The Fundamentals of Lative Logic

P. Eklund, U. Höhle, J. Kortelainen

LINZ 2014, Austria
February 18–22, 2014

‘Lative’ is “motion”, motion ‘to’ and ‘from’, so when terms appear in sentences, terms ‘move into’ sentence, and sentences ‘move away from’ terms. In comparison, ‘ablative’ is “motion away”, and nominative is static. The lative locative case (casus) indeed represents “motion”, whereas e.g. a vocative case is identification with address.

- “Lative logic” is more about “lativity” between various components and building blocks of a logic as a categorical object, rather than traditionally creating “yet another logic”.
- It is also distinct from the “fons et origo” foundational logic, where the roles of metalanguage and object language may be blurred.
- This approach to logic assumes category theory as its metalanguage, and leans on having signatures as a pillar and starting point for “terms”, which in turn are needed in “sentences”, and so on.

- A negation operator $\neg$ can be applied to the term $P(x)$, which indeed is constructed by the operator P, so that $\neg P(x)$ and $P(x)$ are of the same sort, as terms.
- However, as $\exists x.P(x)$ is not a term, but is expected to be a sentence, and it is very questionable whether $\neg$ in $\neg\exists x.P(x)$ and $\exists x.\neg P(x)$ really is the same symbol.
- In $\exists x.\neg P(x)$, it acts an operator, changing a term to term, but in $\neg\exists x.P(x)$ it changes a sentence to a sentence, so it is strictly speaking not an 'operator'.
- Variables may be substituted by terms, but 'sentential' variables make no sense with respect to substitution.

- Assigning uncertainty is far from trivial, and the place where uncertainty should be invoked is also not always clear.
- *Logic*, as a structure, contains *signatures, terms, sentences, theoremata (as structured sets of sentences, or 'structured premises'), entailments, algebras, satisfactions, axioms, theories and proof calculi*.
- It may then be reasonable to assume that *Fuzzy Logic*, again as a structure, contains *fuzzy signatures, fuzzy terms, fuzzy sentences, fuzzy theoremata, fuzzy entailments, fuzzy algebras, fuzzy satisfactions, fuzzy axioms, fuzzy theories and fuzzy proof calculi*, i.e. 'fuzzy' distributes over the operator that glues substructures in logic into a whole.
- This is then the foundational background also for *Fuzzy Logic Programming*.

- We present results on adapting a strictly categorical framework, as a chosen metalanguage, enables us to be very precise about the distinction between terms and sentences, where ‘boolean’ operator symbols, i.e. where the codomain sort of the operator is a ‘boolean’ sort, become part of the underlying signature.
- Implication is not introduced as an operator in the signature, nor as a short name using existing operators, but will appear as integrated into our sentence functors.
- We produce a sentence as a pair $(P(x), Q(y))$ of terms, where they are produced by its own term functors.
- Intuitively, this corresponds to “$P(x)$ is inferred by $Q(y)$”.
- The ‘pairing operation’, i.e., the ‘implication’, is not given in the underlying signature as an operator, but appears as the result of functor composition and product within a ‘sentence constructor’.

Signatures

- The previous talk was using a strictly mathematical, and a 'monoidal biclosed categorical' notation for signatures. Here we adopt the more 'computationally intuitive' notation of a signature, but the content and concept is the same as for the strict one.
- A many-sorted signature $\Sigma = (S, \Omega)$ consists of a set S of sorts (or types), and a tupled set $\Omega = (\Omega_{\mathtt{s}})_{\mathtt{s} \in S}$ of operators. Operators in $\Omega_{\mathtt{s}}$ are written as $\omega : \mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}$.

Signatures over underlying categories

- We indeed restrict to quantales $\mathfrak{Q}$ that are commutative and unital, as this makes the Goguen category $\texttt{Set}(\mathfrak{Q})$ to be a symmetric monoidal closed category and therefore also biclosed.
- This Goguen category carries all structure needed for modelling uncertainty using underlying categories for fuzzy terms over appropriate signatures.
- A signature $(S, (\Omega, \alpha))$ over $\texttt{Set}(\mathfrak{Q})$ then typically has S as a crisp set, and $\alpha : \Omega \longrightarrow Q$ then assigns uncertain values to operators.

Highlights of the term construction

We use the notation

$$\Omega^{\mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}}$$

for the set of operators $\omega : \mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}$ (in $\Omega_{\mathtt{s}}$) and

$$\Omega^{\to \mathtt{s}}$$

for the set of constants $\omega : \to \mathtt{s}$ (also in $\Omega_{\mathtt{s}}$), so that we may write

$$\Omega_{\mathtt{s}} = \coprod_{\substack{\mathtt{s}_1,\ldots,\mathtt{s}_n \\ n \leq k}} \Omega^{\mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}}.$$

For the term functor construction over $\mathtt{Set}(\mathfrak{Q})$ we need objects

$$(\Omega^{\mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}}, \alpha^{\mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}})$$

for the operators $\omega : \mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}$, and

$$(\Omega^{\to \mathtt{s}}, \alpha^{\to \mathtt{s}})$$

for the constants $\omega :\to \mathtt{s}$.

The term functor construction over $\mathtt{Set}$

$$\Psi_{\mathtt{m},\mathtt{s}}((X_{\mathtt{t}})_{\mathtt{t}\in S}) = \Omega^{\mathtt{s}_1\times\ldots\times\mathtt{s}_n\to\mathtt{s}} \otimes \bigotimes_{i=1,\ldots,n} X_{\mathtt{s}_i},$$

changes over $\mathtt{Set}(\mathfrak{Q})$ to

$$\Psi_{\mathtt{m},\mathtt{s}}(((X_{\mathtt{t}},\delta_{\mathtt{t}}))_{\mathtt{t}\in S}) = (\Omega^{\mathtt{s}_1\times\ldots\times\mathtt{s}_n\to\mathtt{s}},\ \alpha^{\mathtt{s}_1\times\ldots\times\mathtt{s}_n\to\mathtt{s}}) \otimes \bigotimes_{i=1,\ldots,n} (X_{\mathtt{s}_i},\delta_{\mathtt{s}_i})$$

$$= (\Omega^{\mathtt{s}_1\times\ldots\times\mathtt{s}_n\to\mathtt{s}} \times \prod_{i=1,\ldots,n} X_{\mathtt{s}_i},\ \alpha^{\mathtt{s}_1\times\ldots\times\mathtt{s}_n\to\mathtt{s}} \odot \bigodot_{i=1,\ldots,n} \delta_{\mathtt{s}_i}).$$

The inductive steps in the construction:

- $\mathsf{T}^1_{\Sigma,\mathtt{s}} = \coprod_{\mathtt{m}\in\hat{\mathtt{S}}} \Psi_{\mathtt{m},\mathtt{s}}$
- $\mathsf{T}^{\iota}_{\Sigma,\mathtt{s}} X_{\mathtt{S}} = \coprod_{\mathtt{m}\in\hat{\mathtt{S}}} \Psi_{\mathtt{m},\mathtt{s}}(\mathsf{T}^{\iota-1}_{\Sigma,\mathtt{t}} X_{\mathtt{S}} \sqcup X_{\mathtt{t}})_{\mathtt{t}\in\mathtt{S}})$, for $\iota > 1$

We have $\mathsf{T}^{\iota}_{\Sigma} X_{\mathtt{S}} = (\mathsf{T}^{\iota}_{\Sigma,\mathtt{s}} X_{\mathtt{S}})_{\mathtt{s}\in\mathtt{S}}$. Further, $(\mathsf{T}^{\iota}_{\Sigma})_{\iota>0}$ is an inductive system of endofunctors, and the inductive limit $\mathsf{F} = \mathrm{ind} \varinjlim \mathsf{T}^{\iota}_{\Sigma}$ exists.

The final term functor:

- $\mathsf{T}_{\Sigma} = \mathsf{F} \sqcup \mathrm{id}_{\mathtt{Set}_{\mathtt{S}}}$

We also have $\mathsf{T}_{\Sigma} X_{\mathtt{S}} = (\mathsf{T}_{\Sigma,\mathtt{s}} X_{\mathtt{S}})_{\mathtt{s}\in\mathtt{S}}$.

Terms and ground terms

In order to proceed towards creating sentences, we need the so called ‘ground terms’ produced by the term monad.

- $\Sigma_0 = (S_0, \Omega_0)$ over $\mathtt{Set}$
- $\mathbf{T}_{\Sigma_0}$ term monad over $\mathtt{Set}_{S_0}$
- $\mathsf{T}_{\Sigma_0}\varnothing_{S_0}$ is the set of ‘ground terms’

'Predicate' symbols as operators in a signature

- We now proceed to clearly separate views of terms and sentences, respectively, in propositional logic and predicate logic.
- In order to introduce 'predicate' symbols as operators in a specific signature, we assume that Σ contains a sort $\texttt{bool}$, which does not appear in connection with any operator in Ω_0, i.e., we set $S = S_0 \cup \{\texttt{bool}\}$, $\texttt{bool} \notin S_0$, and $\Omega = \Omega_0$.
- This means that $\mathrm{T}_{\Sigma,\texttt{bool}} X_S = X_{\texttt{bool}}$, and for any substitution $\sigma_S : X_S \longrightarrow \mathrm{T}_\Sigma X_S$, we have $\sigma_{\texttt{bool}}(x) = x$ for all $x \in X_{\texttt{bool}}$.
- $\texttt{bool}$ is kind of the "predicates as terms" sort.

Propositional logic

Signature:

- Let $\Sigma_{PL} = (S_{PL}, \Omega_{PL})$, where $S_{PL} = S$ and $\Omega_{PL} = \{\mathtt{F}, \mathtt{T} :\to \mathtt{bool}, \& : \mathtt{bool} \times \mathtt{bool} \to \mathtt{bool}, \neg : \mathtt{bool} \to \mathtt{bool}\} \cup \{\mathtt{P}_i : \mathtt{s}_{i_1} \times \cdots \times \mathtt{s}_{i_n} \to \mathtt{bool} \mid i \in I, \mathtt{s}_{i_j} \in S\}$.
- Similarly as `bool` leading to no additional terms, except for additional variables being terms when using Σ, the sorts in S_{PL}, other than `bool`, will lead to no additional terms except variables.
- Adding ‘predicates’ as operators even if they produce no terms seems superfluous at first sight, but the justification is seen when we compose these term functors with T_Σ.

- For the sentence functor, we need the 'tuple selecting' functor $\arg^{\mathtt{s}} : \mathtt{C}_S \longrightarrow \mathtt{C}$ such that $\arg^{\mathtt{s}} X_S = X_{\mathtt{s}}$ and $\arg^{\mathtt{s}} f_S = f_{\mathtt{s}}$.
- We also need the 'variables ignoring' functor $\phi^{\mathtt{s}} : \mathtt{Set}_S \longrightarrow \mathtt{Set}_S$ such that $\phi^{\mathtt{s}} X_S = X'_S$, where for all $\mathtt{t} \in S \backslash \{\mathtt{s}\}$ we have $X'_{\mathtt{t}} = \varnothing$, and $X'_{\mathtt{s}} = X_{\mathtt{s}}$. Actions on morphisms are defined in the obvious way.

Propositional logic 'formulas' as sentences:

- $\mathrm{Sen}_{PL} = \arg^{\mathtt{bool}} \circ \mathrm{T}_{\Sigma_{PL}} \circ \phi^{\mathtt{bool}}$

Notational flexibility and selectivity ...

- $\Sigma_{PL\backslash\neg}$ is the signature where the operator $\neg$ is removed, and $\Sigma_{PL\backslash\neg,\&}$ where both $\neg$ and & are removed
- $\bigcup_{\mathtt{s}\in S}(\mathrm{T}_{\Sigma,\mathtt{s}} \circ \phi^{S\backslash\mathtt{bool}})\varnothing_S$ is the set of all 'non-boolean' sorted terms, i.e., the "unsorted set" of all "ground terms", and corresponds to the so called the "Herbrand universe"
- $\bigcup_{\mathtt{s}\in S}(\mathrm{T}_{\Sigma,\mathtt{s}} \circ \phi^{S\backslash\mathtt{bool}})X_S$ is syntactically the set of all 'non-boolean' sorted terms, i.e., the "unsorted set" of all terms, and corresponds semantically to the "Herbrand interpretation"
- note also how $(\mathrm{arg}^{\mathtt{bool}} \circ \mathrm{T}_{\Sigma_{PL\backslash\neg,\&}} \circ \phi^{\mathtt{bool}})X_S = \{\mathtt{F}, \mathtt{T}\}$

The sentence functor for Horn clause logic (HCL)

$$\begin{aligned}\mathsf{Sen}_{HCL} &= (\mathsf{arg}^{\texttt{bool}})^2 \circ (((\mathsf{T}_{\Sigma_{PL\setminus\neg,\&}} \circ \mathsf{T}_\Sigma) \times (\mathsf{T}_{\Sigma_{PL\setminus\neg}} \circ \mathsf{T}_\Sigma)) \circ \phi^{S\setminus\texttt{bool}}) \\ &= (\mathsf{arg}^{\texttt{bool}})^2 \circ ((\mathsf{T}_{\Sigma_{PL\setminus\neg,\&}} \times \mathsf{T}_{\Sigma_{PL\setminus\neg}}) \circ \mathsf{T}_\Sigma \circ \phi^{S\setminus\texttt{bool}})\end{aligned}$$

- the pair $(h, b) \in \mathsf{Sen}_{HCL} X_S$, as a sentence representing the ‘Horn clause’, means that h is an ‘atom’ and b is a conjunction of ‘atoms’
- $(h, \texttt{T})$ is a ‘fact’
- $(\texttt{F}, b)$ is a ‘goal clause’
- $(\texttt{F}, \texttt{T})$ is a ‘failure’

Modus Ponens as an inference rule then looks more like ...

$$\frac{(\mathtt{F}, b) \quad (h, b)}{(h, \mathtt{T})}$$

This is correctly written since we use sentences only, i.e., not mixing terms and sentences in proof rules, but it is still informal since an inference rule involves ‘theoremata’.

Anticipating the notion of ‘theoremata’ as a structured set of sentences, the following proof rule involves ‘one-sentence theoremata’ in the special case of having the theoremata functor being the powerset functor composed with the sentence functor.

$$\frac{\{(\mathtt{F}, b)\} \ddagger \{(h, b)\}}{\{(h, \mathtt{T})\}}$$

Variable substitutions within sentences

- $\sigma_S : \phi^{S\backslash \mathtt{bool}} X_S \longrightarrow \mathsf{T}_\Sigma \phi^{S\backslash \mathtt{bool}} Y_S$
- $\mu \circ \mathsf{T}_\Sigma \sigma_S : \mathsf{T}_\Sigma \phi^{S\backslash \mathtt{bool}} X_S \longrightarrow \mathsf{T}_\Sigma \phi^{S\backslash \mathtt{bool}} Y_S$

$$\sigma_S^{head} = \mathsf{T}_{\Sigma_{PL\backslash\neg,\&}}(\mu \circ \mathsf{T}_\Sigma \sigma_S) : (\mathsf{T}_{\Sigma_{PL\backslash\neg,\&}} \circ \mathsf{T}_\Sigma)\phi^{S\backslash \mathtt{bool}} X_S \longrightarrow (\mathsf{T}_{\Sigma_{PL\backslash\neg,\&}} \circ \mathsf{T}_\Sigma)\phi^{S\backslash \mathtt{bool}} Y_S$$

$$\sigma_S^{body} = \mathsf{T}_{\Sigma_{PL\backslash\neg}}(\mu \circ \mathsf{T}_\Sigma \sigma_S) : (\mathsf{T}_{\Sigma_{PL\backslash\neg}} \circ \mathsf{T}_\Sigma)\phi^{S\backslash \mathtt{bool}} X_S \longrightarrow (\mathsf{T}_{\Sigma_{PL\backslash\neg}} \circ \mathsf{T}_\Sigma)\phi^{S\backslash \mathtt{bool}} Y_S$$

$$(\sigma_S^{head}, \sigma_S^{body}) = (\mathsf{T}_{\Sigma_{PL\setminus\neg,\&}} \times \mathsf{T}_{\Sigma_{PL\setminus\neg}})(\mu \circ \mathsf{T}_\Sigma \sigma_S) : \\ ((\mathsf{T}_{\Sigma_{PL\setminus\neg,\&}} \times \mathsf{T}_{\Sigma_{PL\setminus\neg}}) \circ \mathsf{T}_\Sigma)\phi^{S\setminus\mathtt{bool}} X_S \longrightarrow \\ ((\mathsf{T}_{\Sigma_{PL\setminus\neg,\&}} \times \mathsf{T}_{\Sigma_{PL\setminus\neg}}) \circ \mathsf{T}_\Sigma)\phi^{S\setminus\mathtt{bool}} Y_S$$

$$\sigma^{HC} = (\sigma_{\mathtt{bool}}^{head}, \sigma_{\mathtt{bool}}^{body}) : \mathsf{Sen}_{HCL} X_S \longrightarrow \mathsf{Sen}_{HCL} Y_S$$

Extending Goguen's and Meseguer's frameworks for institutions and entailment systems

- The term monad can be abstracted by $\Theta\colon \mathtt{Sign} \longrightarrow \mathtt{Mnd}[\mathtt{C}]$ with $\mathtt{Mnd}[\mathtt{C}]$ being the category of monads over $\mathtt{C}$ of 'variable objects'.
- Clearly, a special case is $\Theta(\Sigma) = \mathbf{T}_\Sigma$.

- The Sen functor is abstracted as

$$\mathsf{Sen}\colon \mathtt{Mnd}[\mathtt{C}] \longrightarrow [\mathtt{C}, \mathtt{D}],$$

where $\mathtt{D}$ is monoidal biclosed and $[\mathtt{C}, \mathtt{D}]$ is the functor category, that is, for any monad $\mathbf{F} \in \mathsf{Ob}(\mathtt{Mnd}[\mathtt{C}])$ we have a functor

$$\mathsf{Sen}(\mathbf{F})\colon \mathtt{C} \longrightarrow \mathtt{D}$$

taking some object of variables to sentences over that object.
- Sen_{HCL} is of the form $\mathsf{Sen}(\mathbf{T}_\Sigma)\colon \mathtt{Set}_S \longrightarrow \mathtt{Set}$, where $\Sigma = (S, \Omega)$.
- $\mathsf{Sen}_{HCL}(\mathfrak{Q})$ of the form $\mathsf{Sen}(\mathbf{T}_\Sigma)\colon \mathtt{Set}(\mathfrak{Q})_S \longrightarrow \mathtt{Set}(\mathfrak{Q})$ can be constructed.

- $\mathsf{Sen}(\Theta(\Sigma))\colon \mathtt{C} \longrightarrow \mathtt{D}$
- $\mathsf{Sen}(\mathbf{T}_\Sigma)\colon \mathtt{Set}(\mathfrak{Q})_S \longrightarrow \mathtt{Set}(\mathfrak{Q})$
- Note how the signature is underlying everything, and once the term functor has been abstracted, substitution is really the “fuel” of logic inference.
- Generalized proof calculus can now be done without explicitly saying what the terms are!
- Soundness and completeness, conceptully generalized, can potentially be analysed in a very general sense, and generalized substitution (for terms, not sentences!) is a key issue in this general framework of *Lative Logic*.

A generalized entailment system, $\mathscr{E}$, is a structure $\mathscr{E} = (\mathtt{Sign}, \mathrm{Sen}, \mathbf{\Phi}, L, \vdash)$ where

- $\mathtt{Sign}$ is a category of signatures;
- Sen is the ‘sentence functor’;
- $\mathbf{\Phi} = (\Phi, \eta)$ is a premonad over $\mathtt{C}$ with an object of $\Phi\mathrm{Sen}(\Sigma)$ being called a *theoremata*;
- L is a completely distributive lattice; and
- $\vdash$ is a family of L-valued relations consisting of

$$\vdash_\Sigma : \Phi\mathrm{Sen}(\Sigma) \times \Phi\mathrm{Sen}(\Sigma) \longrightarrow L$$

for each signature $\Sigma \in \mathrm{Ob}(\mathtt{Sign})$ where $\vdash_\Sigma$ is called a Σ*-entailment*.

These are subject to the condition that, for $\Gamma_1, \Gamma_2, \Gamma_3 \in \Phi\mathrm{Sen}(\Sigma)$ (over $\mathtt{Set}$), each $\vdash_\Sigma$

- is reflexive, that is, $(\Gamma_1 \vdash_\Sigma \Gamma_1) = \top$;
- is *axiom monotone*, that is,

$$((\Gamma_1 \vee \Gamma_2) \vdash_\Sigma \Gamma_3) \geq (\Gamma_1 \vdash_\Sigma \Gamma_3) \vee (\Gamma_2 \vdash_\Sigma \Gamma_3);$$

- is *consequent invariant*, i.e.,

$$(\Gamma_1 \vdash_\Sigma \Gamma_2) \wedge (\Gamma_1 \vdash_\Sigma \Gamma_3) = (\Gamma_1 \vdash_\Sigma (\Gamma_2 \vee \Gamma_3));$$

- is transitive in the sense that

$$(\Gamma_1 \vdash_\Sigma \Gamma_2) \wedge ((\Gamma_1 \vee \Gamma_2) \vdash_\Sigma \Gamma_3) \leq (\Gamma_1 \vdash_\Sigma \Gamma_3); \text{ and}$$

- is an $\vdash$-*translation*, meaning that

$$(\Gamma_1 \vdash_\Sigma \Gamma_2) \leq (\Phi\mathrm{Sen}(\sigma)(\Gamma_1) \vdash_{\Sigma'} \Phi\mathrm{Sen}(\sigma)(\Gamma_2))$$

  for all signature morphisms $\sigma \in \mathrm{Hom}_{\mathtt{Sign}}(\Sigma, \Sigma')$.

A generalized institution

$$\mathscr{I} = (\texttt{Sign}, \mathsf{Sen}, \mathsf{Mod}, \mathbf{\Phi}, L, \models)$$

is a structure where

- $\texttt{Sign}$ is a category of signatures;
- Sen is a functor $\mathsf{Sen}\colon \texttt{Sign} \longrightarrow \texttt{Set}$ taking signatures to sentences,
- $\mathsf{Mod}\colon \texttt{Sign} \longrightarrow \texttt{Cat}^{op}$ is a functor with $\mathsf{Mod}(\Sigma)$ representing the category of Σ-*models*;
- L is a completely distributive lattice; and
- $\models$ is a family of L-valued relations consisting of

$$\models_{\Sigma}\colon \ \mathsf{Ob}(\mathsf{Mod}(\Sigma)) \times \Phi\mathsf{Sen}(\Sigma) \longrightarrow L$$

for each signature $\Sigma \in \mathsf{Ob}(\texttt{Sign})$ where $\models_{\Sigma}$ is called a Σ-*satisfaction relation*.

The $\models_\Sigma$ relations must fulfill the *satisfaction condition* that states that for all signature morphisms $\sigma \in \mathrm{Hom}_{\mathtt{Sign}}(\Sigma, \Sigma')$, models $M \in \mathrm{Ob}(\mathrm{Mod}(\Sigma))$ and theoremata $\Gamma \in \Phi\mathrm{Sen}(\Sigma)$, $\models_\Sigma$ must be such that

$$(\mathrm{Mod}(\sigma)(M) \models_\Sigma \Gamma) = (M \models_{\Sigma'} \Phi\mathrm{Sen}(\sigma)(\Gamma)).$$

A logic is a tuple

- $\mathscr{L} = (\mathtt{Sign}, \mathtt{C}, \Theta, \mathtt{D}, \mathsf{Sen}, \mathsf{Mod}, \Phi, L, \vdash, \models)$

and an object in a category of logics, generalizing quite broadly the Burstall-Goguen-Meseguer frameworks of institutions and entailment systems. Doing so we in fact more specific about the sentence functor, which in Burstall-Goguen-Meseguer frameworks are concretized only in specific examples such as for FOL and EL.

More detail can be found in Robert Helgesson's thesis.

Type theory as initiated by Schönfinkel, Curry and Church

- As we have seen, going from one-sorted to many-sorted must be done properly, so that going beyond `Set` can be done properly.
- Schönfinkel was 'untyped', Curry 'simply typed', and Church introduced the intuition about his ι and *o* 'types'.
- They were all unclear about in which signature these 'types' (as sorts) and 'type constructors' (as operators) shold reside.
- The formal description of the conventional set of terms over a signature is clear, but the formalization of the set of λ-terms is less obvious.
- Could we, for instance, avoid the renaming issue with a more strict construction of the set of λ-terms?

- We introduce ‘levels of signatures’ in order to handle the ‘type’ sort (Church’s ι) and type constructors in a signature of its own.
- Further we depart from λ-abstraction in that we say that operators in the underlying signature “owns” their abstractions.
- Note that Church indeed called “λ” an improper symbol.

Levels of signatures for simply typed λ-calculus

1. Level one: The level of 'primitive and underlying' sorts and operations, with a many-sorted signature

$$\Sigma = (S, \Omega)$$

2. Level two: The level of 'type constructors', with a single-sorted signature

$$\lambda_\Sigma = (\{\iota\}, \{\mathtt{s} :\to \iota \mid \mathtt{s} \in S\} \cup \{\Rrightarrow : \iota \times \iota \to \iota\})$$

3. Level three: The level in which we may construct 'λ-terms' based on the signature

$$\Sigma^\lambda = (S^\lambda, \Omega^\lambda)$$

where $S^\lambda = \mathsf{T}_{\lambda_\Sigma}\varnothing$, $\Omega^\lambda = \{\omega^\lambda_{i_1,\ldots,i_n} :\to (\mathtt{s}_{i_1} \Rrightarrow \cdots \Rrightarrow (\mathtt{s}_{i_{n-1}} \Rrightarrow (\mathtt{s}_{i_n} \Rrightarrow \mathtt{s})\cdots) \mid \omega : \mathtt{s}_1 \times \ldots \times \mathtt{s}_n \to \mathtt{s} \in \Omega, (i_1, \ldots, i_n)$ is a permutation of $(1, \ldots, n)\} \cup \{\mathtt{app}_{\mathtt{s},\mathtt{t}} : (\mathtt{s} \Rrightarrow \mathtt{t}) \times \mathtt{s} \to \mathtt{t}\}$

The natural numbers signature in levels

1. Level one:

$$\mathtt{NAT} = (\{\mathtt{nat}\}, \{\mathtt{0} :\to \mathtt{nat}, \mathtt{succ} : \mathtt{nat} \to \mathtt{nat}\})$$

2. Level two:

$$\lambda_{\mathtt{NAT}} = (\{\iota\}, \{\mathtt{nat} :\to \iota, \Rrightarrow : \iota \times \iota \to \iota\})$$

3. Level three:

$$\Sigma^\lambda = (\mathsf{T}_{\lambda_{\mathtt{NAT}}\varnothing}, \Omega^\lambda)$$

where $\Omega^\lambda = \{\mathtt{0}^\lambda :\to \mathtt{nat}, \mathtt{succ}^\lambda_1 :\to (\mathtt{nat} \Rrightarrow \mathtt{nat})\} \cup \{\mathtt{app}_{\mathtt{s},\mathtt{t}} : (\mathtt{s} \Rrightarrow \mathtt{t}) \times \mathtt{s} \to \mathtt{t}\}$

λ-calculus

... so then we can do λ-calculus, fuzzy λ-calculus, λ-calculus with fuzzy, and so on.

See our “Fuzzy terms” paper in the special FSS issue LINZ 2012.

$\Sigma_{\texttt{DescriptionLogic}} = (S, \Omega)$

1. $S = \{\texttt{concept}\}$, and we may add constants like $\texttt{c}_1, \ldots, \texttt{c}_n :\to \texttt{concept}$.
2. We include a type constructor $\texttt{P} : \texttt{type} \to \texttt{type}$ into $\texttt{S}_\Omega$, with an intuitive semantics of being the powerset functor, so that $\texttt{Pconcept}$ is the constructed type for "powerconcept".
3. "Roles" are $\texttt{r} :\to (\texttt{Pconcept} \Rrightarrow \texttt{PPconcept})$, and we need operators $\eta :\to (\texttt{concept} \Rrightarrow \texttt{Pconcept})$ and $\mu :\to (\texttt{PPconcept} \Rrightarrow \texttt{Pconcept})$ in Ω', so that "$\exists r.x$" can be defined as

$$\texttt{app}_{\texttt{PPconcept},\texttt{Pconcept}}(\mu, \texttt{app}_{\texttt{Pconcept},\texttt{PPconcept}}(\texttt{r}, x)).$$

The functor $\mathsf{Q}_S \circ \mathsf{T}_{\Sigma_{\texttt{DescriptionLogic}}}$ over $\texttt{Set}$ then provides a "fuzzy description logic" close to the sense of Yen (1991) and Straccia (1998), and $\mathsf{T}_{\Sigma_{\texttt{DescriptionLogic}}}$ over $\texttt{Set}(\mathfrak{Q})$ is not found in that literature.

Renaming

- In traditional notation, substituting x by $\mathtt{succ}(y)$ in $\lambda y.\mathtt{succ}(x)$ should cause a rename of the bound variable y, e.g., $\lambda z.\mathtt{succ}(\mathtt{succ}(y))$.
- On level 1, we have the substitution (Kleisli morphism) $\sigma_{\mathtt{nat}} : X_{\mathtt{nat}} \longrightarrow \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}\{X_{\mathtt{t}}\}_{\mathtt{t}\in\{\mathtt{nat}\}}$, where $\sigma_{\mathtt{nat}}(x) = \mathtt{succ}(y)$, x being a variable on level 1, and the extension of $\sigma_{\mathtt{nat}}$ is $\mu_{\mathtt{nat}} \circ \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}\sigma_{\mathtt{nat}} : \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}\{X_{\mathtt{t}}\}_{\mathtt{t}\in\{\mathtt{nat}\}} \longrightarrow \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}\{X_{\mathtt{t}}\}_{\mathtt{t}\in\{\mathtt{nat}\}}$.
- On level 3 we have $\sigma'_{\mathtt{nat}} : X_{\mathtt{nat}} \longrightarrow \mathsf{T}_{\mathtt{NAT}',\mathtt{nat}}\{X_{\mathtt{t}}\}_{\mathtt{t}\in S''}$, with $\sigma'_{\mathtt{nat}}(x) = \mathtt{app}_{\mathtt{nat},\mathtt{nat}}(\mathtt{succ}_1^{\lambda}, x)$, x being a variable on level 3, and $\mu'_{\mathtt{nat}} \circ \mathsf{T}_{\mathtt{NAT}',\mathtt{nat}}\sigma'_{\mathtt{nat}}(\mathtt{app}_{\mathtt{nat},\mathtt{nat}}(\mathtt{succ}_1^{\lambda}, x))$ requiring no renaming.

Schönfinkel's *Bausteine* (1920)

The *constancy function* C, defined as $(Ca)y = a$, can be seen as the type constructor $\mathtt{C} : \mathtt{type} \times \mathtt{type} \rightarrow \mathtt{type}$ fulfilling the 'equational condition' $\mathtt{C}(\mathtt{s}, \mathtt{t}) = \mathtt{s}$, and $\mathfrak{A}_{\mathtt{C}_\Sigma}$ would again be a functor fulfilling the corresponding criteria. Additionally, C can also be seen as an operator within Σ' as
$\mathtt{C}_{\mathtt{s},\mathtt{t}} :\rightarrow (\mathtt{s} \Rrightarrow (\mathtt{t} \Rrightarrow \mathtt{s}))$, with
$\mathfrak{A}_{\Sigma'}(\mathtt{C}_{\mathtt{s},\mathtt{t}}) \in \mathrm{Hom}(\mathfrak{A}_{\Sigma'}(\mathtt{s}), \mathrm{Hom}(\mathfrak{A}_{\Sigma'}(\mathtt{t}), \mathfrak{A}_{\Sigma'}(\mathtt{s})))$ so that
$\mathfrak{A}_{\Sigma'}(\mathtt{C}_{\mathtt{s},\mathtt{t}})(x)(y) = x$ for $x \in \mathfrak{A}_{\Sigma'}(\mathtt{s})$ and $y \in \mathfrak{A}_{\Sigma'}(\mathtt{t})$. A sentence, in equational type logic, prescribing the constancy function condition would then look like $\mathtt{app}_{\mathtt{s},\mathtt{t}}(\mathtt{C}_{\mathtt{s},\mathtt{t}}, \mathtt{t}) = \mathtt{s}$.

- Some of Schönfinkel's "operators" I, C, T, Z and S can be 'simply typed' on level two and three (I, C), and some on level three only (T, Z and S).
- See "Modern eyes on λ-calculus" (GLIOC notes, www.glioc.com)

Curry's *functionality* (1934)

Curry, like Schönfinkel, is weak on making distinction between syntax and semantics, so F on signature level two would be $F = \Rrightarrow : \mathtt{type} \to \mathtt{type}$ so that FXY is the term $X \Rrightarrow Y$, with $X, Y :: \mathtt{type}$. Thus, Curry's $\vdash FXYf$, *representing the statement that f belongs to that category*, means f is the constant $f : X \Rrightarrow Y$. Both F and f is by Curry called 'entities', but they are operators within different signatures.

- Curry believes that point that variables *may be introduced into the formal developments without loss of precision*.
- This, in our view, is the “what belongs and what does nt” of variables, leading to fear about ‘loss of precision’.
- Variables were at that time mostly viewed as ‘distinct from constants’.
- Curry writes further that *variables are not the names of any entities whatever, but are “incomplete symbols”, whose function is to indicate possibilities of substitution*.

Church's *simple typing* (1940)

We purposely refrain from making more definite the nature of the types o and ι, the formal theory admitting of a variety of interpretations in this regard. Of course the matter of interpretation is in any case irrelevant to the abstract construction of the theory, and indeed other and quite different interpretations are possible (formal consistency assumed).

- Our $(\beta \Rrightarrow \alpha)$ is Church's $(\beta\alpha)$.
- Speaking in terms of modern type theory involving '$\mathtt{type}$' and '$\mathtt{prop}$', Church's ι, as we have said, is our $\mathtt{type}$ on signature level two, but o is not something like $\mathtt{bool}$, but more like a '$\mathtt{prop}$', which is more unclear.
- We could imagine a $\Rrightarrow_{\mathtt{prop},\mathtt{type},\mathtt{type}}: \mathtt{type} \times \mathtt{type} \to \mathtt{prop}$ corresponding to Church's $o\iota\iota$, but it is not obvious how to deal with it.
- Intuitively, a quantifier may look like $\Pi : \mathtt{type} \times \mathtt{prop} \to \mathtt{prop}$, i.e., like Church's $\Pi_{o(o\alpha)}$, but again, it is not clear how to proceed.
- The algebras of $\mathtt{type}$ and $\mathtt{prop}$ also need to be settled.

- Church's $I_{\alpha\alpha}$ operator is Schönfinkel's identity function I, and Church's $K_{\alpha\beta\alpha}$ operator is Schönfinkel's constancy function C.
- His syntactic definitions of natural numbers $0_{\alpha'}$, $1_{\alpha'}$, $2_{\alpha'}$, $3_{\alpha'}$, etc., is then kind of assuming that the topmost signature Σ is the empty signature.
- Church's 'variable binding' operator, or *choice function*, $\iota_{\alpha(o\alpha)}$, is influence e.g. by Hilbert's ϵ-operator in the ϵ-calculus culminating in Ackermann's thesis 1924.
- The $\iota_{\alpha(o\alpha)}$ operator obviously has its counterpart in our framework as well, but appears differently since variables are only implicitly pointed at by the indices appearing in $\omega^{\lambda}_{i_1,\ldots,i_n}$.

The Brouwer-Heyting-Kolmogorov interpretation

Appears in its well-known form propositionally presented by Komogorov in 1932, *Zur Deutung der Intuitionistischen Logik*:

- Es gilt dann die folgende merkwürdige Tatsache: *Nach der Form fällt die Aufgabenrechnung mit der Brouwersehen, von Herrn Heyting neuerdings formaliaierten, intuitionistischen Logik zusammen*.
- Wit glauben, daß nach diesen Beispielen und Erklärungen die Begriffe "Aufgabe" und "Lösung der Aufgabe" in allen Fällen, welche in den konkreten Gebieten der Mathematik vorkommen, ohne Mißverständnis gebraucht werden können. Die Hauptbegriffe der Aussagenlogik "Aussage" und "Beweis der Aussage" befinden sich nicht in besserer Lage.
- Wenn a und b zwei Aufgaben sind, bezeichnet $a \wedge b$ die Aufgabe "beide Aufgaben a und b lösen", ...

The Curry-Howard isomorphism

Appears in its most well-known form presented by Howard in 1969/1980, *The formulae-as-types notion of construction* and was based e.g. on Curry's and Fey's *Combinatory Logic* from 1958:

- The following consists of notes which were privately circulated in 1969. Since they have been referred to a few times in the literature, it seems worth while to publish them. (Howard,1980)
- Let $P(\supset)$ denote positive implicational propositional logic. By a type symbol is meant a formula of $P(\supset)$. (Howard,1980)
- This can be seen as $\Sigma = (S, \varnothing)$, on level 1, where S is viewed as the set of 'prime formulae', $\mathsf{T}_{\lambda_\Sigma}\varnothing$ is the set of all formulae in $P(\supset)$.

- If we now have $\mathtt{BOOL} = (\{\mathtt{bool}\}, \{\mathtt{a}_i :\to \mathtt{bool} \mid i \in I\} \cup \{\Rightarrow, \wedge : \mathtt{bool} \times \mathtt{bool} \to \mathtt{bool}\})$ on level one, then $\mathtt{BOOL}' = (\mathsf{T}_{\lambda_\Sigma}\varnothing, \{\mathtt{a}_{i_0}^{\lambda} :\to \mathtt{bool} \mid i \in I\} \cup \{\Rightarrow_{1,2}^{\lambda}, \wedge_{1,2}^{\lambda} :\to (\mathtt{bool} \Rrightarrow (\mathtt{bool} \Rrightarrow \mathtt{bool}))\} \cup \{\mathtt{app}_{\mathtt{s},\mathtt{t}} : (\mathtt{s} \Rrightarrow \mathtt{t}) \times \mathtt{s} \to \mathtt{t} \mid \mathtt{s}, \mathtt{t} \in \mathsf{T}_{\lambda_\Sigma}\varnothing\})$ providing $\mathsf{T}_{\mathtt{BOOL}'}\varnothing$ on level three is not to be confused with $\mathsf{T}_{\lambda_\Sigma}\varnothing$ on level two.
- Adding Schönfinkel's $\mathtt{C}_{\mathtt{s},\mathtt{t}} :\to (\mathtt{s} \Rrightarrow (\mathtt{t} \Rrightarrow \mathtt{s}))$ (Curry's $\mathtt{K}$) as an operator on level 3 is then seen as an 'axiom'.

Algebras

- In the two-valued case, $\mathfrak{A}(\texttt{bool})$ is often $\{\textit{false}, \textit{true}\}$, so that $\mathfrak{A}(\texttt{F}) = \textit{false}$ and $\mathfrak{A}(\texttt{T}) = \textit{true}$.
- $\mathfrak{A}(\&) : \mathfrak{A}(\texttt{bool}) \times \mathfrak{A}(\texttt{bool}) \longrightarrow \mathfrak{A}(\texttt{bool})$, is expected to be defined by the usual 'truth table'.
- We may assign for a signature $\Sigma_{PL} = (S_{PL}, \Omega_{PL})$ a pair, the 'many-sorted algebra', $(\mathsf{T}_{\Sigma_{PL}} X_S, (\mathfrak{A}(\omega))_{\omega \in \Omega_{PL}})$, where $X_{\texttt{s}} = \varnothing$ if $\texttt{s} \neq \texttt{bool}$.
- Then, $(\bigcup_{\texttt{s} \in S} (\mathrm{arg}^{\texttt{s}} \circ \mathsf{T}_{\Sigma_{PL}}) X_S, (\texttt{F}, \texttt{T}, \&, \neg))$ serves as a traditional Boolean algebra, when certain equational laws are given.

Programs and their interpretations (paper presented at WILF 2014

- $\Gamma = \{(h_1, b_1), \ldots, (h_n, b_n)\} \subseteq \mathsf{Sen}_{HCL} X_S$
- $(U_\Gamma)_S = \mathsf{T}_\Sigma \varnothing_S = (\mathsf{T}_{\Sigma,\mathtt{s}} \varnothing_S)_{\mathtt{s} \in S}$
- $\bigcup_{\mathtt{s} \in S} (U_\Gamma)_\mathtt{s}$ corresponds to the traditional and unsorted view of the *Herbrand universe*
- $B_\Gamma = (\mathrm{arg}^{\mathtt{bool}} \circ \mathsf{T}_{\Sigma_{PL \setminus \neg, \&}} \circ \mathsf{T}_\Sigma)\ \varnothing_S$ corresponds to the *Herbrand base*
- Herbrand interpretations of a program Γ are subsets $\mathcal{I} \subseteq B_\Gamma$
- we also need what we call the *Herbrand expression base*: $B_\Gamma^\& = (\mathrm{arg}^{\mathtt{bool}} \circ \mathsf{T}_{\Sigma_{PL \setminus \neg}} \circ \mathsf{T}_\Sigma)\ \varnothing_S$
- a Herbrand interpretation $\mathcal{I}$ canonically extends to a *Herbrand expression interpretation* $\mathcal{I}^\& \subseteq B_\Gamma^\&$

Substitution fuzzy Horn clause logic

- fuzzy sets of predicates:
  $\mathsf{L}B_\Gamma = (\mathsf{L} \circ \arg^{\mathtt{bool}} \circ \mathsf{T}_{\Sigma_{PL \setminus \neg,\&}} \circ \mathsf{T}_\Sigma)\, \varnothing_S$
- sentence functor:
  $\mathrm{Sen}_{SFHCL} = (\arg^{\mathtt{bool}})^2 \circ ((\mathsf{T}_{\Sigma_{PL \setminus \neg,\&}} \times \mathsf{T}_{\Sigma_{PL \setminus \neg}}) \circ \mathsf{L}_S \circ \mathsf{T}_\Sigma \circ \phi^{S \setminus \mathtt{bool}})$
- ground predicates over fuzzy sets of terms:
  $B_\Gamma^{\mathsf{L}} = (\arg^{\mathtt{bool}} \circ \mathsf{T}_{\Sigma_{PL \setminus \neg,\&}} \circ \mathsf{L}_S \circ \mathsf{T}_\Sigma)\, \varnothing_S$
- the fuzzy sets of ground predicates is enabled by the ‘swapper’: $\varsigma : \mathsf{T}_{\Sigma_{PL \setminus \neg,\&}} \circ \mathsf{L}_S \longrightarrow \mathsf{L}_S \circ \mathsf{T}_{\Sigma_{PL \setminus \neg,\&}}$

Fixpoints

- considering the effect of substitutions with fuzzy sets of terms: $\varpi^{\mathsf{L}} : \mathsf{L}B_{\Gamma}^{\mathsf{L}} \longrightarrow \mathsf{L}B_{\Gamma}^{\mathsf{L}}$
- $\mathrm{arg}^{\mathtt{bool}}\varsigma_{\mathsf{T}_{\Sigma}\varnothing_{S}} : B_{\Gamma}^{\mathsf{L}} \longrightarrow \mathsf{L}B_{\Gamma}$

$$\varpi^{\mathsf{L}}(\mathcal{I})(\sigma_{\mathtt{bool}}^{\mathsf{L},head}(h)) =$$
$$(\bigvee_{t \in B_{\Gamma}}(\mathrm{arg}^{\mathtt{bool}}\varsigma_{\mathsf{T}_{\Sigma}\varnothing_{S}}(h))(t)) \wedge \mathcal{I}^{\mathsf{L},\&}(\sigma_{\mathtt{bool}}^{\mathsf{L},body}(b))$$

Terminologies, classifications and ontologies in social and health care

- WHO's ICF and ICD-10
- ATC for drugs
- SNOMED which is believed to have description logic as its underlying logic for ontology (health onttology and web ontology is not the same thing!)
- fall risk and fall injury risk

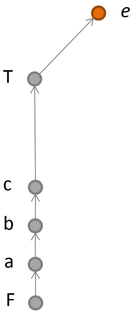
```
Muscle functions (ICF b730-b749)
  Muscle power functions (b730)
    ...
    Power of muscles of all limbs (b7304)
    ...
  Muscle tone functions (b735)
  Muscle endurance functions (b740)
```

The ICF datatypes and its generic scale of quantifiers:

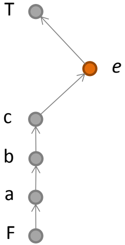
```
xxx.0 NO problem          (none, absent, ...)
xxx.1 MILD problem        (slight, low, ...)
xxx.2 MODERATE problem    (medium, fair, ...)
xxx.3 SEVERE problem      (high, extreme, ...)
xxx.4 COMPLETE problem    (total, ...)
xxx.8 not specified
xxx.9 not applicable
```

Unknown as unital *e* with 5-valued set {F, a, b, c, T} of **truth values**, corresponding to the ICF valuations, including the unknown as 'not specified' (problem qualifier code 8)

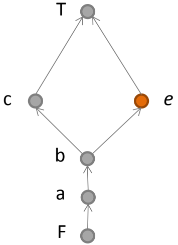
F - a - b - c - T - *e*



F - a - b - c - *e* - T



F - a - b - {c | *e*} - T



ICD-10

```
S52   fracture of forearm
S52.5 fracture of lower end of radius
```

and conflicting ICD-10 extensions, with the ICD-10-CM adopted in the US going further in direction of

```
S52.53   Colles' fracture of radius
S52.532  Colles' fracture of left radius
S52.532D Colles' fracture of left radius,
         subsequent encounter for closed
         fracture with routine healing
```

where “3” for ‘Colles’ means dorsal displacement, “2” and “-” after “53” means ‘left or unspecified arm, and “D” means subsequent encounter for closed fracture with routine healing.

For comparison, in Germany, the ICD-10-GM (2014) uses

```
S52.5  Distale Fraktur des Radius
S52.51 Extensionsfraktur, Colles-Fraktur
```

i.e.,'Colles' now is "51", where the US version says "53". Thus, we have to be "internationally careful" when we see a code like "S52.51".
In Sweden, the ICD-10-SE is only ICD

```
S52.5 Fraktur på nedre delen av radius
```

whereas the Swedish Orthopaedic Association uses

```
S52.50/51 Distal radius (Barton, Colles, Smith)
```

where "0" is left and "1" is right, so the Swedish "S52.51" is different from the German one, and different from the corresponding US code.

Sleeping pills affect the balance so the use of sedatives is a fall risk factor

Anatomic Therapeutic Chemical (ATC) classification of *nitrazepam* (code C08DA01), long-acting drug for insomnia:

N	nervous system	1st level main anatomical group
N05	psycholeptics	2nd level, therapeutic subgroup
N05C	hypnotics and sedatives	3rd level, pharmacological subgroup
N05CD	benzodiazepine derivatives	4th level, chemical subgroup
N05CD02	nitrazepam	5th level

Downton’s Fall Risk Index (DFRI) assessment scale includes the item ‘tranquilizers/sedatives’ under “Medications”, so the user is providing drug information related to a pharmacological subgroup (3rd level), where nitrazepam (5th level) is one of the most fall-risk-increasing drugs (FRIDs). Then again, on interventions it is easy to speak generally about the effect of “withdrawal of psychotropics” (2nd level). Obviously, from formal information management point of view, the health care domain does not always consider data typing and granularity issues.

For ATC, on level two we could have

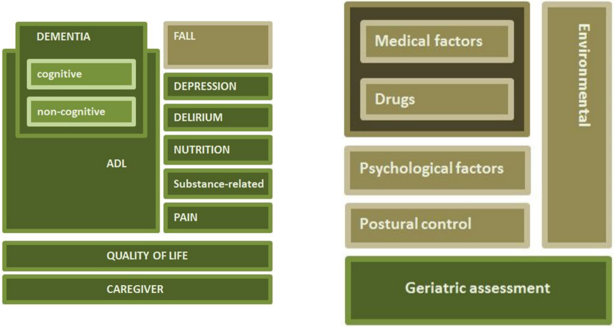
$$\mathtt{1st}, \mathtt{2nd}, \mathtt{3rd}, \mathtt{4th}, \mathtt{5th} :\to \mathtt{type}$$

and on level three

$$\begin{aligned}
\mathtt{PharmacologicIntervention} &:\to \mathsf{P}(\mathtt{3rd}) \\
\mathtt{DrugPrescriptions} &:\to \mathsf{P}(\mathtt{5th}) \\
\textit{hypnotics_and_sedatives} &:\to \mathtt{3rd} \\
\textit{benzodiazepine_derivatives} &:\to \mathtt{4th} \\
\textit{nitrazepam} &:\to \mathtt{5th} \\
\textit{drug} &:\to \mathtt{5th} \\
\phi^{\mathtt{5th}\to\mathtt{4th}} &: \mathtt{5th} \to \mathtt{4th} \\
\phi^{\mathtt{4th}\to\mathtt{3rd}} &: \mathtt{4th} \to \mathtt{3rd} \\
\phi^{\mathtt{5th}\to\mathtt{3rd}} &: \mathtt{5th} \to \mathtt{3rd}
\end{aligned}$$

This then makes a clear distinction between *nitrazepam* as a term of type $\mathtt{5th}$ and $\phi^{\mathtt{5th}\to\mathtt{3rd}}$(*nitrazepam*) as a sedative of type $\mathtt{3rd}$. Further, for the variable *drug*, we can make a substitution with *nitrazepam*, because the types match, but we cannot substitute with *hypnotics_and_sedatives*. For Downton's index the consequence is that $\phi^{\mathtt{5th}\to\mathtt{3rd}}$(*drug*) may appear as a value in the scale, but not *drug*. This is also important in considerations of uncertainty. A relative to a patient may be fairly sure about *hypnotics_and_sedatives*, but not all that certain about that sedative being a *benzodiazepine_derivatives*. Additional operators is required to capture the notion of uncertainty being carried over between ATC levels.

Gerontological and geriatric assessment in general, and fall risk assessment in particular.



Implementations e.g. within the AAL Call 4 project AiB (Ageing in Balance)

Level one:

$$\mathtt{GERONTIUM} = (S, \Omega)$$

where $S = \{\mathtt{nat}, \mathtt{bool}, \mathtt{scale}, \ldots\}$. Operators in Ω can be provided in a number of ways, and is left unspecified at this point.

Level two:

$$\lambda_{\texttt{GERONTIUM}} = (\{\texttt{Observation}, \texttt{Assessment}\}, \lambda_\Omega)$$

λ_Ω:

$$\begin{aligned}
\texttt{s} &: \to \texttt{Observation}, \texttt{s} \in \mathcal{S} \\
\boxtimes &: \texttt{Observation} \times \texttt{Observation} \to \texttt{Observation} \\
\boxplus &: \texttt{Assessment} \times \texttt{Assessment} \to \texttt{Assessment} \\
\texttt{P} &: \texttt{Assessment} \to \texttt{Assessment} \\
\Rrightarrow_{\texttt{Observation}} &: \texttt{Observation} \times \texttt{Observation} \to \texttt{Observation} \\
\Rrightarrow_{\texttt{Assessment}} &: \texttt{Assessment} \times \texttt{Assessment} \to \texttt{Assessment}
\end{aligned}$$

$$
\begin{aligned}
\texttt{CognitiveDementia} &: \to \texttt{Assessment} \\
\texttt{Non-CognitiveDementia} &: \to \texttt{Assessment} \\
\texttt{ADL} &: \to \texttt{Assessment} \\
\texttt{Depression} &: \to \texttt{Assessment} \\
\texttt{Delirium} &: \to \texttt{Assessment} \\
\texttt{Nutrition} &: \to \texttt{Assessment} \\
\texttt{SubstanceRelated} &: \to \texttt{Assessment} \\
\texttt{Pain} &: \to \texttt{Assessment}
\end{aligned}
$$

$$
\texttt{GeriatricAssessment} : \to \texttt{Assessment}
$$

$$\begin{aligned}
\texttt{MedicalFactors} &: \to \texttt{Assessment} \\
\texttt{Drugs} &: \to \texttt{Assessment} \\
\texttt{PsychologicalFactors} &: \to \texttt{Assessment} \\
\texttt{PosturalControl} &: \to \texttt{Assessment} \\
\texttt{EnvironmentalFactors} &: \to \texttt{Assessment}
\end{aligned}$$

$$\texttt{FallRiskAssessment} : \to \texttt{Assessment}$$

Level three:

$$\mathtt{GERONTIUM}^{\lambda} = (T_{\lambda_{\mathtt{GERONTIUM}}\varnothing}, \Omega^{\lambda})$$

Ω^{λ}, including the *Falls Efficacy Scale - International* (FES-I) as an example of an assessment scale:

$$\begin{aligned}\mathtt{FES-I} :\to (&\mathtt{scale4}\natural^{16} \\ &\Rrightarrow (\mathtt{scale64} \boxtimes \mathtt{scale3} \\ &\boxtimes \mathtt{PsychologicalFactors}))\end{aligned}$$

$$\mathtt{Odepression} : \mathtt{P\ Depression} \to \mathtt{Depression}$$

$$\mathtt{OA} :\to \mathtt{P\ CognitiveDementia} \boxplus \ldots$$

$$\mathtt{FallOA} :\to \mathtt{P\ MedicalFactors} \boxplus \ldots$$

$$\mathtt{app}_{\mathtt{s,t}} : (\mathtt{s} \Rrightarrow \mathtt{t}) \times \mathtt{s} \to \mathtt{t}$$