

Lenat – Guha – Pittman – Pratt – Shepherd
1990

CYC: Toward programs with common sense

“Cyc is a bold attempt to assemble a massive knowledge base spanning human consensus knowledge.”

Kalivoda Ágnes

2017. 10. 12.

Lexikális szemantika

Általában a Cyc projektről

- A projekt **1984-ben** kezdődött, és még most is tart (Douglas Lenat, Cycorp Company, USA)
- **Kiindulópont:** egy AI létrehozása 3 lépcsőből áll:
 - reprezentációs nyelv kialakítása
 - következtető eljárások, amikkel a tudás használható
 - tudásbázis építése ← *ez lenne mindennek az alapja*
- **Probléma:** programok merevsége ↔ emberi kreativitás
- **Célkitűzés:** olyan tudásbázis, amely...
 - ismer tényeket, heurisztikákat
 - analóg probléma megoldásra képes
 - kontextusfüggő mikro-elméleteket ismer

A Cyc rendszer felépítése

- 3 fő komponens:
 - **reprezentációs nyelv** (*representation language*): CycL (Lisp alapú), a tudásbázis bővítésével párhuzamosan fejlesztették
 - **következtető rendszer** (*inference engine*): tkp. egy nagy általános megoldás helyett sok kicsi, modus tollens, modus ponens stb. - és ami nagyon gyakorinak bizonyult, arra shortcut-ok
 - **ontológia** (*ontology*): témákat (pl. vásárlás) formalizálnak, az elméletet axiomatikus leírásként rögzítik

A Cyc rendszer felépítése

- 2 szint:
 - **Episztemológiai** (*epistemological*):
 - elsőrendű predikátumlogikai nyelvet használ
 - törekvés az egyszerű szemantikára
 - tkp. ez tartalmazza az összes tudást
 - **Heurisztikus** (*heuristic*):
 - az episztemológiai szint optimalizálása
 - a minél gyorsabb következtetéseket végzi
 - nem célja, hogy egyszerű és átlátható legyen

Cyc ontológia: Alapok

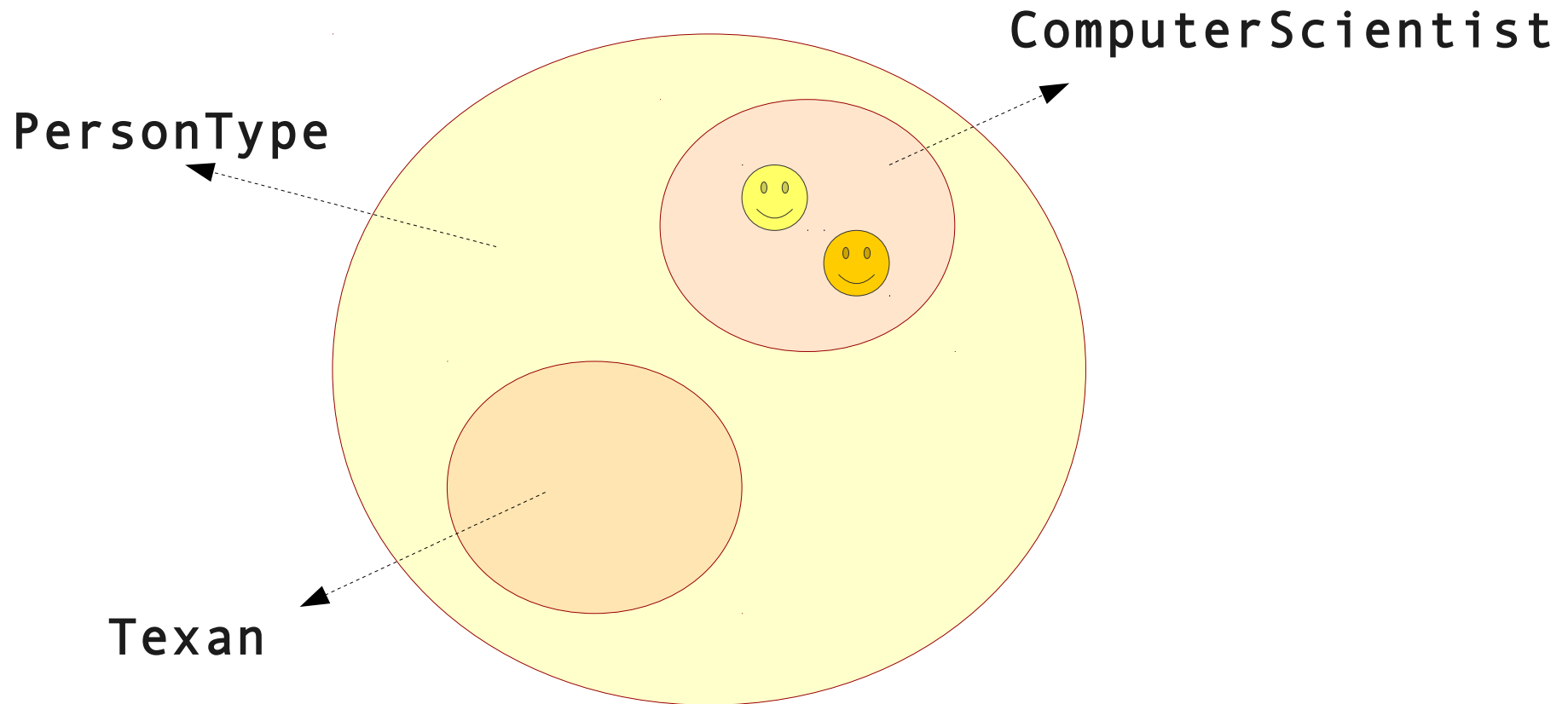
- **Kategóriákra** épül (*categories/classes/collections*), amelyek:
 - Halmazokként képzelhetők el (de nem szigorú matematikai értelemben)
 - Hierarchiába szerveződnek (*generalization ~ supersets, specialization ~ subsets*)
 - A kategóriák elemei a **példányok** (*instances*)
- A hierarchiában **nódusok** (*nodes*) vannak
- Fontos fogalom még a **particionálás** (*partitioning*), azaz ahogyan egy kategóriát részhalmazokra bontunk (többféle felbontás is lehetséges)

Cyc ontológia: Partíciók és predikátumok

- Univerzum halmaz: **Thing**
 - Egyfajta particionálása:
InternalMachineThing ↔ **RepresentedThing**
Lisp beépített elemei CycL-ben reprezentált elemek
 - Másfajta particionálása:
IndividualObject ↔ **Collection**
pl. *Fred* pl. *Thing* [?], *Table*
nem halmazok halmazok
lehetnek részeik lehetnek példányaik
nem lehetnek példányaik
- **Predikátumok** ~ változók, pl. *age(x)*, *weight(x)*
- Ezeket nem lehet **Collections**-re alkalmazni (az Asztalnak, mint halmaznak nincs súlya, csak a példányainak lehet)

Cyc ontológia: Kategóriák (Collections)

- Lehetséges a 'collections of collections', például:



- De a 'collections of collections of collections' már nem!
(régén volt, de csak káoszhoz vezetett, és inkább kivették)

Cyc ontológia: Substances ↔ Individuals

- Ha fogunk egy darab fát, és nekiesünk fejszével, az eredmény sok kis darab fa → **Substances**
ilyen még: homok, mogyoróvaj, sétálás...
- Ha fogunk egy darab asztalt, és ugyanezt megteesszük vele, az eredmény nem sok kis darab asztal lesz → **Individuals**
- **Fontos:** minden individuum valamilyen szubsztanciából áll
ÉS minden szubsztanciát individuumok alkotnak
- Akkor mi legyen pl. Bertrand Russell szubsztanciája?
 - 1) BertrandRussellStuff?
 - 2) AnimalMatter?
 - 3) Substance?

Cyc ontológia: Substances ↔ Individuals

- a **Substances** és **Individuals** halmazok extenzionálisan egyenlőek, de más intenzionális leírásuk van
- A különféle anyagok a **SubstanceType** osztály példányai
- A különféle egyének az **ObjectType** osztály példányai

- Miért jó ez?

Belső (*intrinsic*)
az individuum részeinek
is megvan

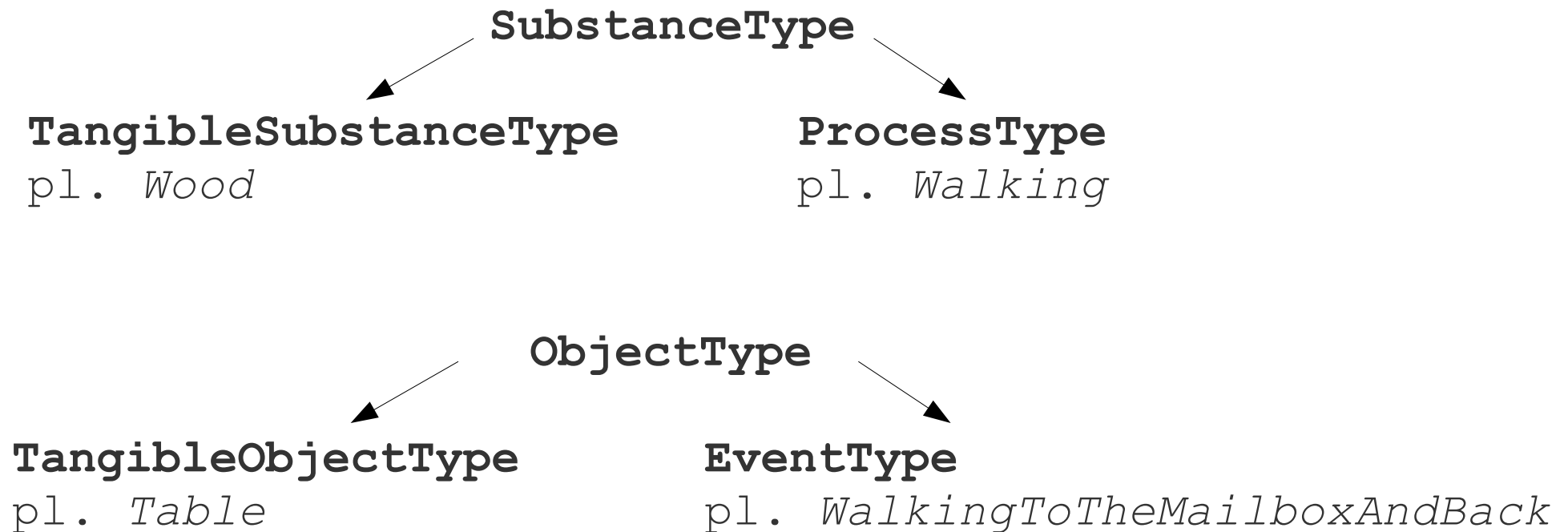
↔

külső (*extrinsic*) tulajdonságok
az individuum részeinek nincs meg



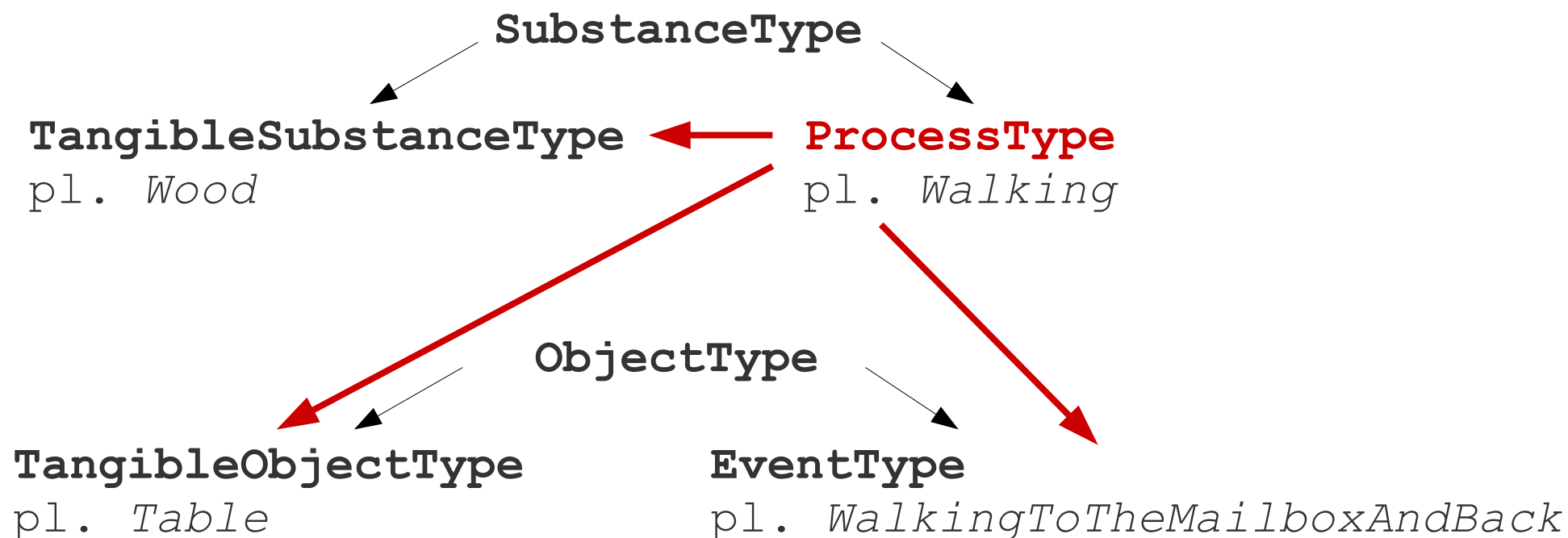
Cyc ontológia: Idő-probléma

- Nem csak térben, de időben is tudunk darabolni!
(*cutting up smth temporally*)



Cyc ontológia: Idő-probléma

- Drámai fordulat: az anyagoknak, tárgyaknak, individuumoknak is van ideje,
pl. az íróasztal a 412-ben addig az enyém, amíg itt dolgozok
(*subAbstraction*)



Cyc ontológia: Események

- Az **Event** bármelyik példányának lehetnek temporális tulajdonságai, pl. *duration*, *endsBeforeTheStartOf(x)* ...
- Ennek kifejezésére 2 idő-absztrakció:
 - Intervallum-alapú:
2 primitívum: *before*, *simultaneousWith(x)*
 - Halmaz-alapú:
pl. ezt akarjuk kifejezni:
“az emberek nem esznek és alszanak egyszerre”
így: az evés és alvás időpontjainak halmaza, és ezek mindig diszjunktak

Cyc ontológia: Történések

- Hogyan ábrázolhatunk egy történést, **Action**-t?
- Hasonlóan ahhoz, ahogy a fizikai objektumokat ábrázoljuk, a törtézés szerkezetét a részeire (aleseményeire) tett megszorítások határozzák meg

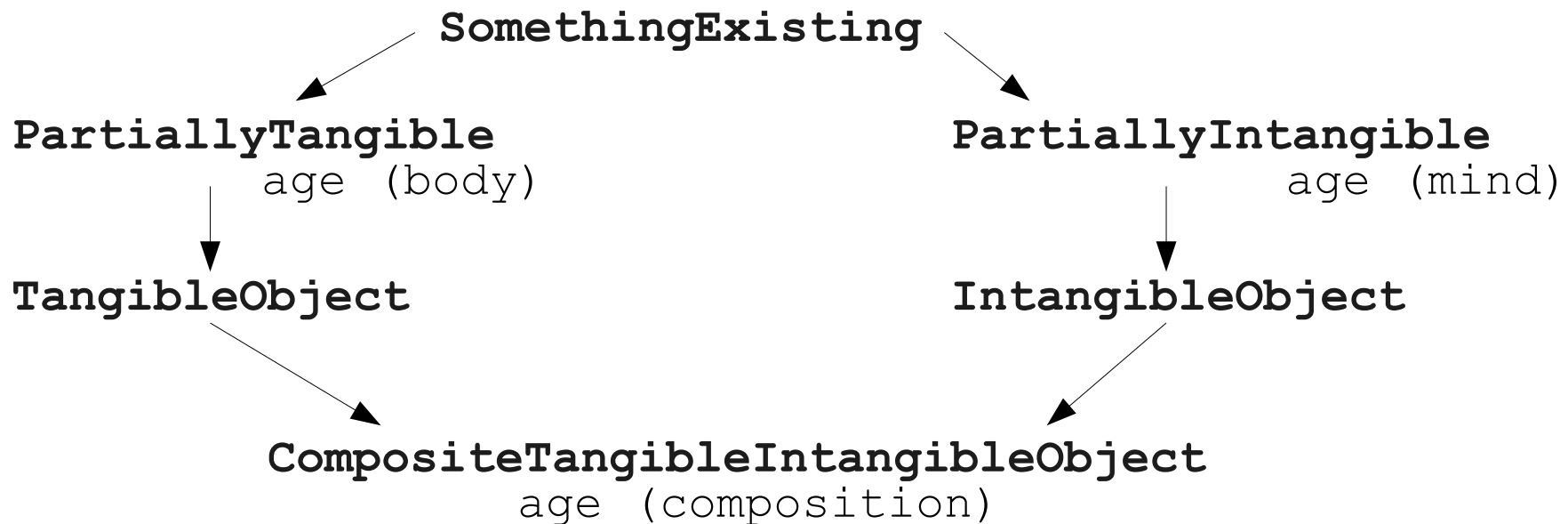
Action =

```
ActorSlots = {
    Actor1 => performer,
    [subAbstractions* for Actor1];
    Actor2 => instrument,
    [subAbstractions* for Actor2]
}
SubEventSlots = {ki-kivel-mit...}
```

`subAbstractions`: az actor állapota a törtézés előtt, közben és után

Cyc ontológia: Összetett objektumok

- Mi a baj a **Frankenstein-szörnyel**?
a teste n éves, az elméje m éves, az összegyűrés k éves
→ ilyenkor mi a teendő: új predikátumok? (pl. *ageOfMyMind*)
... jobb lenne a meglévő *age* predikátumot használni
- Ők úgy oldották meg, hogy új osztályokat hoztak létre:



Zárszó

- Az “egy általános megoldás minden problémára” csak álom
- Sok kis részmegoldás összegyűrése sok esetben hatékonynak bizonyult
- **Kritika:** napjaink egyik legellentmondásosabb mesterséges intelligencia projektje
 - túl bonyolult
 - csak manuálisan bővíthető
 - nincs jól dokumentálva
 - nem egyértelmű, mi a “külső” és “belső” tulajdonság
- De azért csak van benne valami, ha ‘84-től napjainkig tartja magát :)