Matematika u distribuiranom računarstvu: čemu služi i kako se koristi

Ivan Prokić

Fakultet tehničkih nauka, Novi Sad

Seminar LADIS 3.0, Prirodno-matematički fakultet, Novi Sad
16. maj 2023.

Svet distibuiranog računarstva: benefiti

- Povezanost

Svet distibuiranog računarstva: problemi

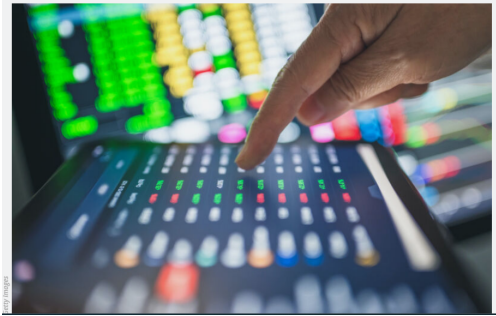
ars TECHNICA — BIZ & IT · TECH · SCIENCE · POLICY · CARS · GAMING & CULTURE · STORE

DIGITAL HEIST —

Really stupid "smart contract" bug let hackers steal $31 million in digital coin

Company says it has contacted the hacker in an attempt to recover the funds. Good luck.

DAN GOODIN - 12/2/2021, 12:41 AM



Getty Images

Meta — Shop · Our technologies · About us · Build with us

Originally published on September 28, 2018 at 9:41AM PT

Security Update

By Guy Rosen, VP of Product Management

On the afternoon of Tuesday, September 25, our engineering team discovered a security issue affecting almost 50 million accounts. We're taking this incredibly seriously and wanted to let everyone know what's happened and the immediate action we've taken to protect people's security.

Our investigation is still in its early stages. But it's clear that attackers exploited a vulnerability in Facebook's code that impacted "View As" a feature that lets people see what their own profile looks like to someone else. This allowed them to steal Facebook access tokens which they could then use to take over people's accounts. Access tokens are the equivalent of digital keys that keep people logged in to Facebook so they don't need to re-enter their password every time they use the app.

- softverske greške

Svet (distibuiranog) računarstva: problemi

- softverske greške

If there is a bug in a distributed algorithm,
no matter how improbable it may seem,
it's not a question of whether it will appear,
it's a question of when it will appear.

(Leslie Lamport, Heidelberg Laureate Forum 2021,
https://www.youtube.com/watch?v=KVs3YFKqclU)

New York is a city in which
one in a million thing
happen to eight people a day.

(Whitfield Diffie, Heidelberg Laureate Forum 2021)

Odgovori na probleme i njihovi problemi

- **testiranje**:

  Program testing can be used to show the presence of bugs, but never to show their absence!

  (Edsger W. Dijkstra, Turing Award in 1972: "The humble programmer")

- **veštačka inteligencija**:

  Yet, because AI does not truly grasp the meaning of things—it can generate natural language text and even computer code that appears to make sense, but contains logic, syntax, and other errors—the lack of close human review ratchets up the risks.

  (Samuel Greengard, "AI Rewrites Coding", Communications of the ACM, March 2023)

- **matematika**:

  ...the expressiveness of the languages involved, as well as the complexity of the systems being modeled, make full formalization a difficult and expensive task.

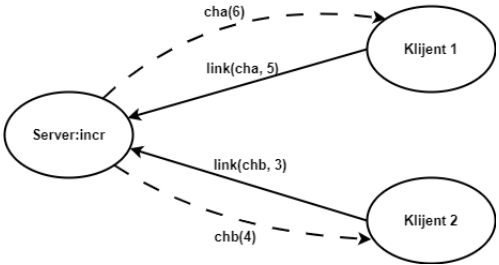
  (Wikipedia, Formal Methods)

Formalne metode

Matematika: formalne metode

- Fomalni modeli služe za **specifikaciju** i **verifikaciju** konkurentnih i distribuiranih sistema
- Postoji veliki broj ovakvih modela:
  - Petrijeve mreže, Carl Adam Petri
  - Model Aktora, Carl Hewitt
  - Temporalna logika akcija (TLA+), Leslie Lamport
  - Komunicirajući sekvencijalni procesi (CSP), Tony Hoare
  - π-**račun**, Robin Milner, Joachim Parrow i David Walker
  - **Tipovi sesija**, Kohei Honda
  - ...

Procesna algebra π-račun (sintaksa)



- Konkurentni procesi:

$$\texttt{Server:incr} \mid \texttt{Klijent1} \mid \texttt{Klijent2}$$

- Slobodna imena (linkova):

$$\texttt{Server:incr} = !\texttt{link}?(x, y).\texttt{x}!(y + 1).0$$

- Privatna (vezana) imena linkova:

$$\texttt{Klijent1} = (\nu\, \texttt{cha})\texttt{link}!(\texttt{cha}, 5).\texttt{cha}?(x).0$$

$$\texttt{Klijent2} = (\nu\, \texttt{chb})\texttt{link}!(\texttt{chb}, 5).\texttt{chb}?(x).0$$

- Paralelna kompozicija ($|$), slanje/primanje ($!/?$), replikacija ($!P$), sekvencijalna kompozicija (tačka), izbor (nema u ovom primeru)

Procesna algebra π-račun (semantika)

- Sinhronizacija slanja i primanja

$$\texttt{Server:incr} \mid \texttt{Klijent1} \mid \texttt{Klijent2}$$

Procesna algebra π-račun (semantika)

- Sinhronizacija slanja i primanja

$$\texttt{Server:incr} \mid \texttt{Klijent1} \mid \texttt{Klijent2}$$

$$\downarrow$$

$$\texttt{Server:incr} \mid (\nu\, \texttt{cha})\big(\texttt{cha}!(6).0 \mid \texttt{cha}?(x).0\big) \mid \texttt{Klijent2}$$

Procesna algebra π-račun (semantika)

- Sinhronizacija slanja i primanja

$$\texttt{Server:incr} \mid \texttt{Klijent1} \mid \texttt{Klijent2}$$

$$\downarrow$$

$$\texttt{Server:incr} \mid (\nu\, \texttt{cha})\big(\texttt{cha}!(6).0 \mid \texttt{cha}?(x).0\big) \mid \texttt{Klijent2}$$

$$\downarrow$$

$$\texttt{Server:incr} \mid (\nu\, \texttt{cha})\big(0 \mid 0\big) \mid \texttt{Klijent2}$$

- Ovo dozvoljava da definišemo bihevioralne ekvivalencije (bisimulacija i druge)

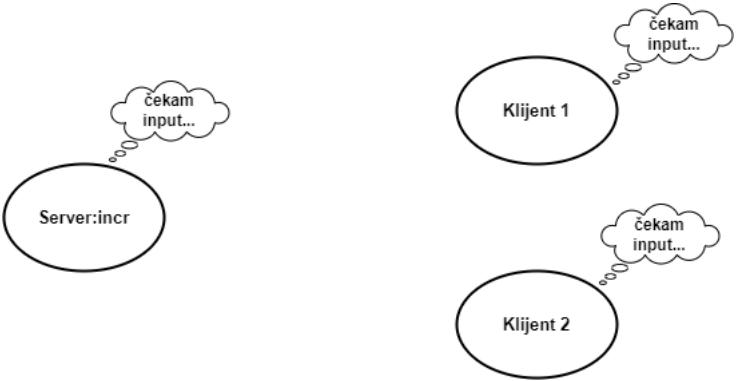
(The pi-calculus: a Theory of Mobile Processes, D Sangiorgi, D Walker, 2003, Cambridge university press)

Procesne algebre: proširenja, fragmenti i varijante π-računa

- π-račun je Tjuring-kompletan (Milner, Robin (1992). "Functions as Processes", MSCS)
- varijante:
  - (The applied pi calculus: Mobile values, new names, and secure communication, M Abadi, B Blanchet, C Fournet - Journal of the ACM (JACM), 2017)
  - (A Calculus for Cryptographic Protocols: The Spi Calculus, M Abadi, AD Gordon - Information and Computation, 1999)
  - (A distributed Pi-calculus, M Hennessy, 2007, Cambridge university press)
  - (An object calculus for asynchronous communication, K. Honda, M. Tokoro, Proc. ECOOP 91, Geneve, 1991.)
  - ...
  - (The C_π-calculus: A model for confidential name passing, I Prokić, HT Vieira, JLAMP 2021)
  - (A calculus for modeling floating authorizations, I Prokić, J Pantović, HT Vieira, JLAMP 2019)
- Zašto ovakvo mnoštvo procesnih algebri? (Expressiveness of process algebras, J Parrow - ENTCS, 2008)

Šta je sa verifikacijom?

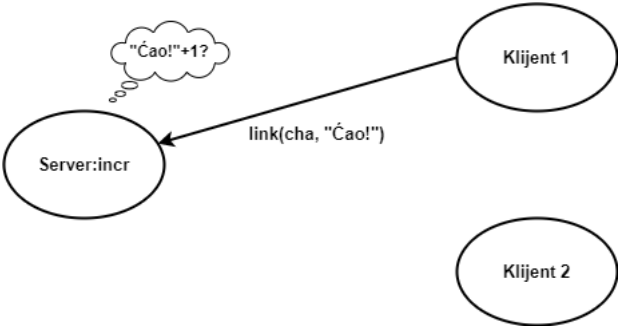
- Neke greške:



Zastoj (eng. deadlock)

Šta je sa verifikacijom?

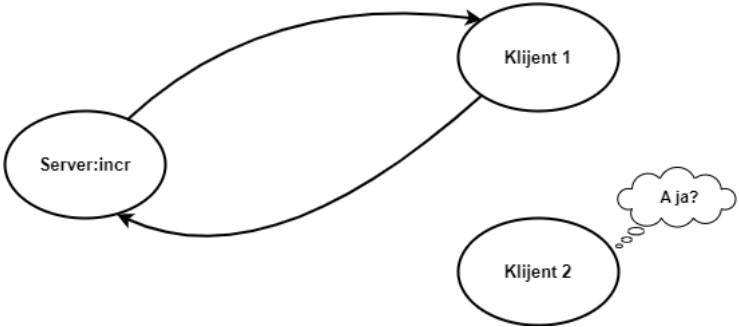
- Neke greške:



Komunikacijska neuskladjenost (eng. communication mismatch)

Šta je sa verifikacijom?

- Neke greške:



Izgladnjivanje (eng. starvation)

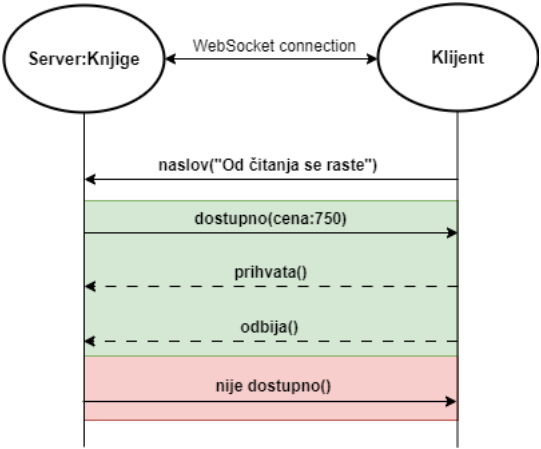
Tipovi (sesija)

- Priča kreće još od Whitehead-a, Russell-a, Ramsey-a, Church-a, Martin-Löf-a...
- Ovde zapravo mislimo na **tipove i tipske sisteme u programskim jezicima**
- *...well typed programs cannot go wrong...* (A Theory of Type Polymorphism in Programming. R Milner, Journal of computer and system sciences, 1978)
- Da probamo malo prezicnije:

  A type system is a tractable syntactic method for proving the absence of certain program behaviors by classifying phrases according to the kinds of values they compute. (Types and Programming Languages, BC Pierce, MIT Press, 2002)
- **Tipovi sesija su tipovi za komunikacijske protokole**:

  The ability to describe complex interaction protocols by means of a formal, simple and yet expressive type language...(A Gentle Introduction to Multiparty Asynchronous Session Types, M Coppo, M Dezani-Ciancaglini, L Padovani, N Yoshida, SFM 2015)

Novi primer, nova varijanta π-računa (sa labelama)

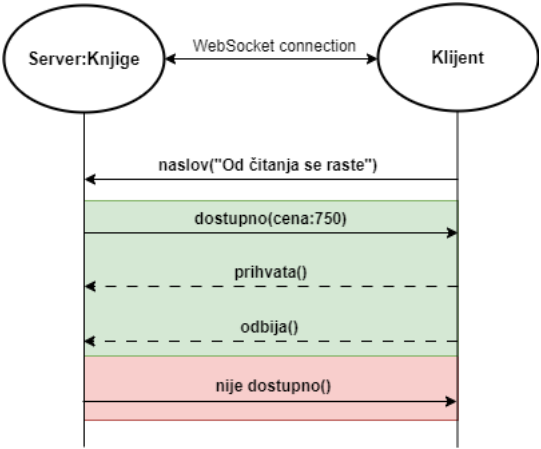


- Procesi:

$$\mathtt{Sr} = \mu X.\mathtt{ch}?nv(x).+\begin{cases}\mathtt{ch}!ds(\mathtt{broj}).+\begin{cases}\mathtt{ch}?pr\\ \mathtt{ch}?od\end{cases}\\ \mathtt{ch}!nd.X\end{cases}$$

$$\mathtt{Kl} = \mu X.\mathtt{ch}!nv(\mathtt{naslov}).+\begin{cases}\mathtt{ch}?ds(x).+\begin{cases}\mathtt{ch}!pr\\ \mathtt{ch}!od\end{cases}\\ \mathtt{ch}?nd.X\end{cases}$$

Tipovi sesija: globalni tip

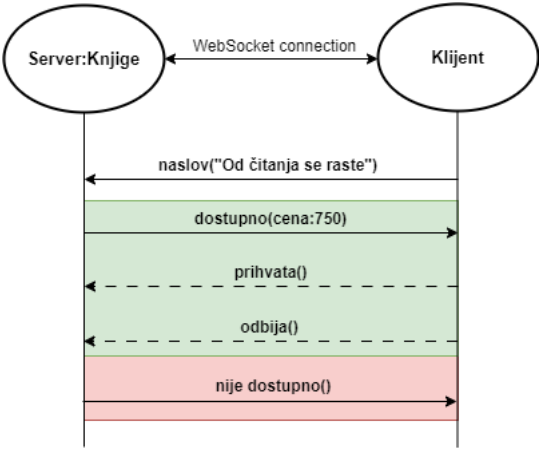


- Globalni tip komunikacije:

$$G = \mu t.\mathtt{Kl}\rightarrow\mathtt{Sr}:nv(\mathtt{str}).\begin{cases}\mathtt{Sr}\rightarrow\mathtt{Kl}:ds(\mathtt{int}).G_{ds}\\ \mathtt{Sr}\rightarrow\mathtt{Kl}:nd().t\end{cases}$$

$$G_{ds} = \begin{cases}\mathtt{Kl}\rightarrow\mathtt{Sr}:pr().\mathtt{end}\\ \mathtt{Kl}\rightarrow\mathtt{Sr}:od().\mathtt{end}\end{cases}$$

Tipovi sesija: lokalni tipovi

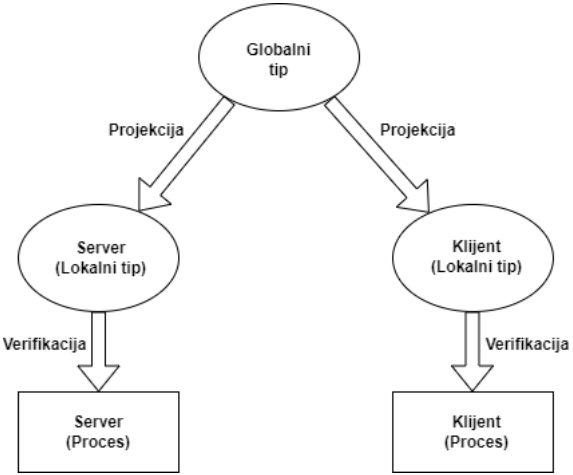


- Projekcijom globalnog tipa na učesnike dobijamo lokalne tipove:

$$\mathsf{T}_{\mathtt{Sr}} = \mu t.\mathtt{Kl}?nv(\mathtt{str}).\oplus\begin{cases}\mathtt{Kl}!ds(\mathtt{int}).\&\begin{cases}\mathtt{Kl}?pr()\\ \mathtt{Kl}?od()\end{cases}\\ \mathtt{Kl}!nd().t\end{cases}$$

$$\mathsf{T}_{\mathtt{Kl}} = \mu t.\mathtt{Sr}!nv(\mathtt{str}).\&\begin{cases}\mathtt{Sr}?ds(\mathtt{int}).\oplus\begin{cases}\mathtt{Sr}!pr()\\ \mathtt{Sr}!od()\end{cases}\\ \mathtt{Sr}?nd().t\end{cases}$$

Kako to radi?



Verifikacija:

- dodeljujemo tipove procesima

$$\mathsf{T}_{\mathtt{Sr}}, \mathsf{T}_{\mathtt{Kl}} \vdash \mathtt{Sr} \mid \mathtt{Kl}$$

- pomoću tipskog sistema, recimo

$$\frac{}{\Theta \vdash \mathtt{n} : \mathsf{nat}} \quad \frac{}{\Theta \vdash (-\mathtt{n}) : \mathsf{int}} \quad \frac{}{\Theta \vdash \mathtt{0} : \mathsf{int}} \quad \frac{}{\Theta \vdash \mathsf{true} : \mathsf{bool}} \quad \frac{}{\Theta \vdash \mathsf{false} : \mathsf{bool}}$$

$$\frac{}{\Theta \vdash () : \mathsf{unit}} \quad \frac{}{\Theta, x : \mathsf{S} \vdash x : \mathsf{S}} \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{nat}}{\Theta \vdash \mathsf{succ}(\mathtt{e}) : \mathsf{nat}} \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{int}}{\Theta \vdash \mathsf{inv}(\mathtt{e}) : \mathsf{int}}$$

$$\frac{\Theta \vdash \mathtt{e} : \mathsf{bool}}{\Theta \vdash \neg\mathtt{e} : \mathsf{bool}} \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{int}}{\Theta \vdash \mathtt{e} > \mathtt{0} : \mathsf{bool}} \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{unit}}{\Theta \vdash \mathtt{e} \approx () : \mathsf{bool}} \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{S} \quad \mathsf{S} \leq: \mathsf{S}'}{\Theta \vdash \mathtt{e} : \mathsf{S}'}$$

$$\frac{}{\vdash \varnothing : \epsilon} \quad \frac{\vdash \mathtt{v} : \mathsf{S}}{\vdash (\mathsf{q}, \ell(\mathtt{v})) : \mathsf{q}!\ell(\mathsf{S})} \quad \frac{\vdash h_1 : \sigma_1 \quad \vdash h_2 : \sigma_2}{\vdash h_1 \cdot h_2 : \sigma_1 \cdot \sigma_2}$$

$$\frac{}{\Theta \vdash \mathtt{0} : \mathsf{end}}\,[\text{T-0}] \quad \frac{\Theta \vdash \mathtt{e}_j : \mathsf{S}_j \quad \Theta \vdash P_j : \mathsf{T}_j \quad j \in I}{\Theta \vdash \mathsf{q}!\ell_j(\mathtt{e}_j).P_j : \bigoplus_{i \in I} \mathsf{q}!\ell_i(\mathsf{S}_i).\mathsf{T}_i}\,[\text{T-OUT}]$$

$$\frac{\forall i \in I \quad \Theta, x_i : \mathsf{S}_i \vdash P_i : \mathsf{T}_i}{\Theta \vdash \sum_{i \in I \cup J} \mathsf{q}?\ell_i(x_i).P_i : \&_{i \in I} \mathsf{q}?\ell_i(\mathsf{S}_i).\mathsf{T}_i}\,[\text{T-EXT}] \quad \frac{\Theta \vdash \mathtt{e} : \mathsf{bool} \quad \Theta \vdash P_i : \mathsf{T} \ (i = 1, 2)}{\Theta \vdash \mathsf{if}\ \mathtt{e}\ \mathsf{then}\ P_1\ \mathsf{else}\ P_2 : \mathsf{T}}\,[\text{T-COND}]$$

$$\frac{\Theta, X : \mathsf{t} \vdash P : \mathsf{T}}{\Theta \vdash \mu X.P : \mu \mathsf{t}.\mathsf{T}}\,[\text{T-REC}] \quad \frac{}{\Theta, X : \mathsf{t} \vdash X : \mathsf{t}}\,[\text{T-VAR}] \quad \frac{\Theta \vdash P : \mathsf{T} \quad \mathsf{T} \leqslant \mathsf{T}'}{\Theta \vdash P : \mathsf{T}'}\,[\text{T-SUB}]$$

$$\frac{\Gamma = \{\mathsf{p}_i : (\sigma_i, \mathsf{T}_i) \mid i \in I\} \quad \forall i \in I \quad \vdash P_i : \mathsf{T}_i \quad \vdash h_i : \sigma_i}{\Gamma \vdash \prod_{i \in I} (\mathsf{p}_i \triangleleft P_i \mid \mathsf{p}_i \triangleleft h_i)}\,[\text{T-SESS}]$$

(Precise Subtyping for Asynchronous Multiparty Sessions, S Ghilezan, J Pantović, I Prokić, A Scalas, N Yoshida, POPL 2021, TOCL 2022)

Neke dobre osobine koje tipovi sesija garantuju

- **Komunikacijska bezbednost** (eng. communication safety) - poslate i primljene poruke uvek su očekivanog tipa
- **Tačnost protokola** (eng. protocol fidelity) - interakcije medju učesnicima u sesiji su tačno one koje se pojavljuju u globalnom tipu
- **Napredak** (eng. progress) - svaka poslata poruka biće primljena i svaki proces koji čeka da primi poruku će je i primiti
- ...

Razne varijante tipova sesija:

- **binarne** (Types for Dyadic Interaction, K Honda, CONCUR 1993) (Language Primitives and Type Disciplines for Structured Communication-based Programming, K Honda, VT Vasconcelos, M Kubo, ESOP 1998)
- **"n-arne"** (Multiparty asynchronous session types, K Honda, N Yoshida, M Carbone, POPL 2008, Journal of the ACM 2016)
- ... razne druge varijante: sinhrone/asinhrone, proširenja i implementacije u programskim jezicima (C, Java, Go, Scala, ...)

Podtipiziranje (u sesijama)

Princip zamene Barbare Liskov u programskim jezicima:

ako je $T' \leqslant T$, tada se svaki **objekat** podtipa T' može bezbedno koristiti u svakom kontekstu gde se **objekat** nadtipa T očekuje

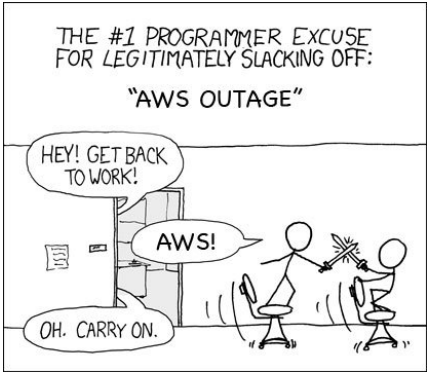
Princip zamene Barbare Liskov u komunikacijskim programima:

ako je $T' \leqslant T$, tada se svaki **proces** podtipa T' može bezbedno koristiti u svakom kontekstu gde se **proces** nadtipa T očekuje

- Podtipiziranje daje veću fleksibilnost tipskog sistema
- Postavlja se pitanje šta sve može da se uradi podtipiziranjem, tj. koja je najveća relacija podtipiziranja (Precise Subtyping for Asynchronous Multiparty Sessions, S Ghilezan, J Pantović, I Prokić, A Scalas, N Yoshida, POPL 2021, TOCL 2022)

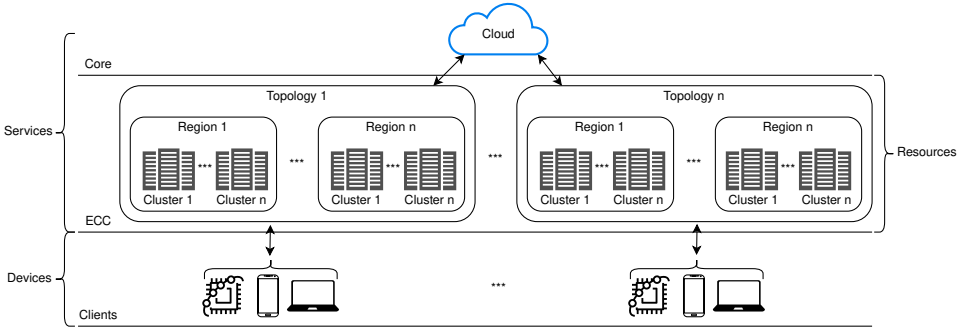
Tipovi sesija u ivičnom računarstvu

Zašto ivično računatstvo (eng. edge computing)?



- Ivično računarstvo je blizu korisnicima i tako skraćuje kašnjenje koje postoji kod računarstva u oblaku (eng. cloud computing)

Predlog organizacije ivičnog računarstva



Sinergija ivičnog i računarstva u oblaku (Towards edge computing as a service: dynamic formation of the micro data-centers, M Simić, I Prokić, J Dedeić, G Sladić, B Milosavljević, IEEE Access, 2021)

- Godišnja nagrada Matematičkog instituta SANU u oblasti računarstva za studente doktorskih studija za 2023. godinu: **Miloš Simić** i Djordje Stakić

Protokoli i njihova formalna specifikacija

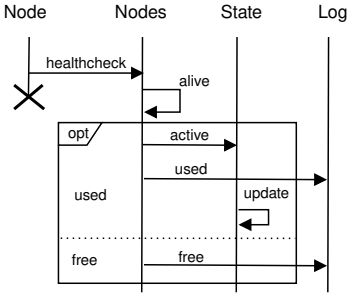
Sistem se oslanja na tri protokola:

- health-check - protokol informiše sistem o stanju svakog čvora
- cluster formation - protokol za formiranje novih klastera
- list detail - protokol prikazuje trenutno stanje sistema korisniku

Trebao nam je formalni model koji je **dovoljno ekspresivan** i **(koliko može) lak za praćenje**:

- Jedan od kandidata je bio (Multiparty asynchronous session types, K Honda, N Yoshida, M Carbone, POPL 2008)
- Medjutim, nije bio dovoljno ekspresivan...
- ... pa smo pitali Nobuko da li ima nešto bolje...
- .... i ona nas je uputila na (Explicit connection actions in multiparty session types, R Hu, N Yoshida, FASE 2017)

Health-check protokol



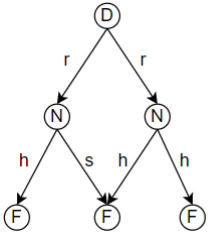
$$G_1 = \texttt{node} \twoheadrightarrow \texttt{nodes}{:}\mathit{health\_check}(T_1).\begin{cases}\texttt{nodes} \twoheadrightarrow \texttt{state}{:}\mathit{active}(T_2).\texttt{nodes} \twoheadrightarrow \texttt{log}{:}\mathit{used}(T_2)\\ \texttt{nodes} \twoheadrightarrow \texttt{log}{:}\mathit{free}(T_2)\end{cases}$$

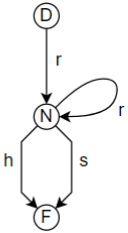
$$S_{\texttt{node}} = \texttt{nodes}!!\mathit{health\_check}(T_1)$$

$$S_{\texttt{nodes}} = \texttt{node}??\mathit{health\_check}(T_1).+\begin{cases}\texttt{state}!!\mathit{active}(T_2).\texttt{log}!!\mathit{used}(T_2)\\ \texttt{log}!!\mathit{free}(T_2)\end{cases}$$

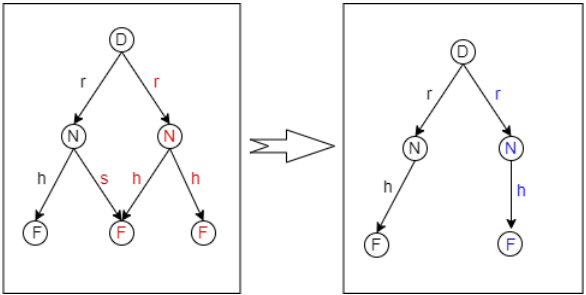
Transformacije grafova

Tipizirani grafovi i njihove transformacije



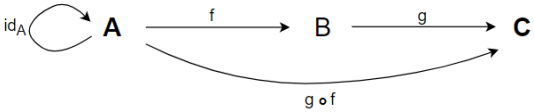


- Levo je tipiziran graf
- Desno njegov tipski graf



- U grafu levo izbrisati crveni deo
- Zatim dodati plavi i dobiti graf desno
- Po kojim pravilima raditi?

Kategorije

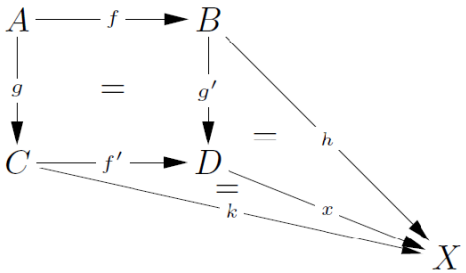


- Klasa objekata A, B, C, ...
- za svaka dva objekta skup strelica (morfizama); kompozicija
- identiteti: $f \circ id_A = f = f \circ id_B$
- asocijativnost $f \circ (g \circ h) = (f \circ g) \circ h$

Primeri kategorija:

- Kategorija skupova **Sets**: objekti su skupovi, morfizmi funkcije, kompozicija je $(g \circ f)(x) = g(f(x))$, identitet je identično preslikavanje $id_A(x) = x$
- Kategorija grafova **Graphs**: objekti su grafovi, morfizmi su homomorfizmi grafova
- Za jedan tipski graf TG, svi grafovi tipizirani sa TG i tipizirani homomorfizmi grafova čine kategoriju **GraphsTG**

Pushout u kategorijama



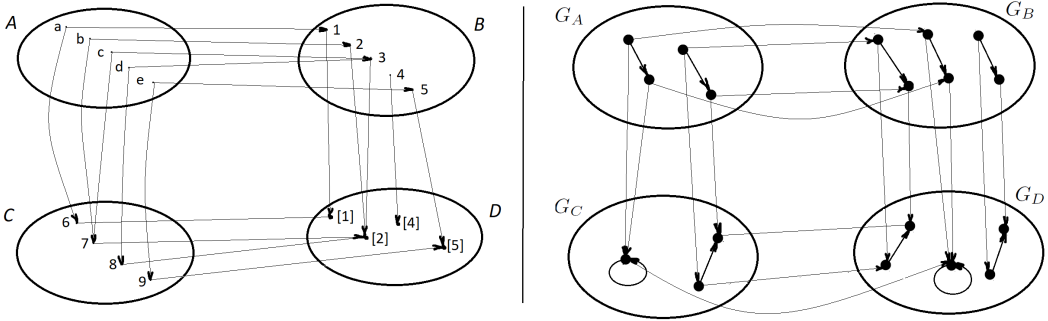
Ako u nekoj kategoriji **C** imamo morfizme $f : A \to B$ i $g : A \to C$, **pushout** (D, f', g') nad f i g je definisan sa

- pushout objektom D i
- morfizmima $f' : C \to D$ i $g' : B \to D$ takvim da $f' \circ g = g' \circ f$

i još mora da važi: Za sve objekte X i morfizme $h : B \to X$ i $k : C \to X$ takve da $k \circ g = h \circ f$, postoji jedinstven morfizam $x : D \to X$ takav da $x \circ g' = h$ i $x \circ f' = k$.

Pushout objekat je jedinstven (do na izomorfizam). Slika preuzeta iz (Fundamentals of Algebraic Graph Transformation, H Ehrig, K Ehrig, U Prange, G Taentzer, Springer-Verlag, 2006)

Pushout u skupovima i grafovima



Pushout = Lepljenje

Transformacija grafova: dupli pushout (DPO)

$$\begin{tikzcd}
L \arrow[d, "m"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "k"] & R \arrow[d, "n"] \\
G & D \arrow[l, "f"'] \arrow[r, "g"] & H
\end{tikzcd}$$

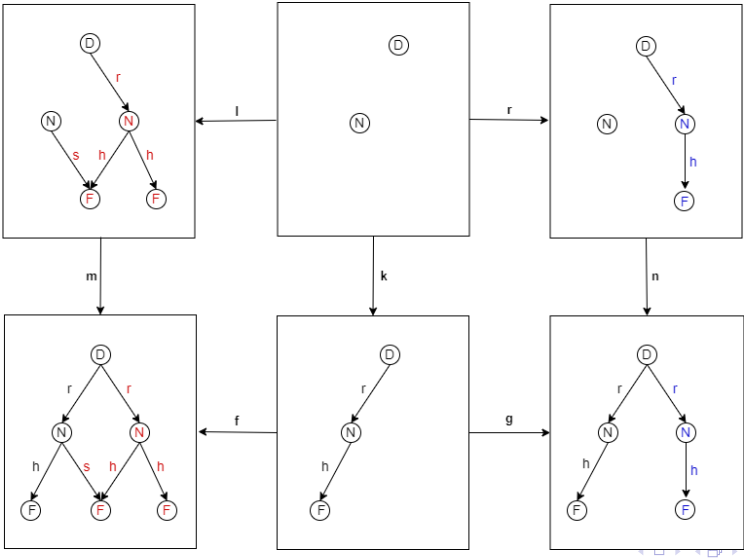
(1) (2)

- Iz grafa G brišemo podgraf $m(L \backslash l(K))$ i preko K (koji je zapravo $L \cap R$) dodajemo podgraf $n(R)$ i tako konačno dobijamo H
- Ovo radi pod uslovom da **sve tačke identifikacije** ($m(x) = m(y)$) **i sve viseće tačke** (čovor x iz L takav da je $m(x)$ incidentat sa granom koja ne pripada $m(L)$) **su tačke lepljenja** (tačke iz $l(K)$),

DPO = lepljenje dva objeka duž zajedničnog podobjekta

Slika preuzeta iz (Fundamentals of Algebraic Graph Transformation, H Ehrig, K Ehrig, U Prange, G Taentzer, Springer-Verlag, 2006)

DPO u akciji



Komentari i pitanja

Hvala na pažnji!