Sveučilište u Zagrebu

Prirodoslovno-matematički fakultet
Matematički odsjek

Ivan Gavran

Logička analiza hibridnih sustava

Diplomski rad

Voditelj rada: prof. dr. sc. Mladen Vuković

Zagreb, 2013.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Posvećujem rad profesorima i kolegama, koji su činili ovaj studij

Sadržaj

1 Uvod

Suvremena razina znanosti i inžinjerstva omogućuje razvoj vrlo složenih sustava. Jednako važnu ulogu kao razvoj tih sustava igra i njihova analiza. Ona omogućuje - već na razini modela - donošenje zaključaka o ponašanju budućih sustava, ispravljanje potencijalnih nedostataka i upotrebu optimalne tehnologije.
Pojam *hibridni sustavi* označava sustave koji u sebi sadrže diskretnu (na primjer digitalnu) i kontinuiranu (na primjer fizikalnu) komponentu. Takvi su sustavi česti u automobilskoj industriji, zračnom i željezničkom prometu, automatizaciji, medicinskim uređajima... U ovom diplomskom radu bit će riječi o logičkoj analizi hibridnih sustava.

1.1 Analiza hibridnih sustava

Hibridni sustavi su oni dinamički sustavi kod kojih se stanje sustava mijenja u vremenu u ovisnosti o preklapajućem utjecaju kontinuiranih i diskretnih faktora. Ideja je obuhvatiti međuodnos kontrolne jedinice koja upravlja kontinuiranim fizikalnim procesom (npr. gibanjem) i samog procesa. Radi velikih mogućnosti koje hibridni sustavi pružaju, oni se koriste sve više u različitim poljima industrije. Unatoč tome, tehnike za analizu hibridnih sustava tek su u začetku i gotovo neupotrebljive. Uzrok je tome to što se otprije poznate tehnike ne mogu primijeniti. Naime, osamdesetih godina prošlog stoljeća došlo je do velikog napretka u analizi konačnih sustava: tada su Clarke i Emerson predstavili tehniku *model checking*. Osnovna je ideja efikasno pretražiti sva moguća stanja u kojima se sustav može naći i detektirati ona *nepoželjna*. Ta je tehnika kasnije proširena i na konačne apstrakcije beskonačnih sustava. Zbog svog principa rada, *model checking* je prikladan za pronalaženje konkretnih protuprimjera. Drugi princip često korišten u analizi je tzv. deduktivna verifikacija. Ideja je pritom pomoću automatskih dokazivača dokazati poželjna svojstva sustava. Oba ova principa su se pokazala neodgovarajućima za analizu hibridnih sustava; prvi zbog prevelike složenosti (*eksplozija* broja stanja, što je i inače problem pri *model-checkingu*, u hibridnim se sustavima ne može nikako kontrolirati), a drugi zbog premalene izražajnosti temporalnih logika.
U ovom se diplomskom radu oslanjamo na doktorsku disertaciju američkog matematičara Andréa Platzera. U njoj se Platzer bavi problemom učinkovite analize hibridnih sustava. Analizu provodi kompozicijskim računom - naime, dokaz poželjnog svojstva razlaže na manje složene dokaze. Za potrebe analize uvodi i tri različite logike. U ovom radu mi se bavimo samo najjednostavnijom od njih - **diferencijalnom dinamičkom logikom**. Za tu logiku koristimo oznaku $\mathsf{d}\mathcal{L}$. Na temelju logike $\mathsf{d}\mathcal{L}$ Platzer je razvio i alat za automatsku verifikaciju *KeYmaera*. KeYmaera je testirana na nekoliko pokaznih primjera. Jedan od njih je ETCS - europski sustav mreže željeznica, koji se postupno implementira, i to će biti glavni motivacijski primjer u ovom diplomskom radu.

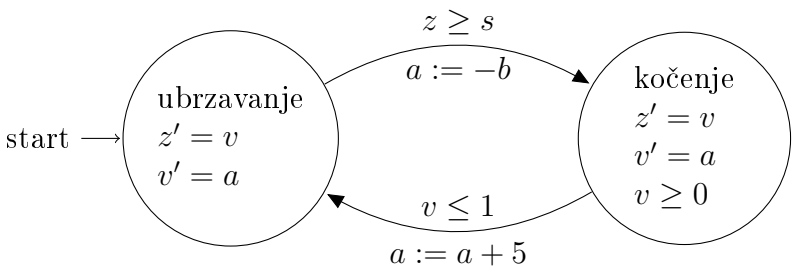
1.2 Pregled rada

Rad se sastoji od sedam poglavlja. Nakon Uvoda, u Poglavlju 2 predstavljamo primjere hibridnih sustava. Na tim ćemo primjerima demonstrirati definicije i tehnike koje će biti uvedene u nastavku rada. Logiku $\mathsf{d}\mathcal{L}$ uvodimo u Poglavlju 3 u kojem zadajemo sintaksu logike - terme, formule, ali i hibridne programe kao dio logike $\mathsf{d}\mathcal{L}$. U Poglavlju 4 definiramo semantiku logike $\mathsf{d}\mathcal{L}$, a u Poglavlju 6 predstavljamo najavljeni kompozicijski račun za analizu hibridnih sustava i dajemo ilustraciju korištenja na jednostavnom primjeru u kontekstu sustava ETCS (taj je sustav detaljnije opisan u Poglavlju 5). Poglavlje 7 opravdava uvedeni račun: u njemu dokazujemo da je on adekvatan i relativno potpun.

2 Pokazni primjeri

U ovom poglavlju uvodimo nekoliko radnih, pokaznih primjera hibridnih sustava. Tim ćemo se pojednostavljenim primjerima uvijek iznova vraćati u narednim poglavljima kako bismo uz njihovu pomoć ilustrirali novouvedene definicije ili tehnike. Također, da bi svrha razvoja $\mathsf{d}\mathcal{L}$ bila jasnija, odmah uz primjere ćemo neformalno predstaviti pojmove koji će biti definirani tek kasnije i uz tako dobivenu širu sliku postaviti pitanja na koja u ostatku rada želimo dati odgovore.

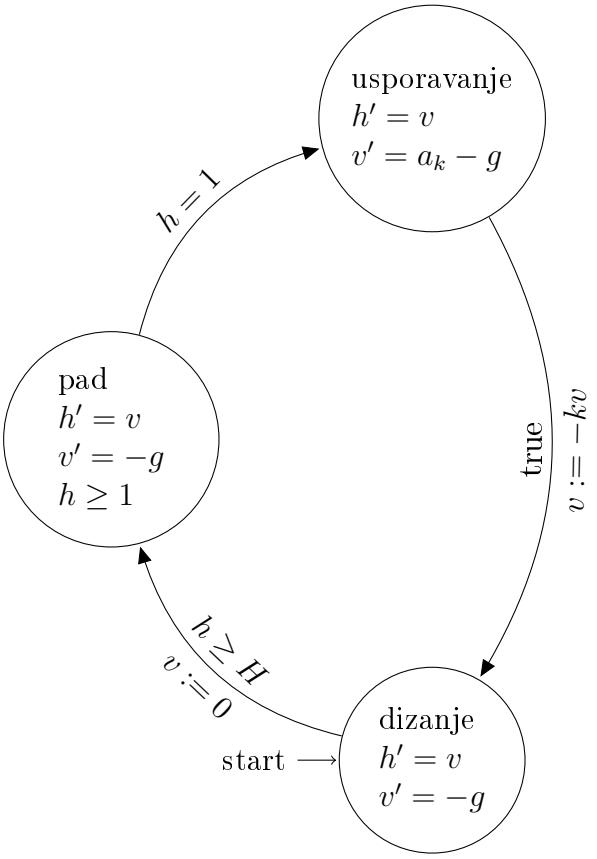
Primjer 2.1. Promotrimo jedan (primitivan) mogući sustav za upravljanje vlakom. Na Slici 1 nalazi se hibridni automat koji ga opisuje. Čvorovi predstavljaju dva različita stanja sustava (ubrzavanje i kočenje). Diferencijalne jednadžbe u njima opisuju gibanje vlaka. Bridovi prikazuju upravljanje sustavom od strane logičke jedinice. Diferencijalna jednadžba u oba čvora je ista (naime, $z' = v, v' = a$) i opisuje da je (vremenska) derivacija položaja z brzina v, a derivacija brzine akceleracija a. Osim ovih jednadžbi, u čvoru *kočenje* nalazi se i ograničenje $v \geq 0$. Ono opisuje dokle se gibanje u skladu s jednadžbama smije odvijati (to odgovara i predodžbi o kretanju vlaka: samo kočenjem se smjer kretanja ne može mijenjati). Sustav nikako ne smije ostati u čvoru (stanju) u kojem ograničenje nije ispoštovano, već mora preći u drugo. Svejedno, to ne znači da u čvoru ostaje sve dok ograničenje vrijedi: može ga napustiti čim je zadovoljen uvjet na nekom od izlaznih bridova. (Da naglasimo još jednom - to je samo nužan uvjet. Kad je uvjet na bridu zadovoljen, tada sustav smije promijeniti čvor, ali i ne mora. Kada će se čvor zaista promijeniti, nije precizirano, radi se o nedeterminističkom sustavu.) Na Slici 1 uvjet za prelazak iz čvora *kočenje* u čvor *ubrzavanje* je da brzina padne ispod 1. To vidimo iznad brida. Ispod brida nalazi se pridruživanje $a := a + 5$. Za vrijeme promjene čvora akceleracija a mijenja svoju vrijednost i gibanje u čvoru *ubrzavanje* se nastavlja uz tako definiranu akceleraciju. Takve diskretne transformacije nazivat ćemo *skokovima*. Za prelazak iz *ubrzavanja* u *kočenje* zahtijevamo da položaj pređe neku unaprijed zadanu veličinu i tada se događa skok akceleracije koja postaje $-b$. (Primijetimo da nikakvo ograničenje unutar čvora *ubrzavanje* ne postoji tako da sustav u tom čvoru može ostati po volji dugo.) Početni čvor sustava je *ubrzavanje* (što je naznačeno riječju start). Kako bismo u potpunosti opisali automat, moramo zadati početne vrijednosti varijabli z i v, početnu vrijednost akceleracije a, kao i konstante s i b.



Slika 1: Hibridni automat za (pojednostavljeni) sustav upravljanja vlakom

U ovako definiranom automatu lako je pronaći manjkavosti: ako je konstanta b dovoljno velika ($b > 5$), u čvoru *ubrzavanje* akceleracija će biti negativna (i nikakvog ubrzanja neće biti). Štoviše, padne li brzina ispod nule, čak i ako je uvjet za prelazak u čvor *kočenje* ispunjen, sustav će biti blokiran u čvoru *ubrzavanje* i nikada više neće moći iz njega izići (zbog ograničenja $v \geq 0$ unutar čvora *kočenje* koje mora biti zadovoljeno i na ulasku u čvor).
Zbog jednostavnosti modela, lako smo uočili greške i lako bismo odabrali *dobre* parametre. Ipak, zanima nas kako u općenitom, složenijem, slučaju biti siguran da sustav neće ostati blokiran u nekom čvoru. Također, kako znati koji su parametri odgovarajući, a koji nisu? Koja je maksimalna brzina koju će sustav razviti i do koje točke će najdalje doći? Odgovore na ta pitanja pružit će nam analiza hibridnih sustava pomoću logike $\mathsf{d}\mathcal{L}$.

Primjer 2.2. U ovom primjeru pomoću hibridnih automata modeliramo jednu od čestih atrakcija u lunaparku, tzv. slobodan pad. Kako ime govori, zamisao je da se posjetitelje podigne na određenu visinu pričvršćene za klupu, zatim klupu pusti da pada da bi je se neposredno pred udarac u tlo zaustavilo, opet podiglo (više ili niže) i tako nekoliko puta.
Ni ovaj put naš model nije potpuno realističan, nego služi da ilustriramo neke od problema koji mogu iskrsnuti. Kako je prikazano na Slici 2, sustav započinje u čvoru *dizanje*. Uzimamo da se uz danu početnu brzinu v_0 penje protiv gravitacije prema gore. U čvoru *pad* pušta se da bez ikakvih vanjskih utjecaja padne (gibajući se po jednadžbi $h'' = -g$). Tako smije padati, kaže ograničenje unutar čvora, sve dok se nalazi na visini većoj od 1 (nakon toga mora preći u čvor *usporavanje*). Primijetimo da to može učiniti samo onda kad se nalazi na visini točno 1 (zbog uvjeta na bridu. Moglo je stajati i $h \leq 1$ s istim rezultatom.). Na bridu koji povezuje čvorove *usporavanje* i *dizanje* nalazi se uvjet *true*. On označava da sustav može preći iz čvora u čvor u bilo kojem trenutnku, bez dodatnih uvjeta. (Oznaka *true* je zapravo izlišna - ako ne napišemo ništa, podrazumijevamo da je uvjet jednak *true*.) Kao i u prethodnom primjeru, za potpunu definiciju potrebno je zadati i početne vrijednosti varijabli h, v kao i vrijednosti a_k i H (g označava akceleraciju prilikom slobodnog pada na Zemlji, tj. $g \sim 9.81 m/s^2$).



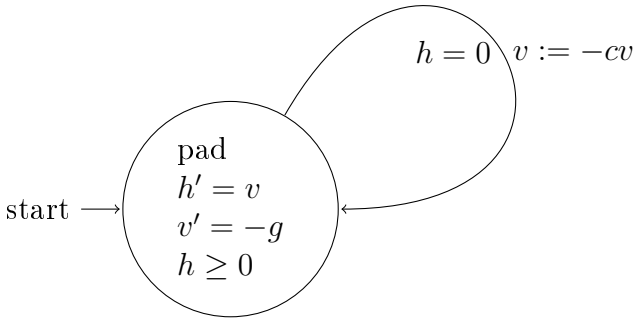
Slika 2: Hibridni automat za *slobodni pad*, atrakciju u lunaparku

U sljedećem primjeru neformalno uvodimo pojam *hibridnog programa*.

Primjer 2.3. Zamislimo loptu koja je ispuštena s određene visine H (na primjer košarkaška lopta kad se provjerava je li dobro napumpana). Taj vrlo jednostavan primjer nije pravi hibridni sustav, ali se može promatrati kao takav (u njemu je *logička komponenta* zemlja koja odbija loptu i mijenja joj smjer). Na Slici 3 je hibridni automat za tu loptu. Dakle, lopta se ispušta iz zraka i giba se pod utjecajem gravitacije ($h'' = -g$). Zadano je ograničenje na $h \geq 0$ (slično kao u Primjeru 2.2, i ovdje smo kombinacijom spomenutog ograničenja i uvjeta na bridu osigurali da se promjena stanja događa točno onda kad je $h = 0$. Dok se stanje mijenja, brzina mijenja smjer, uz $0 < c < 1$, faktor prigušenja. Isti ovaj sustav, osim hibridnim automatom, mogli bismo opisati hibridnim programom.

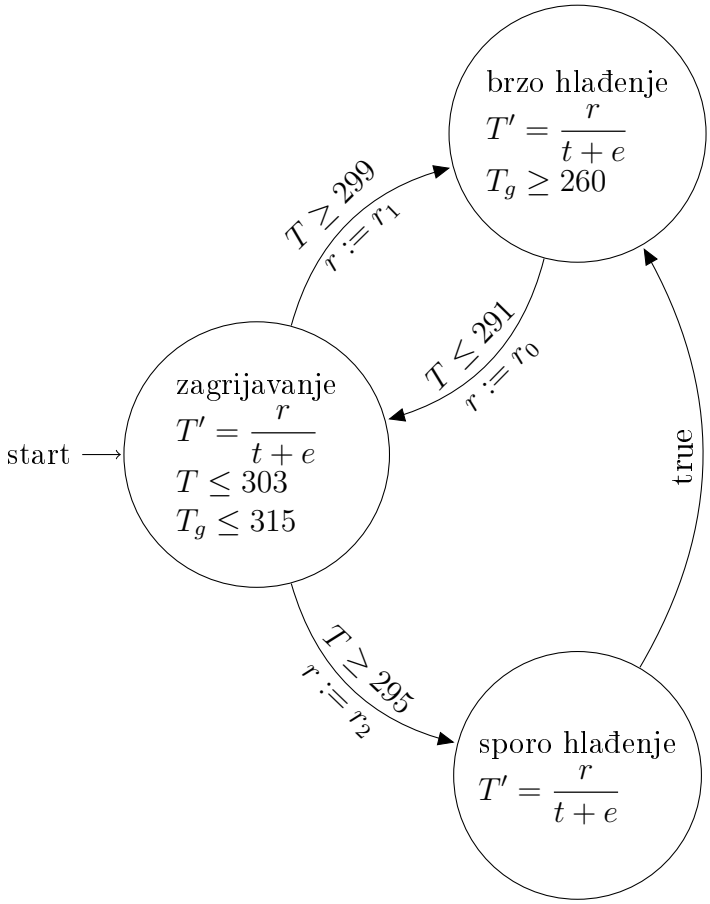
Hibridni program 1
$(h' = v, v' = -g \,\&\, h \geq 0;$
if $(h = 0)$ then
    $v := -cv$
endif
$)^*$

Prva linija programa opisuje kontinuirano gibanje lopte. Ograničenje tog gibanja dano je nakon znaka &. Točka-zarez odvaja taj dio od dijela u kojem je test za prijelaz (if ... then). Ako je uvjet zadovoljen, događa se diskretna promjena vrijednosti varijable v. Na kraju programa nalazi se i regularni operator *, znak da se program može ponoviti po volji mnogo puta. Na ovom primjeru može se pratiti veza između automata i programa, a definicija hibridnog programa doći će u poglavljima koja slijede.



Slika 3: Hibridni automat za loptu koja se odbija od zemlje

Primjer 2.4. Automat prikazan na Slici 4 prikazuje sustav za održavanje temperature u prostoriji. Sustav se sastoji od jednostavnog uređaja koji može biti ili na maksimalnoj temperaturi (zagrijavanje), ili se samo isključiti i prirodno se hladiti (sporo hlađenje) ili se na neki način iznutra hladiti (brzo hlađenje). Simbol T odnosi se na temperaturu prostorije (u Kelvinima, na primjer), a T_g temperaturu uređaja. Ograničenja za T_g odnose se na maksimalnu i minimalnu temperaturu koju uređaj može postići.



Slika 4: Hibridni automat za grijalicu koja zagrijava sobu

Novo u ovom automatu je da se iz čvora *zagrijavanje* može preći i u *brzo hlađenje* i u *sporo hlađenje* i tu se očituje nova mogućnost nedeterminizma u oblikovanju hibridnih automata.

3 Sintaksa

U ovom poglavlju zadajemo i opisujemo sintaksu logike $\mathsf{d}\mathcal{L}$. Kao kod zadavanja sintakse bilo koje logike, polazimo od skupa logičkih varijabli i signature, zatim gradimo terme i na kraju formule logike. Za razliku od drugih logika, u logici $\mathsf{d}\mathcal{L}$ definiramo i hibridne programe koji su ravnopravan sintaktički element i koji grade formule ove logike.

3.1 Termi

Na početku definiramo pojam Skolemovog terma. Radi se o elementu sintakse koji se koristi u procesu *skolemizacije* - eliminacije svih egzistencijalnih kvantifikatora iz formule i uvrštavanjem umjesto njih *varijabli svjedoka*. I Skolemovi termi bit će dio sintakse logike $\mathsf{d}\mathcal{L}$.

Definicija 3.1. Neka je $\mathcal{A}(x_1, x_2, \ldots, x_n, y)$ formula logike prvog reda te neka je $\mathcal{M} = (M, \eta)$ σ-struktura. Term $f(x_1, x_2, \ldots, x_n)$ takav da vrijedi

$$\mathcal{M} \models \exists y \mathcal{A}(x_1, x_2, \ldots, x_n, y) \implies \mathcal{M} \models \mathcal{A}(x_1, x_2, \ldots, x_n, f(x_1, x_2, \ldots, x_n))$$

nazivamo **Skolemovim termom**. Funkciju $f^{\mathcal{M}}\colon M^n \to M$ nazivamo **Skolemovom funkcijom** (iako ponekad taj naziv neprecizno koristimo i za **Skolemov funkcijski simbol** f).

U definiciji alfabeta logike $\mathsf{d}\mathcal{L}$ koja slijedi pobrojat ćemo sve simbole koji se mogu pojaviti kao dio sintakse.

Definicija 3.2. Alfabet logike $\mathsf{d}\mathcal{L}$ je unija skupova $V_1, V_2, \ldots, V_6$ pri čemu su ti skupovi definirani kako slijedi:

- $V_1 = \{x_k : k \in \mathbb{N}\}$
  Elemente skupa V_1 nazivamo **individualnim varijablama**.

- $V_2 = \{R_k^{n_k} : k \in \mathbb{N}\}$
  Elemente skupa V_2 nazivamo **relacijskim simbolima**. Svakom od simbola $R_k^{n_k}$ pridružen je broj $n_k \in \mathbb{N}_0$ koji nazivamo *mjesnošću relacijskog simbola*. Skup V_2 sadrži dvomjesne relacijske simbole koje označavamo s $<$, $>$, $\leq$, $\geq$, $=$.

- $V_3 = \{f_k^{m_k} : k \in \mathbb{R}\}$
  Elemente skupa V_3 nazivamo **funkcijskim simbolima**. Svakom od simbola f^{m_k} pridružen je broj m_k koji nazivamo *mjesnošću funkcijskog simbola*. Skup V_3 sadrži dva 0-mjesna funkcijska simbola, 0 i 1, te dvomjesne funkcijske simbole $+$, $\cdot$, $-$, $/$. Također, sadrži i prebrojivo mnogo Skolemovih funkcijskih simbola $f_{S,i}^{m_i}, i \in \mathbb{N}$. V_3^s je istaknuti podskup skupa V_3 koji sadrži neprebrojivo mnogo funkcijskih simbola mjesnosti nula i čije elemente nazivamo **varijablama stanja**[1].

[1] točnije, broj elemenata skupa V_3^s jednak je broju elemenata skupa $\mathbb{R}$. Motivacija za to je želja da za svaki trenutak (za svaki $t \in \mathbb{R}_0^+$) imamo varijablu stanja. Primijetimo i da kardinalitet skupa V_3^s ne može biti veći od kardinaliteta skupa $\mathbb{R}$ jer je V_3^s podskup skupa V_3.

- $V_4 = \{\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \forall, \exists\}$
  Elemente skupa V_4 nazivamo **logičkim simbolima**.

- $V_5 = \{:=, ', \&, ?, \cup, ;, ^*\}$
  Elemente skupa V_5 nazivamo **simbolima hibridnih programa**.

- $V_6 = \{(,), ,\}$ (lijeva i desna zagrada te zarez)
  Elemente skupa V_6 nazivamo **pomoćnim simbolima**.

Signatura Σ se definira kao unija skupova V_2 i V_3. Dva istaknuta podskupa particioniraju skup Σ: skup $\Sigma_{promjenjivo}$ koji sadrži sve varijable stanja i $\Sigma_{nepromjenjivo} = \Sigma \backslash \Sigma_{promjenjivo}$. Skup varijabli V_1 ćemo ubuduće označavati samo s V.

Napomena 3.1. Već kod zadavanja alfabeta logike $\mathsf{d}\mathcal{L}$ imamo na umu njenu semantiku. Namjera nam je za nosač σ-strukture kojom ćemo interpretirati ove simbole uzeti skup $\mathbb{R}$. Spomenuli smo da Σ sadrži relacijske i funkcijske simbole 0, 1, $+$, -, $\cdot$, $/$, $=$, $\geq$, $\leq$, $<$, $>$... Oni su u alfabetu s namjerom da se interpretiraju standardno za skup realnih brojeva. Mjesnost funkcijskih simbola 0 i 1 je nula (ne primaju argumente), a njima ćemo označavati neutralne elemente za zbrajanje odnosno množenje realnih brojeva. Funkcijski simboli $+$, -, $\cdot$ i $/$ su dvomjesni. Umjesto funkcijske notacije ($\cdot(x, y)$ ili $+(x, y)$), koristit ćemo zapis uobičajen u aritmetici ($x \cdot y$ odnosno $x + y$). Slično je i s relacijskim simbolima $\geq, \leq, <, >$. Razlika između funkcijskih i relacijskih simbola jedino je u tome što će funkcijski simboli označavati funkcije koje primaju vrijednosti argumenata i vraćaju realan broj. S druge strane, relacijski će simboli označavati relacije; one primaju vrijednosti argumenata i vraćaju jednu od *bulovskih* vrijednosti - *istina* ili *laž*.
Primjeri varijabli stanja, kao podskupa skupa funkcijskih simbola, su akceleracija a ili brzina v iz Primjera 2.1.

Sada definiramo terme - argumente za relacijske i funkcijske simbole.

Definicija 3.3. Skup svih **terma** logike $\mathsf{d}\mathcal{L}$, $Trm(\Sigma, V)$, je najmanji skup takav da

- $x \in V \implies x \in Trm(\Sigma, V)$

- ako je $f \in \Sigma$ funkcijski simbol mjesnosti $n \geq 0$ i $\theta_i \in Trm(\Sigma, V)$, $\forall\, 1 \leq i \leq n$, tada je i $f(\theta_1, \theta_2, \ldots, \theta_n) \in Trm(\Sigma, V)$

U sljedećem primjeru dajemo pregled tipičnih terma logike $\mathsf{d}\mathcal{L}$.

Primjer 3.1. Termi logike $\mathsf{d}\mathcal{L}$ su:

- individualne varijable $X \in V$

- varijable stanja $x \in \Sigma$ (kako smo već komentirali, varijabla stanja je funkcijski simbol mjesnosti 0)

- polinomni aritmetički izrazi, na primjer $2x + 4y - 3zw$. Konstante su nepromjenjivi funkcijski simboli mjesnosti 0.

- izrazi u kojima se pojavljuju simboli za Skolemove funkcije, na primjer $x - 3s(X_1, X_2) \cdot 2t(X_3)$. Ovdje je x varijabla stanja, X_1, X_2, X_3 individualne varijable, a s i t (Skolemovi) funkcijski simboli mjesnosti 2 odnosno 1.

- aritmetički izrazi s cjelobrojnim potencijama, na primjer $x^3 - 2y^2$. To se očito može lako preformulirati u $x \cdot x \cdot x - 2 \cdot y \cdot y$.

Sada bismo željeli definirati formule logike $\mathsf{d}\mathcal{L}$. To će biti proširenje formula teorije prvog reda nad signaturom Σ i skupom varijabli V hibridnim programima. Prije no što u sljedećoj točki uvedemo hibridne programe, prisjetimo se definicije formula u teorijama prvog reda. (One će se koristiti i u definiciji hibridnih programa.) U definiciji koristimo oznaku $Trm_{FO}(\Sigma, V)$ za skup svih terma teorije prvog reda nad signaturom Σ i skupom varijabli V (taj pojam ne definiramo jer nema osobitog značenja za logiku $\mathsf{d}\mathcal{L}$).

Definicija 3.4. Skup **formula teorije prvog reda** nad signaturom Σ i skupom varijabli V, $Fml_{FO}(\Sigma, V)$, (oznaka $Fml_{FO}(\Sigma, V)$ motivirana je engleskim *First-Order Formulas*) je najmanji skup takav da:

- ako je $R \in \Sigma$ n-mjesni relacijski simbol, $\theta_i \in Trm_{FO}(\Sigma, V)$, $\forall\, 1 \leq i \leq n$, onda je $R(\theta_1, \theta_2, \ldots, \theta_n) \in Fml_{FO}(\Sigma, V)$

- ako su $\phi, \psi \in Fml_{FO}(\Sigma, V)$, onda su i $\neg\phi, (\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi), (\phi \leftrightarrow \psi) \in Fml_{FO}(\Sigma, V)$

- ako je $\phi \in Fml_{FO}(\Sigma, V)$ i $x \in V$, onda su $(\forall x \phi), (\exists x \psi) \in Fml_{FO}(\Sigma, V)$

Primjer 3.2. Dajemo nekoliko primjera formula prvog reda

- $v^2 \leq 2b(m - z)$ Pretpostavljamo da su v, b, m, z elementi Σ ili V. Ne čini razliku odlučimo li da su oni individualne varijable ili promjenjive varijable stanja ili pak nepromjenjivi funkcijski simboli. To postaje važno tek kad ćemo neku od varijabli htjeti ograničiti i učiniti nepromjenjivom.

- $\forall x \exists y (x^2 < 1 \to 1 < y < x)$ U ovom slučaju mora biti $x, y \in V$ jer po njima kvantificiramo (treća točka u Definiciji 3.4).

- $x^2 + \sqrt{z} = 2$ što stoji za formulu $\exists r (r^2 = z \wedge r \geq 0 \wedge x^2 + r = 2)$.

3.2 Hibridni programi

U ovoj točki opisujemo hibridne programe. Radi se o modelima za hibridne sustave koji objedinjuju kontinuiranu komponentu (izraženu kroz diferencijalne jednadžbe, dio hibridnih programa) i logičku, upravljačku komponentu (koja se ostvaruje kombiniranjem hibridnih programa operacijama $\cup, ^*$ i ;, testiranjem vrijednosti i promjenom vrijednosti varijabli). Dajemo na početku definiciju hibridnih programa, a zatim objašnjavamo pojedine dijelove i (neformalno) njihovu namjenu na semantičkoj razini.

Definicija 3.5. Skup **hibridnih programa**, $HP(\Sigma, V)$, je najmanji skup takav da

(1) ako su $x_1, x_2, \ldots, x_n \in \Sigma$ međusobno različite varijable stanja, a $\theta_i \in Trm(\Sigma, V)$ term za $1 \leq i \leq n$ onda je $(x_1 := \theta_1, x_2 := \theta_2, \ldots, x_n := \theta_n) \in HP(\Sigma, V)$. Izraz $(x_1 := \theta_1, x_2 := \theta_2, \ldots, x_n := \theta_n)$ nazivamo **diskretnim skokom**.

(2) neka je $x_i \in \Sigma$ varijabla stanja, a $\theta_i \in Trm(\Sigma, V)$ za $1 \leq i \leq n$. Ako je $\chi \in Fml_{FO}(\Sigma, V)$, onda je $(x_1' = \theta_1, x_2' = \theta_2, \ldots, x_n' = \theta_n \,\&\, \chi) \in HP(\Sigma, V)$. Izraz $(x_1' = \theta_1, x_2' = \theta_2, \ldots, x_n' = \theta_n \,\&\, \chi)$ nazivamo **neprekidnom evolucijom**.

(3) ako je $\chi \in Fml_{FO}(\Sigma, V)$, onda je $(?\chi) \in HP(\Sigma, V)$. Izraz $(?\chi)$ zove se **test**

(4) ako su $\alpha, \beta \in HP(\Sigma, V)$, onda je $(\alpha \cup \beta) \in HP(\Sigma, V)$. Izraz $(\alpha \cup \beta)$ nazivamo **nedeterminističkim izborom**.

(5) ako su $\alpha, \beta \in HP(\Sigma, V)$, onda je $(\alpha; \beta) \in HP(\Sigma, V)$. Izraz $(\alpha; \beta)$ nazivamo **nizanjem**.

(6) ako je $\alpha \in HP(\Sigma, V)$, onda je $(\alpha^*) \in HP(\Sigma, V)$. Izraz (α^*) nazivamo **nedeterminističkim ponavljanjem**.

U napomeni koja slijedi govorit ćemo više o namijenjenoj semantičkoj ulozi svakog od pobrojanih šest izraza iz Definicije 3.5 (nazivat ćemo ih operacijama). Za to će nam biti potreban pojam *toka diferencijalne jednadžbe.*

Definicija 3.6. Neka je zadana diferencijalna jednadžba $f'(t) = h(t)$. Funkcija $g\colon \mathbb{R}^2 \to \mathbb{R}$ je **tok diferencijalne jednadžbe** $f'(t) = h(t)$ ako vrijedi $g(t, f(0)) = f(t)$. Tako definirana funkcija g je jedinstvena.

Napomena 3.2. U ovoj napomeni detaljnije opisujemo intuitivno značenje svake od šest operacija koje čine hibridne programe. Prva operacija o kojoj govorimo je **diskretan skok** - $(x_1 := \theta_1, x_2 := \theta_2, \ldots, x_n := \theta_n)$
Željeni efekt diskretnog skoka je *istovremeno* mijenjanje interpretacije varijabli x_i u odgovarajuće θ_i. (Naglasak na riječi istovremeno odnosi se na to da se svi θ_i, koje mogu ovisiti o nekom x_j, izvrijedne na početku, prije promjene bilo kojeg x_j.)
Slijedi operacija koja opisuje kontinuirano mijenjanje hibridnog sustava, **neprekidna evolucija** - $(x_1' = \theta_1, x_2' = \theta_2, \ldots, x_n' = \theta_n \,\&\, \chi)$
Ovdje $x_i' = \theta_i$ označava vremensku derivaciju varijable x_i, tj. $\dfrac{dx_i(t)}{dt} = \theta_i(t)$. Neprekidna evolucija označava promjenu vrijednosti varijabli $x_1, x_2, \ldots, x_n$ za različite vrijednosti t u skladu sa zadanim diferencijalnim jednadžbama, a početne vrijednosti su vrijednosti varijabli $x_1, \ldots, x_n$ neposredno prije početka neprekidne evolucije. Evolucija se može zaustaviti *u bilo kojoj točki unutar* χ - tj. za bilo koju vrijednost t za koju χ vrijedi, ali ne smije napustiti χ. (Tu se očituje nedeterminiranost ove operacije.) Ako evoluciju ne želimo ograničiti na određenu domenu, tada umjesto $x' = \theta \,\&\, true$ pišemo samo $x' = \theta$. Zahtijevamo (radi pravila računa logike $\mathsf{d}\mathcal{L}$ koja će biti opisana u Poglavlju 4) da se tok (ili njegova aproksimacija) svake diferencijalne jednadžbe

može definirati formulom prvog reda.
Operacija **test** omogućuje nam grananje programa. Semantika namijenjena operaciji test oblika $(?\chi)$ je nepromijenjeno izvršavanje hibridnog programa ako formula χ vrijedi u danom stanju. U suprotnom, program je blokiran i ne može nastaviti izvršavanje.
Namijenjena semantika operacije **nedeterministički izbor** oblika $(\alpha \cup \beta)$ jest da se nedeterministički odabire hoće li hibridni sustav slijediti program α ili β.
Za operaciju **nizanje** oblika $\alpha; \beta$ želimo da bude potpuno deterministička; program β započinje s izvršavanjem tek nakon što je α završio.
Posljednja operacija koju opisujemo je **nedeterminističko ponavljanje**. Oznaka za nju motivirana je sličnom operacijom iz regularnih izraza - α^*. Namijenjena joj je semantika da se hibridni program α izvršava n puta, gdje je $n \in \mathbb{N}_0$ nedeterministički odabran broj.
Od šest operacija o kojima smo govorili, tri su nedeterminističke (neprekidna evolucija, nedeterministički izbor i nedeterminističko ponavljanje). To nam omogućuje modeliranje mnogih realnih situacija hibridnim programima (na primjer, vlak zahtijeva komunikaciju s kontrolnim tornjem. Ovisno o snazi signala, veza može biti uspostavljena, ali i ne mora).
Također, zbog spomenutih nedeterminističkih operacija, vrijednost varijabli nakon izvršavanja hibridnog programa nije jednoznačno određena. Stoga možemo govoriti o *svim izvršavanjima hibridnog programa* ili o *nekom izvršavanju hibridnog programa*. Što je točno izvršavanje hibridnog programa trenutno predstavljamo samo na intuitivnoj razini, a bit će definirano u Poglavlju 4.

Primjer 3.3. Promotrimo vrlo jednostavan hibridni program koji opisuje gibanje vlaka po jednadžbi $z' = v, v' = a$.

Hibridni program 2
$((a := -b) \cup (?v < 8; a := A)); z' = v, v' = a)^*$

Na početku sustav nedeterministički odabire (što je određeno simbolom $\cup$) hoće li postaviti akceleraciju na $-b$, akceleraciju kočenja, $(a := -b)$ ili će, samo ako je brzina manja od 8, postaviti akceleraciju na pozitivnu vrijednost A $((?v < 8; a := A))$. Nakon toga se sustav ravna prema početnoj diferencijalnoj jednadžbi po volji dugo. Operacija nedeterminističkog ponavljanja na kraju (simbol *) označava da se cijeli postupak može ponoviti po volji mnogo puta.
Primijetimo još da smo dio programa koji opisuje gibanje prema diferencijalnoj jednadžbi ostavili izvan zagrada. Takvu ćemo sintaktičku nepreciznost i dalje upotrebljavati, kada je smisao jasan, radi jednostavnosti.

Primjer 3.4. Prisjetimo se hibridnog automata iz Primjera 2.4 koji modelira rad upravljačke jedinice sobne grijalice. Hibridni program za njega izgleda ovako

Hibridni program 3

$$((T' = \frac{r}{t+e} \,\&\, T \leq 203 \wedge T_g \leq 215); (?T \geq 199; r := r_1) \cup$$
$$\cup (?T \geq 195; r := r_2; T' = \tfrac{r}{t+e}); (T' = \tfrac{r}{t+e} \,\&\, T_g \geq 160); ?T \geq 199; r := r_1)^*$$

Ako se prisjetimo Hibridnog programa 1 iz Primjera 2.3 koji je opisivao odskakivanje košarkaške lopte ispuštene s visine, u oči upada to da smo tamo koristili *if* naredbu za kontrolu toka o kojoj nema spomena u definiciji hibridnih programa. Radi se zapravo o pokrati - sve se klasične naredbe za kontrolu toka programa daju izraziti pomoću hibridnih programa.

Primjer 3.5. U ovom primjeru ilustriramo kako se neke od klasičnih naredbi za kontrolu toka programa izražavaju pomoću hibridnih programa. Redom dajemo hibridne programe za operacije *if-then-else*, *if-then*, *while-do*, *repeat-until* (koje su standardne u svim programskim jezicima i čije je značenje jasno) te *abort*. Operacija *abort* (pojavljuje se npr. u programskom jeziku Java) prekida izvršavanje programa.

- provjera *if* χ *then* α *else* β izražava se hibridnim programom $(?\chi; \alpha) \cup (?\neg\chi; \beta)$ Iako se ovdje prividno radi o nedeterminističkom izboru, nije tako: točno jedan od uvjeta χ ili $\neg\chi$ može vrijediti pa je sustav prisiljen odabrati njega.
- provjera *if* χ *then* α izražava se hibridnim programom $(?\chi; \alpha) \cup (?\neg\chi)$; Primijetimo da je bilo bitno dodati i dio nakon znaka $\cup$, $(?\neg\chi)$. U suprotnom bi sustav, kad χ ne vrijedi, samo stao i program se ne bi mogao izvršiti
- petlja *while* χ *do* α izražava se hibridnim programom $(?\chi; \alpha)^*; ?\neg\chi$
- petlja *repeat* α *until* χ izražava se hibridnim programom $\alpha(?\neg\chi; \alpha)^*; ?\chi$
- operacija *abort* može se izraziti hibridnim programom $?\mathit{false}$

Zaista, možemo reći i više - da se hibridnim programima može simulirati rad Turingovog stroja. Dokaz te tvrdnje dan je u Poglavlju 4.

3.3 Formule logike d$\mathcal{L}$

Formule logike d$\mathcal{L}$ definirane su slično kao u Definiciji 3.4, uz dodatak hibridnih programa kao elemenata formula. Ponovno ćemo nakon definicije dati nekoliko primjera koji ilustriraju definiciju formula i pojašnjavaju njihovu upotrebu u logici d$\mathcal{L}$.

Definicija 3.7. Skup **formula logike d$\mathcal{L}$**, $Fml(\Sigma, V)$, je najmanji skup takav da:

1. ako je $R \in \Sigma$ neki n-mjesni relacijski simbol, $\theta_i \in Trm(\Sigma, V)$, $\forall\, 1 \leq i \leq n$, onda je $R(\theta_1, \theta_2, \ldots, \theta_n) \in Fml(\Sigma, V)$
2. ako su $\phi, \psi \in Fml(\Sigma, V)$, onda su i $\neg\phi, (\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi), (\phi \leftrightarrow \psi) \in Fml(\Sigma, V)$
3. ako je $\phi \in Fml(\Sigma, V)$ i $x \in V$, onda su $(\forall x \phi), (\exists x \phi) \in Fml(\Sigma, V)$

4. ako je $\phi \in Fml(\Sigma, V)$ i $\alpha \in HP(\Sigma, V)$, tada su i $[\alpha]\phi, \langle\alpha\rangle\phi \in Fml(\Sigma, V)$

Napomena 3.3. Kao i kod hibridnih programa, dajemo pregled operatora korištenih u izgradnji formula i njihovo (neformalno) značenje. I dok je namijenjena semantika operatora koji se pojavljuju i u formulama teorija prvog reda standardna, zanimljivo će biti objašnjenje za operatore *box* ($[]$) i *diamond* ($\langle\rangle$). Ovdje oznaka $\cdot^{\mathcal{M}}$ stoji za semantičku interpretaciju danih sintaktičkih elemenata. O kakvoj se točno interpretaciji radi, bit će izloženo u Poglavlju 4. Redom navodimo operatore i njihova intuitivna značenja.

a) **atomarna formula**, $R(\theta_1, \theta_2, \ldots, \theta_n)$, je istinita ako je $(\theta_1^{\mathcal{M}}, \theta_2^{\mathcal{M}}, \ldots, \theta_n^{\mathcal{M}}) \in R^{\mathcal{M}}$.

b) **negacija**, $\neg\phi$, je istinita ako je ϕ neistinita

c) **konjunkcija**, $\phi \wedge \psi$, je istinita ako su i ϕ i ψ istinite

d) **disjunkcija**, $\phi \vee \psi$, je istinita ako je ϕ ili ψ istinita

e) **kondicional**, $\phi \to \psi$, je istinit ako je ϕ neistinita ili ψ istinita

f) **bikondicional**, $\phi \leftrightarrow \psi$, je istinit ako su ϕ i ψ istovremeno istinite ili neistinite

g) **univerzalni kvantifikator**, $\forall x\phi$, je istinit ako je ϕ istinita za sve (realne) vrijednosti varijable x

h) **egzistencijalni kvantifikator**, $\exists x\phi$, je istinit ako postoji vrijednost varijable x za koju je ϕ istinita

i) $[\cdot]$ **(*box*) modalnost**, $[\alpha]$, je istinita ako je ϕ istinita za sva izvršavanja hibridnog programa α s početkom u trenutnom stanju

j) $[\cdot]$ **(*diamond*) modalnost**, $\langle\alpha\rangle$, je istinita ako je ϕ istinita barem za jedno izvršavanje hibridnog programa α s početkom u trenutnom stanju

Kao kod ranijih sintaktičkih elemenata, i ovdje ćemo često izostavljati zagrade radi preglednosti i pouzdati se u prioritete: najviši prioritet imaju negacija ($\neg$), kvantifikatori ($\exists, \forall$) i modalnosti ($[\cdot], \langle\cdot\rangle$). Zatim slijede konjunkcija ($\wedge$) i disjunkcija ($\vee$), a na kraju su kondicional ($\to$) i bikondicional ($\leftrightarrow$).
Nakon što smo definirali formule i objasnili namijenjenu im semantiku, možemo jezikom logike $\mathsf{d}\mathcal{L}$ formulirati neka od pitanja koja smo već postavili u početnim primjerima, u Poglavlju 2.

Primjer 3.6. Prisjetimo se hibridnog automata iz Primjera 2.4 i pripadnog hibridnog programa iz Primjera 3.4. Nazovimo taj hibridni program *grijalica*. Sada tvrdnju da temperatura, uz danu početnu temperaturu, nikad neće pasti ispod 280 K možemo izraziti formulom

$$T_0 = 290 \to [grijalica]T \geq 280$$

Ako je ta formula istinita, tvrdnja vrijedi. Ekvivalentno, možemo provjeriti je li sljedeća formula neistinita

$$T_0 = 290 \rightarrow \langle grijalica \rangle T < 280$$

Primjer 3.7. Neka *vlak* označava neki hibridni program koji modelira kretanje vlaka (moguće i onaj iz Primjera 3.3). Tada formulom

$$v \geq 0 \wedge z < m \rightarrow \langle vlak \rangle z \geq m$$

(pri čemu je z pozicija vlaka, a m zadana točka na pruzi) izražavamo pitanje je li moguće da vlak prijeđe točku m ako svoje kretanje započne nenegativnom brzinom prije točke m. Ponovno, isto pitanje možemo dualno izraziti formulom

$$v \geq 0 \wedge z < m \rightarrow [vlak] z < m$$

Primjer 3.8. Budući da su hibridni programi sastavni dio formula logike $\mathsf{d}\mathcal{L}$, oni se mogu kombinirati s kvantifikatorima, gnijezditi ili nizati. Tako možemo pisati $[\langle \beta \rangle; ?\phi]\psi$ i time izraziti da za sva izvršavanja hibridnog programa koji se sastoji od izvršavanja hibridnog programa β nakon kojeg vrijedi ϕ, vrijedi i ψ.
Formula $[\alpha]\langle \beta \rangle \phi$ kaže da što god činio program α, postoji reakcija programa β takva da ϕ vrijedi. Ili, formulom $\exists p [\alpha] \phi$ kazujemo da možemo odabrati parametar p tako da nakon svakog izvršavanja programa α formula ϕ vrijedi.

Radi bolje čitljivosti, uvijek ćemo pretpostavljati da su diferencijalne jednadžbe ili diskretni skokovi koje upotrebljavamo dobro definirani. Konkretno, pretpostavljamo da uz sve izraze oblika $\frac{p}{q}$ stoji uvjet koji osigurava da je $q \neq 0$.
U ovom smo poglavlju definirali sintaksu logike $\mathsf{d}\mathcal{L}$ i dali naslutiti koja će biti semantika svakog od elemenata. U sljedećem poglavlju formalno definiramo semantiku.

4 Semantika

U poglavlju o semantici sintaktičkim elementima želimo pridijeliti značenja. Ta će značenja slijediti intuiciju o kojoj smo već govorili u prethodnim poglavljima. Ipak, definicije će biti strogo formalne, stoga i detaljne i duge. Zbog toga će veći dio poglavlja biti nizanje definicija. Između definicija ubacit ćemo nekoliko primjera koji će ih ilustrirati.

4.1 Valuacija terma

Nelogičke simbole logike $\mathsf{d}\mathcal{L}$ podijelili smo u tri različite kategorije:

- individualne varijable (skup V): njihova se vrijednost ne će mijenjati izvršavanjem hibridnog programa, a moći će biti kvantificirane univerzalno ili egzistencijalno
- promjenjivi simboli (skup $\Sigma_{promjenjivo}$): zovemo ih i varijable stanja jer će se njihova vrijednost mijenjati izvšavanjem hibridnih programa, ovisno o stanju u kojem se sustav nalazi
- nepromjenjivi simboli (skup $\Sigma_{nepromjenjivo}$): simboli čija se vrijednost nikad ne će promijeniti niti će se po njima kvantificirati, na primjer 0, 1, $+$, $\cdot$

Svakoj od triju pobrojanih kategorija simbola vrijednost pridružujemo posebnom funkcijom. Individualnim varijablama vrijednost dajemo *pridruživanjem* $\eta\colon V \to \mathbb{R}$. Istu ulogu za nepromjenjive simbole igra *interpretacija* I čija je domena skup $\Sigma_{nepromjenjivo}$. Ideja je da interpretacija I znakova $+$, - ili relacijskih simbola ne ovisi o stanju ν, već da za svako stanje znači isto. Funkciju $\nu\colon \Sigma_{promjenjivo} \to \mathbb{R}$ nazivamo *stanjem*. Konačno, valuaciju terma, $val_{I,\eta}(\nu, \cdot)$ definiramo pomoću pridruživanja, interpretacije i stanja (koje je argument valuacije).

Definicija 4.1. Valuacija terma s obzirom na interpretaciju I, valuaciju η i stanje ν definira se induktivno s

- $val_{I,\eta}(\nu, x) = \eta(x)$, ako je $x \in V$
- $val_{I,\eta}(\nu, a) = \nu(a)$, ako je $a \in \Sigma_{promjenjivo}$ (varijabla stanja)
- $val_{I,\eta}(\nu, f(\theta_1, \ldots, \theta_n)) = I(f)(val_{I,\eta}(\nu, \theta_1), \ldots, val_{I,\eta}(\nu, \theta_n))$, ako je $f \in \Sigma_{nepromjenjivo}$ nepromjenjivi funkcijski simbol mjesnosti $n \geq 0$.

Primjer 4.1. Neka je $g \in \Sigma$ jednomjesni funkcijski simbol, $X \in V$ individualna varijabla te $a \in \Sigma$ varijabla stanja. Ako je g interpretiran s $I(g)(t) = t^2$, varijabli X pridružena vrijednost $\eta(X) = 2.8$, a u stanju ν vrijednost varijable stanja a je

$\nu(a) = 1.2$, onda se term $g(X + a)$ valuira s

$$\begin{aligned} val_{I,\eta}(\nu, g(X + a)) &= I(g)(val_{I,\eta}(\nu, X + a)) \\ &= I(g)(val_{I,\eta}(\nu, X) + val_{I,\eta}(\nu, a)) \\ &= I(g)(\eta(X) + \nu(a)) \\ &= I(g)(2.8 + 1.2) = I(g)(4) \\ &= 4^2 = 16 \end{aligned}$$

Primijetimo da smo ovdje prešutno koristili interpretaciju znaka + kao zbrajanja. To ćemo i ubuduće smatrati unaprijed zadanim. Usto, valja spomenuti i da je za ovaj primjer bilo potpuno irelevantno što je varijabla stanja, što nepromjenjivi funkcijski simbol, a što individualna varijabla. To će igrati ulogu tek kada, unutar formula, budemo valuirali modalnosti ili kvantificirane varijable.

4.2 Semantika hibridnih programa

Semantika hibridnih programa je semantika prijelaza iz jednog stanja u drugo; za svaki hibridni program definirat ćemo relaciju $\rho_{I,\eta}$. Dva će stanja biti u relaciji ako se izvršavanjem hibridnog programa iz jednog stanja može preći u drugo.
U Definiciji 4.3 definiramo tranzicijsku semantiku hibridnih programa. U njoj koristimo pokratu $\nu[x \mapsto d]$ za semantičku modifikaciju stanja ν. Taj jednostavan pojam dan je u sljedećoj definiciji.

Definicija 4.2. Semantička modifikacija stanja ν jest stanje koje označavamo $\nu[x_1 \mapsto d_1, \ldots, x_n \mapsto d_n]$. To se stanje, kako oznaka sugerira, u svemu slaže s ν, osim u interpretaciji simbola $x_1, \ldots, x_n \in \Sigma_{promjenjivo}$ koje redom interpretira kao $d_1, \ldots, d_n \in \mathbb{R}$.

Sada smo spremni za najavljenu definiciju valuacije hibridnog programa.

Definicija 4.3. Valuacija hibridnog programa α u odnosu na interpretaciju I i pridruživanje η, u oznaci $\rho_{I,\eta}(\alpha)$, je dvomjesna relacija čiji su elementi uređeni parovi stanja, $\rho_{I,\eta}(\alpha) \subset Sta(\Sigma) \times Sta(\Sigma)$ (gdje je $Sta(\Sigma)$ skup svih stanja). Budući da su hibridni programi definirani induktivno sa šest osnovnih operacija, za istih šest operacija definiramo $\rho_{I,\eta}(\cdot)$.

(1) $(\nu, \omega) \in \rho_{I,\eta}(x_1 := \theta_1, x_2 := \theta_2, \ldots, x_n := \theta_n)$ ako i samo ako je stanje ω dobiveno odgovarajućom semantičkom modifikacijom stanja ν,

$$\omega = \nu[x_1 \mapsto val_{I,\eta}(\nu, \theta_1)][x_2 \mapsto val_{I,\eta}(\nu, \theta_2)] \ldots [x_n \mapsto val_{I,\eta}(\nu, \theta_n)]$$

(2) $(\nu, \omega) \in \rho_{I,\eta}(x_1' = \theta_1, x_2' = \theta_2, \ldots, x_n' = \theta_n \,\&\, \chi)$ ako i samo ako postoji funkcija $f\colon [0, r] \to Sta(\Sigma)$ takva da:

- $f(0) = \nu, f(r) = \omega$
- f zadovoljava zadane diferencijalne jednadžbe, tj. za sve varijable stanja x_i valuacija $val_{I,\eta}(f(t), x_i) = f(t)(x_i)$ u stanju $f(t)$ je neprekidna po $t \in [0, r]$ i ima derivaciju $val_{I,\eta}(f(t), \theta_i)$ za sve $t \in \langle 0, r\rangle$
- vrijednosti svih ostalih varijabli su nepromijenjene
- f poštuje ograničenje χ, tj. $val_{I,\eta}(f(t), \chi) = istina$ za sve $t \in [0, r]$

(3) $\rho_{I,\eta}(?\chi) = \{(\nu, \nu) \colon val_{I,\eta}(\nu, \chi) = \text{istina}\}$

(4) $\rho_{I,\eta}(\alpha \cup \beta) = \rho_{I,\eta}(\alpha) \cup \rho_{I,\eta}(\beta)$

(5) $\rho_{I,\eta}(\alpha; \beta) = \{(\nu, \omega) \colon (\nu, \mu) \in \rho_{I,\eta}(\alpha), (\mu, \omega) \in \rho_{I,\eta}(\beta)\}$

(6) $(\nu, \omega) \in \rho_{I,\eta}(\alpha^*)$ ako i samo ako postoji prirodan broj n i stanja $\nu = \nu_0, \ldots, \nu_n = \omega$ takva da $(\nu_i, \nu_{i+1}) \in \rho_{I,\eta}(\alpha)$ za sve $0 \leq i < n$

Uređeni par (ν, ω) za koji vrijedi $(\nu, \omega) \in \rho_{I,\eta}(\alpha)$ nazivamo **izvršavanjem hibridnog programa α s početkom u stanju ν**. Stanje ω nazivamo **stanjem nakon izvršavanja hibridnog programa α**. Za dano stanje ν, stanje ω nije jednoznačno određeno programom α i relacijom $\rho_{I,\eta}$ (razlog su nedeterminističke operacije u hibridnim programima).

Iako je Definicija 4.3 na prvi pogled vrlo složena, ona samo formalizira intuitivni smisao operatora u hibridnim programima opisan u Napomeni 3.2.
U Poglavlju 3 tvrdili smo da se hibridnim programima može izračunati sve što se može izračunati Turingovim strojevima - sada tu tvrdnju i dokazujemo.

Definicija 4.4. Kažemo da hibridni program α **izračunava** funkciju $f \colon \mathbb{N}^k \to \mathbb{N}$ ako za sve $x_1, \ldots, x_k$ i x takve da je $f(x_1, \ldots, x_k) = x$ i za sva izvršavanja (ν, ω) hibridnog programa α vrijedi $\nu(t_i) = x_i, \forall 1 \leq i \leq k$, a $\omega(t_{fin}) = x$. Ovdje su t_i i t_{fin} su varijable stanja rezervirane za ulaznu i izlaznu vrijednost izračunavanja.

Definicija 4.5. Za funkciju $f \colon \mathbb{N}^k \to \mathbb{N}$ kažemo da je **Turing-izračunljiva** ako postoji Turingov stroj koji je izračunava.

Propozicija 4.1. Svaku Turing izračunljivu funkciju moguće je izračunati nekim hibridnim programom.

Dokaz. Pretpostavimo da je dan Turingov stroj $\mathcal{T} = (Q, \Gamma, b, \Sigma_T, \delta, q_s, F)$[2] koji računa funkciju f, a funkcija prijelaza je definirana s $\delta(q_j, \alpha) = (q'_j, \alpha', p)$. Opišimo hibridni program koji izračunava funkciju f. Neka je $\nu(t_i)$ znak upisan na i-tom polju trake stroja $\mathcal{T}$. Promotrimo hibridni program α.

[2]Turingov stroj $\mathcal{T}$ određen je svojim skupom stanja Q, alfabetom vrpce stroja Γ čiji su elementi i prazan simbol b i ulazni simboli iz skupa Σ_T. Zatim, δ je funkcija prijelaza, a početno stanje i skup završnih stanja su redom q_s i F. Usto, pretpostavljamo da je rezultat izračunavanja funkcije Turingovim strojem upisan u nultom polju vrpce.

$$\alpha \equiv i := 0; q := q_s; (?(q \in F); \bigcup_{j \in \mathcal{J}} (?(q = q_j)(q, t_i, p) := \delta(q_j, t_i); i := i + p))^*$$

$$\cup ?(q \in F); t_{fin} := t_0$$

Neka je ω takvo stanje da vrijedi $(\nu, \omega) \in \rho_{I,\eta}(\alpha)$. Tvrdimo da je $\omega(t_{fin})$ jednaka vrijednosti izračunavanja Turingovog stroja $\mathcal{T}$. Da se u to uvjerimo, prođimo redom kroz naredbe programa α. Na početku se varijabla i, koja predstavlja poziciju glave stroja $\mathcal{T}$, postavlja na 0, a varijabla q, koja označava trenutno stanje stroja, na početno stanje stroja $\mathcal{T}$, q_s. Nakon toga se uzastopno, sve dok nije $q \in F$, uspoređuje vrijednost varijable q s vrijednostima konstanti q_i, koje predstavljaju stanja stroja $\mathcal{T}$. (Simbol $\mathcal{J}$ korišten u ovom programu označava konačan skup indeksa svih stanja q_i.) Zatim se vrijednostima q, t_i i p_i pridružuju vrijednosti definirane u konstantskom simbolu $\delta(q_j, t_i)$ (simboli δ su popisane funkcijske vrijednosti funkcije prijelaza δ). Na kraju, kada vrijedi $q \in F$, varijabla t_{fin} poprima vrijednost t_0, vrijednost na nultoj poziciji i rezultat iztračunavanja stroja $\mathcal{T}$. ■

4.3 Semantika formula logike d$\mathcal{L}$

Sada možemo definirati semantiku formula iz d$\mathcal{L}$. Slično kao što smo to učinili za semantičku modifikaciju stanja, uvodimo oznaku $\eta[x \mapsto d]$ za semantičku modifikaciju pridruživanja η koja se slaže s η u svim varijablama osim varijable $x \in V$ kojoj je pridružen $d \in \mathbb{R}$.

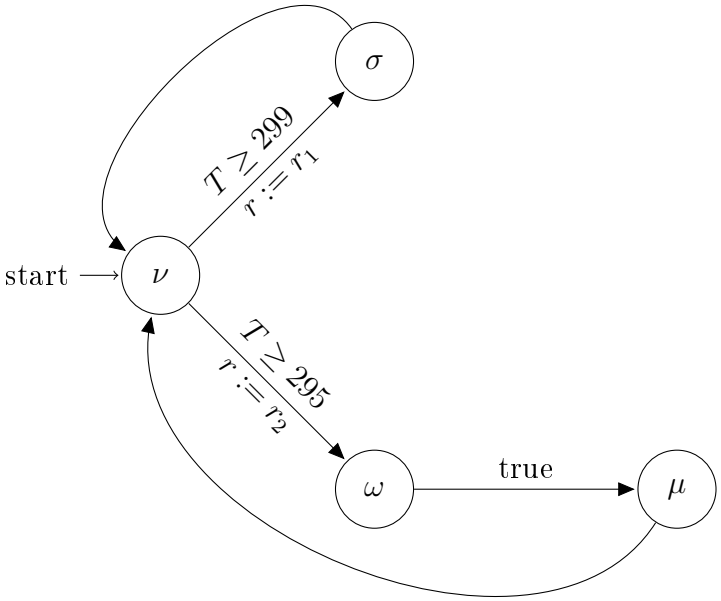
Definicija 4.6. Istinosnu vrijednost formula definiramo rekurzivno, po strukturi formula. Da je formula ϕ istinita u stanju ν u odnosu na pridruživanje η i interpretaciju I označavamo s $I, \eta, \nu \models \phi$ (istinosnu vrijednost formule ψ označavamo s $val_{I,\eta}(\nu, \psi)$).

- $I, \eta, \nu \models R(\theta_1, \ldots, \theta_n)$ ako i samo ako $(val_{I,\eta}(\nu, \theta_1), \ldots, val_{I,\eta}(\nu, \theta_n)) \in I(R)$
- $I, \eta, \nu \models \phi \wedge \psi$ ako i samo ako $I, \eta, \nu \models \phi$ i $I, \eta, \nu \models \psi$
- $I, \eta, \nu \models \phi \vee \psi$ ako i samo ako $I, \eta, \nu \models \phi$ ili $I, \eta, \nu \models \psi$
- $I, \eta, \nu \models \neg\phi$ ako i samo ako $I, \eta, \nu \not\models \phi$
- $I, \eta, \nu \models \phi \rightarrow \psi$ ako i samo ako $I, \eta, \nu \not\models \phi$ ili $I, \eta, \nu \models \psi$
- $I, \eta, \nu \models \forall x \phi$ ako i samo ako $I, \eta[x \mapsto d], \nu \models \phi$ za sve $d \in \mathbb{R}$
- $I, \eta, \nu \models \exists x \phi$ ako i samo ako postoji $d \in \mathbb{R}$ takav da $I, \eta[x \mapsto d], \nu \models \phi$
- $I, \eta, \nu \models [\alpha]\phi$ ako i samo ako $I, \eta, \omega \models \phi$ za sva stanja ω za koja vrijedi $(\nu, \omega) \in \rho_{I,\eta}(\alpha)$

- $I, \eta, \nu \models \langle\alpha\rangle\phi$ ako i samo ako postoji stanje ω takvo da vrijedi $(\nu, \omega) \in \rho_{I,\eta}(\alpha)$ i $I, \eta, \omega \models \phi$

Za **skup formula** Φ kažemo da je **istinit** u stanju ν u odnosu na pridruživanje η i interpretaciju I (i to označavamo s $I, \eta, \nu \models \Phi$) ako vrijedi $I, \eta, \nu \models \phi, \forall \phi \in \Phi$. Za formulu $\phi \in Fml(V, \Sigma)$ kažemo da je **ispunjiva** ako postoji interpretacija I, stanje ν i pridruživanje η za koje vrijedi $I, \eta, \nu \models \phi$. Ako postoje interpretacija I i stanje ν takvi da za sva pridruživanja η vrijedi $I, \eta, \nu \models \phi$, onda kažemo da je (I, ν) **model** za formulu ϕ. Ako za sve I, η, ν vrijedi $I, \eta, \nu \models \phi$, tada formulu ϕ nazivamo **valjanom**.

Primjer 4.2. Prisjetimo se Hibridnog programa 3 iz Primjera 3.4. Neka je veza između T i T_g dana s $T_g \equiv T$ i pokušajmo valuirati formulu iz Primjera 3.6, $T_0 = 290 \rightarrow \langle grijalica\rangle T < 280$.



Slika 5: Skica tijeka Hibridnog programa 3

Interpretacija I dodjeljuje sljedeće vrijednosti nepromjenjivim simbolima r_0 r_1 i r_2: $I(r_0) = 2$, $I(r_1) = -8$, $I(r_2) = -1$. Neka je u početnom stanju ν varijabli stanja T pridijeljena vrijednost $\nu(T) = 290$, a varijabli r vrijednost $\nu(r) = I(r_0) = 2$. Simbol T_0 iz formule koju valuiramo je oznaka za vrijednost varijable stanja T u početnom stanju, dakle $T_0 = 290$. Budući da je sada lijeva strana implikacije istinita, želimo li pokazati da je formula istinita, moramo pronaći izvršavanje programa *grijalica* koje nas vodi do stanja μ takvog da $\mu(T) < 270$.
Na Slici 5 prikazana je struktura izvršavanja programa. Početni čvor označen je

s ν, kao i početno stanje programa. Neka sustav ostane u početnom čvoru jednu minutu. Za to se vrijeme temperatura T mijenjala prema jednadžbi $T' = \frac{2}{t+e}$ čije je rješenje $T(t) = 2ln(t+e) + 288$. Nakon 60 sekundi izvršavanja, temperatura iznosi $T(60) = 296.28K$ i sustav može preći u stanje ω. (U tom trenutku još ne može otići u stanje σ, ali mogao bi još dulje ostati u ν.) Stanje ω se od ν razlikuje u tome što je $\omega(T) = 296.28$ i $\omega(r) = I(r_2) = -1$ U stanju ω temperatura T se smanjuje prema jednadžbi $T' = \frac{-1}{t+e}$ čije je rješenje $T(t) = -ln(t+e) + 297.28$. Nakon dva sata u tom stanju, temperatura T iznosi $T(7200) = 279.51$. Prelaskom u stanje μ dobivamo vrijednost $\mu(T) = 279.51 < 280$ (a time i $val_{I,\eta}(\mu, T) < 280$).
Time smo potpuno odredili istinitost zadane formule. Iako je rješenje bilo vrlo očito (zbog površnog oblikovanja ovog sustava za održavanje temperature), već iz ovog primjera jasno je da je valuacija formula prema definiciji vrlo zamoran i neučinkovit posao. Ne samo to, nego smo ovime vrednovali formulu samo za jednu početnu vrijednost. Voljeli bismo moći vrednovati za sve početne vrijednosti, i za sve moguće nedeterminističke izbore unutar programa. To je motivacija za račun logike $\mathsf{d}\mathcal{L}$ kojeg predstavljamo u Poglavlju 6.

5 ETCS

U ovom poglavlju predstavljamo jednu značajnu primjenu logike $\mathsf{d}\mathcal{L}$. Logika $\mathsf{d}\mathcal{L}$ dala je važan doprinos u analizi sigurnosti europske željezničke mreže. Zato dajemo kratak opis novouspostavljenog zajedničkog europskog standarda, sustava ETCS, a potom uz pomoć logike $\mathsf{d}\mathcal{L}$ analiziramo sigurnosna svojstva na pokaznom primjeru.

5.1 European Train Control System

Mreža željeznica u Europi nije razvijena kao *europska* mreža, već kao više mreža različitih država. Zbog različitih standarda, pri prelasku (nekadašnjih, za zemlje EU) državnih granica potrebno je promijeniti lokomotivu i strojovođu što značajno usporava promet. Usto, razvojem brzih vlakova, signalizacija semaforima na pruzi postala je nedovoljno dobra i teško uočljiva.
Sustav sigurnosti željeznice bio je ranije ostvaren pomoću takozvanih *fiksnih blokova* - dijelova pruge koje je mogao zauzeti samo jedan vlak ili podignuta rampa u danom trenutku. Uz pretpostavku o korektnom funkcioniranju pružnih signala, takav sustav ostvaruje sigurnost ali i bitno usporava promet (zbog statičnosti blokova).
Zbog svega navedenog, Europska je Unija 1991. započela s uvođenjem novog željezničkog standarda za signalizaciju i sigurnost koji će omogućiti veću razinu automatizacije prometa, nazvanog ETCS - *European Train Control System*. Sustav se sastoji od četiri razine (0 - 3), a implementiran je tek djelomično (do razine 2). U nastavku opisujemo svaku razinu standarda:

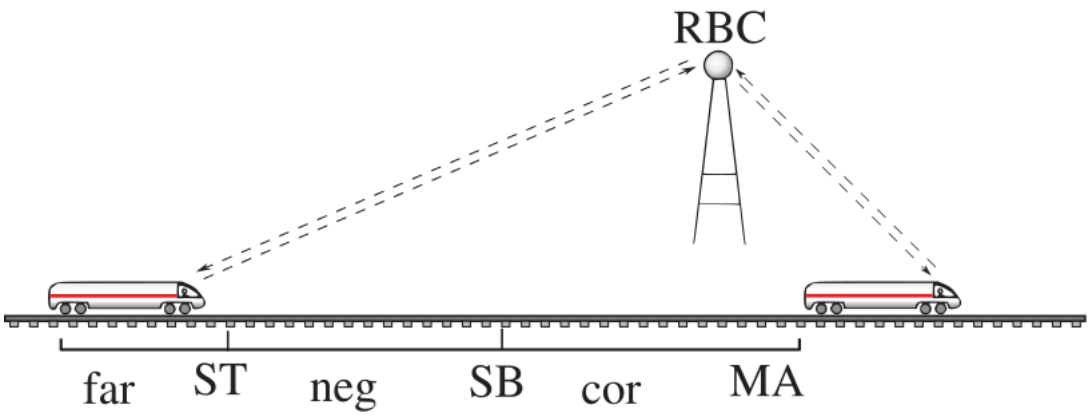
- razina 0
  Odnosi se na vlakove opremljene ETCS opremom koji voze po pruzi bez takve opreme. Tada oprema nadzire samo maksimalnu brzinu vlaka, dok strojovođa upravlja vlakom prema tradicionalnim signalima uz prugu.

- razina 1
  Na ovoj razini se ETCS oprema nadograđuje na postojeću, transformira signale i šalje ih sustavu koji postoji u vlaku. Sustav na temelju primljenih podataka (koji uključuju odobrenje prolaska i rutu) izračunava brzinu te po potrebi vrijeme i snagu kočenja. Budući da se podatci prenose samo prelaskom preko mjesta gdje je signalizacija instalirana, oni se ne dobivaju stalno, već tek po prelasku tzv. *eurobalize*. Razina 1 omogućava prelazak granice između država u kojima isti signali imaju različita značenja bez straha.

- razina 2
  Informacije od pruge do vlaka i obrnuto kontinuirano se prenose radio valovima. Na taj način strojovođa u svakom trenutku u svojoj kabini može vidjeti dokle ima slobodan prolazak. Ipak, blokovi i dalje ostaju fiksni.

- razina 3
  Ova se razina još razvija, a uključuje dinamičke blokove za prolazak.

U Hrvatskoj se razina 1 implementira na pruzi Vinkovci - Tovarnik.



Slika 6: Ilustracija ETCS sustava sigurnosti uz korištenje dinamičkih blokova

5.2 Analiza sigurnosti sustava ETCS pomoću logike d$\mathcal{L}$

Promotrimo idealizirani prikaz ETCS sustava razine 3 i pokušajmo postaviti pitanja o njegovim ključnim sigurnosnim pitanjima jezikom logike d$\mathcal{L}$. Na Slici 6 (preuzetoj iz [1]) dan je shematski prikaz protokola. Svi vlakovi s jednog šireg područja informacije primaju od kontrolne jedinice (RBC - radio block controller) zadužene za to područje. RBC svakom vlaku šalje *dozvolu za prolaz* u obliku najudaljenije točke koju smije dosegnuti, koja je na Slici 6 označena s MA (movement authority). Po primanju te informacije, sustav u vlaku računa od koje točke i koliko treba početi usporavati da bi ostao unutar dozvoljenog područja. Dok se vlak nalazi u području označenom s *far*, može mirno voziti. U točki ST već je dovoljno blizu granici područja za koje ima dozvolu za prolaz pa šalje zahtjev za dobivanjem nove dozvole. Ukoliko je ne dobije, u točki SB vlak počinje usporavati (razlog za neproduljivanje dozvole za kretanje može biti što je neki drugi vlak već dobio dozvolu za isti dio pruge ili što na tom dijelu pruge rampa još nije spuštena). Ako kontrolne jedinice daju dozvole za prolaz korektno (tako da se prolasci vlakova ili podignute rampe međusobno isključuju), onda je za sigurnost dovoljno pokazati da svaki pojedini vlak ostaje unutar granica u kojima ima dozvolu za prolaz. Kako to specificirati jezikom logike d$\mathcal{L}$, navodimo u sljedećem primjeru.

Primjer 5.1. Pretpostavimo da se vlak trenutno nalazi u točki z, a da je dobio dozvolu za prolaz do točke m. Trenutna brzina kretanja vlaka je v. Točku SB (od koje vlak mora početi kočiti) predstavljamo sigurnosnom udaljenošću s koja predstavlja udaljenost točke SB od točke m. Uz ove oznake želimo provjeriti hoće li vlak uvijek ostati unutar dozvoljenog područja kretanja. Ako ćemo ograničenja dana na parametre označiti sa ψ, program koji opisuje komunikaciju vlaka s kontrolnom jedinicom s *ctrl*, a program koji opisuje kretanje vlaka nakon dobivene informacije s

drive, onda je traženo sigurnosno svojstvo opisano $\mathsf{d}\mathcal{L}$ formulom

$$\psi \rightarrow [(ctrl; drive)^*]z \leq m \tag{1}$$

Ona govori da, uz dan raspon parametara iz ψ, nakon svakog niza uzastopnih izvođenja programa *ctrl* i *drive* vlak ostaje unutar dopuštenih granica. Ključno je, naravno, kako ćemo definirati programe *ctrl* i *drive*. Opišimo prvo *ctrl*.

$$ctrl \equiv (?m - z \leq s; a := -b) \cup (?m - z > s; a := A)$$

U ovom se programu radi o jednoj *if-then-else* provjeri: naime, ako je vlak već prešao točku SB ($m - z \leq s$), on počinje kočiti akceleracijom $-b$. U suprotnom, postavlja akceleraciju na pozitivnu vrijednost A. (Kontrola je ovdje prilično gruba: vlak ili usporava maksimalno, ili ubrzava maksimalno. Ipak, lako je zamisliti kako bi se ova formula profinila s tim da vlak bira proizvoljnu vrijednost usporavanja ili ubrzavanja unutar intervala $[-b, 0]$ odnosno $[0, A]$.) Nakon postavljanja akceleracije u *ctrl*, slijedi program *drive* koji se može dizajnirati ovako

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \,\&\, v \geq 0 \wedge \tau \leq \epsilon)$$

Ovaj program opisuje kretanje prema jednadžbi $z'' = a$ koje traje sve dok je brzina v veća od nule. Novouvedena varijabla τ služi ograničavanju trajanja nenadzirane vožnje: na početku programa *drive* ona se inicijalizira na 0, a vožnja može trajati sve dok je $\tau < \epsilon$. Vrijednost konstante definirana je unutar ψ i označava koliko često vlak prima informacije od kontrolne jedinice. Kako bismo mogli provesti realističnu analizu, moramo pretpostaviti da je ϵ najveći mogući i dokazati da i uz takav ϵ formula (1) vrijedi. Parametre zadane u ψ nećemo, kao kod provjere modela (*model checking*) zadavati unaprijed pa provjeriti, već ćemo pomoću računa logike $\mathsf{d}\mathcal{L}$ odrediti sve vrijednosti parametara za koje formula vrijedi. O računu logike $\mathsf{d}\mathcal{L}$ govorimo u sljedećem poglavlju.

6 Račun diferencijalne dinamičke logike

U ovom poglavlju opisujemo račun logike $\mathsf{d}\mathcal{L}$. Pravila računa će nam omogućiti dokazivanje teorema logike $\mathsf{d}\mathcal{L}$. Kao i u računima svih drugih logika, pravila su oblikovana tako da se poklapaju sa semantikom. To važno svojstvo (bez kojeg bi račun bio besmislen) dokazujemo u Poglavlju 7. Račun će se sastojati od propozicijskih pravila i pravila s kvantifikatorima (standardna pravila u računu svih teorija prvog reda), pravila za modalnosti (ta pravila opisuju djelovanja hibridnih programa) i pravila koja uključuju realnu eliminaciju kvantifikatora.

6.1 Supstitucija

Račun koji opisujemo koristi supstitucije - istovremenu zamjenu varijabli (individualnih varijabli ili varijabli stanja) y termom θ pri svakoj pojavi varijable y unutar formule ϕ. Ipak, kako bi definicija supstitucija čuvala istinitost formule, potreban nam je još jedan uvjet - *dopustivost* supstitucije. Pogledajmo sljedeći primjer:

Primjer 6.1. Neka formula ϕ glasi $a \geq b \rightarrow \langle (b := b+1)^* \rangle (b > a)$ i neka supstitucija σ zamjenjuje svaku pojavu varijable b varijablom (istovremeno i termom) a. Tada dobivamo za $\sigma(\phi)$ sljedeću formulu

$$a \geq a \rightarrow \langle (a := a+1)^* \rangle (a > a)$$

Formula ϕ je očito istinita, a $\sigma(\phi)$ očito je lažna. To nije ponašanje koje želimo od supstitucije. U čemu je problem? Unutar hibridnog programa $\langle (a := a+1)^* \rangle$, varijabla a je mijenjana i zato je narav tvrdnje koju izriče ϕ (*broj a veći je ili jednak broju b. Nakon dovoljno povećavanja b, a će biti strogo manji od njega*) potpuno različita od one koju izriče $\sigma(\phi)$ (*a je veći ili jednak sebi samom. Povećamo li ga dovoljno puta, prerast će sam sebe.*)

Kako bismo izbjegli situacije kao u Primjeru 6.1, definiramo kakve supstitucije smatramo poželjnima.

Definicija 6.1. Kažemo da je **primjena supstitucije σ na formulu ϕ dopustiva** ako se nijedna od varijabli x koje σ supstituira sa $\sigma(x)$ ne nalazi u dosegu kvantifikatora ili *modalnog vezivanja* varijable x ili bilo koje varijable terma $\sigma(x)$. Kažemo da modalnost (operatori $[]$ i $\langle\rangle$) **veže** varijablu stanja x ako sadrži diskretan skok koji pridružuje vrijednost varijabli x ili diferencijalnu jednadžbu koja mijenja vrijednost x (npr. $x := \theta$ ili $x' = \theta$).

Odsad, govoreći o supstitucijama, govorimo isključivo o dopustivim primjenama. Tako je i u sljedećoj definiciji u kojoj definiramo primjenu supstitucije na formule logike $\mathsf{d}\mathcal{L}$.

Definicija 6.2. Neka je σ (dopustiva) supstitucija koja simultano zamjenjuje sve pojave varijabli y_i termom θ_i, za $1 \leq i \leq m$ primijenjena na formulu ζ (to označavamo

s $\zeta^{\theta_1,\ldots,\theta_m}_{y_1,\ldots,y_m}$). Djelovanje supstitucije opisujemo induktivno, po svim gradivnim elementima formula logike $\mathsf{d}\mathcal{L}$ - termima, formulama s manjim brojem veznika i hibridnim programima.

- $\sigma(y_i) = \theta_i$, za $1 \leq i \leq m$
- $\sigma(z) = z$, za $z \notin \{y_1, \ldots, y_m\}$
- $\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$, za f funkcijski simbol
- $\sigma(R(t_1, \ldots, t_n)) = R(\sigma(t_1), \ldots, \sigma(t_n))$, za R relacijski simbol
- $\sigma(\neg\phi) = \neg\sigma(\phi)$, za $\phi \in Fml(\Sigma, V)$
- $\sigma(\phi \wedge \psi) = \sigma(\phi) \wedge \sigma(\psi)$, za $\phi, \psi \in Fml(\Sigma, V)$
- $\sigma(\phi \vee \psi) = \sigma(\phi) \vee \sigma(\psi)$, za $\phi, \psi \in Fml(\Sigma, V)$
- $\sigma(\phi \to \psi) = \sigma(\phi) \to \sigma(\psi)$, za $\phi, \psi \in Fml(\Sigma, V)$
- $\sigma(\forall x\phi) = \forall x\sigma(\phi)$, za $\phi \in Fml(\Sigma, V)$
- $\sigma(\exists x\phi) = \exists x\sigma(\phi)$, za $\phi \in Fml(\Sigma, V)$
- $\sigma([\alpha]\phi) = [\sigma(\alpha)]\sigma(\phi)$, za $\alpha \in HP(\Sigma, V), \phi \in Fml(\Sigma, V)$
- $\sigma(\langle\alpha\rangle\phi) = \langle\sigma(\alpha)\rangle\sigma(\phi)$, za $\alpha \in HP(\Sigma, V), \phi \in Fml(\Sigma, V)$
- $\sigma(x_1 := t_1, \ldots, x_n := t_n) = (x_1 := \sigma(t_1), \ldots, x_n := \sigma(t_n))$
- $\sigma(x_1' = t_1, \ldots, x_n' = t_n \,\&\, \chi) = (x_1' = \sigma(t_1), \ldots, x_n' = \sigma(t_n) \,\&\, \sigma(\chi))$
- $\sigma(?\chi) = ?\sigma(\chi)$
- $\sigma(\alpha; \beta) = \sigma(\alpha); \sigma(\beta)$
- $\sigma(\alpha \cup \beta) = \sigma(\alpha) \cup \sigma(\beta)$
- $\sigma(\alpha^*) = \sigma(\alpha)^*$

Napomena 6.1. Dok se u nekim slučajevima nedopustiva primjena supstitucije može lako pretvoriti u dopustivu jednostavnim preimenovanjem vezanih varijabli u formuli u nove, koje se ne koriste u supstituciji (takva je, na primjer, formula $\phi \equiv z > x \wedge y > 1 \wedge x > 0 \to [z := z + xy]z > x$ iz koje preimenovanjem varijable z u varijablu u tamo gdje je vazana dobivamo $\phi' \equiv z > x \wedge y > 1 \wedge x > 0 \to [u := z + xy]u > x$), negdje takvo preimenovanje ne može pomoći (takva je bila formula iz Primjera 6.1).

Priželjkivana veza između (sintaktički) dopustive primjene supstitucije i semantičke modifikacije postoji i to je sadržaj Leme 6.1. Naime, tvrdimo da je jednako hoćemo li formulu ϕ modificirati supstitucijom σ ili ćemo na odgovarajući način promijeniti interpretaciju simbola koje bi σ zamijenila u formuli ϕ.

Lema 6.1. Neka je σ dopustiva supstitucija za term ili formulu ϕ i neka σ supstituira *samo individualne varijable*. Tada za sve I, η, ν vrijedi

$$val_{I,\eta}(\nu, \sigma(\phi)) = val_{I,\sigma^*(\eta)}(\nu, \phi)$$

gdje $\sigma^*(\eta)$ označava pridruživanje koje se definira s $\sigma^*(\eta)(x) := val_{I,\eta}(\nu, \sigma(x))$, za sve $x \in V$.

Dokaz. Ova tvrdnja proizlazi iz činjenice da su supstitucija i valuacija definirane na jednak način: induktivno od djelovanja na osnovne gradivne elemente formula. Takva induktivna izgradnja polazi od varijabli - to je baza koja vrijedi zbog definicije pridruživanja σ^*. Dalje se dokaz provodi indukcijom - prvo po duljini terma, a onda po duljini formule. Budući da se radi o vrlo tehničkom dokazu, ispuštamo ga. □

Lema 6.1 dala je poveznicu između semantike i sintakse supstitucije, ali samo za individualne varijable. Ipak, dvostrukom primjenom te leme sličnu tvrdnju dobivamo i za varijable stanja.

Korolar 6.1. Neka je σ dopustiva primjena supstitucije za term ili formulu ϕ. Tada za sve I, η, ν vrijedi

$$val_{I,\eta}(\nu, \sigma(\phi)) = val_{I,\sigma^*(\eta)}(\sigma^*(\nu), \phi)$$

gdje se $\sigma^*(\nu)$ definira sa $\sigma^*(\nu)(x) = val_{I,\eta}(\nu, \sigma(x))$, za sve varijable stanja $x \in \Sigma$, a $\sigma^*(\eta)$ definira se kao u Lemi 6.1.

Dokaz. Bez smanjenja općenitosti promatramo supstituciju σ koja zamjenjuje samo varijablu stanja x termom θ (prema ranije uvedenim oznakama, $\sigma(\phi) = \phi_x^\theta$). Neka je $z \in V$ individualna varijabla koja se ne pojavljuje u ϕ. Budući da je σ dopustiva za ϕ, x nije vezan u ϕ. Dakle, možemo smatrati da je ϕ oblika ψ_z^x za formulu ψ koja se u svemu poklapa s ϕ osim što sadrži z gdje god ϕ sadrži x. Uvedimo dvije pokrate

- $val_{I,\eta}(\nu, \theta)$ označavamo s e
- $val_{I,\eta}(\nu[x \mapsto e], x)$ označavamo s d

i primijetimo odmah da je $d = e$ (jer izraz koji označavamo s d zapravo govori o valuaciji varijable stanja x u stanju ν koje varijabli x pridružuje baš e). Sada imamo slijed jednakosti

$$\begin{aligned}
val_{I,\eta}(\nu, \phi_x^\theta) &= val_{I,\eta}(\nu, \psi_{z\,x}^{x\,\theta}) && \text{formula } \phi \text{ izražena preko } \psi \\
&= val_{I,\eta}(\nu, \psi_z^\theta) && \text{skraćivanje oznake } x \textit{ stoji umjesto } z, \\
& && \text{a } \theta \text{ stoji umjesto } x \\
& && \text{u oznaku } \theta \textit{ stoji umjesto } z \\
&= val_{I,\eta[z\mapsto e]}(\nu, \psi) && \text{primjena Leme 6.1} \\
&= val_{I,\eta[z\mapsto d]}(\nu[x \mapsto e, \psi) && d = e \text{ i } x \text{ se ne pojavljuje u } \psi \\
&= val_{I,\eta}(\nu[x \mapsto e], \psi_z^x) && \text{primjena Leme 6.1} \\
&= val_{I,\eta}(\nu[x \mapsto e], \phi) \\
&= val_{I,\sigma^*}(\sigma^*(\nu), \phi)
\end{aligned}$$

□

Odavde slijedi jednostavan, a važan korolar - da valjanost ostaje očuvana nakon supstitucije.

Korolar 6.2. Neka je ϕ valjana formula logike $\mathsf{d}\mathcal{L}$, tj. $I, \eta, \nu \models \phi$ za sve I, η, ν. Tada je valjana i formula $\sigma(\phi)$, za bilo koju dopustivu supstituciju σ.

6.2 Pravila računa logike $\mathsf{d}\mathcal{L}$

U ovoj točki predstavljamo sustav pravila za dokazivanje u logici $\mathsf{d}\mathcal{L}$. Pravila dajemo u obliku računa sekvenata (u *Gentzenovu stilu*, prema njemačkom matematičaru Gerhardu Gentzenu, koji je prvi uveo ovo poopćenje prirodne dedukcije). *Sekvent* je oblika $\Gamma \vdash \Delta$ gdje su Γ i Δ konačni skupovi formula. Zapis $\Gamma \vdash \Delta$ znači da ako znamo da vrijede sve formule iz Γ, možemo dokazati jednu od formula iz Δ. Semantički, voljeli bismo da taj zapis bude ekvivalentan s $\bigwedge_{\phi \in \Gamma} \phi \to \bigvee_{\psi \in \Delta} \psi$. (Sva su ova objašnjenja neprecizna i služe samo kako bi se stekla osnovna intuicija - dokazivanje je strogo sintaktičko.) Formule u *antecedensu* Γ i u *konzekvensu* Δ odvajamo zarezom, ali u prvom zarez ima značenje konjunkcije, dok u drugom ima značenje disjunkcije. Promotrimo jednostavan sekvent $\Gamma, \phi \vdash \phi, \Delta$. Taj sekvent mora vrijediti bez obzira na to što su Γ i Δ (tj. iz ϕ možemo dokazati ϕ). Pravi smisao računa je da za kompliciranije sekvente odredi vrijede li ili ne vrijede. Dokazivanje se provodi pomoću stabala dokazivanja[3] gdje su aksiomi u listovima, a tvrdnja koju dokazujemo u korijenu. Tipično pravilo izgleda ovako $(\vee r)$ $\dfrac{\vdash \phi, \psi}{\vdash \phi \vee \psi}$ i čita se *ako možemo dokazati ϕ ili ψ (disjunktivno značenje zareza u konzekvensu), onda možemo dokazati i $\phi \vee \psi$*. Pritom se sekvent iznad crte (u ovom slučaju $\vdash \phi, \psi$)naziva **premisom**, a sekvent ispod crte (ovdje $\vdash \phi \vee \psi$) **konkluzijom**.
Ipak, najčešće ćemo pravila upotrebljavati *unatrag*. Dakle, počinjemo od tvrdnje koju želimo dokazati i pokušavamo *zatvoriti* svaku granu, tj. doći do aksioma. U tom smislu prethodno pravilo čitamo kao *kako bismo dokazali $\phi \vee \psi$ moramo dokazati ϕ ili ψ*.
Za zadavanje računa logike $\mathsf{d}\mathcal{L}$ pomoću pravila, potreban nam je još jedan pomoćni pojam - eliminacija kvantifikatora.

Definicija 6.3. Funkciju $QE\colon Fml(\Sigma, V) \to Fml(\Sigma, V)$ nazivamo **eliminacijom kvantifikatora** ako za svaku formulu $\phi \in Fml(\Sigma, V)$ postoji njoj ekvivalentna formula $QE(\phi) \in Fml(\Sigma, V)$ (tj. $\phi \leftrightarrow QE(\phi)$ je valjana formula) u kojoj nema kvantifikatora te se u njoj ne uvode dodatne slobodne varijable ili funkcijski simboli.

Primjer 6.2. Promatramo formulu koja izriče da postoji nultočka kvadratne jednadžbe s parametrima a, b i c, $\phi \equiv (\exists x(ax^2 + bx + c = 0))$. Tada je $QE(\phi)$ oblika $QE(\phi) \equiv (a \neq 0 \wedge b^2 - 4ac \geq 0) \vee (a = 0 \wedge (b = 0 \to c = 0))$

Prije no što opišemo pravila računa logike $\mathsf{d}\mathcal{L}$, želimo reći kako ta pravila grade dokaze i definirati ključne pojmove kao što su izvodivost ili teorem.

[3] u konkretnom slučaju logike $\mathsf{d}\mathcal{L}$ radit će se o maloj modifikaciji stabla u usmjereni graf

Definicija 6.4. Dokaz u sustavu računa logike $\mathsf{d}\mathcal{L}$ je konačan, acikličan usmjeren graf s jedinstvenim vrhom bez roditelja - korijenom. Vrhovi grafa označeni su sekventima tako da je za svaki vrh skup oznaka njegove djece jednak skupu sekvenata iz premise nekog pravila računa logike $\mathsf{d}\mathcal{L}$, a skup oznaka roditelja te djece mora biti jednak skupu sekvenata konkluzije tog istog pravila. Za formulu ψ kažemo da je **dokaziva** iz konačnog skupa formula Φ (označavamo s $\Phi \vdash_{\mathsf{d}\mathcal{L}} \psi$) ako je korijen označen sa ψ, a svi listovi (vrhovi bez djece) sekventima iz skupa Φ. Kažemo da je ϕ **teorem** računa logike $\mathsf{d}\mathcal{L}$ ako je korijen označen s ϕ, a svi vrhovi bez djece s $(*)$.
Sve usmjerene puteve dokaza (konačnog acikličnog grafa) čiji završni vrh nije označen s $(*)$ ili nekom formulom iz skupa pretpostavki nazivamo **otvorenim granama dokaza**.

U sljedećoj definiciji dajemo sheme pravila računa logike $\mathsf{d}\mathcal{L}$. Iz tih ćemo shema graditi sva moguća pravila računa. Dajemo 32 sheme u obliku sekvenata. S lijeve strane svakog sekventa nalazi se oznaka za tu shemu.

Definicija 6.5. Sheme pravila računa logike $\mathsf{d}\mathcal{L}$ definiraju se s:

$$(\neg r)\ \frac{\phi \vdash}{\vdash \neg\phi} \qquad (\vee r\ )\ \frac{\vdash \phi, \psi}{\phi \vee \psi} \qquad (\wedge r)\ \frac{\vdash \phi \quad \vdash \psi}{\vdash \phi \wedge \psi} \qquad (\to r)\ \frac{\phi \vdash \psi}{\vdash \phi \to \psi} \qquad (\mathrm{ax})\ \frac{*}{\phi \vdash \phi}$$

$$(\neg l)\ \frac{\vdash \phi}{\neg\phi \vdash} \qquad (\vee l)\ \frac{\phi \vdash \quad \psi \vdash}{\phi \vee \psi \vdash} \qquad (\wedge l)\ \frac{\phi, \psi \vdash}{\phi \wedge \psi \vdash} \qquad (\to l)\ \frac{\vdash \phi \quad \psi \vdash}{\phi \to \psi \vdash} \qquad (cut)\ \frac{\vdash \phi \quad \phi \vdash}{\vdash}$$

$$(\langle;\rangle)\ \frac{\langle\alpha\rangle\langle\beta\rangle\phi}{\langle\alpha;\beta\rangle\phi} \qquad (\langle^{*n}\rangle)\ \frac{\phi \vee \langle\alpha\rangle\langle\alpha^*\rangle\phi}{\langle\alpha^*\rangle\phi} \qquad (\langle:=\rangle)\ \frac{\phi^{\theta_1 \ldots \theta_n}_{x_1 \ldots x_n}}{\langle x_1 := \theta_1, \ldots, x_n := \theta_n\rangle}$$

$$([;])\ \frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi} \qquad ([^{*n}])\ \frac{\phi \vee [\alpha][\alpha^*]\phi}{[\alpha^*]\phi} \qquad ([:=])\ \frac{\phi^{\theta_1 \ldots \theta_n}_{x_1 \ldots x_n}}{[x_1 := \theta_1, \ldots, x_n := \theta_n]}$$

$$(\langle\cup\rangle)\ \frac{\langle\alpha\rangle\phi \vee \langle\beta\rangle\phi}{\langle\alpha \cup \beta\rangle\phi} \qquad (\langle?\rangle)\ \frac{\chi \wedge \psi}{\langle?\chi\rangle\psi} \qquad (\langle'\rangle)\ \frac{\exists t \geq 0((\forall t : 0 \leq \tilde{t} \leq t \langle\mathcal{S}_{\tilde{t}}\rangle\chi) \wedge \langle\mathcal{S}_t\rangle\phi)}{\langle x_1' = \theta_1, \ldots, x_n' = \theta_n \,\&\, \chi\rangle\phi}$$

$$([\cup])\ \frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi} \qquad ([?])\ \frac{\chi \to \psi}{[?\chi]\psi} \qquad ([']) \frac{\exists t \geq 0((\forall t : 0 \leq \tilde{t} \leq t [\mathcal{S}_{\tilde{t}}]\chi) \to [\mathcal{S}_t]\phi)}{[x_1' = \theta_1, \ldots, x_n' = \theta_n \,\&\, \chi]\phi}$$

$$(\forall r)\ \frac{\vdash \phi(s(X_1, \ldots, X_n))}{\vdash \forall x \phi(x)} \qquad (\exists r)\ \frac{\vdash \phi(X)}{\vdash \exists x \phi(x)}$$

$$(\exists l)\ \frac{\phi(s(X_1, \ldots, X_n)) \vdash}{\exists x \phi(x) \vdash} \qquad (\forall l)\ \frac{\phi(X) \vdash}{\forall x \phi(x) \vdash}$$

$$(i\forall)\ \frac{\vdash QE(\forall X(\phi(X) \vdash \psi(X)))}{\phi(s(X_1, \ldots, X_n)) \vdash \psi(s(X_1, \ldots, X_n))} \qquad (i\exists)\ \frac{\vdash QE(\exists X \bigwedge_i (\Phi_i \vdash \Psi_i))}{\Phi_1 \vdash \Psi_1 \ \ldots \ \Phi_n \vdash \Psi_n}$$

$$([]gen)\ \frac{\vdash \forall^{\alpha}(\phi \to \psi)}{[\alpha]\phi \vdash [\alpha]\psi} \qquad (\langle\rangle gen)\ \frac{\vdash \forall^{\alpha}(\phi \to \psi)}{\langle\alpha\rangle\phi \vdash \langle\alpha\rangle\psi}$$

$$(ind)\ \frac{\vdash \forall^{\alpha}(\phi \to [\alpha]\phi)}{\phi \vdash [\alpha^*]\phi} \qquad (con)\ \frac{\vdash \forall^{\alpha}\forall v > 0(\rho(v) \to \langle\alpha\rangle\rho(v-1))}{\exists v\rho(v) \vdash \langle\alpha^*\rangle\exists v \leq 0\rho(v)}$$

Pritom:

- primjene svih supstitucija su dopustive
- u pravilima $(\langle' \rangle)$ i $([']) $ t i $\tilde{t}$ su novouvedene individualne varijable, a $\mathcal{S}_t$ je diskretan skok $x_1 := y_1(t), \ldots, x_n := y_n(t)$ gdje su $y_1, \ldots y_n$ rješenja pripadnih diferencijalnih jednadžbi iz konkluzije uz simboličke vrijednosti x_i kao početne (tj. valuacija varijabli x_i prije početka neprekidne evolucije je početna vrijednost).
- u pravilima $(\forall r)$ i $(\exists l)$ simbol s je još neiskorišten Skolemov funkcijski simbol, a $X_1, \ldots, X_n$ su sve slobodne individualne varijable izraza $\forall x\phi(x)$
- u pravilima $(i\forall)$, $(\exists r)$ i $(\forall l)$ X je novouvedena individualna varijabla
- u pravilima $(i\exists)$ i $(i\forall)$ eliminacija kvantifikatora (QE) mora biti definirana za formule u premisi.
- u pravilu $(i\exists)$ varijabla X pojavljuje se samo u granama navedenim u konkluziji. U ostalim otvorenim granama X se ne pojavljuje.
- u pravilu *con* individualna varijabla v ne pojavljuje se u α
- simbol $\forall^{\alpha}$ koji se pojavljuje u pravilima $(\langle\rangle gen)$, $([]gen)$, (ind) i (con) označava univerzalno zatvorenje po svim varijablama stanja vezanim u α (u smislu Definicije 6.1).

Konačno, opisujemo kako iz sheme pravila izvodimo pravila računa logike $\mathsf{d}\mathcal{L}$.

Definicija 6.6. Pravila računa logike $\mathsf{d}\mathcal{L}$ definirana su shemom pravila iz Definicije 6.5 na jedan od sljedeća tri načina:

- za sheme u kojima se pojavljuje simbol $\vdash$ (osim $(i\exists)$) - oblika

$$\frac{\phi_1 \vdash \psi_1 \quad \ldots \quad \phi_n \vdash \psi_n}{\phi_0 \vdash \psi_0}$$

  u pravila računa logike $\mathsf{d}\mathcal{L}$ dodajemo pravila oblika

$$\frac{\Gamma, \langle\mathcal{J}\rangle\phi_1 \vdash \langle\mathcal{J}\rangle\psi_1, \Delta \quad \ldots \quad \Gamma, \langle\mathcal{J}\rangle\phi_n \vdash \langle\mathcal{J}\rangle\psi_n, \Delta}{\Gamma, \langle\mathcal{J}\rangle\phi_0 \vdash \langle\mathcal{J}\rangle\psi_0, \Delta}$$

Pritom su Γ i Δ proizvoljni konačni (moguće i prazni) skupovi dodatnih formula koje čine kontekst i ne mijenjaju se primjenom pravila. $\mathcal{J}$ je diskretan skok (također može biti prazan). $\phi_1, \dots \phi_n, \psi_1, \dots \psi_n$ proizvoljne su formule logike $\mathsf{d}\mathcal{L}$.

- za sheme u kojima se ne pojavljuje simbol $\vdash$, tzv. simetrična pravila oblika $\dfrac{\phi_1}{\phi_0}$ u pravila računa logike $\mathsf{d}\mathcal{L}$ dodajemo pravila oblika

$$\frac{\Gamma \vdash \langle \mathcal{J} \rangle \phi_1, \Delta}{\Gamma \vdash \langle \mathcal{J} \rangle \phi_0, \Delta} \qquad i \qquad \frac{\Gamma, \langle \mathcal{J} \rangle \phi_1 \vdash \Delta}{\Gamma, \langle \mathcal{J} \rangle \phi_0 \vdash \Delta}$$

  i pritom su Γ, Δ konačni skupovi dodatnih formula, $\mathcal{J}$ diskretan skok (svo troje mogu biti prazni), a ϕ_0 i ϕ_1 proizvoljne formule logike $\mathsf{d}\mathcal{L}$.

- ako su $\phi_1 \vdash \psi_1, \dots, \phi_n \vdash \psi_n$ sve otvorene grane dokaza koje sadrže slobodnu varijablu X, tada pravilo iz sheme, $(i\exists)$, postaje pravilo $\mathsf{d}\mathcal{L}$ računa

$$\frac{\vdash QE(\exists X \bigwedge_i (\phi_i \vdash \psi_i))}{\phi_1 \vdash \psi_1 \quad \dots \quad \phi_n \vdash \psi_n}$$

  Pritom su $\phi_1, \dots \phi_n, \psi_1, \dots \psi_n$ proizvoljne formule logike $\mathsf{d}\mathcal{L}$.

Napomena 6.2. Za praktične potrebe koristit ćemo još dvije sheme pravila izvedene iz Definicije 6.5.

$$(ind') \frac{\vdash \phi \qquad \vdash \forall^\alpha(\phi \to [\alpha]\phi) \qquad \vdash \forall^\alpha(\phi \to \psi)}{\vdash [\alpha^*]\psi}$$

$$(con') \frac{\vdash \exists v \rho(v) \qquad \vdash \forall^\alpha \forall v > 0 (\rho(v) \to \langle \alpha \rangle \rho(v-1)) \qquad \vdash \forall^\alpha(\exists v \leq 0 \rho(v) \to \psi)}{\vdash \langle \alpha^* \rangle \psi}$$

Dopustivost gore navedenih pravila lako se dokazuje kombiniranjem pravila $([]gen)$ i (ind) odnosno $(\langle\rangle gen)$ i (con). Za ilustraciju, dajemo dokaz dopustivosti pravila (con').

$$\dfrac{\dfrac{\dfrac{\vdash \forall^\alpha(\forall v > 0)(\rho(v) \to \langle \alpha \rangle \rho(v-1))}{\exists v \rho(v) \vdash \langle \alpha^* \rangle (\exists v \leq 0) \rho(v)}\, con}{\vdash \exists v \rho(v) \to \langle \alpha^* \rangle (\exists v \leq 0) \rho(v)}\, \to r \qquad \dfrac{\vdash \exists v \rho(v) \qquad \dfrac{\vdash \forall^\alpha((\exists v \leq 0)\rho(v) \to \psi)}{\langle \alpha^* \rangle (\exists v \leq 0)\rho(v) \vdash \langle \alpha^* \rangle \psi}\, \langle\rangle gen}{\exists v \rho(v) \to \langle \alpha^* \rangle (\exists v \leq 0)\rho(v) \vdash \langle \alpha^* \rangle \psi}\, \to l}{\vdash \langle \alpha^* \rangle \psi}\, cut$$

Primjer 6.3. Prisjetimo se hibridnog programa za upravljanje vlakom iz Primjera 5.1. Tamo je gibanje vlaka bilo predstavljeno kao uzastopno ponavljanje programa *ctrl* i *drive*, pri čemu su oni bili definirani s

$$ctrl \equiv (?m - z \leq s; a := -b) \cup (?m - z > s; a := A)$$

$$drive \equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \,\&\, v \geq 0 \wedge \tau \leq \epsilon)$$

Uz pretpostavku da je vlak prešao u fazu kočenja ($a := -b$, postavljeno u programu *ctrl* i zatim pokrenut program drive), zanima nas uz koje uvjete je svejedno moguće da vlak pređe granicu m. Pojednostavljeno, zanima nas hibridni program $drive' \equiv \langle z' = v, v' = -b \rangle$ i pitanje vrijedi li (i pod kojim uvjetima) formula $\phi \equiv v \geq 0 \wedge z < m \rightarrow \langle drive' \rangle z > m$. Analizu provodimo pomoću pravila računa koja smo uveli u prethodnim definicijama.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{v \geq 0, z < m \vdash v^2 > 2b(m-z)}\,\rightarrow r, \wedge l}{\vdash v \geq 0 \wedge z < m \rightarrow v^2 > 2b(m-z)}}{v \geq 0, z < m \vdash T \geq 0} \quad \dfrac{\dfrac{}{v \geq 0, z < m \vdash -\frac{b}{2}T^2 + vT + z > m}}{v \geq 0, z < m \vdash \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > m}\,\langle := \rangle}{v \geq 0, z < m \vdash T \geq 0 \wedge \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > m}\,\wedge r}{v \geq 0, z < m \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > m}\,\exists r}{v \geq 0, z < m \vdash \langle z' = v, v' = -b \rangle z > m}\,\langle' \rangle}{\vdash v \geq 0 \wedge z < m \rightarrow \langle z' = v, v' = b \rangle z > m}\,\rightarrow r, \wedge l$$

Analizu započinjemo odozdo, od tvrdnje $\vdash \phi$. Njome izričemo: *moguće je dokazati da postoji situacija u kojoj će vlak, iako koči i nije još prešao točku m, nakon nekog vremena preći točku m i tako izići iz sigurnog područja.* Prvo se primjenom pravila $\rightarrow r$ i $\wedge l$ početna tvrdnja preinačuje pa se u antecedensu nalaze uvjeti, a u konzekvensu neprekidna evolucija položaja z. Pravilo $\langle' \rangle$ zamjenjuje diferencijalnu jednadžbu njenim rješenjem. Egzistencijalni kvantifikator ne možemo eliminirati s QE jer je kvantificirana formula modalna (sadrži *diamond* operator). Ipak, pravilo $\exists r$ nam omogućuje da uvedemo varijablu T kao svjedoka za $\exists t$. Pravilo $\wedge r$ grana stablo dokaza u dvije grane. Nakon što smo pravilom $\langle := \rangle$ eliminiralli modalnost iz desne grane, možemo primijeniti $i\exists$, spojiti obje grane u kojima se pojavljuje T i eliminirati kvanitifikator. Konačno, uz $\exists r, \wedge l$ dolazimo do $v \geq 0, z < m \vdash v^2 > 2b(m-z)$. To nam govori da će se neželjeni scenarij (prelazak točke m) dogoditi ako kvadrat brzine prijeđe umnožak dvostruke akceleracijske konstante i udaljenosti do točke m. Iz ovog primjera vidimo da se upotrebom računa lakše dolazi do puno općenitijih zaključaka nego što smo to mogli ranije.

Sve rečeno u Primjeru 6.3 ne vrijedi mnogo ako ne dokažemo teorem adekvatnosti, tj. da je sve što možemo dokazati računom logike $\mathsf{d}\mathcal{L}$ zapravo istina. O tome će riječi biti u sljedećem poglavlju.

7 Adekvatnost i potpunost sistema d$\mathcal{L}$

U ovom poglavlju dajemo odgovore na ključna pitanja kod uvođenja logike. Prvo, jesu li sve formule koje možemo dokazati pomoću računa logike d$\mathcal{L}$ istinite? Drugo, možemo li sve istinite formule dokazati računom? Pokazat ćemo da je zaista sve što dokazujemo istinito (da je račun logike d$\mathcal{L}$ adekvatan). Pokazat ćemo, ipak, i da ne možemo dokazati sve istinite formule (da račun nije potpun). To bi samo po sebi bilo vrlo loše za d$\mathcal{L}$, ali ne ostajemo na tome: ako i ne možemo dokazati sve istinite tvrdnje uz pomoć računa d$\mathcal{L}$, možemo ih dokazati ako dodamo kao aksiome sve istinite tvrdnje o rješenjima diferencijalnih jednadžbi.

7.1 Teorem adekvatnosti

U ovoj točki dokazujemo teorem adekvatnosti. Okvirno, za svako ćemo pravilo dokazati da je adekvatno, tj. da je konkluzija logička posljedica premise. Sve ove intuitivno jasne pojmove (adekvatan račun, adekvatno pravilo, logička posljedica, istinosna vrijednost sekventa) prvo definiramo.

Definicija 7.1. Za formule $\phi, \psi \in Fml(V, \Sigma)$ kažemo da je ψ **logička posljedica** formule ϕ ako iz činjenice da za sve I, ν postoji η takva da $I, \eta, \nu \models \phi$ slijedi da za sve I', ν' postoji η' takva da $I', \eta', \nu' \models \psi$. Za skupove formula Φ i Ψ kažemo da je skup formula Ψ logička posljedica skupa formula Φ ako je $\bigwedge_{\psi \in \Psi} \psi$ logička posljedica formule $\bigwedge_{\phi \in \Phi} \phi$.

Definicija 7.2. Istinosna vrijednost sekventa $\Phi \vdash \Psi$ definirana je kao istinosna vrijednost formule $\bigwedge_{\phi \in \Phi} \phi \to \bigvee_{\psi \in \Psi} \psi$. Ako je sekvent $\Phi \vdash \Psi$ istinit u stanju ν obzirom na interpretaciju I i pridruživanje η, to označavamo s $I, \eta, \nu \models \Phi \vdash \Psi$. Za sheme sekvenata iz Definicije 6.5 oblika $\Phi \vdash$ istinosnu vrijednost promatramo kao da stoji $\Phi \vdash \bot$, a za one oblika $\vdash \Phi$ kao da stoji $\top \vdash \Phi$ (gdje simbol $\top$ uobičajeno stoji za valjanu formulu, a simbol $\bot$ za formulu koja je neistinita za sve I, η, ν).

Napomena 7.1. Obzirom na definiciju istinosne vrijednosti sekventa, za sve semantičke kategorije (kao što su one definirane u Definiciji 7.1 i Definiciji 4.6) poistovjećujemo sekvent i formulu.

U tekstu koji slijedi definiramo još pojmove adekvatnog pravila, lokalno adekvatnog pravila i adekvatnog računa.

Definicija 7.3. Kažemo da je pravilo računa logike d$\mathcal{L}$

$$\frac{\Phi}{\Psi} = \frac{\phi_1 \quad \phi_2 \quad \dots \quad \phi_n}{\psi_1 \quad \psi_2 \quad \dots \quad \psi_m}$$

adekvatno ako je skup formula konkluzije logička posljedica skupa formula premise. Za isto pravilo kažemo da je **lokalno adekvatno** ako za sve I, η i ν vrijedi da iz $I, \eta, \nu \models \Phi$ slijedi $I, \eta, \nu \models \Psi$.

Kako smo najavili, slijedi dokaz teorema adekvatnosti. Za svaku od shema pravila prikazanih u Definiciji 6.5 dokazujemo da je adekvatna (kako bismo pojednostavili izražavanje, reći ćemo *pravilo je adekvatno*). Ustvari, za sve osim $\forall r, \exists l, i\exists$ moći ćemo dokazati i više, da je pravilo lokalno adekvatno. Ipak, da bi dokaz bio korektan, prije svega dokazujemo jednostavnu propoziciju koja kaže da iz adekvatnosti sheme pravila slijedi adekvatnost pravila računa logike $\mathsf{d}\mathcal{L}$ (s dodanim kontekstom).

Lema 7.1. Ako je shema pravila logike $\mathsf{d}\mathcal{L}$ (lokalno) adekvatna, onda je (lokalno) adekvatno i pripadno pravilo logike $\mathsf{d}\mathcal{L}$

Dokaz. Iz Definicije 6.6 znamo da se pravila dobivaju iz sheme tako da se dodaju formule konteksta u skupovima Γ i Δ i hibridni program s diskretnim skokom. Promatrali smo dvije mogućnosti:

(1) $\dfrac{\phi_1 \vdash \psi_1 \quad \dots \quad \phi_n \vdash \psi_n}{\phi_0 \vdash \psi_0}$ prelazi u
$$\frac{\Gamma, \langle\mathcal{J}\rangle\phi_1 \vdash \langle\mathcal{J}\rangle\psi_1, \Delta \quad \dots \quad \Gamma, \langle\mathcal{J}\rangle\phi_n \vdash \langle\mathcal{J}\rangle\psi_n, \Delta}{\Gamma, \langle\mathcal{J}\rangle\phi_0 \vdash \langle\mathcal{J}\rangle\psi_0, \Delta}$$

(2) $\dfrac{\phi_1}{\phi_0}$ prelazi u $\dfrac{\Gamma \vdash \langle\mathcal{J}\rangle\phi_1, \Delta}{\Gamma \vdash \langle\mathcal{J}\rangle\phi_0, \Delta}$ ili $\dfrac{\Gamma, \langle\mathcal{J}\rangle\phi_1 \vdash \Delta}{\Gamma, \langle\mathcal{J}\rangle\phi_0 \vdash \Delta}$

U oba slučaja kontekst Γ, Δ ne može promijeniti adekvatnost pravila zbog konjunktivnog nizanja pravila u antecedensu i disjunktivnog u konzekvensu sekventa: ako je antecedens neistinit, ostaje neistinit bez obzira na dodani kontekst; ako je pak konzekvens istinit, on ostaje istinit bez obzira na dodani kokntekst. Ta dva slučaja iscrpljuju mogućnosti da sekvent u premisi bude istinit. Sekventi ostaju adekvatni i lokalno adekvatni i dodavanjem programa s diskretnim skokom $\langle\mathcal{J}\rangle$ jer on samo mijenja stanje ν, koje je i u definiciji lokalne adekvatnosti i adekvatnosti univerzalno kvantificirano. □

Izravno iz definicija adekvatnosti i lokalne adekvatnosti slijedi tvrdnja sljedeće leme.

Lema 7.2. Lokalno adekvatno pravilo je i adekvatno.

Nakon svih pomoćnih tvrdnji koje smo dokazali, slijedi teorem adekvatnosti. Ideju dokaza već smo izložili - pokazati da je svaka od shema pravila adekvatna. Ipak, prezentirat ćemo dokaze samo za neke sheme (za ostale se dokazi provode na sličan način).

Teorem 7.1. Račun logike $\mathsf{d}\mathcal{L}$ je adekvatan - ako formulu ϕ možemo dokazati iz skupa formula Ψ, onda je ϕ logička posljedica skupa Ψ.

Dokaz. Želimo prvo dokazati da je svako pravilo računa adekvatno. Za to će nam, prema Lemi 7.1 biti dovoljno dokazati da je svako pravilo iz shema pravila adekvatno (za neka ćemo dokazati i da su lokalno adekvatna).
Prva skupina pravila koja promatramo su tzv. propozicijska pravila (od $(\neg r)$ do (cut)). Ona nisu specifična za račun logike $\mathsf{d}\mathcal{L}$ i njihova se adekvatnost dokazuje kako je uobičajeno. Za ilustraciju dokazujemo lokalnu adekvatnost pravila $(\vee r)\,\dfrac{\vdash \phi, \psi}{\vdash \phi \vee \psi}$. Pretpostavimo da su I, ν, η takvi da je $I, \eta, \nu \models \vdash \phi, \psi$. Iz disjunkcijskog tumačenja zareza u konzekvensu i tumačenja praznog mjesta s lijeve strane sekventa kao $\top$ (kako je opisano u Definiciji 7.2) zaključujemo da vrijedi $I, \eta, \nu \models \phi$ ili $I, \eta, \nu \models \psi$. Ali, to znači da vrijedi i $I, \eta, \nu \models \phi \vee \psi$, a onda i $I, \eta, \nu \models \vdash \phi \vee \psi$. Dakle, pravilo $(\vee r)$ je lokalno adekvatno.
Druga skupina pravila odnosi se na pravila u koja su uključeni hibridni programi izuzev neprekidne evolucije (tj. na pravila s oznakama $(\langle;\rangle)$, $([;])$, $(\langle^{*n}\rangle)$, $([^{*n}])$, $(\langle:=\rangle)$, $([:=])$, $(\langle\cup\rangle)$, $([\cup])$, $(\langle?\rangle)$, $([?])$). Za ilustraciju dokazujemo da su pravila $(\langle;\rangle)\,\dfrac{\langle\alpha\rangle\langle\beta\rangle\phi}{\langle\alpha;\beta\rangle\phi}$ i $([;])\,\dfrac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$ lokalno adekvatna.
Pretpostavimo da vrijedi $I, \eta, \nu \models \langle\alpha\rangle\langle\beta\rangle\phi$. To znači da postoji stanje μ takvo da je $(\nu, \mu) \in \rho_{I,\eta}(\alpha)$ i $I, \eta, \mu \models \langle\beta\rangle\phi$. Iz toga pak zaključujemo da postoji stanje ω takvo da vrijedi $(\mu, \omega) \in \rho_{I,\eta}(\beta)$ i $I, \eta, \omega \models \phi$. Sada iz točke (5) Definicije 4.3 zaključujemo da vrijedi $(\nu, \omega) \in \rho_{I,\eta}(\alpha;\beta)$. Budući da otprije imamo $I, \eta, \omega \models \phi$, zaključujemo da vrijedi $I, \eta, \nu \models \langle\alpha;\beta\rangle\phi$. Na sličan se način dokazuje i $\dfrac{\langle\alpha;\beta\rangle}{\langle\alpha\rangle\langle\beta\rangle\phi}$, svojevrsni obrat pravila čiju smo lokalnu adekvatnost upravo dokazali. Sada pravilo $([;])$ dokazujemo obratom po kontrapoziciji. Neka za I, η, ν vrijedi $I, \eta, \nu \not\models [\alpha;\beta]\phi$. Ponovno iz Definicije 4.3 zaključujemo $I, \eta, \nu \models \langle\alpha;\beta\rangle\neg\phi$. Iz obrata pravila $(\langle;\rangle)$ zaključujemo $I, \eta, \nu \models \langle\alpha\rangle\langle\beta\rangle\neg\phi$ i na kraju, opet po definiciji, $I, \eta, \nu \models \langle\alpha\rangle\neg([\beta]\phi$ i $I, \eta, \nu \not\models [\alpha][\beta]\phi$.
Sljedeća grupa pravila su pravila neprekidne evolucije (dakle, pravila s oznakama $(\langle'\rangle)$, $([′])$). Za ilustraciju dokazujemo da je pravilo

$$(\langle'\rangle)\,\frac{\exists t \geq 0((\forall t : 0 \leq \tilde{t} \leq t\langle\mathcal{S}_{\tilde{t}}\rangle\chi) \wedge \langle\mathcal{S}_t\rangle\phi)}{\langle x_1' = \theta_1, \ldots, x_n' = \theta_n \,\&\, \chi\rangle\phi}$$

lokalno adekvatno. Neka su $y_1, \ldots, y_n$ rješenja sustava diferencijalnih jednadžbi $x_1' = \theta_1, \ldots, x_n' = \theta_n$ sa simboličkim početnim vrijednostima[4] $x_1, \ldots, x_n$. Sjetimo se da je $\mathcal{S}_t$ diskretan skok $x_1 := y_1(t), \ldots, x_n := y_n(t)$. Neka je $\bar{\chi}$ skraćeni zapis za $\forall\tilde{t} : 0 \leq \tilde{t} \leq t\langle\mathcal{S}_{\tilde{t}}\rangle\chi$. Također, neka je η_t^{ζ} pridruživanje koje je u svemu jednako pridruživanju η, osim što varijabli t pridružuje ζ. Pretpostavljamo da postoje I, η, ν takvi da vrijedi $I, \eta, \nu \models \exists t \geq 0((\forall t : 0 \leq \tilde{t} \leq t\langle\mathcal{S}_{\tilde{t}}\rangle\chi) \wedge \langle\mathcal{S}_t\rangle\phi)$. To znači da vrijedi $I, \eta_t^r, \nu \models \bar{\chi} \wedge \langle\mathcal{S}_t\rangle\phi$. Neka je $\mathcal{D}$ pokrata za $x_1' = \theta_1, \ldots, x_n' = \theta_n \,\&\, \chi$. Moramo, dakle, pokazati da je $I, \eta, \nu \models \langle\mathcal{D}\rangle\phi$. Ekvivalentno tome je pokazati $I, \eta_t^r, \nu \models \langle\mathcal{D}\rangle\phi$ (jer je t novouvedena

[4] podsjećamo, to znači da se za početne vrijednosti uzimaju valuacije varijabli $x_1, \ldots, x_n$ u stanju u kojem započinje hibridni program $\langle x_1' = \theta_1, \ldots, x_n' = \theta_n\rangle$

varijabla koja se ne pojavljuje ni u $\mathcal{D}$ ni u ϕ). Da bismo to dokazali, moramo pronaći funkciju koja bi *svjedočila* da postoji stanje ω takvo da je $\nu, \omega \in \rho_{I,\eta_t^r}(\mathcal{D})$ kako je opisano u točki (2) Definicije 4.3. Definiramo funkciju $f\colon [0,r] \to Sta$ tako da je $(\nu, f(\zeta)) \in \rho_{I,\eta_t^\zeta}(\mathcal{S}_t)$. Prema pretpostavci, $f(0) = \nu$ i ϕ vrijedi u $f(r)$. Funkcija $f(\zeta)(x_i), \forall x_i \forall \zeta$ je neprekidna u ζ na segmentu $[0,r]$ i diferencijabilna po ζ na intervalu $\langle 0, r\rangle$ jer su takve i funkcije y_i a vrijedi $val_{I,\eta_t^r}(f(\zeta), x_i) = val_{I,\eta_t^r}(\nu, y_i(t))$. Budući da je y_i rješenje diferencijalne jednadžbe $x_i' = \theta_i$ s pripadnom početnom vrijednosti $\nu(x_i)$ znamo da je derivacija od $val_{I,\eta_t^r}(f(\zeta), x_i)$ jednaka $val_{I,\eta_t^r}(f(\zeta), \theta_i)$. Iz pretpostavke slijedi da je $I, \eta_t^r, \nu \models \bar{\chi}$ pa je onda i $I, \eta_t^r, f(\zeta) \models \chi, \forall \zeta \in [0,r]$. Dakle, zaista je f tražena funkcija.
Sljedeća skupina pravila su pravila s kvantifikatorima (tj. pravila s oznakama $(\forall r)$, $(\exists r)$, $(\exists l)$, $(\forall l)$, $(i\forall)$, $(i\exists)$). Dajemo dokaz adekvatnosti pravila

$$(i\exists)\ \frac{\vdash QE(\exists X \bigwedge_i (\Phi_i \vdash \Psi_i))}{\Phi_1 \vdash \Psi_1\ \ldots\ \Phi_n \vdash \Psi_n}$$

Neka je za proizvoljne I, ν pridruživanje η takvo da je
$I, \eta, \nu \models QE(\exists X \bigwedge\limits_i (\Phi_i \vdash \Psi_i))$. Iz definicije operacije QE znamo da ona daje formulu ekvivalentnu onoj na kojoj je primijenjena pa slijedi $I, \eta, \nu \models \exists X \bigwedge\limits_i (\Phi_i \vdash \Psi_i)$. Neka je d svjedok za egzistencijalni kvantifikator iz formule. Znamo da se X ne pojavljuje slobodan nigdje u premisi (jer pravilo $i\exists$ djeluje na sve slobodne nastupe varijable X). To znači da vrijednost varijable X ne mijenja istinosnu vrijednost premise. Ako definiramo η' kao pridruživanje koje je u svemu jednako η osim što je $\eta(X) = d$, vrijedi $I, \eta', \nu \models \bigwedge\limits_i (\Phi_i \vdash \Psi_i)$, a onda i $I, \eta', \nu \models \Phi_1 \vdash \Psi_1\ \ldots\ \Phi_n \vdash \Psi_n$. Uspjeli smo, dakle, za proizvoljne I, ν konstruirati η' tako da sve konkluzije vrijede. Primijetimo još da smo ovime dokazali adekvatnost (ne i lokalnu adekvatnost) pravila $(i\exists)$.
Posljednja skupina pravila su takozvana *globalna pravila*: $(\langle\rangle gen)$, $([]gen)$, (ind) i (con). Za ilustraciju dajemo dokaz da je pravilo $(\langle\rangle gen)\ \frac{\vdash \forall^\alpha(\phi \to \psi)}{\langle\alpha\rangle\phi \vdash \langle\alpha\rangle\psi}$ lokalno adekvatno. Neka su I, η, ν takvi da $I, \eta, \nu \models \forall^\alpha(\phi \to \psi)$. Budući da $\forall^\alpha$ kvantificira po svim varijablama koje se mogu promijeniti izvršavanjem programa α, vrijedi i $I, \eta, \mu \models \phi \to \psi, \forall \mu \in \rho_{I,\eta}(\alpha)$. Pretpostavimo sada i da $I, \eta, \nu \models \langle\alpha\rangle\phi$. To znači da postoji ν' takav da $(\nu, \nu') \in \rho_{I,\eta}(\alpha)$ i $I, \eta, \nu' \models \phi \to \psi$. Sada slijedi $I, \eta, \nu' \models \psi$, a onda i $I, \eta, \nu \models \langle\alpha\rangle\psi$.
Iz upravo dokazane adekvatnosti svih shema pravila, zaključujemo da su sva pravila logike $\mathsf{d}\mathcal{L}$ adekvatna. Budući da je relacija logičke posljedice tranzitivna, slijedi tvrdnja teorema. □

7.2 Nepotpunost logike $\mathsf{d}\mathcal{L}$

U prethodnoj točki pokazali smo da je sve što možemo dokazati istinito. Jednako važno pitanje je i možemo li dokazati sve što je istinito, to jest je li račun logike $\mathsf{d}\mathcal{L}$

potpun. Pokazat će se da tome nije tako. Dokaz nepotpunosti svodi se na definiranje prirodnih brojeva unutar logike $\mathsf{d}\mathcal{L}$, a zatim tvrdnja slijedi uz pomoć Gödelovog prvog teorema nepotpunosti. Ipak, iako $\mathsf{d}\mathcal{L}$ nije potpuna, na kraju ovog poglavlja dokazujemo da je relativno potpuna obzirom na svojstva rješenja diferencijalnih jednadžbi. Za početak navodimo Gödelov prvi teorem nepotpunosti (zapravo korolar koji nam je potreban u ovom kontekstu).

Teorem 7.2. Niti za jednu logiku koja proširuje aritmetiku prirodnih brojeva ne postoji efektivan račun koji je istovremeno adekvatan i potpun.

U teoremu koji slijedi dokazujemo da se prirodni brojevi mogu definirati unutar logike $\mathsf{d}\mathcal{L}$. (Iz dokaza će biti jasno da se to može učiniti i u diskretnom i u kontinuiranom fragmentu logike $\mathsf{d}\mathcal{L}$. Neformalno koristimo pojmove *diskretni* i *kontinuirani* fragment. Oni se odnose na korištenje diferencijalnih jednadžbi u kontinuiranom fragmentu ili njihovo nekorištenje u diskretnom fragmentu.)

Teorem 7.3. Unutar logike $\mathsf{d}\mathcal{L}$ moguće je definirati prirodne brojeve.

Dokaz. Prirodni brojevi mogu se definirati korištenjem uzastopnog pribrajanja:

$$nat(n) \leftrightarrow \langle x := 0; (x := x + 1)^* \rangle x = n$$

Oni se također mogu definirati pomoću diferencijalnih jednadžbi; koristit ćemo jednadžbe koje opisuju trigonometrijske funkcije i pomoću njihovih nultočaka definirati izomorfnu presliku skupa $\mathbb{N}$.

$$nat(n) \leftrightarrow \exists s \exists c \exists \tau (s = 0 \wedge c = 1 \wedge \tau = 0 \wedge \langle s' = c, c' = -s, \tau' = 1 \rangle (s = 0 \wedge \tau = n))$$

□

Teorem nepotpunosti sada slijedi kao jednostavna posljedica Teorema 7.1, Teorema 7.2 i teorema 7.3.

Teorem 7.4. Račun logike $\mathsf{d}\mathcal{L}$ nije potpun - postoje valjane formule koje pomoću njega ne možemo dokazati.

7.2.1 Relativna potpunost logike $\mathsf{d}\mathcal{L}$

U ovoj podtočki želimo dati odgovor na pitanje: ako račun logike $\mathsf{d}\mathcal{L}$ nije potpun, možemo li ga nekako proširiti da postane potpun? Pokazat će se da možemo; ako u račun logike $\mathsf{d}\mathcal{L}$ dodamo kao aksiome sve valjane formule koje govore o svojstvima diferencijalnih jednadžbi, taj će račun biti potpun. U sljedećoj definiciji formaliziramo tu namjeru. Definirat ćemo logiku diferencijalnih jednadžbi koja se u svemu poklapa s logikom $\mathsf{d}\mathcal{L}$, osim što se njene formule definiraju promjenom točke 4 iz originalne Definicije 3.7.

Definicija 7.4. Skup formula logike diferencijalnih jednadžbi
(u oznaci $Fml_{FOD}(\Sigma, V)$) definira se s:

1. ako je $R \in \Sigma$ neki n-mjesni relacijski simbol, $\theta_i \in Trm(\Sigma, V)$, $\forall\, 1 \leq i \leq n$, onda je $R(\theta_1, \theta_2, \ldots, \theta_n) \in Fml_{FOD}(\Sigma, V)$

2. ako su $\phi, \psi \in Fml_{FOD}(\Sigma, V)$, onda su i $\neg\phi, (\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi), (\phi \leftrightarrow \psi) \in Fml_{FOD}(\Sigma, V)$

3. ako je $\phi \in Fml_{FOD}(\Sigma, V)$ i $x \in V$, onda su $(\forall x\phi), (\exists x\phi) \in Fml_{FOD}(\Sigma, V)$

4. ako je $\phi \in Fml_{FOD}(\Sigma, V)$ tada je i $[x'_1 = \theta_1, \ldots, x'_n = \theta_n]\phi \in Fml_{FOD}(\Sigma, V)$

Napomena 7.2. Primijetimo da je razlika u odnosu na definiciju skupa $Fml(\Sigma, V)$ jedino u točki 4; dok smo u definiciji formula iz $Fml(\Sigma, V)$ dopuštali formule oblika $[\alpha]\phi$ i $\langle\alpha\rangle\phi$ za proizvoljan hibridni program α, u definiciji skupa formula logike diferencijalnih jednadžbi dozvoljavamo samo formule s točno određenim hibridnom programom - neprekidnom evolucijom.
Kao dio skupa Fml_{FOD} uzimamo i formule oblika $\langle x'_1 = \theta_1, \ldots, x'_n = \theta_n\rangle\phi$ jer se radi o pokrati za $\neg[x'_1 = \theta_1, \ldots, x'_n = \theta_n]\neg\phi$

U ostatku poglavlja donosimo dokaz relativne potpunosti logike $\mathsf{d}\mathcal{L}$ s obzirom na FOD. Dokaz je konstruktivan i, u dijelovima, vrlo tehnički. Zato ćemo većinu lema i propozicija navesti bez dokaza, a dokazat ćemo i komentirati samo ključne dijelove.

Sljedeća lema tvrdi da za svaku formulu iz $\mathsf{d}\mathcal{L}$ postoji njoj ekvivalentna formula unutar logike FOD. (Obrnuto vrijedi trivijalno jer je FOD sadržana u $\mathsf{d}\mathcal{L}$.)

Lema 7.3. Za svaku formulu $\phi \in Fml(\Sigma, V)$ logike $\mathsf{d}\mathcal{L}$ postoji formula $\phi^{FOD} \in Fml_{FOD}(\Sigma, V)$, takva da vrijedi $\models \phi \leftrightarrow \phi^{FOD}$.

U definiciji koja slijedi preciziramo što podrazumijevamo kad kažemo da *račun logike $\mathsf{d}\mathcal{L}$ proširujemo aksiomima sa svojstvima rješenja diferencijalnih jednadžbi.*

Definicija 7.5. Kažemo da se formula $\phi \in Fml_{FOD}$ može izvesti uz pomoć računa logike $\mathsf{d}\mathcal{L}$ obogaćenog tautologijama logike FOD ako se ϕ može izvesti pomoću računa logike $\mathsf{d}\mathcal{L}$ kojemu su dodane sve valjane formule logike FOD kao aksiomi. To označavamo s $\vdash_{\mathsf{d}\mathcal{L}^+} \phi$. Slično, ako se sekvent $\Gamma \vdash \Delta$ može izvesti iz tog računa, koristimo oznaku $\Gamma \vdash_{\mathsf{d}\mathcal{L}^+} \Delta$.

Napomena 7.3. Prethodna definicija opisuje račun $\mathsf{d}\mathcal{L}^+$. On se konkretno dobiva tako da pravilima računa logike $\mathsf{d}\mathcal{L}$ pridodamo kao aksiome sve valjane formule logike FOD. Dakle, ako je ψ valjana formula logike FOD, onda u shemu pravila dodajemo $(ax)\,\dfrac{*}{\psi}$.

U nastavku dokaza kvantifikaciju ćemo shvaćati kao pokratu za hibridne programe: egzistencijalni kvantifikator može se prikazati pomoću hibridnog programa kao $\exists x\phi \equiv \langle x' = 1\rangle\phi \vee \langle x' = -1\rangle\phi$, a univerzalni kao $\forall x\phi \equiv [x' = 1]\phi \wedge [x' = -1]\phi$. Također, pretpostavljat ćemo da formule ne sadrže slobodne individualne varijable (umjesto njih neka sadrže varijable stanja). Zato će nam i iduća lema biti korisna.

Lema 7.4. Ako vrijedi $\vdash_{\mathsf{d}\mathcal{L}^+} \phi$ bez korištenja slobodnih individualnih varijabli u dokazu, onda vrijedi i $\vdash_{\mathsf{d}\mathcal{L}^+} \forall x\phi$ i $\vdash_{\mathsf{d}\mathcal{L}^+} \langle x_1 := \theta_1, \ldots, x_n := \theta_n\rangle\phi$

Sljedeća propozicija govori da su relativno potpuna tzv. svojstva sigurnosti. Radi se o formulama oblika $F \to [\alpha]G$ koja se mogu tumačiti ovako: *ako vrijedi F, bez obzira što se događalo u programu* α*, G će uvijek vrijediti.* Motivacija za naziv *svojstva sigurnosti* dolazi od toga što su takva svojstva (koja uvijek vrijede) poželjna za stroge uvjete sigurnosti (na primjer, *vlak će uvijek ostati unutar dozvoljenih granica*). Prije najavljene propozicije, iskazujemo lemu koja će biti od koristi za dokazivanje unutar računa $\mathsf{d}\mathcal{L}^+$

Lema 7.5.

$$\vdash_{\mathsf{d}\mathcal{L}^+} \phi \to \psi \text{ ako i samo ako } \phi \vdash_{\mathsf{d}\mathcal{L}^+} \psi$$

Primijetimo da je jedan smjer prethodne leme zapravo *teorem dedukcije* za račun $\mathsf{d}\mathcal{L}^+$. Teorem dedukcije slijedi izravnom primjenom pravila $(\to r)$. Obrnuti smjer dobiva se kombinacijom pravila (cut) i $(\to l)$.

Propozicija 7.1. Za svaki hibridni program $\alpha \in HP(\Sigma, V)$ i sve $F, G \in Fml_{FOD}(\Sigma, V)$ formule logike FOD vrijedi da $\models F \to [\alpha]G$ povlači $\vdash_{\mathsf{d}\mathcal{L}^+} F \to [\alpha]G$.

Dualno svojstvima sigurnosti, slična tvrdnja vrijedi i za svojstva dostiživosti. Ova su svojstva oblika $F \to \langle\alpha\rangle G$. Motivacija za njihov naziv je to što opisuju da će neka tvrdnja kad tad vrijediti. (Prigodan primjer bio bi: *vlak će u nekom trenutku stići na cilj.*)

Propozicija 7.2. Za svaki hibridni program $\alpha \in HP(\Sigma, V)$ i sve $F, G \in Fml_{FOD}(\Sigma, V)$ formule logike FOD vrijedi da $\models F \to \langle\alpha\rangle G$ povlači $\vdash_{\mathsf{d}\mathcal{L}^+} F \to \langle\alpha\rangle G$.

Primijetimo da iz Propozicije 7.1 primjenom Leme 7.5 slijedi $F \vdash_{\mathsf{d}\mathcal{L}^+} [\alpha]G$. Također, primjenom iste leme iz Propozicije 7.2 dobivamo $F \vdash_{\mathsf{d}\mathcal{L}^+} \langle\alpha\rangle G$.
Uz pomoć navedenih propozicija, možemo dokazati glavni rezultat - teorem o relativnoj potpunosti logike $\mathsf{d}\mathcal{L}$. U dokazu teorema koristit ćemo pojam konjunktivne normalne forme za formulu logike $\mathsf{d}\mathcal{L}$ pa ga prethodno definiramo.

Definicija 7.6. Kažemo da je formula $\phi \in Fml(\Sigma, V)$ u **konjunktivnoj normalnoj formi** ako je oblika: $A_1(A_{1,1}\phi_{1,1} \vee A_{1,2}\phi_{1,2} \vee \ldots \vee A_{1,n_1}\phi_{1,n_1}) \wedge \ldots \wedge A_m(A_{m,1}\phi_{m,1} \vee A_{m,2}\phi_{m,2} \vee \ldots \vee A_{m,n_m}\phi_{m,n_m})$. Pritom su $\phi_{i,j}$ formule iz Fml_{FO} bez kvantifikatora ili negirane takve formule ispred kojih se može i ne mora pojaviti $A_{i,j}$ što stoji za $[\alpha]$ ili $\langle\alpha\rangle$, gdje je α neki hibridni program. Simbol A_i koji se pojavljuje ispred disjunkcija također stoji za $[\alpha]$ ili $\langle\alpha\rangle$ ili prazninu (ne mora se pojaviti).

Naglasimo još jednom - kvantifikatori su samo pokrata za hibridne programe pa se ni u konjunktivnoj normalnoj formi kvantifikatori ne pojavljuju. Nakon svih dosad iskazanih tvrdnji, možemo dokazati tvrdnju o relativnoj potpunosti računa logike $\mathsf{d}\mathcal{L}$.

Teorem 7.5. Račun logike $\mathsf{d}\mathcal{L}$ potpun je obzirom na logiku FOD, tj. za svaku valjanu formulu ϕ logike $\mathsf{d}\mathcal{L}$ vrijedi

$$\vdash_{\mathsf{d}\mathcal{L}^+} \phi$$

Dokaz. Bez smanjenja općenitosti možemo pretpostaviti da je formula ϕ dana u konjunktivnoj normalnoj formi. Dokaz provodimo indukcijom po broju modalnosti koje se pojavljuju u formuli ϕ (taj broj označavamo s $|\phi|$). Baza indukcije je slučaj $|\phi| = 0$. To znači da je ϕ formula prvog reda i stoga vrijedi $\vdash_{\mathsf{d}\mathcal{L}^+} \phi$. Ako je ϕ oblika $\neg\phi_1$, a znamo da je ϕ dana u normalnoj formi pa sve negacije moraju biti unutar modalnosti, onda ponovno vrijedi $|\phi| = 0$ i tvrdnja slijedi.
Pretpostavimo sada da vrijedi $\vdash_{\mathsf{d}\mathcal{L}^+} \phi$ za sve ϕ koji sadrže manje od n modalnosti i promotrimo formulu ϕ koja sadrži n modalnosti.
Neka je ϕ oblika $\phi_1 \wedge \phi_2$. Ako je $|\phi_1| > 0$ i $|\phi_2| > 0$, onda znamo da vrijedi $|\phi_1|, |\phi_2| < |\phi|$. Stoga, prema pretpostavci indukcije vrijedi $\vdash_{\mathsf{d}\mathcal{L}^+} \phi_1$ i $\vdash_{\mathsf{d}\mathcal{L}^+} \phi_2$. Ta dva izvoda kombiniramo pomoću pravila $(\wedge r)$ i dobivamo $\vdash_{\mathsf{d}\mathcal{L}^+} \phi_1 \wedge \phi_2$. Ako jedan od dva konjunkta ne sadrži modalnosti (bez smanjenja općenitosti, neka je to ϕ_1), tada $\vdash_{\mathsf{d}\mathcal{L}^+} \phi_1$ vrijedi prema bazi indukcije, a ϕ_2 je ponovno formula u normalnoj formi s n modalnosti pa promatramo njen oblik. Jednom kad pronađemo [5] dokaz u računu $\mathsf{d}\mathcal{L}^+$ za formulu ϕ_2, dokaz za ϕ ponovno se dobiva kombiniranjem ϕ_1 i ϕ_2 pomoću pravila $(\wedge r)$.
U slučaju pak da se ϕ sastoji samo od disjunkcija, bez smanjenja općenitosti možemo pretpostaviti da je oblika $\phi_1 \vee [\alpha]\phi_2$ ili $\phi_1 \vee \langle\alpha\rangle\phi_2$ (u suprotnom možemo doći do tog oblika koristeći komutativnost i asocijativnost disjunkcije). Budući da se dokaz u oba slučaja odvija potpuno jednako, nastavljamo samo za slučaj $\phi_1 \vee \langle\alpha\rangle\phi_2$. Očito je $|\phi_2| < |\phi|$ i $|\phi_1| < |\phi|$. Prema Lemi 7.3 postoje ϕ_1^{FOD} i ϕ_2^{FOD}, formule logike FOD, za koje vrijedi $\models \phi_1 \leftrightarrow \phi_1^{FOD}$ i $\models \phi_2 \leftrightarrow \phi_2^{FOD}$. Iz toga slijedi da $\models \phi$ povlači $\models \phi_1^{FOD} \vee \langle\alpha\rangle\phi_2^{FOD}$ što je ekvivalentno s $\models \neg\phi_1^{FOD} \rightarrow \langle\alpha\rangle\phi_2^{FOD}$. Sada prema Propoziciji 7.2 (kada bismo dokazivali drugi slučaj, koristili bismo Propoziciju 7.1) slijedi

$$\neg\phi_1^{FOD} \vdash_{\mathsf{d}\mathcal{L}^+} \langle\alpha\rangle\phi_2^{FOD} \tag{2}$$

Nadalje, $\phi_1 \leftrightarrow \phi_1^{FOD}$ povlači $\neg\phi_1 \rightarrow \neg\phi_1^{FOD}$. Budući da, prema pretpostavci indukcije, možemo dokazati ϕ_1 jer vrijedi $|\phi_1| < |\phi|$, možemo dokazati i $\phi_1 \vee \neg\phi_1^{FOD}$ tj. $\neg\phi_1 \rightarrow \neg\phi_1^{FOD}$. Iz Leme 7.5 dobivamo $\neg\phi_1 \vdash_{\mathsf{d}\mathcal{L}^+} \neg\phi_1^{FOD}$. Sada to kombiniramo s (2) koristeći pravilo izvoda (*cut*) i dobivamo

$$\neg\phi_1 \vdash_{\mathsf{d}\mathcal{L}^+} \langle\alpha\rangle\phi_2^{FOD} \tag{3}$$

Slično, $\models \phi_2 \leftrightarrow \phi_2^{FOD}$ povlači $\models \phi_2^{FOD} \rightarrow \phi_2$. Prema pretpostavci indukcije za tu formulu postoji izvod jer vrijedi $|\phi_2| < |\phi|$ pa kao u prethodnom slučaju dokazujemo ϕ_2, a zatim $\neg\phi_2^{FOD} \vee \phi_2$. Sada prema Lemi 7.4 znamo da vrijedi $\vdash_{\mathsf{d}\mathcal{L}^+} \forall^\alpha(\phi_2^{FOD} \rightarrow \phi_2)$ a dalje primjenom pravila $(\langle\rangle gen)$ (u drugom slučaju bismo primijenili pravilo $[]gen$) dobivamo $\langle\alpha\rangle\phi_2^{FOD} \vdash_{\mathsf{d}\mathcal{L}^+} \langle\alpha\rangle\phi_2$. Taj izvod kombiniramo s izvodom sekventa u tvrdnji (3) pomoću pravila (*cut*) čime dolazimo do $\neg\phi_1 \vdash_{\mathsf{d}\mathcal{L}^+} \langle\alpha\rangle\phi_2$ odakle se dobiva $\vdash_{\mathsf{d}\mathcal{L}^+} \phi_1 \vee \langle\alpha\rangle\phi_2$, čime je dokaz završen. □

[5] to će se dogoditi prije ili kasnije jer je broj modalnosti u formuli ϕ veći od nule

Literatura

[1] A. Platzer, *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*, Springer, Berlin, 2010.

[2] D. Harel, *First-Order Dynamic Logic*, Springer, New York, 1979.

[3] D. Harel, D. Kozen, J. Tiuryn, *Dynamic Logic*, MIT Press, Cambridge, 2000.

[4] Encyclopedia of Math, natuknica *Skolem functions*, dostupno na `http://www.encyclopediaofmath.org/index.php/Skolem_function`, pristupljeno u srpnju 2013.

[5] I. Chiswell, W.Hodges, *Mathematical Logic*, Oxford University Press, New York, 2007.

[6] KeYmaera, sustav za automatsku analizu hibridnih sustava, verzija 3.3, dostupno na `http://symbolaris.com/info/KeYmaera.html`, pristupljeno u rujnu 2013.

[7] M. Vuković, *Matematička logika*, Element, Zagreb, 2009.

Sažetak

Hibridni sustavi model su za složene fizikalne sustave koji uključuju interakciju diskretnih i kontinuiranih promjena stanja sustava. (Diskretne promjene u pravilu uzrokuje upravljačka jedinica promjenom određenih veličina, a kontinuirane su rezultat fizikalnog procesa, najčešće gibanja.) Radi velikih mogućnosti koje pružaju, takvi se sustavi često koriste u industriji. Cilj ovog diplomskog rada bio je predstaviti jedan način za logičku analizu takvih sustava i automatsku verifikaciju njihovih svojstava. Središnji dio rada je predstavljanje diferencijalne dinamičke logike, $\mathsf{d}\mathcal{L}$. Logika $\mathsf{d}\mathcal{L}$ kao svoj sastavni dio sadrži *hibridne programe* - izraze koji uključuju i diskretna pridruživanja i diferencijalne jednadžbe te kao takvi izvrsno opisuju hibridne sustave. Usporedno s uvođenjem pojmova i tehnika logike $\mathsf{d}\mathcal{L}$, opisano je nekoliko primjera jednostavnih hibridnih sustava koji služe kao ilustracija za sve uvedene teorijske koncepte. Kao glavni primjer stvarne primjene logike $\mathsf{d}\mathcal{L}$ uzet je sustav europske mreže željeznica, ETCS.
Nakon definiranja sintakse i semantike logike $\mathsf{d}\mathcal{L}$, zadaje se i njen račun. Uz pomoć tog računa se dokazuju ili provjeravaju svojstva sustava. Račun je zadan kompozicijski - dokaze svojstava sustava svodi na dokaze svojstava dijelova sustava. Na kraju se, kako bi uvođenje računa bilo opravdano, dokazuje teorem adekvatnosti računa logike $\mathsf{d}\mathcal{L}$. Pokazuje se i da račun nije potpun, ali da je relativno potpun obzirom na svojstva rješenja diferencijalnih jednadžbi.

Summary

Hybrid systems are models for complex physical systems that combine discrete (e. g. digital) and continuous (e. g. analog or physical) effects. In case of hybrid systems, it is the interaction between discrete component (usually, control unit) and continuous component (some physical process, usally movement) that matters. Due to very powerful system design achievable using hybrid systems, their importance in industry grows rapidly. In spite of that fact, hybrid systems is an area where analytic approaches are still not developed well enough. The main goal of this thesis is to present one approach to logical analysis of hybrid systems' properties. The central part is introducing a differential dynamc logic, $\mathsf{d}\mathcal{L}$. Formulas of $\mathsf{d}\mathcal{L}$ internalise models for hybrid systems - hybrid programs - as first-class citizens. Hybrid programs contain both discrete assignments and continuous evolution along differential equations. This allows us to express statements about hybrid systems as $\mathsf{d}\mathcal{L}$ formulas. Together with introducing notions and techniques of $\mathsf{d}\mathcal{L}$ logic, a number of simplistic examples of hybrid systems is described and used to illustrate all those theoretical concepts. As the central real-life example for verification with tecniques of $\mathsf{d}\mathcal{L}$, we use European Train Control System.
After defining syntax and semantics of $\mathsf{d}\mathcal{L}$, a compositional sequent calculus is presented. Due to its compositional structure, this calculus is suitable for automation. In order to justify the calculus, we present proof of its soundness and completeness relative to handling differential equations.

Životopis

Ivan Gavran rođen je 23. svibnja 1989. godine u Zagrebu. U istom gradu polazi Osnovnu školu Alojzija Stepinca a zatim i Petu gimnaziju. Istovremeno pohađa Osnovnu glazbenu školu Rudolfa Matza te maturira na odjelu za gitaru Glazbene škole Pavla Markovca. Na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu 2008. godine upisuje preddiplomski studij matematike. I sam bivši natjecatelj, u vrijeme studija sudjeluje u radu studentske udruge MNM, posvećene radu sa srednjoškolcima, sudionicima matematičkih natjecanja; drži predavanja u Petoj gimnaziji i na Kampu mladih matematičara u Pazinu te organizira međunarodno matematičko natjecanje za srednjoškolce, *Turnir gradova*. Demonstrator je za više kolegija na studiju. Uz matematičke teme, pokazuje zanimanje i za njima srodne: sudjeluje u ljetnoj školi robotike na Sveučilištu u Bratislavi (STU Bratislava) i ljetnoj školi inžinjerstva u medicini na Katoličkom sveučilištu u Leuvenu (KU Leuven). Na kraju preddiplomskog studija nagrađen je Priznanjem za izniman uspjeh na studiju. Izobrazbu nastavlja na istom fakultetu, na diplomskom studiju *Računarstvo i matematika*. Interes usmjerava na dva područja: računalnu obradu prirodnog jezika i matematičku logiku. O tome svjedoče radovi *Hrvatski kao vfc-jezik*, izrađen zajedno s kolegom Mirom Antonijevićem i web-portal *Makako* za automatsku analizu sentimenta u novinskim člancima izrađen s kolegom Markom Božićem. S istim kolegom, a u suradnji s dr. sc. Marijem Muštrom s Fakulteta elektrotehnike i računarstva, razvija algoritam za automatsku detekciju i klasifikaciju žljezdanog tkiva na mamografskim snimkama. Na ljeto 2012. godine sudjeluje na ESSLLI-ju, europskoj ljetnoj školi logike, teorije jezika i informacija. Za trajanja studija, Ivan Gavran pohađa i nekoliko kolegija na drugim fakultetima Sveučilišta: *Poljski jezik* i *Ruski jezik* na Filozofskom fakultetu i kolegij *Obrada prirodnog jezika* na FER-u. Posljednji semestar studija provodi na razmjeni na Sveučilištu u Beču (Universität Wien). Pred kraj diplomskog studija još jednom je nagrađen Priznanjem za izniman uspjeh na studiju te Priznanjem za iznimne rezultate u izvannastavnim aktivnostima. Za svoj diplomski rad odabire temu iz logike - *Logička analiza hibridnih sustava*.