

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# 10. Amorfn procesiranje

II.Stopnja RI, 2015/2016

Nosilec: prof.dr.Miha Mraz

18. november  
2015



# 1. Uvod

- Amorfnno (brezoblično) procesiranje – (angl. *amorphous computing* – MIT 1996): procesiranje velikega števila identičnih paralelno delujočih entitet, ki temelji na lokalnih interakcijah [1] (podobnost s strukturami celularnih avtomatov)
- AP naj bi dalo odgovor na vprašanja:
  - Kako zagotoviti koherentno (soodvisno) željeno dinamiko na osnovi velikega števila nezanesljivih entitet, povezanih (razporejenih) na neznan in neregularen način, pri čemer se povezave skozi čas spreminjajo?
  - Kako sprogramirati posamezne procesne entitete, da bi skozi lokalne interakcije dosegli željeno globalno dinamiko (globalne vzorce)?



- Osnovne lastnosti amorfnega procesiranja [1]:
  - število entitet v strukturi je izredno veliko (npr.  $10^6$ - $10^{12}$ ), vse pa izvršujejo isti "program",
  - entitete so cenene, procesno in pomnilniško omejene, zasedajo enega od končnih možnih stanj in so sposobne generiranja naključnih števil,
  - število entitet ni odvisno od "programa",
  - entitete se ne zavedajo svoje absolutne lokacije v prostoru (zgolj relativne – sosedstva)
  - procesiranje – interakcija temelji na lokalnih povezavah, moč sporočilnosti pa pada s prepotovano razdaljo sporočila (slišnost sporočila - komunikacijski radij)



- končna dinamika (globalni vzorec) naj bi izkazoval željene zmožnosti strukture (samoorganizacija, samoreplikacija, itd.)
- delovanje posameznih entitet je nezanesljivo, zanesljivost globalne dinamike dosežena z redundanco entitet (angl. *fault tolerant computing*)
- postavitve entitet v prostoru je lahko poljubna (neregularnost strukture)
- entitete procesirajo svoja nova stanja na osnovi vplivov sosedstva
- začetna porazdelitev entitet v prostoru ne vpliva na program posamezne entitete



- Vplivi sosedstva: vpliv prihaja iz sosedstva, ki je pogojen s komunikacijskim redijem  $r$ :
  - $r$  velik glede na velikost entitete in majhen glede na velikost prostora entitet;
  - vpliv entitet v sosedstvu z oddaljenostjo od opazovane entitete pada vse do razdalje  $r$ ; če je razdalja večja od  $r$  je vpliv ničen;
- Zgled aplikacije: „*smart painting*“ za doseganje željenih geometričnih oblik vzorcev, ki jih aplikacija pobarva;



- Koncepti:
  - „*Wave propagation*“: ena od entitet pošlje sporočilo svojemu sosedstvu, le to pa ga propagira (prenaša) naprej – difuzijski val; vpeljemo števec (angl. *hop count*), ki preprečuje vzvratno širjenje vala; z analizo dolžine poti lahko raziskujemo uspešnost in zanesljivost propagacije (sorodno radijskim komunikacijskim omrežjem)
  - „*Pattern formation*“ (Coore, glej [1])



## 2. Izvajanje programa v amorfnih procesnih strukturah

- Začetna stanja vseh entitet so enaka
- Entitete v tem stanju pričakujejo vplive sosedstva, same pa vplivov ne oddajajo (stanje mirovanja, angl. *quiescent state*)
- Iz zunanjega sveta mora priti do inicializacije začetka oddajanja vpliva posameznih entitet – teh inicializacij naj bi bil čim manj
- Stanje amorfnih struktur v času  $t$  si interpretiramo kot množico stanj vseh entitet v omenjeni časovni točki (časovni vzorec)
- Dinamika v posameznih entitetah je pogojena z identičnim "programom"
- Posamezne identitete so mobilne (zmožnost gibanja)



## 3. Amorfnno procesiranje VS Celularni avtomati

- Razlike med področjema:
  - CA: postavitvev modela (celičnega programa) – simulacije – klasifikacija rezultatov (analitični koncept) = usmerjanje na posamezne interakcije
  - AP: program entitete temelji na zasnovi globalne dinamike (sintezni pristop)
  - CA celice so idealno zanesljive, AP entitete ne
  - AP entitete nehomogene v prostoru, CA celice homogene
  - število AP entitet za nekaj velikostnih razredov višje od števila celic v CA
- Več o CA (glej pod ključno besedo „game of life“)





## 4. Programski jeziki za definiranje vzorcev – dinamike prostora entitet

- Jeziki fokusirani na globalno dinamiko (opis željenega globalnega vzorca), ne na dinamiko posameznih entitet (program posamezne entitete)
- Zgodovinski predhodnik: jezik Logo
- Za kompleksno preslikavo željene globalne dinamike v lokalne interakcije (program) naj bi bili zadolženi prevajalniki tovrstnih jezikov
- Primeri jezikov:
  - Growing Point Language (GPL) – 1999
  - Origami Shape Language (OSL)
  - Ecoli (angl. *Extensible calculus of loacal interactions*)
  - ProtoLanguage (2006)
  - Paintable computing language (2002)



## 5. Growing point language

- Avtor: Daniel Coore – MIT – 1999 [2]
- GPL: jezik za abstrakcijo lokalnih interakcij na osnovi globalno definirane dinamike (realizacija – interpreter)
- Osnovni pojmi jezika:
  - **Tropizem** (angl. *tropism*): biološki fenomen, ki indicira rast (helio, chemo, gravi, hidrotropizem, itd.) kot odziv na okoljske stimulanse
  - Primitivni podatkovni tip **feromon** (angl. *pheromone*): služi kot implementacija tropizma; po vplivu radialno simetričen in monotonno padajoč z oddaljenostjo od izvira (brez feromona ni dinamike)
  - **Rastoče točke** (angl. *growing points*): prostorsko omejene množice entitet (v času), ki tvorijo vzorce; po definiciji je rastoča točka v času  $t$  prisotna glede na tropizem le v eni aktivni entiteti; se premikajo, delijo, spajajo, umirajo, itd.



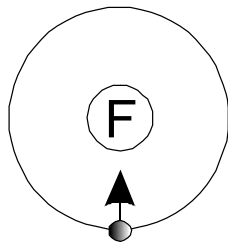
- Rastoča točka pri potovanju po površini ali prostoru (sosedstvu) spreminja stanja entitet, ki jih obišče



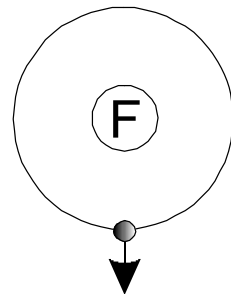
- primitivni podatkovni tip **material** (angl. *marker*): marker beleži prepotovano pot rastoče točke v času (barvanje entitet); rastoče točke ga občutijo kot "material" in se glede na njegovo naravo odočajo o nadaljnji dinamiki (lahko material tudi preoblikujejo)
- pot rastoče točke je vidna le, če pri tem prihaja do odlaganja materialov v aktivnih točkah (barvanja)
- Koncept rastoče točke: skupek aktivnosti, ki prenaša pulz preko sosedov na ciljne pozicije, ki jih predstavljajo feromoni; pri tem rastoča točka skozi čas v aktivnih entitetah odlaga material ali feromone



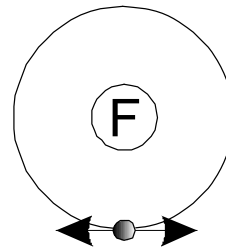
- Vrste feromonov, glede na njihov vplivnost:



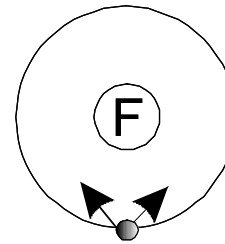
Ortho+



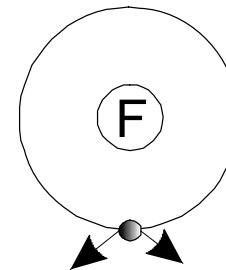
Ortho-



Diam



Plagio+



Plagio-



- Struktura GPL programa:
  - Definicije množice rastočih točk (tendenca: ne preveliko število)
  - Režija za zagon dinamike
- Definicija posamezne rastoče točke:
  - Atributi rastoče točke (deklaracijski del)
  - Instrukcije rastoče točke (programski del): instrukcije se izvajajo sekvenčno po vrstnem redu navedbe v kodi;

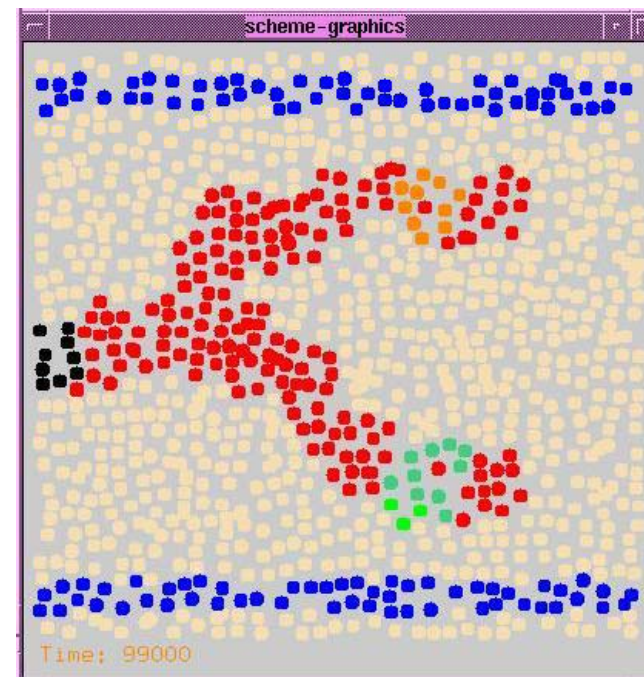
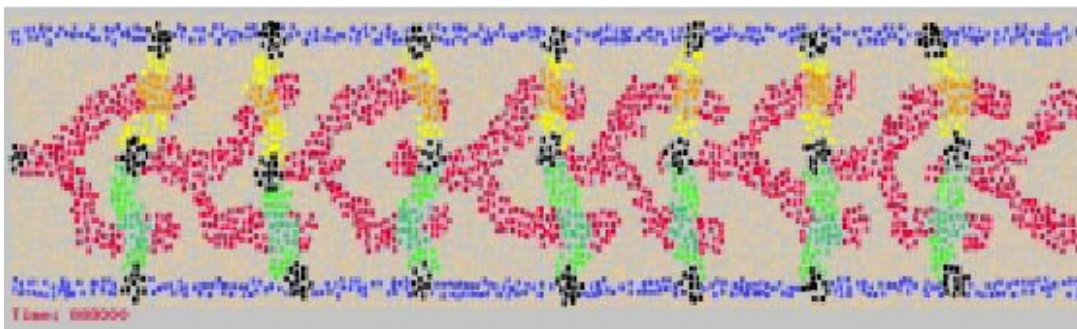


# Zgled 1

```
(define-growing-point(make-red-branch length)
                      // "make-red-branch length" ime rastoče točke
(material red-stuff) // material rastoče točke bo „red-stuff“
(size 5)            // velikost materiala bo 5 (val širjenja materiala bo imel
                  // doseg 5 skokov)
(tropism (and (away-from red-pheromone)
              (and (keep-constant pheromone-1)
                   (keep-constant pheromone-2))))
        // tropizem bo pot rastoče točke odvrčal od visokih
        // koncentracij „red-pheromone“
(avoid green-pheromone)
(actions
 (secret 2 red-pheromone)
 // vse entitete, v katerih je „red-stuff“ material oddajajo
 // skrivnost (secret) „red-pheromone“; s tem se rastoča
 // točka izogiba že prehojeni poti
(when (< length 1)
      (terminate)
      (default
        (propagate (- length 1))))))
```



- Končni vzorec – slika levo (program na prejšnji prosojnici definira le formiranje rdečega materiala)
- “CMOS circuit layout” – slika spodaj

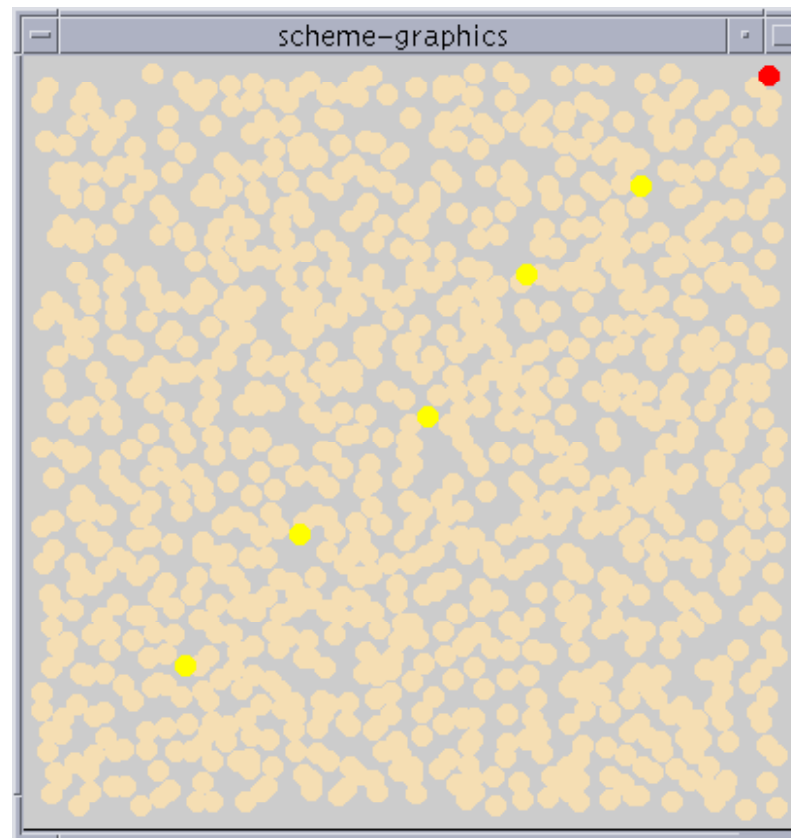
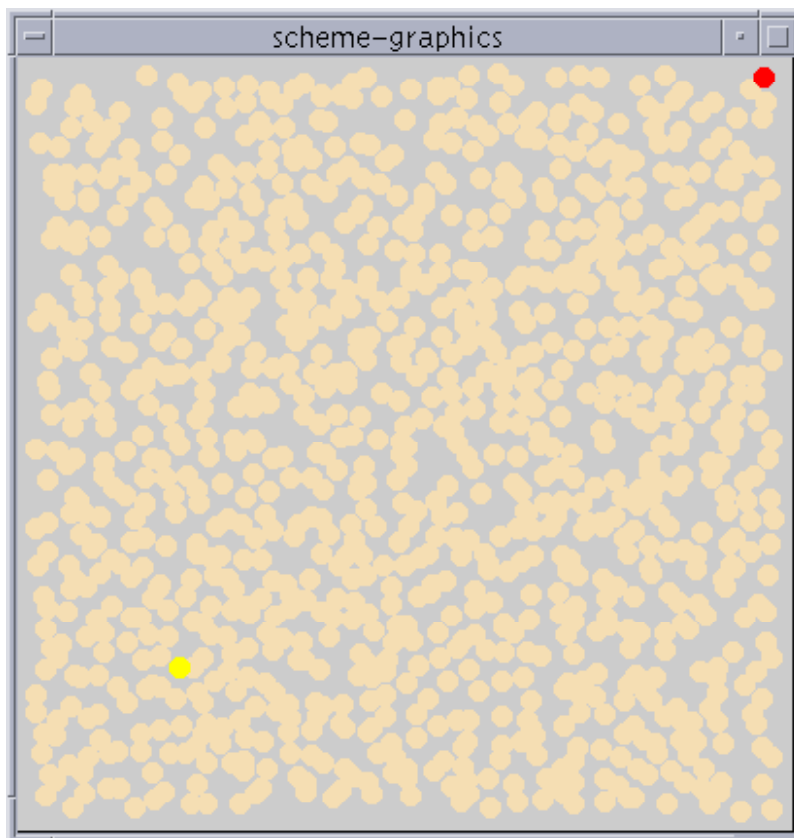






## Zgled 2

```
A1: (define-growing-point(A-to-B)
    // "A-to-B" ime rastoče točke
A1: (material A-material)
A1: (size 0)
A1: (tropism(ortho+ B-pheromone))
    //tropizme lahko spajamo preko AND, NOT-AND,OR in NOT-OR operatorjev
I1: (for each step
    I1: (when ((sensing? B-material)
    I1: (terminate))
    I1: (default(propagate))))
A2: (define-growing-point(B-point)
A2: (material B-material)
A2: (size 0)
I2: (for-each-step
    I2: (secret+ 10 B-pheromone)))
    // "secret" ukaz formira tvorbo feromona
Ini: (color
Ini: ((B-material) "red")
Ini: ((A-material) "yellow")
Ini: (with-locations
Ini: (a b)
Ini: (at a (start-growing-point A-to-B) Ini: (at b (start-growing-point B-point)))
```



10. Amorfno procesiranje



- Nekaj zanimivih ukazov:
  - `avoids`: ukaz omogoča, da dobi feromon inhibirni (odbojni) vpliv;
  - `start-growing-point`: invokacija rastoče točke,
  - `propagate`: iskanje nove lokacije rastoče točke (nove lokacije aktivne točke),
  - `terminate`: zaključek dinamike rastoče točke,
  - `secrete`: definicija koncentracije (obsega vplivnosti) feromona,
  - `when`: pogojni stavek za vejanje izvajanja akcij,
  - `sensing?`: pogojni stavek za detekcijo materiala v aktivni točki,
  - `color`: omogoča barvanje materialov



- Temelji orisa vzorca:
  - Abstraktne entitete (rastoče točke in feromoni)
- GPL prevajalnik prevede izvorno kodo v program posamezne entitete



- Zgledi rezultatov AP na osnovi GPL zapisa:
  - <http://isandtcolloq.gsfc.nasa.gov/fall2002/presentations/abelson.ppt>
  - <http://www.cs.virginia.edu/~evans/cs655/projects/zhong.ppt>
  - <http://necsi.org/events/iccs6/viewpaper.php?id=218>



## 6. Preostali jeziki

- Origami Shape Language:
  - Temelji na principu Origami zgibank (avtor Nagpal, 2001)
  - Program za formacijo vzorca – sekvenca pregibov prvotnega materiala
- Jezik Ecoli:
  - GPL sekvence ukazov se prevedejo v Ecoli notacijo – program posameznih entitet

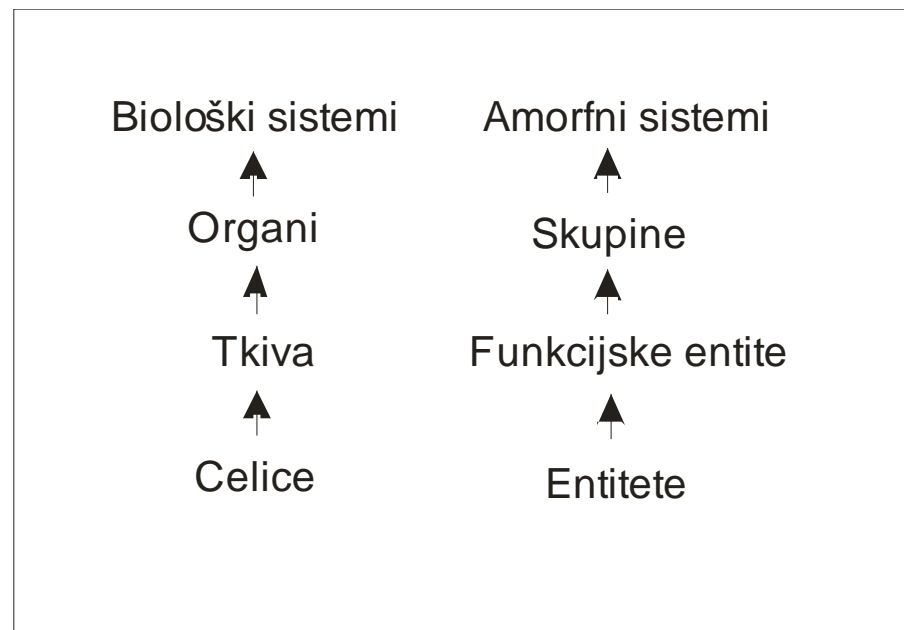


## 7. Ostale značilnosti AP

- Nezanosljivost entitet rešena z redundanco
- Vztrajnostna jedra (množica entitet z ustreznimi materiali, angl. *persistent nodes*): prostorsko strnjene množice entitet, ki so z globalnega vidika urejene; lastnost vztrajnostnega jedra ima takšna množica takrat, ko se ob odvzemu dela entitet vzorec skozi čas obnovi; ni nujno, da se obnovi na istem mestu v prostoru; to se lahko izvede kje drugje v amorfnem substratu;



- Primerjava hierarhije gradnikov v bioloških in amorfnih sistemih:







## 8. Zaključek

- Na nivoju implementacije povezovanje s področji:
  - MEMS, NEMS (nezanesljivi mikro senzorji, aktuatorji, itd.); „*particles that could be mixed with materials, such as paints, gels, and concrete*“; „*computational particles*“
  - Vseprisotno računalništvo (angl. *ubiquitous computing* – ubicomp)
  - „*Cellular engineering*“ kot ena od panog prihodnosti (*chemical factories for the assembly of nanoscale structures*)
  - Sorodno področju senzorskih omrežij;



## 9. Literatura

[1] H.Abelson et.al.: Amorphous computing, Communications of the ACM, May 2000, Vol.43, No.5. (link na članek dostopen iz spl.učilnice)

[2] D.N.Coore: Botanical Computing: A developmental approach to generating Interconnect topologies on an Amorphous Computer (PhD thesis)