TÖL403G GREINING REIKNIRITA

23. Netareiknirit 3

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Sterktengdir hlutar (*strongly connected components*)
  - Skilgreining á sterkri tengingu
  - Hjálparsetning
  - Línuleg reiknirit
  - Reiknirit Kosaraju-Sharir

6.5 – 6.6

Sterk tenging (*strong connectivity*)

- Í stefnuneti eru hnútar *u* og *v* <u>sterktengdir</u> ef það er til vegur frá *u* til *v* **og** líka til vegur frá *v* til *u*

seiling(*u*)

reach(*u*)

Djúpleit á netið sem byrjar í *u*

Getum reiknað *drægi*(*u*) með *DFS*(*u*)

Mengi þeirra hnúta sem hægt er að komast til frá *u*

Ef $v \in$ *drægi*(*u*) og $u \in$ *drægi*(*v*) þá eru *u* og *v* sterktengdir

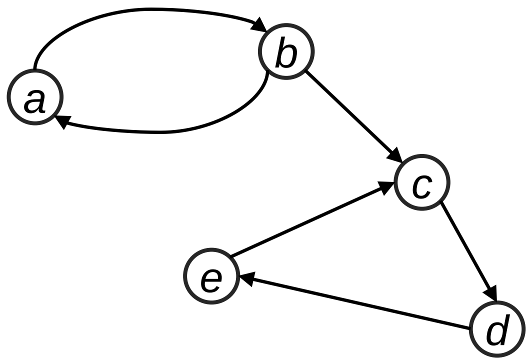
Sterktengdur hluti (*strongly connected component*)

- <u>Sterktengdur hluti</u> (*SCC*) nets er *óstækkanlegt* (*maximal*) mengi hnúta þannig að allir hnútar í menginu eru sterktengdir

Þýðir að ef við bætum hnúti í mengið þá brýtur það skilyrðið um að allir hnútarnir séu sterktengdir

Þetta er ekki endilega stærsta mengið

Dæmi:



Hnútamengið {*a*, *b*} er sterktengdur hluti

Hnútamengið {*c*, *d*, *e*} er sterktengdur hluti

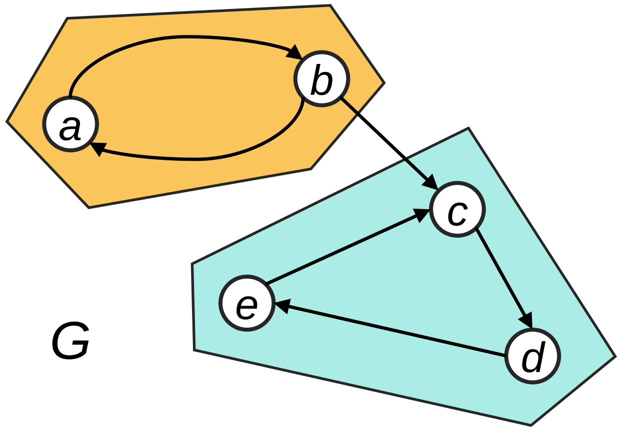
Hvort um sig er óstækkanlegt, þ.e. ekki hægt að bæta í það hnúti og viðhalda sterktengni

Sterktengdir hlutar skilgreina <u>jafngildisvensl</u> (*equivalence relation*)

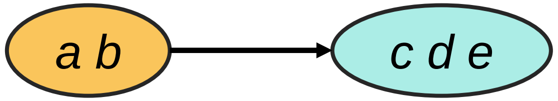
Sterktengnisnet (*strong component graph*)

- Getum búið til net úr sterktengdum hlutum nets *G*, kallast *scc*(*G*)
  - Fæst með því að láta alla sterktengda hluta *G* skreppa saman í einn hnút
  - Stikurnar fást með því að skoða hvaða stikur liggja á milli sterktengdra hluta í *G*

Einfalt dæmi:

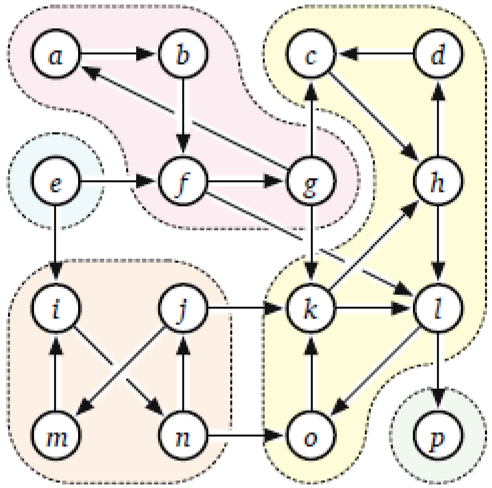


G

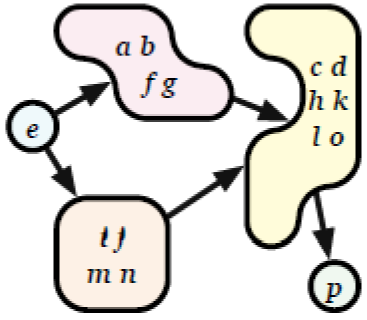


scc(*G*)

Stærra dæmi:



G



scc(*G*)

Hvaða eiginleika hefur *scc*(*G*)?

Útreikningur á sterktengdum hlutum nets

- Auðvelt að finna sterktengdan hluta eins hnútar v á $O(V + E)$ tíma

Finnum fyrst *drægi*(v) með djúpleit

Allir hnútar sem sem geta komist til v

Finnum svo $drægi^{-1}(v) = \{u \mid v \in drægi(u)\}$

Gerum þetta með því að finna *rev*(G), þ.e. net sem er eins og G, nema búið að snúa öllum örvum við

Þá er sterktengdi hluti v sniðmengið $drægi(v) \cap drægi^{-1}(v)$

Getum þá ákvarðað hvort allt netið sé sterktengt á $O(V + E)$ tíma

Þurfum bara að athuga hvort s.t. hluti hnútarins innihaldi alla hnútar netsins

En til að finna alla sterktengda hluta nets þá þurfum við (möguleg) gera þetta fyrir alla hnúta netsins

Tími: $O(V \cdot E)$

Línuleg reiknirit fyrir sterktengda hluta

- Það eru til nokkur $O(V + E)$ tíma reiknirit fyrir sterktengda hluta
- Þau nota flest djúpleit

Eitt fyrsta reikniritið er eftir Robert Tarjan, frá 1972

Það fer eina umferð um netið og hefur lágan fasta

Nokkuð flókið reiknirit og dálítið erfitt að skilja virkni þess

Annað reiknirit er eftir Kosaraju og Sharir

Fann sömu aðferð árið 1981

Lýsti aðferðinni í fyrst árið 1978

Þetta reiknirit notar tvær umferðir um netið

En er auðveldara að lýsa og skilja

Við munum aðeins skoða Kosaraju-Sharir reikniritið

Hjálparsetning

- Lát *T* vera djúpleitartré fyrir stefnunet *G*. Sérhver sterktengdur hluti *C* í *G* hefur <u>nákvæmlega einn</u> hnút sem ekki hefur foreldri (*parent*) í *C*.

Hnúturinn hefur annaðhvort foreldri í öðrum s.t. hluta eða hefur ekkert foreldri í *T* (þ.e. er rótin)

- Sönnun:

Lát *C* vera sterktengdan hluta í *G*

Fyrir sérhverja tvo hnúta *v* og *w* í *C* þá er vegur á milli þeirra sem er algerlega innan *C*

Lát *v* vera þann hnút í *C* sem hefur lægsta *for*-gildi (*v.pre*)

þ.e. *parent*(*v*).*for* < *v*.*for*

Ef *v* hefur foreldri (þ.e. er ekki rótin) þá hefur *parent*(*v*) lægra *for*-gildi

svo *parent*(*v*) er þá ekki í *C*

svo <u>a.m.k. einn</u> hnútur hefur foreldri utan *C*

því *v* er með lægsta *for*-gildið

Rétt áður en kallað er á *DFS*(*v*) þá eru allir hnútarnir í *C* nýjir

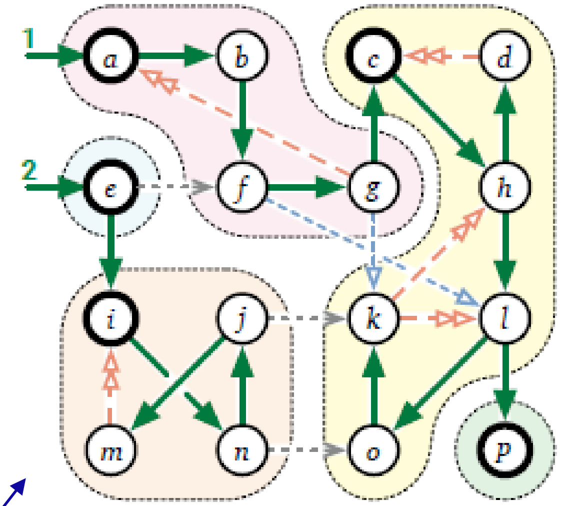
Fyrir alla aðra hnúta *w* í *C* gildir að það er vegur innan *C* frá *v* til w,   <u>svo *w* hefur foreldri í *C*</u>

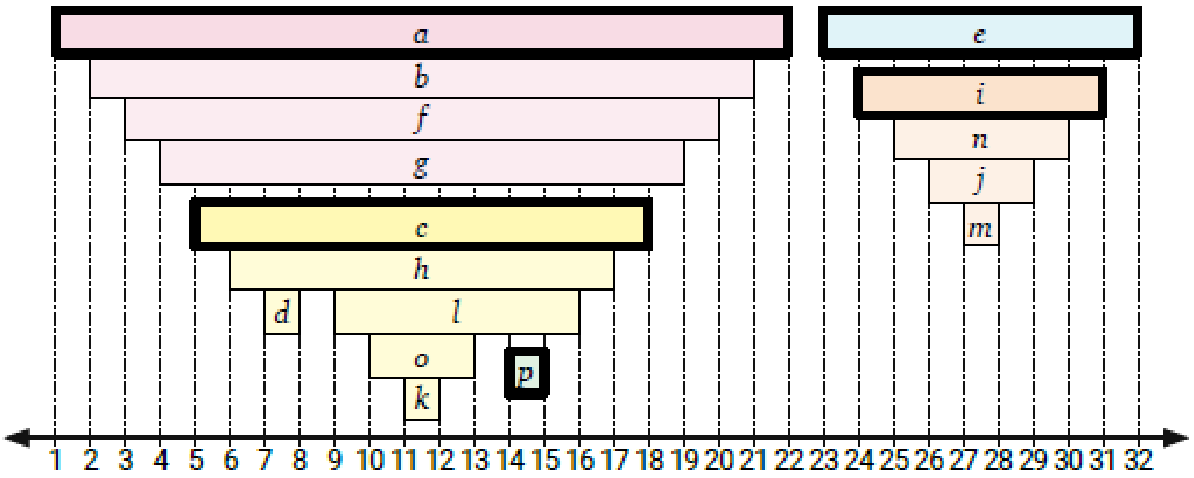
svo <u>enginn annar</u> hnútur hefur foreldri utan *C*

Þýðing hjálparsetningar

- Hver sterktengdur hluti stefnunets *G* skilgreinir <u>samhangandi hluttré</u> í sérhverjum djúpleitarskóg fyrir *G*

<u>Dæmi</u>:



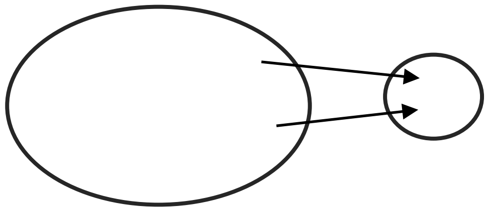


Takið eftir að hver s.t. hluti hefur samhangandi hluttré

Hnútarnir með lægsta *for*-gildið í hverjum s.t. hluta eru með feitari hring

Hugmynd að reikniriti (Kosaraju-Sharir)

svelg-hnútur í *scc*(*G*)



Byrja í svelg-hnúti í *scc*(*G*)

Lát *v* vera hnút þar.

Finna alla hnútana í *drægi*(*v*) og merkja þá með sama númeri

Þeir eru allir í sama s.t. hluta

Henda öllum þessum hnútum út úr *G*

Finna svo svelg-hnút í afgangnum, o.s.frv.

Ekki beint reiknirit vegna

```
STRONGCOMPONENTS(G):
  count ← 0
  while G is non-empty
      C ← ∅
      count ← count + 1
      v ← any vertex in a sink component of G    ⟨⟨Magic!⟩⟩
      for all vertices w in reach(v)
          w.label ← count
          add w to C
      remove C and its incoming edges from G
```

<u>Vandamál</u>:

Ekki auðvelt að finna hnút í svelg-hluta *scc*(*G*)

Djúpleit í hnúti sem er ekki í svelg-hluta gæti farið
yfir í næsta s.t. hluta án þess við vitum af því!

Lindar-hlutar í *scc*(*G*)

(*source-component*)

Það er reyndar auðvelt að finna hnút í *G* sem er í lindar-hluta í *scc*(*G*)

Setning 6.3:

Síðasti hnúturinn í eftirröðun á *G* er í lindar-hluta í *scc*(*G*)

Rökstuðningur: Ef *v* er síðasti hnúturinn í eftirröðun á *G*, þá hlýtur *DFS*(*v*) að vera síðasta *DFS*-kallið í fallinu *DFSAll*

Annars væri búið að merkja v nú þegar

Það getur því engin hnútur úr öðrum s.t. þætti verið með ör yfir *v*

Þessi s.t. hluti hlýtur þá að vera lindar-hluti í *scc*(*G*)

Auðvelt að sjá að *rev*(*scc*(*G*)) = *scc*(*rev*(*G*))

En athugið að lindar-hluti í *scc*(*rev*(*G*)) er svelg-hluti í *scc*(*G*)!

Erfitt að finna hnúta í svelg-hluta *scc*(*G*), svo við snúum bara öllum örvunum við og finnum hnút í lindar-hluta í *scc*(*rev*(*G*))!

Reiknirit Kosaraju-Sharir

- Notum tvær djúpleitir:

  - Djúpleit á *rev*(*G*) og setjum kláraða hnúta á hlaða

Þá eru hnútar í svelg-hluta *scc*(*G*) efst á hlaðanum

þ.e. lindar-hluta *scc*(*rev*(*G*))

  - Djúpleit á *G* í <u>röð eftir hlaðanum</u>

Byrjum þá í svelg-hluta og komumst ekki út úr honum!

Þegar við komum upp úr *DFS* þá er næsti hnútur á hlaðanum í svelg-hluta af restinni af netinu, o.s.frv.

Semsagt:

Fyrri hlutinn raðar hnútunum rétt

svo að seinni hlutinn geti tínt s.t. hlutana hvern af öðrum "aftan af" netinu

Miðað við grannfræðiröð í *scc*(*G*)

Reiknirit Kosaraju-Sharir

```
KosarajuSharir(G):
    S ← new empty stack
    for all vertices v
        unmark v
        v.root ← None
    ⟨⟨Phase 1: Push in postorder in rev(G)⟩⟩
    for all vertices v
        if v is unmarked
            PushPostRevDFS(v, S)
    ⟨⟨Phase 2: DFS again in stack order⟩⟩
    while S is non-empty
        v ← Pop(S)
        if v.root = None
            LabelOneDFS(v, v)
```

Ytra fall fyrir báðar djúpleitir

```
PushPostRevDFS(v, S):
    mark v
    for each edge u→v    ⟨⟨Reversed!⟩⟩
        if u is unmarked
            PushPostRevDFS(u, S)
    Push(v, S)
```

Fyrri djúpleit, sem vinnur á öfugu neti og setur hnútana á hlaða í eftirröðun

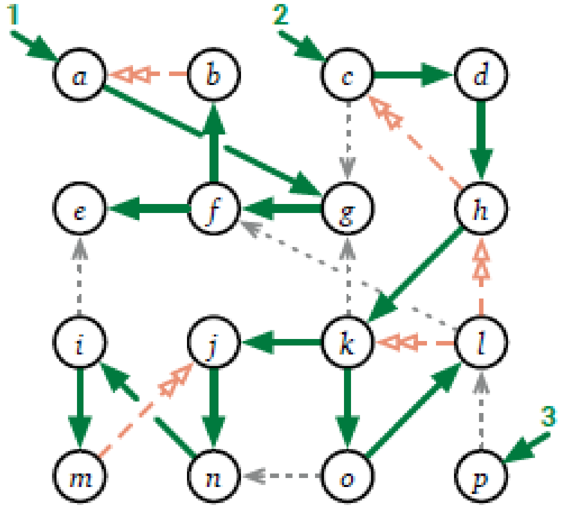
```
LabelOneDFS(v, r):
    v.root ← r
    for each edge v→w
        if w.root = None
            LabelOneDFS(w, r)
```

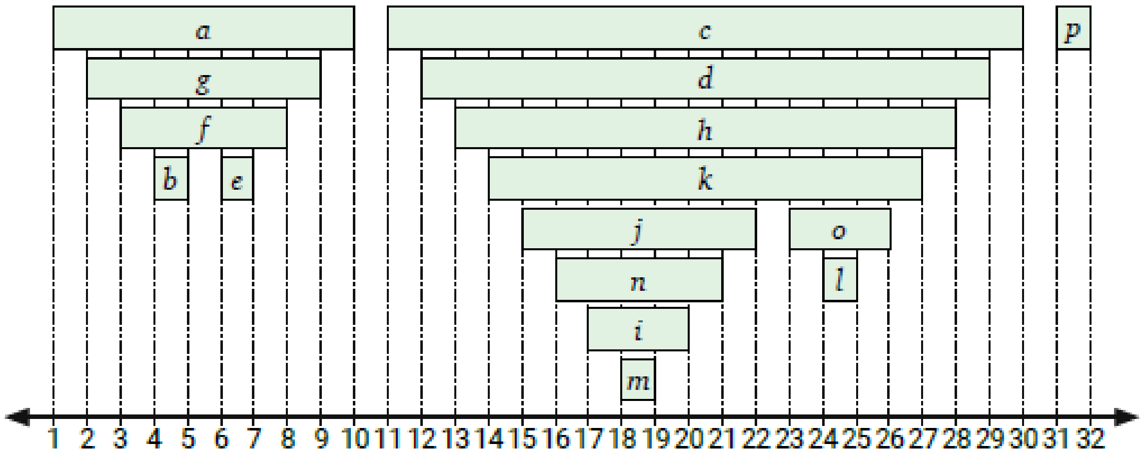
Seinni djúpleit, sem vinnur á G og notar hnútaröðina á hlaðanum

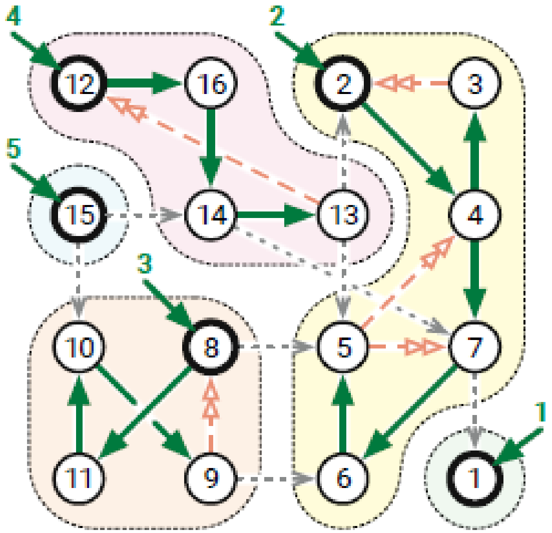
Þá verður síðasti hnúturinn í eftirröðun á G efst á hlaðanum í lokin

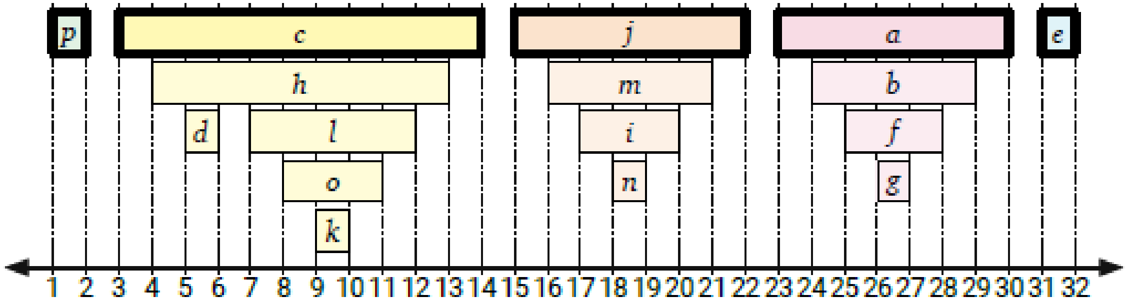
Tökum hnúta af hlaðanum og förum þangað ef þeir hafa ekki þegar fengið s.t. hluta

Sýnidæmi um Kosaraju-Sharir



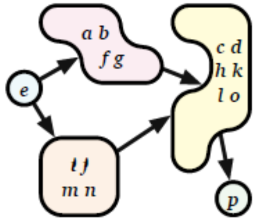






Hlaði:

p	1
c	
d	
h	
k	
o	
l	
j	
n	
i	
m	
a	
g	
f	
e	
b	16



Notkunardæmi

- Þverslaufuskipulag Vefsins

  Þegar Vefurinn hefur verið skannaður og kortlagður hefur komið í ljós áhugavert skipulag

  Hver vefsíða er hnútur og ör frá einni síðu til annarar ef hún hefur hlekk þangað

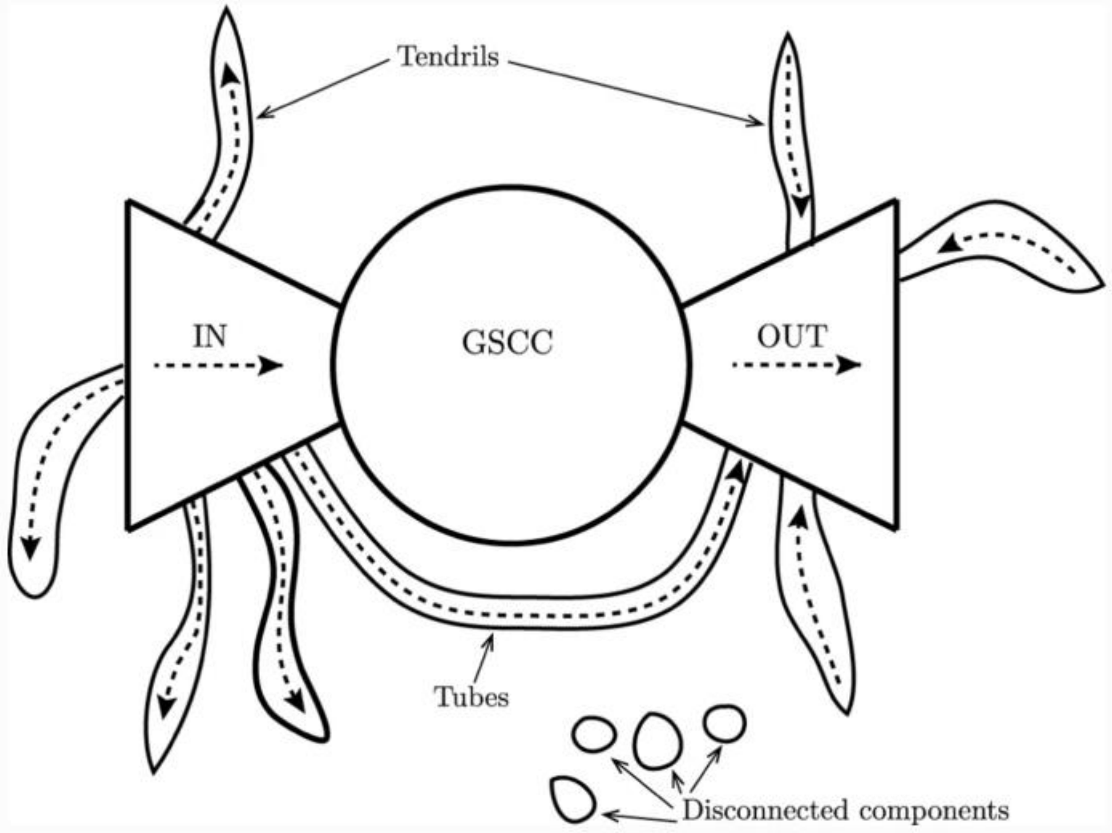
  Það er einn stór sterktengdur hluti: GSCC (~30%)

  Vefsíður sem hafa hlekki inn í GSCC (~15%)

  Vefsíður sem hafa hlekki út úr GSCC (~25)

  Angar af vefsíðum sem liggja út úr IN eða inn í OUT (~30%)

  Restin af síðunum er ekki tengd við aðalhlutann



Ekki víst að þetta skipulag gildi lengur með uppgangi samfélagsmiðla

Notkunardæmi

HÍ

- 2-SAT verkefnið

OG-un á EDA-þáttum með tveimur liðum

Gefin rökyrðing á tilteknu formi, er hægt að finna gildi á breyturnar þannig að rökyrðingin sé sönn?

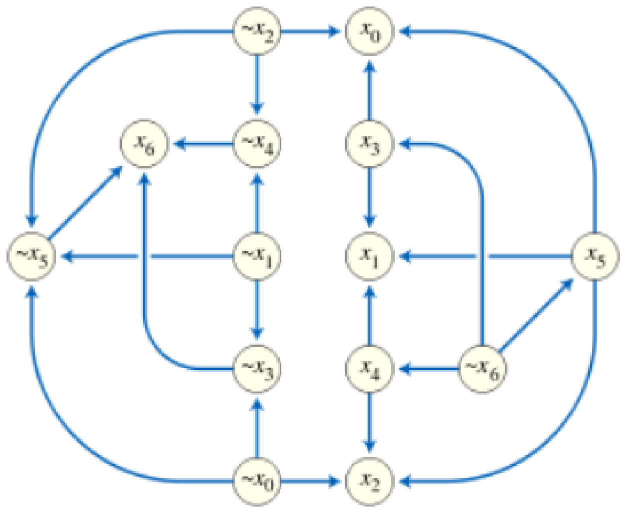
Dæmi: $(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge$
$(x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge$
$(x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$

Umritum hvern þátt $(x_0 \vee x_1)$ sem $(\neg x_0 \rightarrow x_1), (\neg x_1 \rightarrow x_0)$

Skilgreinum "leiðingarnet" (*implication graph*):

Hnútur fyrir hverja breytu og fyrir neitun hennar

Ör fyrir hverja leiðingu í umritun rökyrðingarinnar



frá Wikipedia

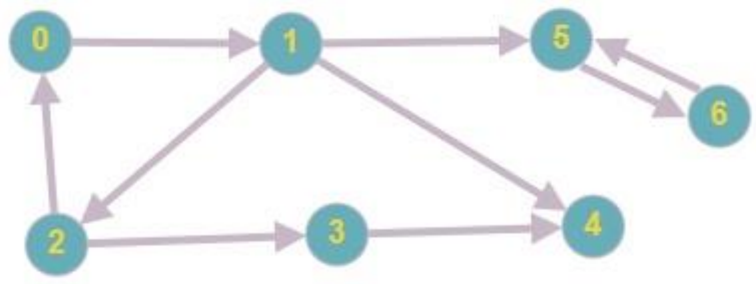
Hægt að sýna að rökyrðingin er fullnægjanleg þþaa
engin breyta og neitun hennar er í sama sterktengda hlut leiðingarnetsins

Aspvall, Plass, Tarjan [1979]

3-SAT verkefnið er NP-complete, en 2-SAT hefur línulegt reiknirit!

Fyrirlestraæfingar

1. Hverjir eru sterktengdir hlutar netsins hér fyrir neðan:



2. Snúið stikunum í netinu hér að ofan við, framkvæmið fyrri umferð í reikniriti Kosaraju-Sharir og sýnið hlaðann.

3. Nefnið dæmi um líklegar vefsíður í hlutunum *IN*, *OUT* og *Disconncted* í Þverslaufuskipulagi vefsins.