Ne ne biti ali ne biti?

Andrej Bauer

Inštitut za matematiko, fiziko in mehaniko
Univerza v Ljubljani

Solomonov seminar
Inštitut Jožef Stefan, marec 2002

Kaj je narobe s temi izreki?

Izrek: Tri točke so kolinearne ali pa ne.

Izrek: Vsak nekonstanten polinom ima kompleksno ničlo.

Izrek: Skoraj vse realne funkcije so povsod nezvezne.

Kaj je narobe s temi izreki?

Izrek: Tri točke so kolinearne ali pa ne.

Numerični testi za kolinearnost so nestabilni.

Izrek: Vsak nekonstanten polinom ima kompleksno ničlo.

Izrek: Skoraj vse realne funkcije so povsod nezvezne.

Kaj je narobe s temi izreki?

Izrek: Tri točke so kolinearne ali pa ne.

Numerični testi za kolinearnost so nestabilni.

Izrek: Vsak nekonstanten polinom ima kompleksno ničlo.

Klasični dokazi ne povedo, kako se ničlo izračuna.

Izrek: Skoraj vse realne funkcije so povsod nezvezne.

Kaj je narobe s temi izreki?

Izrek: Tri točke so kolinearne ali pa ne.

Numerični testi za kolinearnost so nestabilni.

Izrek: Vsak nekonstanten polinom ima kompleksno ničlo.

Klasični dokazi ne povedo, kako se ničlo izračuna.

Izrek: Skoraj vse realne funkcije so povsod nezvezne.

Takih funkcij v računalništvu ne srečamo.

Računalniško usmerjene veje matematike

- Teorija izračunljivih funkcij
- Računska kompleksnost & algoritmi
- Numerična analiza
- Teorija domen
- Kriptografija
- Teorija vrst
- Teorija končnih modelov
- Strojno učenje
- Teorija tipov
- ...

Ste se kdaj vprašali, …

…ali lahko celotno matematiko izdelamo po meri računalništva?

Ste se kdaj vprašali, ...

...ali lahko celotno matematiko izdelamo po meri računalništva?

To lahko napravimo s *teorijo realizabilnosti.*

Kazalo

1. Kako zgradimo svet realizabilnosti

2. Življenje v svetu realizabilnosti

3. Praktična vprašanja

Kako sploh zgradimo matematični svet?

Uporabimo teorijo kategorij.

Kako sploh zgradimo matematični svet?

Uporabimo teorijo kategorij.

Dva koraka:

1. Poiščemo kategorijo, ki opisuje naš računalniški pogled na svet.

2. Kategorijo proučujemo s prijemi kategorne logike.

Kako sploh zgradimo matematični svet?

Uporabimo teorijo kategorij.

Dva koraka:

1. Poiščemo kategorijo, ki opisuje naš računalniški pogled na svet.

   kategorija = objekti + morfizmi

2. Kategorijo proučujemo s prijemi kategorne logike.

   Strukturne lastnosti kategorije določajo logične zakone.

Računalniški pogled na svet

- Vse je narejeno iz Turingovih strojev.

Računalniški pogled na svet

- Vse je narejeno iz Turingovih strojev.
- Vse je narejeno iz ML programov.

Računalniški pogled na svet

- Vse je narejeno iz Turingovih strojev.
- Vse je narejeno iz ML programov.
- Vse je narejeno iz Scottovih domen.

Računalniški pogled na svet

- Vse je narejeno iz Turingovih strojev.
- Vse je narejeno iz ML programov.
- Vse je narejeno iz Scottovih domen.
- *Relativni* pogled:
  1. Katere podatke je mogoče *predstaviti*?
  2. Kako *računamo* s podatki?

Relativna izračunljivost

- Podatki: vsebina neskončnega RAM

- Računanje: (končni) program

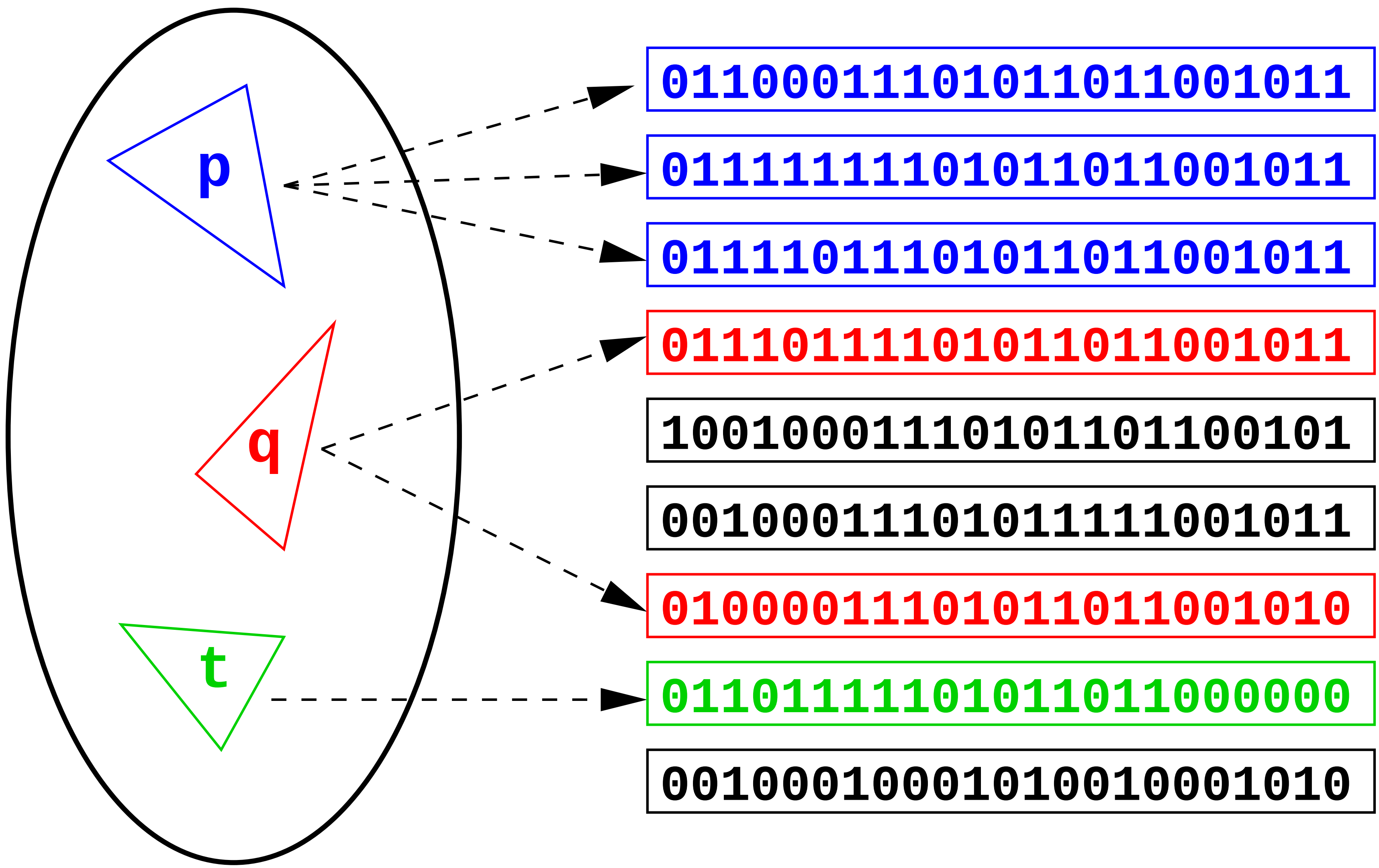
Relativna izračunljivost

- Podatki: vsebina neskončnega RAM
  - dopuščamo *vse* vsebine, tudi neizračunljive
  - lahko bi dodali druge vire podatkov
- Računanje: (končni) program

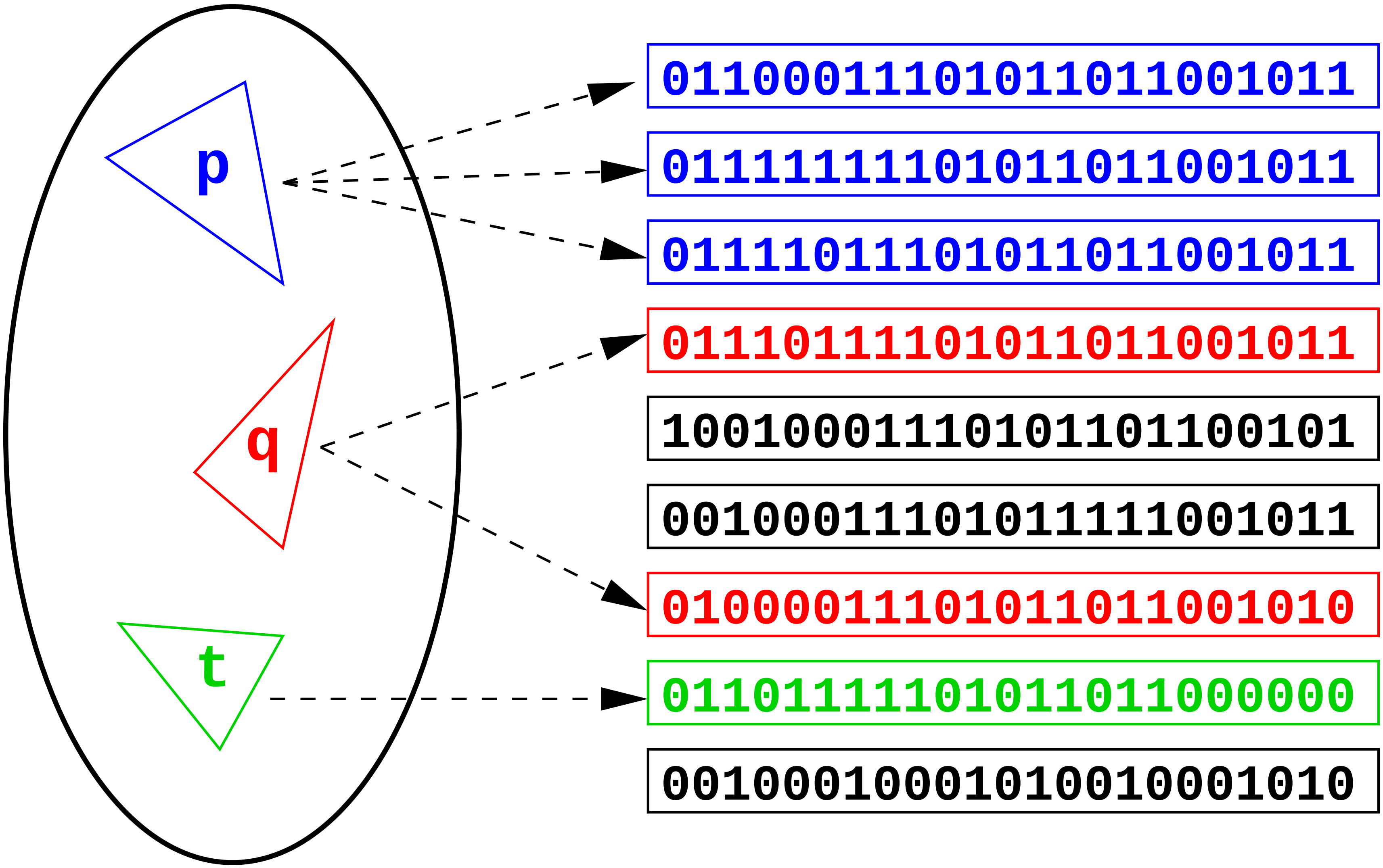
Relativna izračunljivost

- Podatki: vsebina neskončnega RAM
  - dopuščamo *vse* vsebine, tudi neizračunljive
  - lahko bi dodali druge vire podatkov
- Računanje: (končni) program
  - programi napisani v splošnem programskem jeziku
  - različni jeziki dajo različne lastnosti sveta realizabilnosti

Skromne Množice

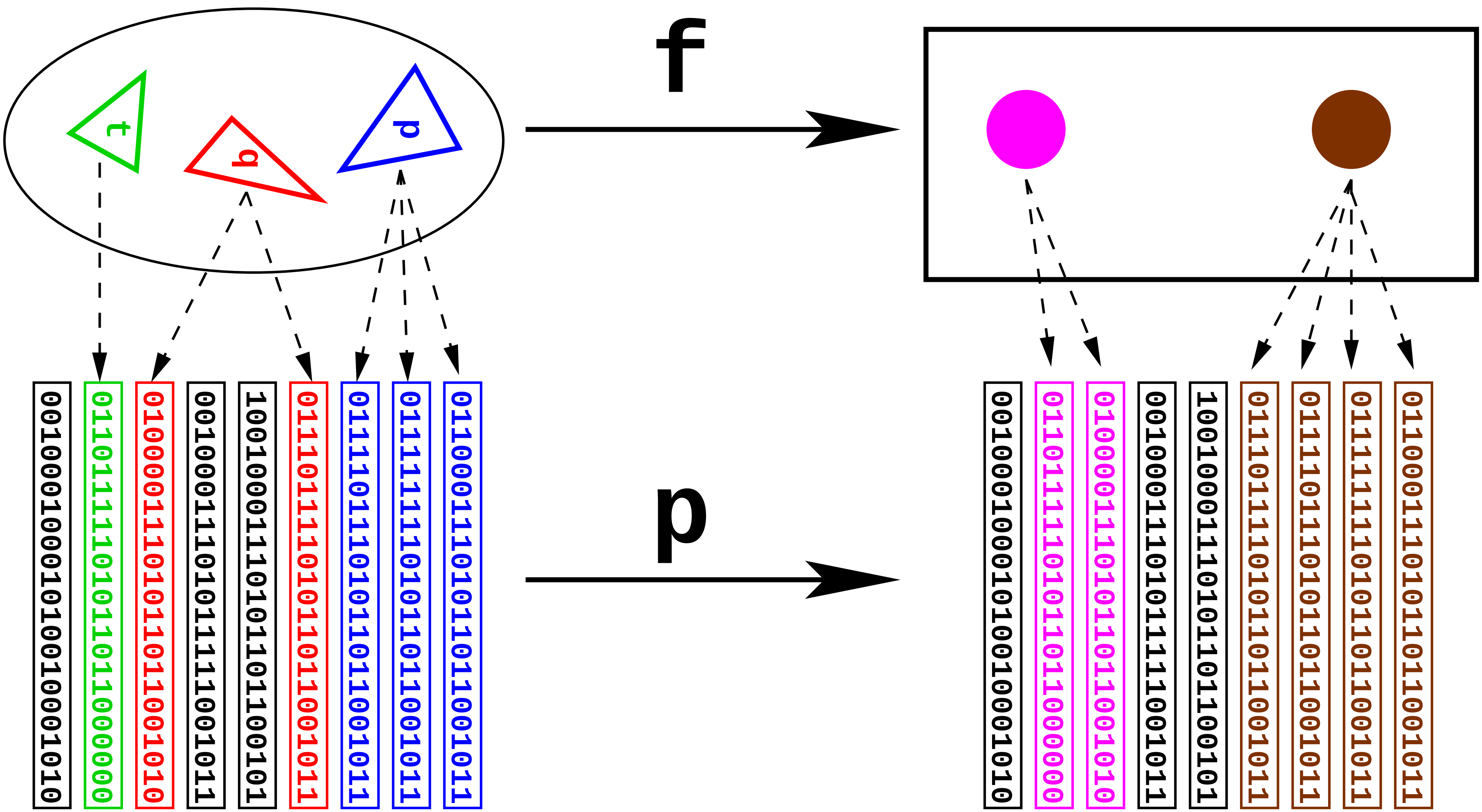


Skromne Množice

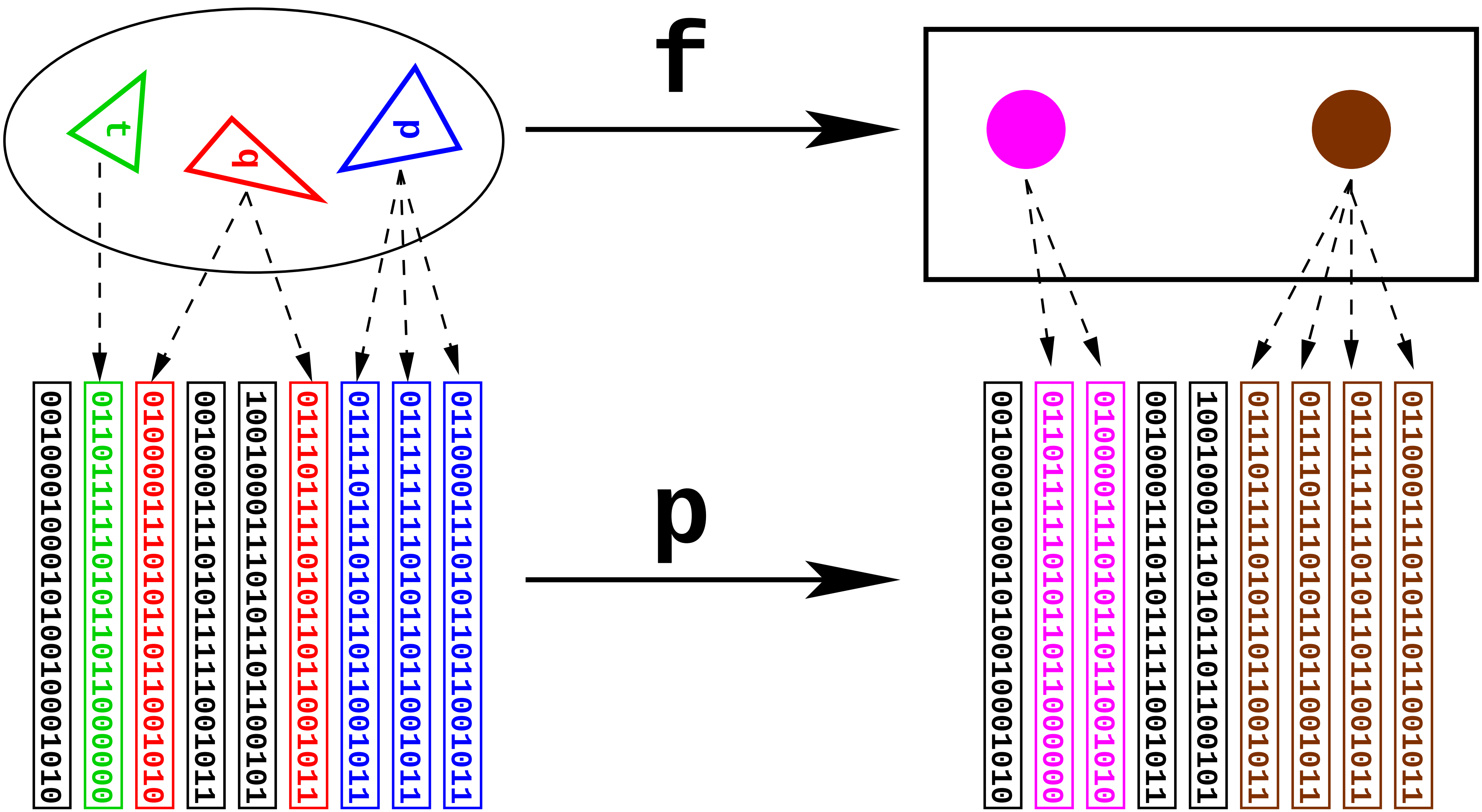


$c \Vdash x$ "c realizira (predstavlja) x"

Realizirane Funkcije



Realizirane Funkcije



Če $\quad c \Vdash x, \quad$ potem $\quad p(c) \Vdash f(x).$

Računalniško osveščene množice in funkcije

Osnova klasične matematike:

množice & funkcije

Računalniško osveščene množice in funkcije

Osnova klasične matematike:

množice & funkcije

Osnove *računalniško osveščene* matematike:

skromne množice & realizirane funkcije

Teorija kategorij na delu

Vsaka matematična struktura, ki je določena z univerzalno lastnostjo, ima do izomorfizma določeno reprezentacijo.

Teorija kategorij na delu

Vsaka matematična struktura, ki je določena z univerzalno lastnostjo, ima do izomorfizma določeno reprezentacijo.

1. Naravna števila

   začetna algebra z eno konstanto in eno unarno operacijo

Teorija kategorij na delu

Vsaka matematična struktura, ki je določena z univerzalno lastnostjo, ima do izomorfizma določeno reprezentacijo.

1. Naravna števila

   začetna algebra z eno konstanto in eno unarno operacijo

2. Realna števila

   polni Arhimedov obseg

Teorija kategorij na delu

Vsaka matematična struktura, ki je določena z univerzalno lastnostjo, ima do izomorfizma določeno reprezentacijo.

1. Naravna števila

   začetna algebra z eno konstanto in eno unarno operacijo

2. Realna števila

   polni Arhimedov obseg

3. Induktivne množice (seznami, drevesa, ...)

Teorija kategorij na delu

Vsaka matematična struktura, ki je določena z univerzalno lastnostjo, ima do izomorfizma določeno reprezentacijo.

1. Naravna števila

   začetna algebra z eno konstanto in eno unarno operacijo

2. Realna števila

   polni Arhimedov obseg

3. Induktivne množice (seznami, drevesa, ...)

4. Kartezični produkt $A \times B$

5. Množica funkcij $A \to B$

Primer: realna števila $\mathbb{R}$

Par $\langle d, e\rangle$ realizira realno število $x \in \mathbb{R}$:

$$\langle d, e\rangle \Vdash_{\mathbb{R}} x \ ,$$

Primer: realna števila $\mathbb{R}$

Par $\langle d, e\rangle$ realizira realno število $x \in \mathbb{R}$:

$$\langle d, e\rangle \Vdash_{\mathbb{R}} x \ ,$$

where:

- mantisa $d = d_0 d_1 d_2 \ldots$, kjer je $d_i \in \{-1, 0, 1\}$

Primer: realna števila $\mathbb{R}$

Par $\langle d, e \rangle$ realizira realno število $x \in \mathbb{R}$:

$$\langle d, e \rangle \Vdash_{\mathbb{R}} x \ ,$$

where:

- mantisa $d = d_0 d_1 d_2 \ldots$, kjer je $d_i \in \{-1, 0, 1\}$
- eksponent $e \in \mathbb{Z}$

Primer: realna števila $\mathbb{R}$

Par $\langle d, e\rangle$ realizira realno število $x \in \mathbb{R}$:

$$\langle d, e\rangle \Vdash_{\mathbb{R}} x \ ,$$

where:

- mantisa $d = d_0 d_1 d_2 \ldots$, kjer je $d_i \in \{-1, 0, 1\}$
- eksponent $e \in \mathbb{Z}$
- *predznačeni* dvojiški zapis:

$$x = 2^e \cdot \sum_{k=0}^{\infty} \frac{d_i}{2^{k+1}}$$

Kazalo

✓ Kako zgradimo svet realizabilnosti

☞ Življenje v svetu realizabilnosti

3. Praktična vprašanja

Teorija delcev v svetu realizabilnosti

“Osnovni gradniki so očesu nevidni realizatorji.”

Teorija delcev v svetu realizabilnosti

"Osnovni gradniki so očesu nevidni realizatorji."

"Osnovna realizatorja sta S in K."

$$\mathrm{K}xy = x \qquad Sxyz = (xz)(yz)$$

Jezik realizabilnosti

Računalniško razumevanje resnice:

> "Izjava je resnična, če jo realizira program."

Jezik realizabilnosti

Računalniško razumevanje resnice:

"Izjava je resnična, če jo realizira program."

Primer: izjavo

$$\forall x \in \mathbb{R}.(x < 0 \lor x \geq 0)$$

Jezik realizabilnosti

Računalniško razumevanje resnice:

"Izjava je resnična, če jo realizira program."

Primer: izjavo

$$\forall x \in \mathbb{R}.(x < 0 \lor x \geq 0)$$

realizira tak program p, da je za $\langle d, e \rangle \Vdash_{\mathbb{R}} x$,

$$\begin{aligned} p(d, e) &= 0 \qquad \text{if } x < 0 \\ p(d, e) &= 1 \qquad \text{if } x \geq 0 \end{aligned}$$

Intuicionistična logika

Logika realizabilnosti je *intuicionistična*.

Intuicionistična logika

Logika realizabilnosti je *intuicionistična*.

Zakon o izključeni tretji možnosti ni veljaven:

$$\varphi \vee \neg\varphi$$

Intuicionistična logika

Logika realizabilnosti je *intuicionistična*.

Zakon o izključeni tretji možnosti ni veljaven:

$$\varphi \vee \neg\varphi$$

Dokazovanje s protislovjem je v splošnem neveljavno:

$$\neg\neg\varphi \Longrightarrow \varphi$$

Intuicionistična logika

Logika realizabilnosti je *intuicionistična*.

Zakon o izključeni tretji možnosti ni veljaven:

$$\varphi \vee \neg\varphi$$

Dokazovanje s protislovjem je v splošnem neveljavno:

$$\neg\neg\varphi \Longrightarrow \varphi$$

Ti zakoni računalničarjem itak povzročajo preglavice!

Princip Markova

Če lahko elemente množice A naštejemo in je $\varphi(x)$ *odločljiv* predikat, potem

$$(\neg\forall x \in A.\, \neg\varphi(x)) \quad \Longrightarrow \quad \exists x \in A.\, \varphi(x)$$

Princip Markova

Če lahko elemente množice A naštejemo in je $\varphi(x)$ *odločljiv* predikat, potem

$$(\neg\forall x \in A.\, \neg\varphi(x)) \implies \exists x \in A.\, \varphi(x)$$

Realiziramo s programom, ki po vrsti za vse elemente $x_1, x_2, \ldots$ of A preveri, ali $\varphi(x_i)$ velja, dokler enega ne najde.

Vse funkcije so zvezne

"Vse funkcije $f : \mathbb{R} \to \mathbb{R}$ so zvezne."

Vse funkcije so zvezne

"Vse funkcije $f : \mathbb{R} \to \mathbb{R}$ so zvezne."

"Prvih n števk rezultata $f(x)$ je odvisnih le od končno mnogo števk argumenta x."

Vse funkcije so zvezne

"Vse funkcije $f : \mathbb{R} \to \mathbb{R}$ so zvezne."

"Prvih n števk rezultata $f(x)$ je odvisnih le od končno mnogo števk argumenta x."

Realiziramo s programom, ki za dane $p \Vdash f$, $\langle d, e \rangle \Vdash x$ in $n \in \mathbb{N}$ poišče tak $k \in \mathbb{N}$, da p prebere kvečjemu k števk $\langle d, e \rangle$ pri računanju prvih n števk $f(x)$.

“Ne ne” prevod

Klasično logiko lahko prevedemo v intuicionistično logiko.

“Ne ne” prevod

Klasično logiko lahko prevedemo v intuicionistično logiko.

Klasično	Intuicionistično
$\varphi \Longrightarrow \psi$	$\neg\neg(\varphi^* \Longrightarrow \psi^*)$
$\varphi \land \psi$	$\neg\neg(\varphi^* \land \psi^*)$
$\varphi \lor \psi$	$\neg\neg(\varphi^* \lor \psi^*)$
$\exists x.\, \varphi(x)$	$\neg\neg\exists x.\, \varphi(x)^*$
$\forall x.\, \varphi(x)$	$\neg\neg\forall x.\, \varphi(x)^*$

Ne ne klasična umetnost

Not not, not to not be, or not to be:
not that is not the question:
Not not, whether not 'tis not nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or not to not take arms against a sea of troubles,
And not by not opposing not not end them?

(not not Hamlet)

Česa prevod ne spremeni?

“Ne ne” stabilne izjave se ne spremenijo.

Česa prevod ne spremeni?

"Ne ne" stabilne izjave se ne spremenijo.

Izjave sestavljene iz $=$, $<$, $\land$, $\Longrightarrow$ and $\forall$.

Česa prevod ne spremeni?

“Ne ne” stabilne izjave se ne spremenijo.

Izjave sestavljene iz $=$, $<$, $\wedge$, $\implies$ and $\forall$.

Primeri “ne ne” stabilnih izjav:

- hitrost izvajanja algoritmov
- hitrost konvergence numeričnih metod
- kombinatorne identitete
- končna kombinatorika

Kazalo

✓ Kako zgradimo svet realizabilnosti

✓ Življenje v svetu realizabilnosti

☞ Praktična vprašanja

Matematika po meri računalništva

Iz dokaza izjave dobimo program, ki izjavo realizira.

Matematika po meri računalništva

Iz dokaza izjave dobimo program, ki izjavo realizira.

Iz konstrukcije množice ali funkcije dobimo podatkovno strukturo ali program, ki jo implementira.

Matematika po meri računalništva

Iz dokaza izjave dobimo program, ki izjavo realizira.

Iz konstrukcije množice ali funkcije dobimo podatkovno strukturo ali program, ki jo implementira.

Pravilnost implementacije preverimo tako, da dokažemo, da realizira želeno specifikacijo.

Vprašanje za strokovnjake za hardware

Realna števila so implementirana z *predznačenim* dvojiškim zapisom.

Vprašanje za strokovnjake za hardware

Realna števila so implementirana z *predznačenim* dvojiškim zapisom.

Ali bi lahko uspešno uporabili negativne števke v aritmetičnih vezjih?

Izziv za računske geometre

Testiranje kolinearnosti ne samo numerično nestabilno, ampak je *nekonstruktivno*.

Izziv za računske geometre

Testiranje kolinearnosti ne samo numerično nestabilno, ampak je *nekonstruktivno*.

Dokazuj izreke brez uporabe

$$\forall x \in \mathbb{R}.\, (x < 0 \lor x \geq 0)$$

Namesto tega uporabi

$$\forall \epsilon > 0.\, \forall x \in \mathbb{R}.\, (x < \epsilon \lor x > -\epsilon)$$

Naloga za načrtovalce programskih jezikov

Navadni `if-then-else` kontrolni mehanizem
ni primeren za realno aritmetiko.

Naloga za načrtovalce programskih jezikov

Navadni `if-then-else` kontrolni mehanizem
ni primeren za realno aritmetiko.

Iznajdi praktične podatkovne strukture za realna števila
in nove kontrolne mehanizme za bolj uspešno
programiranje z realnimi števili.

Nauk zgodbe

Svet realizabilnosti je *vaš* svet.