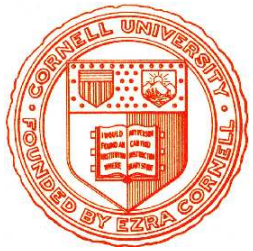
Automatisierte Logik und Programmierung

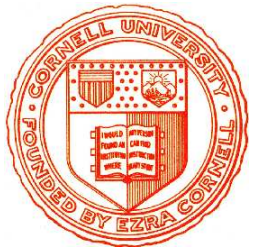
Teil I



Formalisierung von
Logik, Berechnung und Datentypen

Automatisierte Logik und Programmierung

Einheit 2



Formale Logik (Teil 1)

1. Formalisierung in Aussagenlogik
2. Evidenz für logische Aussagen
3. Evidenzkonstruktion mit Refinement Logik
4. Formale Prädikatenlogik
5. Metamathematik der Refinement Logik

Logik präzisiert Schlussfolgerungen

- **Wissenschaft und Programmierung benötigt Vertrauen**
  - Warum sollten wir eine aufgestellte Behauptung akzeptieren?
    - gilt ein mathematisches Theorem wirklich?
    - ist es sicher, daß ein Programm wie vorgesehen funktioniert?
  - Welche Belege können wir für eine Behauptung geben?
    - was ist ein akzeptabler Beweis?
    - wie können wir Software verifizieren?
- **Logik trennt gültige Schlüsse von ungültigen**

  Tatsache: *Alle Säugetiere sind behaart*

  Tatsache: *Alle ~~Affen~~ Teddybären sind behaart*

  Konsequenz: *Also sind alle ~~Affen~~ Teddybären Säugetiere* falsch!

  Ursprüngliche Konsequenz ist zufällig richtig aber nicht allgemeingültig
- **Formalisierung offenbart Struktur logischer Argumente**
  - Begriffe werden durch symbolische Platzhalter ersetzt
  - Abstrakte Symbole beschreiben Zusammenhänge zwischen Aussagen

$$((M \Rightarrow H) \wedge (A \Rightarrow H)) \Rightarrow (A \Rightarrow M)$$ nicht allgemeingültig

Formalisierung in Aussagenlogik

- **Syntax präzisiert natürlichsprachige Konzepte**
  - Sprachdefinition analog zur Beschreibung Programmiersprachen
    - Frei wählbare Symbole als Platzhalter für Aussagen
    - Reservierte Schlüsselwörter/Symbole für logische Bezüge
  - Komplexe Formeln werden induktiv aus einfacheren zusammengesetzt
  - Eindeutige Syntaxdefinition unterstützt spätere Implementierung
- **Erlaubte Symbole**
  - Aussagenvariablen $P, Q, R, P_o, Q_0, R_0,, \ldots$ (Großbuchstaben mit Index)
  - Logische Symbole $\neg$, $\wedge$, $\vee$, $\Rightarrow$ und Klammern
- **Syntax der Aussagenlogik**
  - Jede Aussagenvariable ist eine Formel
  - Sind A und B Formeln, dann auch $\neg A$, $(A \Rightarrow B)$, $(A \wedge B)$, $(A \vee B)$
  - Man kann Konventionen einführen, um Klammern zu sparen (später)

Die Symbole A und B sind Platzhalter für Formeln, d.h. Objekte der "Meta-Sprache"

Formalisierung umgangssprachlicher Aussagen

- *Wenn es friert, fährt die S-Bahn nicht. Die S-Bahn fährt. Also friert es nicht*
  - Mit memnonischen Abkürzungen $(((Friert \Rightarrow \neg SBahn) \wedge SBahn) \Rightarrow \neg Friert)$
  - Aussschließlich mit erlaubten Symbolen $\boldsymbol{(((P \Rightarrow \neg Q) \wedge Q) \Rightarrow \neg P)}$
- *Wenn es friert, fährt die S-Bahn nicht. Es friert es nicht. Also fährt die S-Bahn*
  - $\boldsymbol{(((P \Rightarrow \neg Q) \wedge \neg P) \Rightarrow Q)}$ (leider nicht wahr)
- *Wenn es nicht keinen Frost gibt, dann friert es*
  - $\boldsymbol{\neg\neg P \Rightarrow P}$ (Achtung, regional andere Lesweise "nicht kein"= "absolut nicht")
- *Sein oder nicht sein*
  - $\boldsymbol{P \vee \neg P}$ (der ganze Rest dieses inhaltsschweren Satzes geht verloren)
- *Dieser Satz ist wahr*
  - Nicht formulierbar in Aussagen- oder Prädikatenlogik

Semantik der Aussagenlogik

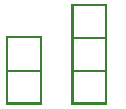
- **Logik ist mehr als nur eine formale Kurzschreibweise**
  - Formeln haben eine intendierte Bedeutung (Semantik)
  - Man kann feststellen, ob eine Aussage/Schlußfolgerung gültig ist
  - Hierzu muß man die Bedeutung von Formeln präzise festlegen
- **Wahrheitstabellen sind kein sinnvoller Weg**
  - Mischen Mathematik mit metaphysischem Konzept der Wahrheit
  - Nehmen an, daß philosphischen Frage '*Was ist Wahrheit?*' geklärt ist
- **Interpretation in Zielsprache ist unintuitiv**
  - Verlagern das Problem nur auf eine andere Sprache
  - Was ist die Bedeutung dieser Zielsprache?
- **Evidenzbasierte Semantik**
  - Beschreibt die Bedeutung logischer Formeln durch Angabe von Belegen (d.h. Rechtfertigungen) für ihre Gültigkeit
  - Umgeht Notwendigkeit philosphische Fragestellungen zu klären

Evidenz

Beleg für die Gültigkeit einer Aussage

- **Das Wort ist eigentlich eine Fehlübersetzung**
  - *evidence* (engl.) $\equiv$ Beweis, Beleg, (juristisch) Indiz, Zeugenaussage
    (Mit dieser Bedeutung hat es sich inzwischen festgesetzt)
  - Korrekt: "das dem Augenschein nach unbezweifelbar Erkennbare"
- **Evidenz für Alltagsargumenten**
  - Die Polizei behauptet, Sie wären zu schnell gefahren
    Was wäre ein Beweis für diese Behauptung? Radarmessung, Photo
  - Nach einem Unfall behauptet Ihr Gegner, Sie hätten Ihr Handy benutzt
    Welche Belege könnte es dafür geben? Daten des Providers, Uhrzeit
  - Vor einer Weile soll es geregnet haben - welches Indiz zeigt dies?
    Die Tatsache, daß die Straße naß ist, ist ein Indiz, aber kein Beweis
  - Wie können Sie beweisen, daß die Anzahl der Personen in diesem Raum entweder gerade oder ungerade ist?
    Das Argument "nichts anderes ist möglich" ist unbefriedigend – zählen ist besser

Evidenz für mathematische Aussagen

- **Beleg für** $2<3$? Einfaches anschauliches Argument 
  - Methodik nur für einfache Beispiele geeignet. Mathematik muß Wege beschreiben, alles auf wenige selbsterklärende Konzepte zu reduzieren
- **Evidenz für** $0=0$**?** selbsterklärend
  - Fundamentale Aussage, nicht durch anderes zu beweisen (Axiom)
- **... für** $1=1$, $2=2$, $3=3$, ... **?** auch selbsterklärend?
  - Im Prinzip könnte man dem zustimmen, aber was ist mit $1111111111111111111111111111111 = 1111111111111111111111111111111$?
  - Besser: Reduktion von Zahlenvergleichen auf Vorgänger und $0=0$
- **... für** $0\neq1$**?** selbsterklärend?
  - Es gibt keinen Beweis für $0\neq1$ oder $0=1$
  - $0=1$ wäre absurd (Arithmetik würde kollabieren). Man *muß* $0\neq1$ fordern
  - $0\neq2$, $0\neq3$,... ist nicht reduzierbar, daher Postulat $0\neq x+1$ $(0\neq s(x))$
- **... für "je zwei Zahlen sind gleich oder verschieden"?** selbsterklärend???
  - Zu sagen "nichts anderes ist möglich", wäre sehr unbefriedigend
  - Besserer Beleg: Reduktionsalgorithmus, der Vergleich durchführt

Evidenzsemantik der Aussagenlogik

- **Konstruktion von Rechtfertigungen für Formeln**
  - Rechtfertigungen werden als formale Evidenz(terme) ausgedrückt
  - Induktive Konstruktion von Evidenz folgt syntaktischen Aufbau

  Achtung: Wie bei Programmiersprachen dürfen ausschließlich bereits erklärte Konstrukte verwendet werden. Gesetze der Boolesche Algebra können erst angewandt werden, wenn hierfür Evidenz vorliegt

- **Formale Schreibweisen**
  - $[A]$: Menge (oder Typ) der Evidenzen für die Formel A
  - $a \in [A]$: Term a ist Evidenz für die Formel A (mathematische Notation)

    $a : [A]$: a ist Evidenz für die Formel A (Programmiersprachennotation)

- **Evidenz für Aussagenvariablen** **$[A]$ bleibt unspezifiziert**
  - Aussagenvariablen sind Platzhalter für unbekannte Aussagen
  - Keine feste Evidenz möglich

  Evidenzkonstruktion nur möglich, wenn man etwas über A in der Hand hat (Folie 20)

Evidenz für Implikationen

- **$A \Rightarrow B$: "Aus A folgt B"** $\qquad [A \Rightarrow B] = [A] \to [B]$
  - Aus Evidenz a für A muß eine Evidenz b für B konstruiert werden
  - Evidenz für $A \Rightarrow B$ muß Funktion f sein mit $f(a) : [B]$, falls $a : [A]$
  - Typ der Evidenzen für $A \Rightarrow B$ ist Typ der Funktionen von $[A]$ nach $[B]$
- **Konkrete Evidenz für $P \Rightarrow P$**
  - Evidenz ist Funktion $f : [P] \to [P]$ mit $f(p) : [P]$, falls $p : [P]$
  - Einfachste Funktion dieser Art ist Identität, d.h. $f(p) = p$
  - Einfachste Notation ist Schreibweise des λ-Kalküls: $\qquad f = \boldsymbol{\lambda p.\, p}$
- **Konkrete Evidenz für $P \Rightarrow (Q \Rightarrow P)$**
  - Evidenz ist $f : [P] \to ([Q] \to [P])$ mit $f(p) = g_p$ für eine Funktion $g_p : [Q] \to [P]$, falls $p : [P]$. Dabei ist $q_p(q) : [P]$, falls $q : [Q]$
  - Einfachste Lösung ist $g_p = \lambda q.\, p$, also $\qquad f = \boldsymbol{\lambda p.\, (\lambda q.\, p)}$
- **Konkrete Evidenz für $P \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)$**
  - Evidenz ist Funktion f mit $f(p) = g_p$ für ein $g_p : ([P] \to [Q]) \to [Q]$, falls $p : [P]$. Dabei ist $q_p(h) : [Q]$, falls $h : [P] \to [Q]$
  - Einfachste Lösung ist $g_p(h) = h(p)$, also $\qquad f = \boldsymbol{\lambda p.\, (\lambda h.\, h(p))}$

Evidenz für Konjunktionen

- **$A \wedge B$: "A und B"** $\qquad [A \wedge B] = [A] \times [B]$
  - Um Evidenz für $A \wedge B$ zu konstruieren, braucht man Evidenzen a für A und b für B, also ein Paar (a, b) mit $a : [A]$ und $b : [B]$
  - Typ der Evidenzen für $A \wedge B$ ist das Produkt von $[A]$ und $[B]$
- **Konkrete Evidenz für $P \Rightarrow (Q \Rightarrow (P \wedge Q))$**
  - Evidenz ist Funktion f mit $f(p) = g_p$ für ein $g_p : [Q] \to ([P] \times [Q])$, falls $p : [P]$. Dabei ist $q_p(q) : [P] \times [Q]$, falls $q : [Q]$
  - Einfachste Lösung ist $g_p(q) = (p, q)$, also $\qquad f = \lambda p.\, (\lambda q.\, (p, q))$
- **Konkrete Evidenz für $P \Rightarrow (Q \Rightarrow (Q \wedge P))$**
  - Dasselbe Argument liefert $\qquad f = \lambda p.\, (\lambda q.\, (q, p))$
- **Konkrete Evidenz für $(P \wedge Q) \Rightarrow P$**
  - Evidenz ist Funktion f mit $f(x) : [P]$ für ein $x : [P] \times [Q]$,
  - x muß ein Paar sein, dessen erste Komponente $x.1$ in $[P]$ liegt und dessen zweite Komponente $x.2$ zu $[Q]$ gehört
  - Der gesuchte Evidenzterm ist damit $\qquad f = \lambda x.\ x.1$

Evidenz für Disjunktionen

- **$A \vee B$: "A oder B"** $\qquad [A \vee B] = [A] + [B]$
  - Um Evidenz für $A \vee B$ zu konstruieren, braucht man Evidenz $a : [A]$ oder Evidenz $b : [B]$ und eine Kennzeichnung, welches von beiden man hat
  - Mögliche Evidenzen sind $\texttt{inl}(a)$ mit $a : [A]$ und $\texttt{inr}(b)$ mit $b : [B]$
  - Typ der Evidenzen für $A \vee B$ ist die direkte Summe von $[A]$ und $[B]$
- **Konkrete Evidenz für $P \Rightarrow (P \vee Q)$**
  - Evidenz ist Funktion f mit $f(p) : [[P] + [Q]$
  - Einfachste Lösung ist $\qquad f = \boldsymbol{\lambda p.}\, \texttt{inl}(\boldsymbol{p})$
- **Konkrete Evidenz für $(P \vee Q) \Rightarrow (Q \vee P)$**
  - Evidenz ist Funktion f mit $f(x) : [Q] + [P]$ für ein $x : [P] + [Q]$
  - x is entweder $\texttt{inl}(p)$ mit $p : [P]$ oder $\texttt{inr}(q)$ mit $q : [Q]$
    Im ersten Fall ist $f(x) = \texttt{inr}(p)$, im zweiten $f(x) = \texttt{inl}(q)$
  - Als geschlossener Term (Pattern-Matching Notation)

$$f = \boldsymbol{\lambda x.}\, (\texttt{case}\ \boldsymbol{x}\ \texttt{of}\ \texttt{inl}(\boldsymbol{p}) \rightarrow \texttt{inr}(\boldsymbol{p}) \mid \texttt{inr}(\boldsymbol{q}) \rightarrow \texttt{inl}(\boldsymbol{q}))$$

Evidenz für Negationen

- **$\neg A$: "nicht A"** $\qquad [\neg A] = [A] \to \{\}$
  - Evidenz für $\neg A$ ist Nachweis(!), daß keine Evidenz für A möglich ist
  - Aus Annahme $a : [A]$ muß Widerspruch konstruiert werden
  - Aussagenlogik hat keine fundamentalen Widerspruche (wie z.B. `0=1`)
    Ergänze konstante Formel f und das Postulat $[\mathsf{f}] = \{\}$ ($\hat{=}$ der leere Typ)
  - Typ der Evidenzen für $\neg A$ ist Typ der Funktionen von $[A]$ nach $\{\}$

- **Konkrete Evidenz für $P \Rightarrow \neg\neg P$**
  - Evidenz ist Funktion f mit $f(p) = g_p$ für ein $g_p : ([P] \to \{\}) \to \{\}$,
    falls $p : [P]$. Dabei ist $q_p(h) : \{\}$, falls $h : [P] \to \{\}$
    Einfachste Lösung ist $g_p(h) = h(p)$, also $\qquad f = \boldsymbol{\lambda p.\,(\lambda h.\,h(p))}$
    Dies bedeutet, daß es keine Funktion $h : [P] \to \{\}$ geben kann, also $\neg P$
    keine Evidenz hat, da man sonst ein Element des leeren Typs konstruieren könnte

- **Konkrete Evidenz für $\neg(P \vee Q) \Rightarrow \neg P$**
  - Evidenz ist Funktion f mit $f(h) = g_h : [P] \to \{\}$ falls $h : ([P] + [Q]) \to \{\}$.
    Dabei ist $q_h(p) : \{\}$, falls $p : [P]$
  - Lösung ist $g_h(p) = h(\mathtt{inl}(p))$, also $\qquad f = \boldsymbol{\lambda h.\,(\lambda p.\,h(\mathtt{inl}(p)))}$

Evidenz für komplexere Formeln

- **Evidenz für $((P \vee Q) \wedge ((P \Rightarrow R) \wedge (Q \Rightarrow R))) \Rightarrow R$**
  - Intuitiv einfach: wenn P gilt, beweise R mit $P \Rightarrow R$, sonst mit $Q \Rightarrow R$
  - Evidenzkonstruktion ist wie Programmierung mit Typbedingungen:
    Finde einen Term in $(([P]+[Q])\times(([P]\to[R])\times([Q]\to[R]))) \to [R]$
  - Evidenz ist Funktion f mit $f(x) = r : [R]$ für eine Eingabe
    x aus $([P]+[Q])\times(([P]\to[R])\times([Q]\to[R]))$
  - x ist ein Tripel $(z, (g, h))$ mit $z : [P]+[Q]$, $g : [P]\to[R]$, $h : [Q]\to[R]$
  - z ist entweder $\texttt{inl}(p)$ mit $p : [P]$ oder $\texttt{inr}(q)$ mit $q : [Q]$
    Im ersten Fall ist $r = g(p)$, im zweiten $r = h(q)$
  - Lösung wäre $\lambda(z, (g, h)).\ \texttt{case}\, z\, \texttt{of}\ \texttt{inl}(p) \to g(p) \mid \texttt{inr}(q) \to h(q)$
  - Dies ist kein korrekter Term (kein Pattern Matching bei λ erlaubt)
    Verwende $z = x.1,\ g = x.2.1,\ h = x.2.2$
  - Evidenz ist also $\lambda x.(\texttt{case}\, x.1\, \texttt{of}\ \texttt{inl}(p) \to x.2.1(p) \mid \texttt{inr}(q) \to x.2.2(q)$
- **Evidenz für $(P \vee Q) \Rightarrow ((P \Rightarrow R) \Rightarrow ((Q \Rightarrow R) \Rightarrow R))$**
  - Evidenz ist $\lambda z.\lambda g.\lambda h.\ \texttt{case}\, z\, \texttt{of}\ \texttt{inl}(p) \to g(p) \mid \texttt{inr}(q) \to h(q)$
  - Argumente kommen der Reihe nach statt im Block (currying)
    Alternativ $\lambda z.\ \texttt{case}\ z\ \texttt{of}\ \texttt{inl}(p) \to (\,\lambda g.\lambda h.\, g(p)\,) \mid \texttt{inr}(q) \to (\,\lambda g.\lambda h.\, h(q)\,)$

Ein paar überraschende Ergebnisse

- **Evidenz für $P \vee \neg P$**
  - Evidenz muß $\texttt{inl}(p)$ für ein $p : [P]$ oder $\texttt{inr}(\lambda p.\, x_p)$ für ein $x_p : \{\}$ sein.
  - Solange man nichts über $[P]$ weiß, kann man beides nicht konstruieren
  - **Keine universelle Evidenz**
- **Evidenz für $\neg\neg P \Rightarrow P$**
  - Evidenz ist Funktion f mit $f(h) = p : [P]$, falls $h : ([P] \to \{\}) \to \{\}$
  - Es gibt keinen allgemeinen Weg, p aus h zu konstruieren
  - **Keine universelle Evidenz**
- **Evidenz für $(P \Rightarrow Q) \Rightarrow (\neg P \vee Q)$**
  - Evidenz ist Funktion f mit $f(h) = x : ([P] \to \{\}) + [Q]$ falls $h : [P] \to [Q]$.
  - x muß $\texttt{inl}(\lambda p.\, x_p)$ für ein $x_p : \{\}$ oder $\texttt{inr}(q)$ für ein $q : [Q]$ sein
  - Solange man nichts über $[P]$ weiß, kann man beides nicht konstruieren
  - **Keine universelle Evidenz**

Was ist das denn für eine Logik?

- **Bekannte logische Gesetze gelten nicht?**
  - Gesetz vom ausgeschlossenen Dritten $P \vee \neg P$
  - Gesetz der Doppelten Negation $\neg\neg P \Rightarrow P$
  - Zusammenhang Implikation und Disjunktion $(P \Rightarrow Q) \Rightarrow (\neg P \vee Q)$

    . . . und viele andere mehr
- **Was bedeutet $P \vee \neg P$ denn genau?**
  - "*Jede beliebige Aussage ist ~~wahr oder nicht~~*"? Das steht da nicht!

    "*Jede beliebige Aussage ist gültig oder ihre Negation ist gültig*"
  - Begriff der "Wahrheit" verwendet metaphysische Ideen und Einsichten

    Mit solchen Konzepten kann man in der Informatik nicht arbeiten
  - $P \vee \neg P$ würde bedeuten, daß man jede (!) Aussage entscheiden kann

    Aus der theoretischen Informatik wissen wir, daß dies nicht stimmt

    Heißt das, daß die Formel $\neg(P \vee \neg P)$ gültig ist?
- **Logik heißt "Intuitionistische (konstruktive) Logik"**
  - Vermeidet Wahrheitsbegriff der "klassischen" Logik
  - Akzeptiert nur mathematische Aussagen, die belegbar sind
  - Ist für das Schließen über Programme besser geeignet

Evidenzsemantik – Zusammenfassung

Aussage A	Evidenztyp $[A]$	Evidenzkonstruktion	Dekompositionsterm
$A \Rightarrow B$	$[A] \to [B]$	$\lambda a.b$	$f(a)$
$A \wedge B$	$[A] \times [B]$	(a, b)	$x.1$, $x.2$
$A \vee B$	$[A] + [B]$	$\texttt{inl}(a)$, $\texttt{inr}(b)$	$\texttt{case}\ x\ \texttt{of}\ \texttt{inl}(a) \to s \mid \texttt{inr}(b) \to t$
$\neg A$	$[A] \to \{\}$	$\lambda a.b$	$f(a)$
f	$\{\}$	–	–

- **Aussagen korrespondieren mit Datentyp der Evidenzen**
  - Es gibt Terme, um Evidenz zu konstruieren
  - Es gibt Terme, um komplexe Evidenz zu zerlegen
- **Evidenzterme bilden eine Art Programmiersprache**
  - Logische Formeln sind Spezifikationen dieser Programme
  - Ausdruckstarke Logiken ermöglichen sehr genaue Spezifikationen
- **Man kann mit Evidenztermen rechnen**
  - Konstruktions- und Dekompositionsterme sind invers zueinander
  - $(a, b).1 \hat{=} a$, $(\lambda a.b)(c) \hat{=} b[c/a]$, ... (... mehr dazu in der Einheit zum λ-Kalkül)

Systematische Konstruktion von Evidenz

- **Evidenzkonstruktion ist wie Programmieren**
  - Man muß einen Term finden, der eine (Datentyp-)Spezifikation erfüllt
  - Nicht immer einfach, wenn man semantisch argumentiert, da tiefes Verständnis des Zusammenhangs zwischen Ein- und Ausgabe nötig
  - Die Konstruktion einer Evidenz für $\neg\neg(P \vee \neg P)$ ist nicht trivial
- **Evidenzkonstruktion ist logische Beweisführung**
  - Spezifikationen der Evidenzterme sind logische Formeln
  - Logische Formeln können in Teilformeln zerlegt werden
  - Beweise von Formeln sind reduzierbar auf Beweise der Teilformeln Beweisaufgabe wird verfeinert zu einfacheren Teilaufgaben
- **Evidenzkonstruktion durch schrittweise Verfeinerung**
  - Zerlege Beweisaufgabe in kleinere Teilaufgaben
  - Konstruiere Evidenz für atomare Beweisaufgaben
  - Setze Evidenz einer Formel aus Evidenzen für Teilformeln zusammen

Beweis durch Verfeinerung

- **Informaler Beweis für $P \Rightarrow (Q \Rightarrow (P \wedge Q))$**
  - Wir nehmen an, daß P gilt und müssen $Q \Rightarrow (P \wedge Q)$ zeigen
  - Dafür nehmen wir an, daß zusätzlich Q gilt und müssen $P \wedge Q$ zeigen
  - Da P und Q gilt, gilt auch $P \wedge Q$
- **Beweis für $P \Rightarrow (Q \Rightarrow (P \wedge Q))$ mit Evidenzkonstruktion**
  - Wir nehmen $p : [P]$ an und müssen $f_p : [Q \Rightarrow (P \wedge Q)]$ konstruieren
  - Dafür nehmen wir $q : [Q]$ an und müssen $x : [P \wedge Q]$ konstruieren
  - Um x zu konstruieren, brauchen wir ein $p_0 : [P]$ und ein $q_0 : [Q]$.
  - Da wir $p : [P]$ und $q : [Q]$ haben, können wir $p_0 = p : [P]$ wählen
  - Da wir $p : [P]$ und $q : [Q]$ haben, können wir $q_0 = q : [Q]$ wählen
  - Damit ist $x = (p, q) : [P \wedge Q]$ und $f_p = \lambda q.\,(p, q) : [Q \Rightarrow (P \wedge Q)]$
    $\boldsymbol{\lambda p.\,\lambda q.\,(p, q) : [P \Rightarrow (Q \Rightarrow (P \wedge Q))]}$ ist die gesuchte Evidenz
- **Methodik läßt sich durch formale Regeln beschreiben**
  - Regeln zerlegen logische Formeln und setzen Evidenzterme zusammen
  - Regeln sind implementierbar durch Pattern Matching und Instantiierung

Beweisen durch Verfeinerung logischer Formeln

Notationen und Begriffe

– Kalkül verwaltet zu beweisende Formel und Annahmen

– Regeln operieren auf Beweiszielen (Sequenzen) der Form $H \vdash C$

  Lesart: *Konklusion C folgt aus Liste der Annahmen (Hypothesen) H*

– Initialziel ist $\vdash A$, d.h. Beweis der Formel A ohne weitere Annahmen

– Regeln transformieren Beweisziele in Listen von Teilzielen

  Regeln werden als
  Regelschemata dargestellt
  mit Platzhaltern für Formeln

$$\frac{H \vdash G}{\begin{array}{c} H_1 \vdash G_1 \\ \vdots \\ H_n \vdash G_n \end{array}}$$

  Beweisbarkeit der Teilziele impliziert Beweisbarkeit des Hauptziels

- **Regelschema für Konjunktionen**

$$\begin{array}{l} H \vdash A \wedge B \\ \quad H \vdash A \\ \quad H \vdash B \end{array} \quad \texttt{andR}$$

$$\begin{array}{ll} \vdash\ (P \Rightarrow Q) \wedge (R \Rightarrow Q) & \texttt{BY andR} \\ 1.\ \vdash\ P \Rightarrow Q & \\ 2.\ \vdash\ R \Rightarrow Q & \end{array}$$

  - Beweisbarkeit von $A \wedge B$ folgt aus Beweisbarkeit von A und von B
  - Anwendung der Regel `andR` auf konkrete Formel $(P \Rightarrow Q) \wedge (R \Rightarrow Q)$ instantiiert A mit $P \Rightarrow Q$ und B mit $R \Rightarrow Q$
  - Entstehende Teilziele werden numeriert

- **Regelschema mit Evidenzkonstruktion**
  - Beweisbarkeit der Teilziele impliziert Beweisbarkeit des Hauptziels
  - Evidenz des Hauptziels entsteht aus Evidenz für Teilziele

$$\begin{array}{ll} H \vdash A \wedge B & \texttt{ev} = (a, b) \\ \quad H \vdash A & \texttt{ev} = a \\ \quad H \vdash B & \texttt{ev} = b \end{array} \quad \texttt{andR}$$

  - Evidenz für $H \vdash A \wedge B$ ist (a, b), wenn a Evidenz für $H \vdash A$ ist und b Evidenz für $H \vdash B$

Regeln für Aussagenvariablen

- **Aussagenvariablen können nicht zerlegt werden**
  - Kein fester Beweis für A, solange nichts über A bekannt ist
  - Aber A kann bewiesen werden, wenn A eine der Hypothesen ist

$$\boxed{H, A, H' \vdash A} \quad \texttt{axiom}$$

  - H und H' sind (möglicherweise leere) Listen von Formeln
  - Es werden keine Teilziele generiert, da Sequenz selbsterklärend ist

- **Evidenzkonstruktion benötigt Labels für Hypothesen**

$$\boxed{H, a{:}A, H' \vdash A \quad \texttt{ev} = a} \quad \texttt{axiom}$$

  - Label der verwendeten Hypothese ist "Variable" der Evidenzsprache
  - Evidenz für Konklusion A ist Label der verwendeten Hypothese

Regeln für Implikationen (I)

- **Implikation auf rechter Seite einer Sequenz**
  - Um $H \vdash A \Rightarrow B$ zu zeigen, nimmt man A an und zeigt B
  - A wird zusätzliche Hypothese im Teilziel mit Variable a als Label

$$\frac{H, a{:}A \vdash B \quad \mathtt{ev} = b}{H \vdash A \Rightarrow B \quad \mathtt{ev} = \lambda a.b}\ \texttt{impliesR}$$

  - Regel nimmt an, daß Evidenz b für das Teilziel $H, a{:}A \vdash B$ existiert d.h. es gibt generische Methode, $b : [B]$ aus $a : [A]$ zu konstruieren
  - Evidenz für $H \vdash A \Rightarrow B$ muß Funktion $\lambda a.b : [A] \to [B]$ sein

- **Beweis für $P \Rightarrow P$**

$\vdash P \Rightarrow P$	$\mathtt{ev} = \lambda p.\, p$	BY	impliesR
1. $p{:}P \vdash P$	$\mathtt{ev} = p$	BY	axiom

  - impliesR erzeugt Teilziel mit neuer Hypothese $p{:}[P]$
  - axiom beweist Teilziel mit Evidenz p
  - impliesR konstruiert hieraus Evidenz $\lambda p.\, p$ für $P \Rightarrow P$

Regeln für Implikationen (II)

- **Implikation auf linker Seite einer Sequenz**
  - Um C unter der Annahme $A \Rightarrow B$ zu zeigen, benötigt man Evidenz für A und kann dann die Annahme B verwenden, um C zu zeigen

$$\begin{array}{ll} H, f{:}A \Rightarrow B, H' \vdash C & \texttt{ev} = c[f(a)/b] \\ H, f{:}A \Rightarrow B, H' \vdash A & \texttt{ev} = a \\ H, b{:}B, H' \vdash C & \texttt{ev} = c \end{array} \quad \texttt{impliesL}$$

  - Annahme $A \Rightarrow B$ benötigt Label f
  - B wird zusätzliche Hypothese im Teilziel 2 mit Variable b als Label
  - Regel nimmt an, daß Evidenzen $a : [A]$ bzw. $c : [C]$ existieren es gibt Methode, $c : [C]$ aus beliebigen $b : [B]$ zu konstruieren und $f(a)$ ist konkrete Evidenz in $[B]$
  - Anwendung von $\lambda b.c$ auf $f(a)$ liefert Evidenz für C im Hauptziel Evidenz $(\lambda b.c)(f(a))$ wird evaluiert zu reduzierter Form $c[f(a)/b]$
  - Annahme $A \Rightarrow B$ wird im Teilziel 1 möglicherweise noch benötigt

Anwendung der Implikationsregeln

- **Beweis für $P \Rightarrow (Q \Rightarrow P)$**

$\vdash\ P \Rightarrow (Q \Rightarrow P)$	ev = $\lambda p.\,(\lambda q.\,p)$	BY impliesR
1. $p{:}P \vdash\ Q \Rightarrow P$	ev = $\lambda q.\,p$	BY impliesR
1.1. $p{:}P, q{:}Q \vdash\ P$	ev = p	BY axiom

  – Zwei Anwendungen von `impliesR` erzeugen Beweisbaum der Tiefe 2

  – Numerierung `1.1.` beschreibt erstes Teilziel des Teilziels 1

- **Beweis für $P \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)$**

$P \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)$	ev = $\lambda p.\,(\lambda h.\,(h(p)))$	BY impliesR
1. $p{:}P \vdash\ (P \Rightarrow Q) \Rightarrow Q$	ev = $\lambda h.\,h(p)$	BY impliesR
1.1. $p{:}P, h{:}(P \Rightarrow Q) \vdash\ Q$	ev = $h(p)$	BY impliesL
1.1.1. $p{:}P, h{:}(P \Rightarrow Q) \vdash\ P$	ev = p	BY axiom
1.1.2. $p{:}P, q{:}Q \vdash\ Q$	ev = q	BY axiom

  – Evidenz $h(p)$ in Schritt 1.1 ist reduzierte Form von $(\lambda q.q)(h(p))$

Regeln für Konjunktion

- **Konjunktion auf rechter Seite einer Sequenz**
  - Um $H \vdash A \wedge B$ zu zeigen, muß A und B gezeigt werden

$$\begin{array}{l} H \vdash A \wedge B \quad \mathtt{ev} = (a, b) \\ \quad H \vdash A \quad \mathtt{ev} = a \\ \quad H \vdash B \quad \mathtt{ev} = b \end{array} \quad \texttt{andR}$$

  - Regel setzt Evidenzen a und b der Teilziele zu (a, b) zusammen
- **Konjunktion auf linker Seite einer Sequenz**
  - Die Annahme $A \wedge B$ ist äquivalent zu den beiden Annahmen A und B

$$\begin{array}{l} H, x{:}A \wedge B, H' \vdash C \quad \mathtt{ev} = c[x.1, x.2/a, b] \\ \quad H, a{:}A, b{:}B, H' \vdash C \quad \mathtt{ev} = c \end{array} \quad \texttt{andL}$$

  - Label x für $A \wedge B$ entspricht Paar der Labels $a{:}A$ und $b{:}B$
  - Evidenz c hängt im Teilziel von a und b ab
  - Im Hauptziel muß a durch $x.1$ und b durch $x.2$ ersetzt werden
    Evidenz $((\lambda a.\,(\lambda b.c))(x.1))(x.2)$ wird evaluiert zu $c[x.1, x.2/a, b]$

Anwendung der Konjunktionsregeln

- **Beweis für $P \Rightarrow (Q \Rightarrow (P \wedge Q))$**

$\vdash\ P \Rightarrow (Q \Rightarrow (P \wedge Q))$	ev = $\lambda p.\,(\lambda q.\,(p, q))$	BY impliesR
1. $p{:}P \vdash\ Q \Rightarrow (P \wedge Q)$	ev = $\lambda q.\,(p, q)$	BY impliesR
1.1. $p{:}P, q{:}Q \vdash\ P \wedge Q$	ev = (p, q)	BY andR
1.1.1. $p{:}P, q{:}Q \vdash\ P$	ev = p	BY axiom
1.1.2. $p{:}Q, q{:}Q \vdash\ P$	ev = q	BY axiom

  – Naheliegender Beweis liefert gleiche Evidenz wie zuvor

- **Beweis für $(P \wedge Q) \Rightarrow P$**

$\vdash\ (P \wedge Q) \Rightarrow P$	ev = $\lambda x.\,x.1$	BY impliesR
1. $x{:}(P \wedge Q) \vdash\ P$	ev = $x.1$	BY andL
1.1. $p{:}P, q{:}Q \vdash\ P$	ev = p	BY axiom

  – Evidenz $x.1$ in Schritt 1 ist reduzierte Form von $((\lambda p.\,(\lambda q.\,p))(x.1))(x.2)$

Regeln für Disjunktionen

- **Disjunktion auf rechter Seite einer Sequenz**
  - Um $H \vdash A \vee B$ zu zeigen, muß A oder B gezeigt werden
  - Zwei Regeln ermöglichen es, eine Wahl zu treffen

$$\frac{H \vdash A \vee B \quad \texttt{ev} = \texttt{inl}(a)}{H \vdash A \quad \texttt{ev} = a}\ \texttt{orR1} \qquad \frac{H \vdash A \vee B \quad \texttt{ev} = \texttt{inr}(b)}{H \vdash B \quad \texttt{ev} = b}\ \texttt{orR2}$$

  - Regeln kennzeichnen Herkunft der Evidenzen a / b mit `inl` / `inr`
- **Disjunktion auf linker Seite einer Sequenz**
  - Um C unter der Annahme $A \vee B$ zu zeigen, muß C unter der Annahme A und unter der Annahme B gezeigt werden können (Fallanalyse)

$$\frac{\begin{array}{ll} H, x{:}A \vee B, H' \vdash C & \texttt{ev} = \texttt{case}\ x\ \texttt{of}\ \texttt{inl}(a) \to c_1 \\ & \quad \mid \texttt{inr}(b) \to c_2 \\ H, a{:}A, H' \vdash C & \texttt{ev} = c_1 \\ H, b{:}B, H' \vdash C & \texttt{ev} = c_2 \end{array}}{}\ \texttt{orL}$$

  - Label x für $A \vee B$ ist entweder $\texttt{inl}(a)$ mit $a{:}A$ oder $\texttt{inr}(b)$ mit $b{:}B$
  - Evidenz c_1 hängt von a, Evidenz c_2 von b ab
  - Evidenz im Hauptziel wird durch Fallanalyse zusammengesetzt

Anwendung der Disjunktionsregeln

- **Beweis für $P \Rightarrow (P \vee Q)$**

$\vdash\ P \Rightarrow (P \vee Q)$	$\mathtt{ev} = \lambda p.\, \mathtt{inl}(p)$	BY impliesR
1. $p{:}P \vdash\ P \vee Q$	$\mathtt{ev} = \mathtt{inl}(p)$	BY orR1
1.1. $p{:}P \vdash\ P$	$\mathtt{ev} = p$	BY axiom

- **Beweis für $(P \vee Q) \Rightarrow (Q \vee P)$**

$(P \vee Q) \Rightarrow (Q \vee P)$	$\mathtt{ev} = \lambda x.\, (\mathtt{case}\ x\ \mathtt{of}\ \mathtt{inl}(p) \to \mathtt{inr}(p) \mid \mathtt{inr}(q) \to \mathtt{inl}(q))$	BY impliesR
1. $x{:}(P \vee Q) \vdash\ Q \vee P$	$\mathtt{ev} = \mathtt{case}\ x\ \mathtt{of}\ \mathtt{inl}(p) \to \mathtt{inr}(p) \mid \mathtt{inr}(q) \to \mathtt{inl}(q)$	BY orL
1.1. $p{:}P \vdash\ Q \vee P$	$\mathtt{ev} = \mathtt{inr}(p)$	BY orR2
1.1.1. $p{:}P \vdash\ P$	$\mathtt{ev} = p$	BY axiom
1.2. $q{:}Q \vdash\ Q \vee P$	$\mathtt{ev} = \mathtt{inl}(q)$	BY orR1
1.2.1. $q{:}Q \vdash\ Q$	$\mathtt{ev} = q$	BY axiom

Regeln für Negation

Spezialisierte Implikationsregeln, da $\neg A \;\hat{=}\; A \Rightarrow \mathsf{f}$

- **Negation auf rechter Seite einer Sequenz**
  - Um $H \vdash \neg A$ zu zeigen, muß aus Annahme A ein Widerspruch folgen

$$\frac{H, a{:}A \vdash \mathsf{f} \quad \mathtt{ev} = b}{H \vdash \neg A \quad \mathtt{ev} = \lambda a.b}\ \texttt{notR}$$

  - Es gibt keine direkte Methode, Evidenz für f zu konstruieren
- **Negation auf linker Seite einer Sequenz**
  - Um C unter Annahme $\neg A$ zu zeigen, benötigt man Evidenz für A
  - Aus dem resultierenden Widerspruch folgt C ohne weiteren Beweis (!)

$$\frac{H, f{:}\neg A, H' \vdash A \quad \mathtt{ev} = a}{H, f{:}\neg A, H' \vdash C \quad \mathtt{ev} = \mathtt{any}(f(a))}\ \texttt{notL}$$

  - Evidenz $\mathtt{any}(f(a))$ drückt aus, daß aus Widerspruch alles folgt
  - Typisierung ist $\mathtt{any}\colon \{\} \to [C]$ für beliebige Formeln C
  - Eingabe für any beschreibt Quelle des Widerspruchs,
    Der Term $f(a)$ konstruiert ein Element, das es gar nicht geben kann

Anwendung der Negationsregeln

- **Beweis für $P \Rightarrow \neg\neg P$**

$\vdash\ P \Rightarrow \neg\neg P$	ev = $\lambda p.\,(\lambda h.\,\mathtt{any}(h(p)))$	BY impliesR
1. $p{:}P \vdash\ \neg\neg P$	ev = $\lambda h.\,\mathtt{any}(h(p))$	BY notR
1.1. $p{:}P, h{:}(\neg P) \vdash\ \mathsf{f}$	ev = $\mathtt{any}(h(p))$	BY notL
1.1.1. $p{:}P, h{:}(\neg P) \vdash\ P$	ev = p	BY axiom

  - Beweis konstruiert Evidenz für $P \Rightarrow (\neg P \Rightarrow Q)$ für beliebige Q
  - Direkt entwickelte Evidenz $\lambda p.\,(\lambda h.\,h(p))$ benötigt $Q = \mathsf{f}$

- **Beweis für $\neg(P \vee Q) \Rightarrow \neg P$**

$\vdash\ \neg(P \vee Q) \Rightarrow \neg P$	ev = $\lambda h.\,(\lambda p.\mathtt{any}(h(\mathtt{inl}(p))))$	BY impliesR
1. $h{:}\neg(P \vee Q) \vdash\ \neg P$	ev = $\lambda p.\mathtt{any}(h(\mathtt{inl}(p)))$	BY notR
1.1. $h{:}\neg(P \vee Q), p{:}P \vdash\ \mathsf{f}$	ev = $\mathtt{any}(h(\mathtt{inl}(p)))$	BY notL
1.1.1. $h{:}\neg(P \vee Q), p{:}P \vdash\ P \vee Q$	ev = $\mathtt{inl}(p)$	BY orR1
1.1.1.1. $h{:}\neg(P \vee Q), p{:}P \vdash\ P$	ev = p	BY axiom

  - Beweis konstruiert Evidenz für $\neg(P \vee Q) \Rightarrow (P \Rightarrow R)$ für beliebige R

Ein komplexerer Beweis

$\vdash\ ((P \vee Q) \wedge ((P \Rightarrow R) \wedge (Q \Rightarrow R))) \Rightarrow R$ ev = $\lambda x.\,(\texttt{case}\ x.1\ \texttt{of}\ \texttt{inl}(p) \to x.2.1(p) \mid \texttt{inr}(q) \to x.2.2(q))$ BY impliesR

1. $x{:}(P \vee Q) \wedge ((P \Rightarrow R) \wedge (Q \Rightarrow R)) \vdash R$ ev = $\texttt{case}\ x.1\ \texttt{of}\ \texttt{inl}(p) \to x.2.1(p) \mid \texttt{inr}(q) \to x.2.2(q)$ BY andL

1.1. $z{:}P \vee Q, y{:}(P \Rightarrow R) \wedge (Q \Rightarrow R) \vdash R$ ev = $\texttt{case}\ z\ \texttt{of}\ \texttt{inl}(p) \to y.1(p) \mid \texttt{inr}(q) \to y.2(q)$ BY andL

1.1.1. $z{:}P \vee Q, g{:}P \Rightarrow R, h{:}Q \Rightarrow R \vdash R$ ev = $\texttt{case}\ z\ \texttt{of}\ \texttt{inl}(p) \to g(p) \mid \texttt{inr}(q) \to h(q)$ BY orL

1.1.1.1. $p{:}P, g{:}P \Rightarrow R, h{:}Q \Rightarrow R \vdash R$ ev = $g(p)$ BY impliesL g

1.1.1.1.1. $p{:}P, g{:}P \Rightarrow R, h{:}Q \Rightarrow R \vdash P$ ev = p BY axiom

1.1.1.1.2. $p{:}P, r{:}R, h{:}Q \Rightarrow R \vdash R$ ev = r BY axiom

1.1.1.2. $q{:}Q, g{:}P \Rightarrow R, h{:}Q \Rightarrow R \vdash R$ ev = $h(q)$ BY impliesL h

1.1.1.2.1. $q{:}Q, g{:}P \Rightarrow R, h{:}Q \Rightarrow R \vdash Q$ ev = q BY axiom

1.1.1.2.2. $q{:}Q, g{:}P \Rightarrow R, r{:}R \vdash R$ ev = r BY axiom

Was passiert mit $P \vee \neg P$, $\neg\neg P \Rightarrow P$, etc.?

- **Beweisansätze für $P \vee \neg P$**

```
⊢ P ∨ ¬P          BY orR1
1. ⊢ P            BY ?????
```

```
⊢ P ∨ ¬P          BY orR2
1. ⊢ ¬P           BY notR
1.1. p:P ⊢ f      BY ?????
```

  – Beide Ansätze können nicht fortgesetzt werden

- **Beweisansatz für $\neg\neg P \Rightarrow P$**

```
⊢ ¬¬P ⇒ P                  BY impliesR
1. f:¬¬P ⊢ P               BY notL
1.1. f:¬¬P ⊢ ¬P            BY notR
1.1.1. f:¬¬P, p:P ⊢ f      BY  ?????
```

  – Keine sinnvolle Fortsetzung möglich

- **Beweisansatz für $(P \Rightarrow Q) \Rightarrow (\neg P \vee Q)$**

```
⊢ (P ⇒ Q) ⇒ (¬P ∨ Q)       BY impliesR
1. f:P ⇒ Q ⊢ ¬P ∨ Q        BY impliesL?, orR1?, orR2?
```

  – Keine der drei möglichen Fortsetzungen führt zum Erfolg

Refinement Logik – Zusammenfassung

Links			Rechts		
$H, f{:}A \Rightarrow B, H' \vdash C$	$\mathtt{ev} = c[f(a)/b]$	impliesL	$H \vdash A \Rightarrow B$	$\mathtt{ev} = \lambda a.b$	impliesR
$H, f{:}A \Rightarrow B, H' \vdash A$	$\mathtt{ev} = a$		$H, a{:}A \vdash B$	$\mathtt{ev} = b$	
$H, b{:}B, H' \vdash C$	$\mathtt{ev} = c$				
$H, x{:}A \wedge B, H' \vdash C$	$\mathtt{ev} = c[x.1, x.2/a, b]$	andL	$H \vdash A \wedge B$	$\mathtt{ev} = (a, b)$	andR
$H, a{:}A, b{:}B, H' \vdash C$	$\mathtt{ev} = c$		$H \vdash A$	$\mathtt{ev} = a$	
			$H \vdash B$	$\mathtt{ev} = b$	
$H, x{:}A \vee B, H' \vdash C$	$\mathtt{ev} = \mathtt{case}\ x\ \mathtt{of}\ \mathtt{inl}(a) \to c_1 \mid \mathtt{inr}(b) \to c_2$	orL	$H \vdash A \vee B$	$\mathtt{ev} = \mathtt{inl}(a)$	orR1
$H, a{:}A, H' \vdash C$	$\mathtt{ev} = c_1$		$H \vdash A$	$\mathtt{ev} = a$	
$H, b{:}B, H' \vdash C$	$\mathtt{ev} = c_2$		$H \vdash A \vee B$	$\mathtt{ev} = \mathtt{inr}(b)$	orR2
			$H \vdash B$	$\mathtt{ev} = b$	
$H, f{:}\neg A, H' \vdash C$	$\mathtt{ev} = \mathtt{any}(f(a))$	notL	$H \vdash \neg A$	$\mathtt{ev} = \lambda a.b$	notR
$H, f{:}\neg A, H' \vdash A$	$\mathtt{ev} = a$		$H, a{:}A \vdash \mathtt{f}$	$\mathtt{ev} = b$	
			$H, a{:}A, H' \vdash A$	$\mathtt{ev} = a$	axiom