Záróvizsga tételsor
12. Logika és számításelmélet

Ancsin Ádám

Logika és számításelmélet

Ítéletkalkulus és elsőrendű predikátumkalkulus: szintaxis, szemantika, ekvivalens átalakítások, a szemantikus következmény fogalma, rezolúció. – A kiszámíthatóság fogalma és a Church-Turing tézis. A Turing-gép. Rekurzív és rekurzívan felsorolható nyelvek. Eldönthetetlen problémák. Nevezetes idő- és tárbonyolultsági osztályok: P, NP, PSPACE. NP-teljes problémák.

1 Logika

1.1 Alapfogalmak

A logika tárgya az emberi gondolkodási folyamat vizsgálata és helyes gondolkodási formák keresése, illetve létrehozása.

Fogalmak:

1. **Állítás**: Olyan kijelentés, melynek logikai értéke (igaz volta) eldönthető, tetszőleges kontextusban igaz vagy hamis. Azt mondjuk, hogy egy állítás igaz, ha információtartalma megfelel a valóságnak (a tényeknek), és hamis az ellenkező esetben.

   A mindennapi beszédben használt kijelentő mondatok legtöbbször nem állítások, mivel a mondat tartalmába a kontextus is beleszámít: időpont, környezet állapota, általános műveltség bizonyos szintje, stb. (pl. nem állítás az, hogy "ma reggel 8-kor sütött a nap", de állítás pl. az, hogy "minden páros szám osztható 2-vel").

2. **Igazságérték**: Az igazságértékek halmaza $\mathbb{L} = \{igaz, hamis\}$.

3. **Gondolkodási forma**: Gondolkodási forma alatt egy olyan (F, A) párt értünk, ahol A állítás, $F = \{A_1, A_2, ..., A_n\}$ pedig állítások egy halmaza.

   A gondolkodásforma helyes, ha minden esetben, amikor F minden állítása igaz, akkor A is igaz.

1.2 Ítéletkalkulus

1.2.1 Az ítéletlogika szintaxisa

Az ítéletlogika ábécéje

Az ítéletlogika ábécéje $V_0 = V_v \cup \{(,)\} \cup \{\neg, \wedge, \vee, \supset\}$, ahol V_v az ítéletváltozók halmaza. Tehát V_0 az ítéletváltozókat, a zárójeleket, és a logikai műveletek jeleit tartalmazza.

Az ítéletlogika nyelve

Az ítéletlogika nyelve ($\mathcal{L}_0$) ítéletlogikai formulákból áll, amelyek a következőképpen állnak elő:

1. Minden ítéletváltozó ítéletlogikai formula. Ezek az úgynevezett prímformulák (vagy atomi formulák).

2. Ha A ítéletlogikai formula, akkor $\neg A$ is az.

3. Ha A és B ítéletlogikai formulák, akkor $(A \wedge B)$, $(A \vee B)$ és $(A \supset B)$ is ítéletlogikai formulák.

4. Minden ítéletlogikai formula az 1-3. szabályok véges sokszori alkalmazásával áll elő.

Literál: Ha X ítéletváltozó, akkor az X és $\neg X$ formulák literálok, amelyek alapja X.

Közvetlen részformula:

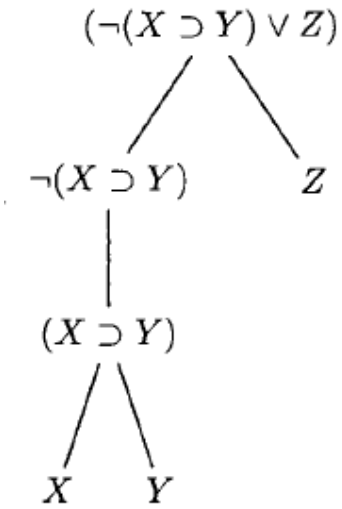
1. Prímformulának nincs közvetlen részformulája.
2. $\neg A$ közvetlen részformulája A.
3. $A \circ B$ ($\circ$ a $\wedge, \vee, \supset$ binér összekötőjelek egyike) közvetlen részformulái A (bal oldali) és B (jobb oldali).

Részformula: Legyen $A \in \mathcal{L}_0$ egy ítéletlogikai formula. Ekkor A részformuláinak halmaza a legszűkebb olyan halmaz, melynek

1. eleme az A, és
2. ha a C formula eleme, akkor C közvetlen részformulái is elemei.

Szerkezeti fa: Egy C formula szerkezeti fája egy olyan véges rendezett fa, melynek csúcsai formulák,

1. gyökere C,
2. a $\neg A$ csúcsának pontosan egy gyermeke van, az A,
3. a $A \circ B$ csúcsának pontosan két gyermeke van, rendre az A és B formulák,
4. levelei prímformulák.



ábra 1: Példa szerkezeti fára.

Logikai összetettség: Egy formula logikai összetettsége a benne található logikai összekötőjelek száma.

Művelet hatásköre: Egy művelet hatásköre a formula részformulái közül az a legkisebb logikai összetettségű részformula, melyben az adott művelet előfordul.

Fő logikai összekötőjel: Egy formula fő logikai összekötőjele az az összekötőjel, amelynek hatásköre maga a formula.

Precedencia: A logikai összekötőjelek precedenciája csökkenő sorrendben a következő: $\neg, \wedge, \vee, \supset$.

A definíciók alapján egyértelmű, hogy egy *teljesen zárójelezett formulában* mi a logikai összekötőjelek hatásköre és mi a fő logikai összekötőjel. Most megmutatjuk, hogy egy formulában milyen esetekben és mely részformulákat határoló zárójelek hagyhatóak el úgy, hogy a logikai összekötőjelek hatásköre ne változzon. A részformulák közül a prímformuláknak és a negációs formuláknak nincs külső zárójelpárja,

ezért csak az $(A \circ B)$ alakú részformulákról kell eldöntenünk, hogy írható-e helyettük $A \circ B$. A zárójelek elhagyását mindig a formula külső zárójelpárjának (ha van ilyen) elhagyásával kezdjük. Majd ha egy részformulában már megvizsgáltuk a külső zárójelelhagyás kérdését, utána ezen részformula közvetlen részformuláinak külső zárójeleivel foglalkozunk. Két eset lehetséges:

1. A részformula egy negációs formula, melyben az $(A \circ B)$ alakú közvetlen részformula külső zárójelei nem hagyhatók el.

2. A részformula egy $(A \bullet B)$ vagy $A \bullet B$ alakú formula, melynek A és B közvetlen részformuláiban kell dönteni a külső zárójelek sorsáról. Ha az A formula $A_1 \circ A_2$ alakú, akkor A külső zárójelpárja akkor hagyható el, ha $\circ$ nagyobb precedenciájú, mint $\bullet$. Ha a B formula $B_1 \circ B_2$ alakú, akkor B külső zárójelpárja akkor hagyható el, ha $\circ$ nagyobb vagy egyenlő precedenciájú, mint $\bullet$.

3. Ha egy $(A \wedge B)$ vagy $A \wedge B$ alakú formula valamely közvetlen részformulája szintén konjunkció, illetve egy $(A \vee B)$ vagy $A \vee B$ alakú formula valamely közvetlen részformulája szintén diszjunkció, akkor az ilyen részformulákból a külső zárójelpár elhagyható.

Formulaláncok: A zárójelek elhagyására vonatkozó megállapodásokat figyelembe véve úgynevezett konjunkciós, diszjunkciós, illetve implikációs formulaláncokat is nyerhetünk. Ezek alakja $A_1 \wedge ... \wedge A_n$, $A_1 \vee ... \vee A_n$, illetve $A_1 \supset ... \supset A_n$ Ezeknek a láncformuláknak a fő logikai összekötőjelét a következő zárójelezési megállapodással fogjuk meghatározni: $(A_1 \wedge (A_2 \wedge ... \wedge (A_{n-1} \wedge A_n)...))$, $(A_1 \vee (A_2 \vee ... \vee (A_{n-1} \vee A_n)...))$, illetve $(A_1 \supset (A_2 \supset ... \supset (A_{n-1} \supset A_n)...))$

1.2.2 Az ítéletlogika szemantikája

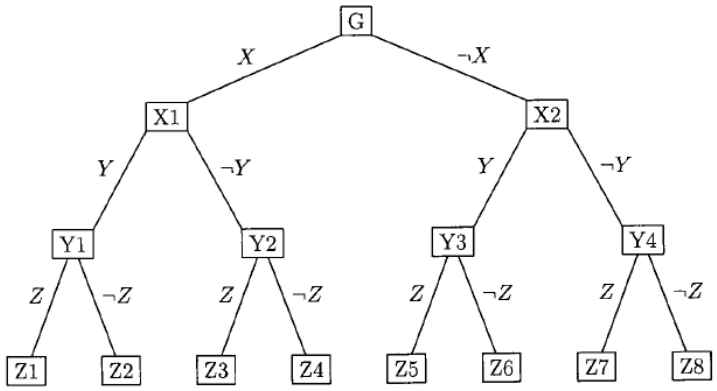
Interpretáció: $\mathcal{L}_0$ interpretációján egy $\mathcal{I} : V_v \to \mathbb{L}$ függvényt értünk, mely minden ítéletváltozóhoz egyértelműen hozzárendel egy igazságértéket.

Boole-értékelés: $\mathcal{L}_0$-beli formulák $\mathcal{I}$ *interpretációbeli Boole-értékelése* a következő $\mathcal{B}_\mathcal{I} : \mathcal{L}_0 \to \mathbb{L}$ függvény:

1. ha A prímformula, akkor $\mathcal{B}_\mathcal{I}(A) = \mathcal{I}(A)$,
2. $\mathcal{B}_\mathcal{I}(\neg A)$ legyen $\neg\mathcal{B}_\mathcal{I}(A)$,
3. $\mathcal{B}_\mathcal{I}(A \wedge B)$ legyen $\mathcal{B}_\mathcal{I}(A) \wedge \mathcal{B}_\mathcal{I}(B)$,
4. $\mathcal{B}_\mathcal{I}(A \vee B)$ legyen $\mathcal{B}_\mathcal{I}(A) \vee \mathcal{B}_\mathcal{I}(B)$,
5. $\mathcal{B}_\mathcal{I}(A \supset B)$ legyen $\mathcal{B}_\mathcal{I}(A) \supset \mathcal{B}_\mathcal{I}(B)$,

Bázis: A formula ítéletváltozóinak egy rögzített sorrendje.

Szemantikus fa: Egy formula különböző interpretációit szemantikus fa segítségével szemléltethetjük. A szemantikus fa egy olyan bináris fa, amelynek i. szintje ($i >= 1$) a bázis i. ítéletváltozójához tartozik, és minden csúcsából két él indul, az egyik a szinthez rendelt ítéletváltozóval, a másik annak negáltjával címkézve. Az X ítéletváltozó esetén az X címke jelentse azt, hogy az X *igaz* az adott interpretációban, a $\neg X$ címke pedig azt, hogy *hamis* az adott interpretációban. A szemantikus fa minden ága egy-egy lehetséges interpretációt reprezentál. Egy n változós formula esetén minden ág n hosszú, és a fának 2^n ága van és az összes lehetséges interpretációt tartalmazza.



ábra 2: Az X,Y,Z ítéletváltozókat tartalmazó formula szemantikus fája.

Igazságtábla: Egy n változós formula igazságtáblája egy $n+1$ oszlopból és 2^n sorból álló táblázat. A táblázat fejlécében az i. oszlophoz ($1 <= i <= n$) a formula bázisának i. ítéletváltozója, az $n+1$. oszlophoz maga a formula van hozzárendelve. Az első n oszlopban az egyes sorokhoz megadjuk rendre a formula különböző interpretációit, majd a formula oszlopába minden sorba beírjuk a formula - a sorhoz tartozó interpretációbeli Boole-értékeléssel kapott - igazságértékét.

A logikai műveletek igazságtáblája:

X	Y	$\neg X$	$X \wedge Y$	$X \vee Y$	$X \supset Y$
i	i	h	i	i	i
i	h	h	h	i	h
h	i	i	h	i	i
h	h	i	h	h	i

Igazhalmaz, hamishalmaz: Egy A formula igazhalmaza (A^i) azon interpretációk halmaza, melyen a formula igazságértékelése igaz. Az A formula hamishalmaza (A^h) pedig azon interpretációk halmaza, melyekre a formula igazságértékelése hamis.

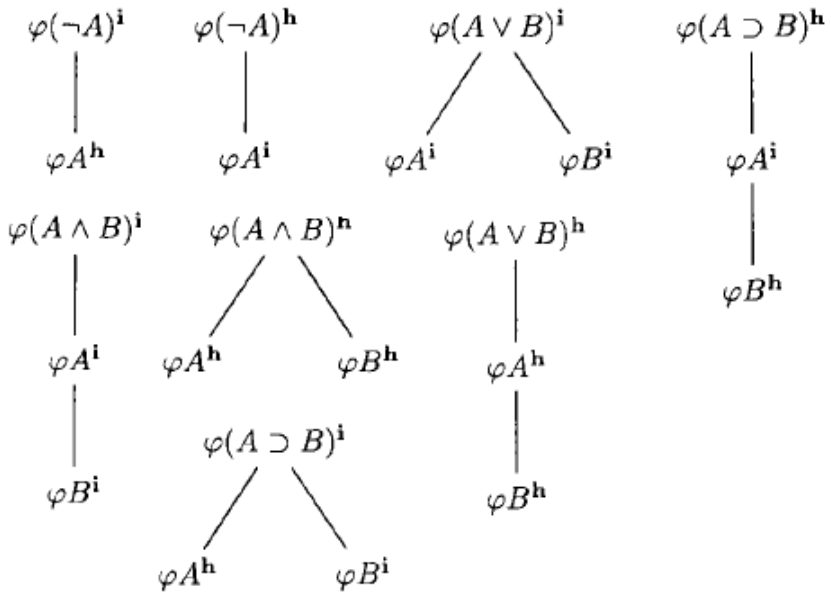
Igazságértékelés függvény: Olyan függvény, amely minden formulához hozzárendeli az igazhalmazát (φA^i) vagy a hamishalmazát (φA^h).

Legyen A egy tetszőleges ítéletlogikai formula. Határozzuk meg A-hoz az interpretációira vonatkozó φA^i, illetve φA^h feltételeket a következőképpen:

1. Ha A prímformula, a φA^i feltételt pontosan azok az $\mathcal{I}$ interpretációk elégítik ki, melyekre $\mathcal{I}(A) = igaz$, a φA^h feltételt pedig pontosan azok melyekre $\mathcal{I}(A) = hamis$.
2. A $\varphi(\neg A)^i$ feltételek pontosan akkor teljesülnek, ha teljesülnek a φA^h feltételek.
3. A $\varphi(A \wedge B)^i$ feltételek pontosan akkor teljesülnek, ha a φA^i és a φB^i feltételek egyszerre teljesülnek.
4. A $\varphi(A \vee B)^i$ feltételek pontosan akkor teljesülnek, ha a φA^i vagy a φB^i feltételek teljesülnek.
5. A $\varphi(A \supset B)^i$ feltételek pontosan akkor teljesülnek, ha a φA^h vagy a φB^i feltételek teljesülnek.

Tétel: Tetszőleges A ítéletlogikai formula esetén a φA^i feltételeket pontosan az A^i-beli interpretációk teljesítik.

Igazságértékelés-fa: Egy A formula φA^i, illetve φA^h feltételeket kielégítő interpretációit az igazságértékelés-fa segítségével szemléltethetjük. Az igazságértékelés-fát a formula szerkezeti fájának felhasználásával állítjuk elő. A gyökérhez hozzárendeljük, hogy A melyik igazságértékre való igazságértékelés-feltételeit keressük, majd a gyökér alá A közvetlen részformulái kerülnek a megfelelő feltétel-előírással, az alábbiak szerint:



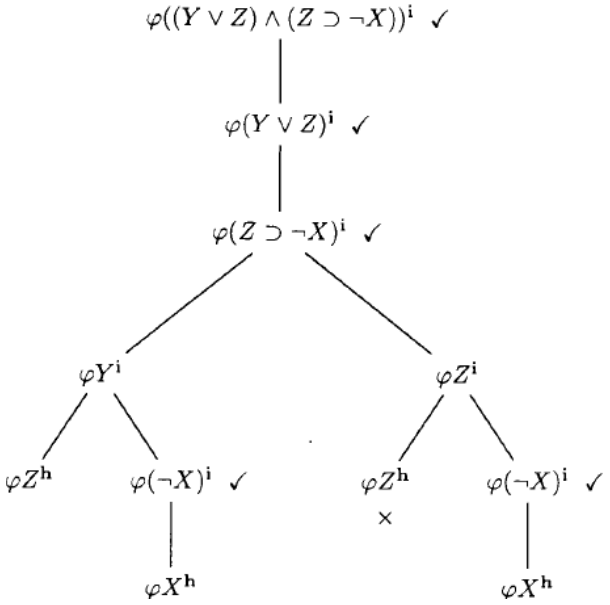
ábra 3: Igazságértékelés-fa feltétel-előírásai.

Ezután a gyökérhez a ✓(feldolgozott) jelet rendeljük. Az eljárást rekurzívan folytatjuk, amíg egy ágon a fel nem dolgozott formulák

- (a) mind ítéletváltozók nem lesznek, vagy
- (b) ugyanarra a formulára egymásnak ellentmondó előírás nem jelenik meg.

Az (a) esetben az ágon előforduló ítéletváltozóknak az ágon rögzített igazságértékeit tartalmazó n-esek mind elemei φA^i gyökér esetén a formula igazhalmazának, φA^h gyökér esetén a formula hamishalmazának.

A (b) esetben nem áll elő ilyen igazságérték n-es.



ábra 4: Az $(Y \vee Z) \wedge (Z \supset \neg X)$ formula igazságértékelés-fája.

A fenti példában a formula igazhalmaza az igazságértékelés-fa alapján: $\{(i,i,h),(h,i,i),(h,i,h),(h,h,i)\}$

Kiterjesztett igazságtábla: Egy igazságtáblában a formula igazságértéke kiszámításának megkönnyítésére vezették be a kiterjesztett igazságtáblát. A kiterjesztett igazságtáblában az ítéletváltozókhoz és a formulához rendelt oszlopokon kívül rendre a formula részformuláihoz tartozó oszlopok is megjelennek. Tulajdonképpen a szerkezeti fában megjelenő részformulák vannak felsorolva.

X	Y	Z	$Y \vee Z$	$\neg X$	$Z \supset \neg X$	$(Y \vee Z) \wedge (Z \supset \neg X)$
i	i	i	i	h	h	h
i	i	h	i	h	i	i
i	h	i	i	h	h	h
i	h	h	h	h	i	h
h	i	i	i	i	i	i
h	i	h	i	i	i	i
h	h	i	i	i	i	i
h	h	h	h	i	i	h

ábra 5: Az $(Y \vee Z) \wedge (Z \supset \neg X)$ formula kiterjesztett igazságtáblája.

Formula kielégíthetősége, modellje: Egy A ítéletlogikai formula *kielégíthető*, ha létezik olyan $\mathcal{I}$ interpretáció, melyre $\mathcal{I} \models_0 A$, azaz a $\mathcal{B}_\mathcal{I}$ Boole-értékelés A-hoz igaz értéket rendel. Egy ilyen interpretációt A *modelljének* nevezünk. Ha A-nak nincs modellje, akkor azt mondjuk, hogy *kielégíthetetlen.*

Ha A igazságtáblájában van olyan sor, amelyben a formula oszlopában igaz érték szerepel, akkor a formula kielégíthető, különben kielégíthetetlen. Ugyanígy, ha φA^i nem üres, akkor kielégíthető, különben kielégíthetetlen.

Ítéletlogikai törvény, tautológia: Egy A ítéletlogikai formula *ítéletlogikai törvény* vagy másképpen *tautológia*, ha $\mathcal{L}_0$ minden interpretációja modellje A-nak. (jelölés: $\models_0 A$)

Eldöntésprobléma: Eldöntésproblémának nevezzük a következő feladatokat:

1. Döntsük el tetszőleges formuláról, hogy tautológia-e!
2. Döntsük el tetszőleges formuláról, hogy kielégíthetetlen-e!

Tautologikusan ekvivalens formulák: Az A és B ítéletlogikai formulák *tautologikusan ekvivalensek* (jelölés: $A \sim_0 B$), ha $\mathcal{L}_0$ minden $\mathcal{I}$ interpretációjában $\mathcal{B}_\mathcal{I}(A) = \mathcal{B}_\mathcal{I}(B)$.

Formulahalmaz kielégíthetősége, modellje: $\mathcal{L}_0$ formuláinak egy tetszőleges Γ halmaza kielégíthető, ha van $\mathcal{L}_0$-nak olyan $\mathcal{I}$ interpretációja, melyre: $\forall A \in \Gamma : \mathcal{I} \models_0 A$. Egy ilyen $\mathcal{I}$ interpretáció modellje Γ-nak. Ha Γ-nak nincs modellje, akkor Γ kielégíthetetlen.

Lemma: Egy $\{A_1, A_2, ..., A_n\}$ formulahalmaznak pontosan azok az $\mathcal{I}$ interpretációk a modelljei, amelyek a $A_1 \wedge A_2 \wedge ... \wedge A_n$ formulának. Következésképpen $\{A_1, A_2, ..., A_n\}$ pontosan akkor kielégíthetetlen, ha az $A_1 \wedge A_2 \wedge ... \wedge A_n$ formula kielégíthetlen.

Szemantikus következmény: Legyen Γ ítéletlogikai formulák tetszőleges halmaza, B egy tetszőleges formula. Azt mondjuk, hogy a B formula *tautologikus következménye* a Γ formulahalmaznak (jelölés: $\Gamma \models_0 B$), ha minden olyan interpretáció, amely modellje Γ-nak, modellje B-nek is. A Γ-beli formulákat feltételformuláknak, vagy premisszáknak, a B formulát következményformulának (konklúziónak) hívjuk.

Tétel: Legyen Γ ítéletlogikai formulák tetszőleges halmaza, A,B,C tetszőleges ítéletlogikai formulák. Ha $\Gamma \models_0 A$, $\Gamma \models_0 B$ és $\{A, B\} \models_0 C$, akkor $\Gamma \models_0 C$.

Tétel: Legyenek $A_1, A_2, ..., A_n, B$ tetszőleges ítéletlogikai formulák. $\{A_1, A_2, ..., A_n\} \models_0 B$ pontosan akkor, ha a $\{A_1, A_2, ..., A_n, \neg B\}$ formulahalmaz kielégíthetetlen, azaz a $A_1 \wedge A_2 \wedge ... \wedge A_n \wedge \neg B$ formula kielégíthetetlen.

Tétel: Legyenek $A_1, A_2, ..., A_n, B$ tetszőleges ítéletlogikai formulák. $\{A_1, A_2, ..., A_n\} \models_0 B$ pontosan akkor, ha $\models_0 A_1 \wedge A_2 \wedge ... \wedge A_n \supset B$.

Ekvivalens átalakítások

Fogalmak:

1. Egy prímformulát (ítéletváltozót), vagy annak a negáltját közös néven *literálnak* nevezünk. A prímformula a *literál alapja*. Egy literált bizonyos esetekben *egységkonjunkciónak* vagy *egységdiszjunkciónak* (*egységklóznak*) is hívunk.

2. *Elemi konjunkció* az egységkonjunkció, illetve a különböző alapú literálok konjunkciója ($\wedge$ kapcsolat a literálok között). *Elemi diszjunkció* vagy *klóz* az egységdiszjunkció és a különböző alapú literálok diszjunkciója ($\vee$ kapcsolat a literálok között). Egy elemi konjunkció, illetve elemi diszjunkció *teljes* egy n-változós logikai műveletre nézve, ha mind az n ítéletváltozó alapja valamely literáljának.

3. *Diszjunktív normálformának* (DNF) nevezzük az elemi konjunkciók diszjunkcióját. *Konjunktív normálformának* (KNF) nevezzük az elemi diszjunkciók konjunkcióját. *Kitüntetett* diszjunktív, illetve konjunktív normálformákról (KDNF, ileltve KKNF) beszélünk, ha a bennük szereplő elemi konjunkciók, illetve elemi diszjunkciók teljesek.

Tetszőleges logikai műveletet leíró KDNF, KKNF előállítása: Legyen $b : \mathbb{L}^n \to \mathbb{L}$ egy n-változós logikai művelet. Adjuk meg b műveletábláját. Az első n oszlop fejlécébe az X_1, X_2, ... X_n ítéletváltozókat írjuk.

A b-t leíró KDNF előállítása:

1. Válasszuk ki azokat a sorokat a műveletáblában, ahol az adott igazságérték n-eshez b *igaz* értéket rendel hozzá. Legyenek ezek a sorok rendre $s_1, s_2, ..s_r$. Minden ilyen sorhoz rendeljünk hozzá egy $X'_1 \wedge X'_2 \wedge ... \wedge X'_n$ teljes elemi konjunkciót úgy, hogy az X'_j literál X_j vagy $\neg X_j$ legyen aszerint, hogy ebben a sorban X_j *igaz* vagy *hamis* igazságérték szerepel. Az így nyert teljes elemi konjunkciók legyenek rendre $k_{s_1}, k_{s_2}, ..k_{s_r}$.

2. Az így kapott teljes elemi konjunkciókból készítsünk egy diszjunkciós láncformulát: $k_{s_1} \vee k_{s_2} \vee ... \vee k_{s_r}$. Ez a formula lesz a b művelet kitüntetett diszjunktív normálformája (KDNF).

X	Y	Z	b		a teljes elemi konjunkciók
i	i	i	h		
i	i	h	i	$\star$	$X \wedge Y \wedge \neg Z$
i	h	i	h		
i	h	h	i	$\star$	$X \wedge \neg Y \wedge \neg Z$
h	i	i	i	$\star$	$\neg X \wedge Y \wedge Z$
h	i	h	h		
h	h	i	i	$\star$	$\neg X \wedge \neg Y \wedge Z$
h	h	h	i	$\star$	$\neg X \wedge \neg Y \wedge \neg Z$

ábra 6: Egy háromváltozós b logikai művelet műveletáblája és az előállított teljes elemi konjunkciók.

A fenti példa b műveletének kitüntetett diszjunktív normálformája a következő formula:
$(X \wedge Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge \neg Z) \vee (\neg X \wedge Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge \neg Z)$.

A b-t leíró KKNF előállítása:

1. Válasszuk ki azokat a sorokat a műveletáblában, ahol az adott igazságérték n-eshez b *hamis* értéket rendel hozzá. Legyenek ezek a sorok rendre $s_1, s_2, ..s_r$. Minden ilyen sorhoz rendeljünk hozzá egy $X'_1 \vee X'_2 \vee ... \vee X'_n$ teljes elemi diszjunkciót úgy, hogy az X'_j literál X_j vagy $\neg X_j$ legyen aszerint, hogy ebben a sorban X_j *hamis* vagy *igaz* igazságérték szerepel. Az így nyert teljes elemi diszjunkciók legyenek rendre $d_{s_1}, d_{s_2}, ..d_{s_r}$.

2. Az így kapott teljes elemi diszjunkciókból készítsünk egy konjunkciós láncformulát: $d_{s_1} \wedge d_{s_2} \wedge ... \wedge d_{s_r}$. Ez a formula lesz a b művelet kitüntetett konjunktív normálformája (KKNF).

X	Y	Z	b		a teljes elemi diszjunkciók
i	i	i	h	$\star$	$\neg X \vee \neg Y \vee \neg Z$
i	i	h	i		
i	h	i	h	$\star$	$\neg X \vee Y \vee \neg Z$
i	h	h	i		
h	i	i	i		
h	i	h	i		
h	h	i	h	$\star$	$X \vee Y \vee \neg Z$
h	h	h	i		

ábra 7: Egy háromváltozós b logikai művelet művelettáblája és az előállított teljes elemi diszjunkciók.

A fenti példa b műveletének kitüntetett konjunktív normálformája a következő formula:
$(\neg X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Y \vee \neg Z) \wedge (X \vee Y \vee \neg Z)$.

KNF, DNF egyszerűsítése: Egy ítéletlogikai formula logikai összetettségén a formulában szereplő logikai összekötőjelek számát értettük. Ugyanazt a logikai műveletet leíró formulák közül azt tekintjük egyszerűbbnek, amelynek kisebb a logikai összetettsége (azaz kevesebb logikai összekötőjelet tartalmaz).

Legyen X egy ítéletváltozó k egy az X-et nem tartalmazó elemi konjunkció, d egy X-et nem tartalmazó elemi diszjunkció. Ekkor az

- (a) $(X \wedge k) \vee (\neg X \wedge k) \sim_0 k$ és
- (b) $(X \vee d) \wedge (\neg X \vee d) \sim_0 d$

egyszerűsítési szabályok alkalmazásával konjunktív és diszjunktív normálformákat írhatunk át egyszerűbb alakba.

Klasszikus Quine–McCluskey-féle algoritmus KDNF egyszerűsítésére:

1. Soroljuk fel a KDNF-ben szereplő összes teljes elemi konjunkciót az L_0 listában, $j := 0$.
2. Megvizsgáljuk az L_j-ben szereplő összes lehetséges elemi konjunkciópárt, hogy alkalmazható-e rájuk az (a) egyszerűsítési szabály. Ha igen, akkor a két kiválasztott konjunkciót ✓-val megjelöljük, és az eredmény konjunkciót beírjuk a L_{j+1} listába. Azok az elemi konjunkciók, amelyek az L_j vizsgálata során nem lesznek megjelölve, nem voltak egyszerűsíthetők, tehát bekerülnek az egyszerűsített diszjunktív normálformába.
3. Ha az L_{j+1} konjunkciólista nem üres, akkor $j := j + 1$. Hajtsuk végre újból a 2. lépést.
4. Az algoritmus során kapott, de meg nem jelölt elemi konjunkciókból készítsünk egy diszjunkciós láncformulát. Így az eredeti KDNF-el logikailag ekvivalens, egyszerűsített DNF-et kapunk.

Rezolúció

Legyenek $A_1, A_2, ..., A_n, B$ tetszőleges ítéletlogikai formulák. Azt szeretnénk bebizonyítani, hogy $\{A_1, A_2, ..., A_n\} \models_0 B$, ami ekvivalens azzal, hogy $\{A_1, A_2, ..., A_n, \neg B\}$ kielégíthetetlen. Írjuk át ez utóbbi formulahalmaz formuláit KNF alakba! Ekkor a $\{KNF_{A_1}, KNF_{A_2}, ..., KNF_{A_n}, KNF_{\neg B}\}$ formulahalmazt kapjuk, ami pontosan akkor kielégíthetetlen, ha a halmaz formuláiban szereplő klózok halmaza kielégíthetetlen.

A klózokra vonatkozó egyszerűsítési szabály szerint ha X ítéletváltozó, C pedig X-et nem tartalmazó klóz, akkor $(X \vee C) \wedge (\neg X \vee C) \sim_0 C$. Az X és a $\neg X$ egységklózok (azt mondjuk, hogy X és $\neg X$ komplemens literálpár) konjunkciójával ekvivalens egyszerűbb, egyetlen literált sem tartalmazó klóz az üres klóz, melyet a $\square$ jellel jelölünk és definíció szerint minden interpretációban hamis igazságértékű.

Legyenek most C_1 és C_2 olyan klózok, melyek pontosan egy komplemens literálpárt tartalmaznak, azaz $C_1 = C_1' \vee L_1$ és $C_2 = C_2' \vee L_2$, ahol L_1 és L_2 az egyetlen komplemens literálpár (C_1' és C_2' üres klózok is lehetnek). Világos, hogy ha a két klózban a komplemens literálpáron kívül is vannak literálok, és ezek nem mind azonosak, az egyszerűsítési szabály alkalmazhatósági feltétele nem áll fenn.

Tétel: Ha $C_1 = C_1' \vee L_1$ és $C_2 = C_2' \vee L_2$, ahol L_1 és L_2 komplemens literálpár, akkor $\{C_1, C_2\} \models_0 C_1' \vee C_2'$

Rezolvens: Legyenek C_1 és C_2 olyan klózok, melyek pontosan egy komplemens literálpárt tartalmaznak, azaz $C_1 = C_1' \vee L_1$ és $C_2 = C_2' \vee L_2$, ahol L_1 és L_2 a komplemens literálpár, a $C_1' \vee C_2'$ klózt a (C_1, C_2) klózpár (vagy a $C_1 \vee C_2$ formula) *rezolvensének* nevezzük. Ha $C_1 = L_1$ és $C_2 = L_2$ (azaz C_1' és C_2' üres klózok), rezolvensük az üres klóz ($\square$). Az a tevékenység, melynek eredménye a rezolvens, a *rezolválás.*

	klózpár	rezolvens
(a)	$(X \vee Y,\ \neg Y \vee Z)$	$X \vee Z$
(b)	$(X \vee \neg Y,\ \neg Y \vee Z)$	nincs: mindkét azonos alapú literál negált
(c)	$(X \vee \neg Y,\ Z \vee \neg V)$	nincs: nincs azonos alapú literál
(d)	$(\neg X \vee \neg Y,\ X \vee Y \vee Z)$	nincs: két komplemens literálpár van
(e)	$(X,\ \neg X)$	$\square$

ábra 8: Példák klózpárok rezolválhatóságára, rezolvensére.

Tétel: Ha a C klóz a (C_1, C_2) klózpár rezolvense, akkor azon $\mathcal{I}$ interpretációk a $\{C_1, C_2\}$ klózhalmazt nem elégíthetik ki, amelyekben C igazságértéke hamis, azaz $\mathcal{B}_{\mathcal{I}}(C) = hamis$.

Rezolúciós levezetés: Egy S klózhalmazból a C klóz rezolúciós levezetése egy olyan véges $k_1, k_2, ..., k_m (m \geq 1)$ klózsorozat, ahol minden $j = 1, 2, ..., m$-re

1. vagy $k_j \in S$,
2. vagy van olyan $1 \leq s, t \leq j$, hogy k_j a (k_s, k_t) klózpár rezolvense,

és a klózsorozat utolsó tagja, k_m, éppen a C klóz.

Megállapodásunk szerint a rezolúciós kalkulus eldöntésproblémája az, hogy levezethető-e S-ből $\square$. A rezolúciós levezetés célja tehát $\square$ levezetése S-ből. Azt, hogy $\square$ levezethető S-ből, úgy is ki lehet fejezni, hogy létezik S-nek rezolúciós cáfolata.

Példa: Próbáljuk meg levezetni $\square$-t az $S = \{\neg X \vee Y, \neg Y \vee Z, X \vee Z, \neg V \vee Y \vee Z, \neg Z\}$ klózhalmazból. A levezetés bármelyik S-beli klózból indítható.

1.	$\neg V \vee Y \vee Z$	[$\in S$]
2.	$\neg Z$	[$\in S$]
3.	$\neg V \vee Y$	[1, 2 rezolvense]
4.	$\neg Y \vee Z$	[$\in S$]
5.	$\neg Y$	[2, 4 rezolvense]
6.	$\neg V$	[3, 5 rezolvense]
7.	$X \vee V$	[$\in S$]
8.	X	[6, 7 rezolvense]
9.	$\neg X \vee Y$	[$\in S$]
10.	Y	[8, 9 rezolvense]
11.	$\square$	[5, 10 rezolvense]

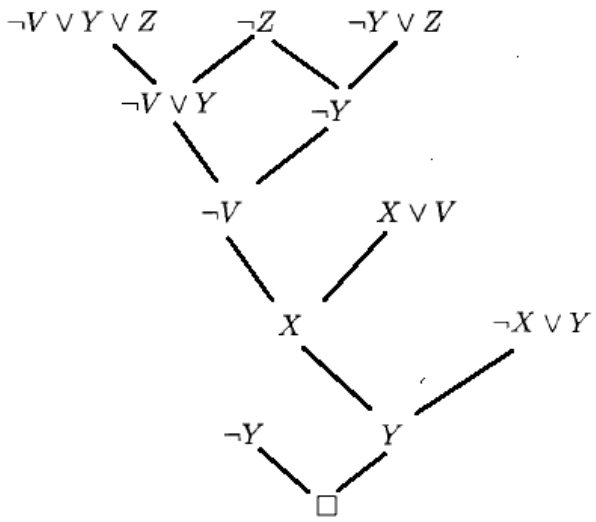
ábra 9: $\square$ rezolúciós levezetése S-ből.

Lemma: Legyen S tetszőleges klózhalmaz. S-ből történő rezolúciós levezetés esetén bármely S-ből levezetett klóz tautologikus következménye S-nek.

A rezolúciós kalkulus helyessége: A rezolúciós kalkulus *helyes*, azaz tetszőleges S klózhalmaz esetén amennyiben S-ből levezethető $\square$, akkor S *kielégíthetetlen*.

A rezolúciós kalkulus teljessége: A rezolúciós kalkulus *teljes*, azaz bármely véges, kielégíthetetlen S klózhalmaz esetén S-ből levezethető $\square$.

Levezetési fa: Egy rezolúciós levezetés szerkezetét *levezetési fa* segítségével szemléltethetjük. A levezetési fa csúcsai klózok. Két csúcsból pontosan akkor vezet él egy harmadik, közös csúcsba, ha az a két klóz rezolvense.



ábra 10: Az előző példa levezetési fája.

Rezolúciós stratégiák:

- *Lineáris rezolúció*: Egy S klózhalmazból való lineáris rezolúciós levezetés egy olyan $k_1, l_1, k_2, l_2, ..., k_{m-1}, l_{m-1}, k_m$ rezolúciós levezetés, amelyben minden $j = 2, 3, ..., m$-re k_j a (k_{j-1}, l_{j-1}) klózpár rezolvense. A k_j klózokat centrális klózoknak, az l_j klózokat mellékklózoknak nevezzük.

  Tetszőleges rezolúciós levezetés átírható lineárissá, azaz a lineáris rezolúciós kalkulus teljes.

- *Lineáris inputrezolúció*: Egy S klózhalmazból való lineáris inputrezolúciós levezetés egy olyan $k_1, l_1, k_2, l_2, ..., k_{m-1}, l_{m-1}, k_m$ lineáris rezolúciós levezetés, amelyben minden $j = 1, 2, ..., m-1$-re $l_j \in S$, azaz a lineáris inputrezolúciós levezetésben a mellékklózok S elemei.

  A lineáris inputrezolúciós stratégia nem teljes, de megadható olyan formulaosztály, melyre az. A legfeljebb egy negált literált tartalmazó klózokat Horn-klózoknak nevezzük, a Horn-formulák pedig azok a formulák, melyek konjunktív normálformája Horn-klózok konjunkciója. A lineáris inputrezolúciós stratégia Horn-formulák esetén teljes.

1.3 Predikátumkalkulus

1.3.1 Elsőrendű logikai nyelvek szintaxisa

Egy elsőrendű logikai nyelv ábécéje logikai és logikán kívüli szimbólumokat, továbbá elválasztójeleket tartalmaz. A logikán kívüli szimbólumhalmaz megadható $< Srt, Pr, Fn, Cnst >$ alakban, ahol:

1. Srt nemüres halmaz, elemei fajtákat szimbolizálnak,
2. Pr nemüres halmaz, elemei predikátumszimbólumok,
3. az Fn halmaz elemei függvényszimbólumok,

4. $Cnst$ pedig a függvényszimbólumok halmaza.

Az $< Srt, Pr, Fn, Cnst >$ ábécé szignatúrája egy $< \nu_1, \nu_2, \nu_3 >$ hármas, ahol

1. minden $P \in Pr$ predikátumszimbólumhoz ν_1 a predikátumszimbólum alakját, azaz a $(\pi_1, \pi_2, ..., \pi_k)$ fajtasorozatot,

2. minden $f \in Fn$ függvényszimbólumhoz ν_2 a függvényszimbólum alakját, azaz a $(\pi_1, \pi_2, ..., \pi_k, \pi)$ fajtasorozatot és

3. minden $c \in Cnst$ konstansszimbólumhoz ν_3 a konstansszimbólumhoz alakját, azaz (π)-t

rendel ($k > 0$ és $\pi_1, \pi_2, ..., \pi_k, \pi \in Srt$).

Logikai jelek az ítéletlogikában is használt logikai összekötőjelek, valamint az univerzális ($\forall$) és egzisztenciális ($\exists$) kvantorok és a különböző fajtájú individuumváltozók. Egy elsőrendű nyelv ábécéjében minden $\pi \in Srt$ fajtához szimbólumoknak megszámlálhatóan végtelen $v_1^\pi, v_2^\pi, ...$ rendszere tartozik, ezeket a szimbólumokat nevezzük π fajtájú változóknak. Elválasztójel a nyitó és csukó zárójelek, és a vessző.

Az elsőrendű logikai nyelvekben az elválasztójelek és a logikai jelek mindig ugyanazok, viszont a logikán kívüli jelek halmaza, illetve ezek szignatúrája nyelvről nyelvre lényegesen különbözhet. Ezért mindig megadjuk a $< Srt, Pr, Fn, Cnst >$ négyest és ennek $< \nu_1, \nu_2, \nu_3 >$ szignatúráját, amikor egy elsőrendű logikai nyelv ábécéjére hivatkozunk. Jelölése $V[V_\nu]$, ahol V_ν adja meg a $< \nu_1, \nu_2, \nu_3 >$ szignatúrájú $< Srt, Pr, Fn, Cnst >$ négyest.

Termek: A $V[V_\nu]$ ábécé feletti termek halmaza $\mathcal{L}_t[V_\nu]$, ami a következő tulajdonságokkal bír:

1. Minden $\pi \in Srt$ fajtájú változó és konstans π fajtájú term.

2. Ha az $f \in Fn$ függvényszimbólum $(\pi_1, \pi_2, ..., \pi_k, \pi)$ alakú és $t_1, t_2, ..., t_k$ – rendre $\pi_1, \pi_2, ..., \pi_k$ fajtájú – termek, akkor az $f(s_1, s_2, ..., s_k)$ egy π fajtájú term.

3. Minden term az 1-2. szabályok véges sokszori alkalmazásával áll elő.

Formulák: A $V[V_\nu]$ ábécé feletti elsőrendű formulák halmaza $\mathcal{L}_f[V_\nu]$, ami a következő tulajdonságokkal bír:

1. Ha a $P \in Pr$ predikátumszimbólum $(\pi_1, \pi_2, ..., \pi_k)$ alakú és az $t_1, t_2, ..., t_k$ – rendre $\pi_1, \pi_2, ..., \pi_k$ fajtájú – termek, akkor a $P(t_1, t_2, ..., t_k)$ szó egy elsőrendű formula. Az így nyert formulákat atomi formuláknak nevezzük.

2. Ha S elsőrendű formula, akkor $\neg S$ is az.

3. Ha S és T elsőrendű formulák és $\circ$ binér logikai összekötőjel, akkor $(S \circ T)$ is elsőrendű formula.

4. Ha S eleme elsőrendű formula, Q kvantor ($\forall$ vagy $\exists$) és x tetszőleges változó, akkor QxS is elsőrendű formula. Az így nyert formulákat kvantált formuláknak nevezzük, a $\forall xS$ alakú formulák univerzálisan kvantált formulák, a $\exists xS$ alakú formulák pedig egzisztenciálisan kvantált formulák. A kvantált formulákban Qx a formula prefixe, S pedig a magja.

5. Minden elsőrendű formula az 1-4. szabályok véges sokszori alkalmazásával áll elő.

A $V[V_\nu]$ ábécé feletti elsőrendű logikai nyelv $\mathcal{L}[V_\nu] = \mathcal{L}_t[V_\nu] \cup \mathcal{L}_f[V_\nu]$, azaz $\mathcal{L}[V_\nu]$ minden szava vagy term, vagy formula.

A negációs, konjunkciós, diszjunkciós, implikációs (ezek jelentése ua., mint nulladrendben) és kvantált formulák összetett formulák.

Az elsőrendű logikai nyelv prímformulái az atomi formulák és a kvantált formulák.

Változóelőfordulás fajtái: Egy formula x változójának egy előfordulása:

- szabad, ha nem esik x-re vonatkozó kvantor hatáskörébe,
- kötött, ha x-re vonatkozó kvantor hatáskörébe esik.

Változó fajtái: Egy formula x változója:

- szabad, ha minden előfordulása szabad,
- kötött, ha minden előfordulása kötött, és
- vegyes, ha van szabad és kötött előfordulása is.

Formula zártsága, nyíltsága: Egy formula:

- zárt, ha minden változója kötött,
- nyílt, ha legalább egy változójának van szabad előfordulása és
- kvantormentes, ha nincs benne kvantor

Megjegyzés: a zárt formulák elsőrendű állításokat szimbolizálnak (egy elsőrendű állítás nem más, mint elemek egy halmazára megfogalmazott kijelentő mondat).

1.3.2 Az elsőrendű logika szemantikája

Matematikai struktúra: Matematikai struktúrán egy $< U, R, M, K >$ négyest értünk, ahol:

1. $U = \bigcup_\pi U_\pi$ nem üres alaphalmaz (univerzum),
2. R az U-n értelmezett logikai függvények (relációk) halmaza,
3. M az U-n értelmezett matematikai függvények (alapműveletek) halmaza,
4. K az U kijelölt elemeinek (konstansainak) halmaza (lehet üres).

Interpretáció: Az interpretáció egy $< U, R, M, K >$ matematikai struktúra és $\mathcal{I} =< \mathcal{I}_{Srt}, \mathcal{I}_{Pr}, \mathcal{I}_{Fn}, \mathcal{I}_{Cnst} >$ függvénynégyes, ahol:

- az $\mathcal{I}_{Srt} : \pi \mapsto U_\pi$ függvény megad minden egyes $\pi \in Srt$ fajtához egy U_π nemüres halmazt, a π fajtájú individuumok halmazát,
- az $\mathcal{I}_{Pr} : P \mapsto P^{\mathcal{I}}$ függvény megad minden $(\pi_1, \pi_2, ..., \pi_k)$ alakú $P \in Pr$ predikátumszimbólumhoz egy $P^{\mathcal{I}} : U_{\pi_1} \times U_{\pi_2} \times ... \times U_{\pi_k} \to \mathbb{L}$ logikai függvényt (relációt),
- az $\mathcal{I}_{Fn} : f \mapsto f^{\mathcal{I}}$ függvény hozzárendel minden $(\pi_1, \pi_2, ..., \pi_k, \pi)$ alakú $f \in Fn$ függvényszimbólumhoz egy $P^{\mathcal{I}} : U_{\pi_1} \times U_{\pi_2} \times ... \times U_{\pi_k} \to U_\pi$ matematikai függvényt (műveletet),
- az $\mathcal{I}_{Cnst} : c \mapsto ct^{\mathcal{I}}$ pedig minden π fajtájú $c \in Cnst$ konstansszimbólumhoz az U_π individuumtartománynak egy individuumát rendeli, azaz $c^{\mathcal{I}} \in U_\pi$.

Változókiértékelés: Legyen az $\mathcal{L}[V_\nu]$ nyelvnek $\mathcal{I}$ egy interpretációja, az interpretáció univerzuma legyen U és jelölje V a nyelv változóinak halmazát. Egy olyan $\kappa : V \to U$ leképezést, ahol ha x π fajtájú változó, akkor $\kappa(x) \in U_\pi$, $\mathcal{I}$-beli változókiértékelésnek nevezünk.

$\mathcal{L}_t[V_\nu]$ **szemantikája**: Legyen az $\mathcal{L}[V_\nu]$ nyelvnek $\mathcal{I}$ egy interpretációja és κ egy $\mathcal{I}$-beli változókiértékelés. Az $\mathcal{L}[V_\nu]$ nyelv egy π fajtájú t termjének értéke $\mathcal{I}$-ben a κ változókiértékelés mellett az alábbi – $|t|^{\mathcal{I},\kappa}$-val jelölt – U_π-beli individuum:

1. ha $c \in Cnst$ π fajtájú konstansszimbólum, akkor $|c|^{\mathcal{I},\kappa}$ az U_π-beli $c^{\mathcal{I}}$ individuum,
2. ha x π fajtájú változó, akkor $|x|^{\mathcal{I},\kappa}$ az U_π-beli $\kappa(x)$ individuum,
3. ha $t_1, t_2, ..., t_k$ rendre $\pi_1, \pi_2, ..., \pi_k$ fajtájú termek és ezek értékei a κ változókiértékelés mellett rendre az U_{π_1}-beli $|t_1|^{\mathcal{I},\kappa}$, az U_{π_2}-beli $|t_2|^{\mathcal{I},\kappa}$ ... és az U_{π_k}-beli $|t_k|^{\mathcal{I},\kappa}$ individuumok, akkor egy $(\pi_1, \pi_2, ..., \pi_k, \pi)$ alakú $f \in Fn$ függvényszimbólum esetén $|f(t_1, t_2, ..., t_k)|^{\mathcal{I},\kappa}$ az U_π-beli $f^{\mathcal{I}}(|t_1|^{\mathcal{I},\kappa}, |t_2|^{\mathcal{I},\kappa}, ..., |t_k|^{\mathcal{I},\kappa})$ individuum.

Változókiértékelés x-variánsa: Legyen x egy változó. A κ^* változókiértékelés a κ változókiértékelés x-variánsa, ha $\kappa^*(y) = y$ minden x-től különböző y változó esetén.

Elsőrendű logikai formula logikai értéke: Legyen az $\mathcal{L}[V_\nu]$ nyelvnek $\mathcal{I}$ egy interpretációja és κ egy $\mathcal{I}$-beli változókiértékelés. Az $\mathcal{L}[V_\nu]$ nyelv egy C formulájához $\mathcal{I}$-ben a κ változókiértékelés mellett az alábbi – $|C|^{\mathcal{I},\kappa}$-val jelölt – igazságértéket rendeljük:

1. $|P(t_1, t_2, ..., t_k)|^{\mathcal{I},\kappa} = \left\{ \begin{array}{ll} igaz & : P^{\mathcal{I}}(|t_1|^{\mathcal{I},\kappa}, |t_2|^{\mathcal{I},\kappa}, ..., |t_k|^{\mathcal{I},\kappa}) = igaz \\ hamis & : kulonben \end{array} \right\}$
2. $|\neg A|^{\mathcal{I},\kappa}$ legyen $\neg|A|^{\mathcal{I},\kappa}$
3. $|A \wedge B|^{\mathcal{I},\kappa}$ legyen $|A|^{\mathcal{I},\kappa} \wedge |B|^{\mathcal{I},\kappa}$
4. $|A \vee B|^{\mathcal{I},\kappa}$ legyen $|A|^{\mathcal{I},\kappa} \vee |B|^{\mathcal{I},\kappa}$
5. $|A \supset B|^{\mathcal{I},\kappa}$ legyen $|A|^{\mathcal{I},\kappa} \supset |B|^{\mathcal{I},\kappa}$
6. $|\forall x A|^{\mathcal{I},\kappa} = \left\{ \begin{array}{ll} igaz & : |A|^{\mathcal{I},\kappa^*} = igaz\ \kappa\ minden\ \kappa^*\ x - variansara \\ hamis & : kulonben \end{array} \right\}$
7. $|\exists x A|^{\mathcal{I},\kappa} = \left\{ \begin{array}{ll} igaz & : |A|^{\mathcal{I},\kappa^*} = igaz\ \kappa\ valamely\ \kappa^*\ x - variansara \\ hamis & : kulonben \end{array} \right\}$

Elsőrendű formula kielégíthetősége: Egy A elsőrendű formula kielégíthető, ha van olyan $\mathcal{I}$ interpretáció és κ változókiértékelés, amelyre $|A|^{\mathcal{I},\kappa} = igaz$ (ekkor azt mondjuk, hogy az $\mathcal{I}$ interpretáció és κ változókiértékelés kielégíti A-t), különben kielégíthetetlen.

Amennyiben az A formula zárt, igazságértékét egyedül az interpretáció határozza meg. Ha $|A|^{\mathcal{I}} = igaz$, azt mondjuk, hogy az $\mathcal{I}$ kielégíti A-t vagy másképpen: $\mathcal{I}$ modellje A-nak ($\mathcal{I} \models A$).

Logikailag igaz elsőrendű formula: Egy A elsőrendű logikai formula logikailag igaz, ha minden $\mathcal{I}$ interpretációban és $\mathcal{I}$ minden κ változókiértékelése mellett $|A|^{\mathcal{I},\kappa} = igaz$. Jelölése: $\models A$.

Szemantikus következmény: Azt mondjuk, hogy a G formula *szemantikus következménye* az $\mathcal{F}$ formulahalmaznak, ha minden olyan $\mathcal{I}$ interpretációra, amelyre $\mathcal{I} \models \mathcal{F}$ fennáll, $\mathcal{I} \models G$ is igaz (jelölés: $\mathcal{F} \models G$).

Tétel: Legyenek $A_1, A_2, ..., A_n, B$ ($n \geq 1$) tetszőleges, ugyanabból az elsőrendű logikai nyelvből való formulák. Ekkor $\{A_1, A_2, ..., A_n\} \models B$ akkor és csak akkor, ha $A_1 \wedge A_2 \wedge ... \wedge A_n \wedge \neg B$ kielégíthetetlen.

Rezolúció: Elsőrendű predikátumkalkulusban is végezhető rezolúció, ráadásul a módszer helyes és teljes is. Nehézséget a klózok kialakítása okozhat, amelyek zárt, univerzálisan kvantált literálok konjunkciójából állnak. Ehhez eszközeink a prenex-, illetve skolem-formák.

2 Számításelmélet

2.1 Kiszámíthatóság

2.1.1 Algoritmusmodellek

- **Gödel**: rekurzív függvények (primitív rekurzív függvények 1931-ben, majd általánosabb 1934-ben)
- **Church**: λ-kalkulus, λ-definiálható függvények: ekvivalensek a rekurzív függvényekkel (bizonyított)
- **Turing**: Turing-gép (1936), a λ-definiálható és a Turing-géppel kiszámítható függvények megegyeznek (bizonyított)

Church-Turing tézis: A kiszámíthatóság különböző matematikai modelljei mind az effektíven kiszámítható függvények osztályát definiálják.

2.1.2 Fogalmak

Kiszámítási problémának nevezünk egy olyan, a matematika nyelvén megfogalmazott kérdést, amire egy algoritmussal szeretnénk megadni a választ. A gyakorlati élet szinte minden problémájához rendelhető, megfelelő absztrakciót használva, egy kiszámítási probléma.

Egy problémát a hozzá tartozó konkrét bementettel együtt a probléma egy példányának nevezzük.

Speciális kiszámítási probléma az eldöntési probléma. Ilyenkor a problémával kapcsolatos kérdés egy eldöntendő kérdés, tehát a probléma egy példányára a válasz "igen" vagy "nem" lesz.

Egy kiszámítási probléma reprezentálható egy $f : A \to B$ függvénnyel. Az A halmaz tartalmazza a probléma egyes bemeneteit, jellemzően egy megfelelő ábécé feletti szóban elkódolva, míg a B halmaz tartalmazza a bemenetekre adott válaszokat, szintén valamely alkalmas ábécé feletti szóban elkódolva. Értelemszerűen, ha eldöntési problémáról van szó, akkor az f értékkészlete, vagyis a B egy két elemű halmaz: $\{igen, nem\}$, $\{1, 0\}$, stb.

Kiszámítható függvény: Egy $f : A \to B$ függvényt *kiszámíthatónak* nevezünk, ha minden $x \in A$ elemre az $f(x) \in B$ függvényérték kiszámítható valamilyen algoritmikus modellel.

Megoldható, eldönthető probléma: Egy kiszámítási probléma *megoldható* (eldöntési probléma esetén azt mondjuk, hogy *eldönthető*), ha az általa meghatározott függvény kiszámítható.

Algoritmusok időigénye: Legyenek $f, g : \mathbb{N} \to \mathbb{N}$ függvények, ahol $\mathbb{N}$ a természetes számok halmaza. Azt mondjuk, hogy f legfeljebb olyan gyorsan nő, mint g (jelölése: $f(n) = \mathcal{O}(g(n))$), ha $\exists c > 0$ és $n_0 \in \mathbb{N}$, hogy $f(n) \leq c * g(n)\ \forall n \geq n_0$. Az $f(n) = \Omega(g(n))$ jelöli azt, hogy $g(n) = \mathcal{O}(f(n))$ teljesül és $f(n) = \Theta(g(n))$ jelöli azt, hogy $f(n) = \mathcal{O}(g(n))$ és $f(n) = \Omega(g(n))$ is teljesül.

Példa: $3n^3 + 5n^2 + 6 = \mathcal{O}(n^3)$, $n^k = \mathcal{O}(2^n)\ \forall k \geq 0$, stb.

Tétel: Minden polinomiális függvény lassabban nő, mint bármely exponenciális függvény, azaz minden $p(n)$ polinomhoz és $c > 0$-hoz $\exists n_0$ egész szám, hogy $\forall n \geq n_0$ esetén $p(n) \leq 2^{cn}$

Kiszámítási probléma megfeleltetése eldöntési problémának: Tekintsünk egy P kiszámítási problémát és legyen $f : A \to B$ a P által meghatározott függvény. Ekkor megadható P-hez egy P' eldöntési probléma úgy, hogy P' pontosan akkor eldönthető, ha P kiszámítható. Állítsuk párba ugyanis minden $a \in A$ elemre az a és $f(a)$ elemeket, és kódoljuk el az így kapott párokat egy-egy szóban. Ezek után legyen P' az így kapott szavakból képzett formális nyelv. Nyilvánvaló, hogy ha minden $a \in A$ és $b \in B$ elemre az $(a, b) \in P'$ tartalmazás eldönthető (azaz P' eldönthető), akkor P kiszámítható és fordítva. E megfeleltetés miatt a továbbiakban jellemzően eldöntési problémákkal foglalkozunk.

2.2 Turing-gépek

Hasonlóan a véges automatához vagy a veremautomatához, a Turing-gép is egy véges sok állapottal rendelkező eszköz. A Turing-gép egy két irányban végtelen szalagon dolgozik. A szalag cellákra van osztva, tulajdonképpen ez a gép (korlátlan) memóriája. Kezdetben a szalagon csak a bemenő szó van, minden cellán egy betű. A szalag többi cellája egy úgynevezett blank vagy szóköz ($\sqcup$) szimbólumokkal van feltöltve. Kezdetben a gép úgynevezett író-olvasó feje a bemenő szó első betűjén áll és a gép a kezdőállapotában van. A gép az író-olvasó fejet tetszőlegesen képes mozgatni a szalagon. Képes továbbá a fej pozíciójában a szalag tartalmát kiolvasni és átírni. A gépnek van két kitüntetett állapota, a q_i és a q_n állapotok. Ha ezekbe az állapotokba kerül, akkor rendre elfogadja illetve elutasítja a bemenő szót. Formálisan a Turing-gépet a következő módon definiáljuk.

A Turing-gép formális definíciója: A Turing-gép egy olyan $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ rendszer, ahol:

- Q az állapotok véges, nem üres halmaza,
- $q_0, q_i, q_n \in Q$, q_0 a kezdőállapot, q_i az elfogadó állapot, q_n pedig az elutasító állapot,

- Σ és Γ ábécék, a bemenő jelek és a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\Gamma - \Sigma$ tartalmaz egy speciális $\sqcup$ szimbólumot,
- $\delta : (Q - \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$ az átmenetfüggvény.

Úgy mint a veremautomaták esetében, egy M Turing-gép működésének fázisait is konfigurációkkal írhatjuk le.

Turing-gép konfigurációja: Az M Turing-gép konfigurációja egy olyan uqv szó, ahol $q \in Q$ és $u, v \in \Gamma^*$, $v \neq \varepsilon$. Ez a konfiguráció az M azon állapotát tükrözi amikor a szalag tartalma uv (uv előtt és után a szalagon már csak $\sqcup$ van), a gép a q állapotban van, és az író-olvasó fej a v első betűjére mutat. M összes konfigurációjának halmazát $\mathcal{C}_M$-el jelöljük.

Turing-gép kezdőkonfigurációja: M kezdőkonfigurációja egy olyan $q_0u\sqcup$ szó, ahol u csak Σ-beli betűket tartalmaz.

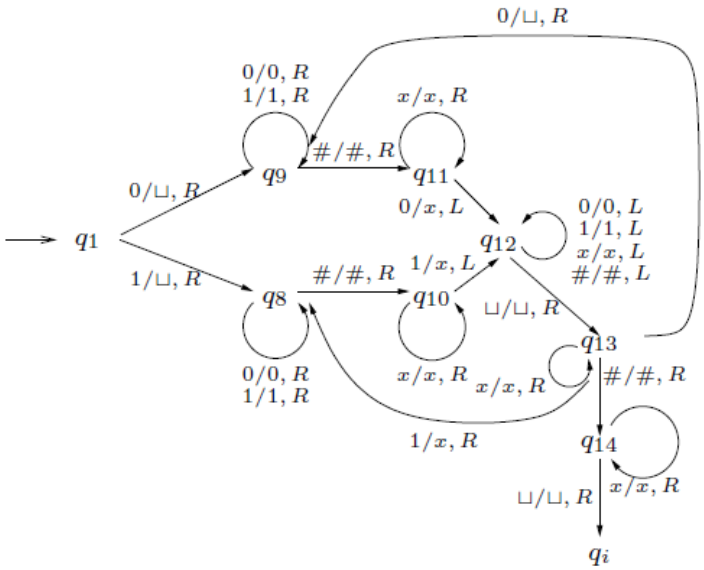
Turing-gép konfigurációátmenete: M konfigurációátmenete egy olyan $\vdash \subseteq \mathcal{C}_M \times \mathcal{C}_M$ reláció, amit a következőképpen definiálunk. Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$ és $u, v \in \Gamma^*$. A következő három esetet különböztetjük meg:

1. Ha $\delta(q, a) = (r, b, S)$, akkor $uqav \vdash urbv$.
2. Ha $\delta(q, a) = (r, b, R)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$.
3. Ha $\delta(q, a) = (r, b, L)$, akkor $uqav \vdash u'rcbv$, ahol $u'c = u$ valamely $u' \in \Gamma^*$-ra és $c \in \Gamma$-ra, ha $u \neq \varepsilon$, egyébként pedig $u' = \varepsilon$, $c = \sqcup$.

Azt mondjuk, hogy M véges sok lépésben eljut a C konfigurációból a C' konfigurációba (jele $C \vdash^* C'$), ha létezik olyan $n \geq 0$ és $C_1, ...C_n$ konfigurációsorozat, hogy $C_1 = C$, $C_n = C'$ és minden $1 \leq i < n$-re $C_i \vdash C_{i+1}$.

Ha $q \in \{q_i, q_n\}$, akkor azt mondjuk, hogy az uqv konfiguráció egy megállási konfiguráció. Továbbá, $q = q_i$ esetében elfogadó, míg $q = q_n$ esetében elutasító konfigurációról beszélünk.

Turing-gép által felismert nyelv: Az M Turing-gép által felismert nyelv (jelölése $L(M)$) azoknak az $u \in \Sigma^*$ szavaknak a halmaza, melyekre igaz, hogy $q_0u\sqcup \vdash^* xq_iy$ valamely $x, y \in \Gamma^*$, $y \neq \varepsilon$ szavakra.



ábra 11: Egy, az $L = \left\{u\#u \mid u \in \{0,1\}^+\right\}$ felismerő Turing-gép.

Turing-gépek ekvivalenciája: Két Turing-gépet ekvivalensnek nevezünk, ha ugyanazt a nyelvet ismerik fel.

Turing-felismerhető nyelv, rekurzívan felismerhető nyelvek osztálya: Egy $L \subseteq \Sigma^*$ nyelv Turing-felismerhető, ha $L = L(M)$ valamely M Turing-gépre. A Turing-felismerhető nyelveket szokás *rekurzívan felsorolhatónak* is nevezni. A rekurzívan felsorolható nyelvek osztályát RE-vel jelöljük.

Turing-eldönthető nyelv, rekurzív nyelvek osztálya: Egy $L \subseteq \Sigma^*$ nyelv Turing-eldönthető, ha létezik olyan Turing-gép, amely minden bemeneten megállási konfigurációba jut és felismeri L-et. A Turing-felismerhető nyelveket szokás *rekurzívnak* is nevezni. A rekurzív nyelvek osztályát R-rel jelöljük.

Turing-gép futási ideje, időigénye: Tekintsünk egy $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ Turing-gépet és annak egy $u \in \Sigma^*$ bemenő szavát. Azt mondjuk, hogy M futási ideje (időigénye) az u szón n ($n \geq 0$), ha M a $q_0 u \sqcup$ kezdőkonfigurációból n lépésben el tud jutni egy megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u szón végtelen.

Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M időigénye $f(n)$ (vagy azt, hogy M egy $f(n)$ időkorlátos gép), ha minden $u \in \Sigma^*$ input szóra M időigénye az u szón legfeljebb $f(l(u))$.

2.2.1 Többszalagos Turing-gépek

A többszalagos Turing-gépek, értelemszerűen, egynél több szalaggal rendelkeznek. Mindegyik szalaghoz tartozik egy-egy író-olvasó fej, melyek egymástól függetlenül képesek mozogni a szalagon.

Többszalagos Turing-gép definíciója: Legyen $k > 1$. Egy k-szalagos Turing-gép egy olyan $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ rendszer, ahol a komponensek a δ kivételével megegyeznek az egyszalagos Turing-gép komponenseivel, δ pedig a következőképpen adódik. $\delta : (Q - \{q_i, q_n\}) \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k$. Legyenek $q, p \in Q$, $a_1, a_2, ..., a_k, b_1, b_2, ..., b_k \in \Gamma$ és $D_1, D_2, ..., D_k \in \{L, R, S\}$. Ha $\delta(q, a_1, a_2, ..., a_k) = (p, b_1, b_2, ..., b_k, D_1, D_2, ..., D_k)$, akkor a gép akkor a gép a q állapotból, ha a szalagjain rendre az $a_1, a_2, ..., a_k$ betűket olvassa, át tud menni a p állapotba, miközben az $a_1, a_2, ..., a_k$ betűket átírja a $b_1, b_2, ..., b_k$ betűkre és a szalagokon a fejeket $D_1, D_2, ..., D_k$ irányokba mozgatja.

A többszalagos Turing-gép konfigurációja, a konfigurációátmenet valamint a felismert illetve eldöntött nyelv definíciója az egyszalagos eset értelemszerű általánosítása. A többszalagos Turing-gép időigényét is az egyszalagoshoz hasonlóan definiáljuk.

Többszalagos és egyszalagos gépek ekvivalenciája: Minden k-szalagos, $f(n)$ időkorlátos Turing-géphez van vele ekvivalens $\mathcal{O}(n * f(n))$ időkorlátos egyszalagos Turing-gép.

2.2.2 Nemdeterminisztikus Turing-gépek

Egy M nemdeterminisztikus Turing-gép állapotfüggvénye $\delta : (Q - \{q_i, q_n\}) \times \mathcal{P}(\Gamma \to Q \times \Gamma \times \{L, R\})$ alakú. Tehát M minden konfigurációjából néhány (esetleg nulla) különböző konfigurációba mehet át. Ily módon M számítási sorozatai egy u szón egy fával reprezentálhatók. A fa csúcsa M kezdőkonfigurációja, a szögpontjai pedig M konfigurációi. A fa minden levele megfelel M egy számítási sorozatának az u-n. M akkor fogadja el u-t, ha a fa valamelyik levele elfogadó konfiguráció. Nevezzük ezt a most leírt fát az M nemdeterminisztikus számítási fájának az u-n. Az M által felismert nyelv a determinisztikus esethez hasonlóan definiálható, a gép által eldöntött nyelv pedig a következőképpen.

Nemdeterminisztikus Turing-gép által eldöntött nyelv: Azt mondjuk, hogy egy nemdeterminisztikus M Turing-gép eldönt egy $L \subseteq \Gamma^*$ nyelvet, ha felismeri, és minden $u \in \Sigma*$ szóra M számítási sorozatai végesek és elfogadási vagy elutasítási konfigurációba vezetnek.

Nemdeterminisztikus Turing-gép időigénye: Legyen $f : \mathbb{N} \to \mathbb{N}$ függvény, M egy nemdeterminisztikus Turing-gép. Az M időigénye $f(n)$, ha egy n hosszú u bemeneten nincsenek M-nek $f(n)$-nél hosszabb számítási sorozatai, azaz az M számítási fája az u-n legfeljebb $f(n)$ magas.

Determinisztikus és nemdeterminisztikus Turing-gépek ekvivalenciája: Minden M nemdeterminisztikus Turing-géphez megadható egy ekvivalens M' determinisztikus Turing-gép. Továbbá, ha M $f(n)$ időigényű valamely $f : \mathbb{N} \to \mathbb{N}$ függvényre, akkor M' $2^{\mathcal{O}(f(n))}$ időigényű.

2.3 Eldönthetetlen problémák

Ebben a fejezetben megmutatjuk, hogy bár a Turing-gép a lehető legáltalánosabb algoritmus modell, mégis vannak olyan problémák, melyek nem számíthatók ki Turing-géppel.

Emlékeztető: A rekurzívan felsorolható (Turing-felismerhető) nyelvek osztályát RE-vel, a rekurzív (Turing-eldönthető) nyelvek osztályát R-rel jelöljük.

Világos, hogy $R \subseteq RE$. A célunk az, hogy megmutassuk: az R valódi részhalmaza az RE-nek, azaz van olyan nyelv (probléma) ami Turing-felismerhető, de nem eldönthető.

Csak olyan Turing-gépeket fogunk vizsgálni, melyek bemenő ábécéje a $\{0,1\}$ halmaz. Ez nem jelenti az általánosság megszorítását, hiszen ha találunk egy olyan $\{0,1\}$ feletti nyelvet, melyet nem lehet eldönteni ilyen Turing-géppel, akkor ezt a nyelvet egyáltalán nem lehet eldönteni.

2.3.1 Turing-gépek kódolása

A $\{0,1\}$ feletti szavak felsorolhatóak (vagyis megszámlálhatóak). Valóban, tekintsük azt a felsorolást, amelyben a szavak a hosszuk szerint követik egymást, és két egyforma hosszú szó közül pedig az van előbb, amelyik az alfabetikus rendezés szerint megelőzi a másikat. Ily módon a $\{0,1\}^*$ halmaz elemeinek egy felsorolása a következőképpen alakul: $w_1 = \varepsilon$, $w_2 = 0$, $w_3 = 1$, $w_4 = 00$, $w_5 = 01$ és így tovább. Ebben a fejezetben tehát a w_i szóval a $\{0,1\}^*$ i. elemét jelöljük.

Legyen továbbá M egy $\{0,1\}$ inputábécé feletti Turing-gép. Van olyan $k > 0$ szám, hogy Q-t felírhatjuk $Q = \{p_1, ...p_k\}$ alakban, ahol $p_1 = q_0$, $p_{k-1} = q_i$, $p_k = q_n$. Továbbá, van olyan $m > 0$ szám, hogy Γ-t felírhatjuk $\Gamma = \{X_1, ...X_m\}$ alakban, ahol $X_1 = 0$, $X_2 = 1$, $X_3 = \sqcup$, és $X_4, ...X_m$ az M további szalagszimbólumai. Nevezzük végül az L, R, S szimbólumokat (amelyek irányokat jelölnek) rendre D_1, D_2 és D_3-nak. Ezek után M egy $\delta(p_i, X_j) = (p_r, X_s, D_t)$ $(0 \leq i, r \leq k,\ 1 \leq j, s \leq m$ és $1 \leq t \leq 3)$ átmenete elkódolható a $0^i10^j10^r10^s10^t$ szóval. Mivel minden 0-s blokk hossza legalább 1, az átmenetet kódoló szóban nem szerepel az 11 részszó. Tehát az M összes átmenetét kódoló szavakat összefűzhetjük egy olyan szóvá, melyben az átmeneteket az 11 részszó választja el egymástól. Az így kapott szó pedig magát M-et kódolja.

A továbbiakban M_i-vel jelöljük azt a Turing-gépet, amelyet a w_i szó kódol $(i \geq 1)$. Amennyiben w_i nem a fent leírt kódolása egy Turing-gépnek, akkor tekintsük M_i-t olyannak, ami minden input esetén azonnal a q_n állapotba megy, azaz $L(M_i) = \emptyset$.

A későbbiekben szükségünk lesz arra, hogy elkódoljunk egy (M, w) Turing-gép és bemenet párost egy $\{0,1\}$ feletti szóban. Mivel a Turing-gépek kódolása nem tartalmazhat 111-et, ezért (M, w) kódja a következő: M kódja után írunk 111-et, majd utána w-t.

2.3.2 Egy nem rekurzívan felsorolható nyelv

Az $L_{átló}$ nyelv: Az $L_{átló}$ nyelv azon $\{0,1\}$ feletti Turing-gépek bináris kódjait tartalmazza, melyek nem fogadják el önmaguk kódját, mint bemenő szót, azaz $L_{átló} = \{w_i \mid i \geq 1, w_i \notin L(M_i)\}$

Tétel: $L_{átló} \notin RE$.

2.3.3 Egy rekurzívan felsorolható, de nem eldönthető nyelv

Az L_u nyelv: Tekintsük azon (M, w) párok halmazát (egy megfelelő bináris szóban elkódolva), ahol M egy $\{0,1\}$ bemenő ábécé feletti Turing-gép, w pedig egy $\{0,1\}$ feletti szó úgy, hogy $w \in L(M)$, azaz M elfogadja w-t. Ezt a nyelvet jelöljük L_u-val. $L_u = \{\langle w_i, w_j \rangle \mid i, j \geq 1, w_j \in L(M_i)\}$

Tétel: $L_u \in RE$.

Tétel: $L_u \notin R$.

2.3.4 További tételek

1. Legyen L egy nyelv. Ha $L, \bar{L} \in RE$, akkor $L \in R$. Következmény: a rekurzívan felsorolható nyelvek nem zártak a komplementerképzésre.

2. Ha $L \in R$, akkor $\bar{L} \in R$, azaz a rekurzív nyelvek zártak a komplementerképzésre.

2.3.5 További eldönthetetlen problémák

Kiszámítható függvény: Legyen Σ és Δ két ábécé és f Σ^*ból Δ^*-ba képző függvény. Azt mondjuk, hogy f kiszámítható, ha van olyan M Turing-gép, hogy M-et egy $w \in \Sigma^*$ szóval a bemenetén elindítva, M úgy áll meg, hogy a szalagján a $f(w) \in \Delta^*$ szó van.

Eldöntési problémák visszavezetése: Legyen $L_1 \subseteq \Sigma^*$ és $L_2 \subseteq \Delta^*$ két eldöntési probléma. L_1 visszavezethető L_2-re ($L_1 \leq L_2$), ha van olyan $f : \Sigma^* \to \Delta^*$ kiszámítható függvény, hogy minden $w \in \Sigma^*$ szóra $w \in L_1$ pontosan akkor teljesül, ha $f(w) \in L_2$ is teljesül.

Tétel: Legyen $L_1 \subseteq \Sigma^*$ és $L_2 \subseteq \Delta^*$ két eldöntési probléma és tegyük fel, hogy L_1 visszavezethető L_2-re. Ekkor igazak a következő állítások:

1. Ha L_1 eldönthetetlen, akkor L_2 is.

2. Ha $L_1 \notin RE$, akkor $L_2 \notin RE$.

A megállási probléma: Legyen $L_h = \{\langle M, w\rangle \mid M \textit{ megáll a } w \textit{ bemeneten}\}$, azaz L_h azon $\langle M, w\rangle$ Turing-gép és bemenet párosokat tartalmazza elkódolva, melyekre M megáll a w bemeneten. L_h eldönthetetlen (L_u visszavezethető L_h-ra), viszont $L_h \in RE$.

Az $L_{üres}$ probléma: Legyen $L_{üres} = \{\langle M\rangle \mid L(M) = \emptyset\}$. $L_{üres}$ eldönthetetlen (L_u visszavezethető $L_{üres}$-re), valamint $L_{üres} \notin RE$.

Rekurzívan felsorolható nyelvek (nem triviális) tulajdonsága: Ha $\mathcal{P}$ a rekurzívan felsorolható nyelvek egy halmaza, akkor $\mathcal{P}$ a rekurzívan felsorolható nyelvek egy tulajdonsága. Ha $\mathcal{P} \neq \emptyset$ és $\mathcal{P} \neq RE$, akkor $\mathcal{P}$ nem triviális tulajdonsága a rekurzívan felsorolható nyelveknek.

Rice tétele: Adott $\mathcal{P}$ tulajdonságra jelöljük $L_{\mathcal{P}}$-vel azon Turing-gépek kódjainak halmazát, amelyek $\mathcal{P}$-beli nyelvet ismernek fel. Ha $\mathcal{P}$ a rekurzívan felsorolható nyelvek egy nem triviális tulajdonsága, akkor $L_{\mathcal{P}}$ eldönthetetlen.

Post Megfelelkezési Probléma (röviden PMP): A PMP problémát a következőképpen definiáljuk. Legyen Σ egy legalább két betűt tartalmazó ábécé és legyen $D = \left\{[\frac{u_1}{v_1}], ..., [\frac{u_n}{v_n}]\right\}$ egy dominóhalmaz, melyben $n \geq 1$ és $u_1, ..., u_n, v_1, ..., v_n \in \Sigma^+$. A kérdés az, hogy van-e egy olyan $1 \leq i_1, ..., i_m \leq m$ ($m \geq 1$) indexsorozat, melyre teljesül, hogy a $[\frac{u_{i_1}}{v_{i_1}}], ..., [\frac{u_{i_m}}{v_{i_m}}]$ dominókat egymás mellé írva alul és felül ugyanaz a szó adódik, azaz $u_{i_1}...u_{i_m} = v_{i_1}...v_{i_m}$. Ebben az esetben a fenti dominósorozatot a D egy megoldásának nevezzük.

Formális nyelvként a következőképpen definiálhatjuk a PMP-t: PMP $= \{\langle D\rangle \mid D - \textit{nek van megoldása}\}$. PMP eldönthetetlen.

2.4 Bonyolultságelmélet

A bonyolultságelmélet célja a megoldható (és ezen belül az eldönthető) problémák osztályozása a megoldáshoz szükséges erőforrások (jellemzően az idő és a tár) mennyisége szerint.

2.4.1 Időbonyolultsági fogalmak

TIME: Legyen $f : \mathbb{N} \to \mathbb{N}$ függvény. $\mathbf{TIME}(f(n)) = \{L \mid L \textit{ eldönthetõ } \mathcal{O}(f(n)) \textit{ idõigényû Turing} - \textit{géppel}\}$

$\mathbf{P} = \bigcup_{k\geq 1} \mathbf{TIME}(n^k)$. Tehát **P** azon nyelveket tartalmazza, melyek eldönthetőek polinom időkorlátos determinisztikus Turing-géppel. Ilyen például a jól ismert ELÉRHETŐSÉG probléma, melynek bemenete egy G gráf és annak két kitüntetett csúcsa (s és t). A kérdés az, hogy van-e a G-ben út s-ből t-be. Ha

az ELÉRHETŐSÉG problémára nyelvként tekintünk, akkor írhatjuk azt, hogy

$$\text{ELÉRHETŐSÉG} = \{\langle G, s, t\rangle \mid G - ben\ van\ út\ s - bõl\ t - be\}.$$

Könnyen megadható az ELÉRHETŐSÉG problémáját polinom időben eldöntő determinisztikus Turing-gép, tehát ELÉRHETŐSÉG $\in \mathbf{P}$.

NTIME: Legyen $f : \mathbb{N} \to \mathbb{N}$ függvény.
$\mathbf{NTIME}(f(n)) = \{L \mid L\ eldönthetõ\ \mathcal{O}(f(n))\ idõigényû\ nemdeterminisztikus\ Turing - géppel\}$

$\mathbf{NP} = \bigcup_{k\geq 1} \mathbf{NTIME}(n^k)$. Az **NP**-beli problémák rendelkeznek egy közös tulajdonsággal az alábbi értelemben. Ha tekintjük egy **NP**-beli probléma egy példányát és egy lehetséges "bizonyítékot" arra nézve, hogy ez a példány "igen" példánya az adott problémának, akkor ezen bizonyíték helyességének leellenőrzése polinom időben elvégezhető. Ennek megfelelően egy **NP**-beli problémát eldöntő nemdeterminisztikus Turing-gép általában úgy működik, hogy "megsejti" a probléma bemenetének egy lehetséges megoldását, és polinom időben leellenőrzi, hogy a megoldás helyes-e.

Tekintsük a SAT problémát, amit a következőképpen definiálunk. Adott egy ϕ ítéletlogikai KNF. A kérdés az, hogy kielégíthető-e. Annak a bizonyítéka, hogy a ϕ kielégíthető, egy olyan változó-hozzárendelés, ami mellett kiértékelve a ϕ-t igaz értéket kapunk. Egy tetszőleges változó-hozzárendelés tehát a ϕ kielégíthetőségének egy lehetséges bizonyítéka .Annak leellenőrzése pedig, hogy ez a hozzárendelés tényleg igazzá teszi-e ϕ-t, polinom időben elvégezhető. A SAT **NP**-beli probléma.

Az a definíciókból következik, hogy fennáll a $\mathbf{P} \subseteq \mathbf{NP}$ tartalmazás.

2.4.2 NP-teljes problémák

Polinom időben kiszámítható függvény: Legyen Σ és Δ két ábécé és f Σ^*ból Δ^*-ba képző függvény. Azt mondjuk, hogy f polinom időben kiszámítható, ha kiszámítható egy polinom időigényű Turing-géppel.

Eldöntési problémák polinom idejű visszavezetése: Legyen $L_1 \subseteq \Sigma^*$ és $L_2 \subseteq \Delta^*$ két eldöntési probléma. L_1 polinom időben visszavezethető L_2-re ($L_1 \leq_p L_2$), ha $L_1 \leq L_2$ és a visszavezetésben használt f függvény polinom időben kiszámítható.

Tétel: Legyen L_1 és L_2 két probléma úgy, hogy $L_1 \leq_p L_2$. Ha L_2

1. **P**-beli, akkor L_1 is **P**-beli.
2. **NP**-beli, akkor L_1 is **NP**-beli.

NP-teljes probléma: Legyen L egy probléma. Azt mondjuk, hogy L **NP**-teljes, ha

1. **NP**-beli, és
2. minden további **NP**-beli probléma polinom időben visszavezethető L-re.

Tétel: Legyen L egy **NP**-teljes probléma. Ha $L \in \mathbf{P}$, akkor $\mathbf{P} = \mathbf{NP}$.

Megjegyzés: Jelenleg **NEM** tudunk **P**-beli **NP**-teljes problémáról!!!

Tétel: Legyen L_1 egy **NP**-teljes, L_2 pedig **NP**-beli probléma. Ha $L_1 \leq_p L_2$, akkor L_2 is **NP**-teljes.

Cooke tétele: SAT **NP**-teljes.

Legyen $k \geq 1$. kSAT $= \{\langle\phi\rangle \mid \phi\ minden\ tagjában\ k\ literál\ van.\}$

Tétel: 3SAT **NP**-teljes, ugyanis SAT $\leq_p$ 3SAT.

TELJES RÉSZGRÁF $= \{\langle G, k\rangle \mid G\ véges\ gráf, k \geq 1,\ G - nek\ \exists\ k\ csúcsú\ részgráfja\}$. Tehát a TELJES RÉSZGRÁF azon G és k párokat tartalmazza, megfelelő ábécé feletti szavakban elkódolva, melyekre igaz,

hogy G-ben van k csúcsú teljes részgráf, azaz olyan részgráf, melyben bármely két csúcs között van él.

TELJES RÉSZGRÁF= $\{\langle G,k\rangle \mid G\ véges\ gráf, k \geq 1,\ G-nek\ \exists\ k\ csúcsú\ részgráfja\}$. Tehát a TELJES RÉSZGRÁF azon G és k párokat tartalmazza, megfelelő ábécé feletti szavakban elkódolva, melyekre igaz, hogy G-ben van k csúcsú teljes részgráf, azaz olyan részgráf, melyben bármely két csúcs között van él.

FÜGGETLEN CSÚCSHALMAZ = $\{\langle G,k\rangle \mid G\ véges\ gráf, k \geq 1,\ G-nek\ \exists\ k\ elemű\ független\ csúcshalmaza\}$. Vagyis a FÜGGETLEN CSÚCSHALMAZ azon G és k párokat tartalmazza, melyekre igaz, hogy G-ben van k olyan csúcs, melyek közül egyik sincs összekötve a másikkal.

CSÚCSLEFEDÉS =
$$\left\{\langle G,k\rangle \mid \begin{array}{l} G\ véges\ gráf, k \geq 1,\ G-nek\ van\ olyan\ k\ elemű\ csúcshalmaza, \\ mely\ tartalmazza\ G\ minden\ élének\ legalább\ 1\ végpontját. \end{array}\right\}.$$

TELJES RÉSZGRÁF, FÜGGETLEN CSÚCSHALMAZ és CSÚCSLEFEDÉS **NP**-teljesek (TELJES RÉSZGRÁF $\leq_p$ FÜGGETLEN CSÚCSHALMAZ $\leq_p$ CSÚCSLEFEDÉS).

UTAZÓÜGYNÖK =
$$\left\{\langle G,k\rangle \mid \begin{array}{l} G\ véges\ irányítatlan\ gráf,\ az\ éleken\ egy-egy\ pozitív\ egész\ súllyal\ és \\ van\ G-ben\ legfeljebb\ k\ összsúlyú\ Hamilton\ kör \end{array}\right\}.$$

Tétel: Az UTAZÓÜGYNÖK probléma **NP**-teljes.

2.4.3 Tárbonyolultság

A tárbonyolultságot egy speciális, úgynevezett offline Turing-gépen vizsgáljuk.

Off-line Turing-gép: Offline Turing-gépnek nevezünk egy olyan többszalagos Turing-gépet, mely a bemenetet tartalmazó szalagot csak olvashatja, a többi, ún. munkaszalagokra pedig írhat is. Az offline Turing-gép tárigényébe csak a munkaszalagokon felhasznált terület számít be.

A továbbiakban Turing-gép alatt minidig offline Turing-gépet értünk. Most definiáljuk a tárbonyolultsággal kapcsolatos nyelvosztályokat.

SPACE(f(n)) = $\{L|L\ eldönthetõ\ \mathcal{O}(f(n))\ tárigényû\ determinisztikus\ Turing-géppel\}$

NSPACE(f(n)) = $\{L|L\ eldönthetõ\ \mathcal{O}(f(n))\ tárigényû\ nemdeterminisztikus\ Turing-géppel\}$

PSPACE = $\bigcup_{k>0}$ **SPACE**(n^k)

NPSPACE = $\bigcup_{k>0}$ **NSPACE**(n^k)

L = **SPACE**$(\log_2 n)$

NL = **NSPACE**$(\log_2 n)$

Savitch tétele: Ha $f(n) \geq \log n$, akkor **NSPACE**$(f(n)) \subseteq$ **SPACE**$(f^2(n))$.