Estonian Winter School in Computer Science, 2004

Proof Polynomilas

"For it is far better to know something about everything than to know all about one thing. This universality is the best."

Blaise Pascal, *Penses*

(Lectures 3-4)

Sergei Artemov
Graduate Center of the City University of New York

Modal logic: time and knowledge

Propositional logic is decidable but too restrictive. First order and higher order logics have unlimited expressive power but are not decidable. Modal logic appeared as an attempt to extend propositional logic by additional connectives preserving certain nice features like decidability.

Minimal format: propositional connectives plus unary connective "modality" $\Box$. Intended readings of new atoms $\Box F$ are
1. Epistemic - existential: "F is known", "F is provable", etc,
2. Temporal - universal: "F holds in all possible situations", "in the future F will always hold", etc.

Usually preserves decidability.

History and Applications

C.I. Lewis (1918): Initiated the modern analysis of modality. He developed the logical systems **S1**-**S5**.

JCC McKinsey (1941): used algebraic methods (Boolean algebras with operators) to prove the decidability of Lewis' **S2** and **S4**.

McKinsey-Tarski (1948): topological semantics $\Box F = \textit{interior}(F)$, provides a mathematical model for intuitionism, logic of approximate measurements, leads to logics for dynamic systems, etc.

Kripke (1959): possible worlds à la Leibniz, by far the most widely used semantics.

Hoare (1969): partial correctness statements $A\{G\}B$ =“if A holds before the execution of G then B holds afterward”, a classic of program verification. Recently Tony Hoare was knighted by the British Queen.

Pratt (1976): logic of programs, $[C]\varphi = \varphi$ holds while C is executed, each $[C]$ is an **S4**-modality. Kripke style semantics where possible worlds are machine states. Stanford University Network = (SUN).

Pnueli (1977): branching temporal logic = logic of concurrency. The language of verification and model checking. Turing award in CS.

Logic of Knowledge: *a core AI topic,* $K_A(\varphi)$ = *“agent* A *knows* φ*”, multiple modalities.* Logical Omniscience Problem: build a logic of knowledge that distinguishes hard and easy problems. Prime factorization example.

Basic systems of modal logic

System **K**:

A1. Propositional axioms and rules

A2. $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$ *(distribution)*

Nec. Necessitation rule: $\dfrac{\vdash F}{\vdash \Box F}$

System **K4** is **K** +

A3. $\Box F \longrightarrow \Box\Box F$ *(positive introspection/transitivity)*

System **S4** is **K4** +

A4. $\Box F \longrightarrow F$ *(reflexivity)*

System **S5** is **S4** +

A5. $\neg\Box F \longrightarrow \Box(\neg\Box F)$ *(negative introspection)*

Some of derivations in **K** (hence in all other modal logics).

Theorem: $\Box$ *and* $\wedge$ *commute*

$$\begin{array}{l}
A \to (B \to A \wedge B) \\
\Box(A \to (B \to A \wedge B)) \\
\Box A \to \Box(B \to A \wedge B) \\
\Box A \to (\Box B \to \Box(A \wedge B)) \\
(\Box A \wedge \Box B) \to \Box(A \wedge B)
\end{array}$$

$$\begin{array}{l}
A \wedge B \to A \\
\Box(A \wedge B \to A) \\
\Box(A \wedge B) \to \Box A \\
\Box(A \wedge B) \to \Box B \\
\Box(A \wedge B) \to (\Box A \wedge \Box B)
\end{array}$$

Theorem: $\Box$ *factors out through* $\vee$:

$$\begin{array}{l}
A \to A \vee B \\
\Box(A \to A \vee B) \\
\Box A \to \Box(A \vee B) \\
\Box B \to \Box(A \vee B) \\
(\Box A \vee \Box B) \to \Box(A \vee B)
\end{array}$$

But not $\Box(A \vee B) \to (\Box A \vee \Box B)$! Consider B to be $\neg A$. Whatever intended reading of modality you take $\Box(A \vee \neg A) \to (\Box A \vee \Box\neg A)$ cannot be valid.

Modality dual to $\Box$: $\Diamond F \;\equiv\; \neg\Box\neg F$.

Intended semantics is derivative from the one for $\Box F$:

if $\Box F$ denotes “F holds in all possible situations”,
then $\Diamond F$ stands for “F holds in at least one possible situation”

(the latter has been usually described as $\Box F$ denotes “F is necessary” and $\Diamond F$ stands for “F is possible”)

Exercise: **S4** $\vdash A \to \Diamond A$ (thus **S4** $\vdash \Box A \to \Diamond A$).
Indeed: **S4** $\vdash \Box\neg A \to \neg A$, **S4** $\vdash \neg\neg A \to \neg\Box\neg A$, **S4** $\vdash A \to \neg\Box\neg A$.

In many respects modal logics behave like normal logical systems. In particular, they are closed under substitution:

If $\Gamma(p) \vdash F(p)$ *then* $\Gamma(p/A) \vdash F(p/A)$ *for any* A

Modal logics admit equivalent substitution:
For **L=K**, **K4**, **S4**, **S5**, *if* **L** $\vdash A \leftrightarrow B$ *then* **L** $\vdash F(p/A) \leftrightarrow F(p/B)$ *for any formula* $F(p)$

NOTE: Deduction Theorem fails for **L=K**, **K4**, **S4**, **S5**. Indeed, in all of those logics $A \vdash \Box A$, by Necessitation, however, none of them derives $A \rightarrow \Box A$. To prove that we need to develop some sort of negative test for **L**, for example, some sort of formal semantics true/false in a certain class of models along with a corresponding *soundness theorem*. Then by showing that F is false we can establish that F is not derivable.

One more example:

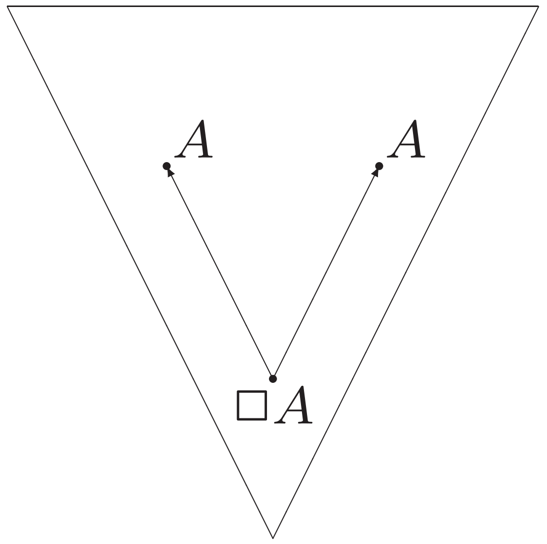
Derivation in **K4**, **S4**, **S5** that $F \rightarrow \Box F$ holds not only for $F \equiv \Box A$ (transitivity axiom), but for $F \equiv \Box A \vee \Box B$ as well.

$\Box A \rightarrow \Box A \vee \Box B$
$\Box(\Box A \rightarrow \Box A \vee \Box B)$
$\Box A \rightarrow \Box\Box A$
$\Box\Box A \rightarrow \Box(\Box A \vee \Box B)$
$\Box A \rightarrow \Box(\Box A \vee \Box B)$
$\Box B \rightarrow \Box A \vee \Box B$
$\Box(\Box B \rightarrow \Box A \vee \Box B)$
$\Box B \rightarrow \Box\Box B$
$\Box\Box B \rightarrow \Box(\Box A \vee \Box B)$
$\Box B \rightarrow \Box(\Box A \vee \Box B)$
$\Box A \vee \Box B \rightarrow \Box(\Box A \vee \Box B)$

Possible Worlds Semantics by Saul Kripke.

Classical logic, propositional and quantified alike, gives a static picture of the world. A classical interpretation (model) is an assignment of truth values to atoms of the language. Modal logic has a striking ability to capture adequately a very natural semantics of "possible worlds" which can be traced back to Leibniz. The possible worlds universe consists of a collection of classical models W connected by a binary accessibility relation $R(a, b)$ "world b is accessible from world a". In other worlds, the possible worlds constitute an ordered graph, not necessarily finite. Whereas classical connectives operate within individual worlds (i.e. nodes in W), modality reaches out to all the worlds accessible from a given one (possible worlds):

$\Box F$ *holds in* a *iff* F *holds in all* b*'s accessible from* a.



Model Kripke is a triple $K = (W, R, \models)$, where W is a nonempty set (elements of which are called "possible worlds"), R a binary relation on W, and $\models$ a truth assignment having form: "world$\models$formula" such that each propositional letter gets some truth value in any world from W. We assume also that for any $x \in W$ both $x \models$ `true` and $x \not\models$ `false`.

The definition of $x \models F$ (read as *a formula F is true in a world x*, or x *forces* F) goes by induction on F:

$x \models A \wedge B$ iff "$x \models A$ and $x \models B$"

$x \models A \vee B$ iff "$x \models A$ or $x \models B$"

$x \models \neg A$ iff "$x \not\models A$"

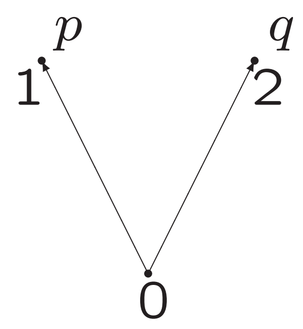
$x \models \Box A$ iff "$y \models A$ for all y such that $R(x, y)$"

By default, we assume that $A \to B$ stands for $\neg A \vee B$, thus imposing the classical truth tables on boolean connectives at every given node. From the definition it is clear that a Kripke model is a collection of classical models connected by some sort of binary "accessibility" relation.
Modality $\Box$ is the only connective able to reach out to other possible worlds, i.e. nodes of the model accessible from a given one.
We may regard $\Box F$ as a sort of restricted universal quantifier "for all possible worlds F holds". It turns out that such limited quantification enables us to express some important features like time and process termination without compromising the decidability of the propositional logic.
$\Diamond F$ holds in x iff F holds in some y accessible from x.

Example

Consider a three-element "V-shaped" model with $W = \{0, 1, 2\}$ given by an oriented graph below. According to this graph, $R(0, 1)$, $R(0, 2)$, and neither of $R(1, 2)$, $R(2, 1)$, $R(1, 0)$, $R(2, 0)$, $R(0, 0)$, $R(1, 1)$, $R(2, 2)$ holds.

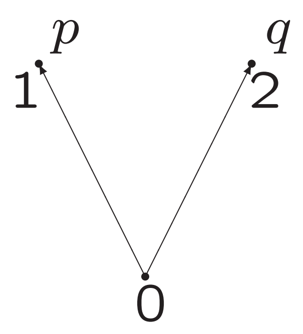
Notational convention: we label the nodes with propositional variables true at a given node. By default, all variables not listed next to a node are assumed false at this node. In particular, $1 \models p$, $2 \models q$, $1 \not\models q$, $2 \not\models p$, $0 \not\models p$, $0 \not\models q$, and all other variables are false at all nodes.



Question: for each of the formulas $\Box p$, $\Box q$, $\Box(p \wedge q)$, $\Box p \wedge \Box q$, $\Box(p \vee q)$, $\Box p \vee \Box q$, list the nodes where this formula is true.

Answer:

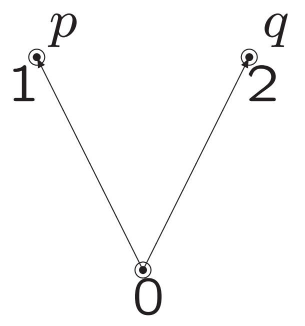
$\Box p$ is true at 1 and 2, but not at 0. Indeed, the set of accessible worlds for either 1 or 2 is empty, thus *FOR ALL* worlds accessible from each of them p holds. $\Box p$ is false at 0, since p fails at 2 which is accessible from 0.
Likewise, $\Box q$ holds at 1 and 2, but not in 0.



Formula $p \wedge q$ is false at every node. Formula $\Box(p \wedge q)$ is true at 1 and 2, but not at 0, so do $\Box p \wedge \Box q$ and $\Box p \vee \Box q$.
Formula $p \vee q$ is true at 1 and 2, but not at 0. Formula $\Box(p \vee q)$ is true at every node. Indeed, it is true at 1 and 2 by trivial reasons (above), hence it also true at 0, since $p \vee q$ is true at every possible world for it.
Note, that $0 \not\models \Box(p \vee q) \rightarrow (\Box p \vee \Box q)$!. Hence we have found a model where this formula fails.

Truth value of a modal formula very much depends upon specific details of accessibility relation.

For example, consider the same model as above, but with all nodes made *reflexive*, i.e. $R(0,0)$, $R(1,1)$, and $R(2,2)$ (we denote reflexive worlds by "circled" nodes, as on the picture). The same formulas now have quite a different meaning.



In particular, $\Box p$ is true at 1, but not at 0 and 2. Likewise, $\Box p$ is true at 2, but not at 0 and 1.
It turned out that each of the modal logics under consideration is complete with respect to a corresponding class of Kripke models which can be characterized by the property of accessibility relation only.

Definition. A formula F is true in a model K (notation: $K \models F$) if F holds at every node of K. A formula F is valid (in a given class of models) if it is true in every model (of this class).

Consolidated Soundness Theorem

- If **K** $\vdash F$ then F is valid in all models.
- If **K4** $\vdash F$ then F is valid in all transitive models.
- If **S4** $\vdash F$ then F is valid in all transitive reflexive models.
- If **S5** $\vdash F$ then F is valid in all transitive reflexive symmetric models .

Proof. A pretty straightforward induction on the length of derivation in a given logic. We first prove that axioms are true in every model. Then we check that rules when applied to formulas true in all models (of a given class) produce a formula true in every such model as well.

Soundness of **K**.

A1. Propositional axioms
are true at every node since each node is a classical model.

A2. $\Box(F \to G) \to (\Box F \to \Box G)$ *(distribution)*
We have to prove that A2 is true at every node x of every model. Suppose $x \models \Box(F \to G)$ and $x \models \Box F$, then for every y accessible from x both $F \to G$ and F hold, hence G does. Since G holds for every y accessible from x, the formula $\Box G$ holds at x.

Modus Ponens: $\dfrac{F \to G, \quad F}{G}$. Obviously holds at each node.

Nec.: $\dfrac{\vdash F}{\vdash \Box F}$ By contrapositive, suppose there is a model K, where $\Box F$ is false at some node x. Then there should be a node y (accessible from x), where F is false. Therefore, F is false in K.

Soundness of **K4**

A3. $\Box F \to \Box\Box F$ *(positive introspection/transitivity)*

Suppose $x \models \Box F$. In order to establish that $x \models \Box\Box F$ consider any y accessible from x and check that $y \models \Box F$. To do this, we have to consider any z accessible from y and prove that $z \models F$. The latter holds since z is also accessible from x (transitivity!), and thus $x \models \Box F$ yields $z \models F$.

Soundness of **S4**

A4. $\Box F \to F$ *(reflexivity)*

Suppose $x \models \Box F$. Then $y \models F$ for all y accessible from x, in particular, for $y = x$. Thus $x \models F$.

Soundness of **S5**

A5. $\neg\Box F \to \Box(\neg\Box F)$ *(negative introspection)*

Suppose $x \models \neg\Box F$, then $y \not\models F$ for some y accessible from x. In order to establish that $x \models \Box\neg\Box F$ consider any z accessible from x and check that $z \models \neg\Box F$. Since accessibility here is symmetric, x is accessible from z. By transitivity, y is also accessible from z. Thus we have found a node y accessible from z and such that $y \not\models F$. Thus $z \models \neg\Box F$.

To show that $p \to \Box p$ is not derivable in modal logic, it now suffices to build a countermodel $K = (W, R, \models)$ for this formula. Consider $W = \{0, 1\}$ and let accessibility be a complete graph on W, i.e. $R(0,0)$, $R(0,1)$, $R(1,0)$, $R(1,1)$. Put $0 \models p$ and $1 \not\models p$. Clearly, K is a legitimate **S5** model, since R is an equivalence relation on W.

Moreover, $0 \models p$, but $0 \not\models \Box p$, since $1 \not\models p$ and 1 is accessible from 0. Therefore, $0 \not\models p \to \Box p$.

By the soundness theorem, **S5** $\nvdash p \to \Box p$, thus none of the other logics **K**, **K4**, **S4** does.

Completeness Theorem

- **K** $\vdash F$ iff F is valid in all models.
- **K4** $\vdash F$ iff F is valid in all transitive models.
- **S4** $\vdash F$ iff F is valid in all transitive reflexive models.
- **S5** $\vdash F$ iff F is valid in all transitive reflexive symmetric models .

Proof. By the maximal consistent sets construction (sometimes called *canonical model*. We will establish completeness of **S4** along with cut-elimination below.

Exercise. Prove that all logics **K**, **K4**, **S4**, **S5** are distinct. Hint: show that each next axiom is not derivable in the previous system, use models.

Genten style proof systems contain the usual classical rules and modal rules for **K**:

$$\frac{\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}$$

for **K4**:

$$\frac{\Gamma, \Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}$$

Genten style proof system for **S4** is called **S4G** and contains the classical rules plus two modal rules

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta} \ (\Box \Rightarrow)$$

and

$$\frac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A} \ (\Rightarrow \Box)$$

where $(\Box\{A_1, \ldots, A_n\} = \{\Box A_1, \ldots, \Box A_n\})$.

By **S4**$^-$ we mean the system **S4** without cut rule.

Consolidated Completeness Theorem for **S4**

The following are equivalent

1. $\mathbf{S4G}^{-} \vdash \Gamma \Rightarrow \Delta$
2. $\mathbf{S4G} \vdash \Gamma \Rightarrow \Delta$
3. $\mathbf{S4} \vdash \wedge\Gamma \Rightarrow \vee\Delta$
4. $\wedge\Gamma \Rightarrow \vee\Delta$ *is true at a every* **S4***-model.*
5. $\wedge\Gamma \Rightarrow \vee\Delta$ *is true at a every finite* **S4***-model.*

Corollary for **S4**: Kripke completeness, cut elimination, equivalence of Gentzen Hilbert systems, finite model property.

Proof. $1 \Rightarrow 2$ and $4 \Rightarrow 5$ are trivial, $2 \Rightarrow 3$ is an easy exercise, $3 \Rightarrow 4$ is the soundness theorem shown above. It suffices now to show $5 \Rightarrow 1$. The latter is proven by the contrapositive: if $\mathbf{S4G}^- \nvdash \Gamma \Rightarrow \Delta$, then $\wedge\Gamma \Rightarrow \vee\Delta$ has a finite **S4**-countermodel.

(The proof is given in class)

Gödel's embedding of **Int** into **S4**:

1. translate **Int**-formula F into a classical language $\Box$:
$$tr(F) = \text{“\textit{box each subformula of} } F\text{”},$$
2. test the translation in **S4**.

Theorem (Gödel (1933), McKinsey & Tarski (1948))

$$\textbf{Int} \textit{ proves } F \quad \Leftrightarrow \quad \textbf{S4} \textit{ proves } tr(F)$$

Proof. Given in class.

The mission of building *BHK* semantics has not been accomplished yet, since **S4** itself still has not been given an exact provability model

$$\textbf{Int} \;\hookrightarrow\; \textbf{S4} \hookrightarrow ? \;\hookrightarrow \textit{REAL PROOFS}$$

Yet another proof system **Natural Deduction**.

Connections to the rest of the world:

1. Gentzen's sequent $\Gamma \Rightarrow F$ is nothing but an input/output record of a Natural derivation of F from hypotheses Γ.

2. Natural derivations in **Int** are isomorphic to typed λ-terms, where propositional formulas are regarded as types (famous Curry-Howard isomorphism).

On the top of that we will observe that objects in the **typed Combinatory Logic CL$_{\rightarrow}$** (combinatory terms) are nothing but proof terms in the Hilbert style proof system. Typed λ-calculus and typed combinatory logic **CL$_{\rightarrow}$** naturally emulate each other the way Hilbert style Natural Deduction proof systems are mutually interpretable.

Proof threes in **IntN** (= Intuitionistic logic, Natural deduction style) are finite oriented threes with nodes labelled by formulas. Formulas from the leaves denote assumptions of this proof tree, the formula at the root node is the one derived by the proof tree. Assumptions from the leaves may be in open (denoted as A) and closed (discharged) forms denoted as $[A]^u$. The latter stands for auxiliary assumptions which were open once but then have been subsumed by the goal formula the way the Deduction Theorem incorporates a hypothesis into the derived formula. Each assumption is labelled by a special marker *proof variable*, usually suppressed for open assumptions. Distinct assumptions must have distinct markers. Assumptions are discharged in groups corresponding to a given marker.

Induction base: The single-node tree labelled by any formula A is a proof tree with the open assumption A; there are no closed assumptions.

Inductive step. We formulate a number of tree constructing **rules** where the premised are roots of already built proof threes.

$$\dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2\\ A & B\end{matrix}}{A \wedge B}\ \wedge\mathsf{I} \qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ A \wedge B\end{matrix}}{A}\ \wedge\mathsf{E}_R \qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ A \wedge B\end{matrix}}{B}\ \wedge\mathsf{E}_L$$

$$\dfrac{\begin{matrix}\mathcal{D}_1\\ A\end{matrix}}{A \vee B}\ \vee\mathsf{I}_R \qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ B\end{matrix}}{A \vee B}\ \vee\mathsf{I}_L \qquad \dfrac{\begin{matrix} & [A]^u & [B]^v\\ \mathcal{D}_1 & \mathcal{D}_2 & \mathcal{D}_3\\ A \vee B & C & C\end{matrix}}{C}\ \vee\mathsf{E}, u, v$$

$$\dfrac{\begin{matrix}[A]^u\\ \mathcal{D}_1\\ B\end{matrix}}{A \to B} \to \mathsf{I}, u \qquad \dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2\\ A \to B & A\end{matrix}}{B} \to\mathsf{E} \qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ \bot\end{matrix}}{A}\ \bot$$

Examples of natural derivations:

$$\dfrac{\dfrac{[A \land B]^u}{A}}{A \land B \to A}\,u \qquad \dfrac{\dfrac{[A]^u}{A \lor B}}{A \to A \lor B}\,u \qquad \dfrac{\dfrac{[A]^u}{B \to A}}{A \to (B \to A)}\,u$$

$$\dfrac{\dfrac{\dfrac{[A]^v \qquad [A \to \bot]^u}{\bot}}{(A \to \bot) \to \bot}\,u}{A \to \neg\neg A}\,v \qquad \dfrac{\dfrac{\dfrac{\dfrac{[A]^v \qquad [A \to \bot]^u}{\bot}}{B}}{(A \to \bot) \to B}\,u}{A \to (\neg A \to B)}\,v$$

More examples:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^u \qquad [A\to B]^w}{B} \qquad [B\to\perp]^v}{\perp}}{\neg A}\,u}{\neg B\to\neg A}\,v}{(A\to B)\to(\neg B\to\neg A)}\,w$$

More:

$$\dfrac{\dfrac{\dfrac{[A]^v \qquad [A\to\bot]^u}{\bot}}{(A\to\bot)\to\bot}\,u \qquad [((A\to\bot)\to\bot)\to\bot]^w}{\dfrac{\dfrac{\bot}{A\to\bot}\,v}{\neg\neg\neg A\to\neg A}\,w}$$

And yet more, with multiple markers:

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \wedge B]^u}{B} \qquad \dfrac{\dfrac{[A \wedge B]^u}{A} \qquad [A \to (B \to C)]^v}{B \to C}}{C}}{(A \wedge B) \to C}\;u}{(A \to (B \to C)) \to ((A \wedge B) \to C)}\;v$$

Finally, something about introducing disjunction:

$$\dfrac{\dfrac{\dfrac{\dfrac{[A\vee B]^w \qquad \dfrac{[A]^u \qquad [A\to C]^y}{C} \qquad \dfrac{[B]^v \qquad [B\to C]^x}{C}}{C}\; u, v}{A\vee B\to C}\; w}{(B\to C)\to(A\vee B\to C)}\; x}{(A\to C)\to((B\to C)\to(A\vee B\to C))}\; y$$

Sequents are input/output snapshots of construction steps of a natural deduction. Translation table (multiple weakenings are suppressed for brevity):

Natural Deduction	Gentzen's derivation
A	$A \Rightarrow A$
$\dfrac{A \quad B}{A \wedge B}$	$\dfrac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}$
$\dfrac{A \wedge B}{A}$	$\dfrac{\Gamma \Rightarrow A \wedge B \qquad \dfrac{A \Rightarrow A}{A \wedge B \Rightarrow A}}{\Gamma \Rightarrow A}\,\text{Cut}$

Natural Deduction | Gentzen's derivation

$$\frac{\bot}{A} \qquad\qquad \frac{\Gamma \Rightarrow \bot \qquad \bot \Rightarrow A}{\Gamma \Rightarrow A}\,\text{Cut}$$

$$\frac{A \to B \qquad A}{B} \qquad\qquad \frac{\Gamma \Rightarrow A \to B \qquad \dfrac{\Gamma \Rightarrow A \qquad B \Rightarrow B}{A \to B, \Gamma \Rightarrow B}}{\Gamma \Rightarrow B}\,\text{Cut}$$

Disjunction cases: exercise!

From Gentzen to Natural Deduction.

An obvious obstacle: how to understand empty succedents (right parts of sequents), since there are no Natural derivations without conclusions.

Lemma. *If sequent* $\Gamma \Rightarrow A$ *is provable in* **IntG**, *then there is a proof which exclusively contains sequents with a single succedents. If* $\Gamma \Rightarrow$ *is provable, then* $\Gamma \Rightarrow A$ *is provable for any* A.

Proof "by example".

Main (and the only) idea: move weakenings up to axioms. If succedents remain empty till the root sequent, plug $\bot$ to all empty succedents:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A \qquad \bot \Rightarrow}{A, \neg A \Rightarrow}}{A, A \wedge \neg A \Rightarrow}}{A \wedge \neg A, A \wedge \neg A \Rightarrow}}{A \wedge \neg A \Rightarrow}}{A \wedge \neg A \Rightarrow B}}{\Rightarrow (A \wedge \neg A) \to B} \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A \qquad \bot \Rightarrow B}{A, \neg A \Rightarrow B}}{A, A \wedge \neg A \Rightarrow B}}{A \wedge \neg A, A \wedge \neg A \Rightarrow B}}{A \wedge \neg A \Rightarrow B}}{A \wedge \neg A \Rightarrow B}}{\Rightarrow (A \wedge \neg A) \to B}$$

From Gentzen to Natural Deduction.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{A \Rightarrow A \qquad B \Rightarrow B}{A \to B, A \Rightarrow B} \qquad A \to B, A, \bot \Rightarrow \bot}{A \to B, A, B \to \bot \Rightarrow \bot}}{A \to B, A \land \neg B, \neg B \Rightarrow \bot}}{A \to B, A \land \neg B, A \land \neg B \Rightarrow \bot}}{A \to B, A \land \neg B \Rightarrow \bot}}{A \to B \Rightarrow \neg(A \land \neg B)}}{\Rightarrow (A \to B) \to \neg(A \land \neg B)}$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{[A \land \neg B]^u}{A} \qquad [A \to B]^v}{B} \qquad \cfrac{[A \land \neg B]^u}{B \to \bot}}{\bot}}{\neg(A \land \neg B)}\, u}{(A \to B) \to \neg(A \land \neg B)}\, v$$

Normalization in Natural Deduction.

Obvious redundancies in Natural derivations are caused by an I rule followed by an E rule in the same position. Here is a complete list of such configurations along with the proposed conversions to eliminate a given redundancy. Here *conv* means “converts to”. A derivation is **normal** if none of those redundancies occurs.

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 \\ A_1\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ A_2\end{matrix}}{A_1 \wedge A_2}}{A_i} \qquad \text{conv} \qquad \begin{matrix}\mathcal{D}_i \\ A_i\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ A_i\end{matrix}}{A_1 \vee A_2} \qquad \begin{matrix}[A_1]^u\\ \mathcal{D}_1\\ C\end{matrix} \qquad \begin{matrix}[A_2]^v\\ \mathcal{D}_2\\ C\end{matrix}}{C} \qquad \text{conv} \qquad \begin{matrix}\mathcal{D}\\ [A_i]\\ \mathcal{D}_i\\ C\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}[A]^u\\ \mathcal{D}\\ B\end{matrix}}{A \to B} \qquad \begin{matrix}\mathcal{D}_1\\ A\end{matrix}}{B} \qquad \text{conv} \qquad \begin{matrix}\mathcal{D}_1\\ A\\ \mathcal{D}\\ B\end{matrix}$$

In a normal Natural derivation each Elimination rule precedes each Introduction rule.

Theorem. *Every Natural Deduction derivation has a normal form.*

Proof. In principle, in this form the claim can be deduced from the normalization theorem for Gentzen proof system. Indeed, there is an algorithm (cf. BPT) of transforming Cut-free Gentzen derivations and translating them to a normal Natural derivations (cf, also "Logic and Structure" by van Dalen). A stronger claim that the reduction system above eliminates ALL redundancies is a bit harder to establish.

Propositions as types interpretation.

Atomic propositions can be read as data types, i.e. *Nat*, Real, *Boolean*, Lists$_T$ over a given type T.

Implications $A \to B$ denote a **functional type**, i.e. the type of all functions from type A to type B. In particular, type

$$(\mathit{Nat} \to \mathit{Nat}) \to \mathit{Nat}$$

stands for functionals over natural numbers, i.e. a function which maps integer functions to natural numbers. Type

$$\mathit{Nat} \to (\mathit{Nat} \to \mathit{Nat})$$

may be interpreted as encoding an integer function of two variables: given integer $x = n$ we get a function $F_n(y)$. Given another integer $y = m$ we get the integer output $F_n(m)$.

Simple Types. To begin talking about data types and functions is sufficed to have "implication" only. Types generated by implications are called **simple types** $\mathcal{T}$. We consider propositional variables $\mathcal{V} = \{p_0, p_1, p_2, \ldots\}$ as *type variables* and define $\mathcal{T}$ by the following grammar:

$$\mathcal{T} = \mathcal{V} \mid \mathcal{T} \to \mathcal{T}$$

A usual reading of this definition is a) each variable is a simple type, b) if A and B are simple types, then $(A \to B)$ is again a simple type. Notational convention: $A_1 \to A_2 \to A_3 \to \ldots \to A_n$ reads

$$(A_1 \to (A_2 \to (A_3 \to \ldots \to A_n)) \ldots)$$

Lambda-terms.

Consider a famous mathematical joke. From the elementary calculus we know that the derivative of x^2 is $2x$, i.e. $(x^2)' = 2x$. Plug in $x = 1$, and get $(1^2)' = 2 \cdot 1$. On the other hand, the derivative of a constant is zero, thus $(1^2)' = (1)' = 0$, hence $2 = 0$ and mathematics is inconsistent. The problem here appears in messing two different meanings of x^2 as a term having type *Real*, and a function of type *Real* $\rightarrow$ *Real*. Students learn this defaults, sometimes with difficulties, and often without a real understanding of the difference. Computers do not understand our defaults and require absolute precision of the language used.

Let us reserve x^2, $2x$ for terms of type *Real*, and $\lambda x.x^2$, $\lambda x.2x$ for functions from x to x^2 and from x to $2x$ on reals, respectively. The above equality should be read as $D[\lambda x.x^2] = \lambda x.2x$, where D denotes a differentiation operator. In this expression variable x is **bound** i.e. **local**, in particular, one cannot substitute a numeric value for x.

Simply typed λ-terms = prototype functional programs.

Notational convention: for *term* t *of type* S we will use both t^S and $t\!:\!S$.

1. Variables $x_0^A, x_1^A, x_2^A, \ldots$ for each simple type A are λ-terms of type A.

2. Given $t^{A\to B}$ and s^A one may construct a new term $(t^{A\to B} \cdot s^A)^B$ (application). An alternative notation here is $(t^{A\to B} s^A)^B$

3. Given a term t^B and a variable x^A there is a new term $(\lambda x^A.t^B)^{A\to B}$ (lambda abstraction). Variable x^A is **bound** in $(\lambda x^A.t^B)^{A\to B}$, other free (not bound) variables of t^B remain free in $(\lambda x^A.t^B)^{A\to B}$.

In general, we shall omit type-indications whenever it is possible to recover them from a context. Notational convention: str should be read as $(st)r$. Term $\lambda x_1(\lambda x_2(\ldots(\lambda x_n.t)\ldots))$ reads as $\lambda x_1 x_2 \ldots x_n.t$. Application binds more strongly than abstraction, thus $\lambda x.st$ is $\lambda x.(st)$ rather than $(\lambda x.s)t$. We do not distinguish λ-terms that differ only in names of bound variables, e.g. $\lambda x^A.y^{A\to B}x^A$ is the same as $\lambda z^A.y^{A\to B}z^A$.

Curry vs. Church typing, applications.

Curry's definition introduces terms without types first and then addresses the problem of recovering types ("implicit typing"). This corresponds to the programming paradigm (cf. **ML**), when a compiler should check whether a type can be assigned to the program. Church suggested "explicit typing", when a program (λ-term) should be written together with its type (**ALGOL 68**, **PASCAL**). The definition above is the Church style one.

Type checking is an important verification (debugging) tool in programming languages.

Curry-Howard isomorphism: a preview.

Simple types = propositional formulas (in the implicational fragment of **Int**).

Typed λ-term of type T = natural style intuitionistic derivation of T.

Natural deduction step	λ-term construction step
A^u	$u : A$
$\dfrac{A \to B \qquad A}{B}$	$\dfrac{s : (A \to B) \qquad t : A}{st : B}$
$\dfrac{\begin{matrix}[A]^u \\ \vdots \\ B\end{matrix}}{A \to B}$	$\dfrac{\begin{matrix}u : A \\ \vdots \\ t(u) : B\end{matrix}}{\lambda u.t(u) : (A \to B)}$

Example on Curry-Howard isomorphism.

Given Natural derivation of $(A \to B) \to (A \to C)$ with an open assumption $A\to(B\to C)$ we build a λ-term of type $(A\to B)\to(A\to C)$ with a free variable $z:(A\to(B\to C))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^x \quad (A\to(B\to C))^z}{B\to C} \qquad \dfrac{[A\to B]^y \quad [A]^x}{B}}{C}}{A\to C}\,x}{(A\to B)\to(A\to C)}\,y$$

$$x:A, \quad y:(A\to B), \quad z:(A\to(B\to C)), \quad zx:(B\to C), \quad yx:B, \quad zx(yx):C$$
$$\lambda x.zx(yx):(A\to C), \quad \lambda yx.zx(yx):((A\to B)\to(A\to C))$$

Implicational fragment of **Int**.

Three equivalent definitions:

1) all implications-only formulas derivable in **Int**,

2) Hilbert style system with *Modus Ponens* and axioms

A1. $A \rightarrow (B \rightarrow A)$

A2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

3) Gentzen style system with rules concerning implication

$$\frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B}\,(\Rightarrow, \rightarrow) \qquad \frac{\Gamma \Rightarrow A \qquad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta}\,(\rightarrow, \Rightarrow)$$

Exercise: prove that those three definitions are equivalent. Prove that for the classical logic those definitions are not equivalent.

Combinatory Logic

So far we have noticed that for intiutionistic logic Natural derivations (essentially equivalent to Gentzen style derivations) has an isomorphic computational copy known as typed λ-terms. Do Hilbert style proofs have a similar computational analogue? The answer to this question is positive: typed **Combinatory Logic CL$_\to$**, which is as old as λ-calculus, provides an isomorphic copy of Hilbert style proofs.

Definition of Combinatory terms.

1. Variables $x_0^A, x_1^A, x_2^A, \dots$ for each simple type A are terms of type A.
2. Constants $\mathbf{k}^{A\to(B\to A)}$ for all types A, B
3. Constants $\mathbf{s}^{(A\to(B\to C))\to((A\to B)\to(A\to C))}$ for all types A, B, C
4. Given $t^{A\to B}$ and s^A one may construct a new term $(t^{A\to B}s^A)^B$ (application).

Set theoretical semantics of types and computational semantics of terms.

Atomic types may be interpreted as sets, function types $A \to B$ as sets of all total functions from A to B (sorry for an enormous cardinality explosion). Not all of those functions are computable, or otherwise efficient. Computable functions are build from a certain initial supply of those by a given set of operations. Combinatory logic over simple types may be regarded as a rudimentary computational model for arbitrary sets. Constants k and s provide simple examples of constructive objects (e.g. functions). Application applies a function to an appropriate input of the matching type and produces again a constructive object. A type is *inhabitant* is it contains a constructive object. Universally inhabitant types are exactly theorems of intuitionistic logic. In specific interpretations additional data can make some other types also inhabitant. Though ridiculously simple, combinatory terms present “almost” universal computational model: if we ignore type (untyped Combinatory Logic) emulates all recursive functions.

Finding Hilbert style derivations and building combinatory terms are similar tasks.

Example "Good old" derivation of $A \to A$.

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ (axiom 2)
2. $A \to ((A \to A) \to A)$ (axiom 1)
3. $(A \to (A \to A)) \to (A \to A)$ (from 1., 2., by MP)
4. $A \to (A \to A)$ (axiom 1)
5. $A \to A$ (from 3. and 4., by MP)

The corresponding combinatory term:

1. **s**:$[(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))]$
2. $\mathbf{k}_1$:$[A \to ((A \to A) \to A)]$
3. $\mathbf{s}\mathbf{k}_1$:$[(A \to (A \to A)) \to (A \to A)]$
4. $\mathbf{k}_2$:$[A \to (A \to A)]$ (axiom 1)
5. $\mathbf{s}\mathbf{k}_1\mathbf{k}_2$:$[A \to A]$

Axioms = combinatory constants
Modus Ponens = application
Open assumptions of a proof = free variables of the term, notation

$$x_1{:}A_1, x_2{:}A_2, \ldots, x_n{:}A_n \vdash t(x_1, x_2, \ldots, x_n){:}B,$$

where block $x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$ is called *basis* or *context* and contains all free variable of $t(x_1, x_2, \ldots, x_n)$. From the programming point of view the context is the list of all inputs of the program $t(x_1, x_2, \ldots, x_n)$ with their types.

Example of a derivation from hypotheses translated to combinatory logic

1. $x{:}(A \to B)$ (a hypothesis)
2. $y{:}(B \to C)$ (a hypothesis)
3. $\mathbf{k}{:}((B \to C) \to (A \to (B \to C)))$ (axiom 1)
4. $\mathbf{k}y{:}(A \to (B \to C))$ (from 2 and 3, by MP)
5. $\mathbf{s}{:}((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$ (axiom 2)
6. $\mathbf{s}(\mathbf{k}y){:}((A \to B) \to (A \to C))$ (from 4 and 5, by MP)
7. $\mathbf{s}(\mathbf{k}y)x{:}(A \to C)$ (from 1 and 6, by MP)

Combinatory Logic and λ-calculus are mutually interpretable.

Hilbert style proof system derives the same formulas as Natural deduction. Likewise, there are algorithms that translate Combinatory terms to λ-terms and vice versa. To translate **CL**$_{\rightarrow}$ to λ-calculus it suffices to build λ-terms corresponding to constants **k and s, which we leave as a useful exercise.**

To translate λ-calculus to **CL**$_{\rightarrow}$ one has to mimic λ-abstraction in **CL**$_{\rightarrow}$. We will put this question in a formal setting later.

Meanwhile both **CL**$_{\rightarrow}$ and λ-calculus are capable of internalizing **Int**-derivations: if $A_1, \ldots, A_n \vdash B$ in **Int** then $x_1{:}A_1, \ldots, x_n{:}A_n \vdash t(x_1, \ldots, x_n){:}B$ in **CL**$_{\rightarrow}$.

Conversions in Combinatory Logic

The canonical set-theoretical model of $\mathbf{CL}_{\rightarrow}$ suggests some further specifications of operations on combinatory terms. Consider term $\mathbf{k}xy$, where, as usual, combinator $\mathbf{k}:(A \rightarrow (B \rightarrow A))$, hence $x:A$ and $y:(B \rightarrow A)$. An easy computation shows that $\mathbf{k}xy{:}A$, i.e. term $\mathbf{k}xy$ denotes some element A. As a matter of a fact, we can specify such an element immediately without even applying any combinators: it is $x{:}A$! This semantical consideration suggests the conversion

$$\mathbf{k}xy \quad \textit{cont} \quad x$$

Here $\mathbf{k}xy$ is called the *redex* of reduction and x a *contractum*. The same conversion is suggested by the proof interpretation of $\mathbf{CL}_{\rightarrow}$: we are looking for a proof $\mathbf{k}xy$ of A, and $x{:}A$ provides such a (possibly, different) proof.

Consider term $\mathbf{s}xyz$, where combinator

$$\mathbf{s}{:}((A \to (B\to C)) \to ((A \to B)\to(A\to C))),$$

hence $x:(A \to (B \to C))$, $y:(A \to B)$, $z:A$. Term $\mathbf{s}xyz:C$ denotes some element of C, and again we can specify such an element without using any combinators. Indeed, $xz(yz){:}C$, which suggests a conversion

$$\mathbf{s}xyz \quad \textit{cont} \quad xz(yz)$$

(redex $\mathbf{s}xyz$, contractum $xz(yz)$).

We say $t \succeq_1 t'$ if t' is obtained from t by replacing one redex by the corresponding contractum. Let also $\succeq$ be the reflexive and transitive closure of $\succeq'$. Finally, the equality "$=$" on combinatory terms is a symmetric and transitive closure of $\succeq$.

Normal forms of combinatory terms.

A term t is *in normal form*, if t does not contain a redex. t has a normal form if there is a normal t' such that $t \succeq t'$. A term is *strongly normalizing*, if its reduction tree is finite, i.e. each reduction sequence is finite.

Theorem. Each term in $\mathbf{CL}_{\rightarrow}$ is strongly normalizing and has a unique normal form.

Proof. A pretty tricky induction, cf. BPT, and Barendregt. Uniqueness of normal forms follows from so-called confluence property, a.k.a. Church-Rosser property: if $t \succeq t_1$ and $t \succeq t_2$, then there is a t_3 such that $t_1 \succeq t_3$ and $t_2 \succeq t_3$. Indeed, assume Church-Rosser property, and suppose t has two normal forms t_1 and t_2, in particular, $t \succeq t_1$ and $t \succeq t_2$. By Church-Rosser, there is t_3 such that $t_1 \succeq t_3$ and $t_2 \succeq t_3$. Since both t_1 and t_2 are normal, t_1 coincides with t_3 and t_2 coincides with t_3, t_1 coincides with t_2.

Conversions and normal forms in simply typed λ-calculus.

The most important reduction in λ-calculus is based on β-conversion:

$$(\lambda x^A.t^B)s^A \quad \mathsf{cont}_\beta \quad t^B[x^A/s^A],$$

where $t^B[x^A/s^A]$ denotes the result of substitution of s^A for x^A in t^B. Naturally substitutions should not allow collisions of variables in s^A and t^B.

Theorem. Each λ-term is strongly normalizing and has a unique normal form.

Proof. Even trickier induction, cf. BPT, and Barendregt. Uniqueness again is established as a corollary of the Church-Rosser property.

Emulated λ-abstraction in $\mathbf{CL}_{\rightarrow}$.

Theorem. To each term t of $\mathbf{CL}_{\rightarrow}$ there is a term $\lambda^* x^A.t$ containing no x^A free such that

$$(\lambda^* x^A.t)s^A \succeq t[x^A/s^A]$$

Proof. We define $\lambda^* x^A.t$ by induction on t:

1. $\lambda^* x^A.x := \mathbf{s}^{(A\rightarrow((A\rightarrow A)\rightarrow A))\rightarrow((A\rightarrow(A\rightarrow A))\rightarrow(A\rightarrow A))}\mathbf{k}_1^{A\rightarrow((A\rightarrow A)\rightarrow A)}\mathbf{k}_2^{A\rightarrow(A\rightarrow A)}$
2. $\lambda^* x^A.y^B := \mathbf{k}^{B\rightarrow(A\rightarrow B)}y^B$, for y different from x
3. $\lambda^* x^A.t_1^{B\rightarrow C}t_2^B := \mathbf{s}^{((A\rightarrow(B\rightarrow C))\rightarrow((A\rightarrow B)\rightarrow(A\rightarrow C)))}(\lambda^* x.t_1)(\lambda^* x.t_2)$.

It is an easy exercise to verify that the stated property holds.

Exercise 19. Which of the following is provable in **S4**? Give a proof, if any. Provide a countermodel otherwise.

a) $\Box(A \to \Box B) \to (A \to \Box\Box B)$
b) $\Box(\Box A \to \Box B) \vee \Box(\Box B \to \Box A)$

Exercise 20.

a) Establish the Disjunctive Property for **S4**: $\vdash \Box A \vee \Box B$ yields $\vdash \Box A$ or $\vdash \Box B$.
b) Does the Disjunctive Property hold for **S5**?

Exercise 21. Prove Gödel's Theorem of 1933: **Int** $\vdash F \quad \Rightarrow \quad$ **S4** $\vdash F^{\Box}$, where $F^{\Box}$ is Goedel's translation consisting in prefixing each occurrence of a subformula in F by $\Box$. Hint: induction on a derivation of F in **Int**. Base case corresponds to the axioms of **Int**, prove that their translations are all derivable in **S4**. The Induction step is MP in **Int**.

Exercise 22. Prove McKinsey-Tarski theorem of 1948: for any propositional formula F if **S4** $\vdash F^{\Box}$ then **Int** $\vdash F$. Hint: do the contrapositive, of course, i.e. if **Int** $\nvdash F$ then **S4** $\nvdash F^{\Box}$. Let **Int** does not prove F. Then there is an intuitionistic countermodel Kripke $K = (W, \preceq, \models)$ for F. Note that K may be regarded as an **S4** model K', since it is reflexive and transitive. It now suffices to show by induction of formula F that for each subformula G of F and for each node $x \in W$

$$x \models_K G \quad \textit{iff} \quad x \models_{K'} F^{\Box}$$

Exercise 23. Prove Internalization rule for modal logics (**K**,**K4**,**S4**,**S5**):

$$\frac{A_1, A_2, \ldots, A_n \vdash B}{\Box A_1, \Box A_2, \ldots, \Box A_n \vdash \Box B}$$

Exercise 24. Establish cut-elimination property, Kripke completeness, finite model property for the modal logic **K**. (Hilbert and Gentzen style axioms were given in Lecture 10). Assume proven: Kripke soundness of **K**, and equivalence of Hilbert and Gentzen style proof systems for **K**.

Exercise 25. Which of the following is provable in **S4**? Give a proof, if any. Provide a countermodel otherwise. a) $\Box(A \to \Box B) \to (A \to \Box\Box B)$
b) $\Box(\Box A \to \Box B) \vee \Box(\Box B \to \Box A)$

Exercise 26.
a) Establish the Disjunctive Property for **S4**: $\vdash \Box A \vee \Box B$ yields $\vdash \Box A$ or $\vdash \Box B$.
b) Does the Disjunctive Property hold for **S5**?

Exercise 27. Prove Gödel's Theorem of 1933: **Int** $\vdash F \quad \Rightarrow \quad$ **S4** $\vdash F^{\Box}$, where $F^{\Box}$ is Gödel's translation consisting in prefixing each occurrence of a subformula in F by $\Box$. Hint: induction on a derivation of F in **Int**. Base case corresponds to the axioms of **Int**, prove that their translations are all derivable in **S4**. The induction step is MP in **Int**.

Exercise 28. Prove McKinsey-Tarski theorem of 1948: for any propositional formula F if **S4** $\vdash F^{\Box}$ then **Int** $\vdash F$. Hint: do the contrapositive, of course, i.e. if **Int** $\nvdash F$ then **S4** $\nvdash F^{\Box}$. Let **Int** does not prove F. Then there is an intuitionistic countermodel Kripke $K = (W, \preceq, \models)$ for F. Note that K may be regarded as an **S4**-model K', since it is reflexive and transitive. It now suffices to show by induction on a formula that for each subformula G of F and for each node $x \in W$

$$x \models_K G \quad \textit{iff} \quad x \models_{K'} G^{\Box}$$

Exercise 29. Prove Internalization rule for modal logics (**K**,**K4**,**S4**,**S5**):

$$\frac{A_1, A_2, \ldots, A_n \vdash B}{\Box A_1, \Box A_2, \ldots, \Box A_n \vdash \Box B}$$

Exercise 30. Establish cut-elimination property, Kripke completeness, finite model property for the modal logic **K**. (Hilbert and Gentzen style axioms were given in Lecture 10). Assume proven: Kripke soundness of **K**, and equivalence of Hilbert and Gentzen style proof systems for **K**.

Exercise 31. Find derivations in **IntN**
a. $(p \rightarrow q) \rightarrow \neg(p \wedge \neg q)$
b. $(\neg p \vee q) \rightarrow (p \rightarrow q)$
c. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \wedge q) \rightarrow r)$

Exercise 32. Find a Gentzen style derivation of $\Rightarrow \neg\neg(\neg\neg p \rightarrow p)$. Apply the step-by-step algorithm of transforming Gentzen style derivations into **IntN**-derivations to transform this derivation into a Natural derivation of a formula $\neg\neg(\neg\neg p \rightarrow p)$.

Exercise 33. Find a derivation of $(p \rightarrow q) \rightarrow \neg(p \wedge \neg q)$ in **IntN**. Apply the algorithm of transforming **IntN**-derivations into Gentzen style derivations to transform this derivation into an **IntG**-derivation of a sequent $\Rightarrow (p \rightarrow q) \rightarrow \neg(p \wedge \neg q)$.

Exercise 34. Consider Peirce Law $((p \rightarrow q) \rightarrow p) \rightarrow p$, which is a classical tautology not derivable in **Int** (prove!). Show that the classical logic **Cl** is not conservative w.r.t. its fragment axiomatized by axioms and rules concerning implication only.

Exercise 35. Find a λ-term corresponding to an **IntN**-derivation of $((p\wedge q)\rightarrow r)\rightarrow(p\rightarrow(q\rightarrow r))$.

Exercise 36. Find internalizations of proofs from 3.2 and 3.3. That means, for a derivation $A_1, A_2 \vdash F$ find a combinatory term (or a λ-term) $t(x_1, x_2)$ such that if $x_1^{A_1}$ and $x_2^{A_2}$, then t^F. In other words, if variables x_1, x_2 have types A_1, A_2 respectively, then $t(x_1, x_2)$ has type F. Yet another equivalent formulation of the desired: $x_1{:}A_1, x_2{:}A_2 \vdash t(x_1, x_2){:}F$.

Exercise 37. Find a normal form of combinatory term $\mathbf{s}\cdot x\cdot(\mathbf{k}\cdot y)\cdot z$, where types match properly:
$\mathbf{s}{:}[(A\rightarrow(B\rightarrow C))\rightarrow((A\rightarrow B)\rightarrow(A\rightarrow C))]$
$\mathbf{k}{:}(B\rightarrow(A\rightarrow B))$
$x{:}(A\rightarrow(B\rightarrow C))$, $y{:}B$, $z{:}A$.

Exercise 38.
a) Find an **IntG**-derivation of $\Rightarrow\neg\neg[(A\rightarrow B)\vee(B\rightarrow A)]$
b) Apply the step-by-step algorithm of transforming Gentzen style derivations

into to transform a derivation from (a) into a natural derivation of formula $\neg\neg[(A\to B)\vee(B\to A)]$.

Exercise 39.
a) Find an **IntN**-derivation of $(\neg A\vee B)\to\neg(A\wedge\neg B)$
b) Apply the step-by-step algorithm of transforming natural derivations into **IntG**-derivations to transform a derivation from (a) into an **IntG** derivation of sequent $\Rightarrow(\neg A\vee B)\to\neg(A\wedge\neg B)$. Don't simplify the resulting Gentzen style derivation.

Exercise 40.
a) Find a natural deduction derivation $p\to q\vdash(p\to r)\to(p\to q\wedge r)$
b) Find a λ-term corresponding to this derivation.

Exercise 41.
a) Find Hilbert style derivation of $A\to C\vdash A\to(B\to C)$.
b) Find internalization of this proof as a combinatory term.

Exercise 42. Find the normal forms of the following combinatory terms. Types are suppressed for short, assume they all match properly, x, y, z are variables.

a) $\mathbf{s}(\mathbf{ks})\mathbf{k}x$
b) $\mathbf{s}(\mathbf{ks})\mathbf{k}xy$
c) $\mathbf{s}(\mathbf{ks})\mathbf{k}xyz$
d) $\mathbf{s}(\mathbf{bbs})(\mathbf{kk})xyz$, where $\mathbf{b}$ is $\mathbf{s}(\mathbf{ks})\mathbf{k}$.

Exercise 43. Untyped combinatory terms are built from untyped variables $x, y, z, \ldots$ and two untyped constants $\mathbf{k}$ and $\mathbf{s}$. There are two basic reductions copied from the typed case:

$\mathbf{k}uv$	reduces to	u
$\mathbf{s}uvw$	reduces to	$uw(vw)$.

Show that strong normalization does not hold for untyped combinatory terms.

Exercise 44. Consider a system $\mathbf{IntG}^+$ which is obtained from **IntG** by adding new axiom sequents $\Rightarrow \neg\neg A \to A$ for all A's.
a) Show that $\mathbf{IntG}^+$ is equivalent to the classical logic, i.e. **Cl** proves F iff $\mathbf{IntG}^+$ proves $\Rightarrow F$.
b) Show that Cut rule cannot be eliminated in $\mathbf{IntG}^+$. Hint: check the Disjunctive Property for Cut-free derivations in $\mathbf{IntG}^+$.

Exercise 45. a) Find an **IntH**-derivation of $((A \to B) \to B) \to B \vdash A \to (A \to (A \to B))$. Feel free to use the Deduction Theorem.

b) Internalize this derivation as a combinatory term, possibly using emulated λ-abstraction. Don't bother converting it into a pure combinatory format.

Exercise 46. Church numerals are λ-terms
$\overline{0} = \lambda sz.z$
$\overline{1} = \lambda sz.sz$
$\overline{2} = \lambda sz.s(sz)$
$\overline{3} = \lambda sz.s(s(sz))$
$\overline{4} = \lambda sz.s(s(s(sz)))$
$\ldots$
Consider λ-term $t = \lambda xy.yx$. Find normal forms of the following terms:
a) $t \cdot \overline{2} \cdot \overline{1}$
b) $t \cdot \overline{2} \cdot \overline{n}$
c) $t \cdot \overline{m} \cdot \overline{n}$

Exercise 47. Which of the following is provable in **S4**? Provide a justification.
a) $\Box(\Box A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
b) $(\Box A \rightarrow \Box B) \rightarrow \Box(\Box A \rightarrow B)$
c) $\Box(\Box(A \rightarrow \Box A) \rightarrow A) \rightarrow A$

Exercise 48. Let
$? = \lambda$*abcdefghijklmnopqstuvwxyzr.r(thisisafixedpointcombinator)*;
$ = ??????????????????????????.
Show that $ is a fixed point combinator, i.e. that $\$F = F(\$F)$ holds for all λ-terms F.

Exercise 49. We know that with respect to Gödel's translation

$$tr(F) = \textit{box each subformula of F}$$

S4 corresponds to **Int**. Show that **S5** corresponds to the classical logic **Cl** with respect to the same Gödel's translation:

$$\mathbf{Cl} \vdash F \quad \textit{if and only if} \quad \mathbf{S5} \vdash tr(F).$$