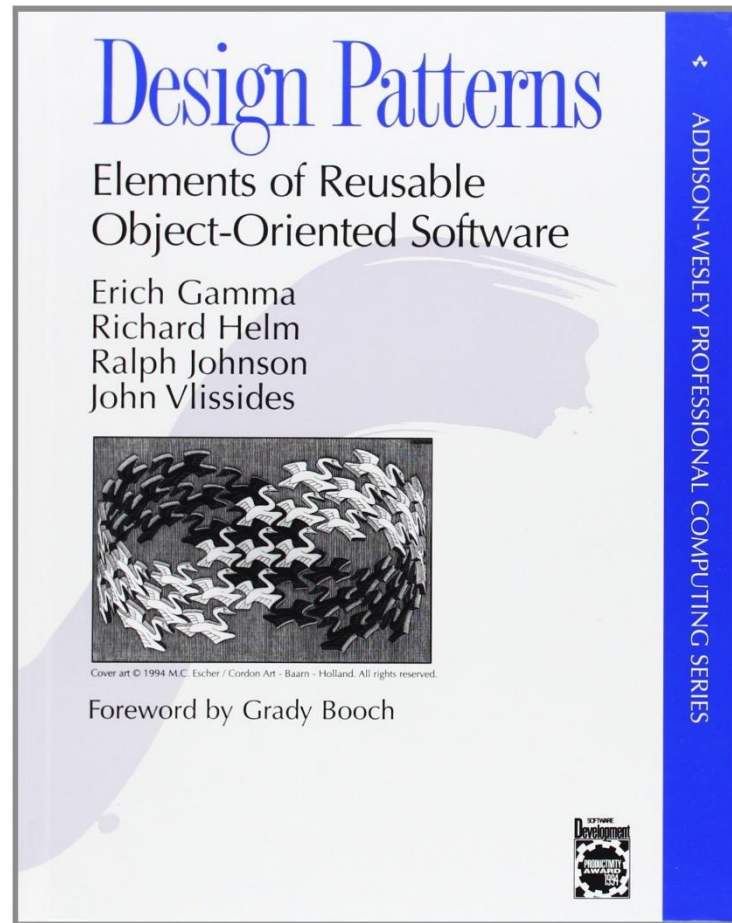


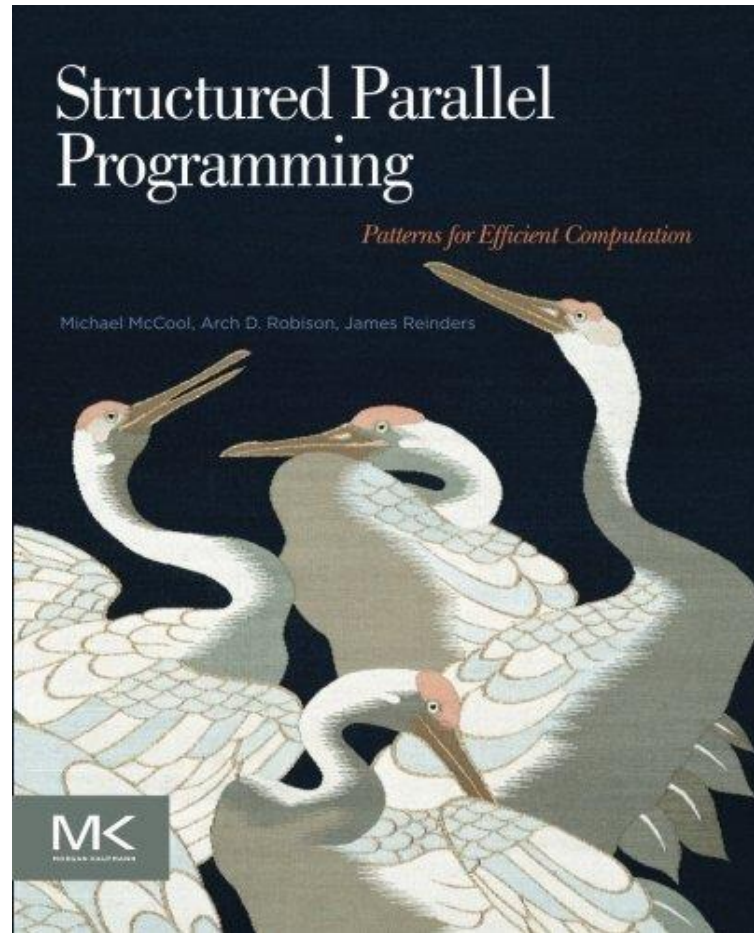
Projektni obrasci u paralelnom programiranju

Literatura – Projektni obrasci „GoF“



https://en.wikipedia.org/wiki/Design_Patterns

Literatura – Paralelni obrasci



<http://parallelbook.com/>

Strukturirano programiranje sa obrascima

- **Projektni obrasci** (engl. *design patterns*) ili, kraće, **obraci** predstavljaju **primere najbolje prakse** (engl. *best practices*) za **rešavanje određenih problema** u softverskom inženjerstvu
 - Prvi put predstavljeni u kontekstu objektno-orijentisanog programiranja u knjizi „Design Patterns: Elements of Reusable Object-Oriented Software“, poznatoj i kao „Gang of Four – GoF“ 1994.
- Obrasci su **algoritamske strukture** koje se često javljaju u **efikasnim programima**, koriste se za **organizaciju algoritama**, poznati i kao **algoritamski skeleti**
- Primena **obrazaca** vodi do **efikasne implementacije** odgovarajućih **algoritamskih struktura**, programi sa obrascima **bolje skaliraju** i **lakše se održavaju**
- **Obrasci** se mogu posmatrati i kao dalje **proširenje ideje strukturiranog programiranja**, gledno „**spolja**“ obrasci su **apstrakcije**, gledano „**iz programa**“ predstavljaju **efikasne implementacione strategije**
- **Obrasci** su **univerzalni**, nisu vezani ni za jednu određenu arhitekturu hardvera, programski jezik ili sistem

Klasifikacija obrazaca

- Osnovni tipovi obrazaca:
 - **Ugnježdavanje** (engl. *nesting pattern*) – omogućava slaganje (kompoziciju) ostalih obrazaca
 - **Serijski i paralelni obrasci za upravljanje tokom programa** (engl. *control patterns*)
 - **Serijski i paralelni obrasci za upravljanje podacima** (engl. *data management patterns*)
 - Ostali obrasci

Strukturirani serijski obrasci

- Osnovu **strukturiranog programiranja** serijskih algoritama čine sledeći **obraci**:
 - **Upravljanje tokom**:
 - ugnježdavanje
 - sekvenca
 - selekcija
 - iteracija
 - rekurzija
 - **Upravljanje podacima**:
 - slučajno čitanje
 - slučajni upis
 - alokacija steka
 - alokacija heap-a
 - objekti
 - zatvaranja (engl. *closures*)
- Primenom ovih obrazaca **naredbe bezuslovnog skoka** (`goto`) mogu se uglavnom **eliminirati iz koda** i na taj način program učiniti efikasnijim, čitljivijim i lakšim za održavanje

Strukturirani serijski obrasci

- **Upravljanje tokom** kod strukturiranog serijskog programiranja se zasniva na obrascima: **sekvenca, selekcija, i iteracija** (kao i **rekurzija**)
- Obrazac **ugnježdavanja** se koristi za hijerarhijsko slaganje pomenutih osnovnih obrazaca
- Poznavanje serijskih obrazaca je važno i kada se radi paralelizacija serijskih algoritama
- **Upravljanje podacima** (engl. *data management*) se odnosi na to kako se podaci alociraju, dele, čitaju, pišu i kopiraju
- Serijski obrasci za upravljanje podacima su: **slučajno čitanje i pisanje, alokacija steka, alokacija heap-a, objekti, zatvaranja**

Strukturirani paralelni obrasci

- Osnovu **strukturiranog paralelnog programiranja** čine sledeći obrasci:
 - superskalarna sekvenca
 - spekulativna selekcija
 - mapiranje (engl. *map*)
 - rekurenca (engl. *reccurence*)
 - skeniranje (engl. *scan*)
 - redukcija (engl. *reduce*)
 - pakovanje-ekspanzija
 - račvanje-spajanje (engl. *fork/join*)
 - protočni sistem (engl. *pipeline*)
 - particija
 - segmentacija
 - šablon (engl. *stencil*)
 - traži-upari (engl. *search/match*)
 - okupljanje (engl. *gather*)
 - rasejavanje (engl. *scatter*)
- Primenom ovih obrazaca, **niti** i **intrinzičke vektorske naredbe** mogu se uglavnom **eliminirati iz koda** i na taj način paralelni program učiniti efikasnijim, čitljivijim i lakšim za održavanje

Paralelni obrasci

- **Paralelni obrazac** se može definisati i kao **kombinacija distribucije zadataka i načina pristupa podacima** koja se **često ponavlja**, a **rešava određeni problem** prilikom projektovanja paralelnih algoritama
- U ovom smislu, **projektovanje paralelnih algoritama** se može svesti na **kompoziciju (slaganje) paralelnih obrazaca**
- **Odnos između serijskih i paralelnih obrazaca:**

Serijski obrazac	Paralelni obrazac
ugnježdavanje	
sekvenca	superskalarna sekvenca
selekcija	spekulativna selekcija
iteracija	mapiranje, redukcija, skeniranje, rekurencija, okupljanje, rasejavanje

Paralelni obrasci za upravljanje tokom programa

Paralelni obrasci za upravljanje tokom

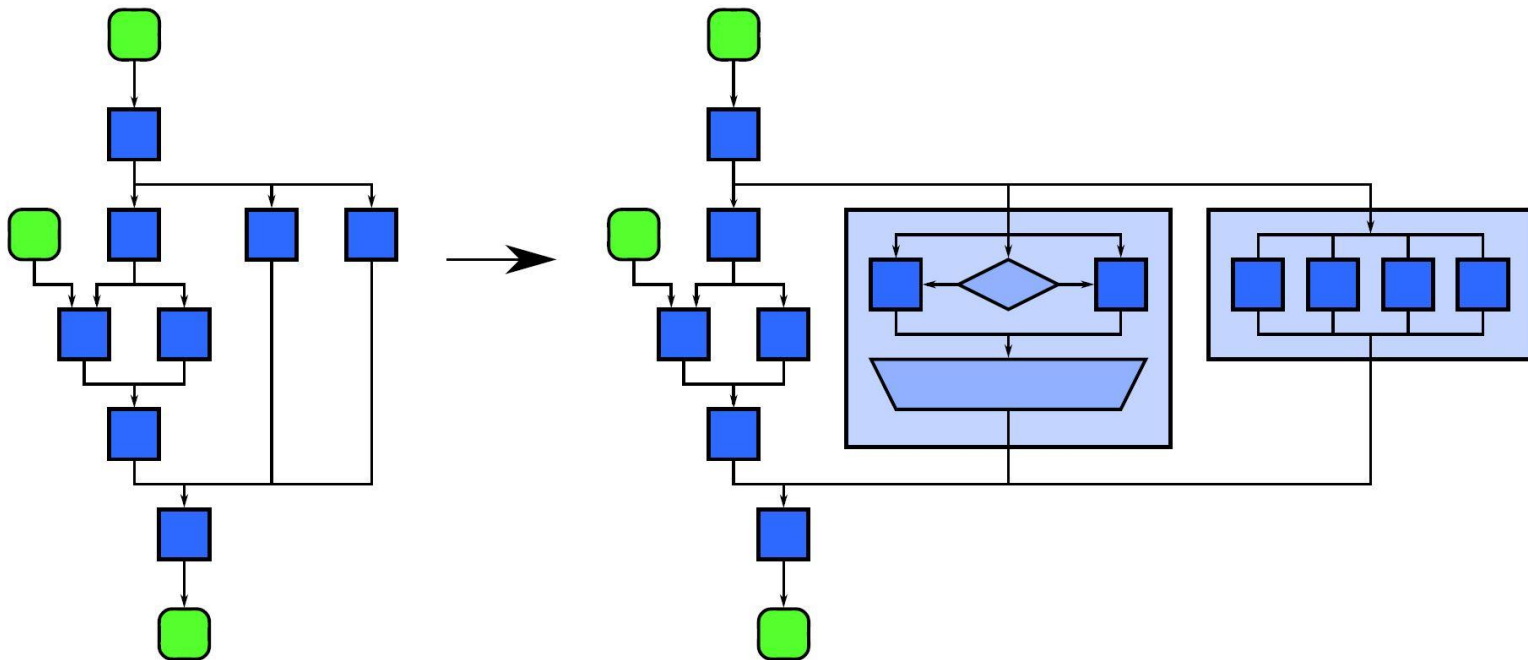
- **Paralelni obrasci za upravljanje tokom programa** proširuju odgovarajuće serijske obrasce
- Svaki paralelni obrazac za upravljanje tokom je u vezi sa najmanje jednim serijskim obrascem, ali sa relaksiranim pretpostavkama u odnosu na serijski slučaj
- Paralelni obrasci za upravljanje tokom programa su:
 - **račvanje-spajanje**
 - **superskalarna sekvenca** (upisuje sekvencu zadataka uređenih samo po zavisnostima)
 - **spekulativna selekcija** (opšta verzija selekcije gde uslov i oba ishoda mogu da se izvršavaju paralelno)
 - **mapiranje**
 - **redukcija**
 - **skeniranje**
 - **šablon**

Ugnježdavanje

- **Ugnježdavanje** je hijerarhijsko slaganje obrazaca
- Ovaj obrazac koristi se prilikom **projektovanja i serijskih i paralelnih algoritama**, omogućava da se **od prostih osnovnih obrazaca realizuju i najkompleksniji algoritmi** koji se mogu izvršiti na bilo kojoj Turingovoj mašini
- **Dijagrami obrazaca** (engl. *pattern diagrams*) se koriste za vizuelni prikaz ideje obrasca, pri čemu svaki **blok zadatka** predstavlja lokaciju opšteg koda u algoritmu
- Svaki blok zadatka može dalje biti još jedan obrazac u obrascu ugnježdavanja do proizvoljne dubine

Ugnježdavanje

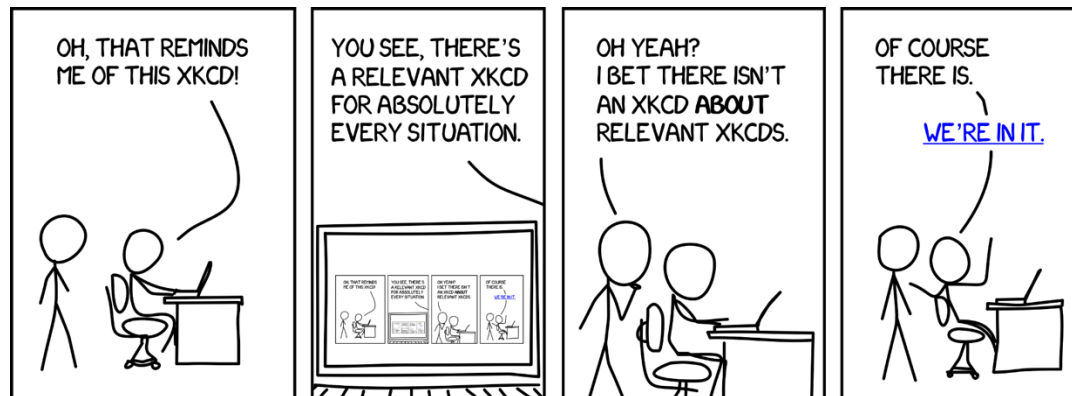
- **Ugnježdavanje je obrazac slaganja.** Omogućava da se drugi obrasci slože u hijerarhiju tako da bilo koji blok u dijagramu može da se zameni obrascem sa istim ulazom/izlazom i zavisnostima



Izvor: <http://parallelbook.com/>

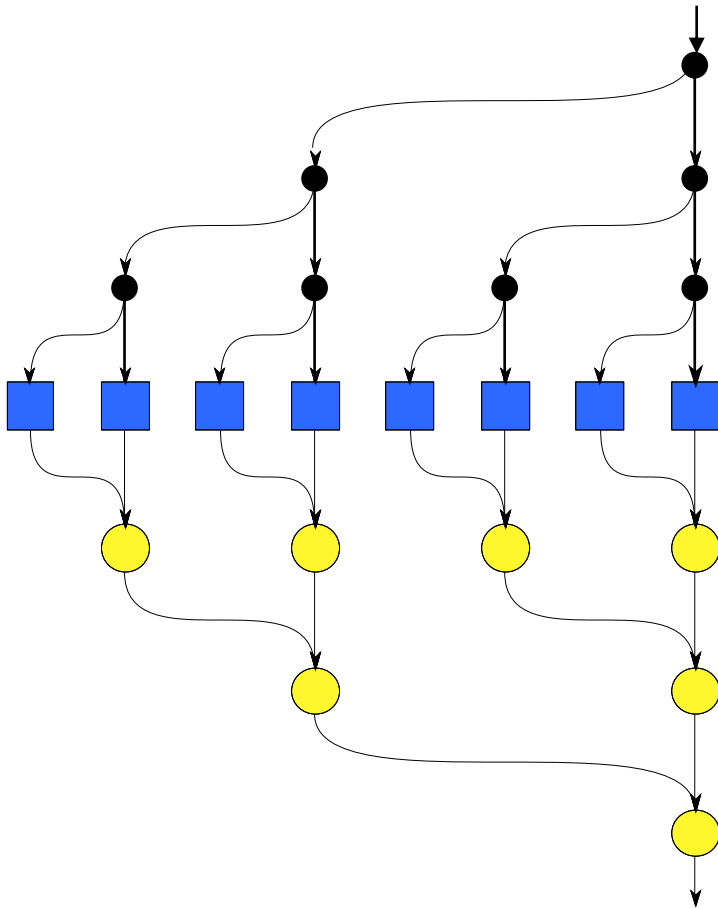
Rekurzija

- Rekurzija je **dinamički vid ugnježdavanja**, omogućava da funkcije **direktno ili indirektno pozivaju same sebe**
- Rekurzija je važan **univerzalni serijski obrazac**
 - **Rekurzija** je osnova **funktionalnog programiranja**
 - **Iteracija** je osnova **proceduralnog programiranja**
- **Strukturalna rekurzija**: ugnježdavanje komponenti
- **Dinamička rekurzija**: ugnježdavanje ponašanja



Izvor: <https://xkcd.com>

Efikasno ugnježdavanje – račvanje-spajanje



- **Račvanje-spajanje** (engl. *fork-join*) je obrazac koji omogućava razdvajanje toka programa u više paralelnih grana, koje se potom mogu spojiti i nastaviti sa sekvencijalnim izvršavanjem, može se ugnježdavati
- Balans između cenu distribucije posla i sinhronizacije
- **Rekurzivna primena** obrasca račvanje-spajanje omogućava **visok stepen paralelizma – fina granularnost** i podeli-i-vladaj (engl. *divide-and-conquer*) pristup

Izvor: <http://parallelbook.com/>

(Serijska) sekvenca

- **Sekvenca:** uređeni niz zadataka koji se izvršavaju u određenom redosledu, prethodni zadatak se mora završiti pre nego naredni krene
- Pretpostavka – prilikom izvršavanja biće praćen redosled zadataka kako su navedeni u programu (očigledno, ali važno kod paralelizacije)

```

1  T = f(A);
2  S = g(T);
3  B = h(S);

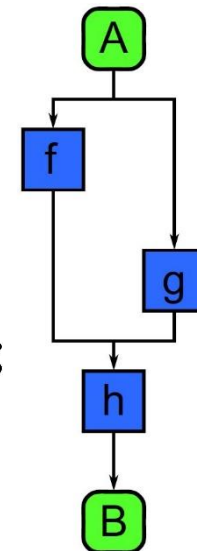
```



```

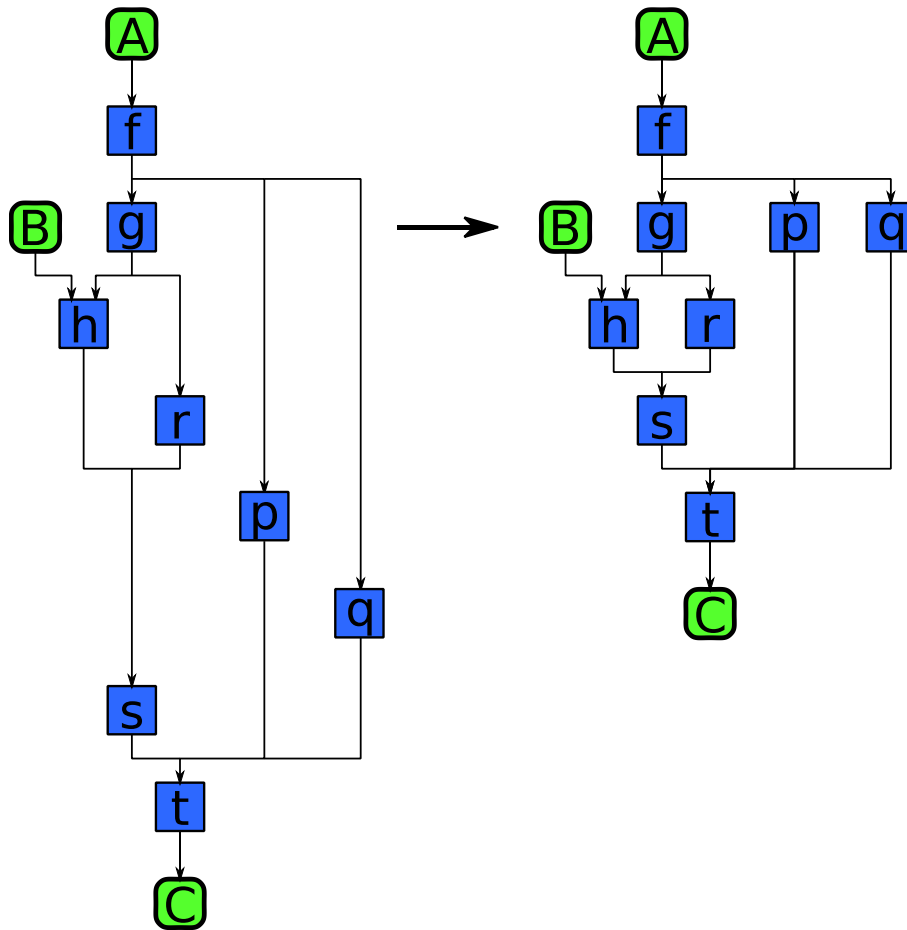
1  T = f(A);
2  S = g(A);
3  B = h(S, T);

```



Sekvenca uređuje operacije u redosledu njihovog pojavljivanja u programskom kodu, čak i kada nema očigledne zavisnosti između zadataka.

Superskalarna sekvenca



Programer piše serijski kod:

```

F = f(A);
G = g(F);
H = h(B,G);
R = r(G);
P = p(F);
Q = q(F);
S = s(H,R);
C = t(S,P,Q);

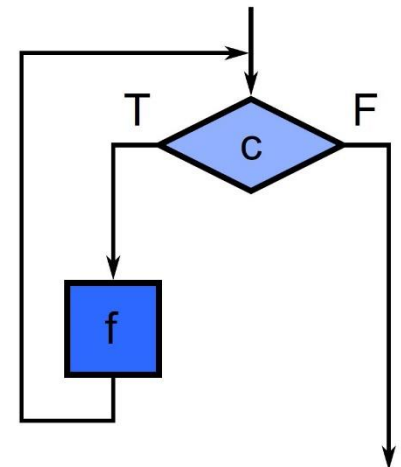
```

Zadaci se uređuju samo prema zavisnostima po podacima i izvršavaju se čim su ulazni podaci spremni

(Serijska) iteracija

- Obrazac **iteracija** ponavlja izvršavanje određenog skupa naredbi dok god postavljeni uslov važi
 - uslov c se evaluira. Ako je rezultat tačan, a se evaluira, i onda se c ponovo evaluira. Proces se ponavlja dok c ne postane netačno
 - Petlja se može prekinuti u bilo kom trenutku na osnovu izračunavanja u bilo kojoj iteraciji
- Komplikacija prilikom paralelizacije: svaka iteracija može zavisiti od vrednosti iz bilo koje od prethodnih iteracija

<pre> 1 for (i = 0; i < n; 2 a; 3 }</pre>	<pre> 1 while (c) { 2 a; 3 }</pre>
---	---

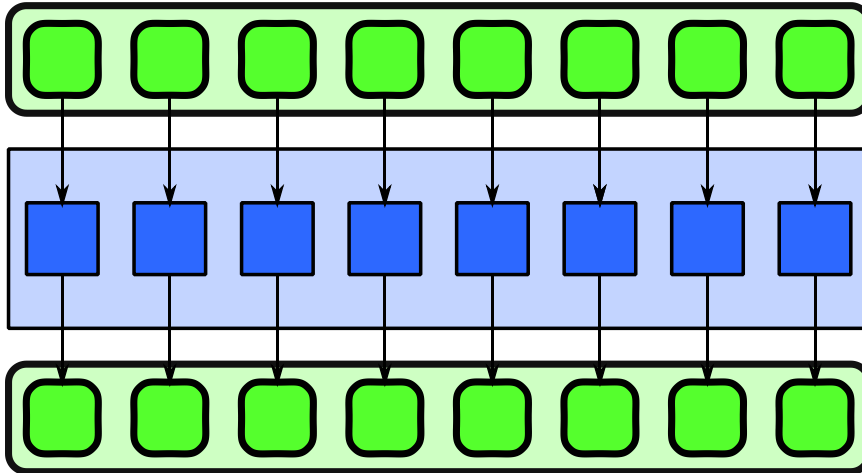


Izvor: <http://parallelbook.com/>

Paralelna „iteracija“

- Serijskom obrascu **iteracija** odgovara **više različitih paralelnih obrazaca** u **zavisnosti** od toga **da li i na koji način iteracije zavise jedne od drugih**
- Najveći broj **paralelnih obrazaca** koji **odgovara serijskoj iteraciji** zahteva **unapred poznat fiksni broj poziva naredbi u telu petlje**
- **Paralelni obrasci koji odgovaraju iteraciji:**
 - **mapiranje**
 - **redukcija**
 - **skeniranje**
 - **rekurencija**
 - **okupljanje**
 - **rasejavanje**

Mapiranje



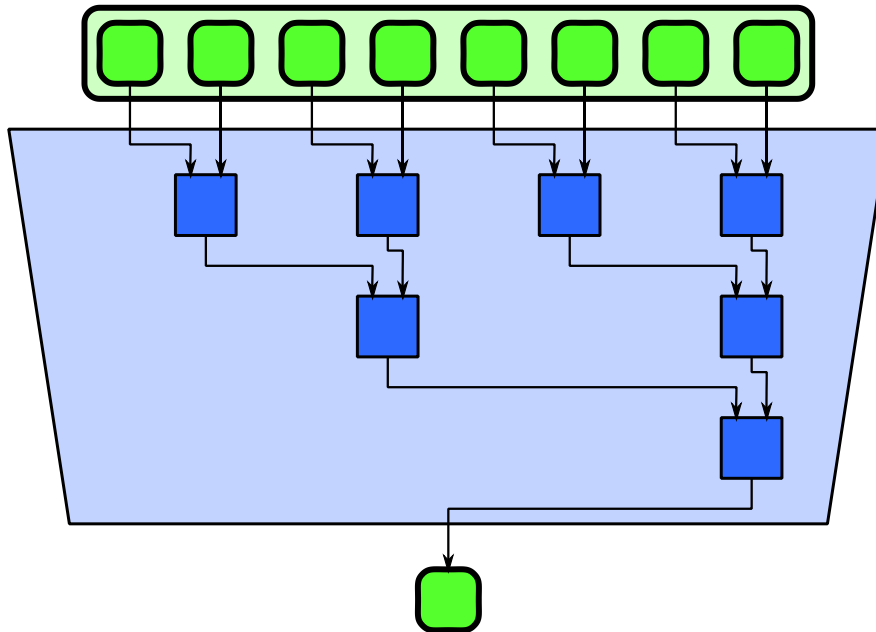
Primeri mapiranja: korekcija game i thresholding kod slika, konverzije modela boja, Monte Karlo uzorkovanje, algoritmi praćenja zraka (engl. *ray tracing*)

- **Mapiranje** replicira funkciju nad svakim elementom indeksnog skupa
- **Indeksni skup** može biti apstraktan ili pridružen elementima niza

```
for (i=0; i<n; ++i) {
    f(A[i]);
}
```

- Mapiranje replicira **jedan specifičan vid iteracije** u serijskim programima – **nezavisne operacije**

Redukcija



Primeri redukcije: matrične operacije, usrednjavanje Monte Karlo uzoraka, provera konvergencije, metrike za poređenje slika

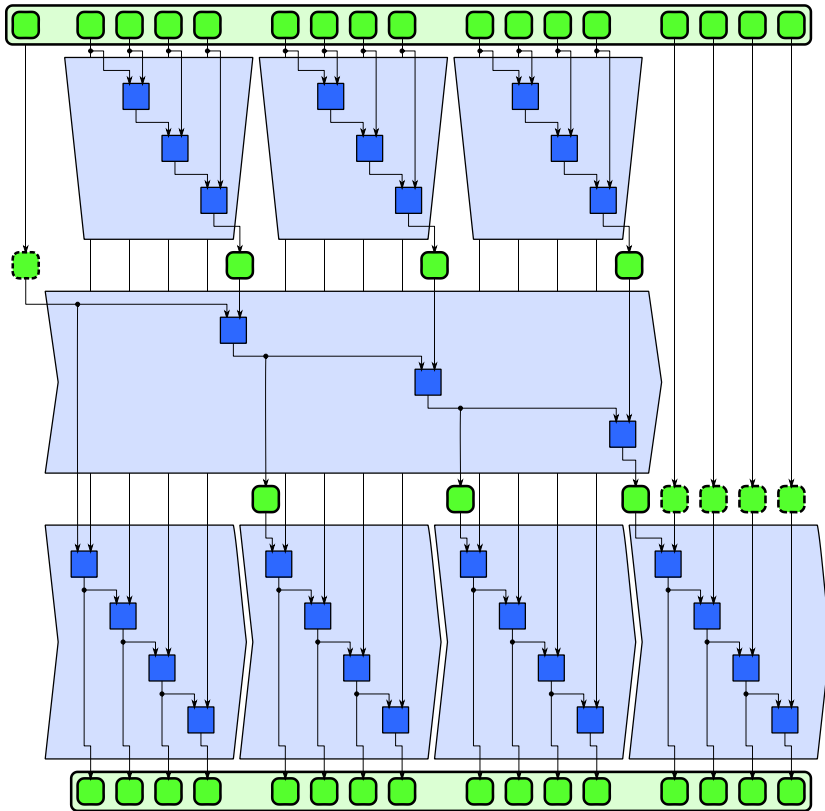
- **Redukcija** kombinuje svaki element kolekcije u jedan element primenom asocijativnog operatora

```
b = 0;
for (i=0; i<n; ++i) {
    b += f(B[i]);
}
```

- **Preuređenje operacija** je često **neophodno** kako bi se omogućila **paralelizacija**

Izvor: <http://parallelbook.com/>

Skeniranje



- **Skeniranje** računa sve parcijalne redukcije kolekcije

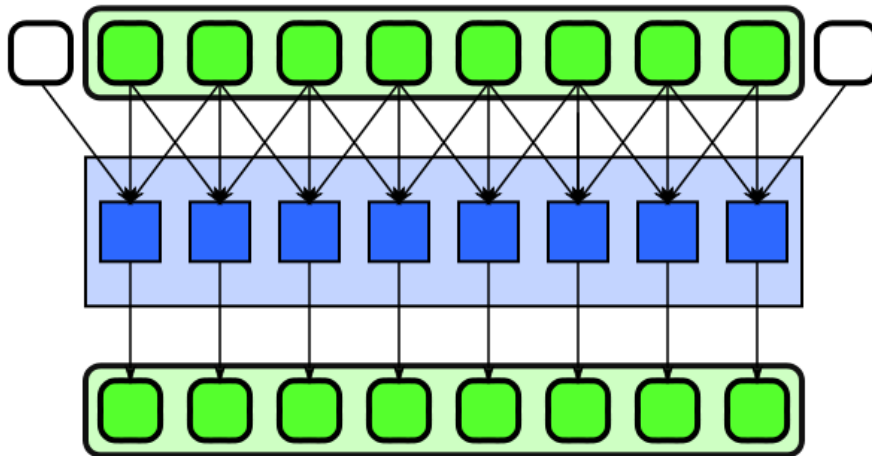
```
A[0] = B[0] + init;
for (i=1; i<n; ++i) {
    A[i] = B[i] + A[i-1];
}
```

- Operator mora biti (najmanje) asocijativan
- Na slici je prikazana jedna moguća paralelna implementacija na bazi skeniranja sa tri faze

Primeri skeniranja: generisanje slučajnih brojeva, analiza vremenskih serija, kompresija podataka, tabelarna integracija

Izvor: <http://parallelbook.com/>

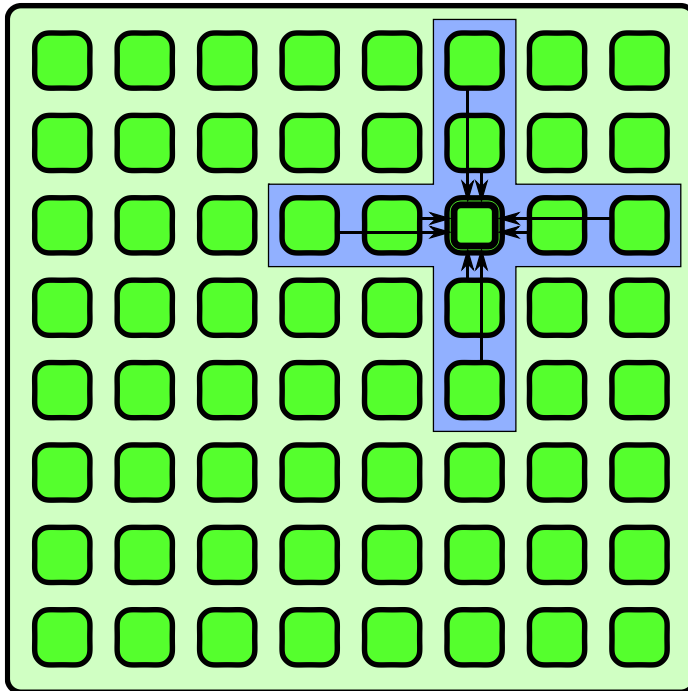
Šablon



Primeri šablona: obrada signala,
spektralne transformacije,
konvolucija

- **Šablon** (engl. *stencil*) primenjuje funkciju (tipično fiksni šablon) nad susedstvima kolekcije
- Šabloni realizuju niz prolazaka (vremenskih koraka) kroz datu kolekciju
- **Susedstva** se definišu preko relativnih pomeraja
- **Granični uslovi** se moraju uzeti u obzir, ali većina izračunavanja su u „unutrašnjosti“

nD šablon



- **nD šablon** primenjuje funkciju nad susedstvima n -dimenzionalnog niza
- **Susedstva** se definišu preko skupa relativnih pomeraja
- **Granični uslovi** se moraju uzeti u obzir
- **von Neumann-ovo** i **Moore-ovo** susedstvo

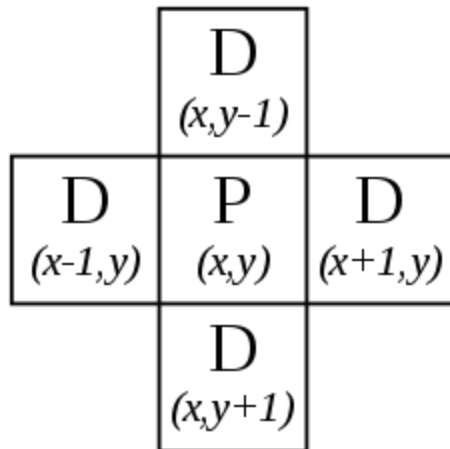
Primeri nD šablona: Jakobijev metod, ćelijski automati, obrada slika, simulacije kao što je npr. protok fluida, rešavanje parcijalnih diferencijalnih jednačina u elektromagnetici i finansijama, kvatna fizika (lattice QCD – *quantum chromodynamics*)

Izvor: <http://parallelbook.com/>

nD šablon – susedstva

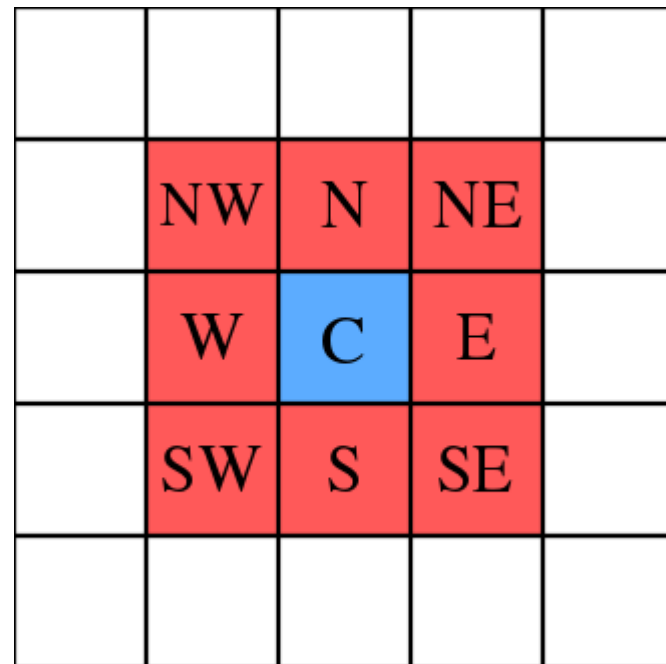
- Susedstva:

- von Neumann



- Moore

- Primer: Conway's Game of Life



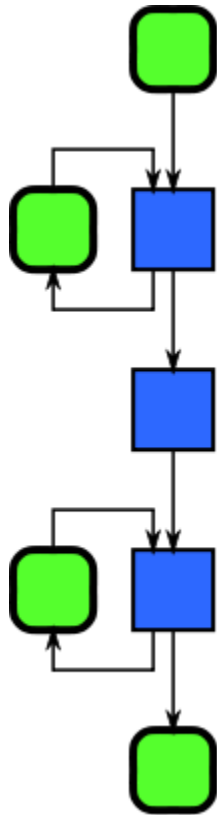
Izvor: https://en.wikipedia.org/wiki/Stencil_code

Paralelni obrasci za upravljanje podacima

Paralelni obrasci za upravljanje podacima

- Kako bi se izbegli **uslovi trke**, od kritične je važnosti znati kada postoji mogućnost da **neki podatak deli i modifikuje više paralelnih niti**
- Neki od paralelnih obrazaca za upravljanje podacima unapređuju **lokalnost podataka**
- Paralelni obrasci za upravljanje podacima su:
 - **protočni sistem** (engl. *pipeline*)
 - **geometrijska dekompozicija/particija**
 - **pakovanje** (engl. *pack*)
 - **okupljanje** (engl. *gather*)
 - **rasejavanje** (engl. *scatter*)

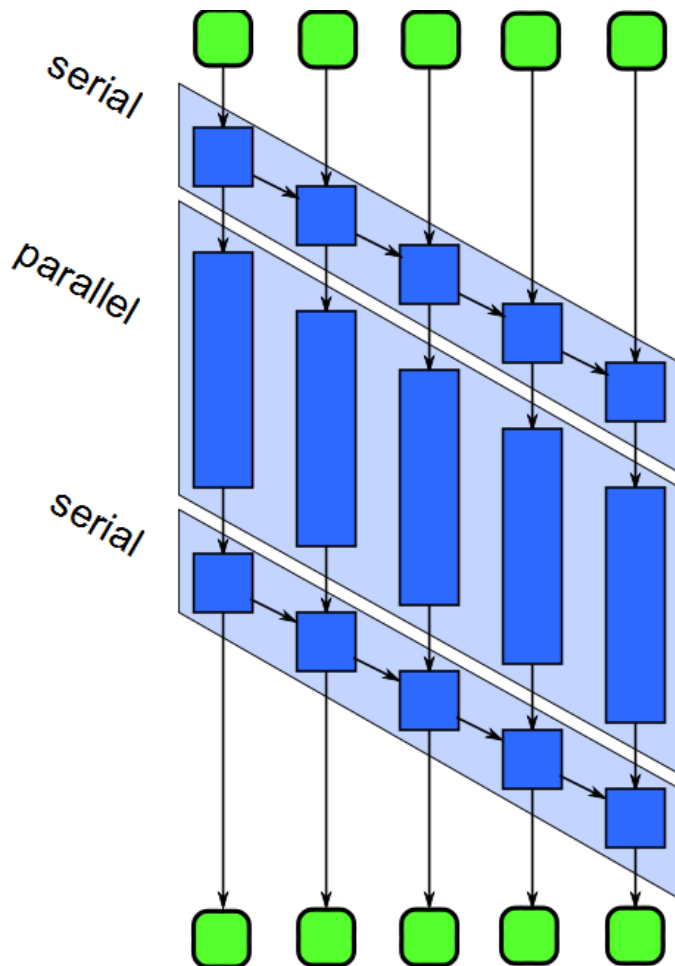
Protočni sistem



- **Protočni sistem** (engl. *pipeline*) koristi sekvencu stepena koji transformišu tok podataka
- Neki od stepena u sistemu mogu pamtit i **stanje** (engl. *stateful*)
- Podaci se mogu koristiti i proizvoditi inkrementalno, tj. „online”

Primeri protočnog sistema: obrada slika, kompresija i dekompresija podataka, obrada signala generalno

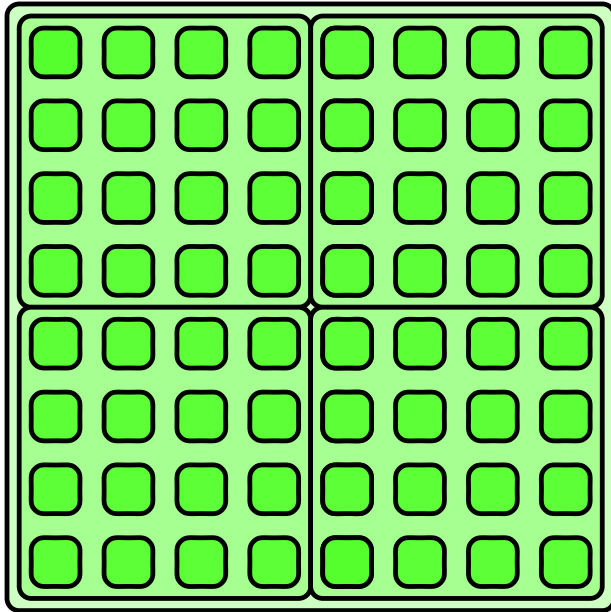
Protočni sistem



- **Paralelizacija protočnog sistema** se može realizovati:
 - **paralelnim izvršavanjem različitih stepena**
 - **paralelnim izvršavanjem više kopija stepena bez stanja** (engl. *stateless*)
- Paralelno izvršavanje više kopija stepena bez stanja zahteva **preuređenje ulaza**
- Mora se voditi računa o baferovanju podataka između stepena

Izvor: <http://parallelbook.com/>

Geometrijska dekompozicija/particija



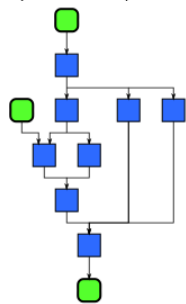
- **Geometrijska dekompozicija** razdvaja ulaznu kolekciju u pod-kolekcije
- **Particija** je specijalni slučaj kod koga nema preklapanja između pod-kolekcija
- Ne premešta podatke, samo daje alternativni pogled na njihovu organizaciju

Primeri geometrijske dekompozicije: JPEG i ostale makroblokovske kompresije, podeli-i-vladaj množenje matrica

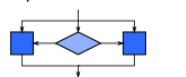
Paralelni obrasci – rezime

Paralelni obrasci

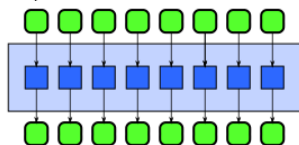
Superscalar sequence



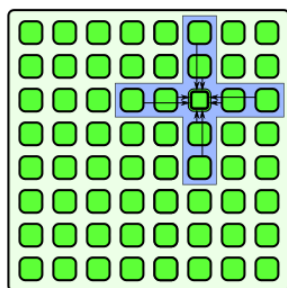
Speculative selection



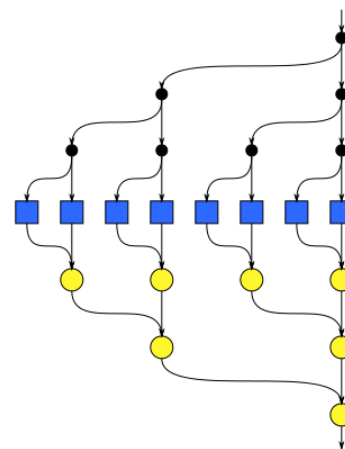
Map



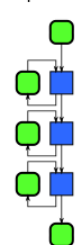
Stencil



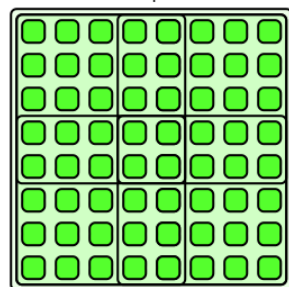
Fork-Join



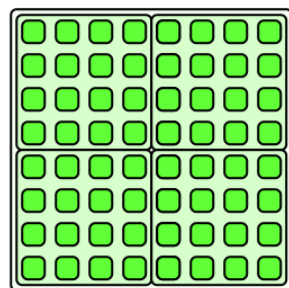
Pipeline



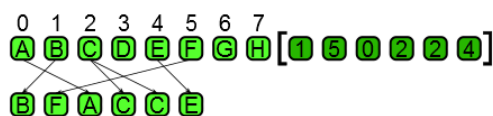
Geometric decomposition



Partition



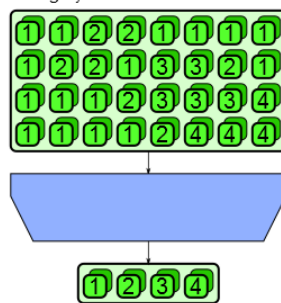
Gather



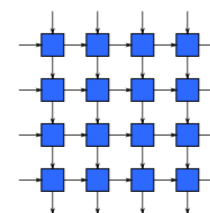
Scatter



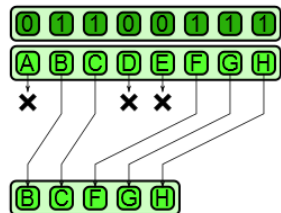
Category Reduction



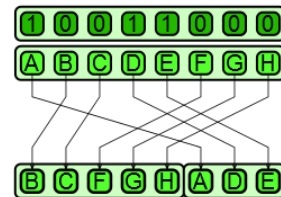
Recurrence



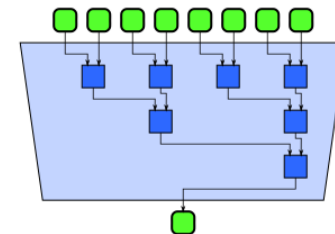
Pack



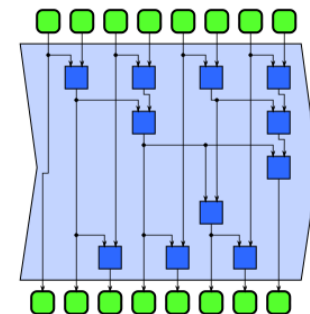
Split



Reduction



Scan



Expand

