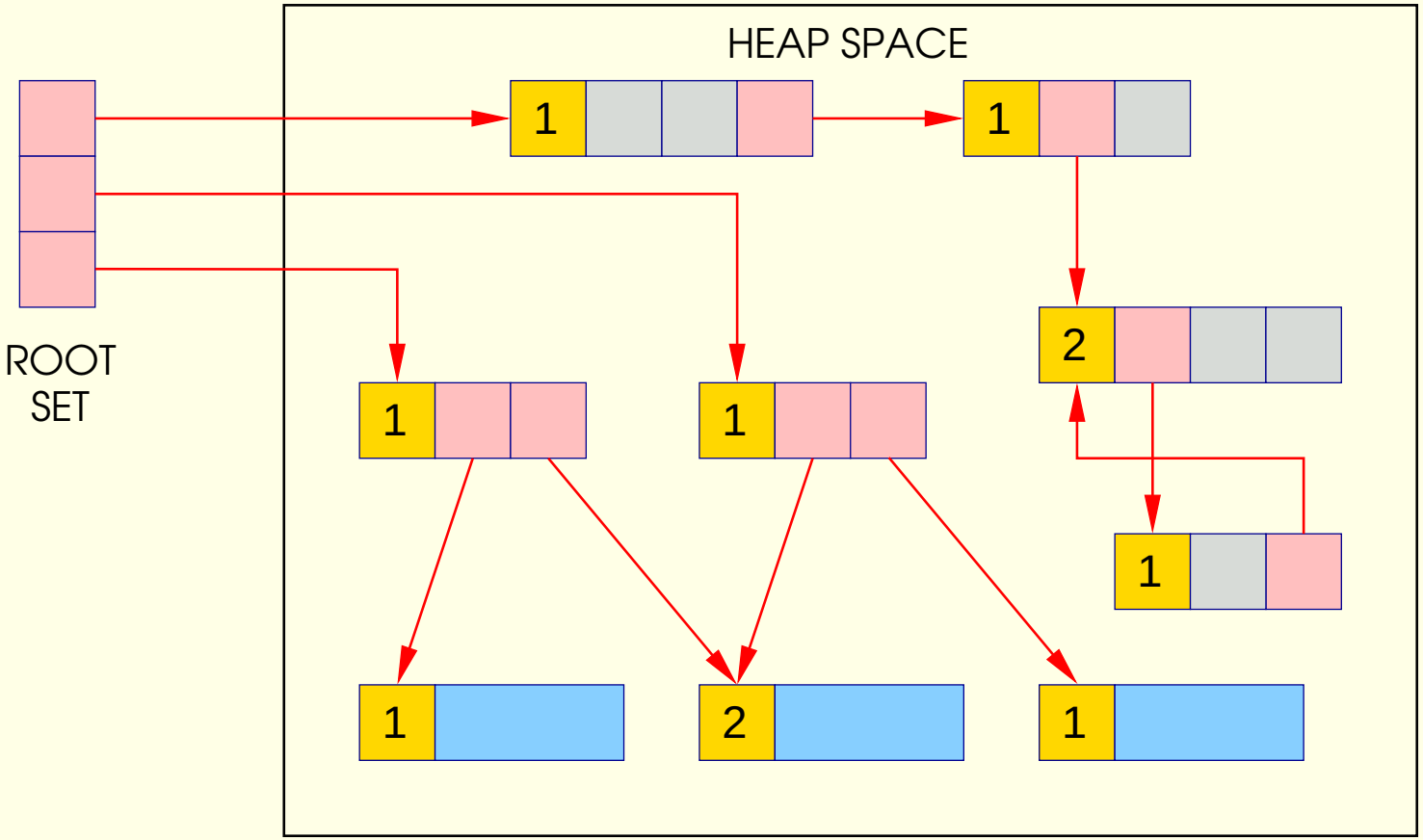
Prügikoristus

- ”Reference-counting” prügikoristuse korral on iga objektiga seotud loendur, mis näitab mitu viita antud objektile on.
- Loenduri muutmine toimub objektile viitade lisamisel/kustutamisel:
  - uue viida lisandumisel loendurit suurendatakse;
  - viida kustutamisel loendurit vähendatakse.
- Kui loendur on null, siis objekt vabastatakse:
  - vabastatud objekt lisatakse vabade objektide listi;
  - kõik viidad, kuhu see objekt viitas, kustutatakse.

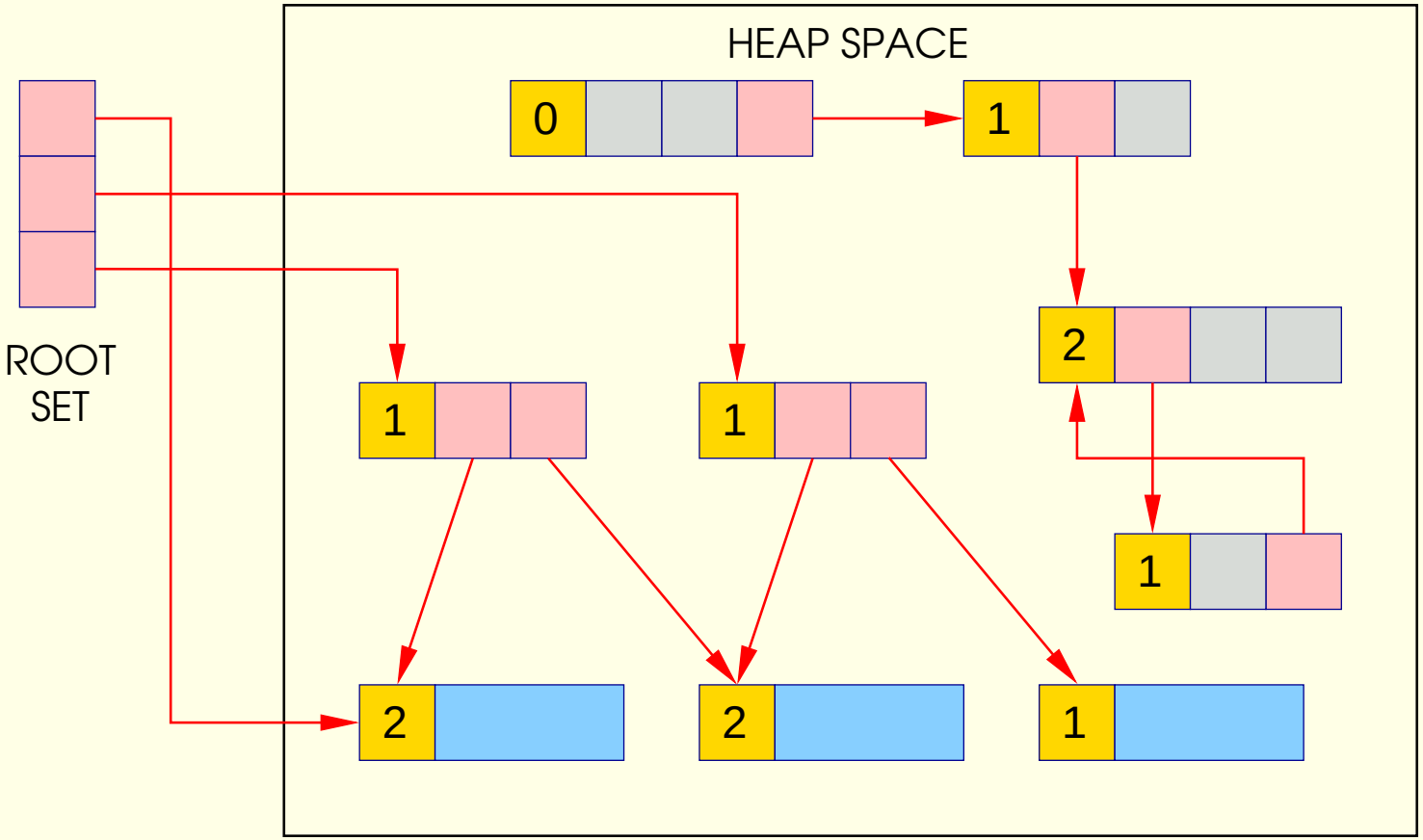
Prügikoristus

”Reference counting”:



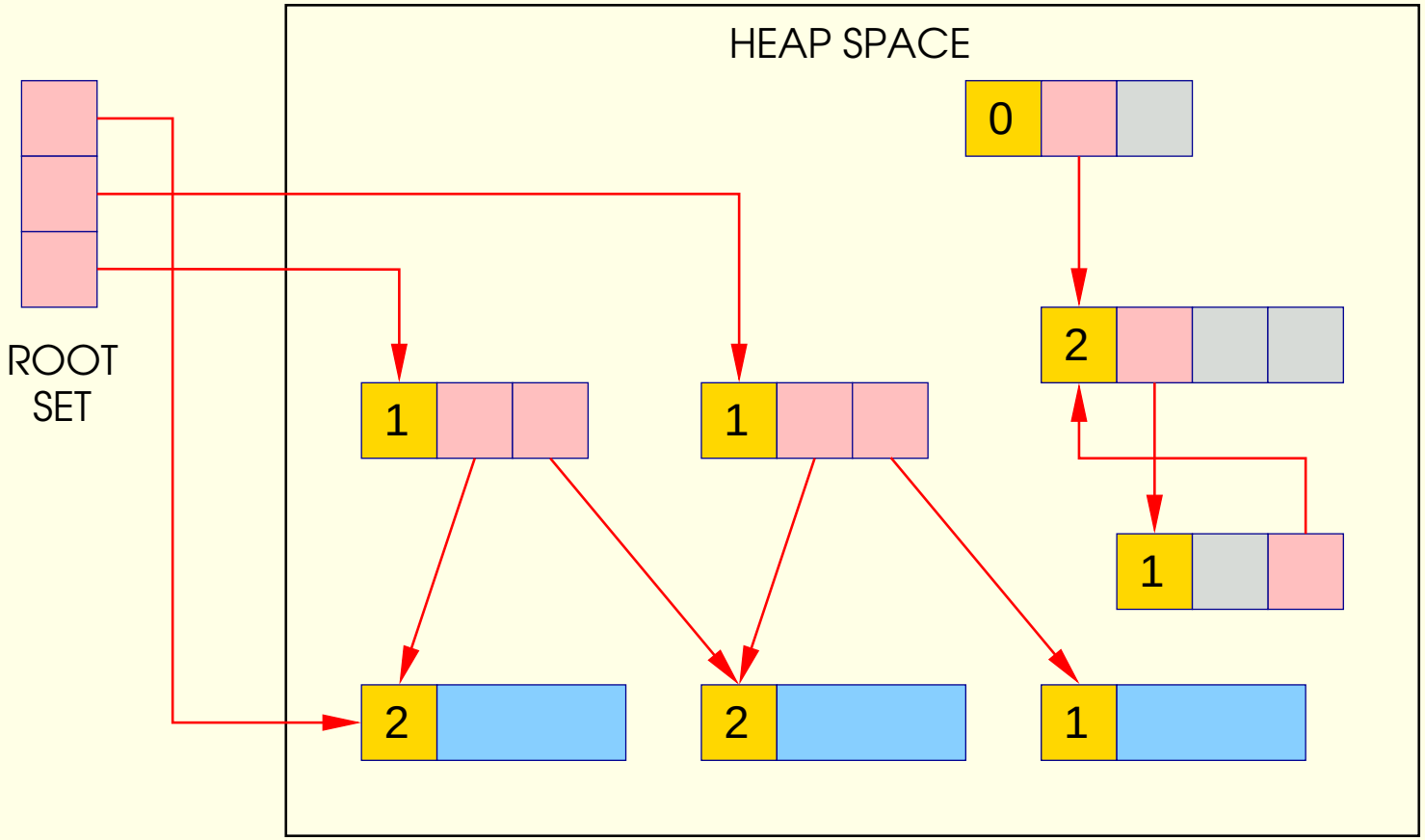
Prügikoristus

”Reference counting”:



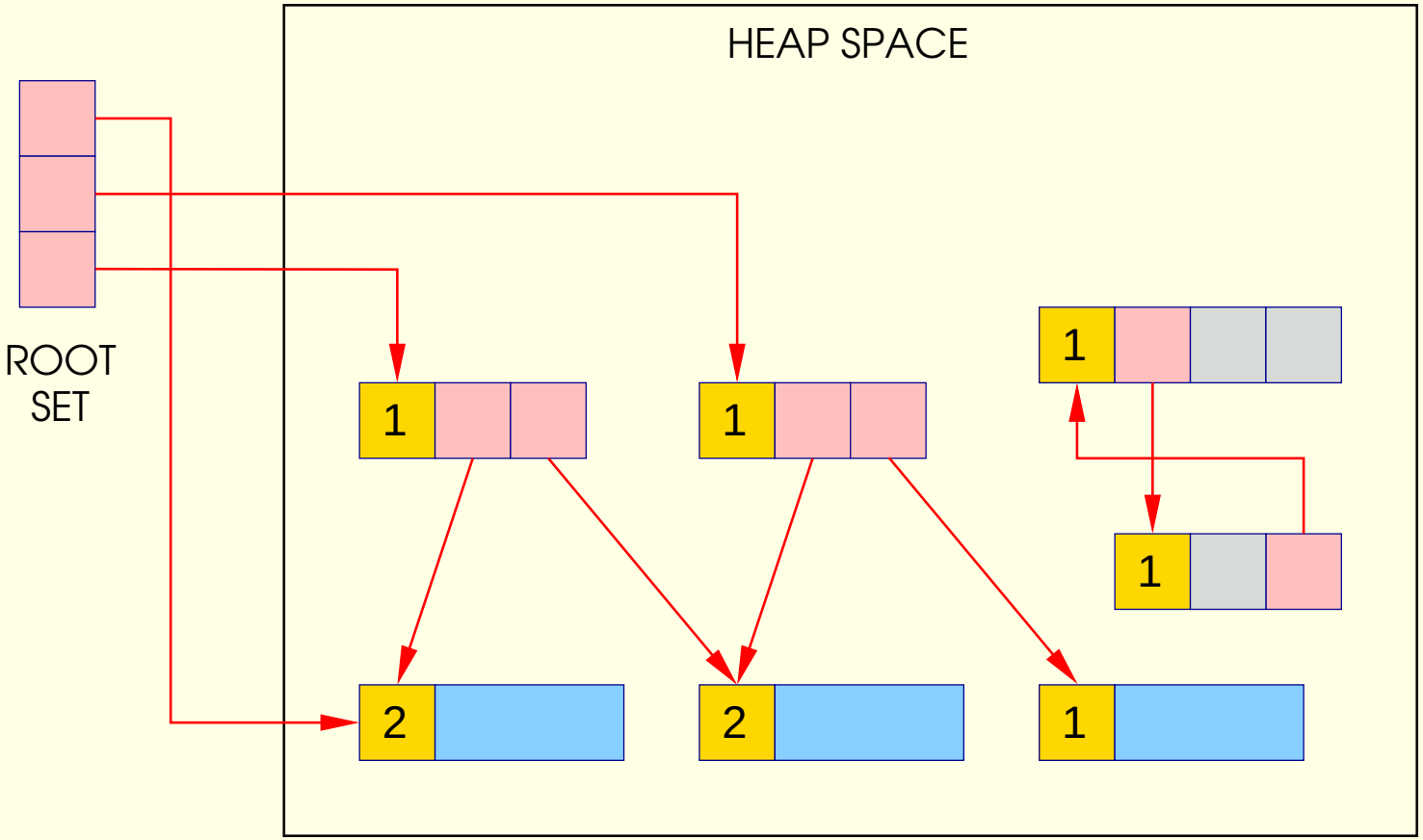
Prügikoristus

”Reference counting”:



Prügikoristus

”Reference counting”:



Prügikoristus

”Reference-counting” prügikoristuse eelised:

- on lihtne realiseerida;
- prügikoristusega seotud toimingud on hajutatud:
  - suhteliselt lihtsalt modifitseeritav inkrementaalseks;
- hea viitade lokaalsus:
  - muudetakse ainult lähte ja sihtviitade loendureid;
- aeg objekti muutumisel prügiks ning tema vabastamise vahel (zombie time) on minimaalne;
- võimaldab lihtsasti realiseerida objektide ”finaliseerimist”.

Prügikoristus

”Reference-counting” prügikoristuse puudused:

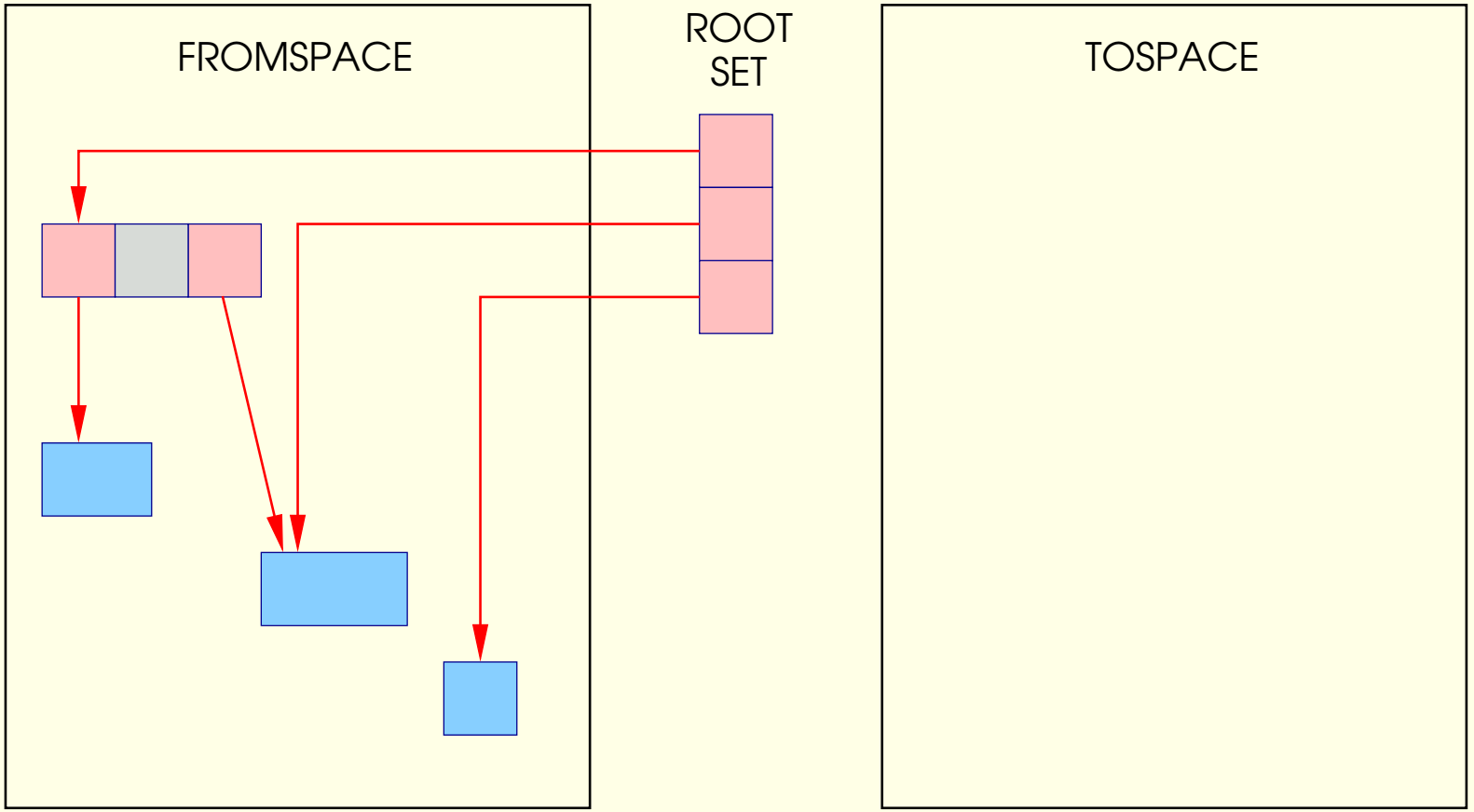
- suhteliselt ebaefektiivne:
  - peab haldama loendureid isegi, kui prügi ei koristata.
- mälu fragmenteerumine:
  - analoogne teiste vabade objektide listil baseeruvate skeemidega;
- paljude väikeste objektide korral võib loendurite peale kuluda suhteliselt palju mälu.
- rekursiivne vabastamine on halvimal juhul tõkestatud kuhja suurusega;
- ei suuda vabastada kogu prügi:
  - tsüklilised andmestruktuurid.

Prügikoristus

- ”Copying” prügikoristuse korral on kuhi jagatud kaheks võrdseks alampiirkonnaks: FromSpace ja ToSpace.
- FromSpace on aktiivselt kasutatav mälupiirkond, kuhu salvestatakse uued objektid.
- Kui FromSpace saab täis, siis teostatakse prügikoristus:
  - elusad objektid kopeeritakse FromSpace’ist ToSpace’i;
  - FromSpace ja ToSpace vahetavad rollid (so. endine ToSpace muutub FromSpace’iks ja vastupidi).

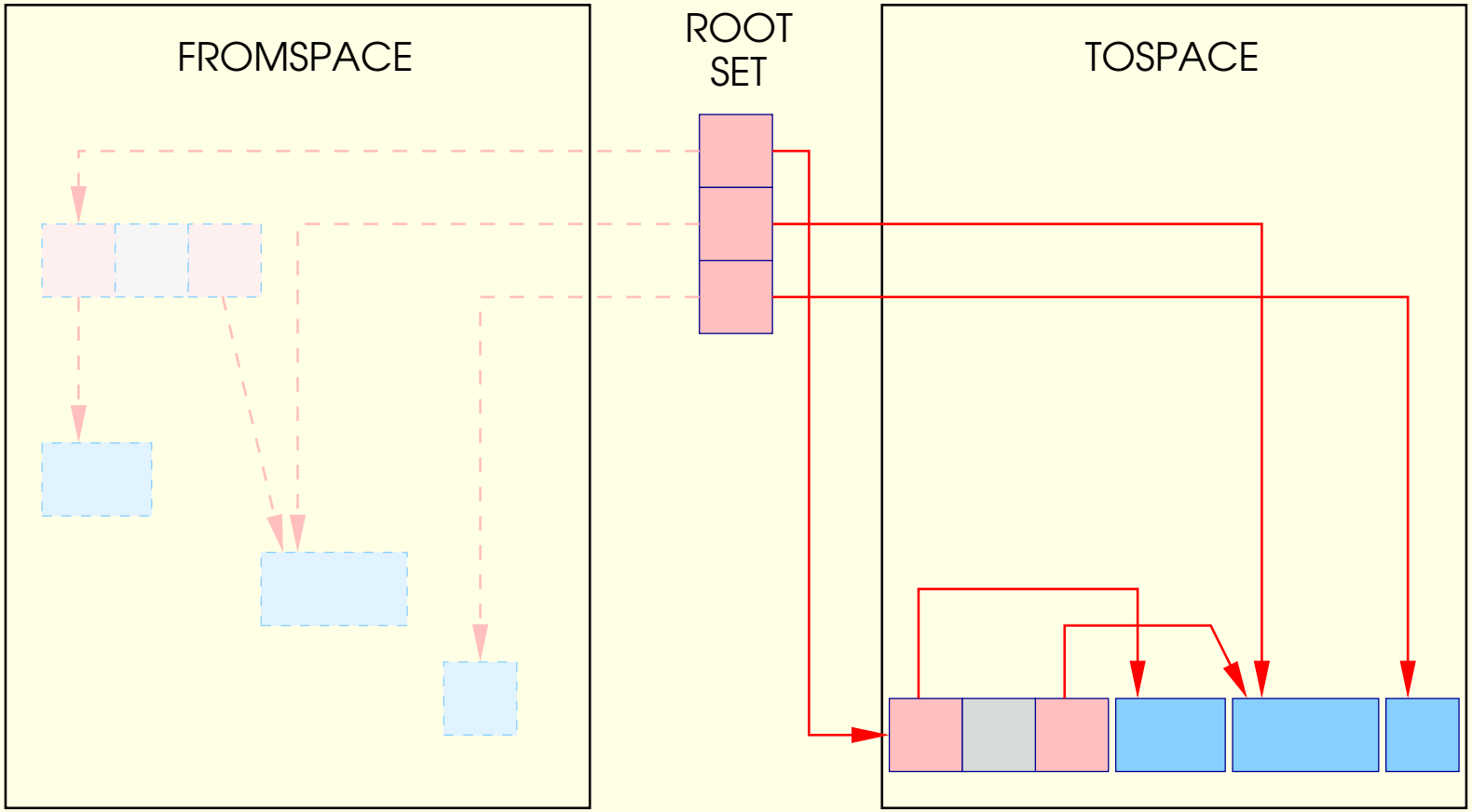
Prügikoristus

”Copying” prügikoristus:



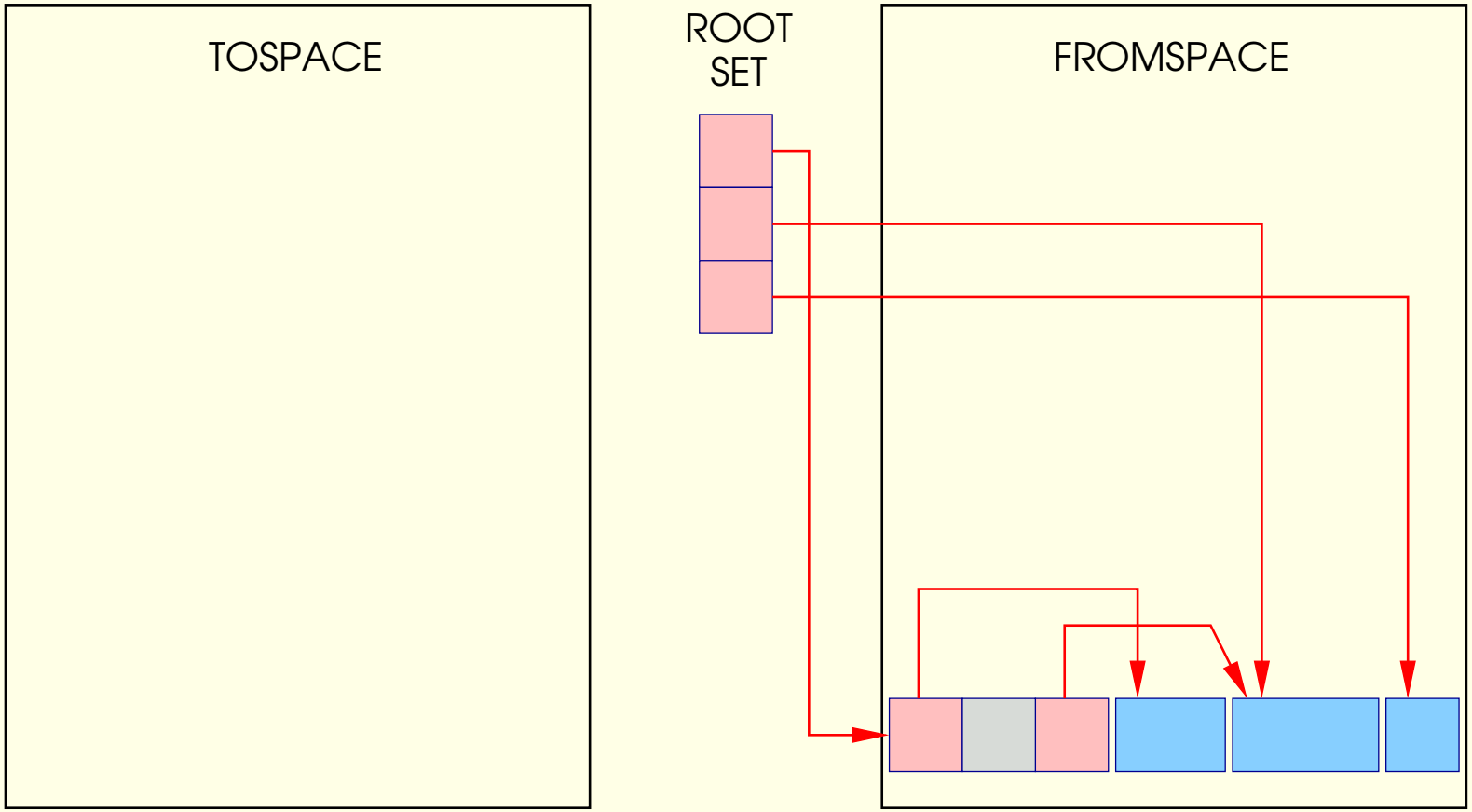
Prügikoristus

”Copying” prügikoristus:



Prügikoristus

”Copying” prügikoristus:

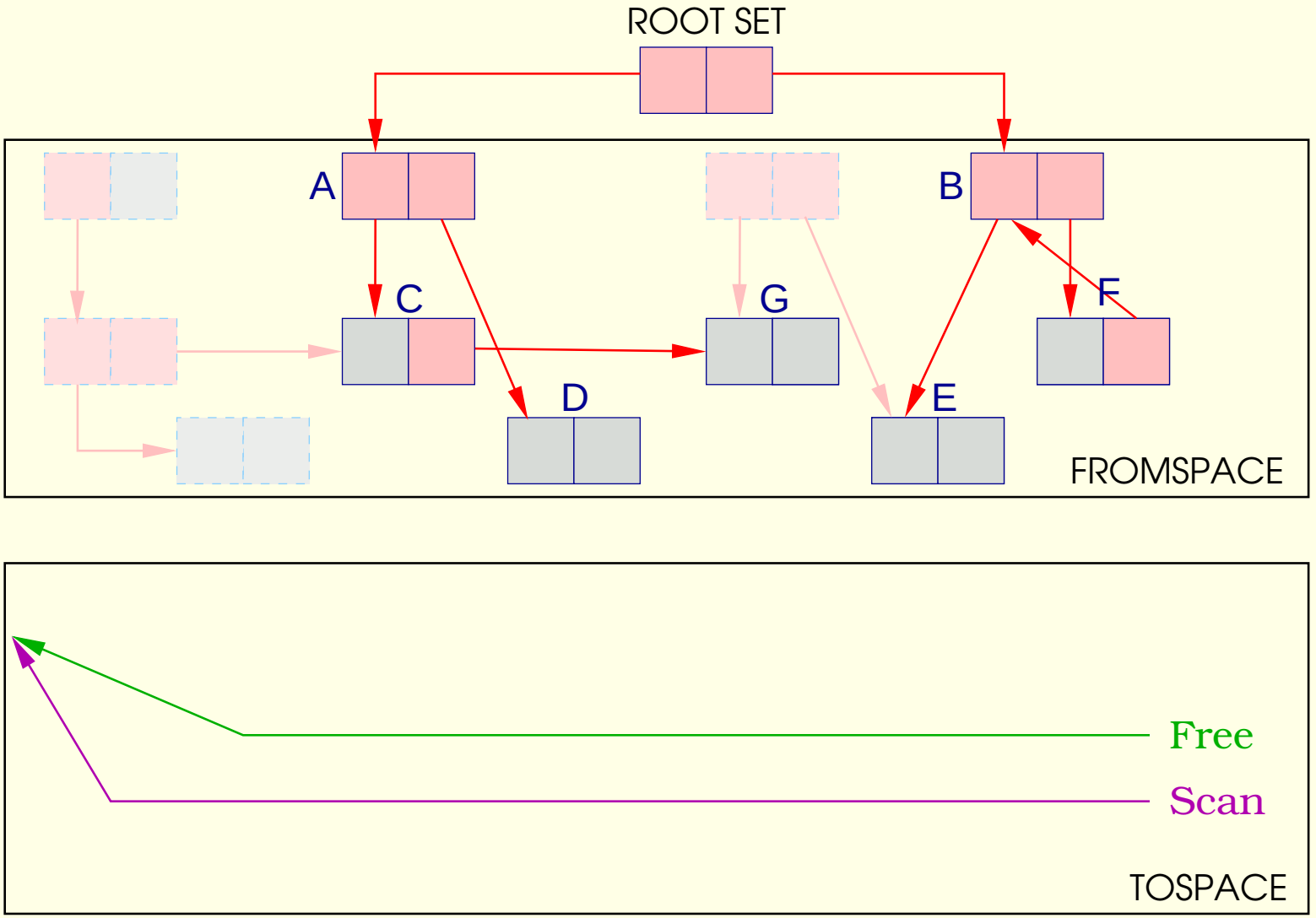


Prügikoristus

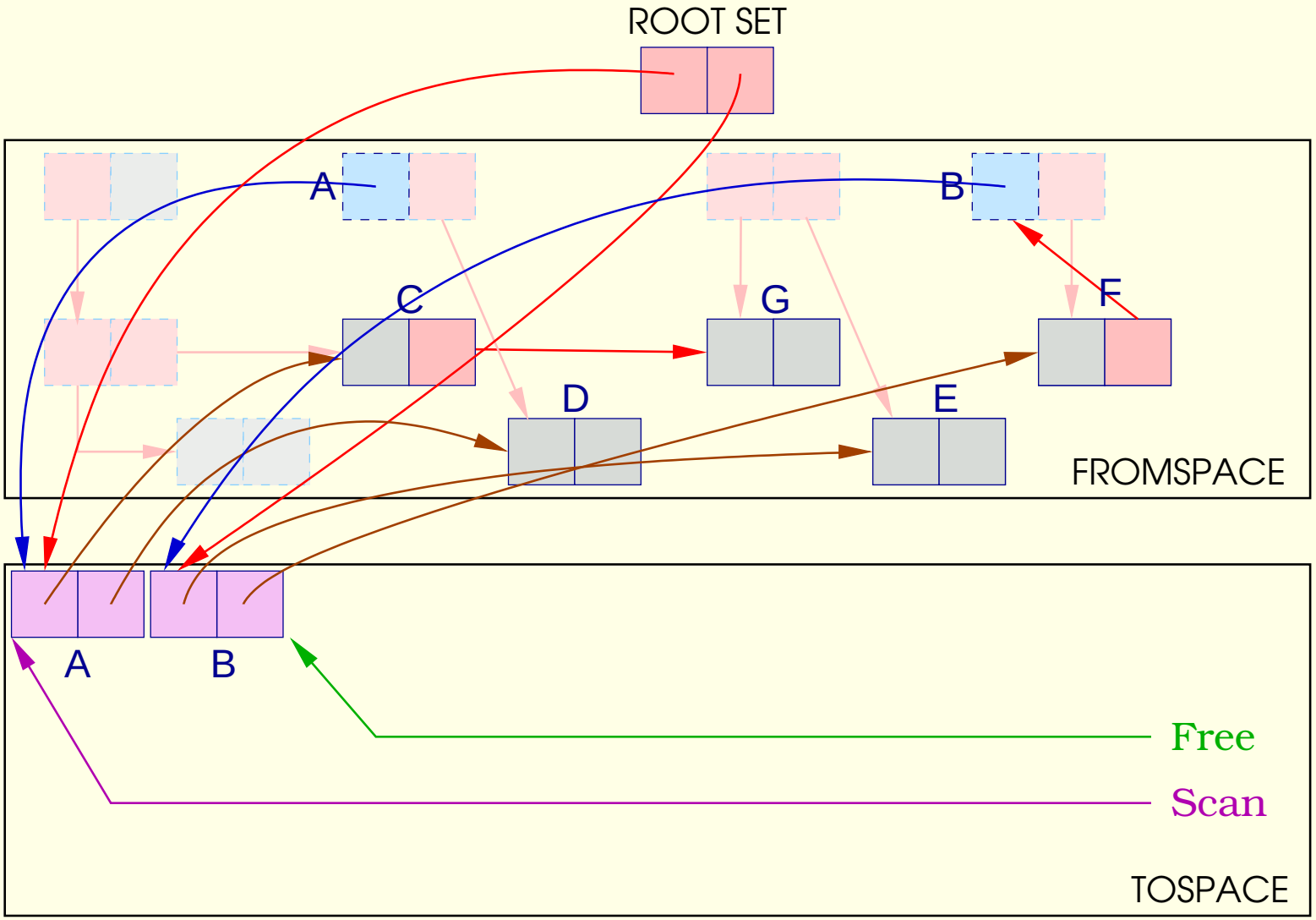
”Copying” prügikoristus, kasutades Cheney algoritmi, toimub kahes (üksteisega vahelduvas) faasis:

- esimese faasis (evacuate) kopeeritakse vahetult kättesaadavad objektid FromSpace’ist ToSpace’i, kasutatud viidad asendatakse viitadega vastavatele uutele objektidele ning vanade objektide asemele installeeritakse *”edasi toimetamise” viidad* (forwarding pointers);
- teises faasis (scavenge) skaneeritakse ToSpace’i kopeeritud objektid lineaarselt läbi ning kõik FromSpace’ist vahetult kättesaadavad objektid *evakueeritakse* ToSpace’i; kui evakueeritav objekt on juba varem kopeeritud, siis objekti uuesti ei kopeerita, vaid asendatatakse järgitav viit selle ”edasi toimetamise” viidaga;
- protsess lõpeb, kui skaneerimisviit jõuab järele kuhja täitumisviidale.

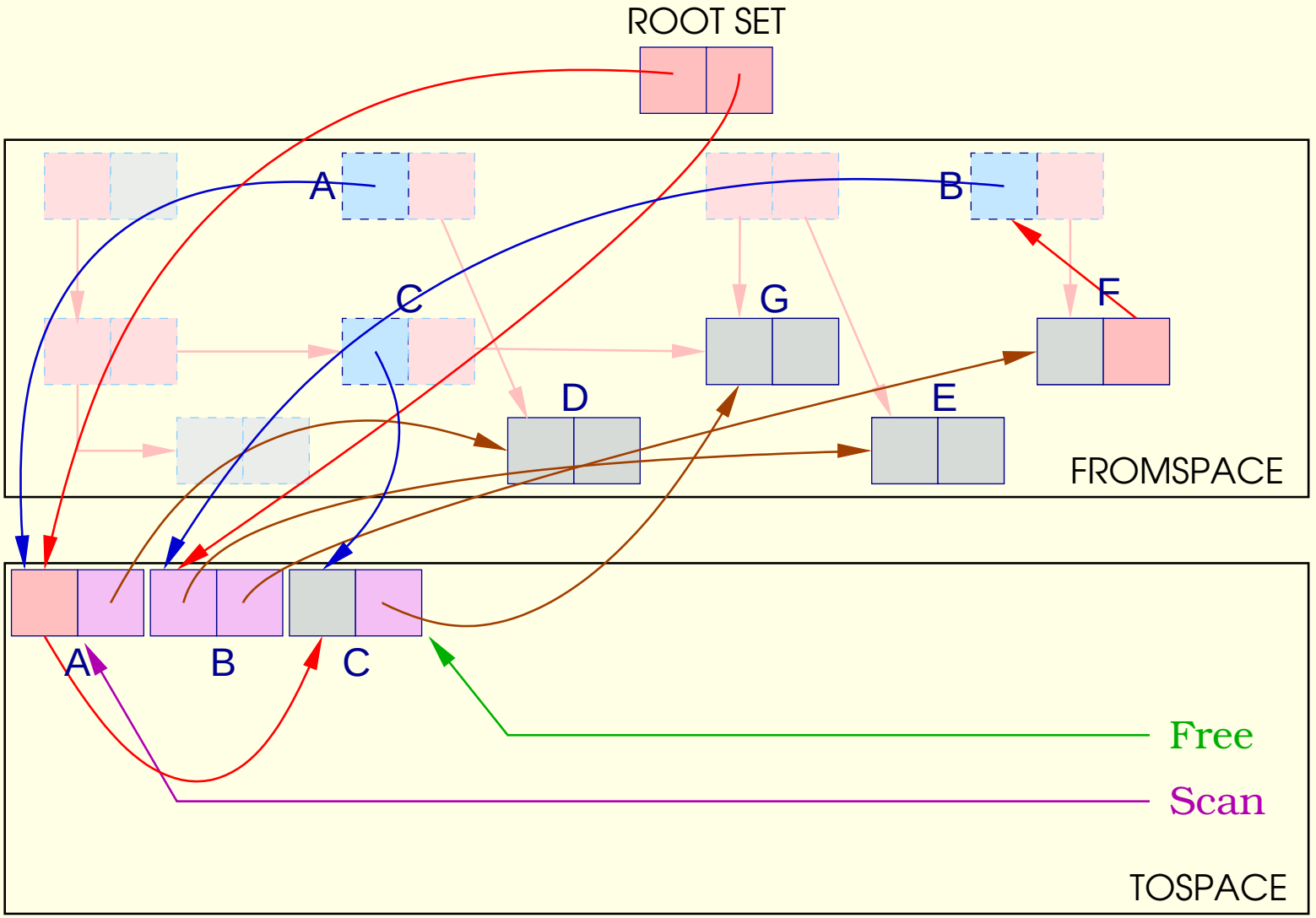
Prügikoristus



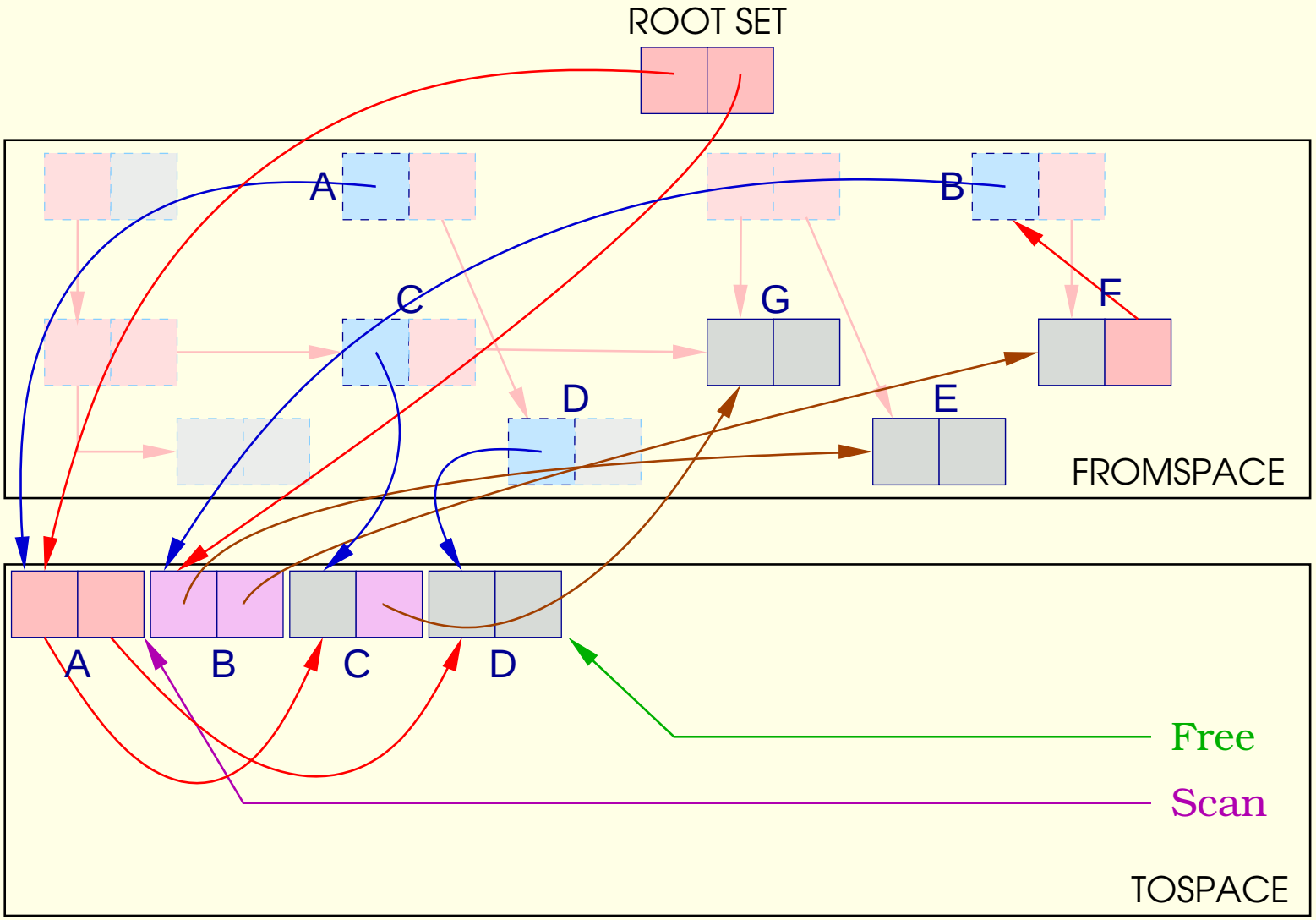
Prügikoristus



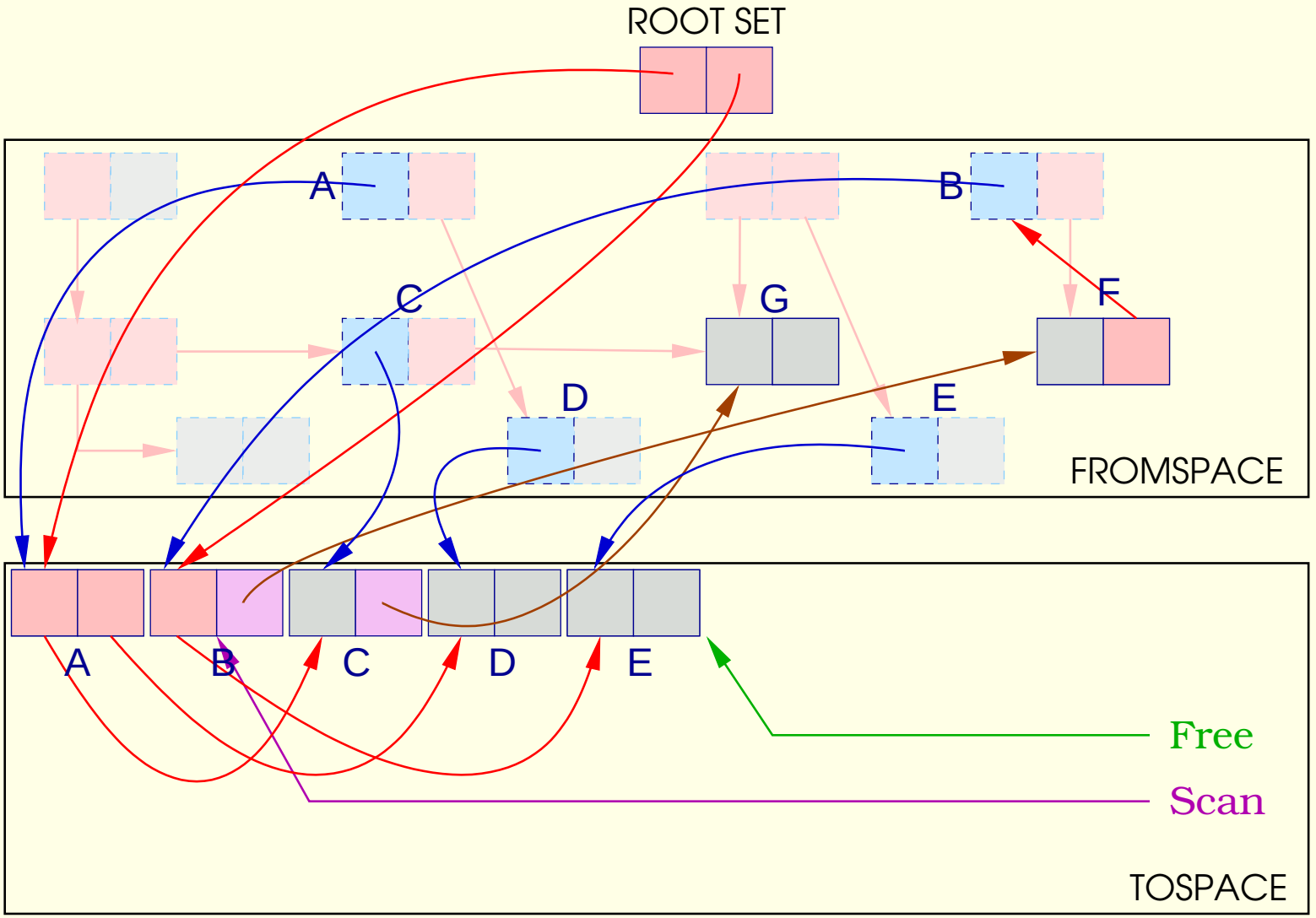
Prügikoristus



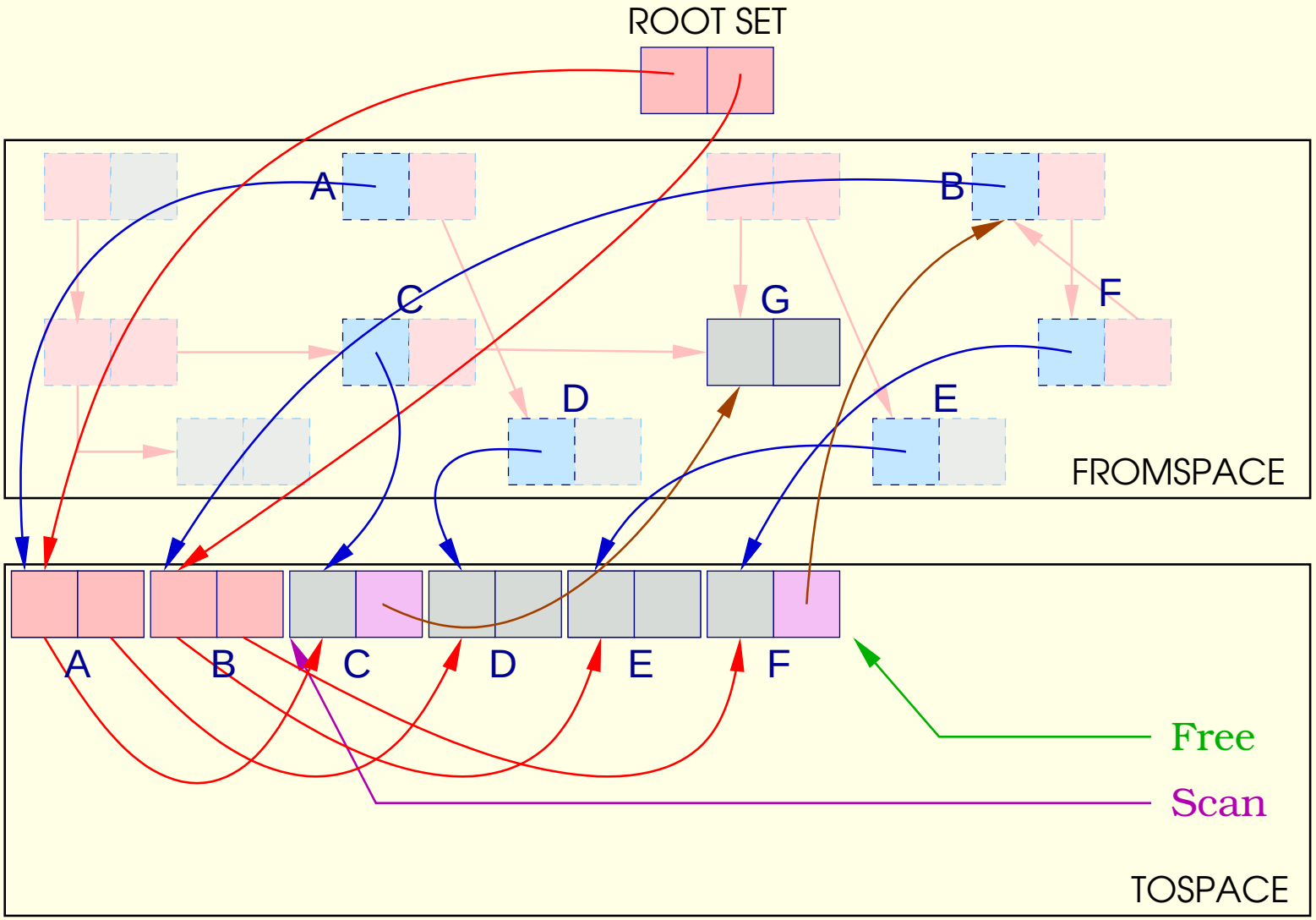
Prügikoristus



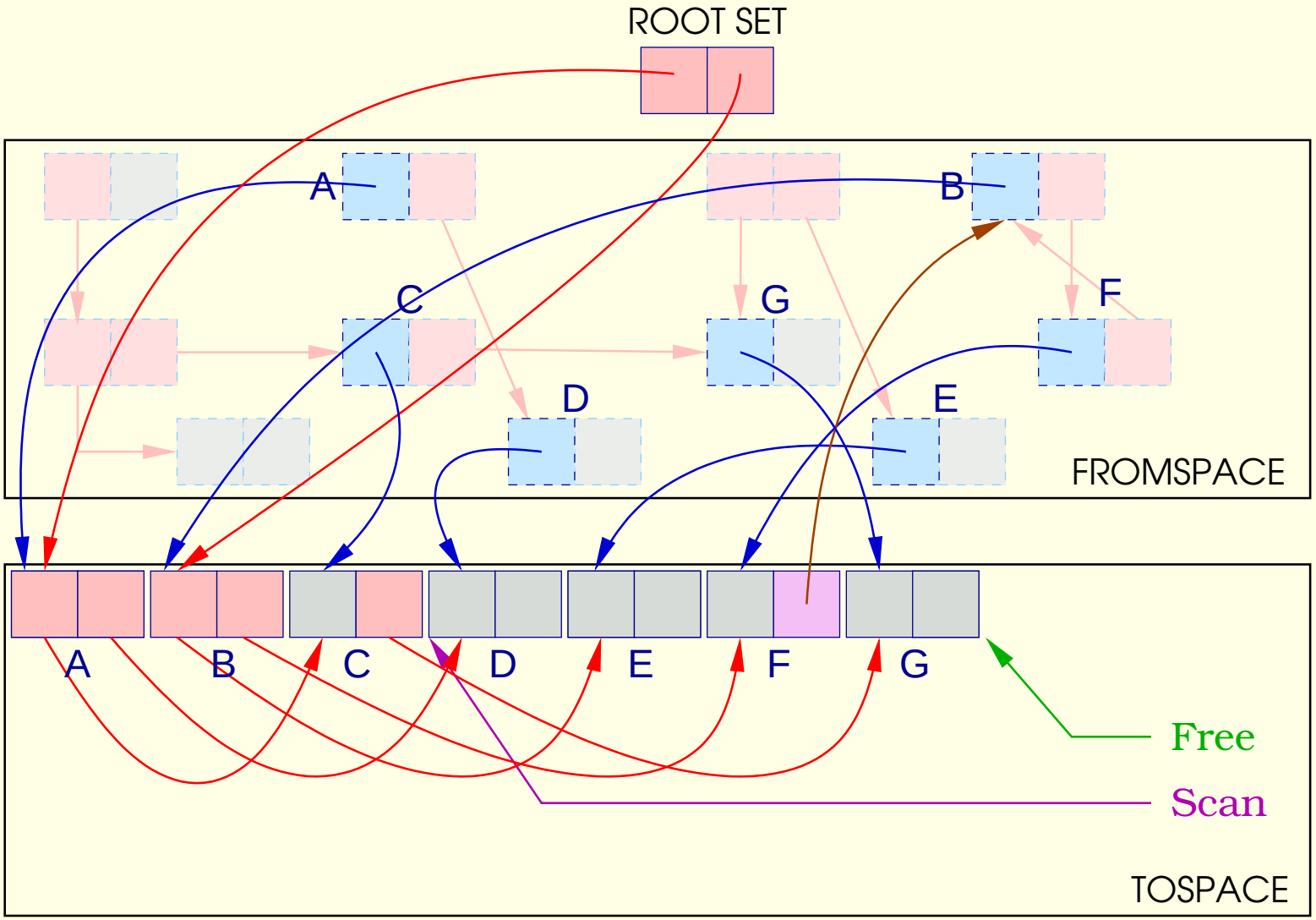
Prügikoristus



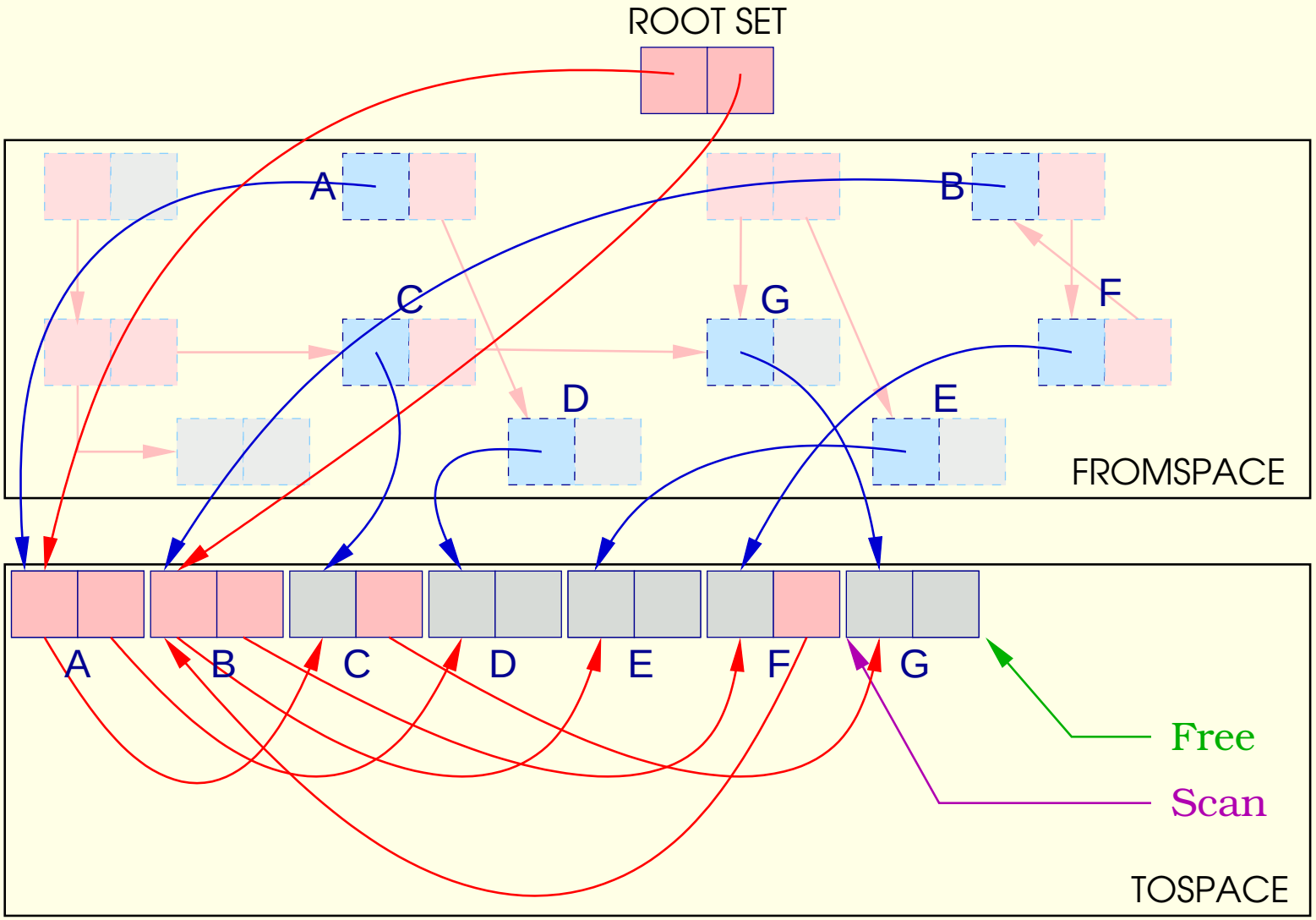
Prügikoristus



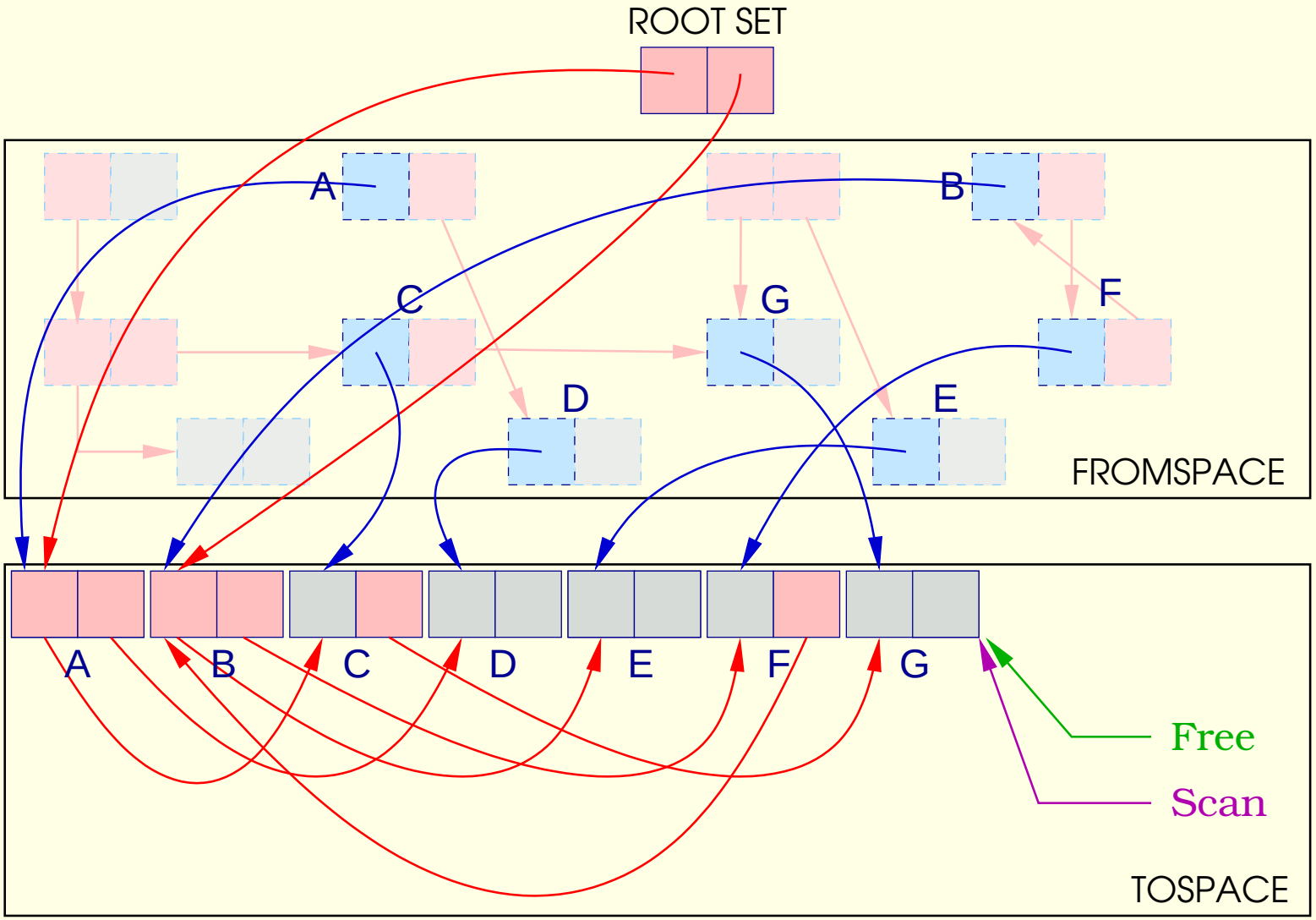
Prügikoristus



Prügikoristus



Prügikoristus



Prügikoristus

”Copying” prügikoristuse eelised:

- kogu vaba mälu on kompaktselt koos;
- suvalise suurusega uute objektide loomine on väga odav:
  - mälu reserveerimine on kuhja täitumisviida suurendamine;
  - kuhja täitumise kontroll on kahe viida võrdlemine;
- inspekteeritakse ainult elusaid objekte:
  - enamus objekte on reeglina suhteliselt lühikese elueaga;
  - elusaid objekte on seetõttu tavaliselt tunduvalt vähem kui prügi;
- teoreetiline amortiseeritud efektiivsus väga hea:
  - kuhja suuruse kasvades lähenevad kopeerimiskulud nullile!

Prügikoristus

”Copying” prügikoristuse puudused:

- kogu töö on kontsentreeritud prügikoristuse ajale:
  - võib tekitada häirivaid pause;
- laiuti läbivaatus võib segamini lüüa lokaalsusmustrid;
- kõik viidad tõstetakse ringi:
  - võib rikkuda mõnd invarianti, mida programm eeldab;
- pool mälust on ”kasutu”;
- pika elueaga objekte kopeeritakse igal prügikorstusel üha uuesti:
  - võib küllaltki kulukaks osutuda suurte pika elueaga objektide korral.