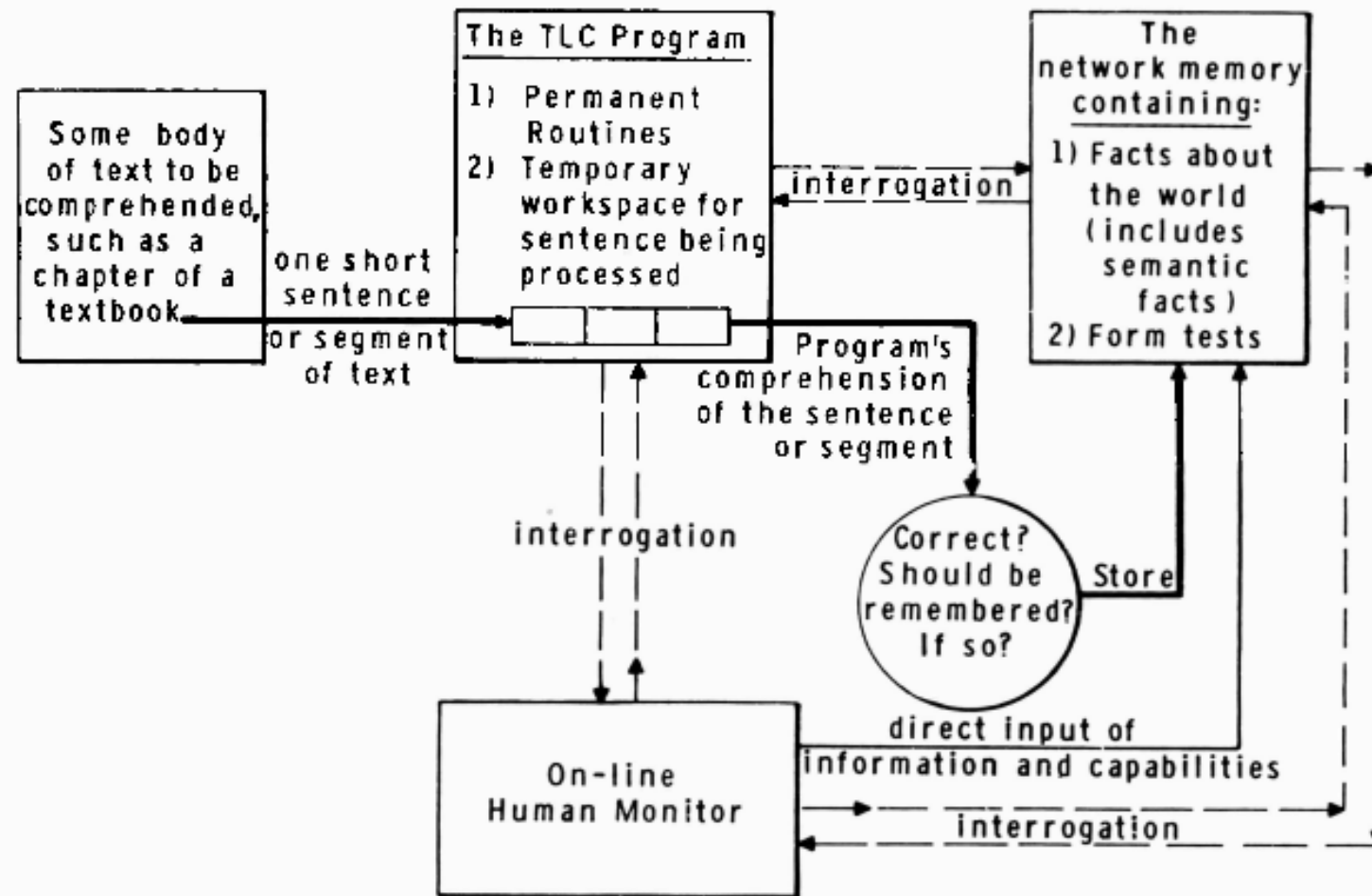


FIG.2 Principle Components and Interactions of the TLC System

A memória

- minden információt egységekben (unit) és tulajdonságokban (property) tárol
- egység: bármi, ami leírható egy szóval, mondattal, NP-vel (az ábrákon szögletes zárójellel)
- tulajdonság: VP, relative clause, jelzők... (az ábrákon kerek zárójellel)



Fig. 3 A Piece of Information In Memory

- az adott dolog általánosabb formája (a példán: ügyfél → ember) a superszet
- ha a lehető legáltalánosabb a szó, akkor a NIL egység a supersetje
- a szavak a memórián kívül vannak egy "szótárban", minden szóhoz hozzá van rendelve egy vagy több pointer a memóriában található olyan egységekre, amelyek a szó jelentését írják le
- az egység első eleme kötelezően egy pointer a supersetre, továbbá tartalmaz pointereket az egységet leíró tulajdonságokra

- ennek a struktúrának a segítségével tetszőleges új egységek hozhatók létre egy meglévő egység mint superset és az új koncepciót leíró tulajdonságok összerendelése által
- egy példa: adott Joe Smith mint egység, ha Joe Smith-ről akarunk beszélni 3 éves korában, létrehozunk egy új egységet, melynek a supersetje Joe Smith, a tulajdonsága pedig az, hogy a kora 3 év, stb.
- ez a program működésének alapja
- fontos még, hogy a tulajdonságok lényegében attribútum-érték párok kibővítve úgy, hogy leírhatók legyenek a prepozíciók, ill. az igék és tárgyuk

Az információ rendszerezése

- mivel a struktúra más egységekre és tulajdonságokra mutató pointeremből áll, a memória lényegében egy hálózatot/gráfot ír le
- minden tárolt koncepciót legfeljebb egy egység írhat le; ha két egység ugyanazt a koncepciót írja le más kontextusban, akkor az eredeti egység lesz a superszetjük (ld. Joe Smith), ez csökkenti a redundanciát

- ez a memóriaszerkezet lehetővé teszi a közös építőelemek (intersections) gyors megtalálását egyfajta párhuzamos keresés által
- a keresés algoritmus a breadth-first search, az egyes iterációkban megjelöli a megtalált egységeket a keresés kiindulópontjával szolgáló koncepcióval (activation tagging)
- először a legközelebbi közös építőelemet találja meg, majd az egyre távolabbiakat
- a program különböző megkötésekkel (iterációk száma, megtalálendő közös építőelemek száma) és részletesebb tagginggel (fel tud ismerni olyat, ami csak tulajdonságokon keresztül kapcsolódik a kiindulási koncepcióhoz) használja ezt az algoritmust

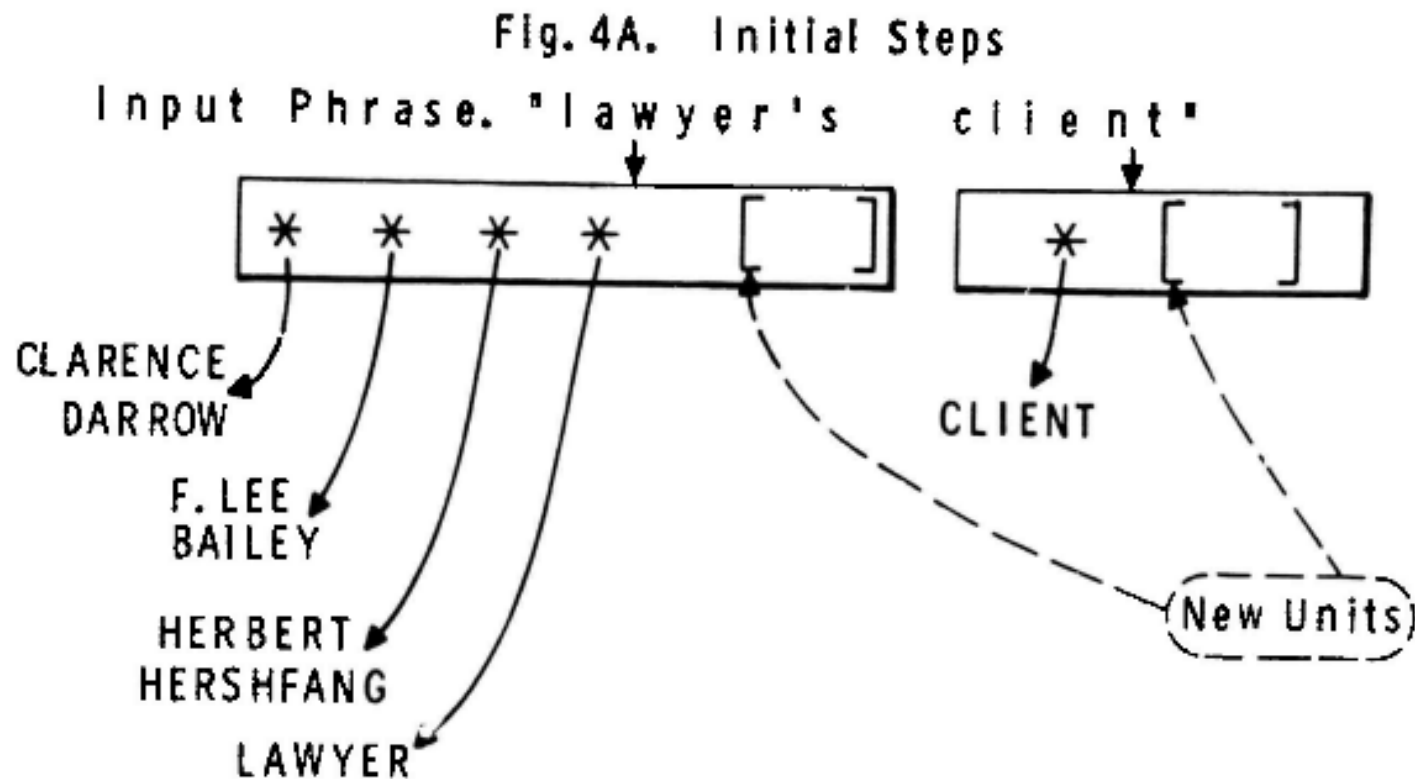
A program működése

- a szerző szerint úgy értjük meg az írott szöveget, hogy az olvasott szavak, kifejezések korábbi ismereteket idéznek fel, ezekhez új kapcsolatokat és tulajdonságokat rendelünk → új ismereteink a régieken alapulnak
- a program megértési modellje: amikor új szóval találkozunk,
 1. létrehoz számára egy új, üres egységet
 2. előkeresi a szó lehetséges értelmezéseit (candidates; ezek egyrészt a szótár elemei, másrészt az anafora referensei)

- 3. megtalálja az új információt reprezentáló tulajdonságokat
- ez a megoldás felvet három problémát: egy szónak több jelentése is lehet; hogyan kell összeállítani az új ismeretet leíró tulajdonságokat; a visszautalásokat is meg kell érteni
- a program ezeket egy folyamat segítségével kívánja megoldani, ez adja a tulajdonképpeni, írott szövegek megértésével kapcsolatos elméletet
- minden értelmezendő szóhoz létrehoz egy új, üres egységet és egy pointerlistát, ezek a lehetséges értelmezések memóriabeli reprezentációira mutatnak

- a létrejött gráfban breadth-first keresést indít a lehetséges értelmezések egységeinek tulajdonságaiból kiindulva, megvizsgálja, hogy az adott tulajdonság vonatkozhat-e az aktuális szóra
- ezt a korábban említett activation tagginggel állapítja meg
- ha talál ilyen tulajdonságot, akkor alkalmazza az új környezetre, ill. az adott lehetséges értelmezés (candidate) lehetséges supersetté válik
- az activation tagging mellett formai teszteket is használ, ezek a szórendre, egyéb szintaktikai információkra vonatkoznak

- a szerző szerint ez megoldja mindhárom problémát



A többértelműség kezelése

- háromféle superset intersection jöhet létre:
- 1. két tulajdonság superset-chainjének a metszete
- 2. két lehetséges értelmezés superset-chainjének a metszete
- 3. egy tulajdonság és egy lehetséges értelmezés superset-chainjének a metszete; a megértés szempontjából ez ad releváns információt a szöveg jelentésére vonatkozóan

- egy példa: John shoots the teacher
- 1. tudunk valamit Johnról és tudunk valami mást a tanárról, amik összefüggenek egymással, de ez indirekt információ, ezért nem releváns **két tulajdonság superszet-chainjének a metszete**
- 2. John is ember, a tanár is ember – itt pedig nem tudtunk meg semmi újat **két lehetséges értelmezés superszet-chainjének a metszete**
- 3. itt kétféle lehetőség van: a tanár gyerekeket tanít, a gyerek ember – John is ember; olyan form testet kell kitalálni, ami ezt elutasítja, azt viszont elfogadja, hogy a fényképésznek a tárgya az ember, és a tanár is ember (ha korábbról tudjuk, hogy John fényképész; ha azt tudnánk, hogy John bűnöző, akkor arra a következtetésre jutna a rendszer, hogy John lelőtte a tanárt **egy tulajdonság és egy lehetséges értelmezés superszet-chainjének a metszete**

Adatok és formai tesztek általánosítása a memóriában

- a memóriában lévő gráf alapján tud következtetni: a lawyer's client alapján megérti azt is, hogy woman's client
- lawyer → professional → person; woman → person
- a formai tesztek is ugyanígy ki tudja következtetni

Komplex mondatok értelmezése

- hosszabb kifejezések elemzéséhez eddig nem tárgyalt technikát alkalmaz a program:
- lawyer's young client: arra, hogy lawyer's young, nem talál értelmes formai tesztet, ezért a hozzá tartozó tulajdonságokat (pl. EMPLOY félreteszi, a következő körben megtalálja, hogy a young az AGE tulajdonságnak egy lehetséges értéke, és a client (PERSON) rendelkezik AGE tulajdonsággal, így összekapcsolódhat; végül formai tesztek alapján ki tudja választani, hogy melyik szó a head

Megoldatlan problémák

- a program többek között nem tud asszociálni konkrétan nem említett koncepciókra (pl. a hímnemű gyermekekről nem tudja, hogy fiú, az öreg férfiről pedig azt, hogy aggastyán)
- nem tud a térbeli elhelyezkedés alapján következtetni
- nem tudja beépíteni a memóriájába a megértett szöveg jelentését (pl. 'Battleships in World War I had 16 inch guns' - nem tudja, hogy mihez kapcsolva tárolja el: világháborúhoz, fegyverekhez, hajózási történelemhez)