Univerzitet u Beogradu
Matematički fakultet

Logike Fiksne Tačke

Master rad

Mentor:
Dr. Nebojša Ikodinović

Kandidat:
Emir Sadiković

Beograd,
2017.

Sadržaj

Predgovor

U ovom tekstu se bavimo proširenjem Logike Prvog Reda operatorima kojima se definiše fiksna tačka preslikavanja zadatog formulom sa slobodnom relacionom promenljivom. Ova logika, pod imenom *Logika fiksne tačke* je posebno pogodna za logičko opisivanje procedura koje su suštinski rekurzivnog karaktera. Biće opisane razne vrste logika sa operatorima(najmanje/inflacione/parcijalne) fiksne tačke, a posebna pažnja će biti posvećena njihovoj primeni u teoriji složenosti, gde se njima, u prisustvu linearnog uređenja, mogu okarakterisati važne klase složenosti.

U prvom poglavlju uvodimo osnovne koncepte i rezultate opšte teorije modela ali sa posebnim akcentom na konačnim modelima, i tome kako se teorija konačnih modela razliku od opšte teorije modela. Uvodimo konstrukciju formula Logike Prvog Reda, a zatim kroz primere predstavljamo njenu izražajnu moć ali i njena ograničenja, što nam priža motivaciju za traganje za njenim proširenjima koja bi bila bolje prilagođena potrebama izražavanja osobina konačnih modela, te u tom kontekstu uvodimo i Logike Fiksne Tačke ali i *Logiku Drugog reda*. U ovom poglavlju se takođe uvode i razmatraju osnovni pojmovi iz Teorija *izračunljivosti* i *složenosti*, te tehnika kodiranja jednog modela izračunavanja (Tjuringovih mašina) konačnim modelima, kao i tehnike kodiranja struktura i formula kao stringova koje bi se mogle ponuditi kao ulaz Tjuringovoj mašini.

Druga glava je rezervisana za dokaz dve temeljne teoreme teorije konačnih modela, koje ćemo dokazati koristeći se aparaturom uvedenom u prvoj glavi, i ispitivanje njihovog značaja i posledica. Tu ćemo biti u prilicu da dokažemo da za konačne modele ne važi teorema potpunosti te da upoznamo prve dokaze kompatibilnosti između određenih klasa složenosti i određenih logika.

Treće poglavlje je centralno poglavlje ovog teksta, u njemu se definišu Logike Fiksne Tačke, dokazuju i kroz primere upoznaju njihove osobine te ispituju veze između ovih logika i klasa složenosti. Tu se, takođe, prikazuje još jedna vrsta proširenja Logike Prvog Reda - Logika Tranzitivnog Zatvorenja, kao i jezik DATALOG, te prikazuju njihove veze sa pojedinim klasama složenosti.

Tekst završava otvorenim pitanjem koje se tiče mogućeg postojanja logike za klasu PTIME nad klasom konačnih struktura, što je, kao što ćemo videti, drugačije preformulisana tvrdnja da čuveni problem "P vs. NP" ima negativan odgovor.

1 Pojmovi

1.1 Logika

Definicija 1.1. *Jezik (signatura, vokabular) σ je skup simbola konstanti $(c_1, c_2, c_3...)$, funkcijskih simbola $(f_1, f_2, f_3...)$ i relacijaskih simbola $(R_1, R_2, R_3...)$. Svakom od funkcijskih i relacijskih simbola je pridružena arnost barem 1 ili veća, dok za simbole konstanti uzimamo da su arnosti 0.*

Jezici sami po sebi nemaju inherentno značenje, oni mogu biti interpretirani tek u odgovarajućoj strukturi:

Definicija 1.2. *Neka je σ jezik. σ-struktura je četvorka*

$$\mathfrak{A} = \left\langle A, \{c_i^{\mathfrak{A}}\}, \{R_j^{\mathfrak{A}}\}, \{f_k^{\mathfrak{A}}\} \right\rangle_{i \in I, j \in J, k \in K}$$

gde je:

- *A neprazan skup, domen ili univerzum od $\mathfrak{A}$*
- *$c_i^{\mathfrak{A}}$ elementi iz A*
- *Svaki relacijski simbol arnosti k je interpretiran nekom k-arnom relacijom na A*
- *Svaki funkcijski simbol arnosti k je interpretiran nekom k-arnom funkcijom $f_k^{\mathfrak{A}} : A^k \longrightarrow A$*
- *I, J i K skupovi indeksa.*

Struktura je konačna ako joj je domen konačan skup. Kao primere jezika (vokabulara) možemo priložiti recimo $\sigma_{numbers} = \{0, S, +, \cdot, <\}$ ili $\sigma_{group} = \{e, \circ, ^{-1}\}$ koje bismo na očigledan način interpretirali respektivno u strukturi prirodnih brojeva i u strukturi neke grupe.

Definicija 1.3. *Neka su $\mathfrak{A}$ i $\mathfrak{B}$ dve σ-strukture. Funkciju $f : A \longrightarrow B$ zovemo homomorfizam ako za sve $a_1, a_2, a_3, ..., a_n, \{c_i^{\mathfrak{A}}\}_{i \in I} \in A$ važi:*

- *$h(c_i^{\mathfrak{A}}) = c_i^{\mathfrak{B}}$ za svako $i \in I$;*
- *$h(f^{\mathfrak{A}}(a_1, a_2, ..., a_n)) = f^{\mathfrak{B}}(h(a_1), h(a_2), ..., h(a_n))$;*
- *$R^{\mathfrak{A}}(a_1, a_2, ..., a_n) \Rightarrow R^{\mathfrak{B}}(h(a_1), h(a_2), ..., h(a_n))$;*

za sve konstante c i sve n-arne funkcijeske (f) i relacijske (R) simbole.

Ako je homomorfizam bijekcija izmedju domena A i B a pride je i njegov inverz homomorfizam za njega kažemo da je *izomorfizam*. *Automorfizam* je izomorfizam iz $\mathfrak{A}$ u $\mathfrak{A}$. Ako je h injekcija zovemo ga *utapanje*. Me nećemo praviti razliku između izomorfnih struktura, one će za naše potrebe biti jednake.

Definicija 1.4. *Kažemo da je $\mathfrak{A}$ podstruktura od $\mathfrak{B}$ ako je $A \subseteq B$ i inkluzivno preslikavanje je utapanje iz $\mathfrak{A}$ u $\mathfrak{B}$, u oznaci:*

$$\mathfrak{A} \subseteq \mathfrak{B}$$

U tom slučaju kažemo da je struktura $\mathfrak{B}$ ekstenzija strukture $\mathfrak{A}$.

Ako su $\mathfrak{A}_i$ podstrukture od $\mathfrak{B}$ njihov presek je - ukoliko nije prazan skup - takođe podstruktura od $\mathfrak{B}$. Odavde sledi da ako je $S \subset B$ tada postoji najmanja podstruktura $\mathfrak{A}$ od $\mathfrak{B}$ koja sadrži S i za nju kažemo da je generisana skupom S.

Mi ćemo na dalje pretpostavljati da je svaki jezik najviše prebrojiv. Kada govorimo o konačnim strukturama podrazumevaćemo da su jezici konačni i relacioni (sadrže samo simbole konstanti i relacijske simbole). Oznakom STRUCT[σ] ćemo označavati skup svih konačnih σ-struktura.

Logika Prvog Reda

Pristupimo sada definisanju formula logike prvog reda (na koju ćemo u daljem tekstu ponekad referisati oznakom FO).

Definicija 1.5. *Neka je zadat jezik σ. σ-term je reč (niz simbola) sastavljena od simbola (konstanti, funcijskih i relacijskih) jezika σ i promenljivih $v_1, v_2, v_3...$ (za koje pretpostavljamo da ih ima prebrojivo mnogo) koristeći sledeća pravila:*

- *Svaka promenljiva i svaki simbol konsante je term;*
- *Ako je f n-arni funkcijski simbol a $t_1, t_2, ..., t_n$ termi tada je i $f(t_1, t_2, ..., t_n)$ takodje term.*

Definišimo sada σ-formule. To su reči izgrađene pomoću simbola jezika σ, zagrada "("i ")"kao i logičkih simbola:
promenljivih $v_1, v_2, v_3...$
simbol jednakosti $=$
simbol negacije $\neg$
simbol konjunkcije $\wedge$
egzistencijalni kvantifikator $\exists$

Definicija 1.6. *Formule se grade na sledeći način:*

- *Ako su t_1, t_2 termi, $t_1 = t_2$ je formula;*
- *Ako je R relacijaski simbol arnosti n, a $t_1, t_2, ..., t_n$ termi tada je $R(t_1, t_2, ..., t_n)$ formula;*
- *Ako je φ formula, tada je i $\neg\varphi$ formula;*
- $(\varphi_1 \wedge \varphi_2)$ *gde su φ_1 i φ_2 formule;*
- $\exists x\ \varphi$, *gde je φ formula a x promenljiva.*

Formule oblika $R(t_1, t_2, ..., t_n)$ i $t_1 = t_2$ zovemo *atomskim* formulama.

Naravno, koristićemo osim navedenih i simbole $\vee, \rightarrow, \leftrightarrow, \forall$ koje definišemo u terminima ranije navedenih simbola:

- $(\varphi_1 \vee \varphi_2) = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$
- $(\varphi_1 \rightarrow \varphi_2) = \neg(\varphi_1 \wedge \neg\varphi_2)$
- $(\varphi_1 \leftrightarrow \varphi_2) = ((\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1))$
- $\forall x\varphi = \neg\exists x\neg\varphi$

Za promenljivu x kažemo da se u formuli φ javlja slobodno ako nije pod uticajem egzistencijalnog kvantifikatora, rečju:

Definicija 1.7. *Promenljiva x koja se javlja u formuli:*

- *$t_1 = t_2$ je slobodna akko x se pojavljuje u t_1 ili t_2;*
- *$R(t_1, t_2, ..., t_n)$ je slobodna akko x se pojavljuje u nekom od t_i;*
- *$\neg\varphi$ je slobodna akko je x slobodna u φ;*
- *$\varphi_1 \wedge \varphi_2$ je slobodna akko x slobodna u φ_1 ili u φ_2;*
- *$\exists y\varphi$ je slobodna akko $x \neq y$ i x je slobodna u φ.*

Ako pojavljivanje promenljive u nekoj formuli nije slobodno kažemo da je ono vezano. Ako je $\vec{x} = (x_1, x_2, ..., x_n)$ vektor slobodnih promenljivih formule φ tada nju označavamo $\varphi(\vec{x})$.

Definišimo sada šta znači da u nekoj σ-strukturi $\mathfrak{A}$ važi formula $\varphi(x_1, x_2, ..., x_n)$ za neko $\vec{b}$. I ova definicija će biti rekurzivna, ali pre toga moramo definisati vrednost terma:

Definicija 1.8. *Za σ-term t, σ-strukturu $\mathfrak{A}$ i vektor $\vec{b} = (b_1, b_2, ...b_n)$ koji je dodeljen promenljivim $v_1, ..., v_n$ definišemo interpretaciju ili vrednost terma $t^{\mathfrak{A}}(\vec{b})$:*

- *Ako je t promenljiva, $v_i^{\mathfrak{A}}(\vec{b}) = b_i$;*
- *Ako je t simbol konstante, $c^{\mathfrak{A}}(\vec{b}) = c^{\mathfrak{A}}$;*
- *Ako je t funkcijski simbol, $f(t_1, ..., t_n)^{\mathfrak{A}}(\vec{b}) = f^{\mathfrak{A}}(t_1^{\mathfrak{A}}(\vec{b}), ..., t_n^{\mathfrak{A}}(\vec{b}))$.*

Definicija 1.9. *Neka je data σ-struktura $\mathfrak{A}$. Za σ-formulu φ i svaki vektor slobodnih promenljivih $\vec{b}$ definišemo relaciju:*

$$\mathfrak{A} \models \varphi(\vec{b})$$

na sledeći način:

- *$\mathfrak{A} \models t_1 = t_2(\vec{b})$ akko $t_1^{\mathfrak{A}}(\vec{b}) = t_2^{\mathfrak{A}}(\vec{b})$;*

- $\mathfrak{A} \models R(t_1, t_2, ..., t_n)(\vec{b})$ *akko* $R^{\mathfrak{A}}(t_1^{\mathfrak{A}}(\vec{b}), ..., t_n^{\mathfrak{A}}(\vec{b}))$*;*
- $\mathfrak{A} \models \neg\varphi(\vec{b})$ *akko* $\mathfrak{A} \not\models \varphi(\vec{b})$*;*
- $\mathfrak{A} \models (\varphi_1 \wedge \varphi_2)(\vec{b})$ *akko* $\mathfrak{A} \models \varphi_1(\vec{b})$ *i* $\mathfrak{A} \models \varphi_2(\vec{b})$*;*
- $\mathfrak{A} \models \exists x \varphi(\vec{b})$ *akko postoji* a *iz* A *takvo da* $\mathfrak{A} \models \varphi(\vec{b}\frac{a}{x})$*,*

  gde

$$\vec{b}\frac{a}{x} = (b_0, ..., b_{i-1}, a, b_{i+1}, ...) \qquad ako\ je\ x = v_i$$

Ako za objekte $\mathfrak{A}$*,* φ *i* $\vec{b}$ *važi* $\mathfrak{A} \models \varphi(\vec{b})$ *kažemo da* φ *važi u* $\mathfrak{A}$ *za* $\vec{b}$ *ili da* $\vec{b}$ *zadovoljava* φ *u* $\mathfrak{A}$*.*

Definicija 1.10. *Formula bez slobodnih promenljivih se zove rečenica.*

Ako rečenica φ *važi u* $\mathfrak{A}$ *kažemo da je* $\mathfrak{A}$ *model za* φ*. Ako rečenica ima model kažemo da je zadovoljiva a valjana je ako je tačna u svakoj strukturi. Ako je* Σ *skup rečenica tada važi da je* $\mathfrak{A}$ *model za* Σ *ako je ona model za svaku rečenici iz* Σ*, u oznaci:*

$$\mathfrak{A} \models \Sigma$$

σ*-teorija* T *je skup rečenica jezika* σ*. Ako teorija ima model kažemo da je zadovoljiva.*

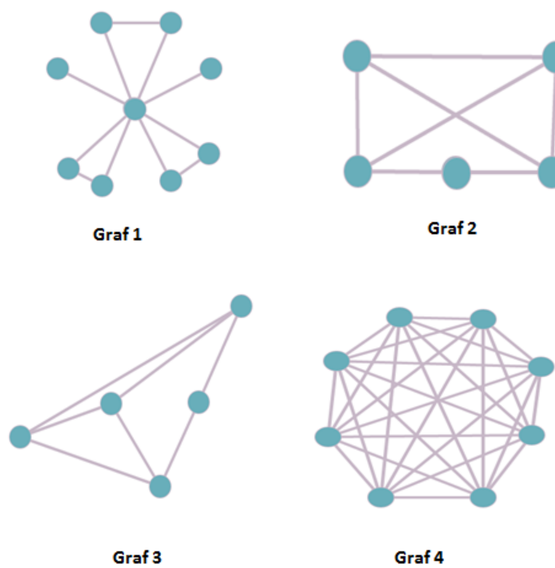
Ako su σ i σ' disjunktni jezici a $\mathfrak{A}$ i $\mathfrak{A}'$ redom σ i σ' strukture nad istim domenom, tada $\sigma \cup \sigma'$ predstavlja novi jezik a $(\mathfrak{A} \cup \mathfrak{A}')$ jednu strukturu u tom jeziku gde je svaki simbol u zavisnosti od toga iz kojeg je od ta dva jezika interpretiran kao u strukturi $\mathfrak{A}$ (ako je iz jezika σ) ili u strukturi $\mathfrak{A}'$ (ako je iz jezika σ').

Jedan specijalan slučaj ove vrste proširenja jezika σ je dodavanje samo još dodatnih simbola konstanti u taj jezik. Ovaj tip proširenja gde je u jezik dodato još n konstanti $(c_1, c_2, ...c_n)$ koje se ne nalaze u σ ćemo obeležavati σ_n. Neka su jeziku σ dodate konstante $c_1, c_2, ..., c_n$ i neka je ϕ rečenica jezika σ_n koja je dobijena iz formule $\phi(x_1, x_2, ..., x_n)$ zamenom svakog x_i sa c_i i neka $(a_1, a_2, ..., a_n) \in A^n$. Tada važi:

$$\mathfrak{A} \models \varphi(a_1, a_2, ...a_n) \text{ akko } (\mathfrak{A}, a_1, ..., a_n) \models \phi$$

Osim logike prvog reda čije smo formiranje upoznali u prethodnom razmatranju, postoje i druge logike, sa složenijom strukturom. Kao što smo videli, u formulama Logike prvog reda možemo samo da kvantifikujemo i da za promenljive uzimamo samo elemente domena određene strukture dok ne mozemo da imamo promenljive koje vrednosti uzimaju iz podskupova domena (odnosno iz relacija na domenu strukture). Za tu potrebu ćemo kasnije u tekstu uvesti *Logiku drugog reda*, ali i brojna druga proširenja FO-logike, koja će nam posluziti za svrhe koje prevazilaze mogućnosti FO-logike.

Primer 1. (Grafovi) Jedan od najvažnijih primera konačne strukture je konačan skup sa jednom binarnom relacijom - odnosno graf. Naime, graf možemo predstaviti kao konačan skup elemenata (čvorova) a binarna relacija E se može interpretirati kao ivica, tako da za elemente a i b važi $E(a, b)$ akko postoji ivica od a do b. Grafički prikaz nekoliko takvih struktura je dat na narednoj slici.



Dakle, jezik grafova je $\sigma = \{E\}$ i svaki od ovih grafova je jedna σ-struktura. Npr. $graf$ 2 je struktura:

$$\mathfrak{G} =< G, E >,$$

gde je $G = \{v_1, v_2, v_3, v_4, v_5\}$ a $E = \{(v_1, v_2), (v_1, v_5), (v_1, v_4), (v_2, v_3), (v_2, v_5), (v_3, v_4), (v_4, v_5), (v_2, v_1), (v_5, v_1), (v_4, v_1), (v_3, v_2), (v_5, v_2), (v_4, v_3), (v_5, v_4)\}$. Primetimo da je relacija E simetrična, što odražava činjenicu da je $graf$ 2 neusmeren.

Primetimo da je $\mathfrak{G}$ takođe i struktura $grafa$ 3, to jest ta dva grafa su za naše potrebe jednaka.

Stazom u grfu $\mathfrak{G}$ između čvorova a i b zovemo niz čvorova $a, v_1, v_2, ...v_k, b$ takav da važi $E(a, v_1), E(v_1, v_2), ...E(v_k, b)$. Kažemo da je graf povezan ako postoji staza izmedju svaka dva čvora u grafu.

Jasno je da je svaka od gornjih struktura model za rečenice $\forall x \neg E(x, x)$, $\forall x \forall y (E(x, y) \leftrightarrow E(y, x))$ i $\forall x \exists y E(x, y)$, koje kažu, redom, da ni jedan čvor nije u relaciji sa samim sobom, da je relacija simetrična i da ni jedan element nije izdvojen od svih drugih elemenata. Dok ni jedan nije model rečenice $\exists x \forall y E(x, y)$. Ova rečenica kaže da u grafu postoji čvor koji je incidentan sa svim čvorovima u grafu. U grafu broj 1 za takvo nešto nam fali samo još da čvor u sredini bude povezan sa samim sobom, a u $grafu$ 4 to važi za svaki od čvorova. Ovu osobinu takođe možemo zapisati pomoću FO-rečenice: $\exists x \forall y(\neg(x = y) \rightarrow E(x, y))$.

Razlika između $grafa$ 1 i $grafa$ 4 je u tome što $graf$ 1 sadrži samo jedan čvor koji je povezan sa svim ostalim (osim samog sebe), dok u $grafu$ 4 tu osobinu imaju svi čvorovi. Ove dve osobine takođe možemo izraziti FO-logikom. Neka je $\varphi(x) = \forall y(\neg(y = x) \rightarrow E(x, y))$. Tada je $graf$ 1 model rečenice:

$$\exists y \varphi(y) \wedge \forall z(\varphi(z) \rightarrow (z = y)),$$

dok $graf$ 4 nije (kao ni preostala dva grafa). $Graf$ 4 jeste model za sledeću rečenicu:

$$\forall x \forall y(\neg(x = y) \rightarrow E(x, y))$$

Svaki graf koji je model za ovu rečenicu se zove *klika*(*clique*), u našem slučaju 8-klika, K_8.

Definicija 1.11. *Grafove (G_1, E_1) i (G_2, E_2) zovemo izomorfnim ako postoji bijekcija f između čvorova ta dva grafa takva da za svaka dva čvora x i y važi $E_1(x, y)$ akko $E_2(f(x), f(y))$.*

Mi smo u prethodnom delu uspeli da FO-rečenicom razdvojimo $graf$ 1 od ostalih grafova, ali postavlja se pitanje da li postoji FO-rečenica koja će tačno opisati $graf$ 1 (odnosno sve grafove njemu izomorfne), odnosno čiji će model biti taj i samo taj graf? Sledeća lema nam daje pozitivan odgovora na ovo pitanje:

Lema 1.1. *Neka je σ alfabet samo sa jednom binarnom relacijom E. Za svaku konačnu σ-strukturu $\mathfrak{A}$ postoji σ-rečenica prvog reda, $\varphi_{\mathfrak{A}}$ koja je zadovoljiva u $\mathfrak{B}$ akko su $\mathfrak{A}$ i $\mathfrak{B}$ izomorfne strukture.*

Dokaz. a) Neka je

$$\varphi_1(\vec{x}) = \bigwedge_{i \neq j} \neg(x_i = x_j)$$

Ova formula, φ_1, će obezbediti da $\mathfrak{A}$ ima barem n elemenata.

b) Neka je

$$\varphi_2(\vec{x}) = \forall y \bigvee_{i} (x_i = y)$$

Ova formula, φ_2, će obezbediti da $\mathfrak{A}$ nema više od n elemenata.

c)

$$\varphi_3(\vec{x}) = \bigwedge_{(a_i, a_j) \in E} E(x_i, x_j)$$

d)

$$\varphi_4(\vec{x}) = \bigwedge_{(a_i, a_j) \notin E} \neg E(x_i, x_j)$$

Sada rečenica $\exists x_1 \exists x_2 \exists x_3 ... \exists x_n(\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4)$ zadovoljava $\mathfrak{A}$ i samo njoj izomorfne strukture. △

U ovom smislu, FO logika je dovoljno snažna za konačne modele, jer se oni njom mogu do kraja opisati. S druge strane, neke od jednostavnih osobina grafova, povezanost, tranzitivno zatvorenje, da li ima paran broj čvorova itd - se ne mogu izraziti jezikom prvog reda, odnosno ne može se naći rečenica prvog reda koja je zadovoljiva samo, recimo, u povezanim grafovima. I inače jedan od centralnih zadataka teorije konačnih modela je istraživanje koje se osobine i, još opštije, koje relacije na strukturi se mogu opisati koristeći se rečenicama određene logike. $\triangle$

Primer 2. (Baza podataka) Drugi važan primer konačne strukture su *relacione baze podataka.* Baza podataka, uopšteno, je bilo koji skup podataka. Relaciona baza podataka je kolekcija tabela, od kojih svaka ima svoje jedinstveno ime, kao i svoje kolone, koje takođe imaju jedinstvena imena (ne moze se desiti da postoje u jednoj tabeli dve kolone sa istim imenom) u kojima se čuvaju slogovi podataka (vrste). Univerzum za rel. bazu podataka predstavlja skup svih elemenata koji se pojavljuju u bilo kojoj od tabela u bilo kojoj koloni. Jezik se sastoji od onoliko relacionih simbola koliko u bazi ima tabela, gde je svaki rel. simbol arnosti koja je jednaka broju kolona koje ima odgovorajuća tabela. Jedna baza, u kojoj se čuvaju podaci o uvozu određene firme bi mogla da izgleda kao što je prikazano na narednoj slici.

Uvoz robe			
vrsta	količina	zemlja porekla	datum
ugalj	1234423	MK	1.1. 2017
celik	12769	MNE	2.1.2017
ugalj	123	BUL	3.1.2017
drvo	5432	MNE	2.1.2017

zemlje	
sifra	zemlja
MK	makedonija
MNE	crna gora
HR	hrvatska
RU	rumunija
BUL	bugarska

Ova baza, kao što vidimo, se sastoji od dve tabele, ”Uvoz robe”i ”zemlje”, te se u tabelu ”Uvoz robe” upisuju i čuvaju podaci u svakoj uvezenoj jedinici bilo kojeg proizvoda, a tabela ”zemlje” je statična i koristi se kao šifrarnik za imena zemalja koja se kao skraćenice upisuju u tabelu "Uvoz robe". Ovu bazu bi predstavljala struktura $\mathfrak{A}$ koja ima 18 elemanata ("ugalj", "čeli", đrvo", "1234423", "12769",...,"1.12017", ..., MNE",...,"bugarska") i na kojoj je zadata jedna binarna relacije (Zemlje - Z) i jedna 4-arne relacije (Uvoz - U) i elementi su u relaciji ako se nalaze u istoj vrsti neke od tabela. Recimo $\mathfrak{A} \models U(''celik'', "12769", "MNE", "2.1.2017")$ itd.

Pomoću rečenica prvog reda možemo formirati i nove relacije. Recimo možemo da izdvojimo vrste proizvoda koji su uvezeni iz Crne Gore, FO-formulom, odnosno "om":

$$\exists x \exists y \exists z U(p, x, y, z) \land Z(y, 'crna\ gora')$$

$\triangle$

Pojam upita, ili globalne relacije, vuče ime upravo iz teorije baza podataka.

Definicija 1.12. *(Upit) n-arni upit na strukturama jezika σ je preslikavanje Q koje svakoj σ-strukturi $\mathfrak{A}$ dodeljuje podskup od A^n i koje je zatvoreno za izomorfizam, odnosno ako je h izomorfizam izmedju A i B onda je $Q(\mathfrak{B}) = h(Q(\mathfrak{A}))$.*

Posebno je važan slučaj kada je $n = 0$. A^0 je skup koji sadrži jedan element pa ima dva podskupa (on sam i prazan skup). Dakle upit Q koji je arnosti 0 slika σ-strukture u skup od dva elementa, kojima možemo nominalno dodeliti vrednosti $True$ i $False$. Ovakvi upiti se zovu Bool-ovski.

Navedimo nekoliko primera upita:

- Tranzitivno zatvorenje (TC) je binarni upit:

$$TC(G) = \{(a, b) \in V^2 \mid \text{postoji putanja od } a \text{ do } b\}$$

- 2-dusjunkne putanje (2DP) je 4-arni upit:

$$2DP(G) = \{(a, b, c, d) \in V^4 \mid \text{postoje disjunktne putanje od } a \text{ do } b \text{ i od } c \text{ do } d\}$$

- Izolovane tačke (IP) je 1-arni upit:

$$IP(G) = \{a \in V \mid a \text{ je izolovani element u grafu } G\}$$

- Planarnost grafa (PL) je 0-arni upit:

$$IP(G) = \begin{cases} 1, & \text{ako je } G \text{ planaran} \\ 0, & \text{ako } G \text{ nije planaran} \end{cases}$$

- Parnost strukture (EVEN) je 0-arni (Bulovski) upit:

$$EVEN(\mathfrak{A}) = \begin{cases} 1, & \text{ako } \mathfrak{A} \text{ ima paran broj elemenata} \\ 0, & \text{ako } \mathfrak{A} \text{ ima neparan broj elemenata} \end{cases}$$

- Povezanost grafa (C) je 0-arni (Bulovski) upit:

$$C(G) = \begin{cases} 1, & \text{ako je } G \text{ povezan} \\ 0, & \text{ako } G \text{ nije povezan} \end{cases}$$

Definicija 1.13. *Neka je $\mathcal{L}$ logika i $\mathcal{C}$ klasa konačnih struktura*

- *k-arni upit Q je definabilan u logici $\mathcal{L}$ ($\mathcal{L}$-definabilan) ako postoji $\mathcal{L}$-formula $\varphi(x_1, ..., x_n)$ u jeziku σ takva da za svaku strukturu $\mathfrak{A} \in \mathcal{C}$ važi:*

$$Q(\mathfrak{A}) = \{(a_1, a_2, ..., a_n) \in A^n \mid \mathfrak{A} \models \varphi(a_1, ..., a_n)\}.$$

- *Bulovski upit Q na $\mathcal{C}$ je $\mathcal{L}$-definabilan ako postoji $\mathcal{L}$-rečenica φ takva da:*

$$Q(\mathfrak{A}) = 1 \quad akko \quad \mathfrak{A} \models \varphi$$

Ako je Q definabilno sa φ pišemo $\varphi(\mathfrak{A})$ umesto $Q(\mathfrak{A})$.

Dakle, upit se može shvatiti kao definisanje i poopstenje pojma osobine. Definišimo još, radi jasnije terminologije u nastavku i pojam osobine neke strukture:

Definicija 1.14. *Osobina končnih struktura $\mathcal{P}$ je bilo koja klasa struktura zatvorena za izomorfizam*

Načelno govoreći teorija konačnih modela se bavi izražajnom moći logika, tj. pitanjima kakve klase struktura su definabilne u datoj logici $\mathcal{L}$, i kako sintaksički uslovi koje postavljamo na φ utiču na semantička ograničenja za $Mod(\varphi) = \{\mathfrak{A} \mid \mathfrak{A} \models \varphi\}$ i td.

Tu je, kao specijalan slučaj definabilnosti interesantno pitanje definabilnosti određene osobine, tj. problem za koje osobine $\mathcal{P}$, za određenu logiku $\mathcal{L}$, postoji rečenica te logike φ takva da važi:

$$\mathfrak{A} \in \mathcal{P} \quad \text{akko} \quad \mathfrak{A} \models \varphi$$

Pogledajmo nekoliko primera definabilnih upita:

- Bulovski upit - da li graf G ima izolovan čvor - je definabilan rečenicom:

$$\exists x \forall y (\neg E(x, y))$$

- Unarni upit "čvor x ima najmanje dva različita suseda" je definabilan rečenicom:

$$\exists y \exists z (\neg(y = z) \wedge E(x, y) \wedge E(x, z))$$

- Pokrivanje čvorova veličine najviše k se može definisati formulom:

$$\exists x_1 \exists x_2 ... \exists x_k (\forall y \forall z (E(x, y) \ \rightarrow \bigvee_{1 \leq i \leq k} y = x_i \ \vee \bigvee_{1 \leq i \leq k} z = x_i))$$

  (ovde x_i i x_j mogu uzeti isti element i za $i \neq j$, te smo stoga rekli da je pokrivanje veličine "najviše" k)

- osobina 3-obojivosti (da se čvorovi grafa mogu obojiti sa tri različite boje tako da svake dva čvora koja su povezana ivicom nisu obojena istom bojom) se može definisati sledećom rečenicom:

$$\begin{aligned}\exists R \subseteq V \exists B \subseteq V \exists G \subseteq V\ \forall x(R(x)\ \vee\ B(x)\ \vee\ G(x)) \wedge \\ \forall x(\neg(R(x) \wedge B(x))\ \wedge\ \neg(R(x) \wedge G(x))\ \wedge\ \neg(B(x) \wedge G(x))) \wedge \\ \forall x \forall y(E(x,y) \rightarrow (\neg(R(x) \wedge R(y))\ \wedge\ \neg(B(x) \wedge B(y)) \wedge \\ \neg(G(x) \wedge G(y))))\end{aligned} \tag{1}$$

  (ovde nam prvi red kaže da je svaki čvor obojen nekom od tri boje, drugi red da je obojen najviše jednom bojom a treći red da su elementi povezani ivicom obojeni različitim bojama)

- Upit o nepovezanosti grafa se može definisati rečenicom:

$$\exists S(\exists x S(x) \wedge \exists y \neg S(y) \wedge \forall z \forall w(S(z) \wedge \neg S(w)\ \rightarrow\ \neg E(z,w)))$$

Obratimo pažnju na činjenicu da smo ovde u prvim primerima kvantifikovali isključivo nad čvorovima grafa odnosno elementima domena strukture te su ti upiti definabilni u logici prvog reda, dok smo u poslednja dva primera kvantifikovali ne samo nad elementima nego i nad skupovima (tačnije podskupovima od domena) i tu već izlazimo izvan dosega logike prvog reda.

Logika Fiksne Tačke

Međutim, mnoge važne upite je moguće definisati u određenim ekstenzijama logike prvog reda. Jedna od najvažnijih takvih konstrukcija je Logika Fiksne Tačke, koju ćemo detaljnije razmotriti u nastavku. Grubo rečeno, reč je o logikama koje proširuju logiku prvog reda operatorima za definisanje fiksnih tačaka relacionih operatora. Pročavanje ovakvih logika vodi poreklo od izučavanja induktivne definabilnosti u kontekstu opšte teorije rekurzija.

Osnovna ideja je sledeća: posmatramo neku fiksiranu relacionu strukturu $\mathfrak{A}$ i n-arni operator $\Phi : P(A^n) \rightarrow P(A^n)$, i želimo da okarakterišemo n-arne relacije nad $\mathfrak{A}$ koje induktivno definiše Φ: Na primer, $R_0 = \emptyset$, $R_{n+1} = \Phi(R_n)$, ili $Q_0 = \emptyset$, $Q_{n+1} = Q_n \cup \Phi(Q_n)$ i sl. Ključni korak u tom cilju jeste na sintaksnom nivou omogućiti definiciju fiksne tačke operatora, tj. n-arnu relaciju $R \subseteq A^n$ takvu da je $\Phi(R) = R$.

Mnogi važni relacioni operatori mogu se definisati formulama koje se grade od sintaksnog materijala logike prvog reda proširenog relacionim promenljivama: za svaki $n \geq 1$ imamo na raspolaganju n-arne relacione promenljive X_1^n, X_2^n, ... Preciznije, za zadati jezik σ, pomenute formule gradimo koristeći sva formacijska pravila Definicije 1.6, uz dodatak:

- Ako je X neka n-arna promenljiva i $t_1, \ldots, t_n$ su termi, onda je $X(t_1, \ldots, t_n)$ formula.

Primetimo da (za sada) nije dozvoljena kvantifikacija po relacionim promenljivama. Ako je φ neka ovako izgradjena formula u kojoj se pojavljuju samo relacione promenljive $X_1, \ldots, X_n$, a (obične) slobodne promenljive su neke od $x_1, \ldots, x_n$, onda pišemo $\varphi(X_1, \ldots, X_n, x_1, \ldots, x_n)$ ili kraće $\varphi(\vec{X}, \vec{x})$.

Formula $\varphi(X, \vec{x})$ jezika σ, sa jednom n-arnom relacionom promenljivom X, i običnim slobodnim promenljivama $\vec{x} = (x_1, \ldots, x_n)$, na svakoj σ-strukturi $\mathfrak{A}$ definiše jedan n-arni operator $\Phi : P(A^n) \to P(A^n)$:

$$\Phi(R) = \{\vec{a} \mid (\mathfrak{A}, R) \models \varphi(\vec{a})\}, \ R \subseteq A^n.$$

Naravno, pri ispitivanju zadovoljenja formule $\varphi(X, \vec{x})$ u $(\mathfrak{A}, R)$ pri valuaciji $\vec{a}$, simbol X posmatramo kao n-arni relacijski simbol čija je interpretacija R. Više o ovome u glavi 3.

Osnovni rezultati logike prvog reda

Nevedimo sada nekoliko centralnih rezultata klasične Teorije modela:

Teorema 1.1. *(Teorema Kompaktnosti) Teorija je zadovoljiva akko je svaki njen konačni podskup zadovoljiv.*

Teorema 1.2. *(Lowenheim-Skolem) Ako T ima beskonačan model onda ona ima prebrojiv model.*

Teorema 1.3. *(Teorema Potpunosti) Neka je T teorija i φ rečenica jezika σ. Ako $T \models \varphi$ onda $T \vdash \varphi$. Oznaka $\models$ ovde znači da φ važi u svim modelima za T a oznaka $\vdash$ da se φ može formalno izvesti iz T.*

Drugim rečima, teorema potpunosti kaže da ako $T \nvdash \varphi$ tada postoji "kontraprimer", odnosno σ-struktura koja jeste model za T ali nije model za φ (odnosno jeste model za $T \cup \neg\varphi$)

Detaljniji prikaz i dokazi ovih tvrđenja se mogu naći u knjizi [9].

Međutim, ispostavlja se da one, iako glavni alat teorije modela, ne važe kada se ograničimo samo na klasu konačnih modela. Dokažimo prvo da za konačne modele ne važi teorema o kompaktnosti:

Teorema 1.4. *Postoji skup T rečenica prvog reda (FO) t.j. teorija prvog reda takva da svaki njen konačan podskup ima konačan model ali T nema konačan model.*

Dokaz. Neka je θ_n rečenica koja kaže da u domenu postoji tačno n elemenata. Definišimo sada $T = \{\neg\theta_n \mid \ n \in \mathbb{N}\}$. Očigledno je da svaki konačan podskup od T ima model međutim T nema konačan model. $\square$

Definicija 1.15. *σ-rečenica φ je konačno zadovoljiva ako postoji konačna σ-struktura $\mathfrak{A}$ takva da $\mathfrak{A} \models \varphi$ a konačno valjana ako je tačna u svakoj konačnoj σ strukturi.*

Jedan od ključnih rezultata ove oblasti, koji ćemo dokazati i detaljnije proanalizirati u poglavlju 2, a iz koje sledi da za konačne modele ne važi teorema potpunosti je:

Teorema 1.5. *(Trakhtenbrot-ova teorema) Konačna zadovoljivost nije odlučiva u logici prvog reda, odnosno za svaki relacioni jezik σ sa barem jednim binarnim relacijskim simbolom važi da skup:*

$$FSAT[\sigma] = \{\varphi \mid \ \varphi \text{ konačno zadovoljiva rečenica prvog reda jezika } \sigma\}$$

nije odlučiv.

Logika Drugog Reda

Definicija 1.16. *(Logika drugog reda (SO)). Logika drugog reda je proširenje FO-logike, gde za razliku od nje postoje promenljive ("promenljive drugog reda") koje vrednosti ne uzimaju iz domena nego iz podskupova i relacija na domenu.*

Neka je dat relacioni jezik σ. Terme i formule SO logike definišemo na sledeći način:

- *Svaki simbol konstante i svaka promenljiva prvog reda (FO-promenljiva) su termi prvog reda.*
- *Atomske formule su:*

  1) atomske formule prvog reda ($t = t_1$ i $R(\vec{t})$ gde su t odgovarajući termi)

  2) $X(t_1, t_2, ..., t_n)$, gde je X SO-promenljiva arnosti n a $t_1, ..., t_n$ termi

- *SO-formule su zatvorene za operacije $\wedge, \vee, \neg$ i kvantifikaciju prvog reda.*
- *Ako je $\varphi(\vec{x}, Y, \vec{X})$ formula onda su $\exists Y \ \varphi(\vec{x}, Y, \vec{X})$ i $\forall Y \ \varphi(\vec{x}, Y, \vec{X})$ takodje formule čije su slobodne promenljive $\vec{x}$ i $\vec{X}$, redom prvog i drugog reda. I inače, formula $\varphi(\vec{x}, \vec{X})$ će nam označavati formulu sa FO-promenljivama $\vec{x}$ i SO-promenljivama $\vec{X}$*

Neka je $\mathfrak{A}$ konačna struktura. Za svaku formulu $\varphi(\vec{x}, \vec{X})$ definisaćemo šta znači $\mathfrak{A} \models \varphi(\vec{b}, \vec{B})$, gde je $\vec{b}$ vektor elemenata iz A iste dužine kao $\vec{x}$, a $\vec{B} = (B_1, ..., B_k)$ uređena k-torka skupova takva da je svaki B_i podskup od A^{n_i}, koje odgvara arnosti od od X_i iz vektora $\vec{X} = (X_1, X_2, ...X_k)$. Opisaćemo samo semantiku konstrukcija koje se razlikuju od onih koje smo opisali za FO-strukture:

- *Ako je $\varphi(\vec{x}, \vec{X})$ oblika $X_i(t_1, t_2, ..., t_k)$ onda $\mathfrak{A} \models \varphi(\vec{b}, \vec{B})$ akko je vektor $(t_1^{\mathfrak{A}}(\vec{b}), ..., t_k^{\mathfrak{A}}(\vec{b})) \in B_i$.*
- *Ako je $\varphi(\vec{x}, \vec{X})$ oblika $\exists Y \ \psi(\vec{x}, Y, \vec{X})$, gde je Y m-arno, tada $\mathfrak{A} \models \varphi(\vec{b}, \vec{B})$ akko postoji neko $C \subseteq A^m$ takvo da važi $\mathfrak{A} \models \varphi(\vec{b}, C, \vec{B})$*
- *Ako je $\varphi(\vec{x}, \vec{X})$ oblika $\forall Y \ \psi(\vec{x}, Y, \vec{X})$, gde je Y m-arno, tada $\mathfrak{A} \models \varphi(\vec{b}, \vec{B})$ akko za svako $C \subseteq A^m$ važi $\mathfrak{A} \models \varphi(\vec{b}, C, \vec{B})$*

Definicija 1.17. *Monadska logika drugog reda, MSO, je restrikcija logike drugog reda gde su sve promeljive drugog reda arnosti 1.*

Definicija 1.18. *Egzistencijalna logika drugog reda, $\exists SO$, je restrikcija logike drugog reda na formule oblika:*

$$\exists X_1 ... \exists X_n \varphi,$$

gde φ ne sadrži kvantifikatore drugog reda (dakle φ je FO-formula). Univerzalna logika drugog reda, $\forall SO$ je restrikcija logike drugog reda na formule oblika:

$$\forall X_1 ... \forall X_n \varphi,$$

gde φ ne sadrži kvantifikatore drugog reda.

Jedan primer SO-rečenice smo videli kod 3-obojivisti, osobine koju smo mogli da prepoznamo SO-rečenicom. Još jedan primer je rečenica koja iskazuje "princip bivalencije": $\forall P \forall x (x \in P \lor x \notin P)$, drugim rečima da za svaku unarnu relaciju važi da se svako x ili nalazi ili ne nalazi u njoj.

Logika drugog reda je značajno ekspresivnija od FO-logike. Recimo, njom se može definisati i povezanost grafa, rečenicom:

$$\forall P ((\exists x P(x) \land \forall x \forall y ((P(x) \land E(x,y)) \ \rightarrow \ P(y))) \ \rightarrow \ \forall z P(z))$$

Ova rečenica kaže da za svako svojstvo čvorova koje ima osobinu da čim važi za neki element važi za sve koje su povezane sa njim, da onda to svojstvo imaju svi elementi. Jasno je da ovu rečenicu zadovoljavaju samo povezani grafovi.

SO-logika (tj. njene određene restrikcije) se ispostavlja vrlo korisno za "hvatanje" određenih klasa složenosti, kako ćemo videti u nastavku teksta.

1.2 Teorija izračunljivosti

Neka je Σ konačan skup simbola (iliti alfabet). Skup svih konačnih nizova (stringova) elemenata iz ovog alfabeta obeležavamo Σ^*. Podskupove od Σ^* zovemo jezicima.

Jedan od najpoznatijih modela računanja je Tjuringova mašina. Neformalno govoreći, radi se o apstraktnoj mašini koja mehanički operiše simbolima na beskonačnoj (obično samo sa jedne, najčešće desne strane) traci. Tačnije rečeno, TM se sastoji od: A) beskonačne trake izdeljene na polja od kojih je u svako upisan neki simbola unapred specifikovanog alfabeta (može biti i blanko); B) glave koja može da čita simbole na traci (jedan simbol u trenutku), briše i upisuje na traku nove simbole i pomera se za po jedno mesto ulevo ili udesno na traci; C) registra stanja, koji može da čuva stanje u kojem se mašina nalazi; D) konačne tabele sa instrukcijama iz koje je moguće pročitati šta glava treba da uradi ako se nalazi u stanju q_i i u polju nad kojim se trenutno nalazi čita simbol a_i - da li da u polje upiše neki novi simbol ili ostavi stari, da li se pomeri levo, desno ili ostane na mestu i u koje stanje treba da se prebaci. Formalno:

Definicija 1.19. *Tjuringova mašina TM je uredjena sedmorka* $TM = (Q, \Sigma, \Gamma, \delta, q_0, Q_{accept}, Q_{reject})$, *gde je Q skup stanja, Σ ulazni alfabet*

(alfabet u kojem se ulazne reči zapisuju) Γ *skup svih znakova koji mogu biti upisani u polja Tjuringove mašine,* δ *tranziciona funkcija,* $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$*,* q_0 *inicijalno stanje dok su* Q_{accept} *i* Q_{reject} *podskupovi od* Q *koji predstavljaju skup prihvatajućih i skup odbijajućih stanja u koja kad dodje mašina prekida sa radom, te znamo da li prihvata ili odbija ulaz u zavisnosti od toga u kojem od dva skupa je završila.*

TM je deterministička ako za svako q *i* a *važi* $|\delta(q,a)| = 1$*, u suprotnom, kada je ova vrednost veća, TM je nedeterministička.*

Konfiguracija Tjuringove mašine predstavlja kompletan sadržaj koji se nalazi na traci u tom trenutku, stanje u kome se mašina nalazi i poziciju na kojoj je nalazi glava. Računanje Tjuringove mašine predstavlja niz konfiguracija $C_0, C_1, C_2, C_3...$ koji može biti konačan, u kojem slučaju je stanje poslednje konfiguracije jedno do stanja iz skupova Q_{accept} ili Q_{reject}, ali i beskonačan. Ako računanje krene iz stanja q_0 sa zapisanom rečju x na traci i završi u nekom od stanja Q_{reject} kažemo da ta TM (i to izračunavanje) odbija x, a ako završi u nekom od stanja iz Q_{accept} da mašina prihvata x. Mašine možemo podeliti na one koje se zaustvaljaju na svim ulazima i one koje za neki ulaz imaju i beskonačno izračunavanje.

Definicija 1.20. *TM M prihvata jezik:*

$$L(M) := \{w \in \Sigma^* \mid M \textit{ prihvata } w\}$$

Jezik L je rekurzibno nabrojiv ako postoji TM M tako da je L(M)=L. Ako je, dodatno, to TM koja se uvek zaustavlja kažemo da je L rekurzivan (odlučiv) jezik, i da ga odlučuje ta TM.

U tom smislu, kada kažemo da je neki problem odlučiv mislimo na to da postoji način da ga kodiramo u nekom rekurzivnom jeziku nekog alfabeta.

Ispostavlja se da su klase determinističkih i nederminističkih Tjuringovih mašina ekvivalentne, u smislu da za svaku nedeterminističku mašinu postoji njoj ekvivalentna deterministička takva da sve jezike koje prihvata ili odbija jedna prihvata i odbija i druga, tako da je definicija regularnosti jezika nezavisna od modela mašine koji uzima u toj definiciji.

Jasno je da je moguće da se Tjuringova mašina zaustavi u jednom trenutku i završi sa radom prihvatajući ili odbijajući odgovor, kao i to da se mašina nikada ne zaustavi (budući da ima beskonačnu traku, iako samo konačan broj stanja). Jedan od najpoznatijih neodlučivih problema je problem zaustavljanja proizvoljne TM na proizvoljnom ulazu ("Halting Problem"):

Lema 1.2. *Jezik:*

$$L_H = \{Code(M) * w \mid \ M \textit{ prihvata } w\}$$

je neodlučiv.

Dokaz. Pretpostavimo suprotno, da postoji TM A koja odlučuje gornji problem. Konstruišimo prvo TM B koja kao potprogram sadrži TM A i ako A prihvata ulaz $Code(M)*w$ (tj. ako se M zaustavlja na ulazu w) tada B ulazi u beskonačnu petlju, a u suprotnom se zaustvlja.

Konstruišimo sada TM C koja za potprogram ima B. C za ulaz prima $Code(M)$, zatim ga duplira, praveći string $Code(M) * Code(M)$ i predaje kao ulaz u TM B.

Razmotrimo kako TM C radi na ulazu $Code(C)$. Ona ga prvo duplira, pravi string $Code(C) * Code(C)$ i zatim pušta mašinu B da odradi izračunavanje nad tim stringom. Ako se mašina C zaustavlja nad stringom $Code(C)$ tada će B ući u beskonačnu petlju, ali kako je on potprogram od C to znači da se C neće zaustaviti nad $Code(C)$, što je kontradikcija.

$\square$

Kodiranje Tjuringovih mašina

Upoznajmo se sada sa tehnikom kodiranja Tjuringovih mašina kao konačnih modela, što će biti jedan od glavnih alata u ovom radu.

Uzmimo, bez gubitka opštost, da se radi o nekoj determinističkoj TM, sa alfabetom $\Gamma = \{0, 1\}$:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_{accept}, Q_{reject}),$$

sa simbolima koji znače isto kao u definiciji 1.13. Pretpostavimo još i da je radni alfabet $\Gamma = \{0, 1\}$. Definišimo sada jezik koji će nam omogućiti da kodiramo TM:

$$\sigma = \{<, \underline{min},\ T_0(\cdot,\cdot),\ T_1(\cdot,\cdot),\ (H_q(\cdot,\cdot))_{q\in Q}\}$$

gde je:

- $<$ je linearno uređenje, a $\underline{\text{min}}$ je simbol konstante koji će predstavljati minimalni element u tom uređenju
- $T_0(\cdot,\cdot)$ *i* $T_1(\cdot,\cdot)$ su relacijski simboli koji će nam opisivati sadržaj TM trake u svakom trenutku; $T_i(p, t)$ će značiti da se simbol i (0 ili 1) nalazi u polju p u trenutku t.
- $H_q(\cdot,\cdot)$ su takodje relacijski simboli pomoću kojih ćemo opisivati položaje glave Tjuringove mašine, pa će nam $H_q(p, t)$ značiti da je u trenutku t glava mašine iznad pozicije p i da se ona nalazi u stanju q.

Sada ćemo napraviti rečenice φ_M tako da elementi od σ u strukturi imaju značenje koje smo im dalu u pasusu iznad. Jedan od načina da to uradimo je da nam za konkretnu TM M φ_M bude konjunkcija sledećih rečenica:

- Rečenica koja kaže da je "$<$" linearno uređenje i da je $\underline{\text{min}}$ minimalni element.
- Rečenice koja definiše početnu konfiguraciju od M. Za prazan ulaz rečenica bi bila:
$$H_{q_0}(\underline{min}, \underline{min})\ \wedge\ \forall p T_0(p, \underline{min}),$$

(kaže da se u početnom ($\underline{min}$) trenutku glava nalazi na prvoj ($\underline{min}$) poziciji na traci i da je mašina u stanju q_0 te da se na svim ostalim poljima nalaze nule.)

- Rečenice koja zabranjuje da se na nekoj poziciji na traci u isto vreme nadje više ili manje od jednog simbola iz radnog alfabeta:

$$\forall p \forall t (T_0(p,t) \leftrightarrow \neg T_1(p,t))$$

- Rečenice koja kaže da u svakom trenutku TM mora biti u jednom i samo jednom stanju i nad tačno jednim poljem:

$$\forall p \exists ! t (\bigvee_{q \in Q} H_q(p,t)) \;\wedge\; \neg \exists p \exists t (\bigvee_{q,q' \in Q, q \neq q'} H_q(p,t) \wedge H_{q'}(p,t))$$

- Rečenice koja postavlja uslov da se tranzicija iz jednog stanja u drugo mora odigravati po pravalima Tjuringove mašine. Na primer, tranziciju $\delta(q,0) = (q',1,L)$ možemo predstaviti sledećim dvema formulama, od kojih se prva bavi situacijom kada se glava u stanju q nalazi iznad polja koje nije početno ($p \neq \underline{min}$) u kojem slučaju menja upisani simbol iz 0 u 1, pomera glavu ulevo a sva druga polja ostaju netaknuta, dok druga formula govori o situaciji kada se glava nalazi iznad početne pozicije na traci ($p = \underline{min}$), u kojem slučaju se ne može ići dalje ulevo u narednom trenutku ($H_{q'}(p,t+1)$):

$$\forall p \forall t \left(\begin{array}{l} p \neq \underline{min} \\ \wedge\ T_0(p,t) \\ \wedge\ H_q(p,t) \end{array} \right) \rightarrow \left(\begin{array}{l} T_1(p,t+1) \\ \wedge\ H_{q'}(p-1,t+1) \\ \wedge\ \forall p'\ [\ p \neq p' \rightarrow \\ (\bigwedge_{i=0,1} T_i(p',t+1) \leftrightarrow T_i(p',t))\ ] \end{array} \right)$$

$$\forall p \forall t \left(\begin{array}{l} p = \underline{min} \\ \wedge\ T_0(p,t) \\ \wedge\ H_q(p,t) \end{array} \right) \rightarrow \left(\begin{array}{l} T_1(p,t+1) \\ \wedge\ H_{q'}(p,t+1) \\ \wedge\ \forall p'\ [\ p \neq p' \rightarrow \\ (\bigwedge_{i=0,1} T_i(p',t+1) \leftrightarrow T_i(p',t))\ ] \end{array} \right)$$

- rečenice koja kaže da se TM u jednom trenutku zaustavlja, odnosno dolazi u neko od stanja Q_{accept} *ili* Q_{reject}::

$$\exists p \exists t \bigvee_{q \in Qaccept \cup Q_{reject}} H_q(p,t)$$

Rečenica φ_M predstavlja konjunkciju ovih rečenica i ona ima konačan model ako i samo ako se TM M zaustavlja na datom ulazu i taj model zapravo predstavlja izračunavanje mašine M.

1.3 Teorija složenosti

Neka je M TM koja se uvek zaustavlja i neka ona prihvata jezik L. Pretpostavimo nadalje da postoji funkcija $f : \mathbb{N} \to \mathbb{N}$ takva da je broj tranzicija TM M pre nego što se zaustavi na proizvoljnoj reči w manji od $cf(|w|)$, za neku konstantu c. U slučaju da je M deterministička kažemo da $L \in DTIME(f)$, a u slučaju nedeterminističke TM, $L \in NTIME(f)$. U cilju oštrije klasifikacije problema po težini u ovom smislu - dakle koliko brzo raste utrošak resursa (u ovom slučaju vremena tj. broja tranzicija TM) - uvodimo neke važne klase problema:

$$PTIME = \bigcup_{k \in \mathbb{N}} DTIME(n^k),$$

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$$

Klase coPTIME je klasa jezika čiji su komplementi u P, dok je coNP klasa jezika čiji su komplementi u NP. PTIME je zatvoren za komplementiranje, odnosno komplement svakog jezika iz PTIME je takođe u PTIME, dok za klasu NP ovakva tvrdnja nije dokazana.

Takođe definišemo klase složenosti i u smislu toga koliko prostornih resursa zahteva rešenje pojedinog problema. Klasa DSPACE(f), za neku funkciju $f(n) \geq n$, je klasa jezika koje prihvata DTM da dužina zapisa na traci mašine za svaku pojedinačnu konfiguraciju za proizvoljan ulaz w ne prelazi $cf(|w|)$, za neku konstantu c.

Tako i ovde imamo klase:

$$PSPACE = \bigcup_{k \in \mathbb{N}} DSPACE(n^k),$$

no, u ovom slučaju se može pokazati da je $PSPACE = \bigcup_{k \in \mathbb{N}} NSPACE(n^k)$.

Prostornu složenost možemo definisati i za sublinearne funkcije, u kojem slučaju koristimo TM sa jos jednom, "radnom"trakom. Prva traka, ulazna nam služi za čitanje i na njoj nije moguće vršiti bilo kakve izmene, dok na radnoj traci obavljamo sve poslove kao i na običnoj TM. Tako definišemo klase DLOG (NLOG) kao skupove jezika za čije odlučivanje ni u jednom trenutku ne treba iskoristiti više od $O(log(|w|))$ ćelije za ulaz w.

Neka je $\Sigma_0^p = \Pi_0^p = PTIME$. Dalje definišemo $\Sigma_i^p = NP^{\Sigma_{i-1}^p}\ za\ i \geq 1$. Definišimo klasu Π_i^p kao klasu svih jezika čiji su komplementi u Σ_i^p. Jasno je da je $\Sigma_1^p = NP\ a\ \Pi_1^p = coNP$.

Polinomijalna hijerarhija je:

$$PH = \bigcup_{i \in \mathbb{N}} \Sigma_i^p = \bigcup_{i \in \mathbb{N}} \Pi_i^p$$

Ovde nam A^B označava klase problema rešivih TM mašinama složenosti A ali koje imaju "orakl" za odlučivanje problema složenosti B, odnosno mogu da reše svaki problem složenosti B u jednom koraku izračunavanja (ovaj model izračunavanja je poznat kao "oracle machine").

Izmedju klasa složenosti postoji sledeći odnos:

$$DLOG \subseteq NLOG \subseteq PTIME \subseteq \{NP, coNP\} \subseteq PH \subseteq PSPACE$$

Poznato je da je NLOG pravi podskup od PSPACE ali nije poznato da li postoji prava inkluzija između bilo koja dva neposredna suseda u gornjem nizu.

Pomenimo još dve klase:

$$EXPTIME = \bigcup_{k \in \mathbb{N}} DTIME(2^{n^k})$$

$$NEXPTIME = \bigcup_{k \in \mathbb{N}} NTIME(2^{n^k})$$

Primer 3. Pretpostavimo da imamo relacionu bazu podataka koja sadrži imena glavnih gradova svih zemalja na svetu, i relacija je takva da su $R(a, b)$ važi ako postoji direktna avionska linija od grada a do grada b. U logici prvog reda možemo postaviti upit da li je moguće od grada a do b doći uz najviše jedno presedanje, formulom: $\varphi(x, y) = E(x, y) \vee \exists z(E(x, z) \wedge E(z, y))$. No u logici prvog reda je nemoguće napraviti upit Q koji bi davao odgovor na pitanje da li je moguće doći od grada a do grada b (koristeći se samo avio-linijama koje imamo u našoj bazi). Primetimo i da postoje razlike u složenosti izračunavanja upita φ i Q. Jedno od pitanja kojim ćemo se baviti u nastavku je pokušaj da se upare određene klase složenosti i određene logike, tj. da damo odgovor na pitanje da li, za sve upite koje je moguće definisati u određenoj logici postoji klasa složenosti tako da je sve takve upite moguće izračunati u datoj složenosti. Drugim rečima, bavićemo se odnosom između definabilnosti i složenosti izračunavanja upita. $\triangle$

Kodiranje konačnih struktura i formula stringovima

Da bismo definisali složenost logike na konačnim modelima treba nam način da kodiramo konačne strukture i formule kao stringove. Formulu možemo pretvoriti u string tako što je zapišemo kao string reprezentaciju njenog sintaksičkog drveta. Ilustrujmo ovu tehniku narednim primerom:

Primer 4. Neka je dat alfabet $\sigma = \{R\}$ gde je R binarna relacija. Sintaksičko drvo sledeće formule:

$$\forall x \forall y(\neg(x = y) \rightarrow E(x, y))$$

Izgleda ovako:

$$\varphi = \forall x(\forall y(\rightarrow ((\neg(= (xy)))(Exy))))$$

Ako usvojimo sledeću konvenciju za predsavljanje simbola koji se javljaju u formulama:

- ($\longrightarrow$ 0001
-) $\longrightarrow$ 0010

- $\to \longrightarrow$ 0011
- $\leftrightarrow \longrightarrow$ 0100
- $\vee \longrightarrow$ 0101
- $\wedge \longrightarrow$ 0110
- $\exists \longrightarrow$ 0111
- $\forall \longrightarrow$ 1000
- $\neg \longrightarrow$ 1001
- $= \longrightarrow$ 1010
- $E \longrightarrow$ 1011
- $v_i \longrightarrow 1111bin(i)1111$

Poslednja stavka znači da će x biti predstavljeno stringom 111111111 a y stringom 1111101111. Dakle, kod naše formule bi bio (bez razmaka, one su ubačene radi preglednosti):

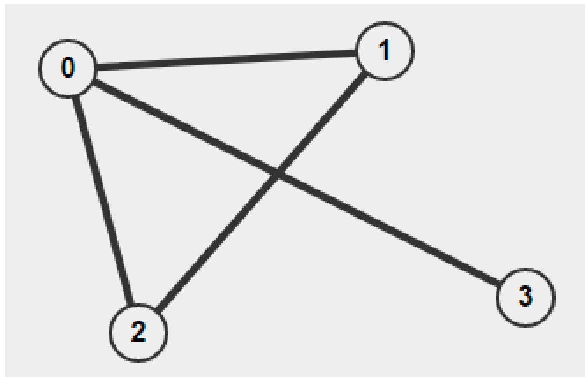
> $Code(\varphi) =$ 1000 111111111 0001 1000 1111101111 0001 0011 0001 0001 1001 0001 1010 0001 111111111 1111101111 0010 0010 0010 0001 1011 111111111 1111101111 0010 0010 0010 0010

$\triangle$

Neka je $\mathfrak{A}$ konačna struktura koja ima n elemenata i njima možemo pridružiti neko uređenje, odnosno numerisati ih, $a_1, a_2, ..., a_n$ i reći da važi $a_1 < a_2 < a_3 < ... < a_n$. Relaciju R arnosti k ćemo kodirati tako što poređamo u alfabetski poredak sve k-torke elemenata iz A (takvih k-torki ima n^k) i R predstavimo kao niz dužine n^k u kome na j-toj poziciji stoji cifra 1 ako je j-ta k-torka u relaciji R a cifra 0 ako nije i taj string ćemo obeležiti sa $Code(R)$. Ako je $\sigma = \{R_1, R_2, ..., R_s\}$ onda će njen kod biti $Code(R_1)Code(R_2)...Code(R_s)$. Za neke modele izračunavanja je potrebno dati još i kardinalnost modela, pa kod od $\mathfrak{A}$ definišemo:

$$Code(\mathfrak{A}) = 0^n 1 \cdot Code(R_1) \cdots Code(R_s)$$

Primer 5. Pokažimo kako bi izgledao kod grafa $G = (E, V)$ datog sledećom šemom:



Deo koda kojim se predstavlja relacija će biti 0111101011001000: prva nula govori da čvor 0 nije u relaciji sa samim sobom, 1 na drugom mestu da $(0,1) \in E$, naredno 1 da $(0,2) \in E$...poslednja 0 da $(3,3) \notin E$. Ispred toga još treba dodati string 00001, koji će označiti broj čvorova, te ukupan kod izgleda:

$$Code(G) = 000010111101011001000$$

$\triangle$

Definicija 1.21. *Neka je $\mathcal{K}$ klasa složenosti i $\mathcal{L}$ logika. Razlikujemo tri vrste složenosti:*

1) Složenost podataka od $\mathcal{L}$ je $\mathcal{K}$ ako za svaku rečenicu ϕ logike $\mathcal{L}$ jezik:

$$\{enc(\mathfrak{A}) \mid \ \mathfrak{A} \models \phi\}$$

pripada klasi $\mathcal{K}$.

2) Izražajna složenost od $\mathcal{L}$ je $\mathcal{K}$ ako za svaku strukturu $\mathfrak{A}$, jezik:

$$\{Code(\phi) \mid \ \mathfrak{A} \models \phi\}$$

pripada klasi $\mathcal{K}$.

3) Kombinovana složenost od $\mathcal{L}$ je $\mathcal{K}$ ako jezik:

$$\{(Code(\mathfrak{A}), Code(\phi)) \mid \ \mathfrak{A} \models \phi \ \}$$

pripada klasi $\mathcal{K}$.

Još kažemo da je složenost podataka (izražajna odnosno kombinovana složenost) od $\mathcal{L}$ $\mathcal{K}$-teška ako je za neko ϕ jezik $\{enc(\mathfrak{A}) \mid \ \mathfrak{A} \models \phi\}$ (za neko $\mathfrak{A}$ jezik $\{Code(\phi) \mid \ \mathfrak{A} \models \phi\}$, odnosno jezik $\{(Code(\mathfrak{A}), Code(\phi)) \mid \ \mathfrak{A} \models \phi \ \}$) $\mathcal{K}$-težak (tj. svaki problem složenosti $\mathcal{K}$ je moguće (sa zadatom složenošću) redukoviati na taj problem).

Za problem koji je u $\mathcal{K}$ a pride je i $\mathcal{K}$-težak kažemo da je $\mathcal{K}$-kompletan

Sada smo u prilici da definišemo složenost logike na konačnim strukturama:

Definicija 1.22. *Neka je $\mathcal{L}$ logika, $\mathcal{K}$ klasa složenosti i $\mathcal{C}$ neka klasa konačnih struktura. $\mathcal{L}$ hvata $\mathcal{K}$ na $\mathcal{C}$ ako važi:*

1. *Složenost podataka od $\mathcal{L}$ na $\mathcal{C}$ je $\mathcal{K}$.*
2. *Za svako svojstvo $\mathcal{P}$ koje imaju strukture iz $\mathcal{C}$ a koje može biti testirano sa koompleksnošću $\mathcal{K}$ postoji rečenica $\Phi_{\mathcal{P}}$ u logici $\mathcal{L}$ takva da za svaku strukturu $\mathfrak{A}$ iz $\mathcal{C}$ važi $\mathfrak{A} \models \Phi_{\mathcal{P}}$ akko $\mathfrak{A}$ ima svojstvo $\mathcal{P}$.*

   Ako je $\mathcal{C}$ skup svih konačnih struktura, kažemo da $\mathcal{L}$ hvata $\mathcal{K}$

Teorija koja se bavi opisivanjem klasa složenosti logikama se zove "Teorija opisne složenosti". Prvi rezultat iz ove teorije je Faginova Teorema (koja će biti dokazana u narednom poglavlju), u kojoj je prvi put klasa složenosti opisana nezavisno od konkretnog modela za izračunavanje, i koja tvrdi da logika $\exists SO$ hvata NP.

2 Tjuringove mašine i konačni modeli

U ovom delu ćemo detaljnije prezentovati i dokazati dve teoreme sa kojima je otpočelo proučavanje teorije konačnih modela kao zasebne oblasti.

2.1 Trakhtenbrot-ova teorema

U ovom odeljku ćemo biti u prilicu da pokažemo da u slučaju konačnih modela ne važe ključni stavovi teorije modela kao što su teoreme o kompaktnosti i potpunosti kao i da ovde nemamo analogon Lowenheim-Skolemove teoreme. Pažnju na takvu mogućnost je prvi put skrenula Trakhtebrot-ova teorema 1950. godine, koja je pokazala da teorema o potpunosti ne važi za konačne modele.

Zapišimo opet teoremu još jednom, radi preglednosti:

Teorema 2.1. *(Trakhtenbrot-ova teorema) Konačna zadovoljivost nije odlučiva u logici prvog reda, odnosno za svaki relacioni jezik σ sa barem jednim binarnim relacijskim simbolom važi da skup:*

$$FSAT[\sigma] = \{\varphi \mid \ \varphi \ konačno\ zadovoljiva\ rečenica\ prvog\ reda\ jezika\ \sigma\}$$

nije odlučiv.

Dokaz. Prvo ćemo dokazati lemu iz teorije izračunljivosti, na koju se dokaz Trakhtebrotove teoreme fundamentalno oslanja:

Lema 2.1. *Problem da li se proizvoljna TM M zaustavlja na praznom ulazu nije odlučiv. Odnosno jezik*

$$L_{H,\lambda} = \{Kod(M) \mid \ M\ se\ zaustavlja\ na\ praznoj\ reči\ \lambda\}$$

nije odlučiv.

Dokaz. Redukovaćemo problem zaustavljanja mašine na prozivoljnom ulazu ("halting problem") na problem zaustavljanja na praznom ulazu. Neka mašina A odlučuje problem zaustavljanja na praznom ulazu, ona za input prima Tjuringovu mašinu (kodiranu) i daje odgovor da li se ona zaustavlja ili ne na praznom ulazu. Neka je $M(w)$ TM koja na prazom ulazu upisuje w i onda simulira rad mašine M na ulazu w. Konstruišimo sada TM B koja odlučuje Halting problem.

1) B za ulaz prima TM M i reč w nad ulaznim alfabetom.

2) B simulira rad mašine A na ulazu $M(w)$, i po pretpostavci A se uvek zaustavlja. Ako A prihvata $M(w)$ tada B prihvata ako A odbija $M(w)$ tada B odbija.

□

Poslužićemo se postupkom kodiranja Tjuringovih mašina koji smo upoznali u prethodnom poglavlju. Uzmimo, bez gubitka opštost, da se radi o determinističkoj TM $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_{accept}, Q_{reject})$, sa simbolima koji znače isto kao u definiciji 1.19. Pretpostavimo još i da je radni alfabet $\Gamma = \{0, 1\}$, budući da će naša mašina raditi na praznom inputu, te neka 0 predstavlja prazan simbol. Rečenica koja definiše početni položaj će biti $H_{q_0}(\underline{min}, \underline{min}) \ \wedge \ \forall p T_0(p, \underline{min})$.

Ako se mašina zaustavlja, posle dolaska u završno stanje konfiguracije se ne menjaju pa bi φ_M (konjunkcija) mogla biti zadovoljena u konačnom modelu. Ako postoji takav model, tj. ako je rečenica φ_M zadovoljiva to znači da se TM M zaustavlja na praznom inputu. Ako bi problem konačne zadovoljivosti proizvoljne rečenice bio odlučiv iz toga bi sledilo da je problem zaustavljanja proizvoljne Tjuringove mašine na praznoj reči odluciv što znamo da nije slučaj, pa postoje i rečinice upravo konstruisanog jezika σ za koje se ne može algoritamski odlučiti da li su zadovoljive.

Do sada smo pokazali da postoji jezik (konstruisali smo ga) koji zadovoljava uslove teoreme. Pređimo sada na drugi deo dokaza, u kojem želimo da pokažemo da ona važi za svaki jezik koji ima barem jedan relacijski simbol. Opisaćemo tehniku logičke redukcije σ-struktura na $\{E\}$-strukture (ili interpretacije σ-struktura u $\{E\}$-strukturama), pri čemu je E binarni relacijski znak.

Neka je $\sigma = \{R_1, \ldots, R_k\}$, pri čemu je R_i relacijski simbol dužine n_i. Pod logičkom redukcijom jezika σ na jezik $\{E\}$ podrazumevaćemo niz formula:

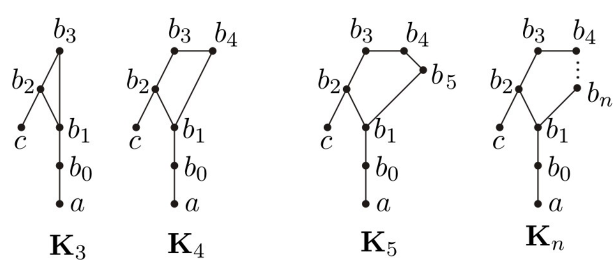
$$(*) \qquad \varphi_0(x), \varphi_1(x_1, \ldots, x_{n_1}), \ldots, \varphi_k(x_1, \ldots, x_{n_k}),$$

koji za svaku $\{E\}$-strukturu $\mathbf{G} = (G, E^{\mathbf{G}})$ odredjuje σ-strukturu $\mathbf{G}^*$ čiji je domen $G_0 = \{a \in G \mid \mathbf{G} \models \varphi_0(a)\}$ i

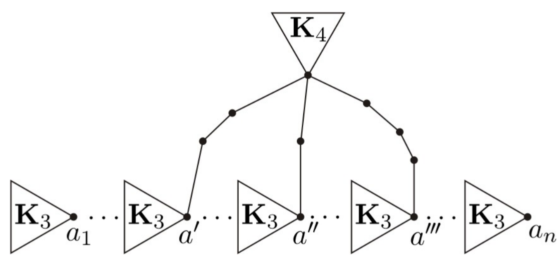
$$R_i^{\mathbf{G}^*} = \{(a_1, \ldots, a_{n_i}) \in G_0 \mid \mathbf{G} \models \varphi_i(a_1, \ldots, a_{n_i})\}, \ i = 1, \ldots, k.$$

Teorema. Za proizvoljan relacioni jezik $\sigma = \{R_1, \ldots, R_k\}$, postoji logička redukcija $(*)$ takva da za svaku σ-strukturu $\mathfrak{A}$ postoji graf $\mathbf{G} = (G, E)$ da je $\mathbf{G}^* \cong \mathfrak{A}$.

Skica dokaza. Logičku redukciju $(*)$ definišemo oslanjajući se na poznatu činjenicu da za svaki konačan graf $\mathbf{K}$ sa n čvorova, postoji $\{E\}$-formula $\varphi_{\mathbf{K}}(x_1, \ldots, x_n)$ takva da za svaki graf $\mathbf{G} = (G, E)$ i sve $v_1, \ldots, v_n \in G$ važi: $\mathbf{G} \models \varphi_{\mathbf{K}}(v_1, \ldots, v_n)$ akko je podgraf od $\mathbf{G}$ generisan sa $\{v_1, \ldots, v_n\}$ izomorfan sa $\mathbf{K}$. Grafove $\mathbf{K}_n$, $n \geq 3$, prikazane na narednoj slici nazivaćemo n-oznakama (sa početkom a):



Neka je $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \dots, R_k^{\mathfrak{A}})$ proizvoljna σ-struktura. Graf (G, E) definišemo polazeći od skupa A, pri čemu za svaki element $a \in A$ lepimo po jednu kopiju grafa $\mathbf{K}_3$ (slika ispod). Relacijske simbole R_i, $i = 1, \dots, k$ kodiramo redom $(i+3)$-oznakama, a zatim za n_i-torke iz $R_i^{\mathfrak{A}}$ lepimo K_{i+3} dodajući, izmedju koordinata n_i-torke i početka grafa $\mathbf{K}_{i+3}$, odgovarajući broj medjučvorova koji opisuju mesto koordinate u n_i-torci. Na primer, ako je R_1 relacijski simbol dužine 3, tj. $n_1 = 3$, i $(a'', a', a''') \in R_1^{\mathfrak{A}}$, onda postupamo kao što je prikazano na narednoj slici.



$\square$

Sada samo u prilici da dokažemo još nekoliko rezultata iz teorije konačnih modela u kojima se oni razlikuju od beskonačnih modela, kako smo to nagovestili u pretodnom poglavlju.

Posledica 2.1. *Ako σ sadrži barem jedan binarni relacijski simbol tada skup:*

$$FVAL = \{\varphi \in FO[\sigma] \mid \ \varphi \text{ rečenica valjana u svim konačnim strukturama}\}$$

nije rekurzivno nabrojiv.

Dokaz. Prvo primetimo da je skup FSAT rekurzivno nabrojiv - prosto za svaku rečenicu jezika φ idemo redom kroz konačne strukture i isprobavamo da li neka od njih zadovoljava φ, ako je rečenica zadovoljiva takva struktura će se pojaviti pre ili kasnije i algoritam će se zaustaviti.

Za svaku rečenicu φ važi:

$$\varphi \notin FSAT[\sigma] \text{ akko } \neg\varphi \in FVAL[\sigma]$$

Ako bi FVAL bio rekurzivno nabrojiv, tada bismo za svaku rečenicu prvog reda jezika σ mogli da uzmemo njenu negaciju i proverimo da li je u jeziku FVAL$[\sigma]$. Iz toga sledi da je $FO[\sigma] \setminus FSAT[\sigma]$ rekurzivno nabrojiv pa možemo napraviti algoritam koji odlučuje FSAT$[\sigma]$: pustimo da istovremeno rade algoritmi koji nabrajaju $FO[\sigma] \setminus FSAT[\sigma]$ i $FSAT[\sigma]$ i taj proces će se za svako φ u jednom trenutku zaustaviti pa ćemo moći da odlučimo taj skup, čemu se protivi Trakhtebrotova teorema. $\square$

Primetimo još da ako bi važila teorema potpunosti to bi impliciralo da je skup FVAL rekurzivno nabrojiv, jer bi smo, za proizvoljnu rečenicu φ mogli da u nekom deduktivnom sistemu algoritamski proizvodimo redom dokaze i onda bi smo u jednom trenutku dobili dokaz za φ (ako ona jeste dokaziva), te bi iz potpunosti sledilo da je ta rečenica valjana. Dakle, iz prethodno dokazane posledice sledi da za konačne modele ne važi teorema potpunosti.

Još jedna važna posledica Trakhtebrotove teoreme je i to što u teoriji konačnih modela ne postoji analogon Lowenheim-Skolemove teoreme, odnosno teorema koja bi odozgo ograničavala veličinu modela za pojedine zadovoljive rečenice:

Posledica 2.2. *Ne postoji rekurzivna funkcija f takva da za svaku rečenicu φ važi da ako ona ima model onda je on kardinalnosti manje of $f(\varphi)$.*

Dokaz. Kada bi takva funkcija postojala omogućila bi nam da za svaku rečenicu odlučimo da li ima konačan model ili ne - proverili bismo sve modele kardinalnosti do $f(\varphi)$, ako tu nema modela onda ga nikako nema. □

2.2 Faginova-ova teorema i NP

Faginova teorema je jedan od prvih rezultat iz deskriptivne teorije složenosti čiji je zadatak karakterizacija klasa složenosti tipovima logika, bez direktnog korišćenja apstraktnih modela za izračunavanje (kakav je Tjuringova mašina). Odnosno, želimo da pokažemo da na određenom domenu struktura $\mathcal{C}$, neka logika $\mathcal{L}$ (npr. logika prvog reda, logika fiksne tačke i slično) *hvata* klasu složenosti $\mathcal{K}$.

Pre prelaska na Faginovu teoremu dokažimo jednu lemu koja će nam biti od koristi u nastavku:

Lema 2.2. *Neka je Φ rečenica prvog reda jezika σ i neka je $\mathfrak{A}$ konačna struktura. Ako je širina rečenice (maksimalan broj slobodnih varijabli koji može imati njena podformula) k tada provera da li $\mathfrak{A} \models \Phi$ može biti urađena u vremenu:*

$$O(\|\Phi\| \times \|\mathfrak{A}\|^k)$$

Dokaz. Pretpostavimo bez umanjenja opštosti da Φ koristi simbole $\wedge$, $\neg$, $\exists$ a ne kosristi $\vee$ ili $\forall$. Neka su $\varphi_1, ...\varphi_m$ sve podformule od Φ, koje mogu imati maksimalno k promenljivih. Konstruišimo sada za svako i (tj. za svaku potformulu φ_i) $\varphi_i(\mathfrak{A})$ (odnosno $Code(\varphi_i(\mathfrak{A}))$), koristeći se indukcijom po složenosti formule. Ako φ_i ima k_i slobodnih promenljivih važiće da je $\varphi_i(\mathfrak{A}) \subseteq A^{k_i}$ i biće reprezentovana stringom nula i jedinica dužine n^{k_i} gde je $n = |A|$.

Ako je φ_i atomska formula $R(x_1, ..., x_{k_i})$ tada je $Code(\varphi_i(\mathfrak{A}))$ kodiranje R u $Code(\mathfrak{A})$.

Ako je φ_i jednako $\neg\varphi_j$ prosto ćemo u reprezentaciji za $\varphi_j(\mathfrak{A})$ samo prebaciti sve nule u jedinice i obrnuto.

Ako je φ_i jednako $\varphi_j \wedge \varphi_l$ imamo dva slučaja:

1) Ako φ_j *i* φ_l imaju sve slobodne promenljive jednake tada $\varphi_i(\mathfrak{A})$ dobijamo kao konjunkciju po bitovima od $\varphi_j(\mathfrak{A})$ i $\varphi_l(\mathfrak{A})$. Prosto da bi bit od $\varphi_i(\mathfrak{A})$

bio 1 odnosno da bi neki vektor zadovoljio φ_i on mora da zadovolji i φ_j i φ_l.

2) Ako ovo iznad nije slučaj, tada $\varphi_i(\vec{x},\vec{y},\vec{z}) = \varphi_i(\vec{x},\vec{y}) \wedge \varphi_i(\vec{x},\vec{z})$, i $\varphi_l(\mathfrak{A})$ dobijama pronalazeći sve vektore $\vec{a} \in A^{\vec{x}}$, $\vec{b} \in A^{\vec{y}}$ i $\vec{c} \in A^{\vec{z}}$ takve da su bitovi koji odgovaraju bitovima vektora $(\vec{a},\vec{b})$ u $\varphi_j(\mathfrak{A})$ i bitovi koji odgovaraju bitovima vektora $(\vec{a},\vec{c})$ u $\varphi_l(\mathfrak{A})$ postavljeni na 1, i onda postavljanjem i onda postavljanjem korespodencije sa bitovima vektora $(\vec{a},\vec{b},\vec{c})$ u $\varphi_i(\mathfrak{A})$ na 1.

3) Ako je $\varphi_i(\vec{x}) = \exists z \varphi_j(z,\vec{x})$ tada prolazimo kroz $\varphi_j(\mathfrak{A})$ i ako je bit koji odgovara $(a,\vec{a})$ jednak 1, tada postavimo bit koji odgovara $\vec{a}$ u $\varphi_i(\mathfrak{A})$ na 1.

Svaki od ovih algoritama je složenosti najviše $O(\|\mathfrak{A}\|^k)$ iz čega sledi naš rezultat. $\square$

Teorema 2.2. *(Faginova teorema) $\exists SO$ hvata NP.*

Dokaz. I) Pokažimo prvo da je klasa složenosti podataka logike $\exists SO$ upravo NP klasa. Neka je $\psi = \exists R_1 \exists R_2 ... \exists R_n \varphi$ rečenice iz $\exists SO$, naš cilj je sada da pokažemo da se može konstruisati NP algoritam M koji bi za svaku strukturu $\mathfrak{A}$, tačnije rečeno za string Code($\mathfrak{A}$) odlučivao da li $\mathfrak{A} \models \psi$. M prvo, za zadatu strukturu $\mathfrak{A}$ nedeterministički pogadja koje su to relacije $R_1, .., R_n$. Relacija R_i je odredjena binarnim stringom dužine n^{r_i}, gde je r_i arnost od R_i $n = |A|$. Tada M odlučuje da li $(\mathfrak{A}, R_1, ..., R_n) \models \varphi$. Kako je φ FO-rečenica ovo može biti odlučeno u polinomijalnom vremenu, po prethodnoj lemi. Dakle M se sastoji od nedeterminističkog pogadjanja binarnog stringa polinomijalne dužine a zatim njegovog proveravanja u polinomijalnom vremenu, pa je klase NP.

II) Dužni smo još da dokažemo drugi uslov, da za svako svojstvo koje imaju konačne strukture a koje se može odlučiti u NP vremenu postoji rečenica iz $\exists SO$ čiji konačni modeli su upravo one (i samo one) strukture $\mathfrak{A}$ koje imaju to svojstvo (u takvoj situaciji možemo reći da je to svojstvo izraženo tom rečenicom).

Neka je dat jezik σ i neko svojstvo $\mathcal{P}$ koje σ-strukture mogu imati i koje možemo da testiramo u NP tj. kada imamo (kodiranu) strukture u NP vremenu možemo da odgovorimo da li ta struktura ima to svojstvo. Neka nam taj problem odlučuje konkretna Tjuringova mašina $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_a, Q_r)$ u standardnom značenju ovih simbola, kako su prezentovani u uvodom poglavlju, i neka ta mašina radi u klasi složenosti NP. Bez umanjenja opštosti možemo pretpostavit da je $\Sigma = \{0, 1\}$ a $\Gamma = \{0, 1, \_\}$, gde će nam "_" predstavljati prazno polje. Pretpostavićemo još i da M svoj posao završava u vremenu n^k i da u radu posećuje sva polja koja je zauzeo ulazni string - što je dužina kodiranja konkretne σ-strukture.

Rečenica koja nama treba, koja će biti zadovoljiva samo u modelima u kojima važi $\mathcal{P}$ je (obeležimo li broj stanja od NTM M sa m):

$$\exists < \exists T_0 \exists T_1 \exists T_2 \exists H_{q_0}, ..., \exists H_{q_{m-1}} \Psi,$$

gde je Ψ rečenica jezika $\sigma \cup \{<, T_1, T_2, T_0, \{H_q\}_{q\in Q}\}$, od datih relacijskih simbola $<$ je binarni a ostali $2k$-arni. Ove simbole interpretiramo na sledeći način:

- $<$ je linearno uređenje domena. Zahvaljujući njemu možemo imati uređenje $\leq_k$ na uređenim k-torkama koje onda možemo iskoristiti za indikaciju mesta na traci ($\vec{p}$) i trenutka u vremenu ($\vec{t}$), kojih može bit maksimalno n^k pa će pa će k-torke biti dovoljne za razlikovanje svakog od njih.

- T_0, T_1, T_2 će nam služiti za opis koji je simbol upisan u koje polje i u kom trenutku, baš kao i u dokazu Trakhtenbrotove teoreme. $T_2(\vec{p}, \vec{q})$ će označavati da je prazan simbol ("_") upisan na mesto $\vec{p}$ u trenutku $\vec{t}$.

- H_q-ovi će takodje imati funkciju kakvu smo im već pridodavali, da opisuju kad se glava nalazi u stanju q, u polju $\vec{p}$ i vremenu $\vec{t}$.

Sada treba da definišemo kako će izgledati rečenica Ψ. Pretpostavimo bez umanjenja opštosti linearno uređenje na A da bude $<$. Rečenica Ψ će biti konjunkcija sledećih rečenica:

- Rečenice koja kaže da je $<$ linearno uređenje.

- Rečenice koje smo imali i u dokazu Trakhtenbrotove teoreme a koje kažu da u svakom trenutku (u svakoj konfiguraciji) svako polje TM sadrži upisan tačno jedan simbol, da je u svakom trenutku glava iznad tačnog jednog polja a mašina u tačno jednom stanju iz Q i da M u jednom trenutku prihvata ulaz tj. dolazi u neko od stanja iz Q_a.

- Rečenice koja govori da prelazak iz jedne konfiguracije u drugu mora poštovati funkciju δ, koje smo imali i kod Trakhtenbrota osim što ovde u igru moramo uvesti i nedeterminizam, pa za svako $a \in \Gamma$ i $q \in Q$ imamo rečenicu:
$$\bigvee_{(q',b,mov)\in\delta(q,a)} \alpha(q,a,q',b,mov),$$
gde je $mov \in \{L, R\}$ a $\alpha(q,a,q',b,mov)$ je rečenica koja opisuje situaciju kada mašina u stanju q pročita a zatim upiše b, pomeri se levo (L) ili desno (R) i predje u stanje q', koja se zapisuje jednako kao i kod Trakhtenbrota.

- Rečenice koja definiše početnu konfiguraciju. Kada bi imali rečenice $\nu(\vec{p})$ i $\xi(\vec{p})$, za koje bi važilo $\mathfrak{A} \models \nu(\vec{p})$ akko je $\vec{p}$-ta pozicija u stringu $Code(\mathfrak{A})$ jednaka 1 i $\mathfrak{A} \models \xi(\vec{p})$ akko je dužina od $Code(\mathfrak{A})$ manja od $\vec{p}$, tada bismo inicijalnu konfiguraciju mogli definisati sledećom rečenicom:
$$\forall\vec{p}\forall\vec{t}\left(\neg\exists\vec{u}\ (\vec{u} \leq_k \vec{t}) \ \rightarrow \ \begin{bmatrix} \big(\nu(\vec{p}) \leftrightarrow T_1(\vec{t},\vec{p})\big) \\ \wedge \big(\xi(\vec{p}) \leftrightarrow T_2(\vec{t},\vec{p})\big) \end{bmatrix}\right),$$
koja kaže da se u početnom trenutku na traci nalazi $Code(\mathfrak{A})$ a da su sva ostala polja prazna.

Formula koju smo napravili, $\exists < \exists T_0 \exists T_1 \exists T_2 \exists H_{q_0}, ..., \exists H_{q_{m-1}} \Psi$ važi u $\mathfrak{A}$ akko M prihvata $Code(\mathfrak{A})$.

Još treba da pokažemo da formule $\nu(\vec{p})$ i $\xi(\vec{p})$ postoje odnosno da možemo da ih konstruišemo. Pokazaćemo ovo za jezik $\sigma = \{R\}$ gde je R binarna relacija, što je situacija koja se lako uopštava na druge slučajeve. Takođe ćemo pretpostavit da je $k = 3$, a za porizvoljno k se slično izvodi. Neka je takodje $A = \{0, 1, ...n-1\}$, pa pozicije zapisujemo uredjenim trojkama elemenata iz A, $\vec{p} = (p_1, p_2, p_3)$. Kako imamo samo jednu relaciju dužine 2, to je $Code(\mathfrak{A})$ dužine $n^2 + n + 1$, dok $\vec{p}$ predstavlja položaj $p_1 n^2 + p_2 n + p_3$, odakle je jasno da p_1 ne može biti veće od 1 ako hoćemo da $\nu(\vec{p_1})$ važi u $\mathfrak{A}$. Ako je $p_1 = 0$ tada $\vec{p}$ označava poziciju $p_2 n + p_3$ a tada p_2 ne može biti 0 da bi $\nu(\vec{p})$ bilo tačno jer prvih $n+1$ mesta u kodu od $\mathfrak{A}$ označavaju broj elemenata od A, prvih n mesta su nule. Ako $p_3 \neq 0$ tada je $p_2 n + p_3 = (p_2 - 1)n + (p_3 - 1) + (n + 1)$ a to je pozicija na stringu $Code(E)$ koja odgovara paru $(p_2 - 1, p_3 - 1)$ (odnosno ona je po alfabentom poretku na mestu broj $(p_2 - 1)n + (p_3 - 1)$ odbrojavajući od mesta $n + 1$), pa je $\nu(0, p_2, p_3) = 1$ ako važi da su $p_2 - 1$ $ip_3 - 1$ u relaciji R (za $p_3 \neq 0$). Ako je $p_3 = 0$ tada ν važi ako je E$(p_2 - 2, n - 1)$. Dakle, formula $\nu(\vec{p}) = \nu(p_1, p_2, p_3)$ može biti predstavljena sledećom formulom:

$$\begin{array}{l} \left[\begin{pmatrix} (p_1 = 0) \\ \wedge\ (p_2 > 1) \end{pmatrix} \wedge \begin{pmatrix} (p_3 \neq 0) \wedge R(p_2 - 1, p_3 - 1) \\ \wedge\ (p_3 = 0) \wedge R(p_2 - 2, n - 1) \end{pmatrix} \right] \\ \vee\ [(p_1 = 0) \wedge (p_2 = 1) \wedge (p_3 = 0)] \vee [(p_1 = 1) \wedge ...] \end{array},$$

za slučaj kad je $p_1 = 1$ se uradi sličan postupak.

Jasno je da postoji formula za $\xi(\vec{p})$, koja prosto kaže da kad se pređe dužina od $Code(\mathfrak{A})$ da su sva polja prazna.

$\square$

Kako je već napomenuto, važnost ove teorem je u tome što je njom klasa složenosti (NP) karakterisana čistim logičkim formalizmom. Kasnije ćemo videti kako se slično može učiniti i za neke druge klase. Ova tehnika nam time otvara novi teren na kome možemo da ispitujemo i dokazujemo stvari vezane za klase složenosti. Npr. da bi razdvojili dve klase složenosti dovoljno je da razdvojimo logike koje ih hvataju.

Kako je negacija svake rečenice iz $\exists SO$ rečenica iz $\forall SO$ a coNP skup komplemenata svih problema koji su u NP to iz Faginove teoreme odmah sledi da $\forall SO$ hvata coNP, te da bi se dokazalo da $NP \neq coNP$ bilo bi dovoljno pokazati $\exists SO \neq \forall SO$.. Pošto znamo da je $P = coP$ taj dokaz bi bio dovoljan da se dokaže da $P \neq NP$.

Pokažimo još jednu posledicu Faginove teoreme:

Posledica 2.3. *SAT (problem zadovoljivosti iskazne formule) je NP-kompletan.*

Dokaz. Jasno je da SAT jeste NP problem. Potrebno je dokazati da se svaki NP problem može svesti na SAT. Neka je $\mathcal{P}$ neki problem iz NP. Mi njega možemo posmatrati i kao klasu konačnih struktura $\mathcal{P}$ (to možemo učiniti sa svakim jezikom $L \subseteq \Gamma^*$, koga možemo posmatrati kao klasu konačnih struktura nad jezikom $\{<\} \cup \{P_a \mid a \in \Gamma\}$; naime, reč $w = w_0 w_1 ... w_{m-1} \in \Gamma^*$ je opisana strukturom $\mathfrak{B}(w)$, čiji je univerzum {0, 1,...,m-1}, $<$ je interpretirano na standardan način a $P_a = \{i \mid w_i = a\}$). Po Faginovoj teoremi postoji FO-rečenica ψ takva da važi:

$$\mathcal{P} = \{\mathfrak{A} \mid \ \mathfrak{A} \models \exists R_1 \exists R_2 ... \exists R_m \psi\}$$

Sada ćemo prikazati redukciju koja svakoj strukturi $\mathfrak{A}$ pridružuje iskaznu formulu $\psi_{\mathfrak{A}}$. Za dato $\mathfrak{A}$ zamenimo u ψ:

- sve podformule $\exists x_i \varphi$ sa $\bigvee_{a_i \in A} \varphi[x_i/a_i]$
- sve podformule $\forall x_i \varphi$ sa $\bigwedge_{a_i \in A} \varphi[x_i/a_i]$
- sve σ-atome P($\vec{a}$) njihovom istinitosnom vrednošću u $\mathfrak{A}$.

Ove se translacije mogu izvršiti dovoljno efikasno budući da je takva evaluacija atomskih formula. Posmatrajući atome $R_i(\vec{a})$ kao iskazne promenljive dobili smo formulu $\psi_{\mathfrak{A}}$ takvu da:

$$\mathfrak{A} \in \mathcal{P} \text{ akko } \mathfrak{A} \models \exists R_1 \exists R_2 ... \exists R_m \psi \text{ akko } \psi_{\mathfrak{A}} \in SAT$$

□

3 Logike fiksne tačke i klase složenosti

Kako smo videli u uvodnom poglavlju, logika prvog reda ima vrlo ograničen ekspresivni potencijal, npr. za ne-lokalne osobine kao i osobine koje intuitivno ukljucuju iterativne ili dinamičke koncepte. Na primer, tranzitivno zatvorenje binarne relacije E je lako generisati rekurzijom na relacionom operatoru $R \to R \cup R \circ R$ gde je $A \circ B := \{(x, y) \mid (x, z) \in A, (z, y) \in B, \text{ za neko } z\}$. Na strukturi $(\mathfrak{A}, E^{\mathfrak{A}})$ kardinalnosti n, iterativna primena ovog operatora počevši od $R_0 = E^{\mathfrak{A}}$ daje niz $R_{i+1} = R_i \cup R_i \circ R_i$. Ovaj postupak se zaustavlja u smislu da dolazi do relacije R_i takvog da $R_i \circ R_i \subseteq R_i$, odnosno fiksne tačke $R_{i+1} = R_i$. Ova konačna vrednost R_i je tranzitivno zatvorenje od $E^{\mathfrak{A}}$.

U ovom odeljku ćemo proučiti proširenja logike prvog reda operatorima za računanje fiksnih tačaka, posmatrati njihovu ekspresivnu moć i pokazati kako na odredjenim strukturama ove logike hvataju neke važne klase složenosti.

3.1 Fiksne tačke operatora na skupovima

Neka je dat skup U i neka je $P(U)$ partitivni skup. Operator na U je preslikavanje $F : P(U) \to P(U)$. F je monoton ako važi:

$$A \subseteq B \;\Rightarrow\; F(A) \subseteq F(B).$$

F je inflacioni ako važi $A \subseteq F(A)$ za svako A.

Definicija 3.1. *Skup X je fiksna tačka operatora F ako važi $F(X) = X$. X je najmanja fiksna tačka od F ($lfp(F)$) ako za svaku drugu fiksnu tačku Y važi $X \subseteq Y$.*

Posmatrajmo, nadalje, niz:

$$X^0 = \emptyset,\; X^{i+1} = F(X^i) \tag{2}$$

F je induktivno ako za svako i važi $X^i \subseteq X^{i+1}$. Jasno je da ako je F monotono da je i induktivno jer $X^0 = \emptyset \subseteq F(X^0)$ i dalje se odmotava indukcijom.

Za induktivne operatore F definišemo:

$$X^\infty = \bigcup_{n \in \mathbb{N}} X^n$$

Budući da mi ovde radimo sa konačnim modelima jasno je da će se ovaj niz jednom stabilizovati, da će se za neko k desiti $X^{k-1} = X^k = X^\infty$.

Dokažimo sada da svaki monotoni operator ima fiksnu tačku

Teorema 3.1. *(Tarski-Knaster) Svaki monotoni operator F: $P(U) \to P(U)$ ima najmanju fiksnu tačku lfp(F) za koju važi:*

$$lfp(F) = \bigcap\{Y \mid Y = F(Y)\} = X^\infty,$$

za $X^\infty = \bigcup_{n\in\mathbb{N}} X^n$

Dokaz. Neka je $W = \{Y \mid F(Y) \subseteq Y\}$. Ovaj skup jeste neprazan jer za U važi $F(U) \subseteq U$. Dokažimo prvo da je skup $S = \bigcap W$ jedna fiksna tačka operatora F, tj. da je $F(S) = S$. Pošto za proizvoljno $Y \in W$ važi $S \subseteq Y$ imamo $F(S) \subseteq F(Y) \subseteq Y$ pa $F(s) \subseteq S$. Primenom operatora F na poslednju nejednakost dobijamo i $F(F(S)) \subseteq F(S)$ te $F(S) \in W$ odakle po definiciji skupa S dobijamo $S \subseteq F(S)$, pa S jeste fiksna tačka.

Neka je sada skup svih fiksnih tačaka $W' = \{Y \mid F(Y) = Y\}$ i neka je $S' = \bigcap W'$. Kako je S element od W to važi $S' \subseteq S$ a kako je $W' \subseteq W$ to važi i $S \subseteq S'$ pa imamo $S = S'$, odnosno da je S presek svih fiksnih tačaka pa je on i najmanja fiksna tačka.

Ostalo je još da dokažemo drugi deo tvrđenja, da je $X^\infty = lfp(F)$. Jasno je da X^∞ jeste fiksna tačka. Dokažimo još da je $X^\infty \subseteq S$ tako što ćemo induktivno pokazati da svaki od skupova X^i pripada svakom od skupova Y iz W. Za X^0 situacija je očigledna. Neka sada za sve Y iz W važi $X^i \subseteq Y$. Tada $X^{i+1} = F(X^i) \subseteq F(Y) \subseteq Y$ za svako Y iz W. □

Neka je G sada proizvoljan operator. Definišimo inflacioni operator $G_{infl}(X) = X \bigcup G(X)$. Samim tim on je i induktivni operator odnosno postoji naš skup X^∞, koji u ovom slučaju zovemo inflaciona fiksna tačka operatora G, $ifp(G)$.

Za proizvoljan operator F niz (2) se može u jednom trenutku stabilizovati, odnosno može postojati neko n takvo da $X^n = X^{n+1}$ ili se može desiti da se taj niz nikad ne stabilizuje. Pošto skup U ima samo konačno mnogo elemenata te konačno mnogo podskupova - $2^{|U|}$, to će se u prvom slučaju niz morati stabilizovati u nekoj od prvih $2^{|U|}$ iteracija, posle čega se neki od skupova mora ponoviti i formirati se beskonačna petlja. Obzirom na ovo razmatranje definišemo *parcijalnu fiksnu tačku*:

$$pfp(f) = \begin{cases} X^n & ako\ X^n = X^{n+1} \\ \emptyset & ako\ X^n \neq X^{n+1}\ za\ svako\ n \leq 2^{|U|} \end{cases}$$

Stav 3.1. *Za monoton operator F važi $lfp(F) = ifp(F) = pfp(F)$*

Dokaz. Jasno je da važi $lfp(F) = pfp(F)$. No niz koji se stabilizuje da bi njegov poslednji element predstavljao ifp(F), dakle niz

$$\emptyset \subseteq \emptyset \cup F(\emptyset) \subseteq \emptyset \cup F(\emptyset) \cup F(\emptyset \cup F(\emptyset))...$$

je tačno jednak, član po član, nizu:

$$\emptyset \subseteq F(\emptyset) \subseteq F(F(\emptyset))$$

jer $\emptyset \subseteq F(\emptyset)$ pa je $\emptyset \cup F(\emptyset) = F(\emptyset)$, dalje zbog monotonosti je $\emptyset \cup F(\emptyset) \subseteq F(\emptyset \cup F(\emptyset))$ pa je $\emptyset \cup F(\emptyset) \cup F(\emptyset \cup F(\emptyset)) = F(\emptyset \cup F(\emptyset)) = F(F(\emptyset))$ i tako dalje. □

3.2 Logike Fiksne Tačke

Pozabavimo se sada time kako proširujemo FO logiku operatorima fiksne tačke. Neka je dat relacioni jezik σ i neka je R dodatni relacijski simbol arnosti

k a $\varphi(R, x_1, ..., x_k)$ formula jezika $\sigma \cup R$. Za konkretnu konačnu σ-strukturu $\mathfrak{A}$ ova rečenica definiše jedan operator na $P(A^k)$ na sledeći način:

$$F_\varphi(X) = \{(a_1, ...a_k) \in A^k \mid \mathfrak{A} \models \varphi(X/R, a_1, ...a_k)\}$$

gde se u $\varphi(X/R, a_1, ...a_k)$ relacija R tumači kao pripadnost skupu X.

Definišimo sada logike IFP i PFP.

Definicija 3.2. *1) IFP logika predstavlja logiku prvog reda proširenu tako da je u njoj formula (sa slobodnim promeljivima u $\vec{t}$) i:*

$$[ifp_{R,\vec{x}}\varphi(R, \vec{x})](\vec{t}),$$

gde je R k-arni relacijski simbol, $\varphi(R, \vec{x})$ formula, $\vec{t}$ uredjena k-torka termi a $|\vec{x}| = k$, sa značenjem:

$$\mathfrak{A} \models [ifp_{R,\vec{x}}\varphi(R, \vec{x})](\vec{a}) \text{ akko } \vec{a} \in ifp(F_\varphi)$$

2) PFP logika predstavlja logiku prvog reda proširenu tako da je u njoj formula (sa slobodnim promeljivima u $\vec{t}$) i:

$$[pfp_{R,\vec{x}}\varphi(R, \vec{x})](\vec{t}),$$

gde je R k-arni relacijski simbol, $\varphi(R, \vec{x})$ formula, $\vec{t}$ uredjena k-torka termova, $|\vec{x}| = k$, sa značenjem:

$$\mathfrak{A} \models [pfp_{R,\vec{x}}\varphi(R, \vec{x})](\vec{a}) \text{ akko } \vec{a} \in pfp(F_\varphi)$$

Videli smo da ma kakav bio operator F_φ, $ifp(F_\varphi)$ i $pfp(F_\varphi)$ će postojati, medjutim mi smo lfp definisali samo za monotone operatore, stoga nije odmah jasno da li bi gornja definicija bila korektna i za definisanje najmanje fiksne tačke. Zapravo:

Lema 3.1. *Problem da li prizvoljna formula prvog reda φ indukuje monoton operator F_φ je neodlučiv.*

Dokaz. Ova lema sledi iz Trakhenbrotove teoreme i činjenice da za proizvoljnu rečenicu ψ i formulu $\varphi(R, x) = (R(x) \to \psi)$ važi:

$$F_\varphi \text{ je monoton u } \mathfrak{A} \text{ akko } \psi \text{ je konačno zadovoljiva u } \mathfrak{A},$$

a desna strana gornje ekvivalencije je neodlučiva. Pokažimo još zašto je ekvivalencija tačna.

=>) Pretpostavimo da ψ nije tačna u nekoj $\mathfrak{A}$. Tada je $\mathfrak{A} \models \varphi(x)$ akko$\neg R(x)$, odnosno $F_\varphi(X) = U/X$ pa $F_\varphi(X)$ nije monotono.

<=) Ako ψ jeste tačna u nekoj $\mathfrak{A}$ tada za svako $X \subseteq A^k$ (k je arnost od R) važi $F_\varphi(X) = U$ pa F_φ jeste monoton u $\mathfrak{A}$.

□

Da bismo izbegli ovu nezgodnu situaciju i uspeli da definišemo lfp, uvedimo pozitivnosti formule u relaciji. Za formulu φ koja sadrži relaciju R kažemo da je pojavljivanje te relacije u formuli pozitivno ako se R nalazi pod dejstvom parnog broja znakova negacije u formuli φ zapisanoj samo pomoću veznika $\neg$, $\wedge$ i $\exists$, a negativno ako se nalizi pod dejstvom neparnog broja ovih znakova u naznačenom obliku formule φ. Na primer, u formuli $\forall x \exists y \neg R_1(x,y) \wedge \neg \exists z((R_2(x,z) \vee \neg R_3(y,z)))$ je pojavljivanje R_1 *i* R_2 negativno a R_3 pozitivno. Za formulu φ kažemo da je pozitivna u R ako u njoj nema negativnih pojavljinja simbola R. Iz sledeće leme će biti jasan značaj ovog pojma za definiciju lfp.

Lema 3.2. *Ako je $\varphi(R,\vec{x})$ pozitivna u R tada je F_φ monotono.*

Dokaz. Neka su X i Y dva proizvoljna podskupa od A^k, takvi da $X \subseteq Y$. Ako se R u formuli φ uopšte ne pojavljuje (u kojem slučaju je formula takodje pozitivna u R) situacija je trivijalna ($F_\varphi(X) = F_\varphi(Y)$).

Neka je sada φ atomska formula $R(x_1, ...x_k)$. U tom slučaju ako je pojavljivanje od R pozitivno možemo smatrati da ispred njega nema znaka negacije. Tada je $F_\varphi(X) = X$ a $F_\varphi(Y) = Y$ pa $F_\varphi(X) \subseteq F_\varphi(Y)$, tj. monotonost važi.

Ako je $\varphi = \varphi_1 \wedge \varphi_2$. Tada i φ_1 i φ_2 moraju biti pozitivni u R, pa su F_{φ_1} i F_{φ_2} monotoni. Uzmimo najopštiji slučaj, da je $\varphi = \varphi(R,\vec{x})$, gde je $\vec{x} = (\vec{x_1}, \vec{x_2}, \vec{x_3})$ takvo da je $\varphi_1 = \varphi_1(R, \vec{x_1}, \vec{x_2}), \varphi_2 = \varphi_2(R, \vec{x_2}, \vec{x_3})$. Neka $(a_1, a_2, a_3) \in F_\varphi(X)$. To znači da $\mathfrak{A} \models \varphi_1(X, \vec{a_1}, \vec{a_2})$ i $\mathfrak{A} \models \varphi_2(X, \vec{a_2}, \vec{a_3})$ pa sledi i da važi $\mathfrak{A} \models \varphi_1(Y, \vec{a_1}, \vec{a_2})$ i $\mathfrak{A} \models \varphi_2(Y, \vec{a_2}, \vec{a_3})$ a odavde i $\mathfrak{A} \models \varphi(Y, \vec{a_1}, \vec{a_2}, \vec{a_3})$, pa je F_φ monoton operator.

Potrebno je još pokazati da tvrdjenje važi za $\varphi = \neg\varphi_1$, no iz pretpostavke sledi da su tada sva pojavljivanja R u φ_1 negativna, tj. sva pojavljivanja od $\neg R$ pozitivna (ako se u formuli ne pojavljuje eksplicitno $\neg R$ tada R možemo zapisati kao $\neg\neg R$) te je funkcija $F_{\varphi_1(\neg R, x)}$ monotona pa je i funkcija $F_{\varphi_1(R,x)}$ (ovo se takođe može pokazati indukcijom) monotona - u opadajućem smislu - pa je i $F_{\neg\varphi_1(R,x)}$ monotona. □

Sada možemo definisati i LFP logiku:

Definicija 3.3. *LFP logika predstavlja logiku prvog reda prošerenu tako da je u njoj formula (sa slobodnim promeljivima u $\vec{t}$) i:*

$$[lfp_{R,\vec{x}}\varphi(R,\vec{x})](\vec{t}),$$

gde je R k-arni relacijski simbol, $\varphi(R,\vec{x})$ formula pozitivna u R, $\vec{t}$ uredjena k-torka termi a $|\vec{x}| = k$, sa značenjem:

$$\mathfrak{A} \models [lfp_{R,\vec{x}}\varphi(R,\vec{x})](\vec{a}) \textit{ akko } \vec{a} \in lfp(F_\varphi)$$

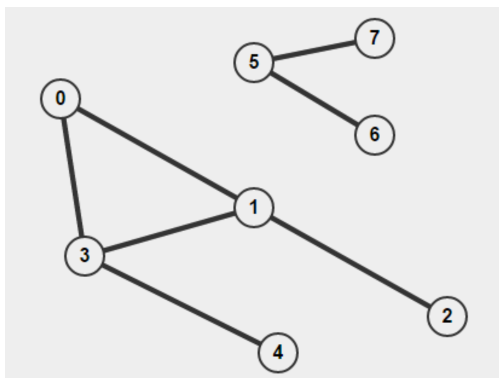
Pozabavimo se sada sa nekoliko primera upita koji se mogu definisati u logici fiksne tačke.

Primer 6. Posmatrajmo prvo rečenicu:

$$\varphi(R, x, y) = E(x,y) \vee \exists z(E(x,z) \wedge R(z,y))$$

gde je E binarna relacija i pokušajmo da odgovorimo na pitanje šta je definisano formulom $[lfp_{R,x,y}\varphi(R,x,y)](u,v)$. U ovom slučaju $F_{\varphi}(X) = E \cup (E \circ X)$ (gde $E \circ X = \{(x,z) \mid \text{postoji } y \text{ takvo da } (x,y) \in E,\ (y,z) \in X\}$). Kada ovaj monotoni operator sukcesivno primenjujemo počevši od praznog skupa dobijemo niz: $\emptyset, E, E \cup E^2, E \cup E^2 \cup E^3 ...$ koji će dostići najmanju fiksnu tačku za neko n u obliku $E \cup E^2 \cup ... \cup E^n$, što predstavlja tranzitivno zatvorenje od E. Dakle formula $[lfp_{R,x,y}\varphi(R,x,y)](u,v)$ definiše tranzitivno zatvorenje od E (svaki element je u relaciji sa svakim elementom do kojeg može "doći" bilo kojim putem putujući relacijom E). Odavde vidimo da je npr. povezanost grafa LFP-definabilna (rečenicom $\forall u \forall v [lfp_{R,x,y}\varphi(R,x,y)](u,v)$).

Prikažimo ovo na primeru sledećeg grafa datog na narednoj slici:



Za ovaj graf važi:

- $X^0 = \emptyset$,
- $X^1 = \{(0,1),(0,3),(1,2),(1,3),(3,4),(5,7),(5,6),(1,0),(3,0),(2,1),$ $(3,1),(4,3),(7,5),(6,5)\}$, dakle u relaciji X^1 se nalaze svi elementi koji su međusobno udaljeni jednu ivicu, tj. povezanani su njome;
- $X^2 = X^1 \cup \{(0,4),(0,2),(1,4),(2,3)(6,7),(4,0),(2,0),(4,1),(3,2)(7,6)\}$, na X^1 su pridodati još i elementi na međusobnom rastojanju od 2 ivice;
- $X^3 = X^2 = X^{\infty} = lfp_{R,x,y}\varphi(R,x,y)$, ne postoje elementi koji su povezani na udaljenisti većoj od 2 ivice, a da se se među njima ne postoji kraći put, i u ovom koraku se naš iterativni proces zaustavlja i $X^2 = X^{\infty}$ predstavlja relaciju tranzitivnog zatvorenja za dati graf.

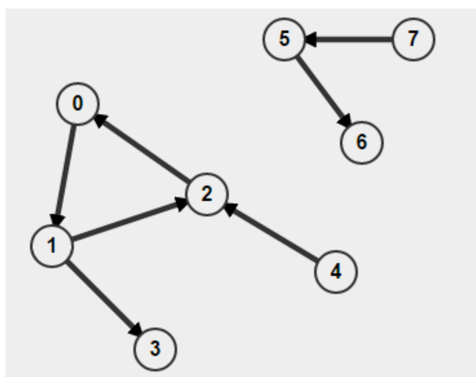
Vidimo da, recimo, $(1,5) \notin lfp_{R,x,y}\varphi(R,x,y)$ te graf nije povezan. $\triangle$

Primer 7. Razmotrimo sada formulu :

$$\varphi(R,x) \equiv \forall y (E(y,x) \to R(y))$$

Formula je pozitivna u R pa indukuje monotonu funkciju F_φ. Kada nju primenimo na prazan skup, dakle F(∅), dobijemo skup - u kontekstu teorije grafova - svih čvorova za koje važi da u njih ne dolazi ni jedna ivica, odnosno čvorova čije je ulazni stepen jednak 0. Kada se na njih primeni F_φ, dobijemo skup čvorova u koje dolaze ivice samo iz čvorova iz skupa F(∅), dakle svi putevi koji se u njima završavaju su dužine maksimalno 1. F(F(F(∅))) su čvorovi u kojima se mogu završiti putevi maksimalne dužine 2 itd. Formula $[lfp_{R,x}\varphi(R,x)](u)$ definiše sve čvorove koji su poslednji na otvorenim putanjama, do kojih je moguće doći polazeći od čvorova u koje ne ulazi ni jedna ivica i ne praveći usput krugove. Odavde je jasno da se pitanjem da li to važi za sve čvorove može saznati da li je graf acikličan, drugim rečima acikličnost grafa je LFT-definabilna (rečenicom $\forall u[lfp_{R,x}\varphi(R,x)](u)$).

Razmotrimo i ovde kako stvari funkcionišu na grafu sličnom onom iz prethodnog primera, samo sa nesimetričnom relacijom E (ako je E simetrična problem je trivijalan: $X^0 = X^1 = X^\infty$):



Za ovaj graf važi:

- $X^0 = \emptyset$,
- $X^1 = \{4, 7\}$
- $X^2 = \{4, 7, 5\}$
- $X^3 = \{4, 7, 5, 6\}$
- $X^4 = \{4, 7, 5, 6\} = X^3 = X^\infty$

Ovde vidimo da, npr. $2 \notin lfp_{R,x}\varphi(R,x)$ te je graf cikličan. $\triangle$

3.3 Osobine logike fiksne tačke

Uvešćemo ovde pojam simultanih fiksnih tačaka, koji će nam omogućiti kompaktiniji i efektniji rad i izvodjenje osobina fiksnih tačaka.

Neka je σ relacioni jezik i neka su $R_1, ..., R_n$ relacijski simboli arnosti redom $k_1, ...k_n$ koji ne pripadaju jeziku σ i neka su $\vec{x_i}$ uređjene k_i-torke promenljivih. Neka je nadalje Φ kolekcija formula $\varphi_1(R_1, ...R_n, \vec{x_1})...\varphi_n(R_1, ...R_n, \vec{x_n})$ formule jezika $\sigma \cup \{R_1, ...R_n\}$, koje su pozitivne u $R_1, ...R_n$. Tada svako φ_i indukuje operator:

$$F_i : P(A^{k_1}) \times ... \times P(A^{k_n}) \to P(A^{k_i})$$

koji radi na sledeći način:

$$F_i(X_1....X_n) = \{(a_1, ...a_{k_i}) \mid \mathfrak{A} \models \varphi_i(X_1/R_1, ..., X_n/R_n, a_1, ..., a_{k_i})\}$$

Ako sve operatore F_i strpamo u jedan dobijemo:

$$\vec{F} : P(A^{k_1}) \times ... \times P(A^{k_n}) \to P(A^{k_1}) \times ... \times P(A^{k_n})$$

koji je, očekivano, definisan na sledeći način:

$$\vec{F}(X_1....X_n) = (F_1(X_1....X_n), F_2(X_1....X_n), ...F_n(X_1....X_n)).$$

Za ovaj operator možemo slično definisati fiksnu tačku, da to bude skup $(X_1, ...X_n)$ za koji važi $\vec{F}(X_1, ...X_n) = (X_1, ...X_n)$, a $(X_1, ...X_n)$ da zovemo najmanjom fiksno tačkom ako za svaku drugu fiksnu tačku $(Y_1, ...Y_n)$ važi da je $X_j \subseteq Y_j$ za svako j. Sada možemo napraviti niz poput onoga koji smo pravili u prethodnom pasusu:

$$X^0 = (\emptyset, \emptyset, ...\emptyset),\ X^1 = \vec{F}(X^0)...,\ X^\infty = \bigcup_{n \in \mathbb{N}} X^n$$

I ovde se može slično pokazati da je X^∞ najmanja fiksna tačka operatora $\vec{F}$. Možemo još dodati da u logici bude i formula:

$$[lfp_{R_i,\phi}](\vec{t})$$

sa semantikom:

$$\mathfrak{A} \models [lfp_{R_i,\Phi}](\vec{a}) \text{ akko } \vec{a} \in X_i^\infty$$

Logiku koju dobijemo na ovaj način ćemo označavati LFP^{simult}.

Primer 8. Pokažimo sada jedan primer korišćenja ove vrste logike u izražavanju važnih osobina grafova. Posmatrajmo graf (konačnu strukturu nad jezikom $\sigma = \{E\}$, gde je E binarna relacija). Neka su R, P i N tri dodatne relacije, prva arnosti 3 a druge dve arnosti 2 i neka je Φ sledeći skup formula:

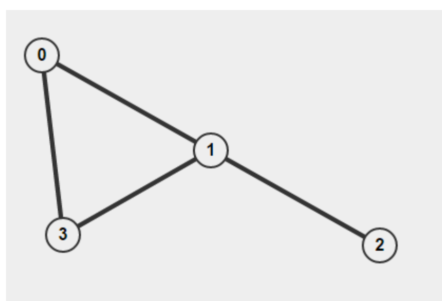
$$\varphi_1(R, P, N, x, y, z) = (E(x, y) \wedge \neg(x = z) \wedge \neg(y = z)) \vee \exists u(E(x, u) \wedge R(u, y, z) \wedge \neg(x = z))$$

$$\varphi_2(R,N,P,x,y) = E(x,y) \ \lor \ \exists u(P(x,u) \land E(u,y) \ \land \ R(x,u,y))$$

$$\varphi_3(R,N,P,x,y) = \exists u(N(x,u) \land E(u,y) \ \land \ R(x,u,y))$$

Operator $\vec{F}$ kreće od argumenta $X^0 = (\emptyset, \emptyset, \emptyset)$, i u svakom koraku dodaje na skup X_1 elemente koji su za po jednu ivicu više udaljeni međusobno, tako da su u k-tom koraku u X_1^k sve trojke (a,b,c) takve da postoji prosta putanja od a do b maksimalne dužine k ivica koja ne prolazi kroz element c. Ako je k neparno ovaj operator na skup X_2 dodaje sve elemente koji su na udaljenosti k, tako da je skup X_2^k skup svih parova (a,b) takvih da postoji prosta putanja između a i b *neparne* dužine maksimalno k, dok skup X_3^k ostaje jednak skupu X_3^{k-1}. Ako je, pak k paran, tada operato $\vec{F}$ na skup X_3 dodaje sve elemente koji su na udaljenosti k, tako da je skup X_3^k skup svih parova (a,b) takvih da postoji prosta putanja između a i b *parne* dužine maksimalno k, dok skup X_2^k ostaje jednak skupu X_2^{k-1}. Ovaj proces će se zaustaviti kada više ne preostane elemenata koji bi mogli da se dodaju u neki od ovih skupova. Kada se to desi, posle nekih n koraka jasno je da tada skupovi X_1^n, X_2^n, X_3^n redom predstavljaju skup svih trojki čvorova (a,b,c) takve da postoji prosta putanja od a do b koja ne prolazi kroz element c, skupo svih čvorova takvih da među njima postoji prosta putanja neparne dužine i skupo svih čvorova takvih da među njima postoji prosta putanja parne dužine.

Prođimo kroz ovaj proces na primeru grafa sa naredne slike:



1) $X^0 = (\emptyset, \emptyset, \emptyset)$

2) $X^1 = (X_1^1, X_2^1, X_3^1)$, gde su:

- $X_1^1 = \{(0,1,2), (1,0,2), (0,1,3), (1,0,3), (0,3,2), (3,0,2), (0,3,1), (3,0,1), (3,1,2), (1,3,2), (3,1,0), ($ $(1,2,3), (2,1,3), (1,2,0), (2,1,0)\}$. U ovom skupu su sve trojke (a,b,c), gde su a i b čvorovi direktno povezani ivicom a c je čvor različit i od a i od b.

- $X_2^1 = \{(0,1),(0,3),(1,2),(1,3),(1,0),(3,0),(2,1),(3,1)\}$. U ovom skupu su svi susedni čvorovi (na međusobnom rastojanju od jedne ivice).
- $X_3^1 = \emptyset$

3) $X^2 = (X_1^2, X_2^2, X_3^2)$, gde su:

   - $X_1^2 = X_1^1 \cup \{(0,1,2),(1,0,2),(0,3,2),(3,0,2),(3,1,2),(1,3,2),$ $(0,2,3),(2,0,3),(3,2,0),(2,3,0)\}$. Na X_1^1 su još dodate trojke (a,b,c) takve da se od a do b može stići iz 2 koraka, ne prelazeći preko elementa c.
   - $X_2^2 = X_2^1$
   - $X_3^2 = \{(0,2),(2,0),(3,2),(2,3),(0,3),(3,0),(0,1),$ $(1,0)(1,3),(3,1)\}$. U ovom skupu su svi parovi elemenata (a,b) takvi da se od a do b mov ze doći iz 2 koraka

4) $X^3 = (X_1^3, X_2^3, X_3^3)$, gde su:

   - $X_1^3 = X_1^2$;
   - $X_2^3 = X_2^2 \cup \{(0,2),(2,0)\}$. Ovde su dodati paravi (a,b) takvi da se od a do b može doći u 3 koraka
   - $X_3^3 = X_3^2$

5) $X^4 = (X_1^4, X_2^4, X_3^4)$, gde su:

   - $X_1^4 = X_1^3 = X_1^\infty$
   - $X_2^4 = X_2^3 = X_2^\infty$
   - $X^4 31 = X_3^3 = X_3^\infty$

$\triangle$

Sada se prirodno postavlja pitanje da li sa ovom logikom dobijamo nešto više u pogledu ekspresivnosti u odnosu na obične logike fiksne tačke. Odgovor daje sledeća teorema:

Teorema 3.2. $LFP^{simult} = LFP$

Dokaz. Dokaz ćemo izvesti za slučaj kada postoje dve formule $\varphi_1(R,S,\vec{x})$ i $\varphi_2(R,S,\vec{y})$, a on se lako može uopštiti. Neka su, prvo, F_1 i F_2 dva monotona operatora, koji su, za skupove U i V, definisani na sledeći način:

$$F_1 : P(U) \times P(V) \to P(U)$$

$$F_2 : P(U) \times P(V) \to P(V)$$

Za $Y \subset U$ one indukuju funkcije:

$$F_2^Y : P(V) \to P(V), \quad F_2^Y(Z) = F_2(Y,Z)$$

$$G_1 : P(U) \to P(U) \quad G_1(X) = F_1(X, lfp(F_2^X))$$

Za ovu situaciju važi sledeća lema

Lema 3.3. *(Bekic-ev princip) Neka je X_1^∞ prva koordinata fiksne tačke operatora (F_1, F_2), zadatog kako je opisano u prethodnom tekstu. Tada važi $X_1^\infty = lfp(G_1)$*

Za dokaz da je $X_1^\infty \subseteq lfp(G_1)$ dokazaćemo da za svako i važi $X_1^i \subseteq lfp(G_1)$ i u tom cilju ćemo iskoristiti indukciju. Za $F_1^0 = \emptyset$ je jasno da ovo važi i pretpostavimo da važi za sve X_1^i za i-ove manje od nekog n. Takođe pretpostavimo da do istog broja važi i $X_2^i \subseteq lfp(F_2^{lfp(G_1)})$. Tada imamo:

$$X_2^n = F_2(X_1^{n-1}, X_2^{n-1}) \subseteq F_2(lfp(G_1), lfp(F_2^{lfp(G_1)}) = F_2^{lfp(G_1)}(lfp(F_2^{lfp(G_1)})) = lfp(F_2^{lfp(G_1)})$$

$$X_1^n = F_1(X_1^{n-1}, X_2^{n-1}) \subseteq F_1(lfp(G_1), lfp(F_2^{lfp(G_1)}) = G_1(lfp(G_1)) = lfp(G_1)$$

što kompletira ovaj deo dokaza.

Još je potrebno dokazati suprotan smer, $lfp(G_1) \subseteq X_1^\infty$. Primetimo prvo da važi $F_2^{X_1^\infty}(X_2^\infty) = F_2(X_1^\infty, X_2^\infty) = X_2^\infty$ pa važi $lfp(F_2^{X_1^\infty}) \subseteq X_2^\infty$. Odatle imamo:

$$G_1(X_1^\infty) = F_1(X_1^\infty, lfp(F_2^{X_1^\infty})) \subseteq F_1(X_1^\infty, X_2^\infty) = X_1^\infty.$$

Kako je $lfp(G_1)$ presek svih skupova A za koje važi $G_1(A) \subseteq A$ a upravo smo pokazali da i X_1^∞ jeste jedan takava skup to važi $lfp(G_1) \subseteq X_1^\infty$.

Ovim smo kompletirali dokaz Bekicevog principa i vratimo se na dokazivanje naše teoreme. Po definiciji operatora G_1 sledi da je $[lfp_{R,\Phi}](\vec{t})]$ ekvivalentno formuli:

$$[lfp_{R,\vec{x}}\varphi_1(R, lfp_{S,\vec{y}}\varphi_2(R, S, \vec{y})]/S, \vec{x})](\vec{t})$$

pošto ova formula definiše lft(G_1) a to je po Bekicevom principu jednako X_1^∞, skupu koji je definisan upravo formulom $[lfp_{R,\Phi}](\vec{t})]$. Dakle formulu logike LFP^{simult} smo zapisali kao formulu logike LFP. Slično možemo postupiti i sa drugom formulom logike LFP^{simult}, tako što bismo sada uzeli $Y \subseteq V$ i definisali

$$F_1^Y : P(U) \to P(U), \quad F_1^Y(X) = F_1(X, Y)$$

$$G_2 : P(V) \to P(V) \quad G_2(Z) = F_2(lfp(F_1^Y), Z)$$

a onda dokazavši da je $X_2^\infty = lfp(G_2)$ dokazali da je $[lfp_{S,\Phi}](\vec{t})$ ekvivalentno sa:

$$[lfp_{S,\vec{y}}\varphi_2([lfp_{R,\vec{x}}\varphi_1(R, S, \vec{x})]/R, S, \vec{y})](\vec{t}).$$

□

Za sada smo definisali LFP^{simult} pa se logično nameće pitanje definisanja simultanih logika inflacione i parcijalne fiksne tačke, što možemo uraditi slično sa definisanjem LFP^{simult}:

Neka je zadat Φ kao kolekcija formula $\varphi_1(R_1, ...R_n, \vec{x_1})...\varphi_n(R_1, ...R_n, \vec{x_n})$ formule jezika $\sigma \cup \{R_1, ...R_n\}$. On generiše formule F_i:

$$F_i : P(A^{k_1}) \times ... \times P(A^{k_n}) \to P(A^{k_i}),$$

$$F_i(X_1....X_n) = X_i \cup \{(a_1, ...a_{k_i}) \mid \mathfrak{A} \models \varphi_i(X_1/R_1, ..., X_n/R_n, a_1, ..., a_{k_i})\}$$

$$\vec{F} = (F_1, ..., F_n)$$

Kao i do sada napravimo niz $X^0 = (\emptyset, \emptyset, ...\emptyset)$, $X^1 = \vec{F}(X^0)...$, $X^\infty = \bigcup_{n\in\mathbb{N}} X^n$

Sada u logiku možemo dodati formulu $[ifp_{R_i,\Phi}](\vec{t})$ za koju važi $\mathfrak{A} \models [lfp_{R_i,\Phi}](\vec{a})$ akko $\vec{a} \in lfp(F_i)$. Ovako dobijenu logiku označavamo IFP^{simult}.

Neka su sada, za istu kolekciju Φ zadate funkcije:

F_i:

$$F_i : P(A^{k_1}) \times ... \times P(A^{k_n}) \rightarrow P(A^{k_i}),$$

$$F_i(X_1....X_n) = \{(a_1, ...a_{k_i}) \mid \mathfrak{A} \models \varphi_i(X_1/R_1, ..., X_n/R_n, a_1, ..., a_{k_i})\}$$

$$\vec{F} = (F_1, ..., F_n),$$

a onda i niz $X^0 = (\emptyset, \emptyset, ...\emptyset)$, $X^1 = \vec{F}(X^0)...$, $X^\infty = (X_1^\infty, ..., X_n^\infty)$ gde je $X_i^\infty = \bigcup_{n\in\mathbb{N}} X_i^n$ za one i za koje ova unija postoja, inače je $X_i^\infty = \emptyset$. Sada je logika PFP^{simult} dobijena dodavanjem formula $[ifp_{R_i,\Phi}](\vec{t})$ za koju važi $\mathfrak{A} \models [lfp_{R_i,\Phi}](\vec{a})$ akko $\vec{a} \in lfp(F_i)$.

Logike IFP^{simult} i PFP^{simult} takođe uvodimo radi kompaktnije i lakše manipulacija jer ni njima ne dobijamo ništa novo u pogledu ekspresivnosti logike, odnosno važi:

Teorema 3.3. *$IFP^{simult} = IFP$ i $PFP^{simult} = PFP$.*

Dokaz. Daćemo dokaz za slučaj sistema Φ od dve formule $\varphi_1(R, S, \vec{x})$ i $\varphi_2(R, S, \vec{x})$, gde su R i S relacijski simboli arnosti $k = |\vec{x}|$. Pretpostavimo još i da se jezik σ sastoji od dva simbola konstanti (c_1 i c_2) i da struktura ima barem dva elementa.

Neka je sada T relacijski simbol arnosti $k + 1$ i neka je $\psi(T, u, \vec{x})$ formula:

$$((u = c_1) \wedge \varphi_1(T(c_1, \vec{z})/R(\vec{z}), T(c_2, \vec{z})/S(\vec{z}), \vec{x}))$$

$$\vee ((u = c_2) \wedge \varphi_2(T(c_1, \vec{z})/R(\vec{z}), T(c_2, \vec{z})/S(\vec{z}), \vec{x}))\,,$$

gde $T(c_1, \vec{z})/R(\vec{z})$ označava da je u formuli svaka pojava od $R(\vec{z})$ zamenjena sa $T(c_1, \vec{z})$. Tada inflaciona i parcijanla fiksna tačka ove formule izračunavaju redom simultanu inflacionu i parcijalnu fiksnu tačku od Φ, jer ona simultana fiksna tačka koja odgovara relaciji $R(S)$ je zapravo skup svih vektora koji za prvu koordinatu imaju $c_1(c_2)$.

$\square$

Vratimo se sada običnoj logici fiksne tačke. Neka je data formula $\varphi(R, \vec{x})$, koja je pozitivna u R. Skupove $X^1 = F_\varphi(\{O\}), X^2 = F_\varphi(X^1), ...$ zovemo nivoi najmanje fiksne tačke. Svaki od njih je LFP-definabilan:

- X^0 formulom $\neg(x = x)$ ...
- X^i formulom $\varphi_i(x) = \varphi(\varphi_{i-1}/R, x)$

Ako $\varphi(R, \vec{x})$ nije pozitivna u R možemo definisati nivoe ifp:

- X^0 formulom $\neg(x = x)$ ...
- X^i formulom $\varphi_i(x) = \varphi(\varphi_{i-1}/R, x) \vee \varphi_{i-1}(x)$

Za formulu φ ćemo uvesti binarnu relaciju na A^k, koja će nam služiti za upoređivanje na kojem stadijumu niza $\emptyset, F_\varphi(\emptyset), ...$ se neki element nalazi i pokazaćemo da je ta relacija lfp-definbilna. Neka je data formula $\varphi(R, \vec{x})$ koja je pozitivna u R. Na datoj strukturi $\mathfrak{A}$ definišemo:

$$|\varphi|^{\mathfrak{A}} = min\{i \mid X^i = X^\infty\}$$

$$|\vec{a}|_\varphi^{\mathfrak{A}} = \begin{cases} min\{i \mid a \in X^i\} & \text{za } a \in X^\infty \\ |\varphi|^{\mathfrak{A}} + 1 & \text{za } a \notin X^\infty \end{cases}$$

Uvedima sada relaciju $\prec^\varphi$ i $\preceq^\varphi$ na A^k (k je arnost od R):

$$\vec{a} \prec^\varphi \vec{b} \text{ akko } |\vec{a}|_\varphi^{\mathfrak{A}} < \left|\vec{b}\right|_\varphi^{\mathfrak{A}}$$

$$\vec{a} \preceq^\varphi \vec{b} \text{ akko } |\vec{a}|_\varphi^{\mathfrak{A}} \leq \left|\vec{b}\right|_\varphi^{\mathfrak{A}} \text{ i } |\vec{a}|_\varphi^{\mathfrak{A}} \leq |\varphi|^{\mathfrak{A}}$$

Razmotrimo opet primer grafa $G = (E, V)$ i formule:

$$\varphi(R, \vec{x}, \vec{y}) = E(x, y) \vee \exists z(E(x, z) \wedge R(z, y))$$

Ona indukuje operator čija najmanja fiksna tačka definiše relaciju tranzitivnog zatvorenja na grafu, kako smo videli u prethodno odeljku. Ali primetimo da za datu formulu relacija $\preceq^\varphi$ definiše upit "udaljenosti", tj. $(a, a') \preceq^\varphi (b, b')$ ako i samo ako postoji putanja od a do a' i najkraća putanja između ove dve tačka je kraća od najkraće putanje između b i b', ili ova ne postoji.

Teorema 3.4. *Za LFP-formulu $\varphi(R, \vec{x})$ relacije $\prec^\varphi$ i $\preceq^\varphi$ (kao i njihove negacije) su LFP-definabilne*

Dokaz. Definišimo još relaciju:

$$\vec{x} \prec_1^\varphi \vec{y} \text{ akko } |\vec{x}|_\varphi^{\mathfrak{A}} + 1 = |\vec{y}|_\varphi^{\mathfrak{A}}$$

Ideja je da definišemo relacije $\prec, \preceq, \prec_1, \not\prec, \not\preceq$ tako da nemamo problema sa negativnim pojavljivanjem ovih relacija (odnosno relacije R) u formuli a zatim, kada imamo fiksnu tačku $(\prec, \preceq, \prec_1, \not\prec, \not\preceq)$ da pokažemo $LFP^{simult}(\prec, \preceq, \prec_1^\varphi, \not\prec, \not\preceq)$=$(\prec^\varphi, \preceq^\varphi, \prec_1^\varphi, \not\prec^\varphi, \not\preceq^\varphi)$.

Da bismo obezbedili pozitivno javljanje relacija u formuli koristićemo sledeće oznake:

- $\varphi(\prec (\vec{y})/R, \vec{x})$ će označavati formulu koja se dobije kada se u formuli φ svako pozitivno pojavljinje relacije $R(\vec{x})$ zameni relacijom $\vec{x} \prec \vec{y}$, a svako negativno pojavljinje relacije R relacijom $\neg\vec{x} \not\prec \vec{y}$. Analogno važi ako umesto $\prec$ u formuli stoji $\preceq$.

- $\varphi(\neg \not\prec (\vec{y})/R, \vec{x})$ će označavati formulu koja se dobije kada se u formuli φ svako pozitivno pojavljinje relacije R zameni sa $\neg\vec{x} \not\prec \vec{y}$, a svako negativno pojavljinje od R sa $(\vec{x} \prec \vec{y})$. Analogno važi ako umesto $\not\prec$ u formuli stoji $\not\preceq$.

Definisaćemo ih na sledeći način:

1) $\psi_1(\prec, \preceq, \prec_1^{\varphi}, \not\prec, \not\preceq, \vec{x}, \vec{y}) \equiv \ \exists z(x \preceq z \prec_1 y)$;

2) $\psi_2(\prec, \preceq, \prec_1^{\varphi}, \not\prec, \not\preceq, \vec{x}, \vec{y}) \equiv \ \varphi(\prec (y)/R, \vec{x})$, u značenju $x \in X^{|y|_{\varphi}^{\mathfrak{A}}}$;

3) $\psi_3(\prec, \preceq, \prec_1^{\varphi}, \not\prec, \not\preceq, \vec{x}, \vec{y}) \equiv \ \varphi(\prec (x)/R, \vec{x}) \wedge \neg\varphi(\neg \not\prec (\vec{x})/R, \vec{y})$ $\wedge \ [\varphi(\preceq (x)/R, \vec{y}) \ \vee \ \forall \vec{z}(\varphi(\preceq (\vec{x})/R, \vec{z}) \to \varphi(\prec (x)/R, \vec{z}))]$. Ovde prva konjunkcija govori da $\vec{x} \in X^{\infty}$, druga da $\vec{y} \notin X^{|x|_{\varphi}^{\mathfrak{A}}}$, a treća da ili $\vec{y} \in X^{|x|_{\varphi}^{\mathfrak{A}}+1}$ ili da je $X^{|x|_{\varphi}^{\mathfrak{A}}} = X^{\infty}$;

4) $\psi_4(\prec, \preceq, \prec_1^{\varphi}, \not\prec, \not\preceq, \vec{x}, \vec{y}) \equiv \ \exists \vec{z}(\vec{x} \not\preceq \vec{z} \prec_1 \vec{y}) \vee \varphi(\emptyset/R, \vec{y}) \vee \forall \vec{z} \neg\varphi(\emptyset/R, \vec{z}))$,gde su poslednje dve disjunkcije u značenju "$y \in X^1$" ili $X^{\infty} = \emptyset$;

5) $\psi_5(\prec, \preceq, \prec_1^{\varphi}, \not\prec, \not\preceq, \vec{x}, \vec{y}) \equiv \ \neg\varphi(\neg \not\prec (\vec{y})/R, \vec{x})$, u značenju "$x \notin X^{|y|_{\varphi}^{\mathfrak{A}}}$";

Označimo sa Ψ ovaj sistem formula. Sva javljanja relacija $\prec, \preceq, \prec_1, \not\prec, \not\preceq$ u Φ su pozitivna. Mi tvrdimo da simultana lfp od Φ definiše $\{\prec^{\varphi}, \preceq^{\varphi}, \prec_1^{\varphi}, \not\prec^{\varphi}, \not\preceq^{\varphi}$ $\}$. Tačnije rečeno potrebno je dokazati da za svako $S_i \in \{\prec, \preceq, \prec_1, \not\prec, \not\preceq\}$ važi $(\vec{a}, \vec{b}) \in lfp_{S_i, \Psi}$ ako i samo ako $\vec{a} S_i^{\varphi} \vec{b}$, gde je S_i^{φ} odgovorajuča relacija iz skupa $\{\prec^{\varphi}, \preceq^{\varphi}, \prec_1^{\varphi}, \not\prec^{\varphi}, \not\preceq^{\varphi}\}$

Implikaciju ćemo dokazati pokazujući da 5-orka relacija $(\prec^{\varphi}, \preceq^{\varphi}, \prec_1^{\varphi}, \not\prec^{\varphi}, \not\preceq^{\varphi}$ $)$ jeste jedna fiksna tačka operatora F_{Ψ}, odnosno da zadovoljava date formule. Za relaciju $\prec^{\varphi}$ stvari su jasne. Za relaciju $\preceq^{\varphi}$ takođe, pošto $a \preceq^{\varphi} b$ akko $\mathfrak{A} \models \varphi(\prec^{\varphi} (\vec{b})/R, \vec{a})$, jer je skup $\{\vec{x} \mid \vec{x} \prec^{\varphi} b\}$ upravo skup X^{k-1} za ono k za koje važi $b \in X^k - X^{k-1}$ .Pretpostavimo da za neke a i b važi $\vec{a} \prec_1^{\varphi} \vec{b}$. Ako je $|\vec{a}|_{\varphi} < |\varphi|$ tada nam treća konjunkcija (u kojoj tada važi prva disjunkcija, $\varphi(\preceq (\vec{a})/R, \vec{\vec{b}})$) iz 3) daje da $\vec{b} \in X^{|\vec{a}|+1}$ a prve dve konjunkcije su očigledno tačne (prva kaže da $\vec{a} \in X^j$, za neko j, a druga da $\vec{b} \notin X^{|\vec{a}|^{\varphi}}$). Ako je $|\vec{a}|^{\varphi} = |\varphi|$ tada iz treće konjunkcije važi druga disjunkcija (koja kaže da je $X^{|\vec{a}|_{\varphi}}$ poslednji nivo) a prve dve konjunkcije takođe važe. Obrnuto, ako za ovu 5-orku relaciji neko $\vec{a}$ i $\vec{b}$ zadovoljavaju ψ_3 to znači da $\vec{a} \in X^j$, za neko j, a $\vec{b} \notin X^{|\vec{a}|_{\varphi}}$, po prve dve konjunkcije, a kako važi i treća to imamo još i da je $\vec{b} \in X^{|\vec{a}|+1}$ ili $\vec{a}$ upada u fiksu taču u poslednjem stadijumu rekurzije te $\vec{b}$ nije u fiksnoj tački tj. $\left|\vec{b}\right|_{\varphi}^{\mathfrak{A}} = |\varphi|^{\mathfrak{A}} + 1 = |\vec{a}|_{\varphi}^{\mathfrak{A}} + 1$. 4) i 5) deo su jasni.

Drugi deo dokaza ćemo izvršiti indukcijom. Dokazaćemo da ako su (a, b) u nekoj od relacija $(\prec^{\varphi}, \preceq^{\varphi}, \prec_1^{\varphi}, \not\prec^{\varphi}, \not\preceq^{\varphi})$ da su tada oni u odgovarajućem završnom stadijumu izračunavanja lfp^{simult}, X^i Baza indukcije $|b|_{\varphi}^{\mathfrak{A}} = 1$. U ovom slučaju a i b ne mogu da se nađu u relacijama $\prec^{\varphi}$ i $\prec_1^{\varphi}$. Ako su u relaciji $a \preceq^{\varphi} b$ to može biti samo ako $|a|_{\varphi}^{\mathfrak{A}} = |b|_{\varphi}^{\mathfrak{A}} = 1$, odnosno imamo da $\mathfrak{A} \models \varphi(\emptyset/R, a)$ (i

$\mathfrak{A} \models \varphi(\emptyset/R, b))$, odnosno $(a,b) \in X_2^1$. Ako su $a \not\prec^\varphi b$ tada važi $\mathfrak{A} \models \varphi(\emptyset/R, b)$, tj. $(a,b) \in X_4^1$. Ako $a \not\preceq^\varphi b$ tada $\mathfrak{A} \models \neg\varphi(\{\emptyset\}, a)$ pošto $|a|_\varphi^\mathfrak{A} \neq 1$, tj. $(a,b) \in X_5^1$.

Induktivni korak: Pretpostavimo da teorema važi za sve b takve da $|b|_\varphi^\mathfrak{A} < j$, za neko j. Neka je $|b|_\varphi^\mathfrak{A} = j$.

- Izvedimo najpre dokaz za relaciju $\prec_1^\varphi$. Neka je $a \prec_1^\varphi b$. Kako je $|a|_\varphi^\mathfrak{A} < |b|_\varphi^\mathfrak{A}$ to $|a|_\varphi^\mathfrak{A} \leq |\varphi|^\mathfrak{A}$ pa važi formula $\varphi(\prec (a)/R, a)$. Iz istog razloga važi $\neg\varphi(\neg \not\prec^\varphi (\vec{a})/R, \vec{b})$ pa po indukcijskoj hipotezi važi $\not\prec^\varphi (a) = \not\prec (a)$ te važi i $\neg\varphi(\neg \not\prec (\vec{x})/R, \vec{y})$.

  Još je ostalo da dokažemo treću konjunkciju. Ako je $|b|_\varphi^\mathfrak{A} \leq |\varphi|^\mathfrak{A}$ tada važi $\varphi(\preceq^\varphi (\vec{a}), \vec{b})$ pa po induktivnoj hipotezi važi $\varphi(\preceq (\vec{a}, \vec{b})$. Ako je $|b|_\varphi^\mathfrak{A} = |\varphi|^\mathfrak{A} + 1$ tj. $|a|_\varphi^\mathfrak{A} = |\varphi|^\mathfrak{A}$ pa važi formula $\forall z(\varphi(\neg \not\prec^\varphi (\vec{a}), \vec{z}) \to \varphi(\prec^\varphi (\vec{a}), (\vec{z})))$, pa po I.H. važi formula $\forall z(\varphi(\neg \not\prec (\vec{a}), \vec{z}) \to \varphi(\prec (\vec{a}), \vec{z}))$, te (a,b) pripadaju fiksnoj tački od ψ_3.

- Neka važi $a \prec^\varphi b$. To znači da postoji važi formula $\exists z(a \preceq^\varphi z \prec_1^\varphi b)$, a kako je $|z|_\varphi^\mathfrak{A} < j$ to po induktivnoj hipotizi i prethodno dokazanom imamo da za to z važi $a \preceq z \prec_1 b$

- Neka važi $a \preceq^\varphi b$. Tu imamo dve mogućnosti, da je $a \prec_1^\varphi b$ i da je $a \preceq^\varphi c$ za neko c za koje važi $|c|_\varphi^\mathfrak{A} < j$, a u oba slučaja možemo iskoristiti induktivnu hipotezi i već dokazane slučajeve, te nam važi $a \preceq b$.

- Neka važi $a \not\prec^\varphi b$. Pretpostavimo da $|b|_\varphi^\mathfrak{A} > 1$ i da u X_4^∞ postoji barem jedan element - u suprotnom bi važila druga odnosno treća disjunkcija u formuli ψ_5. Tada postoji c takvo da $c \prec_1^\varphi b$ a za to c važi i $a \not\preceq^\varphi c$ i $|c|_\varphi^\mathfrak{A} < j$ pa po induktivnoj hipotezi i prvodokazanoj stavci sledi zakljucak da je $a \not\prec b$

- Za $a \not\preceq^\varphi b$ je po definiciji jasno da važi $\neg\varphi(\neg \not\prec (\vec{b}), \vec{a})$ pa važi i $a \not\preceq b$.

□

Posledica 3.1. *(Gurevich-Shelah) IFP = LFP.*

Dokaz. Jasno je da je $LFP \subseteq IFP$. Suprotan smer, $IFP \subseteq LFP$, se dokazuje indukcijom po složenosti formule, no jedini slučaj koji treba razmotriti je formula $ifp_{R,\vec{x}}\varphi(R, \vec{x})$. Uzmemo li, bez gubitka opštosti, da φ generiše induktivan operator tada imamo da je formula $[ifp_{R,\vec{x}}\varphi(R, \vec{x})](\vec{t})$ ekvivalentna formuli $\varphi(\prec^\varphi (\vec{t})/R, \vec{t})$, što je, po upravo dokazanom, formula LFP logike. □

3.4 LFP, PFP - veza sa klasama PTIME i PSPACE

U ovom odeljku ćemo pokazati da logike fiksne tačke - nad uređenim strukturama - hvataju neke važne klase složenosti. Uredjena je ona struktura u kojoj je binarni relacijski simbol interpretiran kao linearno uređenje domena.

Teorema 3.5. *(Immerman-Vardi) LFP (a time, po Gurevich-Shelah teoremi i IFP) nad uređenim strukturama hvata PTIME.*

Dokaz. Potrebno je, dakle, dokazati da se za svaku uređenu strukturu $\mathfrak{A}$ i za svaku LFP-rečenicu φ u složenosti PTIME može testirati da li $\mathfrak{A} \models \varphi$, a zatim dokazati i da za svaku osobinu P uređenih struktura koju je moguće testirati u složenosti PTIME postoji rečenica φ jezika LFP takva da $\mathfrak{A} \models \varphi$ akko $\mathfrak{A}$ ima osobinu P.

Prvi deo dokaza izvodimo indukcijom po složenosti formule. Ovde ćemo razmotriti samo slučaj osnovne formule oblika $lfp_{R,\vec{x}}\varphi$, a slučajevi kvantifikatora i veznika se rešavaju kao što smo uradili u dokazu leme 2.1. Neka je $n = |\mathfrak{A}|$. Računanje fiksne tačke se obavlja najviše u n^k iteracija gde je k broj slobodnih promenljivih od φ. Kako je svaku iteraciju moguće izračunati u polinomijalnom vremenu od n (potrebno je isprobati najviše n^k promenljivih u formuli, a kad ih jednom ubacimo onda je proveru da li ta kombinacija zadovoljava formulu moguće uraditi u polinomijalnom vremenu, kako je pokazano u lemi 2.1.), to iz ove činjenice sledi naš rezultat.

Za drugi deo dokaza ćemo iskoristiti tehniku koju smo upoznali kroz dokaz Trakhtenbrotove i Faginove teoreme. Pretpostavimo neku osobinu $\mathcal{P}$ za koju je, za datu linearnu strukturu $\mathfrak{A}$ moguće u PTIME-u odrediti da li je ona poseduje. Naš zadatak sada postaje da konstruišemo LFP-formulu koja će biti zadovoljiva tačno u onim modelima koji imaju tu osobinu. Neka Tjuringova mašina $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_{accept}, Q_{reject})$, sa oznakamo u smislu na koji samo navikli u ovom tekstu, odlučuje da li data struktura poseduje osobinu $\mathcal{P}$. Bez umanjenja opštosti pretpostavimo da je $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \_\}$, kao i da postoji samo jedno prihvatajuće stanje, q_a. Neka M radi u n^k vremenu i neka je to više nego što je potrebno mesta za kodiranje σ strukture sa n elemenata.

Po isprobanom receptu, a budući da ima linearno uređenje na elementima koje indukuje (leksikografsko) uređenje na k-torkama ($\leq_k$) ćemo napraviti predikate $T_0(\vec{p}, \vec{t}), T_1(\vec{p}, \vec{t}), T_2(\vec{p}, \vec{t})$ koji će nam koristita za iskazivanje da se u trenutku $\vec{t}$ na poziciji $\vec{p}$ nalazi upisana redom 0, 1 ili _ (prazan simbol) kao i predikate $(H_q(\vec{p}, \vec{t}))_{q \in Q}$ koji govore da se glava mašine u trenutku $\vec{t}$ nalazi iznad pozicije $\vec{p}$, a u stanju q.

Konstruišimo sada sistem formula Φ čija će simultana lfp biti $(T_0, T_1, T_2, (H_q)_{q \in Q})$. On će se sastojati od formula $\varphi_i(T_0, T_1, T_2, (H_q)_{q \in Q})$, za sve $i \in \{0, 1, 2\}$ i $\varphi_q(T_0, T_1, T_2, (H_q)_{q \in Q})$ za svako $q \in Q$. Neka su ν i ξ iste one formule iz dokaza Faginove teoreme, dakle $\nu(\vec{p})$ je tačno ako se na $\vec{p}$-toj poziciji ulaznog koda nalazi 1 a $\xi(\vec{p})$ važi ako je $\vec{p}$ veće od broja pozicija na ulazu (koda strukture $\mathfrak{A}$). Formule konstruišemo na sledeći način:

$$\varphi_0 \equiv (\vec{t} = 0 \ \wedge \ \neg\nu(\vec{p}) \ \wedge \ \neg\xi(\vec{p})) \vee (\neg(\vec{t} = 0) \ \wedge \ \alpha_0(\vec{t} - 1, \vec{p}, T_0, T_1, (H_q)_{q \in Q})),$$

što je formula koja će definisati relaciju T_0, gde je α_0 formula zadaje uslove pod kojima se iz jednog momenta ($\vec{t}$-1) prelazi u naredni ($\vec{t}$) u kojem će važiti relacija $T_0(\vec{p}, \vec{t})$:

$$\alpha_0 \equiv [T_0(\vec{p}, \vec{t}-1) \wedge \forall q \in Q \neg H_q(\vec{p}, \vec{t-1})] \vee [\exists q, q_1 \in Q \exists x \in l, r H_q(\vec{p}, \vec{t}-1) \wedge \delta(q, p) = (q_1, 0, x)]$$

Formula koja će definisati T_1 ima sličan oblik:

$$\varphi_1 \equiv (\vec{t} = 0 \ \wedge \ \nu(\vec{p}) \ \wedge \ \neg\xi(\vec{p})) \vee (\neg(\vec{t} = 0) \ \wedge \ \alpha_1(\vec{t} - 1, \vec{p}, T_0, T_1, (H_q)_{q \in Q})),$$

gde je α_1 formula, koja uslovljava tranziciju iz jednog u naredni trenutak $\vec{t}$, u kome važi $T_1(\vec{p}, \vec{t})$:

$$\alpha_1 \equiv [T_1(\vec{p}, \vec{t}-1) \wedge \forall q \in Q \neg H_q(\vec{p}, \vec{t}-1)] \vee [\exists q, q_1 \in Q \exists x \in l, r H_q(\vec{p}, \vec{t}-1) \wedge (\delta(q, p) = (q_1, 1, x)]$$

U istom duhu zapisujemo i formulu za relaciju T_2:

$$\varphi_2 \equiv (\vec{t} = 0 \ \wedge \ \neg\nu(\vec{p}) \ \wedge \ \xi(\vec{p})) \vee (\neg(\vec{t} = 0) \ \wedge \ \alpha_2(\vec{t} - 1, \vec{p}, T_0, T_1, (H_q)_{q \in Q})),$$

gde je α_2 formula:

$$\alpha_2 \equiv [T_2(\vec{p}, \vec{t}-1) \wedge \forall q \in Q \neg H_q(\vec{p}, \vec{t}-1)] \vee [\exists q, q_1 \in Q \exists x \in l, r H_q(\vec{p}, \vec{t}-1) \wedge (\delta(q, p) = (q_1, \_, x)].$$

Formula koja će rekurzivno dati relaciju H_{q_0} ima oblik:

$$\varphi_{q_0} \equiv (\vec{t} = 0 \wedge \vec{p} = 0) \vee (\neg(\vec{t} = 0) \ \wedge \ \alpha_{q_0}(\vec{t} - 1, \vec{p}, T_0, T_1, (H_q)_{q \in Q})),$$

gde je α_{q_0} formula:

$$\alpha_{q_0} \equiv \exists q \in Q \exists x \in \Gamma \exists y \in l, r H_q(\vec{p}, \vec{t} - 1) \wedge (\delta(q, p) = (q_0, x, y)$$

Dok za ostale ralcije $H_q \in Q$, koje po prirodi stvari ne mogu važiti u početnom trenutku formule uzimaju donekle drugačiji oblik:

$$\varphi_q \equiv (\vec{t} > 0) \wedge \ \alpha_q(\vec{t} - 1, \vec{p}, T_0, T_1, (H_q)_{q \in Q})),$$

gde je α_q formula:

$$\alpha_q \equiv \exists q_i \in Q \exists x \in \Gamma \exists y \in l, r H_{q_i}(\vec{p}, \vec{t} - 1) \wedge (\delta(q_i, p) = (q, x, y)$$

Za ovako zadato Φ n-ta iteracija u izračunavanju fiksne tačke računa konfiguracije od M za vremena manja od n. Odatle sledi da formula:

$$\exists \vec{p} \exists \vec{t} [ifp_{H_{q_a}, \Phi}](\vec{p}, \vec{t})$$

u stvari testira da li $\mathfrak{A}$ ima osobinu $\mathcal{P}$, a pošto smo videli da je $IFP^{simult} = IFP = LFP$ odatle sledi i rezultat koji dokazujemo. □

Pozabavimio se sada PFP logikom, koja takođe hvata jednu značajnu klasu složenosti:

Teorema 3.6. *PFP nad uređenim strukturama hvata PSPACE*

Dokaz. Prvo valja pokazati mogućnost evaluacije PFP-formule u PSPACE vremenu. Za formulu $pfp_{R,\vec{x}}\varphi(R, \vec{x})$ postoje dve opcije: 1) da za neko l važi $X^l = X^{l+1}$ u kom slučaju se pfp dostiže, ili da takvo k ne postoji, u kom slučaju je pfp(F_φ)=∅, odnosno formula je netačna. Za evaluaciju je potrebno napraviti distinkciju između ova dva slučaja. To možemo uraditi tako što numerišemo sve moguće podskupove od A^k - a to je moguće uraditi u PSPACE složenosti - tih skupova ima 2^{n^k} a za zapisivanje tog broja je potrebno n^k bitova, a zatim računati niz $X^0, X^1, X^2, ...$ i proveravati da li se neki od podskupova iz A^k ponavlja ili se na kraju niz zaustavlja u kojem slučaju smo izračunali $pfp_{R,\vec{x}}\varphi(R, \vec{x})$. Svaki od medjukoraka kojih ima maksimalno 2^{n^k} je moguće obaviti u PSPACE složenosti čime završavamo ovaj deo dokaza.

S druge strane, neka je M PSPACE Tjuringova mašina zadata kao u dokazu prethodne teoreme. Pretpostavimo, bez umanjenja opštosti da M za ulaz dužine n ne zahteva više od n^k mesta na traci, za neko k.

Konstrušimo prvo formulu takvu da će iteracije $F_\varphi^{i+1}(\emptyset)$ predstavljati i-tu konfiguracije od M na ulazu koji predstavlja string reprezentaciju od $\mathfrak{A}$. Sada nećemo morati da eksplicitno vodimo evidenciju vremenskom trenutku, tj. da imamo posebnu promenljivu za njega. Ako je $x = (q, l, i, b)$ vektor čije koordinate redom predstavljaju stanje, poziciju TM-glave, indeks polja na traci i ono šta je upisano u polju, imamo:

$$\varphi(R, q, l, i, b) := (\neg\exists x R(x) \wedge \varphi_0(q, l, i, b)) \vee (\exists x R(x) \wedge \eta(R, q, l, i, b)),$$

gde φ_0 opisuje početno stanje na ulazu koji predstavlja string reprezentaciju od $\mathfrak{A}$. Formula $\eta(R, q, l, i, b)$ daje opis konfiguacije koja neposredno sledi nakon konfiguracije opisane sa R, te dalje iteracije neće biti prazne, a videli smo, iz prethodnog teksta, da su obe FO-formule (i φ_0 i η). Tada formula:

$$\psi_0(x) := PFP_{R,x,}\varphi$$

opisuje završnu konfiguraciju od M na ulazu $Code(\mathfrak{A})$. Ovako definisana relacija nije prazna pošto je M mašina koja u svakom slučaju dovršava svoje izračunavanje (zaustavlja se). Odatle sledi da:

$$\psi := \exists l \exists i \exists b \psi_0(q_{accept}, l, i, b)$$

definiše da mašina prihvata ulaz $Code(\mathfrak{A})$. $\square$

3.5 DATALOG i logike fiksne tačke

Cilj ovog odeljka je predstavljanje jednog jezika koji vuče koren iz teorije baza podataka i ispitivanje njegovih tesnih veza sa logikama fiksne tačke.

Jedan od najčešćih tipova upita koji se javljaju u radu nad relacionim bazama podataka je upit oblika " SELECT...FROM...WHERE A and B and C and...". Oblik koji se javlja iza reči WHERE ima jednostavnu logičku karakterizaciju:

Definicija 3.4. *FO-formula koja se sastoji od egzistencijalnog kvantifikatora i atomskih formula koje su spojene konjunkcijom se zove konjunktivni upit.*

Dakle konjunktivni upit je upit je formula oblika:

$$\psi(\vec{x}) \equiv \exists \vec{y} \bigwedge_i \alpha_i(\vec{x}, \vec{y}),$$

što možemo zapisati i u obliku:

$$R_\psi(\vec{x}) : -\alpha_1(\vec{x},\vec{y}), \alpha_2(\vec{x},\vec{y}), ..., \alpha_m(\vec{x},\vec{y})$$

Ovako na R_ψ možemo gledati kao na dodatni relcioni simbol. Izraze ovog oblika zovemo "pravila", pri čemu je izraz sa leve strane znaka :- "glava"a sa desne "telo"pravila. DATALOG program sadrži nekolicinu ovakvih pravila:

Definicija 3.5. *DATALOG program nad jezikom σ je par (Π, Q), gde je Π skup pravila oblika:*

$$P(\vec{x}) : -\alpha_1(\vec{x},\vec{y}), \alpha_2(\vec{x},\vec{y}), ..., \alpha_m(\vec{x},\vec{y}),$$

gde P nije relacijski simbol jezika σ, dok su $\alpha_i(\vec{x},\vec{y})$ ili atomske formule ($R(x,y)$, $R \in \sigma$) jezika σ ili glave drugih pravila iz Π, dok je Q takođe glava jednog od pravila iz Π, koja je posebno izdvojena i predstavlja rezultat programa. Relacije koje nisu iz σ a koje sačinjavaju glave pravila se zovu "intenzionalni predikati"a oni iz σ "ekstenzionalni predikati".

Jezik $DATALOG_\neg$ je proširenje jezika DATALOG gde se u telu pravila mogu javiti i negacije atomskih formula ($\neg R(x,y)$).

Neka su, sada, $P_1, .., P_k$ svi intezionalni predikati jezika DATALOG, arnosti redom r_i i neka su sva pravila u kojima je glava P_i sledeća (ima ih recimo l):

$$P_i(\vec{x}) : -\alpha_1^1(\vec{x},\vec{y_1}), \alpha_2^1(\vec{x},\vec{y_1}), ..., \alpha_{m_1}^1(\vec{x},\vec{y_1})$$

$$... \quad ... \quad ... \quad ... \quad ... \quad ....$$

$$P_i(\vec{x}) : -\alpha_1^l(\vec{x},\vec{y_l}), \alpha_2^l(\vec{x},\vec{y_l}), ..., \alpha_{m_l}^l(\vec{x},\vec{y_l}),$$

i neka su $Y_1, ...Y_k$ skupovi gde $Y_j \subseteq A^{r_j}$. Definišemo operator $F_\Pi(Y_1, ...Y_k) = (Z_1, ...Z_k)$:

$$Z_i = \{\vec{a} \in A^{r_i} \mid (\mathfrak{A}, Y_1, Y_2, ..., Y_k) \models \bigvee_{j=1}^{l} \exists \vec{y}_j(\alpha_1^j(\vec{a},\vec{y_j})) \wedge ... \wedge \alpha_{m_j}^j(\vec{a},\vec{y_j})\}$$

Ova funkcija se zove operator neposredne posledice, jer svi elementi skupa Z_i mogu biti izvedeni iz nekog od pravila iz Π čija je glava P_i a intenzioni predikati interpretirani kao skupovi $\vec{Y} = (Y_1, ...Y_k)$.

Tipičan primer DATALOG programa je:

$$trcl(x,y) : -E(x,y)$$

$$trcl(x,y) : -E(x,z), trcl(z,y),$$

gde su pravila, dakle, da su x i y u tranzitivnom zatvorenju ako postoji ivica (x,y), ili postoji ivica (x,z) za neko z koje je u tr. zatvorenju sa y. Ovde vidimo primer pravila koje se rekurzivno pojavljuje. On za rezultat ima Q relaciju trcl.

Usled pozitivnosti u svim intezionalnim predikatima, operator F_Π je monoton pa ima najmanju fiksnu tačku (lfp). Dakle krećući od skupa $(\emptyset, \emptyset, , ...\emptyset)$

primenjujući operator F_Π na kraju dolazimo do $(P_1^\infty, ...P_k^\infty)$. Rezultat od (Π, Q) je

$$Q^\infty = \{\vec{a} \mid \mathfrak{A} \models [lfp_{q,\Phi}](\vec{a})\},$$

gde je Φ kolekcija formula:

$$\varphi_i(\vec{x}, P_1, P_2, ...P_k) \equiv \bigvee_j \exists y_j[\alpha_1^j(\vec{x}, \vec{y}) \wedge ... \wedge (\alpha_{m_j}^j(\vec{x}, \vec{y})].$$

dakle programi DATALOG i DATALOG$_\neg$ se mogu izraziti u logici LFP. Oblik formula φ_i nam daje motivaciju za narednu definiciju i teoremu:

Definicija 3.6. *Neka je dat relacioni jezik σ. Logika egzistencijalne najmanje fiksne tačke ($\exists LFP$) je restrikcija LFP-a tako da:*

- *samo atomske formule mogu biti pod dejstvom negacije*
- *univerzalni kvantifikatori nisu dozvoljeni*

Teorema 3.7. $\exists LFP = DATALOG_\neg$

Dokaz. Jedna strana jednakosti, $DATALOG_\neg \subseteq \exists LFP$, je dokazana u prethodnom razmatranju. Za suprotan smer je potrebno dokazati da se svaka formula iz $\exists LFP$ može prevesti u $DATALOG_\neg$ (Π_φ, Q_φ) koji će kao rezultat za svaku strukturu $\mathfrak{A}$ vraćati tačno $\varphi(\mathfrak{A})$. To radimo na sledeći način:

- Ako je $\varphi(\vec{x})$ atomska formula ili negacija atomske formule tada je $\Pi_\varphi = \{Q_\varphi : -\varphi(\vec{x})\}$
- Ako je $\varphi(\vec{x}) = \varphi_1 \wedge \varphi_2$ tada

$$\Pi_\varphi = \Pi_{\varphi_1} \cup \Pi_{\varphi_2} \cup \{Q_\varphi(\vec{x}) : -Q_{\varphi_1}, Q_{\varphi_2}\}$$

- Ako je $\varphi(\vec{x}) = \varphi_1 \vee \varphi_2$ tada

$$\Pi_\varphi = \Pi_{\varphi_1} \cup \Pi_{\varphi_2} \cup \{Q_\varphi(\vec{x}) : -Q_{\varphi_1}, Q_\varphi(\vec{x}) : -Q_{\varphi_2}\}$$

- Ako je $\varphi(\vec{x}) = \exists\vec{y}\varphi_1(\vec{x}, \vec{y})$ tada

$$\Pi_\varphi = \Pi_{\varphi_1} \cup \{Q_\varphi(\vec{x}) : -Q_{\varphi_1}(\vec{x}, \vec{y})\}$$

- Ako je $\varphi(\vec{x}) = [lfp_{R,\vec{y}}\psi(R, \vec{y})](\vec{x})$ tada za već imamo program za ψ a kako je pojavljivanje od R pozitivno u ψ ono ostaje pozitivno i u φ pa nam je:

$$\Pi_\varphi = \Pi_\psi \cup \{R(\vec{y}) : -Q_\psi(\vec{y}), Q_\varphi(\vec{x}) : -R(\vec{x})\},$$

što nam je dovoljno da kompletiramo dokaz. $\square$

Mi smo do sada videli da LFP nad uređenim strukturama hvata PTIME, ali pitanje od najvećeg interesa - i još uvek otvoreno, kao što ćemo videti - da li postoji logika koja hvata PTIME složenost nad svim konačnim strukturama?

Neka su zadate dve $\mathfrak{A}_1$ i $\mathfrak{A}_2$ strukture istog jezika nad istim domenom, s tim što za svako $R \in \sigma$ važi $R^{\mathfrak{A}_1} \subseteq R^{\mathfrak{A}_2}$. Jasno je da tada, za DATALOG jezik (Π, Q) važi da je $(\Pi, Q)(\mathfrak{A}_1) \subseteq (\Pi, Q)(\mathfrak{A}_2)$. Međutim, postoje osobine (upiti) koje su definabilne u PTIME a nisu monotone, pa DATALOG ne hvata PTIME.

Iz Sličnog razloga ni DATALOG$_\neg$ ne uspeva da uhvati PTIME složenost. Naime, ako opet imamo dve strukture $\mathfrak{A}_1$ *i* $\mathfrak{A}_2$ istog jezika ali koje sada ne moraju biti nad istim domenom već važi $A_1 \subseteq A_2$ a takođe za svaki rel. simbol R važi da je $R^{\mathfrak{A}_1}$ restrikcija od $R^{\mathfrak{A}_2}$ na A_1 tada takođe važi $(\Pi, Q)(\mathfrak{A}_1) \subseteq (\Pi, Q)(\mathfrak{A}_2)$. S druge strane postoje PTIME upiti koji zadovoljavaju uslove ali nisu monotoni.

Ovoj situaciji možemo doskočiti tako što obogatimo strukture relacijom sukcesije ("$\underline{\text{succ}}$") i sa dve dodatne konstante za isticanje minimalnog i maksimalnog elementa ($\underline{\text{min}}$, $\underline{\text{max}}$), što će onemogućiti da se steknu uslovi iz prethodnog pasusa, da $\mathfrak{A}_1$ bude substruktura $\mathfrak{A}_2$. U ovoj situaciji važi sledeća.

Teorema 3.8. *DATALOG$_\neg$ hvata PTIME nad strukturama sa koje imaju relaciju sukcesora i konstante za minimalni i maksimalni element.*

Dokaz. Potrebno je dokazati samo jedan smer (drugi je pokazan): ako je $\mathcal{K}$ neka klasa struktura koje zadovoljavaju uslove teoreme a odlučiva je u PTIME-u, onda ona može biti definisana DATALOG$_\neg$ upitom.

Neka je M PTIME Tjuringova mašina sa skupom stanja Q i alfabetom Σ koja prihvata strukture iz $\mathcal{K}$ i radi u složenosti n^k. Konfiguracije od M ćemo predstaviti rečima $\#w_1w_2\cdots w_{i-1}(qw_i)w_{i+1}\cdots w_{m-1}\,\#$ nad radnim alfabetom $\Gamma = \Sigma \cup (Q \times \Sigma) \cup \{\#\}$, gde je $m = n^k$. Akcije mašine M predstavljamo funkcijom $\delta : \Gamma^3 \to \Gamma$, gde je za konfiguraciju $C = c_0c_1\cdots c_m$ naredna konfiguracija $NEXT(C) = c'_0c'_1\cdots c'_m$ određena pravilima:

$$c'_0 = c'_m = \# \;\; i \;\; c'_i = \delta(c_{i-1}, c_i, c_{i+1}), \quad za\ 1 \le i \le m-1$$

Neka je S $2k$-arni relacioni simbol i neka je Π_S DATALOG$_\neg$ program sa glavom S, koji računa sledbenika od uređene k-torke i neka je $\beta_\omega(\vec{x})$ formula takva da $\mathfrak{A} \models \beta_\omega(\vec{p})$ ako i samo ako se na mestu $\vec{p}$ ulazne konfiguracije nalazi simbol ω. Neka je (Π_ω, Q_ω) DATALOG program koji vraća $\beta_\omega(\vec{x})$.

Predstavimo izračunavanja od M vektorom $\vec{C} = (C_\omega)_{\omega\in\Gamma}$ $2k$-arnih relacija:

$$C_\omega = \{(\vec{p}, \vec{t}) \mid \vec{p} - \text{ti simbol konfiguracije u trenutku } \vec{t} \text{ je } \omega\}$$

DATALOG$_\neg$ program koji odgovara TM M se sastoji od:

1) programa Π_S koji definiše relaciju sukcesora na k-torkama

2) programa Π_ω koji opisuje ulazni string

3) pravila:

$$C_\#(\underline{\vec{min}}, \vec{t})$$
$$C_\#(\underline{\vec{max}}, \vec{t})$$
$$C_\omega(\vec{x}, \vec{0}) : -Q_\omega \vec{x}, \text{ za sve } \omega \in \Gamma - \{\#\}$$

4) Za svako $\tau, \upsilon, \phi, \omega$ za koje važi $\delta(\tau, \upsilon, \phi) = \omega$ pravila:

$$C_\omega(\vec{p}, \vec{t'}) : -S(\vec{x}, \vec{p}), S(\vec{p}, \vec{y}), S(\vec{t}, \vec{t'}), C_\tau(\vec{x}, \vec{t}), C_\upsilon(\vec{p}, \vec{t}), C_\phi(\vec{y}, \vec{t})$$

5) pravila:

$$Acc : -C_{q_{acc}w}(\vec{p}, \vec{t})$$

Pravila 3) nam govore o početnom stanju na traci, pravila 4) o tome kako se menjaju konfiguraracije od M, ako je t' neposredni sledbenik trenutka t a pravilom 5) smo napravili relaciju (Bulovski predikat) Acc koja je tačna ako i samo ako se M zaustavlja u prihvatajućem stanju. Sada M prihvata $Code(\mathfrak{A})$ ako i samo ako je upit (Π_M, Acc) tačan u $\mathfrak{A}$.

□

3.6 Logika Tranzitivnog Zatvorenja

U ovom odeljku ćemo se pozabaviti logikom baziranom na operatorima za izračunavanjem tranzitivnog zatvorenja a onda ćemo pokazati da su i oni ekvivalentni nekim poznatim klasama složenosti.

Definicija 3.7. *Logika tranzitivnog zatvorenja (TrCl) predstavlja ekstenziju logike prvog reda, tako što je u njoj formula i:*

$$[trcl_{\vec{x},\vec{y}}\varphi(\vec{x}, \vec{y}, \vec{z})](\vec{t_1}, \vec{t_2}),$$

gde je $\varphi(\vec{x}, \vec{y}, \vec{z})$ formula sa $|x| = |y| = k$, a slobodne promenljive su $\vec{z}, \vec{t_1}$ i $\vec{t_2}$.

Zadata struktura $\mathfrak{A}$ i vrednosti $\vec{a}, \vec{a_1}$ i $\vec{a_2}$ redom za $\vec{z}, \vec{t_1}$ i $\vec{t_2}$ generišu graf G sa ivicama:

$$\{(\vec{b_1}, \vec{b_2}) \mid \mathfrak{A} \models \varphi(\vec{b_1}, \vec{b_2}, \vec{a})\}$$

Tada:

$\mathfrak{A} \models [trcl_{\vec{x},\vec{y}}\varphi(\vec{x}, \vec{y}, \vec{a})](\vec{a_1}, \vec{a_2})$ *akko* $(\vec{a_1}, \vec{a_2})$ *pripada tranzitivnom zatvorenju od G.*

Teorema 3.9. *TrCl hvata NLogSPACE nad uređenim strukturama.*

Tranzitivno zatvorenje grafa je moguće izračunati u NLogSPACE složenosti (krenemo od jednog čvora i pogađamo sledeći, pri čemu je potrebno samo da pamtimo na kojem smo čvoru i koliko smo ih do sada prošli, dok ne stignemo do željenog čvora, ili odbacimo), te nam u induktivnom dokazivanju toga da se je svaku rečenicu TrCl Logike moguće evaluirati u NlogSPACE-u rečenica koju smo dodali da bi napravili tu logiku neće predstavljati problem. Ono što hoće, međutim je znak negacije koji se može javiti ispred te formule, zbog prirode definicije nedeterminističke mašine. U cilju prevazilaženja ovog problema definišimo logiku PosTrCl u kojoj će se operator tranzitivnog zatvorenja javljati samo sa pozitivnim predznakom. Sada smo u prilici da dokažemo dva tvrđenja, preko kojih ćemo posredno dokaziti i našu teoremu:

T1: Nad uređenim strukturama PosTrCl = TrCl

T2: Nad uređenim strukturama PosTrCl hvata NLogSPACE

Dokažimo prvo tvrđenje T2. Sada imamo situaciju da se negacija može primenjivati samo na formule prvog reda, pa se mogućnost evaluacije PosTrCl formula u NlogSpace izvodi indukcijom kakvu smo već izvodili u ovom radu. Za dokaz drugog smera pretpostavljamo da postoji NTM M koja u složenosti NLogSPACE odlučuje neku osobinu $\mathcal{P}$ σ-struktura i neka M ima dve trake, jednu samo za čitanje i jednu radnu. Ulazna traka prima string $Code(\mathfrak{A})$ a radna traka ne troši više od $c\log(n)$ mesta, za neku konstantu c i kardinalnost univerzuma strukture n.

Položaje na traci označavamo vektorima $\vec{p}$, čija dimenzija zavisi od σ, pa je za njegov opis potrebno $c(\sigma)$ pomenljivih. Stanje ulazne trake za čitanje može biti opisano kao u dokazu Faginove teoreme. Za opis radne trake potrebno je opisati koji simbol je upisan u kom polju, nad kojim poljem se nalazi čitajuća glava i u kom stanju je mašina. Ako se radni alfabet sastoji samo od nula i jedinica tada je broj stringova koji mogu biti zapisani na traci u datom trenutku jednak $2^{c\log(n)} = n^c$, a za njihovo razlikovanje nam je potrebno c promenljivih (odnosno vektor sa c koordinata). Za opis stanja i pozicije glave nam treba $|Q|$ promenljivih, pa nam je ukupan broj promenljivih potreban za razlikovanje svih konfiguracija jedak $c + |Q| + c(\sigma)$.

Ako $\varphi_{init}(\vec{v_0})$ govori da je $\vec{v_0}$ početna konfiguracija, $\varphi_{fin}(\vec{v_a})$ da je $\vec{v_a}$ prihvatajuća konfiguracija a $\varphi_{next}(\vec{v_1}, \vec{v_2})$ da se iz konfiguracije $\vec{v_1}$ po pravilima mašine prelazi u konfiguraciju $\vec{v_2}$, tada je klasa struktura koje prihvata M definisana formulom:

$$\exists \vec{v_0} \exists \vec{v_a} \, (\varphi_{init}(\vec{v_0}) \wedge \varphi_{fin}(\vec{v_a}) \wedge [trcl_{\vec{x},\vec{y}} \varphi_{next}(\vec{x}, \vec{y})](\vec{v_0}, \vec{v_a})) \tag{3}$$

Uvedimo sada nove oznake za minimalni i maksimalni element na skupovima vektora pomoću kojih opisujemo konfiguracije, neka je $\underline{\mathbf{min}}$ minimalni vektor, vektor čije su sve koordinate $\underline{min}$ a $\underline{\mathbf{max}}$ vektor čije su sve koordinate $\underline{max}$. Pomoću njih formulu za prelazak iz jedne konfiguracije u drugu možemo dopuniti tako da postoji opcija da se iz minmalne konfiguracije prelazi u početnu a iz završne u maksimalnu:

$$\varphi'_{next}(\vec{v_1}, \vec{v_2}) = \varphi_{next}(\vec{v_1}, \vec{v_2}) \vee (\vec{v_1} = \underline{\mathbf{min}} \wedge \varphi_{init}(\vec{v_2})) \vee (\vec{v_2} = \underline{\mathbf{max}} \wedge \varphi_{fin}(\vec{v_1}))$$

Sada jednačinu (3) možemo da zapišemo i kao:

$$[trcl_{\vec{x},\vec{y}} \varphi'_{next}(\vec{x}, \vec{y})](\underline{\mathbf{min}}, \underline{\mathbf{max}}),$$

tj. ovom formulom može biti definisana svaka osobina koju je moguće odlučiti NLogSPACE mašinom.

Dokažimo sada i T1. Jasno je da je potrebno rešiti samo slučaj negacije, tj. pokazati da $\neg[trcl_{\vec{x},\vec{y}} \varphi(\vec{x}, \vec{y})](\underline{\mathbf{min}}, \underline{\mathbf{max}})$ ekvivalentna formuli iz PosTrCl.

Za proizvoljnu formulu $\psi(\vec{x},\vec{y})$ i datu strukturu $\mathfrak{A}$ definišemo $d_\psi^\mathfrak{A}(\vec{a},\vec{b})$ kao rastojanje izmedju $\vec{a}$ i $\vec{b}$ u grafu čije relacije definiše $\psi(\mathfrak{A})$. Ako $\vec{a}$ i $\vec{b}$ nisu povezani onda $d_\psi^\mathfrak{A}(\vec{a},\vec{b}) = \infty$. Nadalje, definišemo za svaki element skup elemenata do kojih se od njega može doći:

$$Reach_\psi^\mathfrak{A}(\vec{a}) = \{\vec{b} \in A^k \mid d_\psi^\mathfrak{A}(\vec{a},\vec{b}) \neq \infty\},$$

pa imamo:

$$\mathfrak{A} \models \neg[trcl_{\vec{x},\vec{y}}\varphi(\vec{x},\vec{y})](\underline{\mathbf{min}},\underline{\mathbf{max}}) \text{ akko } \left|Reach_\varphi^\mathfrak{A}(\underline{\mathbf{min}})\right| = \left|Reach^\mathfrak{A}_{\varphi(\vec{x},\vec{y})\wedge\neg(\vec{y}=\underline{\mathbf{max}})}(\underline{\mathbf{min}})\right|$$

Primetimo i da bi formula $\neg[trcl_{\vec{x},\vec{y}}\varphi(\vec{x},\vec{y})](\underline{\mathbf{min}},\underline{\mathbf{max}})$ bila ekvivalentna PosTrCl-formuli:

$$\exists z(\omega_\varphi(\underline{\mathbf{min}},\vec{z}) \wedge \omega_{\varphi(\vec{x},\vec{y})\wedge\neg(\vec{y}=\underline{\mathbf{max}})}(\underline{\mathbf{min}},\vec{z})),$$

ako bi za svaku formulu ψ postojala PosTrCl-formula $\omega_\psi(\vec{x},\vec{z})$ tako da $\mathfrak{A} \models \omega_\psi(\vec{a},\vec{c})$ onda i samo onda kada je $\left|Reach_\psi^\mathfrak{A}(\vec{a})\right| = \vec{c}$, što će nam dati sledeća lema:

Lema 3.4. *Za svaku formulu prvog reda $\psi(\vec{x},\vec{y})$ postoji PosTrCl-formula $\omega_\psi(\vec{x},\vec{z})$ takva da za svaku strukturu $\mathfrak{A}$ važi:*

$$\mathfrak{A} \models \omega_\psi(\vec{a},\vec{c}) \quad \textit{akko} \quad \left|Reach_\psi^\mathfrak{A}(\vec{a})\right| = \vec{c}$$

Dokaz. Definišimo:

$$r_\psi^\mathfrak{A}(\vec{a},\vec{c}) = \left|\{\vec{b} \mid d_\psi^\mathfrak{A}(\vec{a},\vec{b}) \leq \vec{c}\}\right|$$

Ako bi postojala formula $\theta_\psi(\vec{x},\vec{v},\vec{z_1},\vec{z_2})$ sa značenjem:

$$\mathfrak{A} \models \theta_\psi(\vec{a},\vec{e},\vec{c_1},\vec{c_2}) \text{ akko } r_\psi^\mathfrak{A}(\vec{a},\vec{e}) = \vec{c_1} \to r_\psi^\mathfrak{A}(\vec{a},\vec{e}+1) = \vec{c_2},$$

tada je formula ω_ψ definabilna sa:

$$[trcl_{\vec{v_1}\vec{z_1},\vec{v_2}\vec{z_2}}((\vec{v_2} = \vec{v_1}+1) \wedge \theta_\psi(\vec{x},\vec{v_1},\vec{z_1},\vec{z_2}))](\underline{\mathbf{min}},\underline{\mathbf{min}},1\vec{0},\vec{z})$$

Ovde je $1\vec{0}$ vektor koji predstavlja najveću moguću vrednost od $d_\psi^\mathfrak{A}(\vec{a},\vec{b})$, tj. vrednost $|A|^k$. $k+1$ vektor $(c_0,c_1,...,c_k)$ predstavlja udaljenost dužine $c_0n^k + c_1n^{k-1} + ... + c_{k-1}n + c_k$.

Još nam ostaje da vidimo kako se θ_ψ može konstruisati.

Naredna PosTrCl-formula $d_\psi(\vec{x},\vec{y},\vec{z})$ zadovoljava uslov da $\mathfrak{A} \models d_\psi(\vec{a},\vec{b},\vec{c})$ ako i samo ako $d_\psi^\mathfrak{A}(\vec{a},\vec{b}) \leq \vec{c}$:

$$d_\psi(\vec{x},\vec{y},\vec{z}) \equiv [trcl_{\vec{x_1}\vec{z_1},\vec{x_2}\vec{z_2}}(\psi(\vec{x_1},\vec{x_2}) \wedge (\vec{z_1} < \vec{z_2}))](\vec{x},\underline{\mathbf{min}},\vec{y},\vec{z})$$

Nadalje, kako važi da je $r_\psi^\mathfrak{A}(\vec{a},\vec{e}+1) = \vec{c_2}$ ako i samo ako:

$$\vec{c_2} + \left|\{\vec{b} \mid d_\psi^\mathfrak{A}(\vec{a},\vec{b}) > \vec{e}+1\}\right| = 1\vec{0},$$

što naš zadatak pretvara u potragu za PosTrCl formulom koja će izražavati ovaj uslov.

Pre nego što je nađemo, pokažimo da PosTrCl logika ima pogodnu osobinu da nad uređenim strukturama omogućava brojanje, odnosno da za svaku PosTrCl formulu $\varphi(\vec{x})$ postoji PosTrCl formula $\sigma_\varphi(\vec{y})$ za koju važi $\mathfrak{A} \models \sigma_\varphi(\vec{a})$ akko $\left|\{\vec{b} \mid \mathfrak{A} \models \varphi(\vec{b})\}\right| \geq \vec{a}$. Prosto možemo numerisati sve moguće $\vec{b}$-ove i proveriti za svaki od njih da li važi $\mathfrak{A} \models \varphi(\vec{b})$, što može biti urađeno u složenosti NLog, pa i problem da li $\mathfrak{A} \models \sigma_\varphi(\vec{a})$ može bit odlučen u toj složenosti, te može biti izražen PosTrCl logikom.

Formula $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > \vec{e} + 1$ jeste negacija PosTrCl-formule $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) \leq \vec{e} + 1$ (odnosno formule $d_\psi(\vec{a}, \vec{b}, \vec{e} + 1)$) i ona nije u PosTrCl. No, pretpostavimo $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{e}) = \vec{c_1}$. Sada:

- Ako je $\vec{e} = \underline{\mathbf{min}}$ onda $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > 1$ akko $\neg\psi(\vec{a}, \vec{b})$
- Ako $\vec{e} \neq \underline{\mathbf{min}}$ tada $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > \vec{e} + 1$ akko postoji $\vec{c}$ vektora $\vec{f}$ koji su različiti od $\vec{b}$ i takvih da za svako $\vec{f}$ važi $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{f}) < \vec{e}$ i $\neg\psi(\vec{f}, \vec{b})$. Sada se formula d_ψ javlja bez znaka negacije i da bi izrazili $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > \vec{e} + 1$ prebrojimo koliko ima $\vec{f}$-ova koji zadovoljavaju gorenavedeni uslov i dobijeni broj uporedimo sa $\vec{c}$. Kako je to prebrojavanje moguće definisati u PosTrCl to se formula $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > \vec{e} + 1$ može izraziti u PosTrCl.

Sada i formulu θ_ψ takođe možemo da izrazimo u toj logici: krenemo da brojimo od $\vec{c_2}$ i povećavamo za jedan kada god naiđemo na element $\vec{b}$ koji zadovoljava našu formulu, $d_\psi^{\mathfrak{A}}(\vec{a}, \vec{b}) > \vec{e} + 1$, a onda primenimo operator *trcl* da proverimo da li je najudaljeniji element, $1\vec{0}$, dostignut.

□

3.7 Logika za PTIME

Kako postoje logike koje hvataju klase složenosti NP i coNP nad svim konačnim strukturama, postavlja se prirodno pitanje da li postoji takva logika za P? Značaj tog pitanja je jasan, obzirom da se njime "P vs. NP"problem svodi na zadatak da se logike koje hvataju te dve klase složenosti razdvoje nad konačnim strukturama. Takodje je za očekivati da to ne bude lak zadatak tako da za sada imamo samo sledeću Hipotezu:

Hipoteza (Gurevich): *Ne postoji logika koja hvata PTIME nad klasom svih konačnih struktura.*

Problem je formalno definisao i hipotezu dao Yuri Gurevich u radu [5], gde daje preciznu definicuje šta je logika:

Definicija 3.8. *Logika L je funkcija odnosno uredjeni par funkcija (SEN, SAT) koji svakom konačnom jeziku σ pridružuje par rekurzivnih skupova (SEN(σ), SAT(σ)), gde su elementi skupa SEN(σ) L-rečenice jezika σ a SAT(σ) podskup skupa $\{(S, \varphi) \mid S \text{ je konačna FO } \sigma-\text{struktura a } \varphi \in SEN(\sigma)\}$ takav da ako* *su strukture S i S' izomorfne tada $(S,\varphi) \in SAT(\sigma)$ akko $(S',\varphi) \in SAT(\sigma)$. Ako $(S,\varphi) \in SAT(\sigma)$ kažemo da S zadovoljava φ.*

Definicija 3.9. *Ako je L logika a φ L-rečenica jezika σ tada sa MOD(φ) označavamo skup σ-struktura koje zadovoljavaju φ.*

Definicija 3.10. *Logika L hvata PTIME akko:*

1) *Za svaku L-rečenicu φ, MOD(φ) je PTIME odlučivo. Tačnije, za svako σ postoji TM M koja za svaku L-rečenicu φ jezika σ izračunava PTIME TM M(φ) koja izračunava MOD(φ).*

2) *Za svaku PTIME-odlučivu klasu σ-struktura K, ako je K zatvoreno za izomorfizam tada postoji L-rečenica φ jezika σ takva da je $MOD(\varphi)$=K.*

   Ovde dakle PTIME TM M možemo posmatrati kao par (M,p) gde je p polinom sa celobrojnim keoficijentima koji odgovora toj TM.

Ideja ove definicije logike je da ne dozvolimo da se bilo koji skup osobina nazove logikom. Npr, ako pretpostavimo da je Gurevich-eva hipoteza ispravna onda kolekcija svih PTIME osobina ne bi obrazovala logiku.

Ovde ćemo ukratko prikazati nekoliko pokušaja da se pobije Gurevich-eva hipoteza i zašto oni nisu uspeli. Poznato je da se logikama LFP i IFP nad neuređenim strukturam ne mogu izraziti netrivijalne osobine vezane za brojanje (npr osobina da li struktura ima paran broj elemenata) pa je jedan od puteva za osporavanje Gurevich-eve hipoteze bio pokušaj da se na IFP doda mogućnost brojanja te da se takvom logikom pokuša uhvatiti PTIME. Ova logika, označena IFP(Cnt) se može definisati tako što, za σ-strukturu kardinalnosti n uvede dodatni univezum $\{0,1,2,3,...n-1\}$ i dodatni kvantifikator $\exists ix$. Međutim ispostavilo se da je ova logika nedovoljna, odnosno da postoje PTIME osobine koje se ne mogu izraziti u njoj.

Drugačija taktika je bila uvođenje univerzalnih kvanifikatora. Za jezik σ, dodatni relacijski simbol R arnosti k i skup $\{R\}$-struktura $\mathcal{C}$ možemo logiku IFP proširiti geralnim kvantifikatorom $\mathcal{Q}_{\mathcal{C}}$, tako da dobijemo logiku koju zovemo IFP($\mathcal{Q}_{\mathcal{C}}$) u kojoj je, za $|\vec{x}| = k$ formula i:

$$\psi(\vec{y}) = \mathcal{Q}_{\mathcal{C}}\vec{x}\varphi(\vec{x},\vec{y})$$

u značenju:

$$\mathfrak{A} \models \psi(\vec{b}) \ \text{ akko } \ \langle A, \{\vec{a} \mid \mathfrak{A} \models \varphi(\vec{a},\vec{b})\}\rangle \in \mathcal{C}$$

Postavlja se pitanje da li postoji skup generalizovanih kvantifikatora $\mathcal{Q}$ takav da IFP($\mathcal{Q}_{\mathcal{C}}$) hvata PTIME? Sledeći stav nam kaže da bi takav skup bilo teško pronaći čak i da postoji.

Stav: *Ako je $\mathcal{Q}_n$ skup generalisanih kvantifikatori čija arnost ne prelazi n, tada postoji jezik σ_n takav da IFP($\mathcal{Q}$) ne hvata PTIME nad σ_n-strukturam.*

Međutim ovaj rezultat nije dovoljan da isključi mogućnost hvatanja PTIME-a pomoću generalisanih kvantifikatora, budući da se uslovom stava zahteva da arnost kvantifikatora zavisi od n. Tako imamo kolekciju $\mathcal{Q}_{gr}$ binarnih kvantifikatora takvu da IFP($\mathcal{Q}_{gr}$) definiše PTIME osobine grafova. Može se pokazati i da postoji jedan ternarni generalisani kvantifikator $\mathcal{Q}_3$ takav da IFP($\mathcal{Q}_3$) definiše PTIME osobine grafova ali sama $\mathcal{Q}_3$ nije PTIME izračunljiva. Postavljalo se prirodno pitanje postojanja konačne klase $\mathcal{Q}_{fin}$ koja bi hvatala PTIME nad neuređenim grafovima, a i sama bila PTIME izračunljiva, no dokazano je da takva klasa ne postoji, pa je tako propao i ovaj pokušaj pobijanja Gurevich-eve hipoteze, koja je tako ostala da i dalje čeka svoju potvrdu ili pobijanje.

Literatura

[1] Leonid Libkin, Elements of Finite Model Theory, Springer, 2012

[2] Juraj Hromkovič, Theoretical Computer Science: Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography, Springer; 2003.

[3] Ebbinghaus, Heinz-Dieter, Flum, Jörg, Finite Model Theory, Springer; 2nd edition, 1999.

[4] Jouko Väänänen, A Short Course on Finite Model Theory, http://www.math.helsinki.fi/logic/people/jouko.vaananen/shortcourse.pdf

[5] Daniel Leivant, Inductive definitions over finite structures, Carnegie Mellon University (1989.)

[6] Relational Queries Computable in Polynomial Time, Neil Immerman

[7] Logic and the Challenge of Computer Science, Yuri Gurevich,

[8] A First Course in Logic, Shawn Hedman

[9] Mathematical Logic and Model Theory, Alexander Prestel, Charles N. Delzell, Springer-Verlag London Limited 2011