

論理学を（Lean で）つくる

PART.1 古典命題論理 Ver.0

SnO2WMaN (Mashu Noguchi)

2025/12/27

メタ情報

- この文書のリポジトリは <https://github.com/SnO2WMaN/Logic101> です。誤植や訂正などは Issues から連絡してください。
- この文書がコンパイルされた時点のコミット ID は 4e6f72e であり、その時点でのリポジトリの URL は <https://github.com/SnO2WMaN/Logic101/tree/4e6f72e> です。最新版とファイルが更新されている可能性に注意してください。
- この文書は [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) の下でライセンスされています。

1. はじめに

現状のところ、この文書は Ver.0 です。単に古典命題論理の強完全性定理 (定理 53) の証明を Lean で形式化しただけのものであり、解説や導入が非常に不十分です。一応ほとんどの事実に対して証明をつけていますが、ある程度既知っている人向けの内容になっています。実験的に、普通の数式表示 (e.g. φ) とソースコード上の表示 (e.g. φ) を混在させて書いていますが、読みづらいかもかもしれません。ぜひフィードバックをいただければ幸いです。

2. 論理式の導入

この節は Logic101/CPL/Formula.lean に置いてある。

定義 1 (Formula): 型 α の要素を命題変数とした論理式 Formula α を次のように定義する。

1. Formula.atom: 型 α の要素 a は論理式である。論理式であることは $\#a$ と書く。
2. Formula.bot: 論理定項 \perp は論理式である。
3. Formula.and: φ と ψ が論理式であれば、 $\varphi \wedge \psi$ も論理式である。
4. Formula.or: φ と ψ が論理式であれば、 $\varphi \vee \psi$ も論理式である。
5. Formula.imp: φ と ψ が論理式であれば、 $\varphi \rightarrow \psi$ も論理式である。

```
6 inductive Formula ( $\alpha$  : Type u)
7   | atom :  $\alpha$   $\rightarrow$  Formula  $\alpha$ 
8   | bot : Formula  $\alpha$ 
9   | and : Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$ 
10  | or : Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$ 
11  | imp : Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$   $\rightarrow$  Formula  $\alpha$ 
12  deriving DecidableEq
```

```
16 prefix:90 "#"  $\Rightarrow$  Formula.atom
17 notation:max "⊥"  $\Rightarrow$  Formula.bot
18 infixr:85 "  $\rightarrow$  "  $\Rightarrow$  Formula.imp
19 infixl:84 "  $\wedge$  "  $\Rightarrow$  Formula.and
```

```
20 infixl:83 " ∨ " ⇒ Formula.or
```

便宜上、 $\sim\varphi$ を $\varphi \rightarrow \perp$ と定義する。また、 \top を $\sim\perp$ と定義する。

```
22 @[grind] def neg (φ : Formula α) : Formula α := φ → ⊥
23 prefix:90 "~" ⇒ Formula.neg
24
25 @[grind] def top : Formula α := ~⊥
26 notation:max "⊤" ⇒ Formula.top
```

命題変数が列挙可能なら論理式は適当に列挙可能である。この事実は完全性定理の証明に必要となるが、その証明は普通の数学上では自明なものであるし、コード上でも技術的に面倒なだけでつまらないため、証明は省略する。

補題 2 (`Formula.instEncodable`): α が `Encodable` であるなら `Formula α` も `Encodable` である。すなわち、次を満たす。

1. `Formula α` から自然数への全域関数 `encode : Formula α → ℕ` が存在する。
2. 自然数から `Formula α` への部分関数 `decode : ℕ → Option (Formula α)` が存在する。
3. 任意の $\varphi : \text{Formula } \alpha$ について、`decode (encode φ)` は存在し、かつその値は φ と等しい。

```
34 def toNat : Formula α → ℕ
```

```
41 def ofNat : ℕ → Option (Formula α)
```

```
84 lemma ofNat_toNat : ∀ (φ : Formula α), ofNat (toNat φ) = some φ
```

```
89 instance : Encodable (Formula α) where
90   encode := toNat
91   decode := ofNat
92   encodek := ofNat_toNat
```

定義 3 (`Theory`): `Formula α` の集合を理論 `Theory α` と呼ぶ。

```
98 abbrev Theory (α) := Set (Formula α)
```

3. 証明体系の導入

証明体系として Hilbert 流を採用する。やや冗長な量の公理を導入しているが、これは今後直観主義命題論理などを扱うことを企画してのことである。

「証明」と「証明できる」ことを区別していることに注意する。前者はオブジェクトであり、後者は命題である。

定義 4 (`Proof`): `Formula α` 上の証明 `Proof φ` を帰納的に定義する。 `Proof φ` を $\vdash! \varphi$ と書く。

1. 次の論理式たちは公理であり、それ単体で証明になる。

- `andElimL`: $\varphi \wedge \psi \rightarrow \varphi$

- andElimR: $\varphi \wedge \psi \rightarrow \psi$
- andIntro: $\varphi \rightarrow \psi \rightarrow (\varphi \wedge \psi)$
- orIntroL: $\varphi \rightarrow (\varphi \vee \psi)$
- orIntroR: $\psi \rightarrow (\varphi \vee \psi)$
- orElim: $(\varphi \rightarrow \chi) \rightarrow (\psi \rightarrow \chi) \rightarrow (\varphi \vee \psi \rightarrow \chi)$
- impS: $(\varphi \rightarrow \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)$
- impK: $\varphi \rightarrow \psi \rightarrow \varphi$
- efq: $\perp \rightarrow \varphi$
- lem: $\varphi \vee \sim\varphi$

2. mdp: $\varphi \rightarrow \psi$ の証明 と φ の証明から, ψ の証明が得られる.

```

5 inductive Proof : Formula  $\alpha$   $\rightarrow$  Type*
6 | andIntro! { $\varphi \psi$ } : Proof $  $\varphi \rightarrow \psi \rightarrow (\varphi \wedge \psi)$ 
7 | andElimL! { $\varphi \psi$ } : Proof $  $(\varphi \wedge \psi) \rightarrow \varphi$ 
8 | andElimR! { $\varphi \psi$ } : Proof $  $(\varphi \wedge \psi) \rightarrow \psi$ 
9 | orIntroL! { $\varphi \psi$ } : Proof $  $\varphi \rightarrow (\varphi \vee \psi)$ 
10 | orIntroR! { $\varphi \psi$ } : Proof $  $\psi \rightarrow (\varphi \vee \psi)$ 
11 | orElim! { $\varphi \psi \chi$ } : Proof $  $(\varphi \vee \psi) \rightarrow (\varphi \rightarrow \chi) \rightarrow (\psi \rightarrow \chi) \rightarrow \chi$ 
12 | impS! { $\varphi \psi \chi$ } : Proof $  $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)$ 
13 | impK! { $\varphi \psi$ } : Proof $  $\varphi \rightarrow \psi \rightarrow \varphi$ 
14 | mdp! { $\varphi \psi$ } : Proof  $(\varphi \rightarrow \psi) \rightarrow$  Proof  $\varphi \rightarrow$  Proof  $\psi$ 
15 | efq! { $\varphi$ } : Proof $  $\perp \rightarrow \varphi$ 
16 | lem! { $\varphi$ } : Proof $  $\varphi \vee \sim\varphi$ 
17 prefix:40 "⊢!" "  $\Rightarrow$  Proof

```

定義 5 (Provable, Unprovable) : φ の証明 $\vdash!$ φ が存在するとき φ は証明可能 Provable φ である
 と言い, $\vdash \varphi$ と書く.

```

19 def Provable ( $\varphi$  : Formula  $\alpha$ ) := Nonempty ( $\vdash!$   $\varphi$ )
20 prefix:40 "⊢!" "  $\Rightarrow$  Provable

```

証明可能ではないとき, φ は証明不能 Unprovable φ であると言い, $\not\vdash \varphi$ と書く.

```

22 abbrev Unprovable ( $\varphi$  : Formula  $\alpha$ ) :=  $\neg(\vdash \varphi)$ 
23 prefix:40 "⊢!" "  $\Rightarrow$  Unprovable

```

明らかにわかることだが, 証明不能であることは証明が存在しないことと同値である.

以降では証明という概念を陽には扱わず, 証明可能性で議論することにする.

補題 6 (Provable.impId) : $\vdash \varphi \rightarrow \varphi$

証明 :

1. $\varphi \rightarrow (\varphi \rightarrow \varphi) \rightarrow \varphi$ は impK より証明可能.
2. $(\varphi \rightarrow (\varphi \rightarrow \varphi) \rightarrow \varphi) \rightarrow (\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \rightarrow \varphi$ は impS より証明可能.
3. 1. と 2. より mdp から $(\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \rightarrow \varphi$ は証明可能.
4. $\varphi \rightarrow \varphi \rightarrow \varphi$ は impK より証明可能.
5. 3. と 4. より mdp から $\varphi \rightarrow \varphi$ が証明可能.

```

62 lemma impId :  $\vdash \varphi \rightarrow \varphi :=$  by
63   have h1 :  $\vdash \varphi \rightarrow (\varphi \rightarrow \varphi) \rightarrow \varphi :=$  impK;
64   have h2 :  $\vdash (\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \rightarrow \varphi :=$  impS;
65   have h3 :  $\vdash (\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \rightarrow \varphi :=$  mdp h2 h1;
66   have h4 :  $\vdash \varphi \rightarrow \varphi \rightarrow \varphi :=$  impK;
67   exact mdp h3 h4;

```

□

証明可能という概念は技術的に扱いにくいので、もうすこし扱いやすくするために演繹という概念を導入する。

定義 7 (Deduction): Theory α である Γ から φ への演繹 Deduction $\Gamma \varphi$ を帰納的に定義する。

Deduction $\Gamma \varphi$ を $\Gamma \vdash! \varphi$ と書く。

1. $\varphi \in \Gamma$ ならば, $\Gamma \vdash! \varphi$ を得て良い。
2. 証明 $\vdash! \varphi$ から $\Gamma \vdash! \varphi$ を得て良い。
3. $\Gamma \vdash! \varphi \rightarrow \psi$ と $\Gamma \vdash! \varphi$ から, $\Gamma \vdash! \psi$ を得て良い。

```

75 | of_context! { $\Gamma \varphi$ } :  $\varphi \in \Gamma \rightarrow$  Deduction  $\Gamma \varphi$ 
76 | of_proof! { $\Gamma \varphi$ } :  $\vdash! \varphi \rightarrow$  Deduction  $\Gamma \varphi$ 
77 | mdp! { $\Gamma \varphi \psi$ } : Deduction  $\Gamma (\varphi \rightarrow \psi) \rightarrow$  Deduction  $\Gamma \varphi \rightarrow$  Deduction  $\Gamma \psi$ 
78 infix:40 "  $\vdash!$  "  $\Rightarrow$  Deduction

```

定義 8 (Deducible): Γ から φ への演繹 $\Gamma \vdash! \varphi$ が存在するとき, φ は Γ から演繹可能 Deducible $\Gamma \varphi$ であると言い, $\Gamma \vdash \varphi$ と書く。

```

81 infix:40 "  $\vdash$  "  $\Rightarrow$  Deducible

```

そうでないとき, φ は Γ から演繹不能 Underivable $\Gamma \varphi$ であると言い, $\Gamma \not\vdash \varphi$ と書く。

```

84 infix:40 "  $\not\vdash$  "  $\Rightarrow$  Undeducible

```

補題 9 (Deducible.weakening): $\Gamma \subseteq \Delta$ かつ $\Gamma \vdash \varphi$ ならば, $\Delta \vdash \varphi$

証明: 演繹可能性の帰納法による。

(of_context) $\varphi \in \Gamma$ ならば仮定を使えば $\varphi \in \Delta$ であるから, $\Delta \vdash! \varphi$ である。

(of_proof) $\vdash! \varphi$ ならば $\Delta \vdash! \varphi$ である。

(mdp) $\Gamma \vdash! \varphi \rightarrow \psi$ と $\Gamma \vdash! \varphi$ ならば, 帰納法の仮定から $\Delta \vdash! \varphi \rightarrow \psi$ と $\Delta \vdash! \varphi$ であるから, mdp を使えば $\Delta \vdash! \psi$ である。

```

189 lemma weakening (h $\Gamma\Delta$  :  $\Gamma \subseteq \Delta$ ) (h :  $\Gamma \vdash \varphi$ ) :  $\Delta \vdash \varphi :=$  by
190   induction h using Deducible.rec with
191   | of_context h  $\Rightarrow$ 
192     apply of_context;
193     exact h $\Gamma\Delta$  h;
194   | of_provable h  $\Rightarrow$ 

```

```

195   apply of_provable;
196   exact h;
197   | mdp d1 d2 ih1 ih2 =>
198     apply mdp;
199     . apply ih1 hΓΔ;
200     . apply ih2 hΓΔ;

```

□

補題 9 の系として直ちに次が得られる.

系 10 (`Deducible.weakening_insert`): $\Gamma \vdash \varphi$ ならば, `insert ψ $\Gamma \vdash \varphi$`

```

205 @[grind =]
206 theorem deduction_theorem : (insert  $\varphi$   $\Gamma$ )  $\vdash \psi \leftrightarrow \Gamma \vdash (\varphi \rightarrow \psi) := by$ 
```

演繹という概念を導入したのは演繹定理と呼ばれる便利な 定理 11 を証明するためである.

定理 11 (演繹定理, `Deducible.deduction_theorem`): `insert φ $\Gamma \vdash \psi$` と `$\Gamma \vdash \varphi \rightarrow \psi$` は同値である.

証明 :

(\implies) `insert φ $\Gamma \vdash \psi$` 上の帰納法.

(`of_context`) $\psi \in \text{insert } \varphi \Gamma$ ならば場合分けする.

- $\psi = \varphi$ のとき, $\Gamma \vdash \varphi \rightarrow \varphi$ であり, `impId` より成り立つ.
- $\psi \in \Gamma$ のとき, $\Gamma \vdash \psi$ であるから, `afortiori` より $\Gamma \vdash \varphi \rightarrow \psi$ である.

(`of_provable`) $\vdash \psi$ ならば証明可能性上の `afortiori` より $\vdash \varphi \rightarrow \psi$ であり, `of_provable` より $\Gamma \vdash \varphi \rightarrow \psi$ である.

(`mdp`) 帰納法の仮定より $\Gamma \vdash \varphi \rightarrow \psi \rightarrow \chi$ と $\Gamma \vdash \varphi \rightarrow \psi$ であるから, `impS` より $\Gamma \vdash \varphi \rightarrow \chi$ である.

(\impliedby) $\Gamma \vdash \varphi \rightarrow \psi$ なら系 10 より `insert φ $\Gamma \vdash \varphi \rightarrow \psi$` である. これと仮定から `mdp` を使えば `insert φ $\Gamma \vdash \psi$` である.

```

206 theorem deduction_theorem : (insert  $\varphi$   $\Gamma$ )  $\vdash \psi \leftrightarrow \Gamma \vdash (\varphi \rightarrow \psi) := by
207   constructor;
208   . intro h;
209     generalize e $\Delta$  : insert  $\varphi$   $\Gamma = \Delta$  at h;
210     induction h using Deducible.rec with
211       | of_context h =>
212         subst e $\Delta$ ;
213         simp at h;
214         rcases h with (rfl | h);
215         . exact impId;
216         . apply afortiori
217           apply of_context h;
218       | of_provable h =>
219         apply of_provable;$ 
```

```

220     exact Provable.afortiori h;
221   | mdp _ _ ih1 ih2 =>
222     subst eΔ;
223     replace ih1 := ih1 (by grind);
224     replace ih2 := ih2 (by grind);
225     apply ruleImpS ih1 ih2;
226   . intro h;
227     apply mdp (weakening_insert h);
228     apply of_context;
229     simp;

```

□

補題 12 (`Deducible.ruleImpTrans`): $\Gamma \vdash \varphi \rightarrow \psi$ および $\Gamma \vdash \psi \rightarrow \chi$ ならば, $\Gamma \vdash \varphi \rightarrow \chi$

証明:

1. 定理 11 より仮定 $\Gamma \vdash \varphi \rightarrow \psi$ は `insert φ Γ ⊢ ψ` と同値である.
2. もう片方の仮定 $\Gamma \vdash \psi \rightarrow \chi$ は系 10 より `insert φ Γ ⊢ ψ → χ` である.
3. 1. と 2. より `mdp` を使えば `insert φ Γ ⊢ χ` である.
4. 再び 定理 11 より $\Gamma \vdash \varphi \rightarrow \chi$ である.

```

231 lemma ruleImpTrans : Γ ⊢ φ → ψ → Γ ⊢ ψ → χ → Γ ⊢ φ → χ := by
232   intro h1 h2;
233   replace h1 : insert φ Γ ⊢ ψ := deduction_theorem.mpr h1;
234   replace h2 : insert φ Γ ⊢ ψ → χ := weakening_insert h2;
235   exact deduction_theorem.mp $ mdp h2 h1;

```

□

補題 13 (`Deducible.ruleEFQNeg`): $\Gamma \vdash \sim\varphi$ ならば, $\Gamma \vdash \varphi \rightarrow \psi$

証明: $\sim\varphi$ が $\varphi \rightarrow \perp$ であることに注意すれば `efq` と 補題 12 より直ちに従う.

```

237 lemma ruleEFQNeg : Γ ⊢ ~φ → Γ ⊢ φ → ψ := λ h => ruleImpTrans h efq

```

□

補題 14 (`Deducible.ruleDNE`): $\Gamma \vdash \sim\sim\varphi$ ならば, $\Gamma \vdash \varphi$

証明: `ruleOrElim` と `lem` より $\Gamma \vdash \varphi \rightarrow \varphi$ と $\Gamma \vdash \sim\varphi \rightarrow \varphi$ を言えば良い. 前者は `impId` から, 後者は $\Gamma \vdash \sim\varphi$ と 補題 13 より従う.

```

239 lemma ruleDNE (h : Γ ⊢ ~~φ) : Γ ⊢ φ := by
240   apply ruleOrElim lem;
241   . show Γ ⊢ φ → φ;
242     exact impId;
243   . apply ruleEFQNeg h;

```

□

4. 意味論の導入

この節は `Logic101/CPL/Semantics.lean` に置いてある.

定義 15 (Valuation): 写像 $\alpha \rightarrow \text{Prop}$ を付値 Valuation と呼ぶ.

```
6 abbrev Valuation ( $\alpha$ ) :=  $\alpha \rightarrow \text{Prop}$ 
```

定義 16 (Forces): 論理式 $\varphi : \text{Formula } \alpha$ が付値 $V : \text{Valuation } \alpha$ によって強制される Forces $V \varphi$ という関係を次のように定義する. $\text{Forces } V \varphi$ を $V \Vdash \varphi$ と書く.

1. $V \Vdash \#a$ は付値の値 $V a$ である.
2. $V \Vdash \perp$ は成立しない.
3. $V \Vdash \varphi \wedge \psi$ は $V \Vdash \varphi$ かつ $V \Vdash \psi$ のとき成立する.
4. $V \Vdash \varphi \vee \psi$ は $V \Vdash \varphi$ または $V \Vdash \psi$ のとき成立する.
5. $V \Vdash \varphi \rightarrow \psi$ は $V \Vdash \varphi$ ならば $V \Vdash \psi$ のとき成立する.

```
9 def Forces (V : Valuation  $\alpha$ ) : Formula  $\alpha \rightarrow \text{Prop}$ 
10 | #a      => V a
11 |  $\perp$      => False
12 |  $\varphi \wedge \psi$  => Forces V  $\varphi \wedge$  Forces V  $\psi$ 
13 |  $\varphi \vee \psi$  => Forces V  $\varphi \vee$  Forces V  $\psi$ 
14 |  $\varphi \rightarrow \psi$  => Forces V  $\varphi \rightarrow$  Forces V  $\psi$ 
15 infix:50 "  $\Vdash$  " => Forces
```

$\sim\varphi$ は $\varphi \rightarrow \perp$ であり, τ は $\sim\perp$ であることから, 定義から次が成り立つ.

補題 17:

- $V \Vdash \sim\varphi$ は $V \Vdash \varphi$ でないときに成立する.
- $V \Vdash \tau$ は常に成立する.

```
19 @[grind .] lemma forces_top : V  $\Vdash$   $\tau$  := by grind;
20 @[grind =] lemma forces_neg : V  $\Vdash$   $\sim\varphi \leftrightarrow \neg(V \Vdash \varphi)$  := by grind;
```

定義 18 (Valid): 論理式 φ が妥当 Valid φ であるとは, 任意の付値 V に対して $V \Vdash \varphi$ が成立することである. Valid φ を $\models \varphi$ と書く. あるいは φ がトートロジー Tautology であるとも言う.

```
24 def Valid ( $\varphi : \text{Formula } \alpha$ ) : Prop :=  $\forall V, V \Vdash \varphi$ 
25 prefix:50 " $\models$ " => Valid
26 alias Tautology := Valid
```

定義 19 (ConsequenceOn): 付値 V と理論 Γ と論理式 φ について, $\forall \psi \in \Gamma, V \Vdash \psi$ ならば $V \Vdash \varphi$ が成立するとき, V において Γ が φ から帰結する ConsequenceOn $V \Gamma \varphi$ という. ConsequenceOn $V \Gamma \varphi$ を $\Gamma \models[V] \varphi$ と書く.

```
28 @[grind .]
29 def ConsequenceOn (V : Valuation  $\alpha$ ) ( $\Gamma : \text{Theory } \alpha$ ) ( $\varphi : \text{Formula } \alpha$ ) : Prop := ( $\forall \psi \in \Gamma, V \Vdash \psi$ )  $\rightarrow V \Vdash \varphi$ 
30 notation  $\Gamma \models[V] \varphi \Rightarrow \text{ConsequenceOn } V \Gamma \varphi$ 
```

定義 20 (Consequence) : 任意の付値 V について $\text{ConsequenceOn } V \Gamma \varphi$ が成立するとき, Γ が φ から帰結する $\text{Consequence } \Gamma \varphi$ という. $\text{Consequence } \Gamma \varphi$ を $\Gamma \vDash \varphi$ と書く.

```
34 def Consequence (Γ : Theory α) (φ : Formula α) : Prop := ∀ V, Γ ⊨ [V] φ
35 infix:50 " ⊨ " => Consequence
```

論理式が妥当であることと空な理論から意味論的に帰結することは同値である.

補題 21 ($\text{iff_valid_forall_forces}$) : $\vDash \varphi$ は $\emptyset \vDash \varphi$ と同値である.

```
38 lemma iff_valid_forall_forces : ⊨ φ ↔ ∅ ⊨ φ := by simp [Valid, Consequence,
ConsequenceOn]
```

定理 22 (健全性, sound) : $\Gamma \vdash \varphi$ ならば $\Gamma \vDash \varphi$

証明 : 演繹可能性の帰納法による.

(of_context) 任意の付値 V を取り, $\forall \psi \in \Gamma, V \vDash \psi$ を仮定する. このとき $\varphi \in \Gamma$ だから $V \vDash \varphi$ が成り立ち, V の任意性より $\Gamma \vDash \varphi$ が成り立つ.

(andIntroL など) 各公理などに対して $V \vDash \varphi \wedge \psi \rightarrow \varphi$ や $V \vDash \varphi \rightarrow \psi \rightarrow \varphi$ などが成り立つことを確認すれば良い.

(mdp) 任意の付値 V と $\forall \psi \in \Gamma, V \vDash \psi$ を仮定する. 帰納法の仮定より $\Gamma \vDash \varphi \rightarrow \psi$ と $\Gamma \vDash \varphi$ が成り立つから, 特に $V \vDash \varphi \rightarrow \psi$ と $V \vDash \varphi$ が成り立つ. よって $V \vDash \psi$ が成り立つ.

```
40 theorem sound : Γ ⊢ φ → Γ ⊨ φ := by
41   rintro h;
42   induction h using Deducible.rec_provable with
43     | orElim =>
44       intro V hV;
45       grind [Forces];
46     | mdp _ _ ihφψ ihφ =>
47       intro V hV;
48       exact ihφψ V hV $ ihφ V hV;
49     | _ => grind;
```

□

系 23 (sound_provable) : $\vdash \varphi$ ならば $\vDash \varphi$

```
51 lemma sound_provable : ⊢ φ → ⊨ φ := by
```

証明 : 定理 22 で Γ として \emptyset を取り, 補題 21 を用いれば良い. □

系 24 (consistent) : $\not\vdash \perp$. すなわち, どのような論理式でも証明可能である訳では無いという意味で証明体系は矛盾していない.

証明：さもなくば系 23 から $\vDash \perp$ である。適当な付値 v を取れば $v \vDash \perp$ となり、定義に反する。

```
56 theorem consistent :  $\not\vdash$  ( $\perp$  : Formula  $\alpha$ ) := by
57   by_contra h;
58   apply sound_provable h ( $\lambda$  _  $\Rightarrow$  True);
```

□

健全性の逆方向である完全性を示すためには様々な準備が必要である。

定義 25 (Theory.Inconsistent, Theory.Consistent) : 理論 Γ が矛盾している

Theory.Inconsistent Γ とは $\Gamma \vdash \perp$ であることを指す。そうでないとき Γ は無矛盾

Theory.Consistent Γ であると言う。

```
69 abbrev Inconsistent ( $\Gamma$  : Theory  $\alpha$ ) : Prop :=  $\Gamma \vdash \perp$ 
70
71 abbrev Consistent ( $\Gamma$  : Theory  $\alpha$ ) : Prop :=  $\neg$ ( $\Gamma$ .Inconsistent)
```

補題 26 (Theory.iff_inconsistent_insert, Theory.iff_consistent_insert) :

1. insert φ Γ が無矛盾であることは $\Gamma \vdash \sim\varphi$ と同値である。
2. 逆に insert $\sim\varphi$ Γ が矛盾していることは $\Gamma \not\vdash \sim\varphi$ と同値である。

証明：演繹定理(定理 11)の主張そのものである。

```
73 @[grind =] lemma iff_inconsistent_insert : (insert  $\varphi$   $\Gamma$ ).Inconsistent  $\leftrightarrow$   $\Gamma \vdash \sim\varphi$  :=
74   deduction_theorem
75 @[grind =] lemma iff_consistent_insert : (insert  $\varphi$   $\Gamma$ ).Consistent  $\leftrightarrow$   $\Gamma \not\vdash \sim\varphi$  :=
76   deduction_theorem.not
```

□

補題 27 (Theory.either_unprovable_of_consistent) : Γ が無矛盾であるならば、 $\Gamma \not\vdash \varphi$ または $\Gamma \not\vdash \sim\varphi$ のいずれかが成り立つ。

証明：もしそうでないとすると $\Gamma \vdash \varphi$ かつ $\Gamma \vdash \sim\varphi$ である。したがって mdp から $\Gamma \vdash \perp$ であり、 Γ は無矛盾であるという仮定に反する。

```
87 lemma either_unprovable_of_consistent ( $\Gamma$ _consis :  $\Gamma$ .Consistent) ( $\varphi$ ) :  $\Gamma \not\vdash \varphi \vee \Gamma \not\vdash$ 
88    $\sim\varphi$  := by
89   contrapose!  $\Gamma$ _consis;
90   obtain  $\langle h\varphi, hn\varphi \rangle$  :=  $\Gamma$ _consis;
91   apply mdp hn $\varphi$  h $\varphi$ ;
```

□

補題 28 : $\Gamma \not\vdash \varphi$ ならば、insert $\sim\varphi$ Γ は無矛盾である。

証明：対偶を示す。insert $\sim\varphi$ Γ が矛盾しているとすると、補題 26 より $\Gamma \vdash \sim\sim\varphi$ であり、ruleDNE より $\Gamma \vdash \varphi$ である。

```

105 lemma consistent_insert_neg_of_unprovable (h :  $\Gamma \not\vdash \varphi$ ) : (insert ( $\sim\varphi$ )
106  $\Gamma$ ).Consistent := by
107   contrapose! h;
108   apply ruleDNE;
109   apply iff_inconsistent_insert.mp h;

```

□

定義 29 (Maximal): 理論 Γ が極大 Maximal Γ であるとは、任意の論理式 φ について $\varphi \in \Gamma$ または $\sim\varphi \in \Gamma$ が成り立つことである。

```

110 abbrev Maximal ( $\Gamma$  : Theory  $\alpha$ ) : Prop :=  $\forall \varphi, \varphi \in \Gamma \vee \sim\varphi \in \Gamma$ 

```

定義 30 (MaximalConsistentTheory): 無矛盾かつ極大な理論 Γ : Theory α を極大無矛盾理論 MaximalConsistentTheory α と呼ぶ。

```

115 abbrev MaximalConsistentTheory ( $\alpha$ ) := {  $\Gamma$  : Theory  $\alpha$  //  $\Gamma$ .Consistent  $\wedge$   $\Gamma$ .Maximal }

```

次の定理 31 は Lindenbaum 補題と呼ばれる重要な定理であり、完全性定理の証明において中核を成す。この定理の証明には Lean においてメタのレベルで排中律を用いていることに注意されたい。

定理 31 (lindenbaum): Γ が無矛盾であるならば、ある極大無矛盾理論 Δ が存在して $\Gamma \subseteq \Delta$ を満たす。

```

220 lemma lindenbaum ( $\Gamma$  : Theory  $\alpha$ ) ( $\Gamma$ .consis : Theory.Consistent  $\Gamma$ ) :  $\exists \Delta$  :
    MaximalConsistentTheory  $\alpha, \Gamma \subseteq \Delta$ .1 := by

```

定理 31 の証明にはいくつかの補助的な定義や補題が必要である。

定義 32 (lindenbaum_expand): 理論 Γ と論理式 φ について、lindenbaum_expand $\Gamma \varphi$ を次のように定義する。

- insert $\varphi \Gamma$ が無矛盾であるならば、lindenbaum_expand $\Gamma \varphi =$ insert $\varphi \Gamma$
- そうでなければ、lindenbaum_expand $\Gamma \varphi = \Gamma$

```

122 open Classical in
123 def lindenbaum_expand ( $\Gamma$  : Theory  $\alpha$ ) ( $\varphi$  : Formula  $\alpha$ ) : Theory  $\alpha$  := if
    Theory.Consistent (insert  $\varphi \Gamma$ ) then insert  $\varphi \Gamma$  else  $\Gamma$ 

```

注意 33: ここで、insert $\varphi \Gamma$ が無矛盾であるかは现阶段では決定不能 (Decidable ではない) であるが、Classical を用いて決定可能としていることに注意せよ。

補題 34 (consistent_lindenbaum_expand): Γ が無矛盾であるならば、lindenbaum_expand $\Gamma \varphi$ も無矛盾である。

証明: 定義及び仮定から直ちに従う。

```

125 @[grind =>]
126 lemma consistent_lindenbaum_expand (Γ_consist : Theory.Consistent Γ) :
127   Theory.Consistent (lindenbaum_expand Γ φ) := by
128     dsimp [lindenbaum_expand];
129     split <;> assumption;

```

□

以下 α は列挙可能だとすると、補題 2 から論理式は列挙可能であるとしてよい。

定義 35 (`lindenbaum_indexed`): 自然数で添字付けられた理論の族 `lindenbaum_indexed : $\mathbb{N} \rightarrow$ Theory α` を次のように定義する。

- `lindenbaum_indexed 0 = Γ`
- `lindenbaum_indexed (n + 1)` はもし n に対応する論理式 φ が存在するとき `lindenbaum_expand (lindenbaum_indexed n) φ` とし、存在しないときは `lindenbaum_indexed n` とする。

```

132 def lindenbaum_indexed (Γ : Theory α) :  $\mathbb{N} \rightarrow$  Theory α
133   | 0      => Γ
134   | n + 1 =>
135     match (decode n) with
136     | some φ => lindenbaum_expand (lindenbaum_indexed Γ n) φ
137     | none   => lindenbaum_indexed Γ n

```

補題 36 (`consistent_lindenbaum_indexed`): Γ が無矛盾であるならば、任意の自然数 n について `lindenbaum_indexed n` も無矛盾である。

証明: 仮定および補題 34 から n の帰納法によって示される。

```

142 lemma consistent_lindenbaum_indexed (Γ_consist : Theory.Consistent Γ) :
143   Theory.Consistent (lindenbaum_indexed Γ n) := by
144     induction n with
145     | zero => assumption;
146     | succ n ih =>
147       dsimp [lindenbaum_indexed];
148       split;
149       . grind [consistent_lindenbaum_expand];
150       . assumption;

```

□

補題 37 (`monotone_lindenbaum_indexed`): 任意の自然数 $n \leq m$ について、`lindenbaum_indexed n \subseteq lindenbaum_indexed m` が成り立つ。

証明: $n = m$ のとき自明. $n \leq m$ で成立すると仮定して $n \leq m + 1$ の場合を示す. 明らかに定義から `lindenbaum_indexed m \subseteq lindenbaum_indexed (m + 1)` であるから \subseteq の推移律と帰納法の仮定から従う.

```

151 lemma monotonic_lindenbaum_indexed (h : n ≤ m) : lindenbaum_indexed Γ n ⊆
152   lindenbaum_indexed Γ m := by
153   induction h with
154   | refl => simp;
155   | step h ih =>
156     trans;

```

```

156     . exact ih;
157     . dsimp [lindenbaum_indexed, lindenbaum_expand];
158     split;
159     . split <;> grind;
160     . grind;

```

□

定義 38 (`lindenbaum_maximal`): 理論 `lindenbaum_maximal` をすべての自然数 n における `lindenbaum_indexed n` の和集合として定義する.

```

162 def lindenbaum_maximal (Γ : Theory α) : Theory α := U n, lindenbaum_indexed Γ n

```

補題 39: $\varphi \in \Gamma.lindenbaum_maximal$ はある自然数 n が存在して $\varphi \in lindenbaum_indexed\ n$ であることと同値である.

証明: 定義より明らか.

```

164 lemma iff_mem_lindenbaum_maximal : φ ∈ Γ.lindenbaum_maximal ↔ ∃ n, φ ∈
Γ.lindenbaum_indexed n := by
165   simp [lindenbaum_maximal, Set.mem_iUnion]

```

□

補題 40 (`subset_lindenbaum_indexed_lindenbaum_maximal`, `subset_lindenbaum_maximal`):

1. 任意の自然数 n について, `lindenbaum_indexed n` \subseteq `Γ.lindenbaum_maximal` が成り立つ.
2. 特に `Γ` \subseteq `Γ.lindenbaum_maximal` が成り立つ.

証明: 定義より明らか.

```

164 lemma iff_mem_lindenbaum_maximal : φ ∈ Γ.lindenbaum_maximal ↔ ∃ n, φ ∈
Γ.lindenbaum_indexed n := by
165   simp [lindenbaum_maximal, Set.mem_iUnion]
166
167 @[grind .]
168 lemma subset_lindenbaum_indexed_lindenbaum_maximal : lindenbaum_indexed Γ n ⊆
lindenbaum_maximal Γ := by
169   intro φ hφ;
170   apply iff_mem_lindenbaum_maximal.mpr;
171   use n;
172
173 lemma subset_lindenbaum_maximal : Γ ⊆ lindenbaum_maximal Γ :=
subset_lindenbaum_indexed_lindenbaum_maximal (n := 0)

```

□

補題 41 (`exists_deducible_maximal`): $\Gamma.lindenbaum_maximal \vdash \varphi$ ならば, ある自然数 n が存在して `lindenbaum_indexed n` $\vdash \varphi$.

証明: $\Gamma.lindenbaum_maximal \vdash \varphi$ に対して演繹可能性についての帰納法を行う.

(of_context) $\varphi \in \Gamma.lindenbaum_maximal$ ならば, 定義からある自然数 n が存在して $\varphi \in lindenbaum_indexed\ n$ である. よって $lindenbaum_indexed\ n \vdash \varphi$ である.

(of_proof) 証明可能であるならそもそも文脈無しで演繹可能だから, $n = 0$ とすれば良い.

(mdp) $\Gamma.lindenbaum_maximal \vdash \varphi \rightarrow \psi$ と $\Gamma.lindenbaum_maximal \vdash \varphi$ に対して 帰納法の仮定よりある n と m に対して $lindenbaum_indexed\ n \vdash \varphi \rightarrow \psi$ と $lindenbaum_indexed\ m \vdash \varphi$ が成り立つ. このとき所望の自然数は $\max\ n\ m$ とすればよい. なぜなら $n \leq \max\ n\ m$ かつ $m \leq \max\ n\ m$ であるから, 補題 37 より $lindenbaum_indexed\ (\max\ n\ m) \vdash \varphi \rightarrow \psi$ と $lindenbaum_indexed\ (\max\ n\ m) \vdash \varphi$ が成り立ち, mdp より $lindenbaum_indexed\ (\max\ n\ m) \vdash \psi$ が成り立つ.

```

175 lemma exists_deducible_maximal (h :  $\Gamma.lindenbaum\_maximal \vdash \varphi$ ) :  $\exists n,$ 
lindenbaum_indexed  $\Gamma\ n \vdash \varphi := by$ 
```

```

176 generalize e $\Delta$  :  $\Gamma.lindenbaum\_maximal = \Delta$  at h;
177 induction h using Deducible.rec with
178 | of_context h  $\Rightarrow$ 
179   subst e $\Delta$ ;
180   obtain  $\langle n, hn \rangle :=$  iff_mem_lindenbaum_maximal.mp h;
181   use n;
182   apply of_context hn;
183 | of_provable h  $\Rightarrow$ 
184   use 0;
185   exact of_provable h;
186 | mdp d1 d2 ih1 ih2  $\Rightarrow$ 
187   obtain  $\langle n, ih_1 \rangle :=$  ih1 e $\Delta$ ;
188   obtain  $\langle m, ih_2 \rangle :=$  ih2 e $\Delta$ ;
189   use max n m;
190   apply mdp;
191   . exact weakening (monotonic_lindenbaum_indexed (by omega)) ih1;
192   . exact weakening (monotonic_lindenbaum_indexed (by omega)) ih2;

```

□

補題 42 (consistent_lindenbaum_maximal) : Γ が無矛盾なら $\Gamma.lindenbaum_maximal$ も無矛盾である.

証明 : もしそうでないとすると, 補題 41 よりある自然数 n が存在して $lindenbaum_indexed\ n \vdash \perp$ であるが, それは 補題 36 に反する.

```

194 @[grind .]
195 lemma consistent_lindenbaum_maximal ( $\Gamma\_consis$  : Theory.Consistent  $\Gamma$ ) :
Theory.Consistent ( $\Gamma.lindenbaum\_maximal$ ) := by
196   by_contra hC;
197   obtain  $\langle \_, hn \rangle :=$  exists_deducible_maximal hC;
198   apply consistent_lindenbaum_indexed  $\Gamma\_consis$  hn;

```

□

補題 43 (maximal_lindenbaum_maximal) : Γ が無矛盾なら $\Gamma.lindenbaum_maximal$ は極大である.

証明 : 任意に論理式 φ を取る. 補題 42 より $\Gamma.lindenbaum_maximal$ は無矛盾だから, 補題 27 より $\Gamma.lindenbaum_maximal \not\vdash \varphi$ または $\Gamma.lindenbaum_maximal \not\vdash \sim\varphi$ が成り立つ.

$\Gamma.lindenbaum_maximal \not\vdash \varphi$ の場合を考える。このとき $insert(\sim\varphi)(\Gamma.lindenbaum_indexed(\text{encode}(\sim\varphi)))$ は無矛盾でないと仮定すると $\Gamma.lindenbaum_indexed(\text{encode}(\sim\varphi)) \vdash \sim\sim\varphi$ であるから、 $ruleDNE$ より $\Gamma.lindenbaum_indexed(\text{encode}(\sim\varphi)) \vdash \varphi$ が成り立つ。したがって補題 9 と補題 41 より $\Gamma.lindenbaum_maximal \vdash \varphi$ が成り立ち、仮定に反する。したがって $insert(\sim\varphi)(\Gamma.lindenbaum_indexed(\text{encode}(\sim\varphi)))$ は無矛盾であるから、 $\sim\varphi \in \Gamma.lindenbaum_indexed(\text{encode}(\sim\varphi) + 1)$ が成り立ち、補題 39 より $\sim\varphi \in \Gamma.lindenbaum_maximal$ が成り立つ。後者の場合も同様にして $\varphi \in \Gamma.lindenbaum_maximal$ が成り立つ。いずれにせよ $\varphi \in \Gamma.lindenbaum_maximal$ または $\sim\varphi \in \Gamma.lindenbaum_maximal$ が成り立つから $\Gamma.lindenbaum_maximal$ は極大である。

```

200 lemma maximal_lindenbaum_maximal (Γ_consist : Theory.Consistent Γ) : Theory.Maximal
    (Γ.lindenbaum_maximal) := by
201   intro φ;
    rcases Γ.lindenbaum_maximal.either_unprovable_of_consistent
202   (consistent_lindenbaum_maximal Γ_consist) φ with (hC | hC);
    . right;
203   apply iff_mem_lindenbaum_maximal.mpr;
204   use ((encode (~φ)) + 1);
205   suffices (insert (~φ) (Γ.lindenbaum_indexed (encode (~φ)))) .Consistent by
206     simp [lindenbaum_indexed, lindenbaum_expand, this];
207   apply iff_consistent_insert.mpr;
208   contrapose! hC;
209   apply weakening (by grind) (ruleDNE hC);
210   . left;
211   apply iff_mem_lindenbaum_maximal.mpr;
212   use ((encode φ) + 1);
213   suffices (insert φ (Γ.lindenbaum_indexed (encode φ))) .Consistent by
214     simp [lindenbaum_indexed, lindenbaum_expand, this];
215   apply iff_consistent_insert.mpr;
216   contrapose! hC;
217   apply weakening (by grind) hC;
218

```

□

これで定理 31 の証明の準備が整った。

証明 (定理 31) : 無矛盾な Γ を取り、その $\Gamma.lindenbaum_maximal$ を Δ とすれば、 $\Gamma \subseteq \Delta$ であることは補題 40 より従い、 Δ が無矛盾であることは補題 42 より従い、 Δ が極大であることは補題 43 より従う。

```

220 lemma lindenbaum (Γ : Theory α) (Γ_consist : Theory.Consistent Γ) : ∃ Δ :
    MaximalConsistentTheory α, Γ ⊆ Δ.1 := by
221   use ⟨Γ.lindenbaum_maximal, consistent_lindenbaum_maximal Γ_consist,
    maximal_lindenbaum_maximal Γ_consist⟩;
222   exact subset_lindenbaum_maximal;

```

□

注意 44 : 実際には、定理 31 は Mathlib にある Zorn の補題といくつかの集合に関する補題を用意すればこれよりも簡潔に証明できる。

次は極大無矛盾理論に対して性質を見ていく。以下 Γ は極大無矛盾理論であるとする。

補題 45 (`MaximalConsistentTheory.iff_mem_provable`): 論理式 φ について, $\varphi \in \Gamma$ は $\Gamma \vdash \varphi$ と同値である.

証明:

(\implies) `of_context` から明らか.

(\impliedby) `Maximal` であるから $\sim\varphi \notin \Gamma$ であることを言えば良い. もし $\sim\varphi \in \Gamma$ ならば $\Gamma \vdash \sim\varphi$ であるから仮定 $\Gamma \vdash \varphi$ と `mdp` より $\Gamma \vdash \perp$ が言えて, Γ が無矛盾であることに反する.

```
237 constructor;
238 . apply of_context;
239 . intro h;
240 suffices  $\sim\varphi \notin \Gamma.1$  by grind;
241 by_contra hC;
242 apply  $\Gamma$ .consistent;
243 apply mdp;
244 . show  $\Gamma.1 \vdash \sim\varphi$ ;
245 apply of_context hC;
246 . exact h;
```

□

補題 46 (`MaximalConsistentTheory.not_mem_bot`): $\perp \notin \Gamma$.

証明: 補題 45 と Γ の無矛盾性から直ちに従う.

```
248 @[simp, grind.] lemma not_mem_bot :  $\perp \notin \Gamma.1$  := by grind;
```

□

補題 47 (`MaximalConsistentTheory.iff_mem_neg`): $\sim\varphi \in \Gamma$ は $\varphi \notin \Gamma$ と同値である.

証明: 補題 27 と 補題 45 より直ちに従う.

```
250 @[grind =] lemma iff_mem_neg :  $\sim\varphi \in \Gamma.1 \leftrightarrow \varphi \notin \Gamma.1$  := by constructor <> grind;
```

□

補題 48 (`MaximalConsistentTheory.iff_not_mem_provable`): $\varphi \notin \Gamma$ は $\Gamma \vdash \sim\varphi$ と同値である.

証明: 補題 47 と 補題 45 より直ちに従う.

```
252 @[grind =] lemma iff_not_mem_provable :  $\varphi \notin \Gamma.1 \leftrightarrow \Gamma.1 \vdash \sim\varphi$  := by grind;
```

□

補題 49 (`MaximalConsistentTheory.iff_mem_imp`): $\varphi \rightarrow \psi \in \Gamma$ は 「 $\varphi \in \Gamma$ ならば $\psi \in \Gamma$ 」 と同値である.

証明:

(\implies) 補題 45 より $\Gamma \vdash \varphi \rightarrow \psi$, $\Gamma \vdash \varphi$ ならば $\Gamma \vdash \psi$ であることを証明すれば良い。これは mdp そのものである。

(\Leftarrow) 「 $\varphi \in \Gamma$ ならば $\psi \in \Gamma$ 」を仮定することは「 $\varphi \notin \Gamma$ または $\psi \in \Gamma$ 」のどちらかと同値である。

$\varphi \notin \Gamma$ と仮定すると 補題 47 より $\sim\varphi \in \Gamma$ であるから、 $\Gamma \vdash \sim\varphi$ であり、ruleEFQNeg より $\Gamma \vdash \varphi \rightarrow \psi$ が成り立つ。

$\psi \in \Gamma$ と仮定すると $\Gamma \vdash \psi$ であるから、afortiori より $\Gamma \vdash \varphi \rightarrow \psi$ が成り立つ。

いずれにせよ $\Gamma \vdash \varphi \rightarrow \psi$ が成り立つ。

```

255 lemma iff_mem_imp : ( $\varphi \rightarrow \psi$ )  $\in$   $\Gamma.1 \leftrightarrow (\varphi \in \Gamma.1 \rightarrow \psi \in \Gamma.1) := by
256   constructor;
257   . simp only [iff_mem_provable]
258   apply mdp;
259   . rintro h;
260   rcases not_or_of_imp h with (h $\varphi$  | h $\psi$ );
261   . replace h $\varphi$  := iff_mem_neg.mpr h $\varphi$ ;
262     simp only [iff_mem_provable] at h $\varphi$   $\vdash$ ;
263     exact ruleEFQNeg h $\varphi$ ;
264   . simp only [iff_mem_provable] at h $\psi$   $\vdash$ ;
265     exact afortiori h $\psi$ ;$ 
```

□

注意 50 : \Leftarrow において「 A ならば B 」を「 A でない、または B 」の同値だと見做すこと (not_or_of_imp) は Lean 上のメタ論理として古典論理でしか成立しない。

補題 51 (MaximalConsistentTheory.iff_mem_and) : $\varphi \wedge \psi \in \Gamma$ は $\varphi \in \Gamma$ かつ $\psi \in \Gamma$ と同値である。

証明 : 補題 45 より $\Gamma \vdash \varphi \wedge \psi$ は $\Gamma \vdash \varphi$ かつ $\Gamma \vdash \psi$ と同値であることを示せば良い。 \implies 方向は ruleAndElimL と ruleAndElimR から、 \Leftarrow 方向は ruleAndIntro から従う。

```

268 lemma iff_mem_and : ( $\varphi \wedge \psi$ )  $\in$   $\Gamma.1 \leftrightarrow \varphi \in \Gamma.1 \wedge \psi \in \Gamma.1 := by
269   simp only [iff_mem_provable]
270   constructor;
271   . intro h;
272     constructor;
273     . apply ruleAndElimL h;
274     . apply ruleAndElimR h;
275   . simp only [and_imp];
276     apply ruleAndIntro;$ 
```

□

補題 52 (MaximalConsistentTheory.iff_mem_or) : $\varphi \vee \psi \in \Gamma$ は $\varphi \in \Gamma$ または $\psi \in \Gamma$ と同値である。

証明 :

(\Rightarrow) $\varphi \vee \psi \in \Gamma$ なので $\Gamma \vdash \varphi \vee \psi$ である。 $\varphi \notin \Gamma$ かつ $\psi \notin \Gamma$ だと仮定すると、補題 48 より $\Gamma \vdash \sim\varphi$ かつ $\Gamma \vdash \sim\psi$ である。したがって ruleOrElim より $\Gamma \vdash \perp$ であり、 Γ が無矛盾であることに反する。 よって $\varphi \in \Gamma$ または $\psi \in \Gamma$ 。

(\Leftarrow) 補題 45 より $\Gamma \vdash \varphi$ または $\Gamma \vdash \psi$ ならば $\Gamma \vdash \varphi \vee \psi$ であることを示せば良い。 これは ruleOrIntroL または ruleOrIntroR から従う。

```

279 lemma iff_mem_or : ( $\varphi \vee \psi$ )  $\in$   $\Gamma$ .1  $\leftrightarrow$   $\varphi \in \Gamma$ .1  $\vee$   $\psi \in \Gamma$ .1 := by
280   constructor;
281   . intro h;
282     by_contra! hC;
283     obtain  $\langle$ h $\varphi$ , h $\psi$  $\rangle$  := hC;
284     replace h $\varphi$  :  $\uparrow\Gamma \vdash \sim\varphi$  := iff_not_mem_provable.mp h $\varphi$ ;
285     replace h $\psi$  :  $\uparrow\Gamma \vdash \sim\psi$  := iff_not_mem_provable.mp h $\psi$ ;
286     apply  $\Gamma$ .consistent;
287     apply ruleOrElim (iff_mem_provable.mp h) h $\varphi$  h $\psi$ ;
288   . simp only [iff_mem_provable];
289     rintro (h $\varphi$  | h $\psi$ );
290     . apply ruleOrIntroL;
291       exact h $\varphi$ ;
292     . apply ruleOrIntroR;
293       exact h $\psi$ ;

```

□

これで完全性定理を示す全ての準備が整った。

定理 53 (完全性, complete) : $\Gamma \models \varphi$ ならば $\Gamma \vdash \varphi$

証明 : 対偶を示す。 $\Gamma \not\models \varphi$ と仮定すると、補題 28 より $\text{insert } (\sim\varphi) \Gamma$ は無矛盾である。したがって定理 31 より極大無矛盾理論 Δ が存在して $\text{insert } (\sim\varphi) \Gamma \subseteq \Delta$ を満たすものが取れる。

このとき付値 V を $\#a$ が Δ に属するとき成立すると定める。すると、論理式の帰納法および補題 46, 補題 49, 補題 51, 補題 52 より $V \models \psi$ は $\psi \in \Delta$ と同値であることが示せる。

したがって、 $\Gamma \subseteq \Delta$ であるから $\forall \psi \in \Gamma, V \models \psi$ が成り立つ。また、 $\sim\varphi \in \Delta$ であるから $V \models \sim\varphi$ が成り立つ。すなわち $V \models \varphi$ でない。

よって $\Gamma \models \varphi$ は成り立たない。

```

297 theorem complete [Encodable  $\alpha$ ] :  $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$  := by
298   contrapose!;
299   intro h;
300   suffices  $\exists V, (\forall \psi \in \Gamma, V \models \psi) \wedge \neg V \models \varphi$  by simpa [Consequence, ConsequenceOn];
301
302   obtain  $\langle$  $\Delta$ , h $\Gamma\Delta$  $\rangle$  := (insert  $(\sim\varphi) \Gamma$ ).lindenbaum $ by grind
303   [Theory.consistent_insert_neg_of_unprovable];
304
305   let V : Valuation  $\alpha$  :=  $\lambda a \Rightarrow (\#a \in \Delta.1)$ ;
306   have hV :  $\forall \{\psi\}, V \models \psi \leftrightarrow \psi \in \Delta.1$  := by
307     intro  $\psi$ ;
308     induction  $\psi$  <;> grind
309   use V;
310   constructor;
311   . intro  $\gamma$  h $\gamma$ ;
312     apply hV.mpr;
313     apply h $\Gamma\Delta$ ;
314     grind;
315   . apply hV.not.mpr;

```

```

315   apply Δ.iff_mem_neg.mp;
316   apply hΓΔ;
317   grind;

```

□

5. コード全文

```

1  import Mathlib.Logic.Encodable.Basic
2
3  variable {α : Type u}
4
5
6  inductive Formula (α : Type u)
7  | atom : α → Formula α
8  | bot  : Formula α
9  | and  : Formula α → Formula α → Formula α
10 | or   : Formula α → Formula α → Formula α
11 | imp  : Formula α → Formula α → Formula α
12 deriving DecidableEq
13
14
15 namespace Formula
16
17 prefix:90 "#" ⇒ Formula.atom
18 notation:max "⊥" ⇒ Formula.bot
19 infixr:85 " → " ⇒ Formula.imp
20 infixl:84 " ∧ " ⇒ Formula.and
21 infixl:83 " ∨ " ⇒ Formula.or
22
23 @[grind] def neg (φ : Formula α) : Formula α := φ → ⊥
24 prefix:90 "~" ⇒ Formula.neg
25
26 @[grind] def top : Formula α := ~⊥
27 notation:max "⊤" ⇒ Formula.top
28
29 section
30
31 variable [Encodable α]
32 open Encodable
33
34 def toNat : Formula α → ℕ
35 | ⊥      ⇒ (Nat.pair 0 0) + 1
36 | #a    ⇒ (Nat.pair 1 <| encode a) + 1
37 | φ → ψ ⇒ (Nat.pair 2 <| φ.toNat.pair ψ.toNat) + 1
38 | φ ∧ ψ ⇒ (Nat.pair 3 <| φ.toNat.pair ψ.toNat) + 1
39 | φ ∨ ψ ⇒ (Nat.pair 4 <| φ.toNat.pair ψ.toNat) + 1
40
41 def ofNat : ℕ → Option (Formula α)
42 | 0 ⇒ none
43 | e + 1 ⇒
44   let idx := e.unpair.1
45   let c := e.unpair.2
46   match idx with
47   | 0 ⇒ some ⊥
48   | 1 ⇒ (decode c).map (#.)
49   | 2 ⇒
50     have : c.unpair.1 < e + 1 := Nat.lt_succ_iff.mpr
51       $ le_trans (Nat.unpair_left_le _)
52       $ Nat.unpair_right_le _
53     have : c.unpair.2 < e + 1 := Nat.lt_succ_iff.mpr
54       $ le_trans (Nat.unpair_right_le _)
55       $ Nat.unpair_right_le _
56     do
57       let φ <- ofNat c.unpair.1
58       let ψ <- ofNat c.unpair.2
59       return φ → ψ

```

```

60 | 3 =>
61 | have : c.unpair.1 < e + 1 := Nat.lt_succ_iff.mpr
62 | $ le_trans (Nat.unpair_left_le _)
63 | $ Nat.unpair_right_le _
64 | have : c.unpair.2 < e + 1 := Nat.lt_succ_iff.mpr
65 | $ le_trans (Nat.unpair_right_le _)
66 | $ Nat.unpair_right_le _
67 | do
68 | let φ <- ofNat c.unpair.1
69 | let ψ <- ofNat c.unpair.2
70 | return φ ∧ ψ
71 | 4 =>
72 | have : c.unpair.1 < e + 1 := Nat.lt_succ_iff.mpr
73 | $ le_trans (Nat.unpair_left_le _)
74 | $ Nat.unpair_right_le _
75 | have : c.unpair.2 < e + 1 := Nat.lt_succ_iff.mpr
76 | $ le_trans (Nat.unpair_right_le _)
77 | $ Nat.unpair_right_le _
78 | do
79 | let φ <- ofNat c.unpair.1
80 | let ψ <- ofNat c.unpair.2
81 | return φ ∨ ψ
82 | _ => none
83
84 lemma ofNat_toNat : ∀ (φ : Formula α), ofNat (toNat φ) = some φ
85 | #a => by simp [toNat, ofNat, Nat.unpair_pair, encodek];
86 | ⊥ => by simp [toNat, ofNat]
87 | φ → ψ | φ ∧ ψ | φ ∨ ψ => by simp [toNat, ofNat, ofNat_toNat φ, ofNat_toNat ψ]
88
89 instance : Encodable (Formula α) where
90 encode := toNat
91 decode := ofNat
92 encodek := ofNat_toNat
93
94 end
95
96 end Formula
97
98
99 abbrev Theory (α) := Set (Formula α)

```

```

1 import Logic101.CPL.Formula
2
3 variable {α} {φ ψ χ : Formula α} {Γ Δ : Theory α}
4
5 inductive Proof : Formula α → Type*
6 | andIntro! {φ ψ} : Proof $ φ → ψ → (φ ∧ ψ)
7 | andElimL! {φ ψ} : Proof $ (φ ∧ ψ) → φ
8 | andElimR! {φ ψ} : Proof $ (φ ∧ ψ) → ψ
9 | orIntroL! {φ ψ} : Proof $ φ → (φ ∨ ψ)
10 | orIntroR! {φ ψ} : Proof $ ψ → (φ ∨ ψ)
11 | orElim! {φ ψ χ} : Proof $ (φ ∨ ψ) → (φ → χ) → (ψ → χ) → χ
12 | impS! {φ ψ χ} : Proof $ (φ → (ψ → χ)) → (φ → ψ) → (φ → χ)
13 | impK! {φ ψ} : Proof $ φ → ψ → φ
14 | mdp! {φ ψ} : Proof (φ → ψ) → Proof φ → Proof ψ
15 | efq! {φ} : Proof $ ⊥ → φ
16 | lem! {φ} : Proof $ φ ∨ ~φ
17 prefix:40 "⊢!" " => Proof
18
19 def Provable (φ : Formula α) := Nonempty (⊢! φ)
20 prefix:40 "⊢" " => Provable
21
22 abbrev Unprovable (φ : Formula α) := ~(⊢ φ)
23 prefix:40 "⊭" " => Unprovable
24
25 namespace Provable
26
27 open Proof
28

```

```

29 @[grind .] lemma andElimL :  $\vdash (\varphi \wedge \psi) \rightarrow \varphi := \langle \text{andElimL!} \rangle$ 
30 @[grind .] lemma andElimR :  $\vdash (\varphi \wedge \psi) \rightarrow \psi := \langle \text{andElimR!} \rangle$ 
31 @[grind .] lemma andIntro :  $\vdash \varphi \rightarrow \psi \rightarrow (\varphi \wedge \psi) := \langle \text{andIntro!} \rangle$ 
32 @[grind .] lemma orIntroL :  $\vdash \varphi \rightarrow (\varphi \vee \psi) := \langle \text{orIntroL!} \rangle$ 
33 @[grind .] lemma orIntroR :  $\vdash \psi \rightarrow (\varphi \vee \psi) := \langle \text{orIntroR!} \rangle$ 
34 @[grind .] lemma orElim :  $\vdash (\varphi \vee \psi) \rightarrow (\varphi \rightarrow \chi) \rightarrow (\psi \rightarrow \chi) \rightarrow \chi := \langle \text{orElim!} \rangle$ 
35 @[grind .] lemma impS :  $\vdash (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi) := \langle \text{impS!} \rangle$ 
36 @[grind .] lemma impK :  $\vdash \varphi \rightarrow \psi \rightarrow \varphi := \langle \text{impK!} \rangle$ 
37 @[grind .] lemma efq :  $\vdash \perp \rightarrow \varphi := \langle \text{efq!} \rangle$ 
38 @[grind .] lemma lem :  $\vdash \varphi \vee \sim\varphi := \langle \text{lem!} \rangle$ 
39 @[grind!  $\rightarrow$ ] lemma mdp :  $\vdash \varphi \rightarrow \psi \rightarrow \vdash \varphi \rightarrow \vdash \psi := \lambda \langle d_1 \rangle \langle d_2 \rangle \Rightarrow \langle \text{mdp! } d_1 \ d_2 \rangle$ 
40
41 @[induction_eliminator]
42 lemma rec
43   {motive   :  $(\varphi : \text{Formula } \alpha) \rightarrow (\vdash \varphi) \rightarrow \text{Sort}$ }
44   (andElimL :  $\forall \{\varphi \psi\}, \text{motive } ((\varphi \wedge \psi) \rightarrow \varphi) \text{ andElimL}$ )
45   (andElimR :  $\forall \{\varphi \psi\}, \text{motive } ((\varphi \wedge \psi) \rightarrow \psi) \text{ andElimR}$ )
46   (andIntro :  $\forall \{\varphi \psi\}, \text{motive } (\varphi \rightarrow \psi \rightarrow (\varphi \wedge \psi)) \text{ andIntro}$ )
47   (orIntroL :  $\forall \{\varphi \psi\}, \text{motive } (\varphi \rightarrow (\varphi \vee \psi)) \text{ orIntroL}$ )
48   (orIntroR :  $\forall \{\varphi \psi\}, \text{motive } (\psi \rightarrow (\varphi \vee \psi)) \text{ orIntroR}$ )
49   (orElim   :  $\forall \{\varphi \psi \chi\}, \text{motive } ((\varphi \vee \psi) \rightarrow (\varphi \rightarrow \chi) \rightarrow (\psi \rightarrow \chi) \rightarrow \chi) \text{ orElim}$ )
50   (impS     :  $\forall \{\varphi \psi \chi\}, \text{motive } ((\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) \text{ impS}$ )
51   (impK     :  $\forall \{\varphi \psi\}, \text{motive } (\varphi \rightarrow \psi \rightarrow \varphi) \text{ impK}$ )
52   (efq      :  $\forall \{\varphi\}, \text{motive } (\perp \rightarrow \varphi) \text{ efq}$ )
53   (lem      :  $\forall \{\varphi\}, \text{motive } (\varphi \vee \sim\varphi) \text{ lem}$ )
54   (mdp      :  $\forall \{\varphi \psi\}, (\text{h}\varphi\psi : \vdash \varphi \rightarrow \psi) \rightarrow (\text{h}\varphi : \vdash \varphi) \rightarrow \text{motive } (\varphi \rightarrow \psi) \text{ h}\varphi\psi \rightarrow \text{motive } \varphi \text{ h}\varphi$ )
   $\rightarrow \text{motive } \psi \text{ (mdp h}\varphi\psi \text{ h}\varphi)$ 
  :  $\forall \{\varphi\}, (\text{d} : \vdash \varphi) \rightarrow \text{motive } \varphi \text{ d} := \text{by}$ 
  rintro  $\varphi \langle \text{d} \rangle$ ; induction  $\text{d} < >$  grind;
55
56
57
58 @[grind  $\leq$ ]
59 lemma afortiori :  $\vdash \varphi \rightarrow \vdash \psi \rightarrow \varphi := \lambda h \Rightarrow \text{mdp impK } h$ 
60
61 @[grind .]
62 lemma impId :  $\vdash \varphi \rightarrow \varphi := \text{by}$ 
63   have  $h_1 : \vdash \varphi \rightarrow (\varphi \rightarrow \varphi) \rightarrow \varphi := \text{impK}$ ;
64   have  $h_2 : \vdash (\varphi \rightarrow (\varphi \rightarrow \varphi) \rightarrow \varphi) \rightarrow (\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi := \text{impS}$ ;
65   have  $h_3 : \vdash (\varphi \rightarrow \varphi \rightarrow \varphi) \rightarrow \varphi \rightarrow \varphi := \text{mdp } h_2 \ h_1$ ;
66   have  $h_4 : \vdash \varphi \rightarrow \varphi \rightarrow \varphi := \text{impK}$ ;
67   exact mdp  $h_3 \ h_4$ ;
68
69 end Provable
70
71
72
73
74 inductive Deduction : Theory  $\alpha \rightarrow$  Formula  $\alpha \rightarrow$  Type*
75 | of_context! { $\Gamma \varphi$ } :  $\varphi \in \Gamma \rightarrow$  Deduction  $\Gamma \varphi$ 
76 | of_proof!   { $\Gamma \varphi$ } :  $\vdash! \varphi \rightarrow$  Deduction  $\Gamma \varphi$ 
77 | mdp!       { $\Gamma \varphi \psi$ } : Deduction  $\Gamma (\varphi \rightarrow \psi) \rightarrow$  Deduction  $\Gamma \varphi \rightarrow$  Deduction  $\Gamma \psi$ 
78 infix:40 "  $\vdash!$  "  $\Rightarrow$  Deduction
79
80 def Deducible ( $\Gamma : \text{Theory } \alpha$ ) ( $\varphi : \text{Formula } \alpha$ ) := Nonempty ( $\Gamma \vdash! \varphi$ )
81 infix:40 "  $\vdash$  "  $\Rightarrow$  Deducible
82
83 abbrev Undeducible ( $\Gamma : \text{Theory } \alpha$ ) ( $\varphi : \text{Formula } \alpha$ ) :=  $\neg(\Gamma \vdash \varphi)$ 
84 infix:40 "  $\not\vdash$  "  $\Rightarrow$  Undeducible
85
86
87 namespace Deducible
88
89 @[grind  $\leftarrow$ ] lemma of_context (h :  $\varphi \in \Gamma$ ) :  $\Gamma \vdash \varphi := \langle \text{Deduction.of\_context! } h \rangle$ 
90
91 lemma of_proof (h :  $\vdash! \varphi$ ) :  $\Gamma \vdash \varphi := \langle \text{Deduction.of\_proof! } h \rangle$ 
92
93 @[grind  $\leftarrow$ ] lemma of_provable (h :  $\vdash \varphi$ ) :  $\Gamma \vdash \varphi := \text{of\_proof } h.\text{some}$ 
94
95 @[grind!  $\rightarrow$ ] lemma mdp :  $\Gamma \vdash \varphi \rightarrow \psi \rightarrow \Gamma \vdash \varphi \rightarrow \Gamma \vdash \psi := \lambda \langle d_1 \rangle \langle d_2 \rangle \Rightarrow \langle \text{Deduction.mdp! } d_1 \ d_2 \rangle$ 

```

```

96
97
98 @[grind .] lemma andElimL :  $\Gamma \vdash (\varphi \wedge \psi) \rightarrow \varphi := \text{by grind};$ 
99 @[grind .] lemma andElimR :  $\Gamma \vdash (\varphi \wedge \psi) \rightarrow \psi := \text{by grind};$ 
100 @[grind .] lemma andIntro :  $\Gamma \vdash \varphi \rightarrow \psi \rightarrow (\varphi \wedge \psi) := \text{by grind};$ 
101 @[grind .] lemma orIntroL :  $\Gamma \vdash \varphi \rightarrow (\varphi \vee \psi) := \text{by grind};$ 
102 @[grind .] lemma orIntroR :  $\Gamma \vdash \psi \rightarrow (\varphi \vee \psi) := \text{by grind};$ 
103 @[grind .] lemma orElim :  $\Gamma \vdash (\varphi \vee \psi) \rightarrow (\varphi \rightarrow \chi) \rightarrow (\psi \rightarrow \chi) \rightarrow \chi := \text{by grind};$ 
104 @[grind .] lemma impS :  $\Gamma \vdash (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi) := \text{by grind};$ 
105 @[grind .] lemma impK :  $\Gamma \vdash \varphi \rightarrow \psi \rightarrow \varphi := \text{by grind};$ 
106 @[grind .] lemma efq :  $\Gamma \vdash \perp \rightarrow \varphi := \text{by grind};$ 
107 @[grind .] lemma lem :  $\Gamma \vdash \varphi \vee \sim\varphi := \text{by grind};$ 
108
109 @[grind  $\rightarrow$ ]
110 lemma ruleAndElimL :  $\Gamma \vdash \varphi \wedge \psi \rightarrow \Gamma \vdash \varphi := \text{by}$ 
111   intro h;
112   apply mdp _ h;
113   exact andElimL;
114
115 @[grind  $\rightarrow$ ]
116 lemma ruleAndElimR :  $\Gamma \vdash \varphi \wedge \psi \rightarrow \Gamma \vdash \psi := \text{by}$ 
117   intro h;
118   apply mdp _ h;
119   exact andElimR;
120
121 @[grind  $\rightarrow$ ]
122 lemma ruleAndIntro :  $\Gamma \vdash \varphi \rightarrow \Gamma \vdash \psi \rightarrow \Gamma \vdash \varphi \wedge \psi := \text{by}$ 
123   intro h $\varphi$  h $\psi$ ;
124   apply mdp _ h $\psi$ ;
125   apply mdp _ h $\varphi$ ;
126   exact andIntro;
127
128 lemma ruleOrIntroL :  $\Gamma \vdash \varphi \rightarrow \Gamma \vdash \varphi \vee \psi := \text{by}$ 
129   intro h;
130   apply mdp _ h;
131   exact orIntroL;
132
133 lemma ruleOrIntroR :  $\Gamma \vdash \psi \rightarrow \Gamma \vdash \varphi \vee \psi := \text{by}$ 
134   intro h;
135   apply mdp _ h;
136   exact orIntroR;
137
138 @[grind  $\rightarrow$ ]
139 lemma ruleOrElim :  $\Gamma \vdash \varphi \vee \psi \rightarrow \Gamma \vdash \varphi \rightarrow \chi \rightarrow \Gamma \vdash \psi \rightarrow \chi \rightarrow \Gamma \vdash \chi := \text{by}$ 
140   intro h $_1$  h $_2$  h $_3$ ;
141   apply mdp _ h $_3$ ;
142   apply mdp _ h $_2$ ;
143   apply mdp _ h $_1$ ;
144   exact orElim;
145
146 @[grind  $\leq$ ] lemma afortiori :  $\Gamma \vdash \varphi \rightarrow \Gamma \vdash \psi \rightarrow \varphi := \lambda h \Rightarrow \text{mdp impK } h$ 
147
148 @[grind  $\leq$ ] lemma ruleImpS :  $\Gamma \vdash \varphi \rightarrow (\psi \rightarrow \chi) \rightarrow \Gamma \vdash \varphi \rightarrow \psi \rightarrow \Gamma \vdash \varphi \rightarrow \chi := \lambda h_1 h_2 \Rightarrow \text{mdp}$ 
149   (mdp impS h $_1$ ) h $_2$ 
150
151 @[grind .] lemma impId :  $\Gamma \vdash \varphi \rightarrow \varphi := \text{of\_provable Provable.impId}$ 
152
153 @[induction_eliminator]
154 lemma rec
155   {motive   : ( $\Gamma$  : Theory  $\alpha$ )  $\rightarrow$  ( $\varphi$  : Formula  $\alpha$ )  $\rightarrow$  ( $\Gamma \vdash \varphi$ )  $\rightarrow$  Sort}
156   {of_context :  $\forall \{\Gamma \varphi\} (h : \varphi \in \Gamma), \text{motive } \Gamma \varphi (\text{of\_context } h)$ }
157   {of_provable :  $\forall \{\Gamma \varphi\} (h : \vdash \varphi), \text{motive } \Gamma \varphi (\text{of\_provable } h)$ }
158   (mdp      :  $\forall \{\Gamma \varphi \psi\}, (h\varphi\psi : \Gamma \vdash \varphi \rightarrow \psi) \rightarrow (h\varphi : \Gamma \vdash \varphi) \rightarrow \text{motive } \Gamma (\varphi \rightarrow \psi) h\varphi\psi \rightarrow$ 
159   motive  $\Gamma \varphi h\varphi \rightarrow \text{motive } \Gamma \psi (\text{mdp } h\varphi\psi h\varphi)$ )
160   :  $\forall \{\Gamma \varphi\}, (d : \Gamma \vdash \varphi) \rightarrow \text{motive } \Gamma \varphi d := \text{by}$ 
161   rintro  $\Gamma \varphi \langle d \rangle$ ;
162   induction d with
163   | of_context! h  $\Rightarrow$  apply of_context h;

```

```

163 | of_proof! h ⇒ apply of_provable ⟨h⟩;
164 | mdp! d₁ d₂ ih₁ ih₂ ⇒
165   apply mdp;
166   . exact ih₁;
167   . exact ih₂;
168
169 lemma rec_provable
170   {motive : (Γ : Theory α) → (φ : Formula α) → (Γ ⊢ φ) → Sort}
171   (of_context : ∀ {Γ φ} (h : φ ∈ Γ), motive Γ φ (of_context h))
172   (mdp       : ∀ {Γ φ ψ}, (hφψ : Γ ⊢ φ → ψ) → (hφ : Γ ⊢ φ) → motive Γ (φ → ψ) hφψ →
173   motive Γ φ hφ → motive Γ ψ (mdp hφψ hφ))
174   (andElimL  : ∀ {Γ φ ψ}, motive Γ ((φ ∧ ψ) → φ) andElimL)
175   (andElimR  : ∀ {Γ φ ψ}, motive Γ ((φ ∧ ψ) → ψ) andElimR)
176   (andIntro  : ∀ {Γ φ ψ}, motive Γ (φ → ψ → (φ ∧ ψ)) andIntro)
177   (orIntroL  : ∀ {Γ φ ψ}, motive Γ (φ → (φ ∨ ψ)) orIntroL)
178   (orIntroR  : ∀ {Γ φ ψ}, motive Γ (ψ → (φ ∨ ψ)) orIntroR)
179   (orElim    : ∀ {Γ φ ψ χ}, motive Γ ((φ ∨ ψ) → (φ → χ) → (ψ → χ) → χ) orElim)
180   (impS      : ∀ {Γ φ ψ χ}, motive Γ ((φ → (ψ → χ)) → (φ → ψ) → (φ → χ)) impS)
181   (impK      : ∀ {Γ φ ψ}, motive Γ (φ → ψ → φ) impK)
182   (efq       : ∀ {Γ φ}, motive Γ (⊥ → φ) efq)
183   (lem       : ∀ {Γ φ}, motive Γ (φ ∨ ¬φ) lem)
184   : ∀ {φ}, (d : Γ ⊢ φ) → motive Γ φ d := by
185   rintro φ d;
186   induction d using Deducible.rec with
187   | of_provable h ⇒ induction h <;> grind;
188   | _ ⇒ grind;
189
190 lemma weakening (hΓΔ : Γ ⊆ Δ) (h : Γ ⊢ φ) : Δ ⊢ φ := by
191   induction h using Deducible.rec with
192   | of_context h ⇒
193     apply of_context;
194     exact hΓΔ h;
195   | of_provable h ⇒
196     apply of_provable;
197     exact h;
198   | mdp d₁ d₂ ih₁ ih₂ ⇒
199     apply mdp;
200     . apply ih₁ hΓΔ;
201     . apply ih₂ hΓΔ;
202
203 @[grind <=]
204 lemma weakening_insert (h : Γ ⊢ ψ) : insert φ Γ ⊢ ψ := weakening (by grind) h
205
206 @[grind =]
207 theorem deduction_theorem : (insert φ Γ) ⊢ ψ ↔ Γ ⊢ (φ → ψ) := by
208   constructor;
209   . intro h;
210     generalize eΔ : insert φ Γ = Δ at h;
211     induction h using Deducible.rec with
212     | of_context h ⇒
213       subst eΔ;
214       simp at h;
215       rcases h with (rfl | h);
216       . exact impId;
217       . apply afortiori
218         apply of_context h;
219     | of_provable h ⇒
220       apply of_provable;
221       exact Provable.afortiori h;
222     | mdp _ _ ih₁ ih₂ ⇒
223       subst eΔ;
224       replace ih₁ := ih₁ (by grind);
225       replace ih₂ := ih₂ (by grind);
226       apply ruleImpS ih₁ ih₂;
227   . intro h;
228     apply mdp (weakening_insert h);
229     apply of_context;
230     simp;

```

```

231 lemma ruleImpTrans :  $\Gamma \vdash \varphi \rightarrow \psi \rightarrow \Gamma \vdash \psi \rightarrow \chi \rightarrow \Gamma \vdash \varphi \rightarrow \chi :=$  by
232   intro h1 h2;
233   replace h1 : insert  $\varphi$   $\Gamma \vdash \psi :=$  deduction_theorem.mpr h1;
234   replace h2 : insert  $\varphi$   $\Gamma \vdash \psi \rightarrow \chi :=$  weakening_insert h2;
235   exact deduction_theorem.mp $ mdp h2 h1;
236
237 lemma ruleEFQNeg :  $\Gamma \vdash \sim\varphi \rightarrow \Gamma \vdash \varphi \rightarrow \psi := \lambda h \Rightarrow$  ruleImpTrans h efq
238
239 lemma ruleDNE (h :  $\Gamma \vdash \sim\sim\varphi$ ) :  $\Gamma \vdash \varphi :=$  by
240   apply ruleOrElim lem;
241   . show  $\Gamma \vdash \varphi \rightarrow \varphi$ ;
242     exact impId;
243   . apply ruleEFQNeg h;
244
245 end Deducible

```

```

1 import Logic101.CPL.Formula
2 import Logic101.CPL.Hilbert
3
4 variable { $\alpha$  : Type u} { $\varphi \psi \chi$  : Formula  $\alpha$ } { $\Gamma \Delta$  : Theory  $\alpha$ }
5
6 abbrev Valuation ( $\alpha$ ) :=  $\alpha \rightarrow$  Prop
7
8 @[grind]
9 def Forces (V : Valuation  $\alpha$ ) : Formula  $\alpha \rightarrow$  Prop
10 | #a       $\Rightarrow$  V a
11 |  $\perp$       $\Rightarrow$  False
12 |  $\varphi \wedge \psi$   $\Rightarrow$  Forces V  $\varphi \wedge$  Forces V  $\psi$ 
13 |  $\varphi \vee \psi$   $\Rightarrow$  Forces V  $\varphi \vee$  Forces V  $\psi$ 
14 |  $\varphi \rightarrow \psi$   $\Rightarrow$  Forces V  $\varphi \rightarrow$  Forces V  $\psi$ 
15 infix:50 "  $\Vdash$  "  $\Rightarrow$  Forces
16
17 variable {V : Valuation  $\alpha$ }
18
19 @[grind .] lemma forces_top :  $V \Vdash \top :=$  by grind;
20 @[grind =] lemma forces_neg :  $V \Vdash \sim\varphi \leftrightarrow \neg(V \Vdash \varphi) :=$  by grind;
21
22
23 @[grind .]
24 def Valid ( $\varphi$  : Formula  $\alpha$ ) : Prop :=  $\forall V, V \Vdash \varphi$ 
25 prefix:50 "  $\Vdash$  "  $\Rightarrow$  Valid
26 alias Tautology := Valid
27
28
29 @[grind .]
30 def ConsequenceOn (V : Valuation  $\alpha$ ) ( $\Gamma$  : Theory  $\alpha$ ) ( $\varphi$  : Formula  $\alpha$ ) : Prop := ( $\forall \psi \in \Gamma,$ 
31  $V \Vdash \psi$ )  $\rightarrow V \Vdash \varphi$ 
32 notation  $\Gamma$  "  $\Vdash$ [V]" "  $\varphi \Rightarrow$  ConsequenceOn V  $\Gamma \varphi$ 
33
34 @[grind .]
35 def Consequence ( $\Gamma$  : Theory  $\alpha$ ) ( $\varphi$  : Formula  $\alpha$ ) : Prop :=  $\forall V, \Gamma \Vdash[V] \varphi$ 
36 infix:50 "  $\Vdash$  "  $\Rightarrow$  Consequence
37
38 @[grind =]
39 lemma iff_valid_forall_forces :  $\Vdash \varphi \leftrightarrow \emptyset \Vdash \varphi :=$  by simp [Valid, Consequence,
40 ConsequenceOn]
41
42 theorem sound :  $\Gamma \vdash \varphi \rightarrow \Gamma \Vdash \varphi :=$  by
43   rintro h;
44   induction h using Deducible.rec_provable with
45   | orElim  $\Rightarrow$ 
46     intro V hV;
47     grind [Forces];
48   | mdp _ _ ih $\varphi\psi$  ih $\varphi$   $\Rightarrow$ 
49     intro V hV;
50     exact ih $\varphi\psi$  V hV $ ih $\varphi$  V hV;
51   | _  $\Rightarrow$  grind;

```

```

51 lemma sound_provable :  $\vdash \varphi \rightarrow \vDash \varphi :=$  by
52   intro h;
53   apply iff_valid_forall_forces.mpr;
54   exact sound ( $\Gamma := \emptyset$ ) (Deducible.of_provable h);
55
56 theorem consistent :  $\not\vdash (\perp : \text{Formula } \alpha) :=$  by
57   by_contra h;
58   apply sound_provable h ( $\lambda \_ \Rightarrow \text{True}$ );
59
60
61
62
63 open Deducible
64
65 namespace Theory
66
67 variable { $\Gamma : \text{Theory } \alpha$ }
68
69 abbrev Inconsistent ( $\Gamma : \text{Theory } \alpha$ ) : Prop :=  $\Gamma \vdash \perp$ 
70
71 abbrev Consistent ( $\Gamma : \text{Theory } \alpha$ ) : Prop :=  $\neg(\Gamma.\text{Inconsistent})$ 
72
73 @[grind =] lemma iff_inconsistent_insert :  $(\text{insert } \varphi \Gamma).\text{Inconsistent} \leftrightarrow \Gamma \vdash \sim\varphi :=$ 
74   deduction_theorem
75
76 @[grind =] lemma iff_consistent_insert :  $(\text{insert } \varphi \Gamma).\text{Consistent} \leftrightarrow \Gamma \not\vdash \sim\varphi :=$ 
77   deduction_theorem.not
78
79 @[grind  $\Rightarrow$ ]
80 lemma either_consistent_insert_of_consistent ( $\Gamma_{\text{consis}} : \Gamma.\text{Consistent}$ ) ( $\varphi$ ) :  $(\text{insert } \varphi$ 
81  $\Gamma).\text{Consistent} \vee (\text{insert } (\sim\varphi) \Gamma).\text{Consistent} :=$  by
82   contrapose!  $\Gamma_{\text{consis}}$ ;
83   obtain  $\langle h_1, h_2 \rangle := \Gamma_{\text{consis}}$ ;
84   apply ruleOrElim lem;
85   . show  $\Gamma \vdash \varphi \rightarrow \perp$ ;
86     grind;
87   . grind
88
89 @[grind  $\Rightarrow$ ]
90 lemma either_unprovable_of_consistent ( $\Gamma_{\text{consis}} : \Gamma.\text{Consistent}$ ) ( $\varphi$ ) :  $\Gamma \not\vdash \varphi \vee \Gamma \not\vdash \sim\varphi :=$ 
91   by
92     contrapose!  $\Gamma_{\text{consis}}$ ;
93     obtain  $\langle h\varphi, hn\varphi \rangle := \Gamma_{\text{consis}}$ ;
94     apply mdp  $hn\varphi$   $h\varphi$ ;
95
96 @[grind  $\rightarrow$ ]
97 lemma unprovable_neg_of_provable_of_consistent ( $\Gamma_{\text{consis}} : \Gamma.\text{Consistent}$ ) ( $h : \Gamma \vdash \varphi$ ) :
98    $\Gamma \not\vdash \sim\varphi :=$  by grind;
99
100 @[grind  $\rightarrow$ ]
101 lemma unprovable_of_provable_neg_of_consistent ( $\Gamma_{\text{consis}} : \Gamma.\text{Consistent}$ ) ( $h : \Gamma \vdash \sim\varphi$ ) :
102    $\Gamma \not\vdash \varphi :=$  by grind;
103
104 @[grind  $\rightarrow$ ]
105 lemma not_mem_bot_of_consistent ( $\Gamma_{\text{consis}} : \Gamma.\text{Consistent}$ ) :  $\perp \notin \Gamma :=$  by
106   by_contra! h;
107   apply  $\Gamma_{\text{consis}}$ ;
108   exact of_context h;
109
110 @[grind  $\rightarrow$ ]
111 lemma consistent_insert_neg_of_unprovable ( $h : \Gamma \not\vdash \varphi$ ) :  $(\text{insert } (\sim\varphi) \Gamma).\text{Consistent} :=$ 
112   by
113     contrapose! h;
114     apply ruleDNE;
115     apply iff_inconsistent_insert.mp h;
116
117 abbrev Maximal ( $\Gamma : \text{Theory } \alpha$ ) : Prop :=  $\forall \varphi, \varphi \in \Gamma \vee \sim\varphi \in \Gamma$ 

```

```

111
112 end Theory
113
114
115 abbrev MaximalConsistentTheory ( $\alpha$ ) := {  $\Gamma$  : Theory  $\alpha$  //  $\Gamma$ .Consistent  $\wedge$   $\Gamma$ .Maximal }
116
117
118 namespace Theory
119
120 open Encodable
121
122 open Classical in
123 def lindenbaum_expand ( $\Gamma$  : Theory  $\alpha$ ) ( $\varphi$  : Formula  $\alpha$ ) : Theory  $\alpha$  := if Theory.Consistent
124 (insert  $\varphi$   $\Gamma$ ) then insert  $\varphi$   $\Gamma$  else  $\Gamma$ 
125
126 @[grind  $\Rightarrow$ ]
127 lemma consistent_lindenbaum_expand ( $\Gamma$ _consis : Theory.Consistent  $\Gamma$ ) : Theory.Consistent
128 (lindenbaum_expand  $\Gamma$   $\varphi$ ) := by
129   dsimp [lindenbaum_expand];
130   split <;> assumption;
131
132 variable [Encodable  $\alpha$ ]
133
134 def lindenbaum_indexed ( $\Gamma$  : Theory  $\alpha$ ) :  $\mathbb{N}$   $\rightarrow$  Theory  $\alpha$ 
135 | 0  $\Rightarrow$   $\Gamma$ 
136 | n + 1  $\Rightarrow$ 
137   match (decode n) with
138   | some  $\varphi$   $\Rightarrow$  lindenbaum_expand (lindenbaum_indexed  $\Gamma$  n)  $\varphi$ 
139   | none  $\Rightarrow$  lindenbaum_indexed  $\Gamma$  n
140
141 variable { $\Gamma$  : Theory  $\alpha$ } {n :  $\mathbb{N}$ }
142
143 @[grind  $\Rightarrow$ ]
144 lemma consistent_lindenbaum_indexed ( $\Gamma$ _consis : Theory.Consistent  $\Gamma$ ) :
145 Theory.Consistent (lindenbaum_indexed  $\Gamma$  n) := by
146   induction n with
147   | zero  $\Rightarrow$  assumption;
148   | succ n ih  $\Rightarrow$ 
149     dsimp [lindenbaum_indexed];
150     split;
151     . grind [consistent_lindenbaum_expand];
152     . assumption;
153
154 lemma monotonic_lindenbaum_indexed (h : n  $\leq$  m) : lindenbaum_indexed  $\Gamma$  n  $\subseteq$ 
155 lindenbaum_indexed  $\Gamma$  m := by
156   induction h with
157   | refl  $\Rightarrow$  simp;
158   | step h ih  $\Rightarrow$ 
159     trans;
160     . exact ih;
161     . dsimp [lindenbaum_indexed, lindenbaum_expand];
162     split;
163     . split <;> grind;
164     . grind;
165
166 def lindenbaum_maximal ( $\Gamma$  : Theory  $\alpha$ ) : Theory  $\alpha$  := U n, lindenbaum_indexed  $\Gamma$  n
167
168 lemma iff_mem_lindenbaum_maximal :  $\varphi \in \Gamma$ .lindenbaum_maximal  $\leftrightarrow$   $\exists$  n,  $\varphi \in$ 
169  $\Gamma$ .lindenbaum_indexed n := by
170   simp [lindenbaum_maximal, Set.mem_iUnion]
171
172 @[grind .]
173 lemma subset_lindenbaum_indexed_lindenbaum_maximal : lindenbaum_indexed  $\Gamma$  n  $\subseteq$ 
174 lindenbaum_maximal  $\Gamma$  := by
175   intro  $\varphi$  h $\varphi$ ;
176   apply iff_mem_lindenbaum_maximal.mpr;
177   use n;

```

```

172 lemma subset_lindenbaum_maximal :  $\Gamma \subseteq$  lindenbaum_maximal  $\Gamma :=$ 
173 subset_lindenbaum_indexed_lindenbaum_maximal (n := 0)
174
175 lemma exists_deducible_maximal (h :  $\Gamma$ .lindenbaum_maximal  $\vdash$   $\varphi$ ) :  $\exists$  n, lindenbaum_indexed
176  $\Gamma$  n  $\vdash$   $\varphi :=$  by
177   generalize e $\Delta$  :  $\Gamma$ .lindenbaum_maximal =  $\Delta$  at h;
178   induction h using Deducible.rec with
179     | of_context h  $\Rightarrow$ 
180       subst e $\Delta$ ;
181       obtain  $\langle$ n, hn $\rangle :=$  iff_mem_lindenbaum_maximal.mp h;
182       use n;
183       apply of_context hn;
184     | of_provable h  $\Rightarrow$ 
185       use 0;
186       exact of_provable h;
187     | mdp d1 d2 ih1 ih2  $\Rightarrow$ 
188       obtain  $\langle$ n, ih1 $\rangle :=$  ih1 e $\Delta$ ;
189       obtain  $\langle$ m, ih2 $\rangle :=$  ih2 e $\Delta$ ;
190       use max n m;
191       apply mdp;
192       . exact weakening (monotonic_lindenbaum_indexed (by omega)) ih1;
193       . exact weakening (monotonic_lindenbaum_indexed (by omega)) ih2;
194
195 @[grind .]
196 lemma consistent_lindenbaum_maximal ( $\Gamma$ _consis : Theory.Consistent  $\Gamma$ ) :
197 Theory.Consistent ( $\Gamma$ .lindenbaum_maximal) := by
198   by_contra hC;
199   obtain  $\langle$ _, hn $\rangle :=$  exists_deducible_maximal hC;
200   apply consistent_lindenbaum_indexed  $\Gamma$ _consis hn;
201
202 lemma maximal_lindenbaum_maximal ( $\Gamma$ _consis : Theory.Consistent  $\Gamma$ ) : Theory.Maximal
203 ( $\Gamma$ .lindenbaum_maximal) := by
204   intro  $\varphi$ ;
205   rcases  $\Gamma$ .lindenbaum_maximal.either_unprovable_of_consistent
206 (consistent_lindenbaum_maximal  $\Gamma$ _consis)  $\varphi$  with (hC | hC);
207   . right;
208   apply iff_mem_lindenbaum_maximal.mpr;
209   use ((encode ( $\sim\varphi$ )) + 1);
210   suffices (insert ( $\sim\varphi$ ) ( $\Gamma$ .lindenbaum_indexed (encode ( $\sim\varphi$ )))) .Consistent by
211     simp [lindenbaum_indexed, lindenbaum_expand, this];
212   apply iff_consistent_insert.mpr;
213   contrapose! hC;
214   apply weakening (by grind) (ruleDNE hC);
215   . left;
216   apply iff_mem_lindenbaum_maximal.mpr;
217   use ((encode  $\varphi$ ) + 1);
218   suffices (insert  $\varphi$  ( $\Gamma$ .lindenbaum_indexed (encode  $\varphi$ ))) .Consistent by
219     simp [lindenbaum_indexed, lindenbaum_expand, this];
220   apply iff_consistent_insert.mpr;
221   contrapose! hC;
222   apply weakening (by grind) hC;
223
224 lemma lindenbaum ( $\Gamma$  : Theory  $\alpha$ ) ( $\Gamma$ _consis : Theory.Consistent  $\Gamma$ ) :  $\exists$   $\Delta$  :
225 MaximalConsistentTheory  $\alpha$ ,  $\Gamma \subseteq \Delta$ .1 := by
226   use  $\langle$  $\Gamma$ .lindenbaum_maximal, consistent_lindenbaum_maximal  $\Gamma$ _consis,
227 maximal_lindenbaum_maximal  $\Gamma$ _consis $\rangle$ ;
228   exact subset_lindenbaum_maximal;
229
230 end Theory
231
232 namespace MaximalConsistentTheory
233
234 variable { $\Gamma$  : MaximalConsistentTheory  $\alpha$ }
235
236 lemma consistent ( $\Gamma$  : MaximalConsistentTheory  $\alpha$ ) :  $\Gamma$ .1.Consistent :=  $\Gamma$ .2.1

```

```

232
233 @[grind .] lemma maximal : Theory.Maximal  $\Gamma.1 := \Gamma.2.2$ 
234
235 @[grind =]
236 lemma iff_mem_provable :  $\varphi \in \Gamma.1 \leftrightarrow \Gamma.1 \vdash \varphi :=$  by
237   constructor;
238   . apply of_context;
239   . intro h;
240   suffices  $\sim\varphi \notin \Gamma.1$  by grind;
241   by_contra hC;
242   apply  $\Gamma$ .consistent;
243   apply mdp;
244   . show  $\Gamma.1 \vdash \sim\varphi$ ;
245     apply of_context hC;
246     . exact h;
247
248 @[simp, grind .] lemma not_mem_bot :  $\perp \notin \Gamma.1 :=$  by grind;
249
250 @[grind =] lemma iff_mem_neg :  $\sim\varphi \in \Gamma.1 \leftrightarrow \varphi \notin \Gamma.1 :=$  by constructor <=> grind;
251
252 @[grind =] lemma iff_not_mem_provable :  $\varphi \notin \Gamma.1 \leftrightarrow \Gamma.1 \vdash \sim\varphi :=$  by grind;
253
254 @[grind =]
255 lemma iff_mem_imp :  $(\varphi \rightarrow \psi) \in \Gamma.1 \leftrightarrow (\varphi \in \Gamma.1 \rightarrow \psi \in \Gamma.1) :=$  by
256   constructor;
257   . simp only [iff_mem_provable]
258     apply mdp;
259   . rintro h;
260     rcases not_or_of_imp h with (h $\varphi$  | h $\psi$ );
261     . replace h $\varphi :=$  iff_mem_neg.mpr h $\varphi$ ;
262       simp only [iff_mem_provable] at h $\varphi$   $\vdash$ ;
263       exact ruleEFQNeg h $\varphi$ ;
264     . simp only [iff_mem_provable] at h $\psi$   $\vdash$ ;
265       exact afortiori h $\psi$ ;
266
267 @[grind =]
268 lemma iff_mem_and :  $(\varphi \wedge \psi) \in \Gamma.1 \leftrightarrow \varphi \in \Gamma.1 \wedge \psi \in \Gamma.1 :=$  by
269   simp only [iff_mem_provable]
270   constructor;
271   . intro h;
272     constructor;
273     . apply ruleAndElimL h;
274     . apply ruleAndElimR h;
275   . simp only [and_imp];
276     apply ruleAndIntro;
277
278 @[grind =]
279 lemma iff_mem_or :  $(\varphi \vee \psi) \in \Gamma.1 \leftrightarrow \varphi \in \Gamma.1 \vee \psi \in \Gamma.1 :=$  by
280   constructor;
281   . intro h;
282     by_contra! hC;
283     obtain  $\langle h\varphi, h\psi \rangle :=$  hC;
284     replace h $\varphi : \uparrow\Gamma \vdash \sim\varphi :=$  iff_not_mem_provable.mp h $\varphi$ ;
285     replace h $\psi : \uparrow\Gamma \vdash \sim\psi :=$  iff_not_mem_provable.mp h $\psi$ ;
286     apply  $\Gamma$ .consistent;
287     apply ruleOrElim (iff_mem_provable.mp h) h $\varphi$  h $\psi$ ;
288   . simp only [iff_mem_provable];
289     rintro (h $\varphi$  | h $\psi$ );
290     . apply ruleOrIntroL;
291       exact h $\varphi$ ;
292     . apply ruleOrIntroR;
293       exact h $\psi$ ;
294
295 end MaximalConsistentTheory
296
297 theorem complete [Encodable  $\alpha$ ] :  $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi :=$  by
298   contrapose!;
299   intro h;
300   suffices  $\exists V, (\forall \psi \in \Gamma, V \vDash \psi) \wedge \neg V \vDash \varphi$  by simpa [Consequence, ConsequenceOn];
301

```

```

302   obtain ⟨Δ, hΓΔ⟩ := (insert (~φ) Γ).lindenbaum $ by grind
[Theory.consistent_insert_neg_of_unprovable];
303
304   let V : Valuation α := λ a ⇒ (#a ∈ Δ.1);
305   have hV : ∀ {ψ}, V ⊨ ψ ↔ ψ ∈ Δ.1 := by
306     intro ψ;
307     induction ψ <;> grind
308   use V;
309   constructor;
310   . intro γ hγ;
311     apply hV.mpr;
312     apply hΓΔ;
313     grind;
314   . apply hV.not.mpr;
315     apply Δ.iff_mem_neg.mp;
316     apply hΓΔ;
317     grind;

```