1 Boole-algebra

Egy *Boole-algebra* egy $\mathcal{B} = (B, \top, \bot, \wedge, \vee, \neg)$ struktúra, amelyben a B halmaz felett az alábbi műveletek vannak definiálva:

- $\top$: az igaz (*true*),
- $\bot$: a hamis (*false*),
- $\wedge : B \times B \to B$: konjunkció (*and*),
- $\vee : B \times B \to B$: diszjunkció (*or*),
- $\neg : B \to B$: negáció (*negation*).

A Boole-algebrára a következő axiómák érvényesek:

Asszociativitás

$$\forall x, y, z \in B, \quad (x \wedge (y \wedge z)) = ((x \wedge y) \wedge z), \tag{1}$$
$$\forall x, y, z \in B, \quad (x \vee (y \vee z)) = ((x \vee y) \vee z). \tag{2}$$

Kommutativitás

$$\forall x, y \in B, \quad (x \wedge y) = (y \wedge x), \tag{3}$$
$$\forall x, y \in B, \quad (x \vee y) = (y \vee x). \tag{4}$$

Disztributivitás

$$\forall x, y, z \in B, \quad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \tag{5}$$
$$\forall x, y, z \in B, \quad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z). \tag{6}$$

Identitás

$$\forall x \in B, \quad x \wedge \top = x, \tag{7}$$
$$\forall x \in B, \quad x \vee \bot = x. \tag{8}$$

Komplementer

$$\forall x \in B, \quad x \wedge \neg x = \bot, \tag{9}$$
$$\forall x \in B, \quad x \vee \neg x = \top. \tag{10}$$

1.1 Boole-algebra a Boole-típussal

Az alábbiakban definiálunk egy konkrét Boole-algebrát, ahol B a logikai értékek halmaza, egészen pontosan a Boole-típus, azaz $\{\text{true}, \text{false}\}$:

$$\text{bool} ::= \text{true} \mid \ \text{false}$$

Ezt felhasználva az alábbi szereposztásban bool, mint a Coq natív nyelvében lévő típus, a beépített Coq-függvényekkel Boole-algebra:

$$\begin{aligned}
B &:= \text{bool},\\
\top &:= \text{true},\\
\bot &:= \text{false},\\
\wedge &:= \text{andb}, \quad (\text{logikai ÉS}),\\
\vee &:= \text{orb}, \quad (\text{logikai VAGY}),\\
\neg &:= \text{negb} \quad (\text{logikai negáció}).
\end{aligned}$$

2 A halmazok implementációja a Martin-Löf-típuselméletben

A következőkben bemutatjuk a halmazok és halmazműveletek formalizálását Coq-ban. Ebben a formalizmusban a halmazokat függvényekként kezeljük, amelyek egy adott típus elemeihez igazságértékeket rendelnek.

2.1 Halmazok és elemek

Egy U-típusú halmazt (pontosabban, egy halmaz fogalmát) egy olyan függvényként definiálunk, amely egy U-beli elemet egy igaz-hamis állításhoz (vagyis egy logikai értékhez) rendel:

$$\text{SetU}(U) := U \to \text{Prop}.$$

Egy adott $x \in U$ elem A-ban való tagságát az alábbi módon definiáljuk:

$$\text{isin}(x, A) := A(x).$$

Azaz, $x \in A$ akkor igaz, ha az A-hoz tartozó függvény igazat ad vissza x-re.

2.2 Halmazműveletek

Az alábbiakban különféle halmazműveleteket definiálunk a Coq formális rendszerében.

Unió: Két halmaz unióját az alábbi függvény adja meg:

$$\text{union}(A, B) := \text{fun } x \Rightarrow A(x) \vee B(x).$$

Ez azt jelenti, hogy egy $x \in U$ elem az $A \cup B$-ban van, ha x eleme A-nak, vagy x eleme B-nek.

Metszet: Két halmaz metszete az alábbi függvényként értelmezhető:

$$\text{intersection}(A, B) := \text{fun } x \Rightarrow A(x) \wedge B(x).$$

Ez azt jelenti, hogy egy $x \in U$ elem az $A \cap B$-ban van, ha x egyszerre eleme A-nak és B-nek.

Komplementer: Egy halmaz komplementerét a következőképpen definiáljuk:

$$\text{complementer}(A) := \text{fun } x \Rightarrow \neg A(x).$$

Ez azt jelenti, hogy x eleme $\complement A$-nak, ha x nem eleme A-nak.

Üres halmaz: Az üres halmaz egy olyan függvény, amely minden elemhez hamis értéket rendel:

$$\text{empty} := \text{fun } x \Rightarrow \text{False}.$$

Teljes halmaz: A teljes halmaz egy olyan függvény, amely minden elemhez igaz értéket rendel:

$$\text{full} := \text{fun } x \Rightarrow \text{True}.$$

Részhalmaz: Az $A \subseteq B$ reláció azt jelenti, hogy A minden eleme B-nek is eleme:

$$\text{subset}(A, B) := \forall x, A(x) \to B(x).$$

Halmazok egyenlősége: Két halmaz akkor egyenlő, ha kölcsönösen részhalmazai egymásnak:

$$\text{seteq}(A, B) := (\forall x, A(x) \to B(x)) \wedge (\forall x, B(x) \to A(x)).$$

2.3 Halmazok egyenlőségének tulajdonságai

A halmazok egyenlősége reflexív, szimmetrikus és tranzitív. Ezek a tulajdonságok bizonyíthatók Coq-ban az alábbi módon:

Reflexivitás: Minden halmaz egyenlő önmagával:

$$A \equiv A.$$

Szimmetria: Ha $A \equiv B$, akkor $B \equiv A$:

$$A \equiv B \implies B \equiv A.$$

Tranzitivitás: Ha $A \equiv B$ és $B \equiv C$, akkor $A \equiv C$:

$$A \equiv B \wedge B \equiv C \implies A \equiv C.$$

2.4 Problémák az U típusú halmazokkal

Az U-típusú halmazok a Boole-algebra szintaxisa felett értelmes struktúrát képesek alkotni, de ehhez sokat kell dolgozni.

$$\text{setU\_Algebra}(U) := \left(\text{SetU}(U), \top, \bot, \cap, \cup, \complement\right),$$

pl. kommutatív a $\cup$ és $\cap$ esetén, de még ehhez is sokat kell dolgozni.

A halmazok egyenlősége nem teljesen ugyanaz, mint a logikai egyenlőség ($=$). A logikai egyenlőség azt jelenti, hogy két entitás azonos. Ezzel szemben a halmazok egyenlősége ($\equiv$) azt jelenti, hogy a két halmaznak ugyanazok az elemei.

Ez a különbség elmúlasztható az alábbi Coq axióma segítségével, amely összekapcsolja a halmazok egyenlőségét a logikai egyenlőséggel:

$$\texttt{Axiom setequality\_eq} : \forall (AB : \text{SetU } U), (A \equiv B) \to A = B.$$

Ez az axióma azt mondja ki, hogy ha két halmaz A és B elemeik szerint egyenlők ($A \equiv B$), akkor ezek logikailag is egyenlők ($A = B$). Azonban ezt az axiómát explicit módon be kell vezetni, mivel a halmazok egyenlősége ($\equiv$) nem automatikusan jelenti a logikai egyenlőséget ($=$) a Coq rendszerében.

2.4.1 Példa a különbségre

Tegyük fel, hogy két halmazt A és B különböző módon definiálunk, például:

$$A := \text{fun } x \Rightarrow x > 0,$$

$$B := \text{fun } x \Rightarrow x \geq 1.$$

Bár A és B különböző függvények, előfordulhat, hogy ugyanazokkal az elemekkel rendelkeznek, tehát $A \equiv B$, de nem $A = B$, mert a függvények más formában vannak definiálva. Ezt az ellentmondást a `setequality_eq` axióma hidalja át, amely garantálja, hogy $A \equiv B$ esetén $A = B$.

Számos más konstruktivitásból eredő probléma is lehet, de nyilván ezek újabb axiómákkal szintén eliminálhatók. Prop helyett lehet pl. bool-t is használni és akkor az egy másik implementáció, de akkor valóban levezethető lenne, hogy $\left(\text{SetU}(U), \top, \bot, \cap, \cup, \complement\right)$ Boole-alggebra (Boole-halmazalgebra). Ha ezt nem tesszük meg, akkor a negáció másként vislkedik és akkor csak Heyting-algebra.