工学部専門科目「計算と論理」配布資料
自然演繹と Coq

五十嵐 淳
京都大学 大学院情報学研究科 通信情報システム専攻
cal21@fos.kuis.kyoto-u.ac.jp
http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/class/cal/

December 21, 2021

証明体系 (*proof system*) のひとつである**自然演繹** (*natural deduction*) によって，Coq で使われている論理の (一部の) 形式化 (記号化) を行う．

- 証明体系 (proof system):
  - 「(判断・命題が) 証明できるとはどういうことか」「証明が違う・同じとはどういうことか」を考えるための道具
  - 構成要素:
    - 命題・判断の (形式的) 定義
    - 導出 (証明を形式化したもの) の定義
  - 自然演繹，シーケント計算，ヒルベルト流公理系，などの「流儀」の違う証明体系
- 自然演繹 (natural deduction):
  - ゲンツェン (Gerhard Gentzen, 1909–1945) によって作られた証明体系の (流儀の) ひとつ
  - 人間の推論過程を自然に表現することが狙い
  - 導出の定義に使われる規則に特徴: 導入規則と除去規則 (後述)

ここでは，命題の構成要素 (**論理結合子** (*logical connective*) という) として「ならば (`->`)」だけを考える論理から始めて，`Induction.v` までの内容で扱われる論理 (だいたい，スコットの Logic of Computable Functions と呼ばれる体系に相当する) に拡張する．さらに,「かつ」「または」「〜を満たすものが存在する」といった，その他の論理結合子や量化子に拡張する.[1]

Coq の論理体系を含め，この講義で導入する論理は**直観主義論理** (*intuitionistic logic*) と呼ばれるもので，命題の意味を (真/偽を表す)1/0 で与えるふつうの論理 (これを**古典論理** (*classical logic*) と呼ぶ) とは異なるものである．

[1]多くの教科書では，まず「かつ」「または」といった「ならば」以外の論理結合子を導入した**命題論理** (*propositional logic*) を示し,「任意の〜」「ある〜について」といった量化子を導入した**述語論理** (*predicate logic*)，自然数についての述語論理である**算術** (*arithmetic*) へと徐々にコマを進めるが，ここでは，違うルートを取る．

1 (「ならば」に関する) 直観主義論理

1.1 命題，文脈

これまでの Coq の演習で扱った命題は，`n + 2 = 3` や `n * m = m * n` といった，ふたつの式を等号で結んだものを最小の単位として，それらを `->` でつないだり，全称量化 `forall` をつけたりすることで，より複雑な命題を構成してきた．

命題論理では「ならば」「かつ」「または」といった，命題をつなげてより大きな命題を作る論理結合子に着目するため，最小の命題 (これを**原子命題** (*atomic proposition*)) としてどんなものを考えるかについては特に何も約束ごとがない．このため，以下しばらく，原子命題としては単に `A`, `B`, `C` など (これを**命題変数** (*propositional variable*) という) があることとし，原子命題を指す記号としては $\boldsymbol{p}$ を使う．

まず，以下に「ならば」の論理における命題と文脈の定義を示す．

$$
\begin{array}{lcl}
\boldsymbol{H}\ (\text{仮定名}) & \in & \{\texttt{H1}, \texttt{H2}, \texttt{IH}, \ldots\} \\
\boldsymbol{p}, \boldsymbol{q}\ (\text{原子命題}) & \in & \{\texttt{A}, \texttt{B}, \texttt{C}, \ldots\} \\
\boldsymbol{P}, \boldsymbol{Q}\ (\text{命題}) & ::= & \boldsymbol{p} \\
 & | & \boldsymbol{P}\ \texttt{->}\ \boldsymbol{Q} \\
 & & \\
\boldsymbol{\Gamma}(\text{文脈}) & ::= & \bullet \\
 & | & \boldsymbol{\Gamma},\ \boldsymbol{H} : \boldsymbol{P}
\end{array}
$$

- 命題は原子命題を「ならば」を意味する `->` でつないだものである．`->` は右結合である．つまり，`A -> B -> C` は `A -> (B -> C)` を意味する．また，`(A -> B) -> C` を `A -> B -> C` と略記してはいけない．
- 文脈は $\boldsymbol{H} : \boldsymbol{P}$ という形の列であり，$\boldsymbol{P}$ が (名前 $\boldsymbol{H}$ で) 仮定されていることを示している．列中の仮定の名前は相異なる．
- 文脈 $\boldsymbol{\Gamma}$ に現れる仮定の名前の集合を $dom(\boldsymbol{\Gamma})$ と書く．例えば，$\boldsymbol{\Gamma}$ が $\bullet, \texttt{H1} : \texttt{A -> B}, \texttt{H2} : \texttt{A}$ の時，$dom(\boldsymbol{\Gamma}) = \{\texttt{H1}, \texttt{H2}\}$ である．
- 文脈の先頭の $\bullet$ (と，それに続くコンマ) は省略する．つまり，$\bullet, \texttt{H1} : \texttt{A -> B}, \texttt{H2} : \texttt{A}$ は `H1 : A -> B, H2 : A` と略記する．
- 多くの教科書では，文脈における明示的な仮定の名前 (「$\boldsymbol{H}$:」の部分) を省略するが，ここでは Coq の記法に近づけている．

1.2 判断と導出規則

単純型付ラムダ計算などで，型付け関係 $\boldsymbol{\Gamma} \vdash \boldsymbol{M} : \boldsymbol{T}$ を規則を使って定義したのと同様に,「文脈 $\boldsymbol{\Gamma}$ のもとで命題 $\boldsymbol{P}$ が成立する」ということを示す関係 $\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ (これを**判断** (*judgment*) とも呼ぶ) を規則を使って定義する．

自然演繹の特徴は，命題の構成要素 (「ならば」，全称量化，等号) ひとつにつき，それが結論に現れる規則 (**導入規則** (*introduction rule*) と呼ぶ．しばしば規則名に I (introduction の頭文字) をつけ

る)，前提に現れる規則 (**除去規則** (*elimination rule*) と呼ぶ．しばしば規則名に E (elimination の頭文字) をつける) が与えられるところにある．導入規則は，論理結合子によって構成される命題が成立する一般的な条件を示しており，除去規則はその命題を出発点として，そこからどんな結論が導けるかを示している．

$$\frac{(\boldsymbol{H} : \boldsymbol{P} \in \boldsymbol{\Gamma})}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}} \quad \text{(Assumption)}$$

$$\frac{\boldsymbol{\Gamma},\ \boldsymbol{H} : \boldsymbol{P} \vdash \boldsymbol{Q} \qquad \boldsymbol{H} \notin \mathit{dom}(\boldsymbol{\Gamma})}{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \texttt{->} \boldsymbol{Q}} \quad \text{(->I)}$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \texttt{->} \boldsymbol{Q} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{P}}{\boldsymbol{\Gamma} \vdash \boldsymbol{Q}} \quad \text{(->E)}$$

Assumption 規則は導入規則でも除去規則でもない特殊な規則で，文脈で仮定された命題は成立することを示している．->I 規則は,「ならば」の導入規則である．(他に仮定された $\boldsymbol{\Gamma}$ に加えて) $\boldsymbol{P}$ を仮定した文脈で $\boldsymbol{Q}$ が成立するのであれば (規則の前提),「$\boldsymbol{P}$ -> $\boldsymbol{Q}$」が成立しているとしてよい (規則の結論)，と理解できる．->E 規則は,「ならば」の除去規則である．$\boldsymbol{P}$ -> $\boldsymbol{Q}$ と $\boldsymbol{P}$ がともに成立するのであれば，帰結として $\boldsymbol{Q}$ が成立しているとしてよいことを示している．

1.3 判断の導出例

1. 判断 $\vdash$ A -> B -> A の導出:

$$\dfrac{\dfrac{\dfrac{}{\texttt{H1 : A, H2 : B} \vdash \texttt{A}}\ \textsc{Assumption}}{\texttt{H1 : A} \vdash \texttt{B -> A}}\ \texttt{->I}}{\vdash \texttt{A -> B -> A}}\ \texttt{->I}$$

2. 判断 $\vdash$ (A -> B -> C) -> (A -> B) -> (A -> C) の導出:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{A -> B -> C}}\ \textsc{Asm} \quad \dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{A}}\ \textsc{Asm}}{\boldsymbol{\Gamma} \vdash \texttt{B -> C}}\ \texttt{->E} \quad \dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{A -> B}}\ \textsc{Asm} \quad \dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{A}}\ \textsc{Asm}}{\boldsymbol{\Gamma} \vdash \texttt{B}}\ \texttt{->E}}{\boldsymbol{\Gamma} \vdash \texttt{C}}\ \texttt{->E}}{\texttt{H1 : A -> B -> C, H2 : A -> B} \vdash \texttt{A -> C}}\ \texttt{->I}}{\texttt{H1 : A -> B -> C} \vdash \texttt{(A -> B) -> (A -> C)}}\ \texttt{->I}}{\vdash \texttt{(A -> B -> C) -> (A -> B) -> (A -> C)}}\ \texttt{->I}$$

ただし $\boldsymbol{\Gamma}$ は H1 : A -> B -> C, H2 : A -> B, H3 : A とする．Assumption は Asm と略している．また A -> B -> C は A -> (B -> C) のことであることに注意．

2　単純型付ラムダ計算に関する論理

「ならば」の論理を拡張して，だいたい Induction.v までで扱った範囲に相当する論理体系を与える．ここでの原子命題は，単純型付ラムダ計算の項を使って $\boldsymbol{M_1}$ = $\boldsymbol{M_2}$ という形で与えられる．同時に，全称量化を導入する．

$$
\begin{array}{lcl}
\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{H} & \in & \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{H1}, \mathtt{H2}, \ldots\} \\
\\
\boldsymbol{S}, \boldsymbol{T} & ::= & \mathtt{nat} \\
& | & \mathtt{bool} \\
& | & \boldsymbol{S}\ \mathtt{->}\ \boldsymbol{T} \\
\\
\boldsymbol{P}, \boldsymbol{Q} & ::= & \boldsymbol{M_1}\ \mathtt{=}\ \boldsymbol{M_2} \\
& | & \boldsymbol{P}\ \mathtt{->}\ \boldsymbol{Q} \\
& | & \forall \boldsymbol{x} : \boldsymbol{T}, \boldsymbol{P} \\
\\
\boldsymbol{\Gamma} & ::= & \bullet \\
& | & \boldsymbol{\Gamma},\ \boldsymbol{x} : \boldsymbol{T} \\
& | & \boldsymbol{\Gamma},\ \boldsymbol{H} : \boldsymbol{P}
\end{array}
$$

命題中に現れる項は型がついているものでないといけないことはもちろんだが，各項に型がついていても命題としては一貫性が取れていないこともある．例えば，

$$\forall \mathtt{b} : \mathtt{bool}, \forall \mathtt{n} : \mathtt{nat}, \mathtt{b} = \mathtt{n}$$

のような命題は，そもそも意味がないものとして排除する必要がある．そのためにまず命題に対する型付けを考える必要がある．また，文脈に並んでいる命題も型付けできるものである必要がある．

命題に対する型付け関係は

$$\boldsymbol{\Gamma} \vdash \boldsymbol{P} : \mathtt{Prop}$$

と書く．$\boldsymbol{P}$ が正しい/証明できる，という意味の $\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ との区別に注意せよ．例えば，$\vdash \forall \mathtt{n} : \mathtt{nat}, \mathtt{n} = \mathtt{S\ O}$ は導出でき (そうも) ないが，$\vdash \forall \mathtt{n} : \mathtt{nat}, \mathtt{n} = \mathtt{S\ O} : \mathtt{Prop}$ は成り立つ．命題の型付けは単に $\boldsymbol{P}$ が**真偽を議論する対象になりうる**命題であることを示しているだけで，その内容が正しいかどうかとは関係がない．(文献によっては，両者の区別のために，$\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ を $\boldsymbol{\Gamma} \vdash \boldsymbol{P}\ \mathit{true}$ と記述することもある．)

文脈に並んだ命題が型付けできるものであることは

$$\vdash \boldsymbol{\Gamma}\ \mathit{ok}$$

でしめす．

2.1　命題の型付け規則

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{M_1} : \boldsymbol{T} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{M_2} : \boldsymbol{T}}{\boldsymbol{\Gamma} \vdash \boldsymbol{M_1}\ \mathtt{=}\ \boldsymbol{M_2} : \mathtt{Prop}} \quad (=\mathrm{P})$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} : \mathtt{Prop} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{Q} : \mathtt{Prop}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \mathtt{->} \boldsymbol{Q} : \mathtt{Prop}} \tag{->P}$$

$$\frac{\boldsymbol{\Gamma}, \boldsymbol{x} : \boldsymbol{T} \vdash \boldsymbol{P} : \mathtt{Prop}}{\boldsymbol{\Gamma} \vdash \forall \boldsymbol{x} : \boldsymbol{T}, \boldsymbol{P} : \mathtt{Prop}} \tag{∀P}$$

等号については，両辺が同じ型の項でないといけないこと，量化子 $\forall \boldsymbol{x} : \boldsymbol{T}, \boldsymbol{P}$ については，$\boldsymbol{P}$ が変数 $\boldsymbol{x}$ が $\boldsymbol{T}$ である下で命題であることが求められている．

文脈については，以下のような規則を考える．

$$\frac{}{\vdash \bullet \ ok} \tag{EnvEmp}$$

$$\frac{\vdash \boldsymbol{\Gamma} \ ok \qquad \boldsymbol{x} \notin dom(\boldsymbol{\Gamma})}{\vdash \boldsymbol{\Gamma}, \boldsymbol{x} : \boldsymbol{T} \ ok} \tag{EnvTy}$$

$$\frac{\vdash \boldsymbol{\Gamma} \ ok \qquad \boldsymbol{H} \notin dom(\boldsymbol{\Gamma}) \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{P} : \mathtt{Prop}}{\vdash \boldsymbol{\Gamma}, \boldsymbol{H} : \boldsymbol{P} \ ok} \tag{EnvProp}$$

最後の規則は，文脈に新たな仮定 $\boldsymbol{H} : \boldsymbol{P}$ をおく場合には，$\boldsymbol{P}$ がそれまでの仮定のもとで型付けされる命題でなければならないことを意味している．文脈に並ぶ順番は大事で，$\vdash \mathtt{n} : \mathtt{nat}, \mathtt{H} : \mathtt{n} = \mathtt{n}\ ok$ は導出できるが，$\vdash \mathtt{H} : \mathtt{n} = \mathtt{n}, \mathtt{n} : \mathtt{nat}\ ok$ は導出できない．

以下で，判断 $\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ のための (「ならば」の論理に加える) 規則を与えるが，$\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ の形の判断が出現するところでは，常に $\vdash \boldsymbol{\Gamma}\ ok$ かつ $\boldsymbol{\Gamma} \vdash \boldsymbol{P} : \mathtt{Prop}$ であることを前提とする．

2.2 全称量化に関する規則

$$\frac{\boldsymbol{\Gamma},\ \boldsymbol{x} : \boldsymbol{T} \vdash \boldsymbol{P} \qquad (\boldsymbol{x} \notin dom(\boldsymbol{\Gamma}))}{\boldsymbol{\Gamma} \vdash \forall \boldsymbol{x} : \boldsymbol{T}, \boldsymbol{P}[\boldsymbol{x}]} \tag{∀I}$$

$$\frac{\boldsymbol{\Gamma} \vdash \forall \boldsymbol{x} : \boldsymbol{T}, \boldsymbol{P}[\boldsymbol{x}] \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{M} : \boldsymbol{T}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}[\boldsymbol{M}]} \tag{∀E}$$

導入規則 ∀I は，$\boldsymbol{P}$ 中に現れる変数 $\boldsymbol{x}$ について，その型が $\boldsymbol{T}$ であること以外に特に仮定を置かずに $\boldsymbol{P}$ が成り立つ時,「任意の $\boldsymbol{x}$ について $\boldsymbol{P}$ が成り立つ」といってよいことを示している．また，除去規則は,「任意の $\boldsymbol{x}$ について $\boldsymbol{P}$ が成り立つ」ならば，実際，任意に選んだ具体的な型 $\boldsymbol{T}$ の項 $\boldsymbol{M}$ について $\boldsymbol{P}$ が成り立つことを示している．$\boldsymbol{P}[\boldsymbol{x}]$ と $\boldsymbol{P}[\boldsymbol{M}]$ はラムダ計算の簡約規則などと同様に，$\boldsymbol{P}[\boldsymbol{x}]$ 中に $\boldsymbol{x}$ が現れること，その $\boldsymbol{x}$ に $\boldsymbol{M}$ を代入した命題が $\boldsymbol{P}[\boldsymbol{M}]$ であることを示す記法である．

2.3 「等しさ」に関する導出規則

$$\frac{\boldsymbol{M_1} \longleftrightarrow \boldsymbol{M_2} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{M_1} : \boldsymbol{T} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{M_2} : \boldsymbol{T}}{\boldsymbol{\Gamma} \vdash \boldsymbol{M_1} \mathtt{=} \boldsymbol{M_2}} \tag{=I}$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{M_1} \mathtt{=} \boldsymbol{M_2} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{P}[\boldsymbol{M_1}]}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}[\boldsymbol{M_2}]} \tag{=E}$$

$\longleftrightarrow$ はラムダ計算で登場した，簡約関係から導かれる項の間の同値関係である．導入規則 =I は，簡約を通じて等しい ($\longleftrightarrow$)，ということが (この論理での) 等しさの定義であることを示している．除去規則=E は，命題中の項は等しい項で置き換えてよいことを示している．

2.4 自然数に関する導出規則

$$\frac{\Gamma \vdash \mathtt{S}(M_1) = \mathtt{S}(M_2)}{\Gamma \vdash M_1 = M_2} \quad (\textsc{InjS})$$

$$\frac{\Gamma \vdash \mathtt{0} = \mathtt{S}(M)}{\Gamma \vdash P} \quad (\textsc{ContraNat})$$

$$\frac{\Gamma \vdash P[\mathtt{0}] \qquad \Gamma,\ y : \mathtt{nat},\ H : P[y] \vdash P[\mathtt{S}(y)]}{\Gamma \vdash \forall x : \mathtt{nat}, P[x]} \quad (\textsc{IndNat})$$

規則 InjS は，S の injectivity (1-to-1 であること) を，規則 ContraNat は 0 が 1 以上の自然数とは決して等しくない (等しい，という結論が出てきたら矛盾なので，そこから何でも導ける) ことを示している．規則 IndNat は数学的帰納法の原理を推論規則の形式で表したものである．規則中の $H : P[y]$ が帰納法の仮定を表している．

2.5 真偽値に関する導出規則

$$\frac{\Gamma \vdash \mathtt{true} = \mathtt{false}}{\Gamma \vdash P} \quad (\textsc{ContraBool})$$

$$\frac{\Gamma \vdash P[\mathtt{true}] \qquad \Gamma \vdash P[\mathtt{false}]}{\Gamma \vdash \forall x : \mathtt{bool}, P[x]} \quad (\textsc{IndBool})$$

規則 ContraBool は ContraNat と同様である．IndBool は真偽値についての場合わけの原理である．

2.6 導出の例

ここでは，$\forall$x : nat,P は $\forall$x,P と省略する．また，規則名 Assumption も Asm と省略する．

1. 判断 $\vdash \forall$x,S 0 = x -> x + S(S 0) = S(S(S 0)) の導出:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathtt{S\ 0} = \mathtt{x}}\ \textsc{Asm} \quad \dfrac{\mathtt{S\ 0 + S(S\ 0)} \leftrightarrow \mathtt{S(S(S\ 0))} \quad \Gamma \vdash \mathtt{S\ 0 + S(S\ 0)} : \mathtt{nat}}{\Gamma \vdash \mathtt{S\ 0 + S(S\ 0)} = \mathtt{S(S(S\ 0))}}\ =\!\mathrm{E} \quad \Gamma \vdash \mathtt{S(S(S\ 0))} : \mathtt{nat}}{\mathtt{x : nat},\ \mathtt{H : S\ 0 = x} \vdash \mathtt{x + S(S\ 0) = S(S(S\ 0))}}\ =\!\mathrm{I}}{\mathtt{x : nat} \vdash \mathtt{S\ 0 = x -> x + S(S\ 0) = S(S(S\ 0))}}\ \mathtt{->}\mathrm{I}}{\vdash \forall \mathtt{x,S\ 0 = x -> x + S(S\ 0) = S(S(S\ 0))}}\ \forall\mathrm{I}$$

ただし M_1 + M_2 は足し算を表すラムダ項 *plus* を使った *plus* M_1 M_2 の略記であり，Γ は x : nat, H : S 0 = x である．

2. 判断 $\vdash$ `∀x,x + 0 = x` の導出:

$$\dfrac{\dfrac{\vdots}{\texttt{0 + 0} \longleftrightarrow \texttt{0}} \quad \dfrac{\vdots}{\vdash \texttt{0 + 0 : nat}} \quad \dfrac{\vdots}{\vdash \texttt{0 : nat}}}{\vdash \texttt{0 + 0 = 0}}\ =\!\text{I} \qquad \dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{x + 0 = x}}\ \textsc{Asm} \quad \dfrac{\vdots}{\boldsymbol{\Gamma} \vdash \texttt{S x + 0 = S(x+0)}}}{\boldsymbol{\Gamma} \vdash \texttt{S x + 0 = S x}}\ \begin{matrix}=\!\text{I}\\ =\!\text{E}\end{matrix}}{\vdash \texttt{∀x,x + 0 = x}}\ \textsc{IndNat}$$

ただし，$\boldsymbol{\Gamma}$ は `x : nat`, `IH : x + 0 = x` である．

3. 命題 $\boldsymbol{P}$ を `∀x,∀y,x = y -> y = x`，文脈 $\boldsymbol{\Gamma}$ を `sym` : $\boldsymbol{P}$, `z : nat`, `H : S(S(0)) = S(S(0)) * z` とする．この時，判断 $\boldsymbol{\Gamma} \vdash$ `S(S(0)) * z = S(S(0))` の導出:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{∀x,∀y,x = y -> y = x}}\ \textsc{Asm}}{\boldsymbol{\Gamma} \vdash \texttt{∀y,S(S(0)) = y -> y = S(S(0))}}\ \forall\text{E}}{\boldsymbol{\Gamma} \vdash \texttt{S(S(0)) = S(S(0)) * z -> S(S(0)) * z = S(S(0))}}\ \forall\text{E} \qquad \dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{S(S(0)) = S(S(0)) * z}}\ \begin{matrix}\textsc{Asm}\\ \texttt{->E}\end{matrix}}{\boldsymbol{\Gamma} \vdash \texttt{S(S(0)) * z = S(S(0))}}$$

(∀E 規則のふたつめの前提 $\boldsymbol{\Gamma} \vdash$ `S(S(S 0)) : nat` と $\boldsymbol{\Gamma} \vdash$ `z : nat` との導出は省略している．)

4. 判断 $\vdash$ `∀x,x + 0 = 0 -> x = 0` の導出:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma},\ \texttt{H' : S(x) + 0 = 0} \vdash \texttt{S(x) + 0 = 0}}\ \textsc{Asm} \quad \dfrac{\vdots}{\boldsymbol{\Gamma},\ \texttt{H' : S(x) + 0 = 0} \vdash \texttt{S(x) + 0 = S(x+0)}}}{\boldsymbol{\Gamma},\ \texttt{H' : S(x) + 0 = 0} \vdash \texttt{0 = S(x+0)}}\ \begin{matrix}=\!\text{I}\\ =\!\text{E}\end{matrix}}{\boldsymbol{\Gamma},\ \texttt{H' : S x + 0 = 0} \vdash \texttt{S x = 0}}\ \textsc{ContraNat}}{\boldsymbol{\Gamma} \vdash \texttt{S x + 0 = 0 -> S x = 0}}\ \texttt{->I}}{\vdash \texttt{∀x,x + 0 = 0 -> x = 0}}\ \textsc{IndNat}$$

$$\dfrac{\dfrac{\dfrac{\vdots}{\texttt{H : 0 + 0 = 0} \vdash \texttt{0 = 0}}\ =\!\text{I}}{\vdash \texttt{0 + 0 = 0 -> 0 = 0}}\ \texttt{->I}}$$

ただし $\boldsymbol{\Gamma}$ は `x : nat`, `IH : x + 0 = 0 -> x = 0` である．

3 Coq と自然演繹

Coq で行う証明は，ゴールとなる命題 $\boldsymbol{P}$ が与えられた時，$\boldsymbol{\Gamma} \vdash \boldsymbol{P}$ (ただし，$\boldsymbol{\Gamma}$ はそれまでに証明をした定理の集まり) を結論とする導出木を構成するプロセスに他ならない．例えば，上の $\vdash$ `∀x,x = S(0) -> x + S(S(0)) = S(S(S(0)))` の導出を考えてみよう．Coq で

```
Theorem foo : forall x:nat, x = S 0 -> x + S(S 0) = S(S(S 0)).
```

と打った場合，人間に課せられたのは，

$$\dfrac{\dfrac{?}{\vdash \texttt{∀x,x = S(0) -> x + S(S(0)) = S(S(S(0)))}}\ ??}{}$$

という (根しかない) 木の ? 部分を埋めて導出木を完成させることである．

タクティックは，規則を組み合わせて (それまでに部分的に構成された) 木を「成長させる」働きをしている．例えば，`intros` タクティックは ∀I 規則や `->I` 規則に対応しており，上の木に対して，

`intros x.` を実行すると木は成長して

$$\dfrac{\dfrac{?}{\texttt{x : nat} \vdash \texttt{S(O) = x -> x + S(S(O)) = S(S(S(O)))}}\ ??}{\vdash \forall \texttt{x,S(O) = x -> x + S(S(O)) = S(S(S(O)))}}\ \forall\mathrm{I}$$

になる．まだ未完成の部分にある命題 (と文脈) が新しいゴールである．

タクティックは複数の規則を組み合わせを表現していることもある．例えば，文脈に $\texttt{H} : \boldsymbol{M_1} = \boldsymbol{M_2}$ が含まれており，ゴールが $\vdash \boldsymbol{P}[\boldsymbol{M_2}]$ だった時に `rewrite <- H.` を実行すると，

$$\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \boldsymbol{M_1} = \boldsymbol{M_2}}\ \textsc{Assumption} \quad \dfrac{?}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}[\boldsymbol{M_1}]}\ ??}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}[\boldsymbol{M_2}]}\ =\mathrm{E}$$

と木を成長させ，新しいゴールは $\boldsymbol{P}$ 中の $\boldsymbol{M_2}$ を $\boldsymbol{M_1}$ に書き換えた $\boldsymbol{P}[\boldsymbol{M_1}]$ になる．

最後に `apply` タクティックについて見てみよう．$\boldsymbol{P}$ を

$$\forall \texttt{q : nat,}\forall \texttt{r : nat,q = r -> plus q q = mult r 2}$$

とし，$\texttt{H} : \boldsymbol{P}$ が文脈にあり，ゴールが

$$\texttt{plus 2 2 = mult x 2}$$

だったとする.[2]ここで `apply H.` を実行するのは，以下のように導出木を成長させることと考えられる．

$$\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}}\ \textsc{Assumption}}{\boldsymbol{\Gamma} \vdash \forall \texttt{r : nat,(2 = r) -> plus 2 2 = mult r 2}}\ \forall\mathrm{E} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \texttt{O : nat}}\ \text{T-}\textsc{Zero}}{\boldsymbol{\Gamma} \vdash \texttt{S O : nat}}\ \text{T-}\textsc{Succ}}{\boldsymbol{\Gamma} \vdash \texttt{2 : nat}}\ \text{T-}\textsc{Succ}}{\boldsymbol{\Gamma} \vdash \texttt{x : nat}}\ \text{T-}\textsc{Var}}{\boldsymbol{\Gamma} \vdash \texttt{(2 = x) -> plus 2 2 = mult x 2}}\ \forall\mathrm{E} \quad \dfrac{?}{\boldsymbol{\Gamma} \vdash \texttt{2 = x}}\ ??}{\boldsymbol{\Gamma} \vdash \texttt{plus 2 2 = mult x 2}}\ \texttt{->}\mathrm{E}$$

この導出木の未完成の部分は `apply H.` を実行した時の新しいゴールである，`2 = x` になっている．

タクティックと自然演繹の規則には大体，以下のような対応関係がある．

タクティック	規則
`intros`	->I または ∀I
`reflexivity`	=I
`apply`	Assumption, ∀E, ->E の組み合わせ
`rewrite`	Assumption と =E の組合せ
`induction`, `destruct`	IndNat または IndBool
`injection`	InjS, Assumption, ->E の組み合わせ
`discriminate`	ContraNat, Contrabool, Assumption の組み合わせ

[2]アラビア数字表記の `2` は，`S(S O)` の略記である．

4 「ならば」以外の論理結合子

以下に「かつ」(連言),「または」(選言) といった「ならば」以外の論理結合子，そして矛盾という特殊な命題に関連した導出規則を示す．**??**節の論理に対して，これらを加えた論理を命題論理と呼ぶ.

$$
\begin{array}{lll}
\boldsymbol{H}\ (\text{仮定名}) & \in & \{\texttt{H1}, \texttt{H2}, \texttt{IH}, \ldots, \} \\
\boldsymbol{p}, \boldsymbol{q}\ (\text{原子命題}) & & \\
 & & \\
\boldsymbol{P}, \boldsymbol{Q}\ (\text{命題}) & ::= & \ldots \\
 & & \mid \quad \boldsymbol{P} \wedge \boldsymbol{Q} \\
 & & \mid \quad \boldsymbol{P} \vee \boldsymbol{Q} \\
 & & \mid \quad \bot
\end{array}
$$

- $\boldsymbol{P} \wedge \boldsymbol{Q}$ は「$\boldsymbol{P}$ かつ $\boldsymbol{Q}$」を表す.
- $\boldsymbol{P} \vee \boldsymbol{Q}$ は「$\boldsymbol{P}$ または $\boldsymbol{Q}$」を表す.
- $\bot$ は矛盾を表す.

4.1 命題の型付け規則

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} : \texttt{Prop} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{Q} : \texttt{Prop}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \wedge \boldsymbol{Q} : \texttt{Prop}} \quad (\wedge\text{P})$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} : \texttt{Prop} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{Q} : \texttt{Prop}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \vee \boldsymbol{Q} : \texttt{Prop}} \quad (\vee\text{P})$$

$$\frac{}{\boldsymbol{\Gamma} \vdash \bot : \texttt{Prop}} \quad (\bot\text{P})$$

4.2 導出規則

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \qquad \boldsymbol{\Gamma} \vdash \boldsymbol{Q}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \wedge \boldsymbol{Q}} \quad (\wedge\text{-I})$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \wedge \boldsymbol{Q}}{\boldsymbol{\Gamma} \vdash \boldsymbol{P}} \quad (\wedge\text{-E1})$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \wedge \boldsymbol{Q}}{\boldsymbol{\Gamma} \vdash \boldsymbol{Q}} \quad (\wedge\text{-E2})$$

$$\frac{\boldsymbol{\Gamma} \vdash \boldsymbol{P} \wedge \boldsymbol{Q} \qquad \boldsymbol{\Gamma},\ \boldsymbol{H_1} : \boldsymbol{P},\ \boldsymbol{H_2} : \boldsymbol{Q} \vdash \boldsymbol{R} \qquad (\boldsymbol{H_1} \neq \boldsymbol{H_2} \text{ かつ } \boldsymbol{H_1}, \boldsymbol{H_2} \notin \mathit{dom}(\boldsymbol{\Gamma}))}{\boldsymbol{\Gamma} \vdash \boldsymbol{R}} \quad (\wedge\text{-E3})$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \quad (\vee\text{-I1})$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \quad (\vee\text{-I2})$$

$$\frac{\Gamma \vdash P \vee Q \qquad \Gamma, H_1 : P \vdash R \qquad \Gamma, H_2 : Q \vdash R \qquad (H_1, H_2 \notin dom(\Gamma))}{\Gamma \vdash R} \quad (\vee\text{-E})$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash P} \quad (\bot\text{-E})$$

- 「かつ」の除去規則は，通常 E1, E2 のふたつを考えることが多いが，それらの代わりとして E3 を採用してもよい．
- 「または」の除去規則は一種の場合分けの推論を表している．
- 矛盾については導入規則がなく除去規則のみがある．
- 否定 $\neg P$ は P -> $\bot$ の略記とみなす．

5 存在量化子

存在量化子 (*existential quantifier*) とは，「P を満たす x が存在する」といういわゆる存在命題を表すための記法で，$\exists x : T, P$ という形をとる．存在量化子に関する命題の型付け規則と推論規則は以下のように与えることができる．

$$\frac{\Gamma, x : T \vdash P : \mathtt{Prop}}{\Gamma \vdash \exists x : T, P : \mathtt{Prop}} \quad (\exists\text{P})$$

$$\frac{\Gamma \vdash M : T \qquad \Gamma \vdash P[M]}{\Gamma \vdash \exists x : T, P[x]} \quad (\exists\text{-I})$$

$$\frac{\Gamma \vdash \exists x : T, P[x] \qquad \Gamma, x : T, H : P[x] \vdash Q}{\Gamma \vdash Q} \quad (\exists\text{-E})$$

導入規則 ∃-I は，「M が P を満たす」が導出できた時には，「P を満たす x が存在する」ことを導出してよいことを示している．$P[M]$ は $\exists x : T, P$ の P 中の x に具体的な項 M を入れた命題を表している．

一方，除去規則 ∃-E は，存在命題から別の命題 Q を導くためには，x が $P[x]$ を満たすという文脈/仮定のもとで Q を示せればよい (十分である) ことを示している．結論から $\Gamma \vdash Q : \mathtt{Prop}$ が暗黙に仮定されているが，これは，大雑把には，Q に x が現れてはいけない，つまり，$x : T$ と $H : P[x]$ は，Q を導くための局所的な仮定であることを意味している．

判断の導出例

1. $\mathtt{x} : \boldsymbol{T} \vdash \boldsymbol{P}[\mathtt{x}] : \mathtt{Prop}$ かつ $\mathtt{x} : \boldsymbol{T} \vdash \boldsymbol{Q}[\mathtt{x}] : \mathtt{Prop}$ とする．この時，判断

$$\vdash (\forall \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \mathtt{->} \boldsymbol{Q}[\mathtt{x}]) \mathtt{->} (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}]) \mathtt{->} (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{Q}[\mathtt{x}])$$

の導出は以下のようになる．

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma} \vdash \exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}]}\ \textsc{Asm} \qquad \dfrac{\dfrac{\dfrac{}{\boldsymbol{\Gamma}' \vdash \forall \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \mathtt{->} \boldsymbol{Q}[\mathtt{x}]}\ \textsc{Asm}}{\boldsymbol{\Gamma}' \vdash \boldsymbol{P}[\mathtt{y}] \mathtt{->} \boldsymbol{Q}[\mathtt{y}]}\ \forall\mathrm{E} \qquad \dfrac{}{\boldsymbol{\Gamma}' \vdash \boldsymbol{P}[\mathtt{y}]}\ \textsc{Asm}}{\boldsymbol{\Gamma}' \vdash \boldsymbol{Q}[\mathtt{y}]}\ \mathtt{->}\mathrm{E}}{\boldsymbol{\Gamma}' \vdash \exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{Q}[\mathtt{x}]}\ \exists\mathrm{I}}{\boldsymbol{\Gamma} \vdash \exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{Q}[\mathtt{x}]}\ \exists\mathrm{E}}{\mathtt{H1} : \forall \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \mathtt{->} \boldsymbol{Q}[\mathtt{x}] \vdash (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}]) \mathtt{->} (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{Q}[\mathtt{x}])}\ \mathtt{->}\mathrm{I}}{\vdash (\forall \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \mathtt{->} \boldsymbol{Q}[\mathtt{x}]) \mathtt{->} (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}]) \mathtt{->} (\exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{Q}[\mathtt{x}])}\ \mathtt{->}\mathrm{I}$$

ただし

$$\begin{aligned} \boldsymbol{\Gamma} &= \mathtt{H1} : \forall \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \mathtt{->} \boldsymbol{Q}[\mathtt{x}], \mathtt{H2} : \exists \mathtt{x} : \boldsymbol{T}, \boldsymbol{P}[\mathtt{x}] \\ \boldsymbol{\Gamma}' &= \boldsymbol{\Gamma}, \mathtt{y} : \boldsymbol{T}, \mathtt{H3}, \boldsymbol{P}[\mathtt{y}] \end{aligned}$$

である．