

LING5702: Lecture Notes 2

Logical Reasoning

Language seems similar to thought, but we can distinguish them. We'll start with thought — specifically, 'conscious'/'rational' thought: evaluating and implementing but not inventing plans.

Contents

2.1	Decision theory (von Neumann & Morgenstern, 1944)	1
2.2	Basic parts of meaning (ontology) (Church, 1940; Carnap, 1947)	2
2.3	Logical expressions (Church, 1940)	3
2.4	Simplification by variable substitution using beta reduction	4
2.5	Logical operators: implication, conjunction, disjunction and negation	5
2.6	Entailment (Heyting, 1930; Kleene, 1952)	6
2.7	Extra: intuitionistic logic theorems useful for defining grammar	8
2.8	Arithmetic (Peano, 1889; Heyting, 1930; Kleene, 1952)	10
2.9	Generalized quantifiers (Barwise & Cooper, 1981)	11
2.10	Example: grid-world navigation	15
2.11	Intensions (Carnap, 1947; Pollard, 2007)	16
2.12	Example: interpreting a question	17
2.13	Example: following a plan to answer	20

2.1 Decision theory (von Neumann & Morgenstern, 1944)

Conscious thought is a **decision process**: evaluating plans to maximize **average expected utility**. (There's lots of other thought: enjoying, reminiscing, etc., but this more directly helps us survive.)

A decision process assumes a **time** τ , a **plan** π , a **theory** θ about the world, and a **reward** $R(\theta)$.

For example:

- τ may be noon on a particular day,
- π may be a plan to walk to a hill,
- θ may include our location and the knowledge that a step may take us closer to a goal,
- $R(\theta)$ may be the position or status we get from reaching the hill.

We'll also need a special probability over 'trials' v , which are **possible successor** times to τ (we'll use $\text{Suc } \tau$ to define a unique **actual successor** to τ , and $\text{Cur } \tau$ to flag τ as **current**):

$$P_{\tau}(\lambda_v \sigma v \mid \varphi) = P(\lambda_v \sigma v \mid \lambda_v \varphi \wedge \text{PossibleSuccessor } \tau v)$$

(The lambda term λ_v identifies the random variable over trials, and will be further explained later.)

We then use these probabilities in a decision process to calculate time-averaged expected utility.

Average (over time τ) expected reward for a plan π is a sum over **outcomes** ω of **actions** δ in π :

$$U(\pi, \theta, \tau) = \overbrace{R_\tau(\theta)}^{\text{reward}} \cdot \begin{cases} \text{if } \exists_\delta \pi, \theta \wedge \text{Cur } \tau \Rightarrow \delta: \sum_\omega \overbrace{P_\tau(\omega \mid \theta \wedge \delta)}^{\text{prob. of outcome } \omega \text{ of } \delta} \cdot \underbrace{U(\pi, \theta \wedge \delta \wedge \omega (\text{Suc } \tau), \text{Suc } \tau)}_{\text{repeat with } \theta, \delta \text{ and } \omega \text{ as new } \theta} \\ \text{otherwise:} & \frac{1}{\tau} \end{cases}$$

(It assumes the plan π is perfectly specific: at most one next action δ for each theory θ .)

This tells us how good a plan is. We can use it to choose among plans to maximize average reward!

For example, if the goal hill is one step away, we get a reward in one step, so $U(\pi, \theta, \tau) = 1$.

But if it's muddy and we slip half the time and don't go anywhere, then:

$$U(\pi, \theta, \tau) = \begin{cases} .5 \text{ (slip)} & \times \begin{cases} .5 \text{ (slip)} & \times \begin{cases} .5 \text{ (slip)} & \times \dots \\ +.5 \text{ (no slip)} & \times \frac{1}{3} \text{ (arrive in 3 steps)} \end{cases} \\ +.5 \text{ (no slip)} & \times \frac{1}{2} \text{ (arrive in 2 steps)} \end{cases} \\ +.5 \text{ (no slip)} & \times \frac{1}{1} \text{ (arrive in 1 step)} \end{cases} \approx .7$$

So if we have two plans (clear path and muddy path) and we know mud slows us, we can avoid it.

We speculate language provides an evolutionary advantage by letting us *share* plans and theories.

So what are these plans and theories? We'll define them as logical expressions in **lambda calculus**.

2.2 Basic parts of meaning (ontology) (Church, 1940; Carnap, 1947)

Lambda calculus is defined in terms of **objects** of various **types**:

1. **truth values** (type **t**): judgments about statements within a particular theory,
simple examples: **True** (tautology / entailed by theory), **False** (contradiction with theory);
2. **entities** (type **e**): referents of proper nouns within a particular theory,
examples: **Person43**, **Course101**, **GrainOfSand5324**, **InfinitesimalOfVolume3387**, ...;
3. **functions**: (type $(\alpha) \rightarrow \beta$ for any types α and β): associations between inputs and outputs,
examples: **Prof** of type $(e) \rightarrow t$, **Teach** of type $(e) \rightarrow (e) \rightarrow t$, ...
(functions of type $(e) \rightarrow t$ or $(e) \rightarrow (e) \rightarrow t$ or $(e) \rightarrow (e) \rightarrow (e) \rightarrow t$, ... are called **predicates**)
(predicates of type $(e) \rightarrow t$ are called **characteristic functions of sets**, or just **sets**);

4. **propositions** (type **p**): partial descriptions of desired, believed or claimed theories,
examples: **ItsSnowing**, ... (these can also be entire logical expressions, as we'll see later);
5. **individuals** (type **i**): referents that may or may not be distinct in different theories,
examples: **BruceWayne**, **Batman**, ... (they're just names, in case you don't like predicates).

These objects are not necessarily represented in the brain or mentioned in logical expressions.

They're just things we can describe in whatever theory we may reason about.

(This is cognitive modeling, so we must reason about *theories* rather than *worlds*, as in semantics.)

2.3 Logical expressions (Church, 1940)

Now we can define rules to build tree-like **logical expressions**, as with syntax in natural language.

These rules will use the above types, just like syntax rules use syntactic categories (as we'll see).

We start with two kinds of simple expression of any type α , shown as schematized grammar rules:

1. **constants** $\begin{array}{c} \alpha \\ \triangle \\ \varphi \end{array}$ — examples of φ of type α : **Person43** of type **e**, **Prof** of type $(\mathbf{e}) \rightarrow \mathbf{t}$, ...
2. **variables** $\begin{array}{c} \alpha \\ \triangle \\ \chi \end{array}$ — examples of χ of type α : x of type **e**, r of type $(\mathbf{e}) \rightarrow \mathbf{t}$, ...

We'll also define two basic rules to build complex expressions for any types α and β :

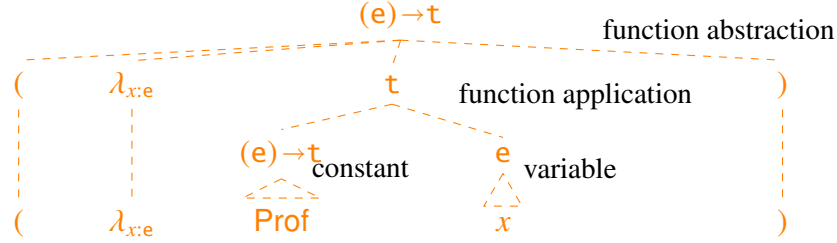
3. **applications** $\begin{array}{c} \beta \\ \swarrow \quad \searrow \\ (\alpha) \rightarrow \beta \quad \alpha \\ \swarrow \quad \searrow \\ \varphi \quad \psi \end{array}$ of functions φ of type $(\alpha) \rightarrow \beta$ to arguments ψ of type α :
example $\varphi \psi$ of type β : **Prof Person43** of type **t**, where **Prof** is type $(\mathbf{e}) \rightarrow \mathbf{t}$ and **Person43** is **e**,
4. **abstractions** $\begin{array}{c} (\alpha) \rightarrow \beta \\ \swarrow \quad \searrow \\ \lambda_{\chi:\alpha} \quad \beta \\ \swarrow \quad \searrow \\ \varphi \end{array}$ defining functions on input variables χ of type α with output φ :
example of $(\lambda_{x:\alpha} \varphi)$ of type $(\alpha) \rightarrow \beta$: $(\lambda_{x:\mathbf{e}} \mathbf{True})$ of type $(\mathbf{e}) \rightarrow \mathbf{t}$, where **True** is type **t**.

We'll also need parentheses to disambiguate scope:

5. **parentheses** $\begin{array}{c} \alpha \\ \swarrow \quad \searrow \\ \alpha \\ \swarrow \quad \searrow \\ \varphi \end{array}$ for any expression φ of type α .

This is called **lambda calculus** because of the abstraction operator.

We can draw trees for expressions in lambda calculus, just like in natural languages like English:



Practice 2.1:

Assuming:

- **Prof** is of type $(e) \rightarrow t$, and
- **Cardinality** is of type $((e) \rightarrow t) \rightarrow e$,

draw a tree for the following lambda calculus expression, showing type labels on all branches:

Cardinality $(\lambda_{x:e} \text{Prof } x)$.

2.4 Simplification by variable substitution using beta reduction

We can simplify application of function abstractions via **beta reduction**.

This is just variable substitution: $(\lambda_{\chi:\alpha} \varphi) \omega = [\varphi]_{\chi \mapsto \omega}$, replacing χ with ω via the following rules:

$$\begin{aligned} [\varphi]_{\chi \mapsto \omega} &= \varphi & \text{if } \varphi \text{ is a constant,} & & [\varphi \psi]_{\chi \mapsto \omega} &= [\varphi]_{\chi \mapsto \omega} [\psi]_{\chi \mapsto \omega}, \\ [\varphi]_{\chi \mapsto \omega} &= \varphi & \text{if } \varphi \text{ is a variable and } \varphi \neq \chi, & & [(\lambda_{\psi:\beta} \varphi)]_{\chi \mapsto \omega} &= (\lambda_{\psi:\beta} [\varphi]_{\chi \mapsto \omega}) & \text{if } \psi \neq \chi, \\ [\chi]_{\chi \mapsto \omega} &= \omega, & & & [(\lambda_{\chi:\alpha} \varphi)]_{\chi \mapsto \omega} &= (\lambda_{\chi:\alpha} \varphi). \end{aligned}$$

For example:

$$\begin{aligned} (\lambda_{x:e} \text{Like } x \text{ } x) \text{Person43} &= [\text{Like } x \text{ } x]_{x \mapsto \text{Person43}} && \text{by definition} \\ &= [\text{Like } x]_{x \mapsto \text{Person43}} [x]_{x \mapsto \text{Person43}} && \text{function application} \\ &= [\text{Like } x]_{x \mapsto \text{Person43}} \text{Person43} && \text{variable matching substituentum} \\ &= [\text{Like}]_{x \mapsto \text{Person43}} [x]_{x \mapsto \text{Person43}} \text{Person43} && \text{function application} \\ &= [\text{Like}]_{x \mapsto \text{Person43}} \text{Person43 Person43} && \text{variable matching substituentum} \\ &= \text{Like Person43 Person43} && \text{constant} \end{aligned}$$

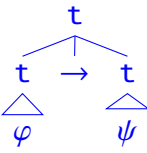
Practice 2.2:

Beta reduce the following expression: $(\lambda_{x:e} x) \text{ Person43}$.

2.5 Logical operators: implication, conjunction, disjunction and negation

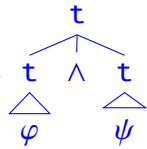
Some functions (on truth values) are so useful, they're just considered part of the logic.

We have 'infix' rules for the following **operators** (regular 'prefix' functions are harder to read):

6. **implication**  of protasis φ to apodosis ψ of type **t**, true if ψ when φ is true:

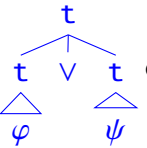
example of $\varphi \rightarrow \psi$: $\text{Prof Person43} \rightarrow \text{Employee Person43}$ of type **t**.

Implications also accommodate beta reduction: $[\varphi \rightarrow \psi]_{\chi \mapsto \omega} = [\varphi]_{\chi \mapsto \omega} \rightarrow [\psi]_{\chi \mapsto \omega}$.

7. **conjunction**  of conjuncts φ and ψ of type **t**, true (only) if φ and ψ are true:

example of $\varphi \wedge \psi$: $\text{Prof Person43} \wedge \text{True}$ of type **t**, where both conjuncts are type **t**.

Conjunctions also accommodate beta reduction: $[\varphi \wedge \psi]_{\chi \mapsto \omega} = [\varphi]_{\chi \mapsto \omega} \wedge [\psi]_{\chi \mapsto \omega}$.

8. **disjunction**  of disjuncts φ and ψ of type **t**, true (only) if φ or ψ are true:

example of $\varphi \vee \psi$: $\text{Prof Person43} \vee \text{True}$ of type **t**, where both disjuncts are type **t**.

Disjunctions also accommodate beta reduction: $[\varphi \vee \psi]_{\chi \mapsto \omega} = [\varphi]_{\chi \mapsto \omega} \vee [\psi]_{\chi \mapsto \omega}$.

We also have a special symbol for negation:

9. **negation**  of statement φ of type **t**, true (only) if φ is false:

example of $\neg \varphi$: $\neg \text{Prof Person43}$ of type **t**, where argument statement is of type **t**.

Negations also accommodate beta reduction: $[\neg \varphi]_{\chi \mapsto \omega} = \neg [\varphi]_{\chi \mapsto \omega}$.

Practice 2.3:

Assuming:

- **Prof** is of type $(e) \rightarrow t$,
- **Person43** is of type e , and
- **True** is of type t ,

draw a tree for the following lambda calculus expression, showing type labels on all branches:

Prof Person43 $\wedge$ True.

Practice 2.4:

Beta reduce the following expression: $(\lambda_{x:e} \text{Prof } x \wedge \text{True}) \text{Person43}$.

2.6 Entailment (Heyting, 1930; Kleene, 1952)

Generally we have two ways of acquiring new knowledge as logical propositions:

- from other people, via language (we'll discuss that in a subsequent lecture),
- from existing knowledge (other logical propositions, often called a 'theory'), via entailment.

Entailment $\pi_1, \pi_2, \dots \Rightarrow \kappa$ is a process of inferring conclusion statements κ from premises $\pi_1, \pi_2, \dots$

These entailments can be chained up in a proof, just like equations in a mathematical derivation.

Basic propositional logic entailments can be defined using just a **modus ponens** rule:

$$\varphi \rightarrow \psi, \varphi \Rightarrow \psi$$

and a definition of negation in terms of implication:

$$\neg\varphi \Leftrightarrow \varphi \rightarrow \text{False}$$

along with a small set of schematized axioms:

- for implication — for all θ, φ, ψ of type t :

$$\varphi \rightarrow (\psi \rightarrow \varphi)$$

implication introduction (II)

$$(\theta \rightarrow \varphi) \rightarrow ((\theta \rightarrow (\varphi \rightarrow \psi)) \rightarrow (\theta \rightarrow \psi))$$

implication self-distribution (ISD)

- for conjunction — for all φ, ψ of type $\mathbf{t}$:

$$\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$$

conjunction introduction (CI)

$$(\varphi \wedge \psi) \rightarrow \varphi \quad (\varphi \wedge \psi) \rightarrow \psi$$

conjunction elimination (CE)

- for disjunction — for all φ, ψ, χ of type $\mathbf{t}$:

$$\varphi \rightarrow (\varphi \vee \psi) \quad \psi \rightarrow (\varphi \vee \psi)$$

disjunction introduction (DI)

$$(\varphi \rightarrow \chi) \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\varphi \vee \psi) \rightarrow \chi))$$

disjunction elimination (DE)

- for negation — for all φ, ψ of type $\mathbf{t}$:

$$(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi)$$

negation introduction (NI)

$$\neg\varphi \rightarrow (\varphi \rightarrow \psi)$$

negation elimination / ex falso quodlibet (EFQ)

These are the axioms of **intuitionistic logic** (Brouwer, 1907; Heyting, 1930; Kleene, 1952).

Here they are used in a simple proof:

$$\text{Prof Kim} \wedge \text{Adult Pat} \Rightarrow \text{Prof Kim} \quad (\text{CE,MP})$$

We can use these to derive **theorems** like transitivity of implication (‘hypothetical syllogism,’ HS):

$$(B \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C)) \quad (1:\text{II})$$

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C))) \quad (2:\text{II}, 3:\text{MP from 1})$$

$$((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)) \rightarrow ((B \rightarrow C) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))) \quad (4:\text{II})$$

$$(A \rightarrow B) \rightarrow (((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)) \rightarrow ((B \rightarrow C) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)))) \quad (5:\text{II}, 6:\text{MP from 4})$$

$$(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)) \quad (7:\text{ISD})$$

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))) \quad (8:\text{ISD}, 9:\text{MP from 6 and 7})$$

$$((B \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C))) \rightarrow (((B \rightarrow C) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))) \quad (10:\text{ISD})$$

$$(A \rightarrow B) \rightarrow (((B \rightarrow C) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))) \quad (11:\text{ISD}, 12:\text{MP from 3 and 10})$$

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)) \quad (13:\text{MP from 9 and 12})$$

We can use HS to conjoin assumptions (let’s call this Theorem 1):

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow (A \rightarrow (B \rightarrow C))) \quad (1:\text{II})$$

$$((B \wedge A) \rightarrow A) \rightarrow (((B \wedge A) \rightarrow (A \rightarrow (B \rightarrow C))) \rightarrow ((B \wedge A) \rightarrow (B \rightarrow C))) \quad (2:\text{ISD})$$

$$(B \wedge A) \rightarrow A \quad (3:\text{CE})$$

$$((B \wedge A) \rightarrow (A \rightarrow (B \rightarrow C))) \rightarrow ((B \wedge A) \rightarrow (B \rightarrow C)) \quad (4:\text{MP from 2 and 3})$$

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow (B \rightarrow C)) \quad (5:\text{HS}, 6:\text{MP from 1 and 4})$$

$$((B \wedge A) \rightarrow B) \rightarrow (((B \wedge A) \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow C)) \quad (7:\text{ISD})$$

$$(B \wedge A) \rightarrow B \quad (8:\text{CE})$$

$$((B \wedge A) \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow C) \quad (9:\text{MP from 7 and 8})$$

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow C) \quad (10:HS, 11:MP \text{ from 6 and 9})$$

We can also use HS to un-conjoin assumptions (let's call this Theorem 2):

$$(B \rightarrow (A \wedge B)) \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C)) \quad (1:ISD)$$

$$A \rightarrow ((B \rightarrow (A \wedge B)) \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C))) \quad (2:II, 3:MP)$$

$$A \rightarrow (B \rightarrow (A \wedge B)) \quad (4:CI)$$

$$A \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C)) \quad (5:MP \text{ from 3 and 4})$$

$$((A \wedge B) \rightarrow C) \rightarrow (A \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C))) \quad (6:II, 7:MP)$$

$$((A \wedge B) \rightarrow C) \rightarrow (B \rightarrow ((A \wedge B) \rightarrow C)) \quad (8:II)$$

$$(B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (A \rightarrow (B \rightarrow ((A \wedge B) \rightarrow C))) \quad (9:II)$$

$$((A \wedge B) \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow ((A \wedge B) \rightarrow C))) \quad (10:HS, 11:MP \text{ from 8 and 9})$$

$$(A \rightarrow (B \rightarrow ((A \wedge B) \rightarrow C))) \rightarrow ((A \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C))) \rightarrow (A \rightarrow (B \rightarrow C))) \quad (12:ISD)$$

$$((A \wedge B) \rightarrow C) \rightarrow ((A \rightarrow ((B \rightarrow ((A \wedge B) \rightarrow C)) \rightarrow (B \rightarrow C))) \rightarrow (A \rightarrow (B \rightarrow C))) \quad (13:HS, 14:MP \text{ from 11 and 12})$$

$$((A \wedge B) \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C)) \quad (15:ISD, 16:MP \text{ from 7 and 14})$$

This conjoining and un-conjoining of assumptions lets us reason from theories as big conjunctions.

Note the conjunctions here are commutative (let's call this Theorem 3):

$$((A \wedge B) \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C)) \quad (1:\text{Theorem 2})$$

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((B \wedge A) \rightarrow C) \quad (2:\text{Theorem 1})$$

$$((A \wedge B) \rightarrow C) \rightarrow ((B \wedge A) \rightarrow C) \quad (3:HS, 4:MP \text{ from 1 and 2})$$

Defining meaning by constructed proof is **proof theory** (Hilbert & Bernays, 1934; Gentzen, 1935).

It beats classical logic and model theory because it allows reasoning without complete world states.

2.7 Extra: intuitionistic logic theorems useful for defining grammar

We can also define a weaker version of ISD if conditions don't match (let's call this Theorem 4):

$$((A \wedge B) \rightarrow C) \rightarrow (((A \wedge B) \rightarrow (C \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow D)) \quad (1:ISD)$$

$$(A \wedge B) \rightarrow A \quad (2:CE)$$

$$((A \wedge B) \rightarrow A) \rightarrow ((A \rightarrow C) \rightarrow ((A \wedge B) \rightarrow C)) \quad (3:HS)$$

$$(A \rightarrow C) \rightarrow ((A \wedge B) \rightarrow C) \quad (4:MP \text{ from 2 and 3})$$

$$((A \rightarrow C) \wedge ((B \wedge C) \rightarrow D)) \rightarrow (A \rightarrow C) \quad (5:CE)$$

$$((A \rightarrow C) \wedge ((B \wedge C) \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow C) \quad (6:HS, 7:MP \text{ from 5 and 4})$$

$$((A \rightarrow C) \wedge (((B \wedge C) \rightarrow D) \rightarrow (((A \wedge B) \rightarrow (C \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow D))) \rightarrow ((A \wedge B) \rightarrow D) \quad (8:HS, 9:MP \text{ from 7 and 1})$$

$$(A \wedge B) \rightarrow B \quad (10:CE)$$

$$((A \wedge B) \rightarrow B) \rightarrow ((B \rightarrow (C \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow (C \rightarrow D))) \quad (11:HS)$$

$$\begin{aligned}
&(B \rightarrow (C \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow (C \rightarrow D)) && (12: \text{MP from 11 and 10}) \\
&((B \wedge C) \rightarrow D) \rightarrow (B \rightarrow (C \rightarrow D)) && (13: \text{Theorem 2}) \\
&((B \wedge C) \rightarrow D) \rightarrow ((A \wedge B) \rightarrow (C \rightarrow D)) && (14: \text{HS}, 15: \text{MP from 13 and 12}) \\
&((A \rightarrow C) \wedge ((B \wedge C) \rightarrow D)) \rightarrow ((B \wedge C) \rightarrow D) && (16: \text{CE}) \\
&((A \rightarrow C) \wedge ((B \wedge C) \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow (C \rightarrow D)) && (17: \text{HS}, 18: \text{MP from 16 and 15}) \\
&((A \rightarrow C) \wedge ((B \wedge C) \rightarrow D)) \rightarrow ((A \wedge B) \rightarrow D) && (19: \text{IIS}, 20: \text{MP from 9 and 18}) \\
&(A \rightarrow C) \rightarrow (((B \wedge C) \rightarrow D) \rightarrow ((A \wedge B) \rightarrow D)) && (18: \text{Theorem 2})
\end{aligned}$$

We can then adapt this to arguments that are implications (let's call this Theorem 5):

$$\begin{aligned}
&(A \rightarrow (C \rightarrow D)) \rightarrow (((B \wedge (C \rightarrow D)) \rightarrow E) \rightarrow ((A \wedge B) \rightarrow E)) && (1: \text{Theorem 4}) \\
&(A \rightarrow (C \rightarrow D)) \rightarrow ((A \wedge C) \rightarrow D) && (2: \text{Theorem 2}) \\
&((A \wedge C) \rightarrow D) \rightarrow (((B \wedge (C \rightarrow D)) \rightarrow E) \rightarrow ((A \wedge B) \rightarrow E)) && (3: \text{HS}, 4: \text{MP from 2 and 1})
\end{aligned}$$

We can then adapt this to arguments that are medial in conditions (let's call this Theorem 6):

$$\begin{aligned}
&((A \wedge D) \rightarrow E) \rightarrow (((B \wedge C \wedge (D \rightarrow E)) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F)) && (1: \text{Theorem 5}) \\
&((C \wedge (D \rightarrow E) \wedge B) \rightarrow F) \rightarrow ((B \wedge C \wedge (D \rightarrow E)) \rightarrow F) && (2: \text{Theorem 3}) \\
&(((C \wedge (D \rightarrow E) \wedge B) \rightarrow F) \rightarrow ((B \wedge C \wedge (D \rightarrow E)) \rightarrow F)) \rightarrow \\
&\quad (((B \wedge C \wedge (D \rightarrow E)) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F)) \rightarrow \\
&\quad (((C \wedge (D \rightarrow E) \wedge B) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F))) && (3: \text{HS}) \\
&(((B \wedge C \wedge (D \rightarrow E)) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F)) \rightarrow \\
&\quad (((C \wedge (D \rightarrow E) \wedge B) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F)) && (4: \text{MP from 3 and 2}) \\
&((A \wedge D) \rightarrow E) \rightarrow (((C \wedge (D \rightarrow E) \wedge B) \rightarrow F) \rightarrow ((A \wedge B \wedge C) \rightarrow F)) && (5: \text{HS}, 6: \text{MP from 1 and 4})
\end{aligned}$$

We can also define a conjunctive version of ISD (let's call this Theorem 7):

$$\begin{aligned}
&((C \wedge A) \rightarrow B) \rightarrow (C \rightarrow (A \rightarrow B)) && (1: \text{Theorem 2}) \\
&((C \rightarrow (A \rightarrow B)) \rightarrow (C \rightarrow B)) \rightarrow (((C \wedge A) \rightarrow B) \rightarrow (C \rightarrow B)) && (2: \text{HS}, 3: \text{MP from 1}) \\
&(C \rightarrow A) \rightarrow ((C \rightarrow (A \rightarrow B)) \rightarrow (C \rightarrow B)) && (4: \text{ISD}) \\
&(C \rightarrow A) \rightarrow (((C \wedge A) \rightarrow B) \rightarrow (C \rightarrow B)) && (5: \text{HS}, 6: \text{MP from 4 and 3})
\end{aligned}$$

We can also define a weakening rule to add empty arguments (let's call this Theorem 8):

$$\begin{aligned}
&(A \rightarrow B) \rightarrow (C \rightarrow (A \rightarrow B)) && (1: \text{II}) \\
&(C \rightarrow (A \rightarrow B)) \rightarrow ((A \wedge C) \rightarrow B) && (2: \text{Theorem 1}) \\
&(A \rightarrow B) \rightarrow ((A \wedge C) \rightarrow B) && (3: \text{HS}, 4: \text{MP from 1 and 2}) \\
&((A \wedge C) \rightarrow B) \rightarrow ((C \wedge A) \rightarrow B) && (5: \text{Theorem 3}) \\
&(A \rightarrow B) \rightarrow ((C \wedge A) \rightarrow B) && (6: \text{HS}, 7: \text{MP from 4 and 5})
\end{aligned}$$

We can also define an identity theorem (let's call this Theorem 9):

$$(A \rightarrow (B \rightarrow A)) \rightarrow ((A \rightarrow ((B \rightarrow A) \rightarrow A)) \rightarrow (A \rightarrow A)) \quad (1: \text{ISD})$$

$$\begin{array}{ll}
(A \rightarrow (B \rightarrow A)) & (2:II) \\
(A \rightarrow ((B \rightarrow A) \rightarrow A)) & (3:II) \\
(A \rightarrow A) & (4:MP \text{ from 1 and 2}, 5:MP \text{ from 4 and 3})
\end{array}$$

We can then extend this identity to conditionals (let's call this Theorem 10):

$$\begin{array}{ll}
(A \rightarrow B) \rightarrow (A \rightarrow B) & (1:\text{Theorem 9}) \\
(A \wedge (A \rightarrow B)) \rightarrow B & (2:\text{Theorem 2}, 3:MP \text{ from 2 and 1}) \\
((A \rightarrow B) \wedge A) \rightarrow B & (4:\text{Theorem 3}, 5:MP \text{ from 4 and 3})
\end{array}$$

Finally, we can define a conditional modus ponens for type raising (let's call this Theorem 11):

$$\begin{array}{ll}
((A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))) & (1:HS) \\
((B \rightarrow C) \rightarrow (A \rightarrow C)) \rightarrow ((A \wedge (B \rightarrow C)) \rightarrow C) & (2:\text{Theorem 1}) \\
((A \rightarrow B) \rightarrow ((A \wedge (B \rightarrow C)) \rightarrow C)) & (3:HS, 4:MP \text{ from 1 and 2})
\end{array}$$

We'll use these later to define a grammar.

2.8 Arithmetic (Peano, 1889; Heyting, 1930; Kleene, 1952)

One of the most fundamental things we can reason about is **cardinality**: the sizes of sets of entities.

For this we need inline operators related to equality, and axiom schemas for them:

10. **equality** $\begin{array}{c} \alpha \\ \triangle \\ \chi \end{array} \begin{array}{c} t \\ \diagup \quad \diagdown \\ \alpha \quad \alpha \\ = \end{array} \begin{array}{c} \alpha \\ \triangle \\ \omega \end{array}$ of non-functions χ and ω of type α , true (only) if χ and ω are equal:

$$\chi = \chi \quad (\chi = \omega) \rightarrow (\sigma \chi \rightarrow \sigma \omega) \quad \text{for all } \sigma \text{ of type } \alpha \rightarrow t$$

11. **inequality** $\begin{array}{c} \alpha \\ \triangle \\ \chi \end{array} \begin{array}{c} t \\ \diagup \quad \diagdown \\ \alpha \quad \alpha \\ \neq \end{array} \begin{array}{c} \alpha \\ \triangle \\ \omega \end{array}$ of non-functions χ, ω of type α , true (only) if χ and ω are not equal:

$$(\chi \neq \omega) \rightarrow \neg(\chi = \omega) \quad \neg(\chi = \omega) \rightarrow (\chi \neq \omega)$$

We also need a successor function **Suc** of type $(e) \rightarrow e$ and axiom schemas for numbers μ and ν :

$$(\text{Suc } \mu = \text{Suc } \nu) \rightarrow (\mu = \nu) \quad \neg(\text{Suc } \nu = 0)$$

This defines numbers as chains of successor functions, e.g.: $1 = \text{Suc } 0$, $2 = \text{Suc } (\text{Suc } 0)$, ...

We also need some inline arithmetic operators for (rational) numbers and axiom schemas for them:

12. **addition** $\begin{array}{c} \text{e} \\ \swarrow \quad \downarrow \quad \searrow \\ \text{e} \quad + \quad \text{e} \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \\ \triangle \quad \triangle \\ \mu \quad \nu \end{array}$ of numbers μ and ν of type e , yielding the sum of μ and ν :

$$\nu + 0 = \nu \quad \mu + \text{Suc } \nu = \text{Suc } (\mu + \nu) \quad \frac{\mu}{\delta} + \frac{\nu}{\epsilon} = \frac{(\mu \cdot \epsilon) + (\nu \cdot \delta)}{\delta \cdot \epsilon}$$

13. **multiplication** $\begin{array}{c} \text{e} \\ \swarrow \quad \downarrow \quad \searrow \\ \text{e} \quad \cdot \quad \text{e} \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \\ \triangle \quad \triangle \\ \mu \quad \nu \end{array}$ of numbers μ and ν of type e , yielding the product of μ and ν :

$$\nu \cdot 0 = 0 \quad \mu \cdot \text{Suc } \nu = \mu + (\mu \cdot \nu) \quad \frac{\mu}{\delta} \cdot \frac{\nu}{\epsilon} = \frac{\mu \cdot \nu}{\delta \cdot \epsilon}$$

14. **total order** $\begin{array}{c} \text{t} \\ \swarrow \quad \downarrow \quad \searrow \\ \text{e} \quad \geq \quad \text{e} \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \\ \triangle \quad \triangle \\ \mu \quad \nu \end{array}$ of numbers μ and ν of type e , true (only) if μ equals ν plus number o :

$$(\mu \geq \nu) \leftrightarrow (\mu = \nu + o) \quad \frac{\mu}{\delta} \geq \frac{\nu}{\epsilon} \leftrightarrow (\mu \cdot \epsilon) \geq (\nu \cdot \delta) \quad \text{where } \varphi \leftrightarrow \psi \text{ means } \varphi \rightarrow \psi \text{ and } \psi \rightarrow \varphi$$

The above operators $\circ$ all accommodate beta reduction: $[\mu \circ \nu]_{\chi \mapsto \omega} = [\mu]_{\chi \mapsto \omega} \circ [\nu]_{\chi \mapsto \omega}$.

The above axioms are called the **Peano axioms** of natural number arithmetic (Peano, 1889).

2.9 Generalized quantifiers (Barwise & Cooper, 1981)

With these basic rules we can define quantifier functions to make general statements over variables.

Generalized quantifiers compare sizes of intersections of restriction ρ and nuclear scope σ sets:

If we don't have all properties of all entities, we can't count set members exactly, so we use bounds:

- **Cardinal** quantifiers, defined recursively for numbers ν of type e , elements χ, ω of type α :

$$\begin{aligned} \text{Count}_{\geq} 1 \rho (\lambda_{\chi:\alpha} \sigma \chi \wedge \text{Count}_{\geq} \nu (\lambda_{\omega:\alpha} \omega \neq \chi \wedge \rho \omega) \sigma) &\leftrightarrow \text{Count}_{\geq} (\text{Suc } \nu) \rho \sigma \\ \text{Count}_{\leq} \nu \rho \sigma \vee \text{Count}_{\geq} 1 \rho (\lambda_{\chi:\alpha} \sigma \chi \wedge \text{Count}_{\leq} \nu (\lambda_{\omega:\alpha} \omega \neq \chi \wedge \rho \omega) \sigma) &\leftrightarrow \text{Count}_{\leq} (\text{Suc } \nu) \rho \sigma \end{aligned}$$

or inferred from cardinality bounds of subgroups:

$$\begin{aligned} \text{Count}_{\geq} \mu (\lambda_{\chi:\alpha} \rho \chi \wedge \tau \chi) \sigma \wedge \text{Count}_{\geq} \nu (\lambda_{\chi:\alpha} \rho \chi \wedge \neg \tau \chi) \sigma &\rightarrow \text{Count}_{\geq} (\mu + \nu) \rho \sigma \\ \text{Count}_{\leq} \mu (\lambda_{\chi:\alpha} \rho \chi \wedge \tau \chi) \sigma \wedge \text{Count}_{\leq} \nu (\lambda_{\chi:\alpha} \rho \chi \wedge \neg \tau \chi) \sigma &\rightarrow \text{Count}_{\leq} (\mu + \nu) \rho \sigma \end{aligned}$$

where μ, ν are numbers of type e ; ρ, σ, τ are sets of type $(\alpha) \rightarrow \text{t}$; χ, ω are elements of type α .

Minorants (lower bounds) can also use predicates over entity constants as a base case:

$$\rho \chi \rightarrow \text{Count}_{\geq 1} \rho (\lambda_{\chi:\alpha} \text{True}) \quad (\text{EI})$$

Base cases for these bounds have analogues in first-order logic:

$$\begin{aligned} \text{Count}_{\geq 1} \rho \sigma &= \exists_{\chi:\alpha} (\rho \chi \wedge \sigma \chi), \\ \text{Count}_{\leq 0} \rho \sigma &= \forall_{\chi:\alpha} (\rho \chi \rightarrow \neg \sigma \chi). \end{aligned}$$

- **Proportional** quantifiers, in terms of **Count**, for all numbers μ, ν and elements χ of type α :

$$\begin{aligned} \text{Count}_{\geq 1} (\lambda_{\mu:e} \text{Count}_{\leq \mu} (\lambda_{\chi:\alpha} \text{True}) \rho) (\lambda_{\mu:e} \text{Count}_{\geq (\mu \cdot \nu)} \rho \sigma) &\leftrightarrow \text{Ratio}_{\geq} \nu \rho \sigma \\ \text{Count}_{\geq 1} (\lambda_{\mu:e} \text{Count}_{\geq \mu} (\lambda_{\chi:\alpha} \text{True}) \rho) (\lambda_{\mu:e} \text{Count}_{\leq (\mu \cdot \nu)} \rho \sigma) &\leftrightarrow \text{Ratio}_{\leq} \nu \rho \sigma \end{aligned}$$

Other thresholds can also be derived, for quantifiers κ , numbers ν and (large) μ , and sets ρ and σ :

- $\kappa_{>} \nu \rho \sigma \leftrightarrow \kappa_{\geq} (\nu \cdot \frac{\mu+1}{\mu}) \rho \sigma$
- $\kappa_{<} \nu \rho \sigma \leftrightarrow \kappa_{\leq} (\nu \cdot \frac{\mu-1}{\mu}) \rho \sigma$
- $\kappa_{=} \nu \rho \sigma \leftrightarrow \kappa_{\geq} \nu \rho \sigma \wedge \kappa_{\leq} \nu \rho \sigma$

So, for the theory **Prof Kim $\wedge$ Adult Pat**, we can infer:

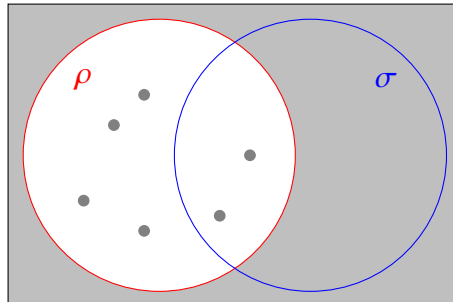
$$\text{Prof Kim} \Rightarrow \text{Count}_{\geq 1} (\lambda_{x:e} \text{Prof } x) (\lambda_{x:e} \text{True})$$

Practice 2.5:

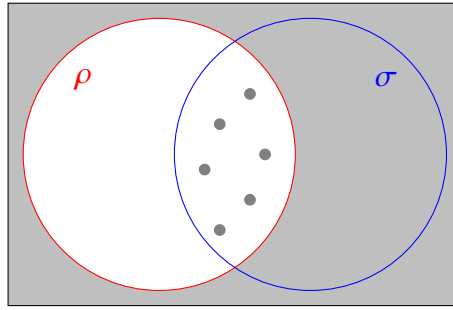
Use the above entailments to obtain a lower bound on cardinality for **Adult** given the theory **Adult Kim $\wedge$ Adult Pat**. Identify the entailment rules you used.

Some examples of generalized quantifiers with fixed thresholds:

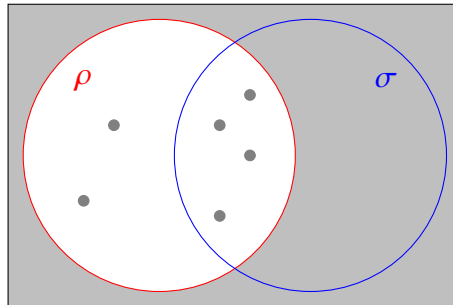
- **Some** $\rho \sigma \leftrightarrow \text{Count}_{\geq 1} \rho \sigma$ — true if provably some of set ρ are in set σ :



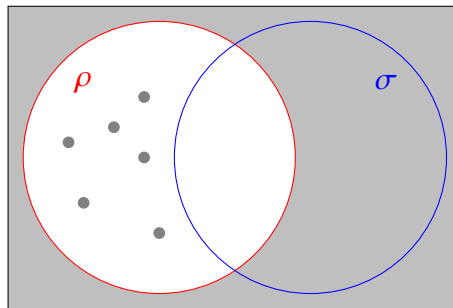
- $All\ \rho\ \sigma \leftrightarrow Ratio_{\geq 1.0}\ \rho\ \sigma$ — true if provably all of set ρ are in set σ :



- $Most\ \rho\ \sigma \leftrightarrow Ratio_{> 0.5}\ \rho\ \sigma$ — true if provably most of set ρ are in set σ :



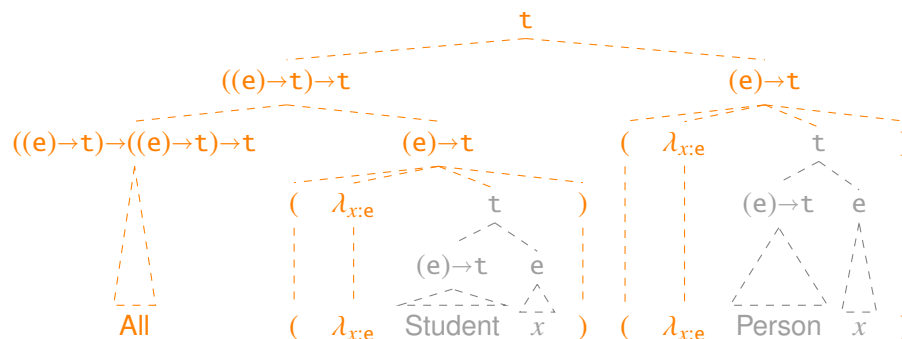
- $None\ \rho\ \sigma \leftrightarrow Count_{\leq 0}\ \rho\ \sigma$ — true if provably none of set ρ are in set σ :



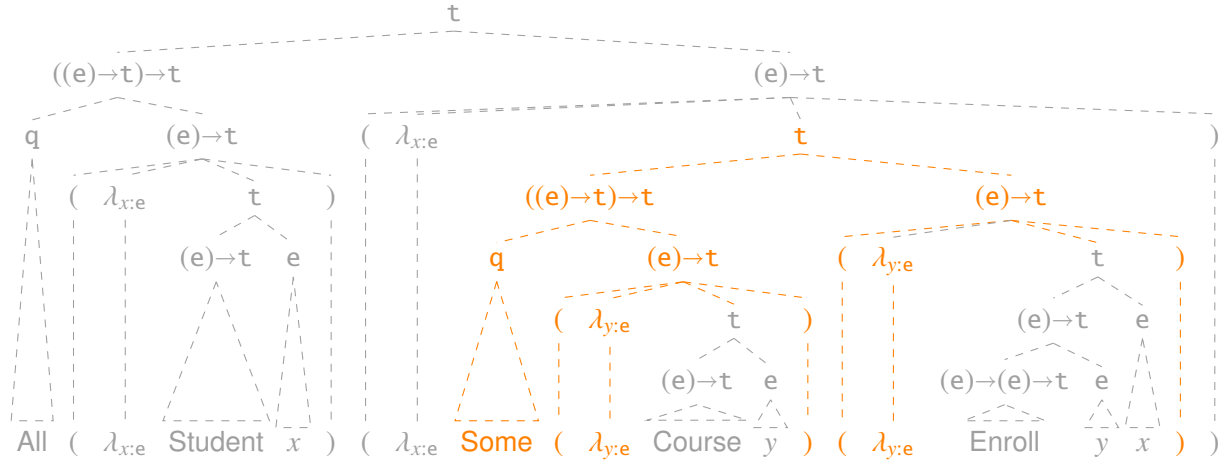
Note the similarity to diagrams of conditional probabilities from the earlier lecture notes.

Generalized quantifiers represent conditional probabilities $P(\sigma|\rho)$, so we use them for reasoning!

Generalized quantifiers with fixed thresholds are just functions of type $((e) \rightarrow t) \rightarrow ((e) \rightarrow t) \rightarrow t$:

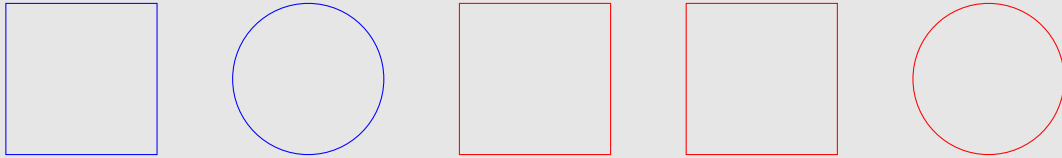


These can be **scoped** or nested (here, to save space, we substitute **q** for $((e) \rightarrow t) \rightarrow ((e) \rightarrow t) \rightarrow t$):



Practice 2.6:

Given this theory of **Shape** entities (where **Red** and **Square** have their usual meanings):



can we prove the following lambda calculus expression (note you are not asked for the proof)?

$$\text{Most } (\lambda_{x:e} \text{ Shape } x \wedge \text{Red } x) (\lambda_{x:e} \text{ Square } x)$$

Practice 2.7:

Given the same set of shapes above, can we prove the following lambda calculus expression?

$$\text{Most } (\lambda_{x:e} \text{ Shape } x) (\lambda_{x:e} \text{ Square } x \wedge \text{Red } x)$$

We also have some axioms for quantifiers where ρ and σ are sets and ω is an entity constant:

- $\rho \omega \rightarrow \text{Count}_{\geq 1} \rho (\lambda_{x:\alpha} \text{ True})$

Existential Introduction (EI)

- **Ratio**_{≥ 1.0} $\rho \sigma \rightarrow (\rho \omega \rightarrow \sigma \omega)$ **Universal Elimination (UE)**

With modus ponens, we can then derive an entailment on the universal quantifier **All**:

- **universal modus ponens (UMP)** — for sets ρ, σ of type $(\alpha) \rightarrow \mathbf{t}$ and elements χ of type α :

$$\text{All } \rho \sigma, \rho \chi \Rightarrow \sigma \chi$$

(this has two premises: the first is the **major premise** and the second is the **minor premise**)

We can also derive entailments for quantifiers κ , numbers ν, μ , sets ρ, σ, τ , and elements χ :

- $(\sigma \chi \rightarrow \tau \chi) \rightarrow (\kappa_{\geq} \nu \rho \sigma \rightarrow \kappa_{\geq} \nu \rho \tau)$ **(right upward entailment, RU)**
- $(\sigma \chi \rightarrow \tau \chi) \rightarrow (\kappa_{\leq} \nu \rho \tau \rightarrow \kappa_{\leq} \nu \rho \sigma)$ **(right downward entailment, RD)**
- $(\rho \chi \rightarrow \tau \chi) \rightarrow (\text{Count}_{\geq} \nu \rho \sigma \rightarrow \text{Count}_{\geq} \nu \tau \sigma)$ **(left upward entailment, LU)**
- $(\rho \chi \rightarrow \tau \chi) \rightarrow (\text{Count}_{\leq} \nu \tau \sigma \rightarrow \text{Count}_{\leq} \nu \rho \sigma)$ **(left downward entailment, LD)**
- $(\rho \chi \rightarrow \tau \chi) \rightarrow (\text{All } \tau \sigma \rightarrow \text{All } \rho \sigma)$ **(left downward entailment, LD)**
- $(\mu \geq \nu) \rightarrow (\kappa_{\geq} \mu \rho \sigma \rightarrow \kappa_{\geq} \nu \rho \sigma)$
- $(\mu \geq \nu) \rightarrow (\kappa_{\leq} \nu \rho \sigma \rightarrow \kappa_{\leq} \mu \rho \sigma)$

For example:

- $\text{Count}_{\geq 2} \text{Red Square} \rightarrow \text{Count}_{\geq 2} \text{Red Shape}$
- $\text{Count}_{\geq 2} \text{Red Square} \rightarrow \text{Count}_{\geq 1} \text{Red Square}$

These entailments are called **natural logic** (van Benthem, 1986).

2.10 Example: grid-world navigation

Now we can make a plan and a theory that moves a person toward a hill in a grid world:

- a **plan** π to move **Person43** to **Hill6** (assuming the following predicates:
 - **CurrentTime** t is true for the most recent time point t in an **AEU** evaluation;
 - **PrecedeOrEqual** $s t$ is true if time s precedes or is equal to t ;
 - **TryMoveToward** $t a x$ is true if a tries to move toward x at time t ;

where ‘All’s iterate over entities, ‘None’s give conditions):

$$\begin{aligned} &\text{All } (\lambda_t \text{CurrentTime } t \wedge \\ &\quad \text{None } (\lambda_s \text{PrecedeOrEqual } 0 s \wedge \text{PrecedeOrEqual } s t) \\ &\quad (\lambda_s \text{At } s \text{Person43 Hill6})) \\ &\quad (\lambda_t \text{TryMoveToward } t \text{Person43 Hill6}) \end{aligned}$$

(Here time ‘0’ is when the plan is created – I have to reach the goal *after* that to succeed.)

In a utility function, this gives us our **actions** δ – in this case: **TryMoveToward** predicates.

- a **theory** θ that moving through mud may fail (assuming the following predicates:
 - **Adjacent** $x y$ is true if grid squares x and y are adjacent;
 - **Aligned** $x y z$ is true if a grid square y lies on a line from x to z ;
 - **Muddy** y and **Clear** y are true if grid square y is muddy or clear, respectively;
 - **PossibleSuccessor** $t u$ is true if time u immediately follows t with some probability;
 where the ‘**Ratio= 0.5**’s provide the probability cost):

All $(\lambda_{t,a,z}$ **TryMoveToward** $t a z$)
 $(\lambda_{t,a,z}$ All $(\lambda_x$ **At** $t a x$)
 $(\lambda_x$ All $(\lambda_y$ **Adjacent** $x y \wedge$ **Aligned** $x y z \wedge$ **Muddy** y)
 $(\lambda_y$ **Ratio= 0.5** $(\lambda_u$ **PossibleSuccessor** $t u$) $(\lambda_u$ **At** $u a y$) $\wedge$
 $\text{Ratio= 0.5 } (\lambda_u$ **PossibleSuccessor** $t u$) $(\lambda_u$ **At** $u a x$))))

and that moving through clear terrain always succeeds:

All $(\lambda_{t,a,z}$ **TryMoveToward** $t a z$)
 $(\lambda_{t,a,z}$ All $(\lambda_x$ **At** $t a x$)
 $(\lambda_x$ All $(\lambda_y$ **Adjacent** $x y \wedge$ **Aligned** $x y z \wedge$ **Clear** y)
 $(\lambda_y$ All $(\lambda_u$ **PossibleSuccessor** $t u$) $(\lambda_u$ **At** $u a y$))))

In a utility function, this gives us our **outcome events** ω – in this case: **At** predicates.

Using these plans and theories in an expected utility function defines a rational decision process. Since they are *simple* and *work*, they are in some sense a ‘natural’ representation of complex ideas. We’ll therefore use these expressions as the complex ideas that get communicated using language.

But note these look very different from how we might represent these ideas in natural language.

2.11 Intensions (Carnap, 1947; Pollard, 2007)

If you want something that’s not true in the current theory, you don’t just want ‘**False**’.

Rather, what you want is for a description of a theory to be true in the current theory.

We therefore define an ‘up’ operator (Montague, 1970) to interpret expressions as descriptions.

15. **intension** $(\uparrow \begin{array}{c} \text{p} \\ \swarrow \quad \searrow \\ \uparrow \quad \text{t} \\ \triangle \\ \varphi \end{array})$ of an expression φ of type t — essentially this is just a type marker.

For example: $(\uparrow \text{Prof Person43})$ of type p , where **Prof Person43** is type t .

Intensions also accommodate beta reduction: $[(\uparrow \varphi)]_{x \mapsto \omega} = (\uparrow [\varphi]_{x \mapsto \omega})$.

We can also model intensions as entities, drawn from a very large domain, similar to infinitesimals.

Entailment predicates can be used to evaluate if a **desired intension** i is in some **class** j :

All $(\lambda_t \text{ CurrentTime } t)$
 $(\lambda_c \text{ All } (\lambda_c \text{ Clerk } c)$
 $(\lambda_a \text{ All } (\lambda_a \text{ Person } a)$
 $(\lambda_x \text{ All } (\lambda_x \text{ Have } t \ c \ x \wedge$
 Some $(\lambda_j \ j = (\uparrow \text{ Count}_{\geq 1} (\lambda_u \text{ ConsecutiveTime } t \ u)$
 $(\lambda_u \text{ Eat } u \ a \ x)))$
 $(\lambda_j \text{ Some } (\lambda_i \text{ Want } t \ a \ i)$
 $(\lambda_i \text{ Entail } i \ j)))$
 $(\lambda_x \text{ Give } t \ c \ a \ x))))$

(If a clerk has something, and someone wants it [perhaps among other things], give it to them.)

Here, even if the intension i that the agent wants contains other conjuncts, it still entails j :

Some $(\lambda_i \ i = (\uparrow \text{ Count}_{\geq 1} (\lambda_u \text{ ConsecutiveTime } 10:00:00 \ u)$
 $(\lambda_u \text{ Eat } u \ \text{Me Apple1} \wedge \text{Drink } u \ \text{Me Juice1})))$
 $(\lambda_i \text{ Want } 10:00:00 \ \text{Me } i)$

So if the above is true, the clerk will recognize that I want to eat an apple and give it to me.

Practice 2.8:

Using the non-intensional and intensional quantifier functions above and the predicates **Kid** k , **Car** c , **Time** t , **Own** $t \ k \ c$, and **Want** $k \ i$, write a lambda calculus expression stating that every kid wants to own a car at some point in time (but they don't have a particular car in mind). Note: only quantifier functions can take lambda functions $(\lambda_x \dots)$ as arguments; all the predicates can only take entity variables as arguments.

2.12 Example: interpreting a question

Now we can reason about what kind of answer a question requires, e.g. using these predicates:

- **CurrentTheory** : $(p) \rightarrow t$, true for some current theory proposition in associative memory
- **CurrentTimeInTheory** : $(p) \rightarrow (e) \rightarrow t$, true for the current time entity in the provided theory
- **ConsecutiveTo** : $(e) \rightarrow (e) \rightarrow t$, true if the second time entity immediately follows the first
- **Admin** : $(e) \rightarrow t$, true for the administrator agent
- **AuthorizedUser** : $(e) \rightarrow t$, true for any authorized user
- **Speaker** : $(e) \rightarrow (e) \rightarrow t$, true for the current speaker at the provided time entity

- **Student** : $(e) \rightarrow t$, true for any student
- **Want** : $(p) \rightarrow (e) \rightarrow (e) \rightarrow t$, true if the first entity wants the proposition at the time entity
- **Believe** : $(p) \rightarrow (e) \rightarrow (e) \rightarrow t$, true if the first entity believes the proposition at the time entity
- **Tell** : $(p) \rightarrow (e) \rightarrow (e) \rightarrow (e) \rightarrow t$, true if the first entity tells the second the proposition at the time

and the following conjunction as a current theory, **Theory1**:

Admin Person1 $\wedge$
 AuthorizedUser Person7 $\wedge$
 Speaker Person7 10:00InTheory1 $\wedge$
 Student Person43 $\wedge$
 Student Person58

and the following conjunction as a top-level theory, **Theory0**, to be combined with other theories:

CurrentTimeInTheory p 10:00In p $\wedge$ (current time in every possible proposition p)
 ConsecutiveTo t In p $(t+1)$ In p $\wedge$ (for every time t in every possible proposition p)
 CurrentTheory ($\uparrow$ Theory1)

If we then add a translation of *Who is a student?*:

All $(\lambda_i$ CurrentTheory i)
 $(\lambda_i$ All $(\lambda_t$ CurrentTimeInTheory i t)
 $(\lambda_t$ All $(\lambda_a$ Speaker a t)
 $(\lambda_a$ All $(\lambda_x$ Student x)
 $(\lambda_x$ All $(\lambda_j$ $j = (\uparrow$ Student $x))$
 $(\lambda_j$ All $(\lambda_k$ $k = (\uparrow$ All $(\lambda_u$ CurrentTimeInTheory k $u)$
 $(\lambda_u$ All $(\lambda_v$ ConsecutiveTo u $v)$
 $(\lambda_v$ Believe j a $v))))$
 $(\lambda_k$ Want k a $t))))$

then we apply universal modus ponens (UMP) using this translation as the major premise:

$\overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{minor premise}}$
 $\Rightarrow$ All $(\lambda_i$ CurrentTheory i)
 $(\lambda_i$ All $(\lambda_t$ CurrentTimeInTheory t i) $(\lambda_t \dots)$),
 CurrentTheory ($\uparrow$ Theory1) (CE)
 $=$ All $(\lambda_i$ CurrentTheory i)
 $(\lambda_i$ All $(\lambda_t$ CurrentTimeInTheory t i) $(\lambda_t \dots)$),
 $(\lambda_i$ CurrentTheory i) ($\uparrow$ Theory1) (beta reduction)
 $\Rightarrow (\lambda_i$ All $(\lambda_t$ CurrentTimeInTheory t i) $(\lambda_t \dots)$) ($\uparrow$ Theory1) (UMP)
 $=$ All $(\lambda_t$ CurrentTimeInTheory t ($\uparrow$ Theory1)) $(\lambda_t \dots)$ (beta reduction)

and then repeat using each conclusion as the major premise of another universal modus ponens:

$$\begin{array}{l}
 \overbrace{\text{All } (\lambda_t \text{ CurrentTimeInTheory } t (\uparrow \text{Theory1})) (\lambda_a \text{ All } (\lambda_a \text{ Speaker } a t) (\lambda_a \dots))}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{minor premise}} \\
 \Rightarrow \text{All } (\lambda_t \text{ CurrentTimeInTheory } t (\uparrow \text{Theory1})) (\lambda_a \text{ All } (\lambda_a \text{ Speaker } a t) (\lambda_a \dots)), \\
 \text{CurrentTimeInTheory 10:00InTheory1 } (\uparrow \text{Theory1}) \quad (\text{CE}) \\
 \Rightarrow \text{All } (\lambda_a \text{ Speaker } a \text{ 10:00InTheory1}) (\lambda_a \dots) \quad (\text{UMP})
 \end{array}$$

$$\begin{array}{l}
 \overbrace{\text{All } (\lambda_a \text{ Speaker } a \text{ 10:00InTheory1}) (\lambda_x \text{ All } (\lambda_x \text{ Student } x) (\lambda_x \dots))}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{minor premise}} \\
 \Rightarrow \text{All } (\lambda_a \text{ Speaker } a \text{ 10:00InTheory1}) (\lambda_x \text{ All } (\lambda_x \text{ Student } x) (\lambda_x \dots)), \\
 \text{Speaker Person7 10:00InTheory1} \quad (\text{CE}) \\
 \Rightarrow \text{All } (\lambda_x \text{ Student } x) (\lambda_x \dots) \quad (\text{UMP})
 \end{array}$$

$$\begin{array}{l}
 \overbrace{\text{All } (\lambda_x \text{ Student } x) (\lambda_j \text{ All } (\lambda_j j = (\uparrow \text{Student } x)) (\lambda_j \dots))}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{minor premise}} \\
 \Rightarrow \text{All } (\lambda_x \text{ Student } x) (\lambda_j \text{ All } (\lambda_j j = (\uparrow \text{Student } x)) (\lambda_j \dots)), \\
 \text{Student Person43} \quad (\text{CE}) \\
 \Rightarrow \text{All } (\lambda_j j = (\uparrow \text{Student Person43})) (\lambda_j \dots) \quad (\text{UMP})
 \end{array}$$

$$\begin{array}{l}
 \overbrace{\text{All } (\lambda_j j = (\uparrow \text{Student Person43})) (\lambda_k \text{ All } (\lambda_k k = (\uparrow \dots)) (\lambda_k \dots))}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1}}^{\text{minor premise}} \\
 \Rightarrow \text{All } (\lambda_j j = (\uparrow \text{Student Person43})) (\lambda_k \text{ All } (\lambda_k k = (\uparrow \dots)) (\lambda_k \dots)), \\
 [(\uparrow \text{Theory1b43}) \stackrel{\text{def}}{=} (\uparrow \text{Student Person43})] \quad (\text{CE}) \\
 \Rightarrow \text{All } (\lambda_k k = (\uparrow \dots)) (\lambda_k \dots) \quad (\text{UMP})
 \end{array}$$

$$\begin{array}{l}
 \text{All } (\lambda_k k = (\uparrow \text{All } (\lambda_u \text{ CurrentTimeInTheory } k u) \\
 (\lambda_u \text{ All } (\lambda_v \text{ ConsecutiveTo } u v) \\
 (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{ Person7 } v)))) \\
 (\lambda_k \text{ Want } k \text{ Person7 10:00InTheory1}), \text{Theory0} \wedge \text{Theory1} \\
 \Rightarrow \text{All } (\lambda_k k = (\uparrow \text{All } (\lambda_u \text{ CurrentTimeInTheory } k u) \\
 (\lambda_u \text{ All } (\lambda_v \text{ ConsecutiveTo } u v) \\
 (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{ Person7 } v)))) \\
 (\lambda_k \text{ Want } k \text{ Person7 10:00InTheory1}), \\
 [(\uparrow \text{Theory1a43}) \stackrel{\text{def}}{=} (\uparrow \text{All } (\lambda_u \text{ CurrentTimeInTheory } (\uparrow \text{Theory1a43}) u) \\
 (\lambda_u \text{ All } (\lambda_v \text{ ConsecutiveTo } u v) \\
 (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{ Person7 } v))))] \quad (\text{CE}) \\
 \Rightarrow \text{Want } (\uparrow \text{Theory1a43}) \text{ Person7 10:00InTheory1} \quad (\text{UMP})
 \end{array}$$

(where $\stackrel{\text{def}}{=}$ simply names a proposition) to obtain a revised **Theory1** with **Want** predicates:

... $\wedge$ (everything in the previous **Theory1**)
Want ($\uparrow$ **Theory1a43**) **Person7** 10:00In**Theory1** $\wedge$
Want ($\uparrow$ **Theory1a58**) **Person7** 10:00In**Theory1**

We can now simplify **Theory1a43** using more universal modus ponens (UMP):

$\overbrace{\text{Theory0} \wedge \text{Theory1a43}}^{\text{major premise}}, \overbrace{\text{Theory0} \wedge \text{Theory1a43}}^{\text{minor premise}}$
 $\Rightarrow \text{All } (\lambda_u \text{ CurrentTimeInTheory } (\uparrow \text{Theory1a43}) u)$
 $\quad (\lambda_u \text{ All } (\lambda_v \text{ ConsecutiveTo } u v)$
 $\quad \quad (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{Person7 } v)),$
 $\quad \text{CurrentTimeInTheory } (\uparrow \text{Theory1a43}) \text{10:00InTheory1a43} \quad \text{(CE)}$
 $\Rightarrow \text{All } (\lambda_v \text{ ConsecutiveTo } \text{10:00InTheory1a43 } v) \quad \text{(UMP)}$
 $\quad (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{Person7 } v)$

$\text{All } (\lambda_v \text{ ConsecutiveTo } \text{10:00InTheory1a43 } v)$
 $\quad (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{Person7 } v), \text{Theory0} \wedge \text{Theory1a43}$
 $\Rightarrow \text{All } (\lambda_v \text{ ConsecutiveTo } \text{10:00InTheory1a43 } v)$
 $\quad (\lambda_v \text{ Believe } (\uparrow \text{Theory1b43}) \text{Person7 } v),$
 $\quad \text{ConsecutiveTo } \text{10:00InTheory1a43 } \text{10:01InTheory1a43} \quad \text{(CE)}$
 $\Rightarrow \text{Believe } (\uparrow \text{Theory1b43}) \text{Person7 } \text{10:01InTheory1a43} \quad \text{(UMP)}$

then **Theory1a43** is:

Believe ($\uparrow$ **Theory1b43**) **Person7** 10:01In**Theory1a43**

and **Theory1b43** is:

Student **Person43**

and **Theory1a58** and **Theory1b58** are the same, but for **Person58**.

This establishes what, specifically, the authorized user wants to know, given who the students are.

2.13 Example: following a plan to answer

Now we consider a plan π to answer authorized users' queries (prior to advancing the clock):

$\text{All } (\lambda_i \text{ CurrentTheory } i)$
 $\quad (\lambda_i \text{ All } (\lambda_t \text{ CurrentTimeInTheory } i t)$
 $\quad \quad (\lambda_t \text{ All } (\lambda_u \text{ ConsecutiveTo } u t)$
 $\quad \quad \quad (\lambda_u \text{ All } (\lambda_a \text{ Admin } a)$
 $\quad \quad \quad \quad (\lambda_a \text{ All } (\lambda_b \text{ AuthorizedUser } b)$
 $\quad \quad \quad \quad \quad (\lambda_b \text{ All } (\lambda_j \text{ Want } j b u)$
 $\quad \quad \quad \quad \quad \quad (\lambda_j \text{ All } (\lambda_v \text{ CurrentTimeInTheory } j v)$
 $\quad \quad \quad \quad \quad \quad \quad (\lambda_v \text{ All } (\lambda_k \text{ Believe } k b v)$
 $\quad \quad \quad \quad \quad \quad \quad \quad (\lambda_k \text{ Tell } k b a t)))))))))$

then after more universal modus ponens:

$$\begin{aligned}
& \pi, \text{Theory0} \wedge \text{Theory1} \\
& \Rightarrow \text{All } (\lambda_i \text{CurrentTheory } i) \\
& \quad (\lambda_i \text{All } (\lambda_t \text{CurrentTimeInTheory } t \ i) (\lambda_t \dots)), \\
& \quad \text{CurrentTheory } (\uparrow \text{Theory1}) \quad \text{(CE)} \\
& \Rightarrow \text{All } (\lambda_t \text{CurrentTimeInTheory } t \ (\uparrow \text{Theory1})) (\lambda_t \dots) \quad \text{(UMP)}
\end{aligned}$$

$$\begin{aligned}
& \text{All } (\lambda_t \text{CurrentTimeInTheory } t \ (\uparrow \text{Theory1})) \\
& \quad (\lambda_t \text{All } (\lambda_u \text{ConsecutiveTo } t \ u) (\lambda_u \dots)), \text{Theory0} \wedge \text{Theory1} \\
& \Rightarrow \text{All } (\lambda_t \text{CurrentTimeInTheory } t \ (\uparrow \text{Theory1})) \\
& \quad (\lambda_t \text{All } (\lambda_u \text{ConsecutiveTo } t \ u) (\lambda_u \dots)), \\
& \quad \text{CurrentTimeInTheory } 10:00 \text{InTheory1 } (\uparrow \text{Theory1}) \quad \text{(CE)} \\
& \Rightarrow \text{All } (\lambda_t \text{ConsecutiveTo } 10:00 \text{InTheory1 } u) (\lambda_t \dots) \quad \text{(UMP)}
\end{aligned}$$

⋮

$$\begin{aligned}
& \text{All } (\lambda_j \text{Want } j \text{ Person7 } 10:00 \text{InTheory1}) \\
& \quad (\lambda_j \text{All } (\lambda_v \text{CurrentTimeInTheory } v \ j) (\lambda_v \dots)), \text{Theory0} \wedge \text{Theory1} \\
& \Rightarrow \text{All } (\lambda_j \text{Want } j \text{ Person7 } 10:00 \text{InTheory1}) \\
& \quad (\lambda_j \text{All } (\lambda_v \text{CurrentTimeInTheory } v \ j) (\lambda_v \dots)), \\
& \quad \text{Want } (\uparrow \text{Theory1a43}) \text{ Person7 } 10:00 \text{InTheory1} \quad \text{(CE)} \\
& \Rightarrow \text{All } (\lambda_v \text{CurrentTimeInTheory } v \ (\uparrow \text{Theory1a43})) (\lambda_v \dots) \quad \text{(UMP)}
\end{aligned}$$

$$\begin{aligned}
& \text{All } (\lambda_v \text{CurrentTimeInTheory } v \ (\uparrow \text{Theory1a43})) \\
& \quad (\lambda_v \text{All } (\lambda_k \text{Believe } k \text{ Person7 } v) (\lambda_k \dots)), \text{Theory0} \wedge \text{Theory1} \\
& \Rightarrow \text{All } (\lambda_v \text{CurrentTimeInTheory } v \ (\uparrow \text{Theory1a43})) \\
& \quad (\lambda_v \text{All } (\lambda_k \text{Believe } k \text{ Person7 } v) (\lambda_k \dots)), \\
& \quad \text{CurrentTimeInTheory } 10:01 \text{InTheory1a43 } (\uparrow \text{Theory1a43}) \quad \text{(CE)} \\
& \Rightarrow \text{All } (\lambda_k \text{Believe } k \text{ Person7 } 10:01 \text{InTheory1a43}) (\lambda_k \dots) \quad \text{(UMP)}
\end{aligned}$$

$$\begin{aligned}
& \text{All } (\lambda_k \text{Believe } k \text{ Person7 } 10:01 \text{InTheory1a43}) \\
& \quad (\lambda_k \text{Tell } k \text{ Person7 Person1 } 10:01 \text{InTheory1}), \text{Theory0} \wedge \text{Theory1} \\
& \Rightarrow \text{All } (\lambda_k \text{Believe } k \text{ Person7 } 10:01 \text{InTheory1a43}) \\
& \quad (\lambda_k \text{Tell } k \text{ Person7 Person1 } 10:01 \text{InTheory1}), \\
& \quad \text{Believe } (\uparrow \text{Theory1b43}) \text{ Person7 } 10:01 \text{InTheory1a43} \quad \text{(CE)} \\
& \Rightarrow \text{Tell } (\uparrow \text{Theory1b43}) \text{ Person7 Person1 } 10:01 \text{InTheory1} \quad \text{(UMP)}
\end{aligned}$$

we obtain a revised **Theory1** with **Tell** predicates:

... $\wedge$ (everything in the previous **Theory1**)
Tell ($\uparrow$ **Theory1b43**) **Person7** **Person1** 10:01In**Theory1** $\wedge$
Tell ($\uparrow$ **Theory1b58**) **Person7** **Person1** 10:01In**Theory1**

This conclusion is for the administrator to tell the user that **Person43** and **Person58** are students. It's a logical consequence of the current theory, the user's stated desire and the administrator plan.

References

- Barwise, J. & Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4.
- Brouwer, L. E. J. (1907). *Over de grondslagen der wiskunde*. PhD thesis, University of Amsterdam, Amsterdam.
- Carnap, R. (1947). *Meaning and Necessity: A Study in Semantics and Modal Logic*. Chicago: University of Chicago Press.
- Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2), 56–68.
- Gentzen, G. (1935). Untersuchungen über das logische schlieSSen. *Mathematische Zeitschrift*, 39, 176–210, 405–431.
- Heyting, A. (1930). Die formalen regeln der intuitionistischen logik I. *Sitzungsberichte der Preussischen Akademie der Wissenschaften. Physikalisch-mathematische Klasse*, (pp. 42–56).
- Hilbert, D. & Bernays, P. (1934). *Grundlagen der Mathematik*, volume 50 of *Grundlagen der mathematischen Wissenschaften*. Berlin, Heidelberg: Springer.
- Kleene, S. C. (1952). *Introduction to Metamathematics*. Princeton, NJ: D. Van Nostrand Company.
- Montague, R. (1970). Pragmatics and intensional logic. *Dialectica*, 24(4), 277–302.
- Peano, G. (1889). *Arithmetices principia: nova methodo exposita*. Augustae Taurinorum [Turin]: Fratres Bocca.
- Pollard, C. (2007). Hyperintensions. *Journal of Logic and Computation*, 18(2), 257–282.
- van Benthem, J. (1986). Semantic automata. In *Essays in Logical Semantics*. Dordrecht, the Netherlands: Kluwer.
- von Neumann, J. & Morgenstern, O. (1944). Theory of games and economic behavior. *Science and Society*, 9(4), 366–369.