Organon F 24 (Supplementary Issue) 2017: 5-26

Mereologická struktura procedur

MARIE DUŽÍ

Katedra informatiky FEI. VŠB-Technická Universita v Ostravě
17. listopadu 15. 708 33 Ostrava. Czech Republic
marie.duzi@vsb.cz

ABSTRACT: The paper deals with properties of structured procedures from a mereological point of view, i.e. with respect to their constitutive elements. As a result, it is argued that procedures amount to structured complexes (wholes) made of uniquely determined parts and that the part-whole relation is of the partial-ordering type. However, such a mereology is not a standard one: the principle of extensionality and the idempotence law do not hold there.

KEYWORDS: Mereology – procedure – structure – TIL construction.

1. Úvod

Téměř před deseti lety (jak ten čas letí) jsem psala článek Duží (2007) do Festschriftu na počest našeho milého kolegy Pavla Cmoreje, a nyní je zde opět jubileum a čas pro další článek. V onom článku z r. 2007 se mimo jiné zamýšlím nad klasickou mereologií, která považuje vztah celek-část za vztah mezi fyzickými individui a ukazuji, že *pokud* je vztah celek-část opravdu vztahem mezi individui, pak je to zcela náhodný vztah a kterákoli individua mohou do takového vztahu vstupovat. Můžeme tedy mít „warrots“, tj. individua skládající se z velryby (*whale*) a mrkve (*carrot*), apod. Navíc tento přístup nevysvětlí problémy klasické mereologie, jako např. odpověď na otázku, jaký je rozdíl mezi kusem kamene a Michelangelovým Davidem. Vždyť obě individua se skládají ze stejných částí, a tedy extensionální princip „stejné části = stejný celek“ nám říká, že zde není žádný rozdíl. Jistě, Michelangelo sice údajně říká, že „v každém kameni je ukryta socha“, ale přesto cítíme, že mezi oním kusem kamene a nádhernou Michelangelovou sochou je značný rozdíl. Podobně, klasický příklad je dítě hrající si s legem. Pokud ono dítě postaví z kostek lega dům, pak jej zboří (jak děti s oblibou dělají), a ze stejných součástí postaví auto, klasická mereologie tvrdí, že se jedná o totéž individuum. Na tuto otázku v podstatě odpovídá Tichý v (1995, 179-180), že zde není žádná záhada, neboť jakožto *individua* jsou onen kus kamene a nádherná socha, či dům nebo auto postavené z kostek lega opravdu totožná, neboť individua jsou *jednoduchá* a *holá*. Nemají žádnou strukturu a žádnou netriviální vlastnost nutně. Pouze ona *procedura* vytesání sochy Davida z kusu kamene či postavení auta nebo domečku z kostek lega je *strukturovaná*, skládá se z částí neboli *konstituentů*.[1]

Cílem tohoto článku je proto zamyslet se nad tím, jaká je struktura procedur a specifikovat její vlastnosti. Jelikož je snad nejvýznačnějším rysem Tichého transparentní intensionální logiky (TIL – viz Tichý 1988) její *procedurální* sémantika, a TIL pracuje s procedurami jako s plnoprávnými objekty, je přirozené, že budu zkoumat mereologickou strukturu procedur právě v systému TIL. Hodlám ukázat, že mereologická struktura procedur splňuje v mnoha ohledech zákony klasické mereologie, ale zároveň je v některých ohledech neklasická.

Zbytek tohoto článku má následující strukturu. V kapitole 2 stručně shrnu základní definice a principy TIL. V kapitole 3 představím mereologickou strukturu procedur obecně a konstrukcí speciálně, což je hlavním cílem tohoto článku. Závěrečná kapitola 4 shrnuje dosažené výsledky.

2. Základní definice a principy TIL

V této kapitole podávám stručné shrnutí nejdůležitějších pojmů a definic systému TIL v podobě, jak je tento systém představen zejména v Duží – Jespersen – Materna (2010) nebo v Duží – Materna (2012).

Z hlediska filosofického je nejdůležitějším rysem systému TIL jeho *procedurální* sémantika. Významem výrazu není množinový objekt označený daným výrazem, pokud vůbec takový objekt existuje, nýbrž *způsob danosti* takového objektu. Tento Fregův způsob danosti je v TIL explikován jako *abstraktní*

---

[1] Na toto téma vedl Pavel Cmorej s Pavlem Tichým velice bohatou, plodnou a krásnou diskusi, která byla publikována v časopise *Organon F*, viz. Cmorej, Tichý (1998a, 1998b).

procedura, která při svém provádění dává na výstupu daný označený objekt, nebo v určitých dobře definovaných případech neprodukuje nic, procedura může být nevlastní, to je může selhávat v prezentaci určitého objektu.

Z formálního hlediska je TIL parciální, typovaný, hyperintensionální λ-kalkul. *Parciální*, jelikož pracuje s parciálními funkcemi, tj. funkcemi, které na některých argumentech nemají žádnou hodnotu. *Typovaný*, protože každý objekt (včetně oněch procedur) má přiřazen určitý typ, do kterého patří. A konečně *hyperintenzionální*, protože λ-termy jazyka TIL neoznačují funkce jakožto množinová zobrazení (čili Churchovy „*functions-in-extension*"), nýbrž algoritmicky strukturované procedury (zhruba Churchovy „*functions-in-intension*"), jejichž produkty jsou ony (parciální) funkce. Toto pojetí je v souladu s původním pojetím λ-kalkulu, viz např. Barendregt (1997, 184).

Tichý definoval šest druhů těchto abstraktních procedur a nazval je *konstrukce.* Jakožto procedury mohou být konstrukce (alespoň v principu) prováděny tak, že operují na vstupních objektech (nižšího řádu) a produkují nanejvýš jeden výstupní objekt, který jsou typovány produkovat. Neprocedurální objekty jako individua, funkce, množiny apod. nemohou být prováděny. Konstituenty dané konstrukce tedy nemohou být neprocedurální objekty, nýbrž pouze její pod-konstrukce. Proto potřebujeme jednoduché konstrukce, které dodávají vstupní objekty, na kterých má daná konstrukce operovat. Takovéto atomické konstrukce jsou dvě, a to *Trivializace* a *Proměnné.*

Operační smysl *Trivializace* je podobný smyslu konstant ve formálních jazycích. Trivializace ^0X objektu X prezentuje či referuje k objektu X bez pomoci jiných procedur. Zde by mohlo být užitečné srovnání s programovacími jazyky. Trivializace ^0X je prostě pointer na reprezentanta objektu X, neboť abychom mohli s objektem X pracovat, musíme jej nejdříve nějak uchopit, specifikovat. Trivializace je takovýto základní mechanismus jednoduché reference k objektu.

Proměnné produkují objekty v závislosti na valuaci, říkáme, že v-konstruují. Přijímáme objektivní verzi Tarského koncepce proměnných. Každému typu (viz Def. 2) je přiřazeno spočetně nekonečně mnoho proměnných, které přes tento typ „rangují", tj. v-konstruují objekty tohoto typu. Navíc, objekty každého typu (alespoň o dvou prvcích) mohou být uspořádány do nekonečně mnoha sekvencí. Valuace v pak vybere jednu z těchto sekvencí objektů daného typu a první proměnná, která je tomuto typu přiřazena, v-konstruuje první objekt sekvence, druhá proměnná druhý objekt, atd.

Tedy provedení Trivializace nebo proměnné nikdy neselhává, vždy dává nějaký objekt. Jinými slovy, proměnné ani Trivializace nejsou nikdy v-nevlastní.

Na druhé straně provádění molekulárních konstrukcí může selhat v produkování objektu, tj. ostatní konstrukce (kromě λ-Uzávěru) mohou být v-nevlastní. Podobně jako v ostatních λ-kalkulech máme v TIL dvě duální procedury, a to proceduru aplikace funkce produkované konstrukcí X na n-tici argumentů produkovaných konstrukcemi $X_1,\ldots,X_m$, což je *Kompozice* $[X\, X_1\ldots X_m]$, a proceduru specifikace funkce λ-abstrakcí od hodnot proměnných, což je *Uzávěr* $[\lambda x_1\ldots x_m\, X]$. Kromě těchto standardních potřebujeme ještě dvě svým způsobem nestandardní procedury, a to Provedení a Dvojí provedení. Potřebujeme je proto, že konstrukce se nemusí vyskytovat pouze v módu provádění, ale také v módu zmiňování, kdy k ní pouze referujeme jakožto k objektu. Takovýto výskyt je zaručen procedurou Trivializace. Chceme-li však efekt Trivializace zrušit, a přejít do módu provádění, použijeme explicitní specifikaci provedení, tj. konstrukci Dvojí Provedení.

Definice 1 (*konstrukce*)

(i) *Proměnné* x, y, … jsou *konstrukce*, které konstruují objekty (prvky jim přiřazeného typu) v závislosti na valuaci, tj. *v-konstruují*.

(ii) Je-li X jakýkoli objekt (i konstrukce), ${}^0X$ je *konstrukce Trivializace*, která *konstruuje* X.

(iii) Nechť X, $Y_1,\ldots,Y_m$ jsou konstrukce. Pak *Kompozice* $[X\, Y_1\ldots Y_m]$ je následující *konstrukce*. Pro libovolnou valuaci v, Kompozice $[X\, Y_1\ldots Y_m]$ je *v-nevlastní*, jestliže některá z konstrukcí X, $Y_1,\ldots,Y_m$ je v-nevlastní, nebo jestliže X v-konstruuje funkci, která není definována na n-tici objektů v-konstruovaných konstrukcemi $Y_1,\ldots,Y_m$. Jinak, jestliže X v-konstruuje funkci f, která je definována na n-tici objektů v-konstruovaných konstrukcemi $Y_1,\ldots,Y_m$, pak $[X\, Y_1\ldots Y_m]$ *v-konstruuje* hodnotu této funkce f na této n-tici objektů.

(iv) Nechť x_1, x_2, …, x_m jsou navzájem různé proměnné a Y konstrukce. Pak (λ-)*Uzávěr* $[\lambda x_1\ldots x_m\, Y]$ je *konstrukce*, která *v-konstruuje* funkci f následujícím způsobem. Nechť B_1, …, B_m jsou objekty, které jsou po řadě v-konstruovány proměnnými x_1, …, x_m. Pak hodnotou funkce f na n-tici objektů B_1, …, B_m je objekt (pokud nějaký), který je $v(B_1/x_1,\ldots,B_m/x_m)$-konstruován konstrukcí Y, kde $v(B_1/x_1,\ldots,B_m/x_m)$ je valuace stejná jako v, až na to, že přiřazuje objekt B_1 proměnné x_1, …, B_m proměnné x_m.

(v) *Provedení* ${}^1X$ je *konstrukce*, která buď *v-konstruuje* objekt v-konstruovaný konstrukcí X, nebo pokud X není konstrukce nebo je v-nevlastní, je rovněž ${}^1X$ *v-nevlastní*, tj. nekonstruuje žádný objekt.

(vi) *Dvojí Provedení* ^2X je *konstrukce*. Jestliže X v-konstruuje konstrukci X' a X' v-konstruuje objekt Y, pak ^2X *v-konstruuje* Y. Jinak, tj. pokud X není konstrukce, nebo X ne v-konstruuje jinou konstrukci, nebo X v-konstruuje v-nevlastní konstrukci, je konstrukce ^2X *v-nevlastní*.

(vii) Nic jiného není *konstrukce* než dle (i) – (vi). □

Pozn.: Provedení ^1X a X nejsou identické konstrukce, jsou pouze ekvivalentní (v tom smyslu, že pro libovolnou valuaci v platí, že ^1X a X v-konstruují jeden a tentýž objekt, nebo jsou obě v-nevlastní), pokud je X konstrukce. Jinak je ^1X nevlastní konstrukce a X je neprocedurální objekt.

TIL ontologie je velice bohatá. Obsahuje konstrukce konstrukcí, konstrukce funkcí, funkčních hodnot, včetně objektů, které nejsou konstrukcemi. Proto je potřebné tuto ontologii uspořádat do hierarchie typů, aby nenastal paradox bludného kruhu a abychom mohli kontrolovat, zda je daná konstrukce sestavena v souladu s typovými omezeními, nebo je nevlastní z důvodu chybného typování. K tomu slouží induktivní definice *rozvětvené hierarchie typů*. Tato hierarchie typů je v podstatě dvourozměrná nekonečná tabulka. V řádcích se zvyšuje stupeň zanoření funkcí při jejich skládání, ve sloupcích pak řád konstrukce.

Definice 2 (*rozvětvená hierarchie typů*)

Nechť B je *báze*, tj. kolekce navzájem různých neprázdných množin. Pak:

$\mathbf{T_1}$ *(typy řádu 1).*

(i) Každý prvek báze B je elementární *typ řádu 1 nad B*.

(ii) Nechť α, β_1, …, β_m ($m > 0$) jsou typy řádu 1 nad B. Pak kolekce (α β_1 … β_m) všech m-árních parciálních funkcí z $\beta_1 \times \ldots \times \beta_m$ do α je funkcionální *typ řádu* 1 *nad B*.

(iii) Nic jiného není *typ řádu 1 nad B* než dle (i) a (ii).

$\mathbf{C}_n$ *(konstrukce řádu n)*

(i) Nechť x je proměnná, která v-konstruuje prvky typu řádu n. Pak x je *konstrukce řádu n nad B*.

(ii) Nechť X je prvek typu řádu n. Pak ^0X, ^1X, ^2X jsou *konstrukce řádu n nad B*.

(iii) Nechť X, X_1,…, X_m ($m > 0$) jsou konstrukce řádu n nad B. Pak $[X\ X_1 \ldots X_m]$ je *konstrukce řádu n nad B*.

(iv) Nechť $x_1,\ldots x_m$, X ($m > 0$) jsou konstrukce řádu n nad B. Pak $[\lambda x_1 \ldots x_m\ X]$ je *konstrukce řádu n nad B*.

(v) Nic jiného není *konstrukce řádu n nad B než dle* $\mathbf{C}_n$ (i) – (iv).

$\mathbf{T}_{n+1}$ *(typy řádu n+*1)

Nechť $*_n$ je kolekce všech konstrukcí řádu *n* nad B. Pak

(i) $*_n$ a každý typ řádu *n* jsou *typy řádu n*+1.

(ii) Jsou-li α, $\beta_1,\ldots,\beta_m$ ($m > 0$) typy řádu *n*+1 nad *B*, pak ($\alpha\ \beta_1 \ldots \beta_m$) je *typ řádu n*+1 *nad B*.

(iii) Nic jiného není *typ řádu n+1 nad B* než dle (i) a (ii). □

Báze obsahuje obvykle kolekce množin *neprocedurálních* objektů, které tedy nejsou konstrukcemi. Pro účely analýzy přirozeného jazyka předpokládáme tuto bázi elementárních typů:

ο: množina pravdivostních hodnot {**P, N**};
ι: množina individuí (konstantní universum diskursu);
τ: množina reálných čísel (a také časových okamžiků);
ω: množina možných světů.

Empirické výrazy označují empirické *podmínky*, které mohou, ale nemusejí, být splněny v daném světě a čase jejich vyhodnocování. Tyto podmínky modelujeme jako *(PWS) intense.* PWS intense jsou objekty patřící do typu ($\beta\omega$): funkce s doménou možných světů ω a hodnotami libovolného typu β. Typ β je často typem *chronologie* α-objektů, tj. funkce typu ($\alpha\tau$). Proto jsou α-intense nejčastěji funkce patřící do typu (($\alpha\tau$)ω), což zkracujeme jako '$\alpha_{\tau\omega}$'. Typickou analýzou empirického výrazu je tedy konstrukce intense, nejčastěji Uzávěr ve tvaru $\lambda w \lambda t$ [… *w t* …], kde proměnná *w* ranguje přes ω a *t* přes τ. *Extense* jsou objekty typu α, kde $\alpha \neq (\beta\omega)$ pro libovolný typ β. Množiny a relace modelujeme jejich charakteristickými funkcemi. Tak například ($o\iota$) je typ množiny individuí, ($o\iota\iota$) je typ relace mezi individui. Příklady intensí jsou *proposice* patřící do typu $o_{\tau\omega}$, *vlastnosti* individuí typu $(o\iota)_{\tau\omega}$, *binární vztahy* mezi individui typu $(o\iota\iota)_{\tau\omega}$, individuové *úřady* (nebo role) typu $\iota_{\tau\omega}$.

Logické objekty jako *pravdivostní funkce* a *kvantifikátory* jsou extensionální: $\wedge$ (konjunkce), $\vee$ (disjunkce), $\supset$ (implikace) jsou typu (ooo), $\neg$ (negace) je typu (oo). *Kvantifikátory* $\forall^\alpha$, $\exists^\alpha$ jsou typově polymorfní totální funkce typu ($o(o\alpha)$), pro libovolný typ α, které jsou definovány takto: *Všeobecný kvantifikátor* $\forall^\alpha$ je funkce, která přiřazuje třídě *A* objektů typu α pravdivostní hodnotu **P**, jestliže *A* obsahuje všechny prvky typu α, jinak **N**. *Existenční kvantifikátor* $\exists^\alpha$ je funkce, která přiřazuje třídě *A* objektů typu α hodnotu **P**, jestliže *A* je neprázdná, jinak **N**.

Aby zápis konstrukcí nebyl zahlcen, píšeme typy objektů většinou zvlášť. K tomu používáme tuto *notaci*:

- 'X/α' znamená, že objekt X je (tj. patří do) typu α.
- '$X/*_n \rightarrow_v \alpha$' znamená, že konstrukce X patří do typu $*_n$, a X je typována v-konstruovat objekt typu α. Není-li řád konstrukce v dané chvíli podstatný, píšeme prostě '$X \rightarrow_v \alpha$', a pokud to, co je v-konstruováno, nezávisí na valuaci v, píšeme '$X \rightarrow \alpha$'.
- V TIL platí, že proměnné w a t jsou užívány jako rangující přes možné světy a časy, tj. $w/*_1 \rightarrow_v \omega$ a $t/*_1 \rightarrow_v \tau$.
- Jestliže $C \rightarrow_v \alpha_{\tau\omega}$, pak pro často užívanou Kompozici $[[C\ w]\ t]$, což je intensionální sestup (neboli extensionalizace) α-intense v-konstruované konstrukcí C, používáme zkrácený zápis 'C_{wt}'.
- Často vynecháváme vnější závorky Uzávěru, pokud to nevede k nedorozumění. Pro logické spojky a identitu (=) často používáme infixní notaci bez Trivializace z důvodu snadnějšího čtení zápisu konstrukcí.

Příklady:

- Čísla 0 a 1, funkce sčítání (+) a dělení (:) nejsou konstrukce, jsou to objekty typu řádu 1: $0/\tau$, $1/\tau$, $+/(\tau\tau\tau)$, $:/(\tau\tau\tau)$.
- Proměnná x, která ranguje přes čísla, je konstrukce řádu 1, $x/*_1 \rightarrow_v \tau$.
- Kompozice $[^0+\ x\ ^01]/*_1 \rightarrow_v \tau$ v-konstruuje následníka čísla, v-konstruovaného proměnnou x.
- Uzávěr $\lambda x\ [^0+\ x\ ^01]/*_1 \rightarrow (\tau\tau)$ konstruuje funkci následníka.
- Kompozice $[^0:\ x\ ^00]/*_1 \rightarrow_v \tau$ je typována v-konstruovat čísla, avšak nekonstruuje nic, je v-nevlastní pro libovolnou valuaci v.
- Uzávěr $\lambda x\ [^0:\ x\ ^00]/*_1 \rightarrow (\tau\tau)$ konstruuje degenerovanou funkci, která nemá hodnotu na žádném ze svých argumentů.
- Je-li $Improper/(o*_1)$ třída konstrukcí řádu 1 v-nevlastních pro libovolnou valuaci v, pak $^0Improper/*_2 \rightarrow (o*_1)$, $^0[^0:\ x\ ^00]]/*_2 \rightarrow *_1$ a Kompozice $[^0Improper\ ^0[^0:\ x\ ^00]]/*_2 \rightarrow o$ konstruuje $\mathbf{P}/o$, tj. $[^0=\ [^0Improper\ ^0[^0:\ x\ ^00]]\ ^0\mathbf{P}]$, kde $=/(ooo)$.

Z výše uvedeného vyplývá, že každý objekt, se kterým v TIL pracujeme či na něm operujeme, včetně konstrukcí, má jednoznačně přiřazen typ, do kterého patří. Avšak dle Def. 2 platí, že každý typ řádu n je rovněž typem řádu n+1. Toto zvyšování řádů je zde (mimo jiné) z toho důvodu, aby typování nebylo příliš restriktivní, čili abychom mohli například kvantifikovat přes množinu konstrukcí řádu n, která bude obsahovat prvky $*_1, *_2, \ldots, *_n$. Avšak každá konstrukce má jednoznačně přiřazen typ $*_n$, kde n je její nejmenší řád.[2]

Navíc, konstrukce jsou typovány v-konstruovat objekt určitého jednoznačně určeného typu. Např. jestliže $x_1 \rightarrow_v \alpha_1, \ldots, x_n \rightarrow_v \alpha_n$, $Y \rightarrow_v \beta$, pak Uzávěr $[\lambda x_1 \ldots x_n\ Y]$, v-konstruuje funkci typu $(\beta\ \alpha_1 \ldots \alpha_n)$. A jestliže konstrukce $X \rightarrow_v (\beta\ \alpha_1 \ldots \alpha_n)$, $Y_1 \rightarrow_v \alpha_1, \ldots, Y_m \rightarrow_v \alpha_m$, pak Kompozice $[X\ Y_1 \ldots Y_m] \rightarrow_v \beta$. Výjimkou jsou pouze ty konstrukce, které jsou v-nevlastní z důvodu nekoherentního typování, a tedy nejen že ne v-konstruují žádný objekt, ale nelze někdy ani určit typ objektů, které by měly v-konstruovat. Například konstrukce $^1 Tom/*_1$, kde Tom/ι, je v-nevlastní pro každou valuaci v, a jelikož individuum nelze provést, nelze ani určit typ objektu, který by tato konstrukce měla v-konstruovat. Jiným příkladem může být konstrukce $[^2c\ ^0 1]$, kde proměnná $c/*_2 \rightarrow_v *_1$. Pokud proměnná c v-konstruuje konstrukci funkce typu $(\tau\tau)$, např. funkci následníka $Succ/(\tau\tau)$, není pro takovou valuaci konstrukce $[^2c\ ^0 1]$ v-nevlastní, protože v-konstruuje číslo $2/\tau$. Ovšem pro valuaci, která přiřadí proměnné c konstrukci funkce jiného typu než $(\alpha\tau)$, pro libovolný typ α, bude tato Kompozice v-nevlastní z důvodu chybného typování, a nemůžeme určit typ, jehož prvky by měla Kompozice $[^2c\ ^0 1]$ v-konstruovat. Aby toto nenastalo, musíme zadat nejen $c/*_2 \rightarrow_v *_1$, ale i navíc $^2c/*_3 \rightarrow_v (\alpha\tau)$.

(λ-)Uzávěr $[\lambda x_1 \ldots x_m\ Y]$ není v-nevlastní pro žádnou valuaci v, protože vždy v-konstruuje funkci. Dokonce i v tom případě, kdy je konstituent Y v-nevlastní pro každou valuaci v, není Uzávěr v-nevlastní. Avšak v takovém případě je konstruován bizarní objekt. Je to tzv. *degenerovaná funkce*, která nemá hodnotu na žádném ze svých argumentů. Příkladem jsou $\lambda x\ [^0{:}\ x\ ^0 0]/*_1 \rightarrow (\tau\tau)$, $\lambda y\ [^0{+}\ y\ [^0 Cotg\ ^0\pi]]/*_1 \rightarrow (\tau\tau)$, kde proměnné $x, y/*_1 \rightarrow_v \tau$, $:/(\tau\tau\tau)$, $+/(\tau\tau\tau)$, $Cotg/(\tau\tau)$ je funkce kotangens, π/τ.

3. Mereologická struktura procedur

V této kapitole podám několik argumentů ve prospěch následujících tezí. Každá procedura je celistvý objekt, který může být jakožto celek prováděn za účelem produkování nanejvýš jednoho objektu, a který může jakožto celek

---

2 Raclavský nazývá tento typ *nativní*, viz Raclavský – Kuchyňka – Pezlar (2015, 54-55).

figurovat jako objekt, na kterém jiné procedury operují. Každá procedura je však *strukturovaný* celek s jednoznačně určenými částmi, které spolu navzájem interagují, přičemž tyto části spojuje dohromady v jeden celek právě ona procedura. Procedura není pouhým výčtem, seznamem či množinou svých vlastních částí. Množina ani výčet nemohou být provedeny, kdežto procedura (alespoň v principu) ano.

3.1. Dva druhy celků

Russell (1903, §136) rozlišuje dva druhy celků.

> [An *aggregate*] is completely specified when all its simple constituents are specified; its parts have no direct connection *inter se*, but only the indirect connection involved in being parts of one and the same whole. But other wholes occur, which contain relations or what may be called predicates, not occurring simply as terms in a collection, but as relating or qualifying. Such wholes are always propositions. These are *not completely specified when their parts are all known*. (Russell 1903, §136, kurzíva moje)

Tedy části strukturovaného celku na sebe vzájemně působí, spolu interagují. Prvky množiny jsou na sobě zcela nezávislé, nijak neinteragují. Teorie množin řeší pouze problém externí, jak vymezit hranice, které odliší její prvky od všeho ostatního v universu. Mereologie vhodná pro opravdu strukturované celky jako procedury řeší navíc interní problém, jak zajistit, aby části navzájem spolupracovaly tak, že vytvoří jeden celek.

Toto bylo zřejmé již Bernardovi Bolzanovi. Bolzano (1837) ukazuje, že pouhým výčtem komponent obsahu pojmu není pojem definován.[3] Musíme vzít v úvahu *způsob spojení* těchto komponent.[4] Bolzano vypracoval systematickou realistickou teorii pojmů (*Vorstellungen an sich*), jakožto objektivních entit obdařených strukturou. V tradiční teorii pojmu založené na Port Royal logice je pojem definován jako dvojice (extense, intense).[5] Intense (obsah) pojmu C je souhrn atributů $C_1,\dots,C_n$, zatímco extense (rozsah) pojmu C je

---

[3] Pojmy explikujeme v TIL jako uzavřené procedury. Viz zejména Materna (1998; 2004).

[4] V originálu: „die *Art, wie* diese Theile untereinander verbunden sind“ (Bolzano 1837, §244)

[5] Často také (extent, intent), nebo rozsah (*Umfang*) a obsah (*Inhalt*). Nadále budu používat v češtině vžité termíny „rozsah“ a „obsah“.

průnik množin objektů, které tyto jednotlivé atributy splňují. Proto samozřejmě platí zákon *inverse rozsahu a obsahu* (čím větší obsah, tím menší rozsah, a naopak). Například rozsah pojmu „občané Prahy“ je větší než rozsah pojmu „občané Prahy hovořící německy“.

Bolzano kritizuje tento zákon v Bolzano (1837, §120), kde uvádí tento příklad:

1. Člověk, který rozumí všem evropským jazykům
2. Člověk, který rozumí všem *živým* evropským jazykům

První pojem má menší obsah než druhý, avšak (kontra zákon inverze) jeho rozsah je rovněž menší než rozsah pojmu druhého. Tímto příkladem chce Bolzano zřejmě ukázat, že *způsob* spojení jednotlivých částí obsahu je důležitý. Klasická Port-Royal škola uvažuje pouze spojení konjunktivní. Pak ovšem zákon inverze je triviálním důsledkem teorie množin.[6] Pokud je však spojení jiné než konjunktivní, zákon platit nebude. Např. pro pojmy „občané Prahy hovořící česky“ a „občané Prahy hovořící česky *a* německy“ zákon inverze platí. Avšak samozřejmě neplatí pro pojmy „občané Prahy hovořící česky“ a „občané Prahy hovořící česky *nebo* německy“.

Bolzano také ukazuje, že pojem opravdu není definován pouze svým obsahem. Jako příklad uvádí dvojici matematických pojmů 3^5 a 5^3. Tyto pojmy mají přesně stejný obsah, totiž pojem čísla 3, pojem čísla 5 a pojem funkce umocňování. Přesto jsou to rozdílné pojmy, protože *procedura* spočívající v aplikaci funkce umocňování na dvojici $\langle 3, 5\rangle$ je jiná než procedura aplikace funkce mocnina na dvojici $\langle 5, 3\rangle$.

Jakožto mnozí géniové, Bolzano trochu předběhl svou dobu a nebyl plně pochopen a oceněn. Např. Bar-Hillel (1950) kritizuje Bolzanův (1837, §148) příklad dvou různých pojmů trojúhelníka. Dle Bolzana je pojem trojúhelníka jakožto geometrického obrazce o třech *stranách* rozdílný od pojmu trojúhelníka definovaného jako geometrický obrazec o třech *úhlech* takových, že součet jejich velikostí je roven 180°. Avšak Bar-Hillel ignoruje fakt, že jde o dva různé pojmy specifikující stejnou množinu geometrických obrazců (tj. trojúhelníků), zaměřuje se právě jen na tento společný produkt obou pojmů. Bar-Hillel říká:

[6] Těmito kritickými poznámkami nemíním snižovat užitečnost této teorie. Má řadu užitečných aplikací, např. tzv. „formální konceptuální analýza“, která je hojně využívána v informatice. Viz např. Ganter – Wille (1999).

> [it]s uncritical acceptance may lead to strange, even contradictory formulations. … The two occurrences of the word 'triangle', though differently defined, express both the property Triangle, have the property Triangle as their intension, so that the property Triangle is different from the property Triangle. (Bar-Hillel 1950, 108)

Ovšem Bolzano zde nemluví o vlastnosti být trojúhelníkem. Mluví o dvou různých pojmech, které mají různý obsah ale stejný rozsah čili o dvou různých procedurách, které produkují tutéž vlastnost (množinu trojúhelníků). A je samozřejmé, že někdo může vědět, že trojúhelník má tři strany aniž by věděl, že součet jeho vnitřních úhlů je roven 180°.

V této kapitole se tedy zaměřím na takové celky, jejichž vlastní části jsou jednoznačně určeny, je dán způsob spojení těchto částí do jednoho celku a tyto části spolu interagují. Jinými slovy, budu zkoumat strukturu *procedur*.

3.2. Konstrukce jako strukturované celky

Konstrukce *C* jakožto abstraktní procedura se skládá z určitých jednoznačně určených prováděcích kroků, tj. konstituentů neboli částí *C*, které musí být jednotlivě provedeny má-li být provedena celá konstrukce *C*. Tyto konstituenty operují na vstupních objektech, což mohou být objekty typu řádu 1 (tj. neprocedurální objekty) nebo konstrukce nižšího řádu, které pak ovšem nemají výskyt v módu provádění, nýbrž jsou pouze zmiňovány jako objekty, na kterých jiné konstituenty operují. Je ovšem důležité si uvědomit, že vstupní či výstupní objekty nejsou konstituenty dané konstrukce. Konstituenty mohou být pouze ty její podkonstrukce, které mají výskyt v módu provádění. Navíc, jednotlivé konstituenty dané konstrukce spolu vzájemně interagují. Konstruují funkce, které jsou aplikovány na argumenty, které dodávají jiné konstituenty. Tedy výstup jednoho konstituentu se stává argumentem funkce, která je výstupem jiného konstituentu, atd.

Každá konstrukce *C* jakožto celek tedy může samotná figurovat coby objekt, na kterém jiné konstrukce operují neboli argument funkce, která je výstupem jiného prováděného konstituentu. V tom případě se daná konstrukce *C* vyskytuje *hyperintenzionálně*. Typickým příkladem jsou významy vět vyjadřujících propoziční či pojmové postoje, jak zpozoroval již Carnap v (1947, §§13*ff*). Carnap říká, že komplement postoje není ani extensionální ani intensionální, neboť zde selhává substituce logicky ekvivalentních klauzulí. Podobně Ludwig a Ray říkají:

> In general, one term can be substituted for another in 'that'-clauses salva veritate only if they are synonymous. Perhaps the most popular solution to the problem of providing a compositional semantics for natural languages aims to exploit this fact by treating 'that'-clauses as referring to intensional entities – entities (at least) as finely individuated as the meanings of sentences. (Ludwig – Ray 1998, 141)

Autoři zde užívají termín 'intensional entities' v obecném smyslu tak, jak jej užíval Church. V naší terminologii bychom užili termín 'hyperintensional entities', protože termín 'intensional' byl okupován PWS-sémantikou pro (extensionální) funkce s doménou možných světů.

Uvedu nyní jednoduchý příklad takového hyperintensionálního výskytu dané konstrukce v příslušné nadkonstrukci. Provedeme analýzu věty

„Tom počítá kotangent π"

a aplikujeme přitom *metodu analýzy* spolu s derivačním stromem příslušné konstrukce, přiřazené naší větě jakožto její význam. Nejprve však krátká úvaha. Jakého typu je objekt označený výrazem „počítá"? Je to jistě vztah individua (v našem případě Toma) k něčemu. Ovšem to něco nemůže být číslo, protože funkce kotangent není definována na argumentu π. Ale i kdyby byla definována, nemá smyslu počítat číslo, bez toho, že by byl nějak specifikován způsob zadání tohoto čísla. Počítání je proto vztah k tomuto způsobu zadání, tedy ke konstrukci.

a) *Typová analýza*: *Tom*/ι; *Počítá*/$(o\iota *_1)_{\tau\omega}$: vztah individua ke konstrukci; *Cot(angent)*/$(\tau\tau)$; π/τ.

b) *Syntéza.* Abychom mohli aplikovat vztah *Počítá* na individuum a konstrukci, musíme jej nejprve extensionalizovat: $[[{}^0\mathit{Počítá}\ w]\ t] \rightarrow_v (o\iota *_1)$, neboli zkráceně ${}^0\mathit{Počítá}_{wt}$. Jelikož Tom má vztah přímo ke *konstrukci*, tj. proceduře aplikace funkce kotangent na argument π, $[{}^0\mathit{Cot}\ {}^0\pi]$, musíme tuto Kompozici Trivializovat: ${}^0[{}^0\mathit{Cot}\ {}^0\pi]/*_2 \rightarrow *_1$. Rovněž k individuu Tom musíme referovat pomocí Trivializace. Dostáváme $[{}^0\mathit{Počítá}_{wt}\ {}^0\mathit{Tom}\ {}^0[{}^0\mathit{Cot}\ {}^0\pi]] \rightarrow_v o$. Nakonec abstrakcí od hodnot proměnných *w*, *t* konstruujeme propozici označenou naší větou:

$$\lambda w \lambda t\ [{}^0\mathit{Počítá}_{wt}\ {}^0\mathit{Tom}\ {}^0[{}^0\mathit{Cot}\ {}^0\pi]] \rightarrow_v o_{\tau\omega}$$

Všimněme si, že typy objektů, konstruovaných konstrukcemi $^0\pi$, ^0Cot a $[^0Cot\ ^0\pi]$, tj. po řadě τ, $(\tau\tau)$ a τ, jsou zde irelevantní, protože tyto konstrukce *nejsou* konstituenty celé konstrukce. Kompozice $[^0Cot\ ^0\pi]$ se zde vyskytuje pouze jako (druhý) argument relace *v*-konstruované Kompozicí $^0Počítá_{wt}$. Jinými slovy, chceme-li vyhodnotit pravdivostní hodnotu dané propozice v daném stavu světa *w*, *t*, pak stačí pouze ověřit, zda je Tom ve vztahu k této Kompozici $[^0Cot\ ^0\pi]$, avšak k tomu nemusíme provádět (bezvýslednou) aktivitu zjišťování hodnoty funkce kotangent na argumentu π, to je záležitost či problém Toma.

Abychom tyto obecné úvahy upřesnili, potřebujeme ještě několik definic.

Definice 3 (*podkonstrukce*)
Nechť *C* je konstrukce. Pak
(i) *C* je *podkonstrukce C*.
(ii) Je-li *C* konstrukce ^0X, ^1X nebo ^2X a *X* je konstrukce, pak *X* je *podkonstrukce C*.
(iii) Je-li *C* Kompozice $[X\ X_1 \ldots X_n]$, pak X, X_1, …, X_n jsou *podkonstrukce C*.
(iv) Je-li *C* Uzávěr $[\lambda x_1 \ldots x_n\ Y]$, pak *Y* je *podkonstrukce C*.
(v) Je-li *A* podkonstrukce *B* a *B* je *podkonstrukce C*, pak *A* je *podkonstrukce C*.
(vi) Nic jiného není podkonstrukce konstrukce *C* než dle bodů (i) – (v). □

Jak již bylo uvedeno, je důležité rozlišit výskyt podkonstrukce v módu provádění a v módu zmiňování, tj. výskyt jako objekt, na kterém jiné konstrukce operují. Je tomu tak proto, že částmi dané konstrukce jsou pouze ty podkonstrukce, které se vyskytují v módu provádění, a ne objekty, na kterých se mají tyto operace provádět. Nejprve však ještě jednou tento rozdíl pouze charakterizujeme, rigorosní definice následuje.

- Konstrukce *C* se může vyskytovat v módu zmiňování pouze v rámci nějaké její nadkonstrukce *D*;
- samotná konstrukce *C* musí tedy být (*v*-)konstruovaná nějakou podkonstrukcí *C'* konstrukce *D*;
- toto rozlišení dvou módů je nutno definovat pro *výskyty* jednotlivých podkonstrukcí, neboť jedna a tatáž podkonstrukce *C* se může vyskytovat v *D* jak v módu provádění, tak zmiňování.

Proto stručná charakteristika rozdílu výskytu v módu provádění a v módu zmiňování je: Nechť *C* je podkonstrukce konstrukce *D*. Pak výskyt *C* je *zmíněn v D*, jestliže provedení konstrukce *D* nevyžaduje provedení tohoto výskytu *C*. Jinak, ty výskyty podkonstrukcí konstrukce *D*, které je nutno provést, aby byla provedena celá *D*, včetně *D* samotné, jsou výskyty v módu provádění. Opět jednoduchý příklad:

(Calc) Tilman počítá 2 + 5
2 + 5 = 7
———————————
Tilman počítá 7.

Závěr je nesmyslný, neboť nelze počítat číslo, aniž by to číslo bylo nějak aritmeticky specifikováno. Důvodem, proč zde substituce selhává, je to, že je podstatný rozdíl v užití významu termu '2 + 5' v první a druhé premise. V první premise je tento význam pouze zmíněn, neboť Tilman není vztažen k výsledku, nýbrž k samotné proceduře aplikace funkce + na argumenty 2 a 5. Chce tuto proceduru provést a zjistit, co je jejím výsledkem. Tedy výskyt oné procedury je v první premise pouze zmíněn. Chceme-li v libovolném světě *w* a čase *t* vyhodnotit pravdivostní podmínky vyjádřené první premisou, stačí ověřit, zda Tilman tuto proceduru opravdu provádí, nemusíme proto sami počítat, kolik je 2+5. Na druhé straně, chceme-li vyhodnotit pravdivost druhé premisy, musíme provést proceduru aplikace funkce + na argumenty 2 a 5, a zjistit, zda je výsledek identický s číslem 7. Jednotlivé kroky analýzy premis P_1, P_2 jsou:

P_1: (a) $[^0{+}\ {}^02\ {}^05]$ $/*_1, \rightarrow \tau$
(b) $^0[^0{+}\ {}^02\ {}^05]$ $/*_2, \rightarrow *_1$ (kompozici $[^0{+}\ {}^02\ {}^05]$)
(c) $[^0Po\check{c}\acute{\imath}t\acute{a}_{wt}\ {}^0Tilman\ {}^0[^0{+}\ {}^02\ {}^05]]$ $/*_2, \rightarrow_v o$
(d) $\lambda w \lambda t\ [^0Po\check{c}\acute{\imath}t\acute{a}_{wt}\ {}^0Tilman\ {}^0[^0{+}\ {}^02\ {}^05]]$ $/*_2, \rightarrow o_{\tau\omega}$

P_2: (a) $[^0{+}\ {}^02\ {}^05]$ $/*_1, \rightarrow \tau$
(b) ^{07} $/*_1, \rightarrow \tau$
(c) $^0{=}$ $/*_1, \rightarrow (o\tau\tau)$
(d) $[^0{=}\ [^0{+}\ {}^02\ {}^05]\ {}^07]$ $/*_1, \rightarrow o$.

Typy: *Tilman*/ι; *Počítá*/$(o\iota *_1)_{\tau\omega}$; +/$(\tau\tau\tau)$; 2, 5, 7/$\tau$; =/$(o\tau\tau)$.

Je tedy zřejmé, že identita (d) neumožňuje substituovat ^{07} za $^0[^0{+}\ {}^02\ {}^05]]$. Došlo by k typové chybě. *Konstituenty* premisy P_1, tj. *části* procedury P_1, jsou:

1) $\lambda w \lambda t\ [{}^0Počítá_{wt}\ {}^0Tilman\ {}^0[{}^0{+}\ {}^02\ {}^05]]$
2) $[{}^0Počítá_{wt}\ {}^0Tilman\ {}^0[{}^0{+}\ {}^02\ {}^05]]$
3) ${}^0Počítá_{wt}$
4) $[{}^0Počítá\ w]$
5) ${}^0Počítá$
6) w
7) t
8) ${}^0Tilman$
9) ${}^0[{}^0{+}\ {}^02\ {}^05]$

Všimněme si ještě, že každá konstrukce je částí sebe sama, proto konstrukce ad 1) je konstituentem konstrukce $\lambda w \lambda t\ [{}^0Počítá_{wt}\ {}^0Tilman\ {}^0[{}^0{+}\ {}^02\ {}^05]]$. Ostatní konstituenty jsou *vlastní části* této konstrukce. Kompozice $[{}^0{+}\ {}^02\ {}^05]$ není částí této konstrukce, její výskyt je v módu zmiňování, stejně tak jako výskyt konstrukcí ${}^0{+}$, ^{02}, ^{05}.

Konstrukce jsou zmiňovány Trivializací. Jak je vidět na našich příkladech, všechny podkonstrukce takto zmíněného výskytu jsou rovněž v módu zmiňování. Říkáme také, že se vyskytují v hyperintensionálním kontextu. Mohlo by se tedy zdát, že k tomu, abychom určili, zda je daný výskyt podkonstrukce hyperintensionální, tj. v módu zmiňování, stačí určit, zda je to výskyt v dosahu Trivializace. Trivializace konstrukce prostě zvyšuje kontext na hyperintensionální úroveň. Je zde však ještě jedna malá komplikace. Dle Definice 1, vi) Dvojí Provedení ruší účinek Trivializace. Platí tedy tento axiom: ${}^{20}C = C$. Proto definujeme:

Definice 4 (*výskyt konstrukce v módu zmiňování a provedení*)
Nechť D je podkonstrukce konstrukce C. Pak

(i) Je-li D identická s C, pak *výskyt D* je v C v *módu provedení*.
(ii) Je-li C identická s $[X_1\ X_2 \ldots X_m]$ a D je jedna z konstrukcí $X_1, X_2, \ldots, X_m$, pak *výskyt D* je v C v *módu provedení*.
(iii) Je-li C identická s $[\lambda x_1 \ldots x_m\ X]$ a D je X, pak *výskyt D* je v C v *módu provedení*.
(iv) Je-li C identická s ${}^1X$ a D je X, pak *výskyt D* je v C v *módu provedení*.
(v) Je-li C identická s ${}^2X$ a D je X, nebo výskyt ${}^0D$ je v X v módu provedení, pak *výskyt D* je v C v *módu provedení*.
(vi) Je-li výskyt D v módu provedení v C' a tento výskyt C' je v módu provedení v C, pak *výskyt D* je v C v *módu provedení*.

(vii) Pokud výskyt *D* v *C* není v módu provedení, pak tento *výskyt D* je v módu *zmiňování* v *C*.

(viii) Pouze výskyty dle i)-vii) jsou v módu provedení nebo zmiňování. □

Definice 5 (*Konstituent neboli část konstrukce*)
Nechť *D* je podkonstrukce konstrukce *C*. Pak každý výskyt *D*, který je v *C* v módu provedení je *konstituentem* neboli *částí C*.

Důsledek: Vlastní části konstrukce *C* jsou výskyty *D* dle bodů ii), iii), iv) a v), Def. 4. Tedy vlastní části konstrukce *C* nejsou identické s *C*. Výskyt *D* dle bodu i) je nevlastní částí *C*, je to konstrukce identická s *C*.

Definice 6 (*atomické a molekulární konstrukce*)
Konstrukce je *atomická*, pokud neobsahuje žádný jiný konstituent než sebe sama. Jinak je daná konstrukce *molekulární*.

Důsledek. Konstrukce *C* je atomická, pokud *C* je

- Proměnná, nebo
- Trivializace ^0X, kde *X* je objekt jakéhokoli typu včetně typu konstrukce
- Jednoduché Provedení ^1X, kde *X* je neprocedurální objekt typu řádu 1, tj. *X* není konstrukce
- Dvojí Provedení ^2X, kde *X* je objekt typu řádu 1, tj. *X* není konstrukce.□

To, že atomická konstrukce je strukturovaný celek je triviálně pravda, neboť atomická konstrukce je částí sebe sama (Def. 4, i). Otázkou nyní je, co vytváří z *vlastních* částí molekulární konstrukce jeden celek. Tímto sjednocujícím faktorem molekulární konstrukce je právě to, že konstrukce je procedura. Každá konstrukce, ať již atomická nebo molekulární je celek, který může být (alespoň v principu) jakožto procedura proveden. Tedy konstrukce (a obecně procedury) nejsou pouhé agregáty nebo souhrny svých částí, jako je tomu u množin. Množina nemůže být provedena za účelem získání nějakého výstupu, a její prvky spolu nijak neinteragují. A jak jsme uvedli v úvodu, právě ono „*direct connection inter se*“ je to, co charakterizuje části strukturovaného celku.

Zbývá tedy ukázat, jakým způsobem spolu interagují vlastní části molekulární konstrukce. To je však jednoduché a zřejmé. V procesu skládání funkcí (konstrukce Kompozice) nebo duálně v procesu deklarace funkcí (konstrukce Uzávěr) probíhá interakce tak, že výstup (produkt) jednoho konstituentu *C* dané molekulární konstrukce se stává vstupem jiného konstituentu *D* této konstrukce čili argumentem funkce konstruované konstituentem *D*. Pokud některý z konstituentů selhává v produkování příslušného objektu (konstituent je *v*-nevlastní), pak celá konstrukce selhává v produkování objektu, je *v*-nevlastní, neboť tímto je proces interakce mezi částmi přerušen a nemůže být dokončen.

Jednoduchý příklad.[7] Kompozice $[^0{+}\ ^02\ ^05]$ je procedura, která produkuje číslo 7. Její části spolu interagují v tomto procesu takto. Konstituent $^0{+}$ jakožto procedura volá dva další konstituenty ^{02} a ^{05}, neboť potřebuje jejich výstup čili čísla 2 a 5, které slouží jako argumenty funkce + produkované procedurou $^0{+}$. Tímto způsobem celá Kompozice jakožto procedura spojuje své vlastní části $^0{+}$, ^{02}, ^{05} v jeden celek, který produkuje hodnotu funkce + na argumentu (2,5), což je právě číslo 7. Podobně Uzávěr $\lambda x\ [^0{+}\ x\ ^01]$, produkuje při svém provedení funkci následníka, a to takto. Je to procedura s formálním parametrem *x*. Kdykoli je valuací proměnné *x* dodán skutečný argument *n*, je tento argumentem *n* dosazen v těle procedury za formální parametr *x*, toto tělo je provedeno jakožto procedura aplikace funkce + (produkt konstituentu $^0{+}$) na argumenty *n* a 1, tedy výsledkem je číslo n+1. Jelikož takto je Uzávěr prováděn pro *libovolnou* valuaci proměnné *x*, abstrakcí od jejích hodnot (λx) je produkováno zobrazení, které libovolnému číslu přiřadí jeho následníka.

Tvrzení 1

Relace být částí (konstituentem) dané konstrukce je částečné uspořádání.

Důkaz.

a) *Reflexivita* plyne z Def. 4, i).
b) *Transitivita* plyne z Def. 4, vi).
c) *Antisymetrie*. Předpokládejme, že C_1 je částí C_2 a C_2 je částí C_1, a že C_1 není identická s C_2. Pak ale nelze aplikovat bod i) Def. 4, tedy C_1 je *vlastní* částí C_2 a C_2 je *vlastní* částí C_1. To však dle Def. 4 není možné. Tedy C_1 je identická s C_2.

Nyní můžeme porovnat mereologickou strukturu procedur / konstrukcí s klasickou mereologií. Cotnoir (2013) rekapituluje tři hlavní principy klasické mereologie (Classical Extensional Mereology, CEM) takto:

[7] Pro názornost popíšu tuto interakci v žargonu programovacích jazyků.

Extenzionalita. Jestliže x a y mají stejnou mereologickou strukturu („make-up“), pak x a y jsou identické.
Antisymetrie. Je-li x částí y a y částí x, pak x a y jsou identické.
Idempotence. Jestliže x je vlastní částí y, pak suma x a y je identická s y.

Princip antisymetrie je obsahem tvrzení 1. Budeme se tedy nadále zabývat problémem extenzionality a idempotence.

3.2.1. Extenzionalita

Extenzionalita, tj. princip „stejné části = stejný celek“, je v klasické mereologii předmětem mnoha diskusí, a je ve všeobecnosti považována za největší problém CEM. Cotnoir se k tomuto problému vyjadřuje takto:

> The classic counterexample to extensionality involves objects (e.g., a statue) and the matter which constitutes them (e.g., a lump of clay). They presumably have different properties: e.g., the clay can survive squashing whereas the statue cannot. They must, therefore, be different objects. Yet every part of one appears to be part of the other. Their structure (insofar as mereology is concerned, anyway) is exactly the same. Another example involves the construction of two objects by a rearrangement of the same parts. Suppose my son builds a house out of some lego bricks. He then destroys the house (as he often does) and proceeds to build a boat from the same lego bricks. Is the house identical to the boat? Or are they distinct? Extensionality would seem to force us to identify the two. (Cotnoir 2013, 835)

Naše odpověď je, samozřejmě, že jde o různé objekty, protože jsou *konstruovány* sice ze stejných vlastních částí, ale různým *způsobem*. Avšak tyto objekty jako Lego domeček či loďka, nejsou strukturované, jsou jednoduché. Tichý k tomu říká:

> [A] *car* is a simple entity. But is this not a *reductio ad absurdum*? Are cars not complex, as anyone who has tried to fix one will readily testify?
>
> No, they are not. If a car were a complex, then it would be legitimate to ask: Exactly how complex is it? Now how many parts does a car consist of? One plausible answer which may suggest itself is that it has three parts: an engine, a chassis, and a body. But an equally plausible answer can be given in terms of a much longer list: several spark plugs, several pistons, a starter, a carburettor, four tyres, two axles, six windows, etc. Despite being longer the latter list does not overlap with the former: neither the engine, nor

> the chassis nor the body appears on it. How can that be? How can an engine, for example, both be and not be a part of one and the very same car?
>
> There is no mystery, however. It is a commonplace that a car can be *decomposed* in several alternative ways. … Put in other words, a car can be *constructed* in a very simple way as a mereological sum of three things, or in a more elaborate way as a mereological sum of a much larger set of things. (Tichý 1995, 179-180)

Pro naši mereologii *strukturovaných* procedur, princip extenzionality platí triviálně, *pokud* jej aplikujeme na vlastní i *nevlastní* části. Princip „stejné části (i nevlastní) = stejný celek" platí pro procedury triviálně, neboť je ekvivalentní reflexivitě. Formulujme tedy silnější princip extenzionality:

> *Extensionalita vlastních částí*. Jestliže *x* a *y* jsou objekty složené ze stejných *vlastních* částí, pak *x* a *y* jsou identické.

V tomto případě princip extenzionality pro procedury *neplatí*, což je však naprosto v pořádku. Vždyť jistě umělecká socha a hrouda hlíny jsou naprosto rozdílné objekty. Jak jsem uvedla výše, pro specifikaci procedury nestačí uvést výčet jejích vlastních částí, protože identita procedury se odvíjí od *způsobu spojení* těchto částí do jednoho celku, tj. do celé procedury. Připomeňme si znovu Bolzanův příklad dvou rozdílných pojmů se stejnými (vlastními) částmi, a to 3^5 vs. 5^3. Jistěže to jsou rozdílné procedury, ačkoliv mají stejné vlastní části, neboť procedura umocnění čísla tři na pátou je rozdílná od procedury umocnění čísla pět na třetí:

$$^0[^0Mocnina\ ^03\ ^05] \neq {}^0[^0Mocnina\ ^05\ ^03]$$

Typy: $Mocnina/(\tau\tau\tau)$; $3,5/\tau$; $[^0Mocnina\ ^03\ ^05]$, $[^0Mocnina\ ^05\ ^03]/*_1 \rightarrow \tau$; $^0[^0Mocnina\ ^03\ ^05]/*_2 \rightarrow *_1$; $^0[^0Mocnina\ ^05\ ^03]/*_2 \rightarrow *_1$; $\neq/(o*_1*_1)$: relace *ne*-identity mezi konstrukcemi řádu 1.

Dokonce i v případě, kdy procedury C_1 a C_2 produkují jeden a tentýž objekt a mají stejné vlastní části, mohou to být rozdílné procedury. Například dle Def. 1 *nejsou* Kompozice $[^0+\ ^02\ ^05]$ a $[^0+\ ^05\ ^02]$ identické, ačkoliv mají stejné vlastní části a konstruují stejný objekt (číslo 7):

$$[^0+\ ^02\ ^05] = [^0+\ ^05\ ^02]$$

ale

$^0[^0{+}\ ^02\ ^05] \neq {}^0[^0{+}\ ^05\ ^02]$

Typy: +/(τττ); 2,5/τ; $[^0{+}\ ^02\ ^05]$, $[^0{+}\ ^05\ ^02]/*_1 \rightarrow \tau$; =/(οττ); $^0[^0{+}\ ^02\ ^05]$, $^0[^0{+}\ ^05\ ^02]/*_2 \rightarrow *_1$; $\neq/(o*_1*_1)$.

Opět však je tomu tak správně, má to takto být, neboť pouhé porozumění výrazům '2+5' a '5+2' nestačí k tomu, abychom určili, že jsou ekvivalentní (tj. označují stejný objekt). Musíme navíc *dokázat*, že funkce sčítání je komutativní.

3.2.2. Idempotence

Princip idempotence je formulován poněkud nejasně. Jaký je význam výrazu „suma" v principu „jestliže *x* je vlastní částí *y*, pak suma *x* a *y* je identická s *y*"? Můžeme jej přeformulovat takto: Není možné přidat k objektu *Y* jeho další vlastní část *X*, neboť takovéto přidání bude pohlceno již existujícím výskytem vlastní části *X*. Takovýto princip se zdá být platný. Musíme však rozlišit mezi idempotencí ve smyslu mereologickém a ve smyslu stejných pravdivostních podmínek, což v TIL odpovídá rozdílu mezi *intensionální* a *hyperintensionální* úrovní. Ukážeme si to na příkladu.

Nechť A, $B \rightarrow_v$ ο jsou libovolné konstrukce pravdivostní hodnoty. Pak na intensionální úrovni idempotence platí: $A = [^0\wedge\ A\ A]$. Uvažme dále konstrukci

$Y = [^0\supset [A\ B]]$

a nechť $X = A$. Přidejme X a $^0\wedge$ ke konstrukci Y. Obdržíme

$Y^+ = [^0\supset [^0\wedge\ A\ A]\ B]$.

Avšak konstrukce Y je *odlišná* od konstrukce Y^+. Pro libovolnou konstrukci A je Kompozice $[^0\wedge\ A\ A]$ rozdílná od A.

Mohli bychom namítnout, že v tomto případě jsme přidali *dvě* různé části ke konstrukci Y, a to konstrukci X a Trivializaci $^0\wedge$, a tedy je samozřejmé, že konstrukce Y^+ se musí lišit od konstrukce Y. Problém však spočívá v tom, jak jinak bychom mohli přidat k dané proceduře další výskyt procedurální vlastní části, bez toho, že bychom přidali rovněž nějakou operaci, která to umožní (v tomto případě spojka konjunkce).

Jak jsem uvedla v úvodu tohoto článku, zatímco klasická mereologie se zabývá vztahem celek-část mezi fyzickými individui, tj. zcela náhodným vztahem, do kterého mohou vstupovat kterákoli individua a vytvářet tak nesmyslné objekty jako „warrots“, tj. individua skládající se z velryby (*whale*) a mrkve (*carrot*), mereologie strukturovaných procedur je v tomto smyslu velmi *restriktivní.* V TIL je typově jasně vymezeno, jaké typy konstrukcí a jaké typy objektů mohou být smysluplně spojeny jakožto části jednoho celku, tj. strukturované procedury, kterou lze provádět. Tedy idempotence *neplatí* na úrovni hyperintensionální *pro strukturované procedury*.

4. Závěr

V tomto článku jsem se zabývala mereologickou strukturou procedur, kterou jsem explikovala v rámci Tichého systému Transparentní intensionální logiky, TIL. Ukázala jsem, že TIL konstrukce nejsou jednoduché objekty, nýbrž strukturované celky, skládající se z jednoznačně určených částí, tj. konstituentů, které je nutno provést, chceme-li provést celek jakožto proceduru produkující na výstupu nanejvýš jeden objekt. Mereologie takovýchto strukturovaných celků má vlastnosti částečného uspořádání, je založena na rigorosních a jasných principech, je však neklasická v tom smyslu, že pro ni neplatí princip extenzionality a idempotence. Zároveň jsem však ukázala, že právě tyto neklasické rysy jsou pro strukturu procedur správné a žádoucí.

Poděkování

Tato práce byla podporována grantovou agenturou České republiky v rámci projektu GA15-13277S, „Hyperintensionální logika pro analýzu přirozeného jazyka“ a rovněž interní agenturou VŠB-TU Ostrava v rámci projektu SGS No. SP2016/100.

Děkuji P. Maternovi za cenné poznámky k práci Bolzanově a B. Jespersenovi za cenné připomínky k idempotenci.

Literatura

BAR-HILLEL, Y. (1950): Bolzano’s Definition of Analytic Propositions. *Methodos II*, No. 5, 342-355. (Republished in *Theoria* 16, 1950, 91-117.)

BARENDREGT, H. P. (1997): The Impact of the Lambda Calculus. *Bulletin of Symbolic Logic* 3, 181-215.

BOLZANO, B. (1837): *Wissenschaftslehre*. Sulzbach: von Seidel.

CARNAP, R. (1947): *Meaning and Necessity*. Chicago: Chicago University Press.

CMOREJ, P. – TICHÝ, P. (1998a): Complexes I. *Organon F* 5, č. 2, 139-161.

CMOREJ, P. – TICHÝ, P. (1998b): Complexes II. *Organon F* 5, č. 3, 266-289.

COTNOIR, A. J. (2013): Strange Parts: The Metaphysics of Non-Classical Mereologies. *Philosophy Compass* 8/9, 834-845.

DUŽÍ, M. (2007): Properties on the Edge. In: Zouhar, M. (ed.): *Svet jazyka a svet za jazykom*. Bratislava: Filozofický ústav SAV, 42-68.

DUŽÍ, M. – JESPERSEN, B. – MATERNA, P. (2010): *Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Trasnsparent Intensional Logic*. Berlin: Springer.

DUŽÍ, M. – MATERNA, P. (2012): *TIL jako procedurální logika (průvodce zvídavého čtenáře Transparentní intensionální logikou)*. Bratislava: Aleph.

GANTER, B. – WILLE, R. (1999): *Formal Concept Analysis. Mathematical Foundations*. Springer.

LUDWIG, K. – RAY, G. (1998): Semantics for Opaque Contexts. *Philosophical Perspectives* 12, 141-166.

MATERNA, P. (1998): *Concepts and Objects*. Helsinki: Acta Philosophica Fennica, vol. 63.

MATERNA, P. (2004): *Conceptual Systems*. Berlin: Logos.

RACLAVSKÝ, J. – KUCHYŇKA, P. – PEZLAR, I. (2015): *Transparentní intenzionální logika jako charakteristika universalis a calculus ratiocinator*. Brno: Masarykova univerzita.

RUSSELL, B. (1903/1996): *The Principles of Mathematics.* New York – London: W. W. Norton & Company.

TICHÝ, P. (1988): *The Foundations of Frege's Logic*. Berlin – New York: De Gruyter.

TICHÝ, P. (1995): Constructions as the Subject-Matter of Mathematics. In: DePauli-Schimanovich, W. – Köhler, E. – Stadler, F. (eds.): *The Foundational Debate: Complexity and Constructivity in Mathematics and Physics.* Dordrecht – Boston – London – Vienna: Kluwer, 175-185. Reprinted in Tichý (2004, 873-885).

TICHÝ, P. (2004): *Collected Papers in Logic and Philosophy*. Svoboda, V. – Jespersen, B. – Cheyne, C. (eds.), Prague: Filosofia, Czech Academy of Sciences – Dunedin: University of Otago Press.