Greedy Rectilinear Drawings

Philipp Kindermann
Cheriton School of Computer Science
University of Waterloo

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Patrizio Angelini
Michael A. Bekos

Walter Didimo
Luca Grilli
Alessandra Tappini

Antonios Symvonis

KIT
Karlsruher Institut für Technologie

Tamara Mchedlidze
Roman Prutkin

Greedy Rectilinear Drawings

Philipp Kindermann
Cheriton School of Computer Science
University of Waterloo

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

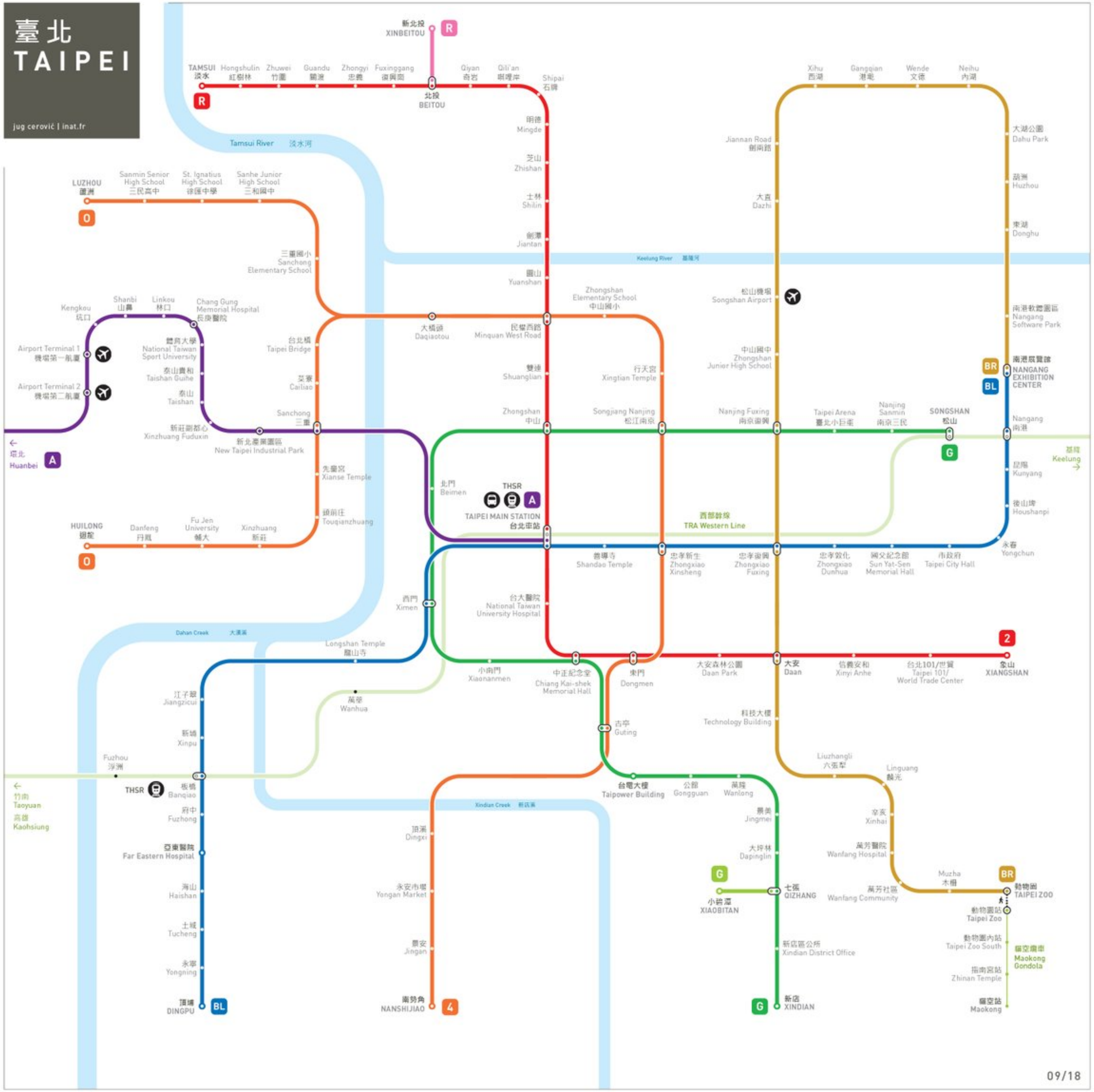
KIT
Karlsruher Institut für Technologie

Patrizio Angelini
Michael A. Bekos

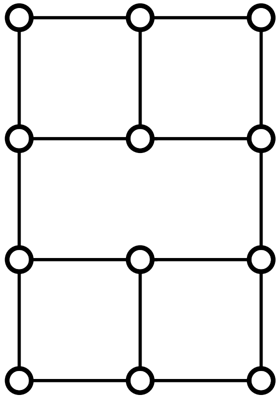
Walter Didimo
Luca Grilli
Alessandra Tappini

Antonios Symvonis

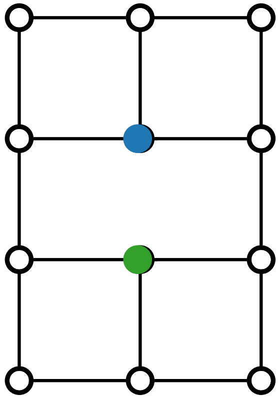
Tamara Mchedlidze
Roman Prutkin


臺北
TAIPEI
jug cerović | inat.fr
新北投 XINBEITOU R
TAMSUI 淡水 R
Hongshulin 紅樹林
Zhuwei 竹圍
Guandu 關渡
Zhongyi 忠義
Fuxinggang 復興崗
北投 BEITOU
Qiyan 奇岩
Qili'an 唭哩岸
Shipai 石牌
明德 Mingde
芝山 Zhishan
士林 Shilin
劍潭 Jiantan
圓山 Yuanshan
民權西路 Minquan West Road
雙連 Shuanglian
Zhongshan 中山
Tamsui River 淡水河
Keelung River 基隆河
Dahan Creek 大漢溪
Xindian Creek 新店溪
LUZHOU 蘆洲 O
Sanmin Senior High School 三民高中
St. Ignatius High School 徐匯中學
Sanhe Junior High School 三和國中
三重國小 Sanchong Elementary School
大橋頭 Daqiaotou
台北橋 Taipei Bridge
菜寮 Cailiao
Sanchong 三重
先嗇宮 Xianse Temple
頭前庄 Touqianzhuang
HUILONG 迴龍 O
Danfeng 丹鳳
Fu Jen University 輔大
Xinzhuang 新莊
Kengkou 坑口
Shanbi 山鼻
Linkou 林口
Chang Gung Memorial Hospital 長庚醫院
Airport Terminal 1 機場第一航廈
Airport Terminal 2 機場第二航廈
體育大學 National Taiwan Sport University
泰山貴和 Taishan Guihe
泰山 Taishan
新莊副都心 Xinzhuang Fuduxin
新北產業園區 New Taipei Industrial Park
← 環北 Huanbei A
北門 Beimen
THSR
A
TAIPEI MAIN STATION 台北車站
Zhongshan Elementary School 中山國小
行天宮 Xingtian Temple
Songjiang Nanjing 松江南京
Xihu 西湖
Gangqian 港墘
Wende 文德
Neihu 內湖
Jiannan Road 劍南路
大直 Dazhi
松山機場 Songshan Airport
中山國中 Zhongshan Junior High School
Nanjing Fuxing 南京復興
Taipei Arena 臺北小巨蛋
Nanjing Sanmin 南京三民
SONGSHAN 松山 G
大湖公園 Dahu Park
葫洲 Huzhou
東湖 Donghu
南港軟體園區 Nangang Software Park
南港展覽館 NANGANG EXHIBITION CENTER BR BL
Nangang 南港
基隆 Keelung →
昆陽 Kunyang
後山埤 Houshanpi
永春 Yongchun
西部幹線 TRA Western Line
善導寺 Shandao Temple
忠孝新生 Zhongxiao Xinsheng
忠孝復興 Zhongxiao Fuxing
忠孝敦化 Zhongxiao Dunhua
國父紀念館 Sun Yat-Sen Memorial Hall
市政府 Taipei City Hall
西門 Ximen
台大醫院 National Taiwan University Hospital
Longshan Temple 龍山寺
小南門 Xiaonanmen
中正紀念堂 Chiang Kai-shek Memorial Hall
東門 Dongmen
大安森林公園 Daan Park
大安 Daan
信義安和 Xinyi Anhe
台北101/世貿 Taipei 101/ World Trade Center
2 象山 XIANGSHAN
萬華 Wanhua
江子翠 Jiangzicui
新埔 Xinpu
Fuzhou 浮洲
← 竹南 Taoyuan 高雄 Kaohsiung
THSR
板橋 Banqiao
府中 Fuzhong
亞東醫院 Far Eastern Hospital
海山 Haishan
土城 Tucheng
永寧 Yongning
頂埔 DINGPU BL
古亭 Guting
科技大樓 Technology Building
台電大樓 Taipower Building
公館 Gongguan
萬隆 Wanlong
Liuzhangli 六張犁
Linguang 麟光
頂溪 Dingxi
永安市場 Yongan Market
景安 Jingan
南勢角 NANSHIJIAO 4
景美 Jingmei
大坪林 Dapinglin
G 小碧潭 XIAOBITAN
七張 QIZHANG
新店區公所 Xindian District Office
新店 XINDIAN G
辛亥 Xinhai
萬芳醫院 Wanfang Hospital
萬芳社區 Wanfang Community
Muzha 木柵
BR 動物園 TAIPEI ZOO
動物園站 Taipei Zoo
動物園內站 Taipei Zoo South
指南宮站 Zhinan Temple
貓空站 Maokong
貓空纜車 Maokong Gondola
09/18

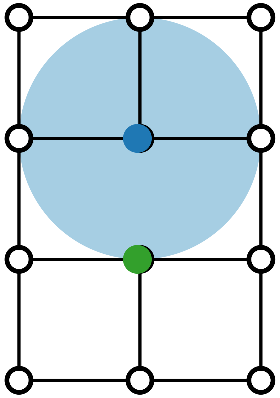
Greedy Rectilinear Drawings



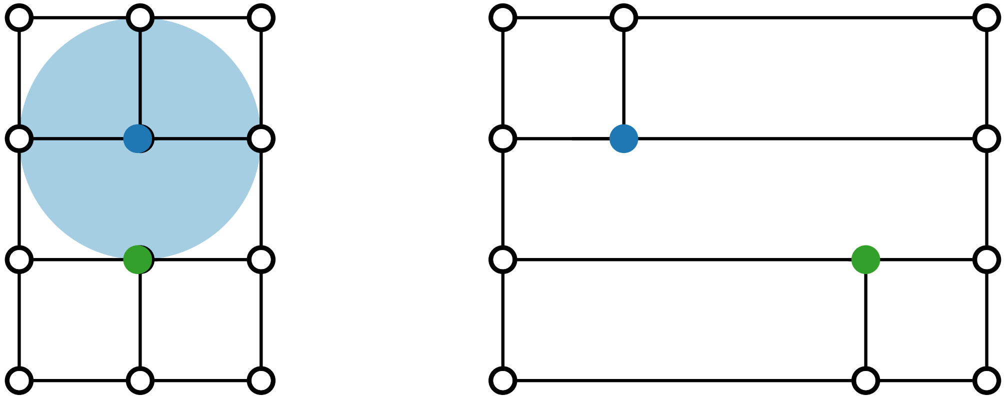
Greedy Rectilinear Drawings



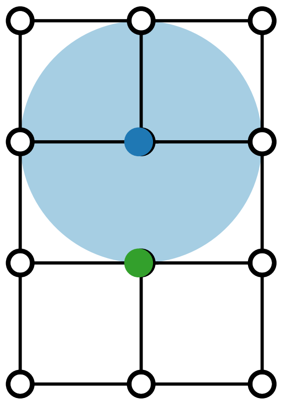
Greedy Rectilinear Drawings

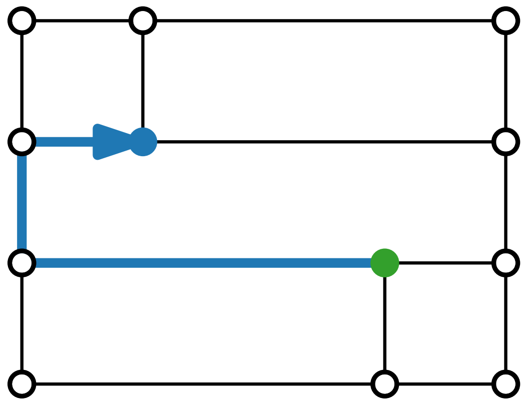


Greedy Rectilinear Drawings

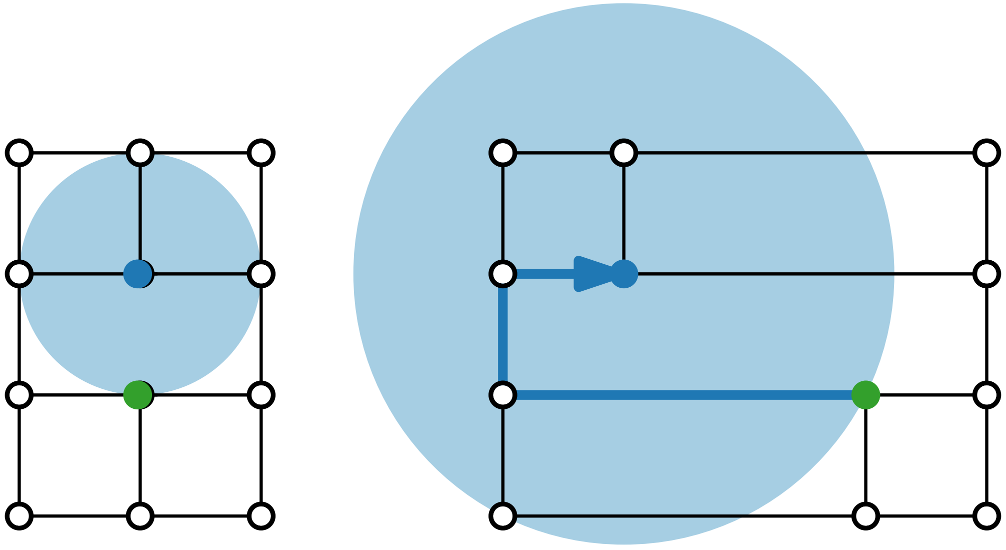


Greedy Rectilinear Drawings

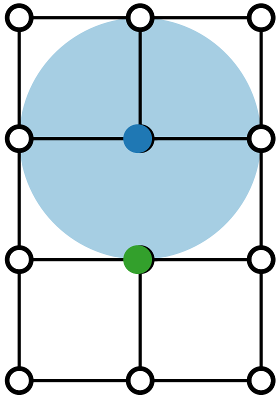


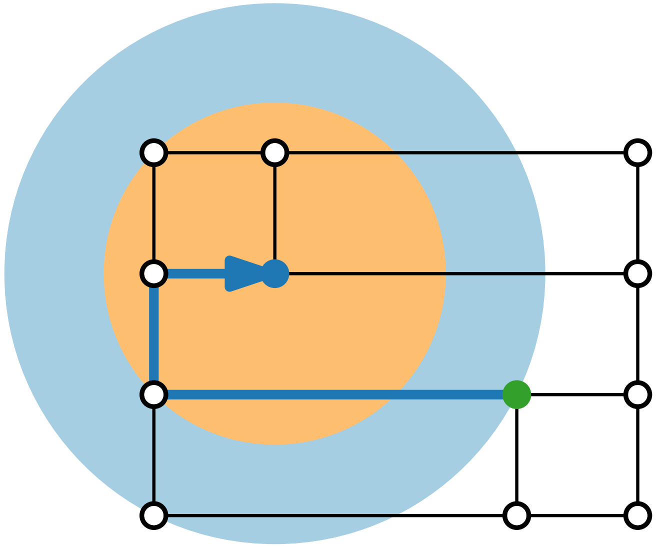


Greedy Rectilinear Drawings

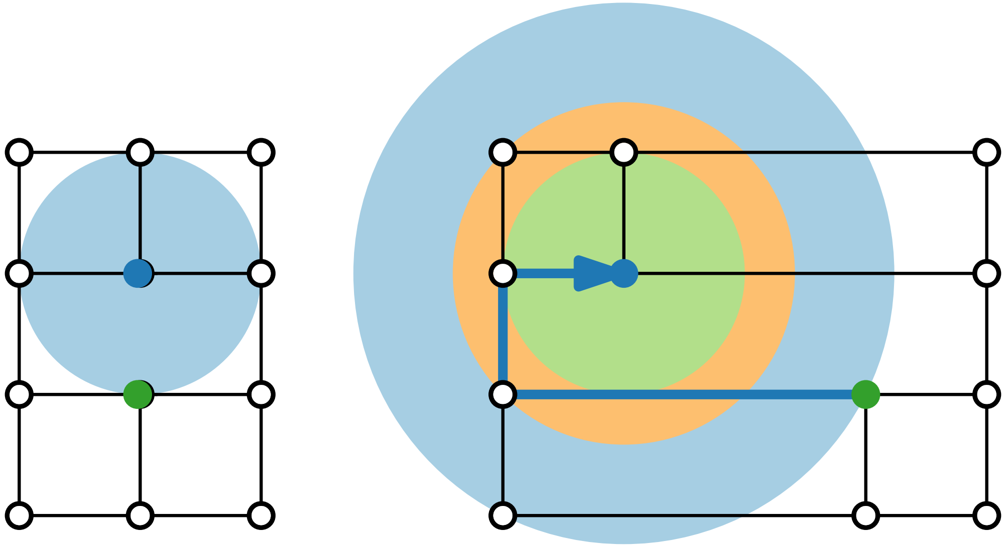


Greedy Rectilinear Drawings

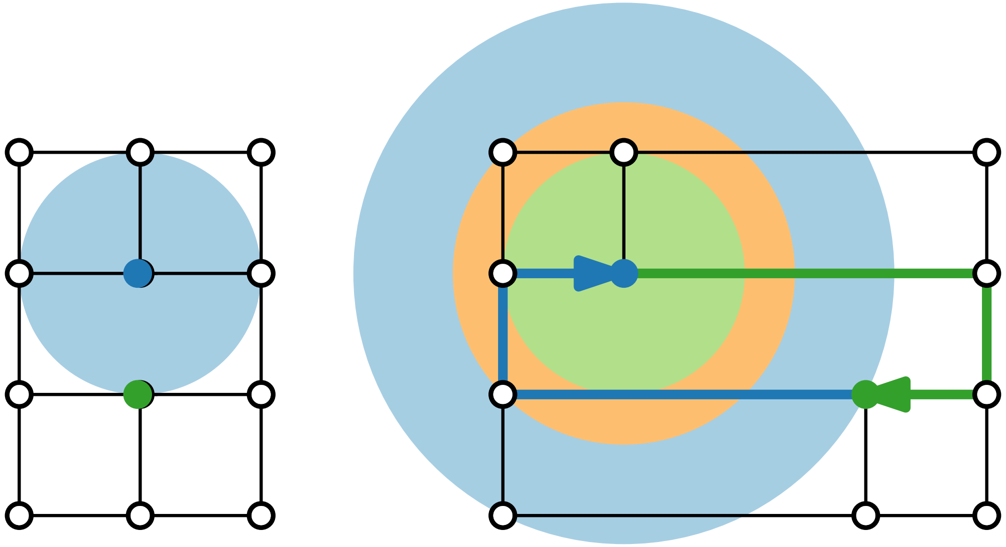




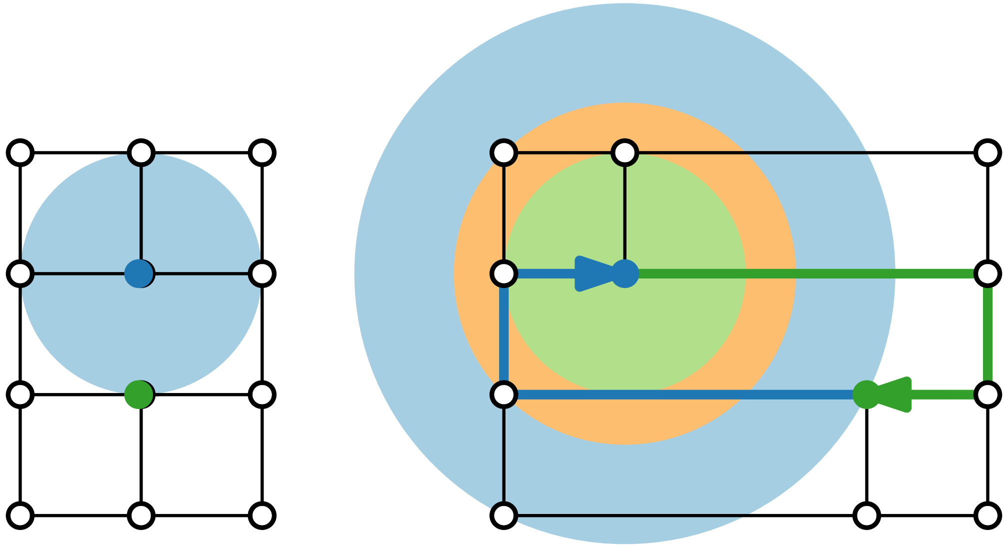
Greedy Rectilinear Drawings



Greedy Rectilinear Drawings



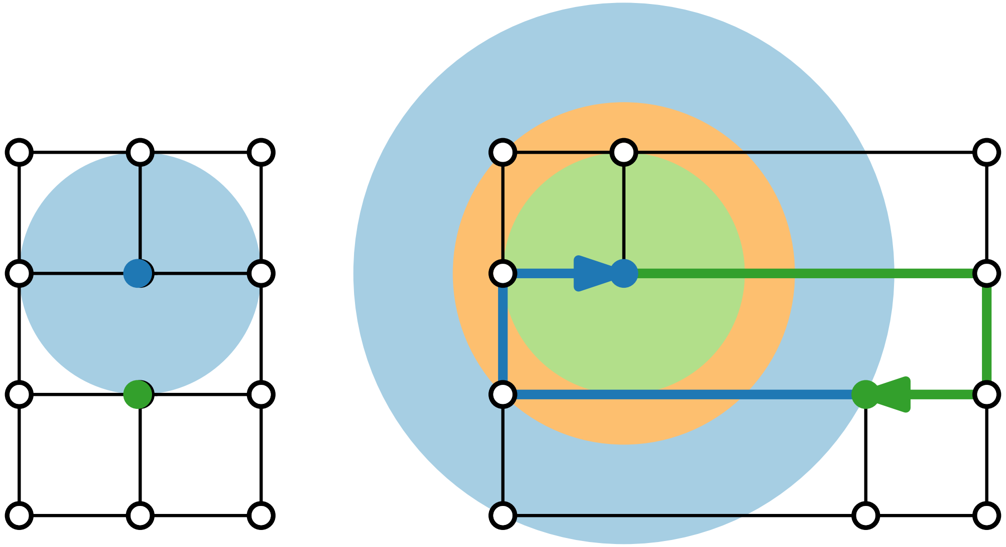
Greedy Rectilinear Drawings



[Papadimitrou, Ratajczak]

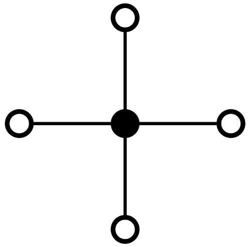
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

Greedy Rectilinear Drawings

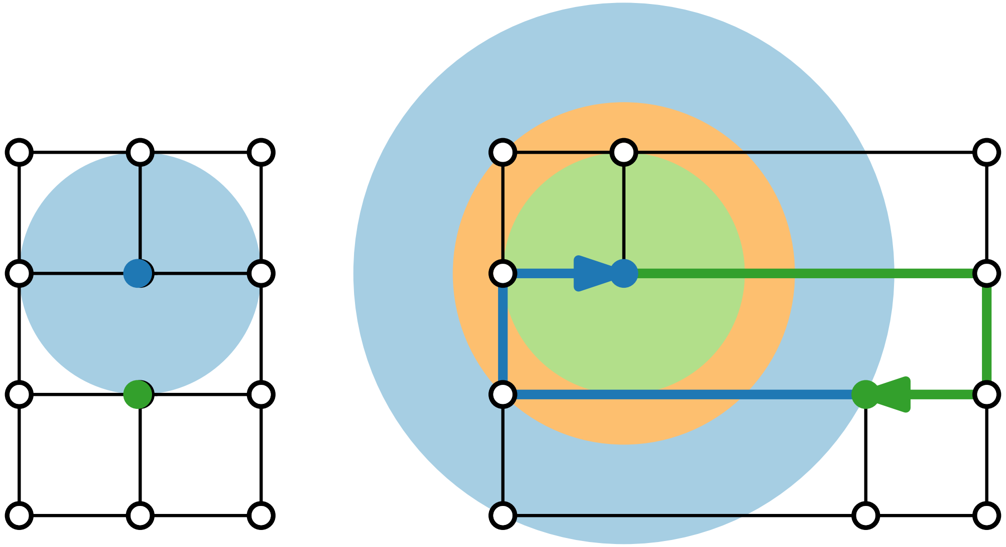


[Papadimitrou, Ratajczak]

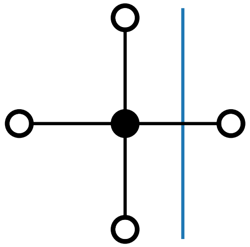
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



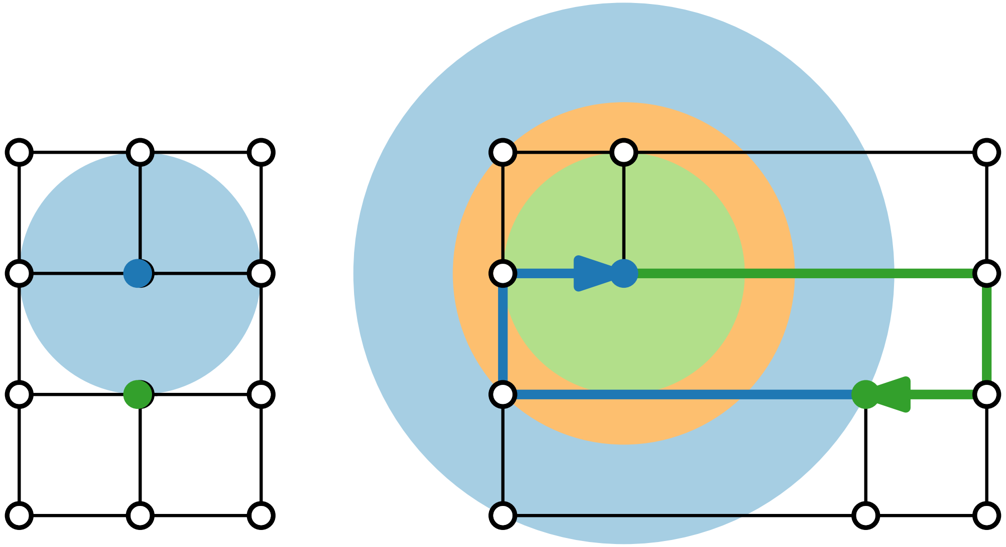
Greedy Rectilinear Drawings



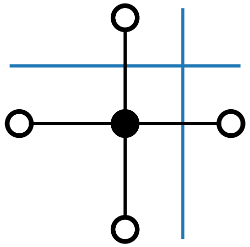
[Papadimitrou, Ratajczak]
Drawing greedy ⇔ cell(v) empty $\forall v$



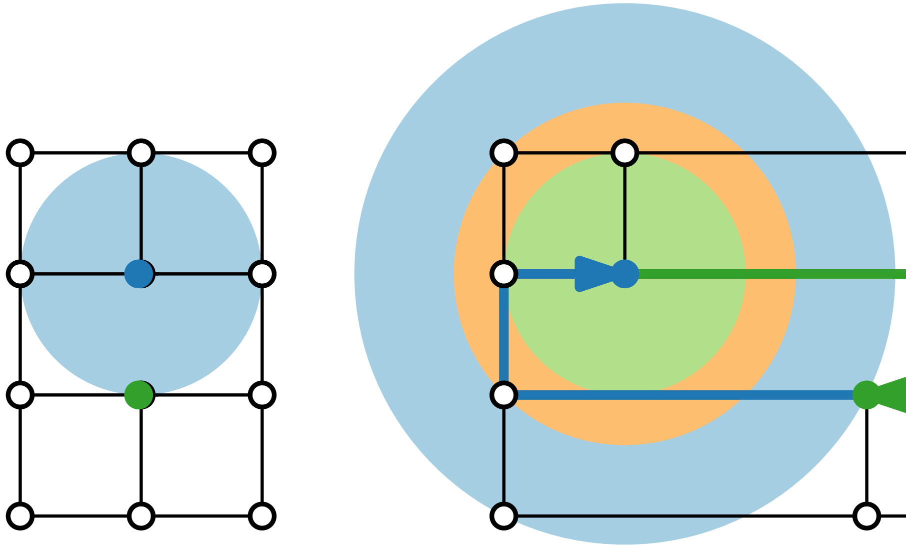
Greedy Rectilinear Drawings



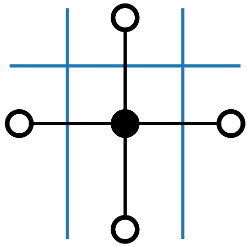
[Papadimitrou, Ratajczak]
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



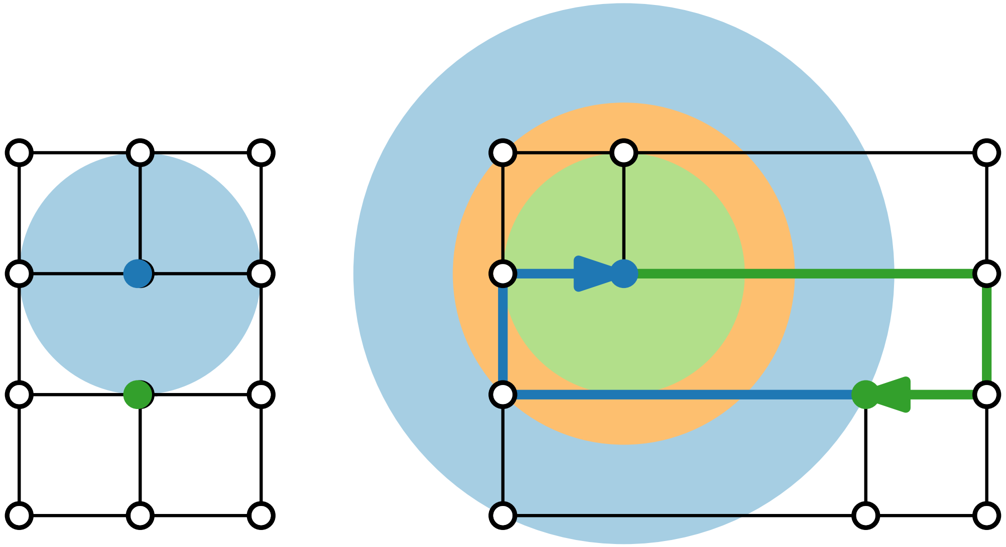
Greedy Rectilinear Drawings



[Papadimitrou, Ratajczak]
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

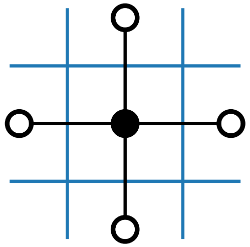


Greedy Rectilinear Drawings

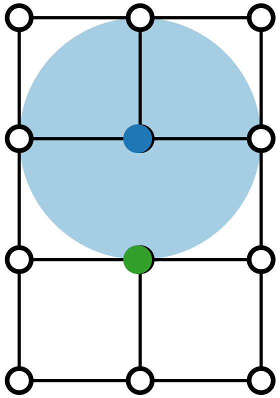


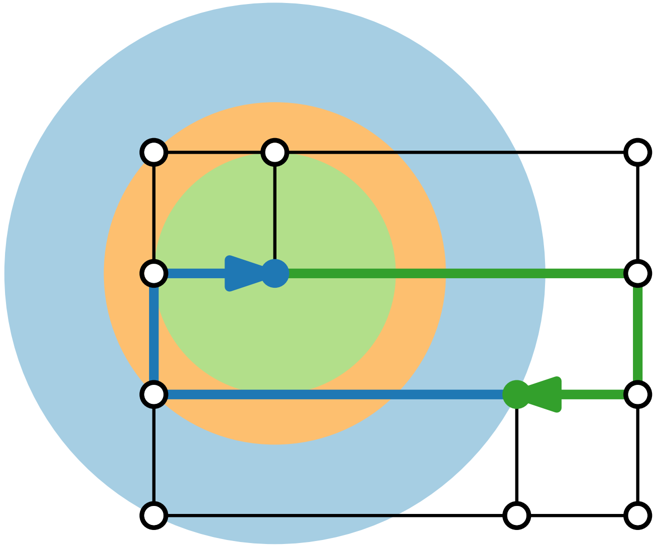
[Papadimitrou, Ratajczak]

Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



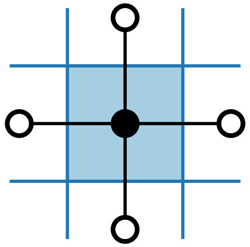
Greedy Rectilinear Drawings



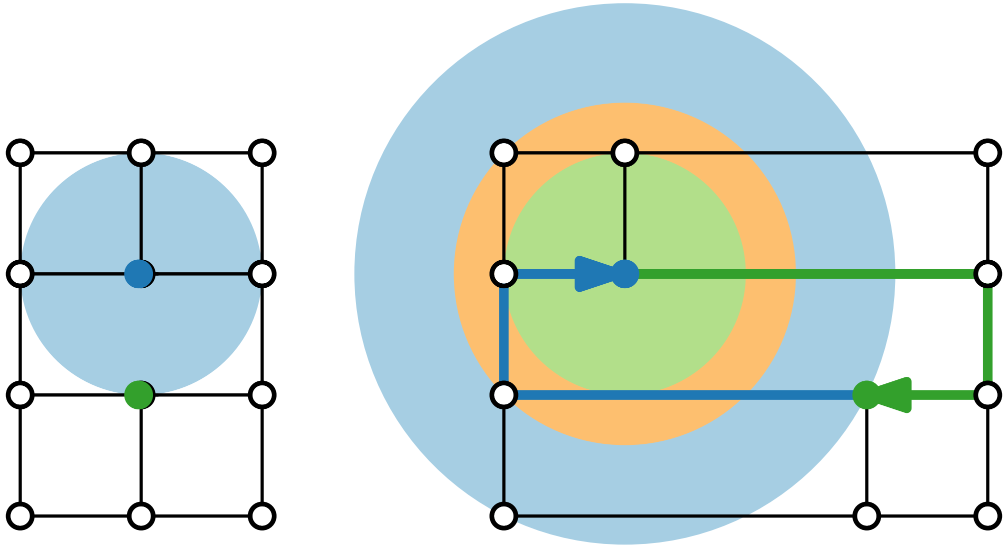


[Papadimitrou, Ratajczak]

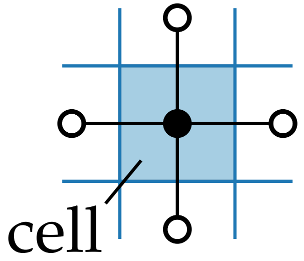
Drawing greedy ⇔ cell(v) empty $\forall v$



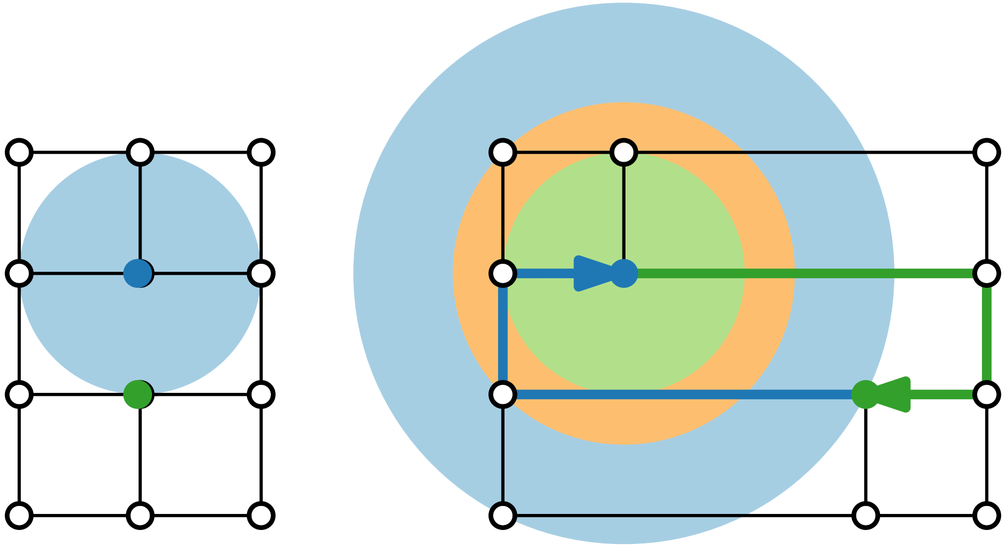
Greedy Rectilinear Drawings



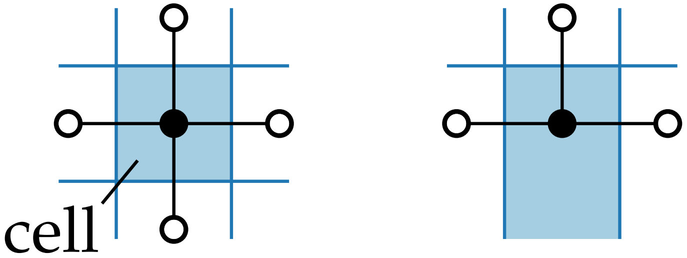
[Papadimitrou, Ratajczak]
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



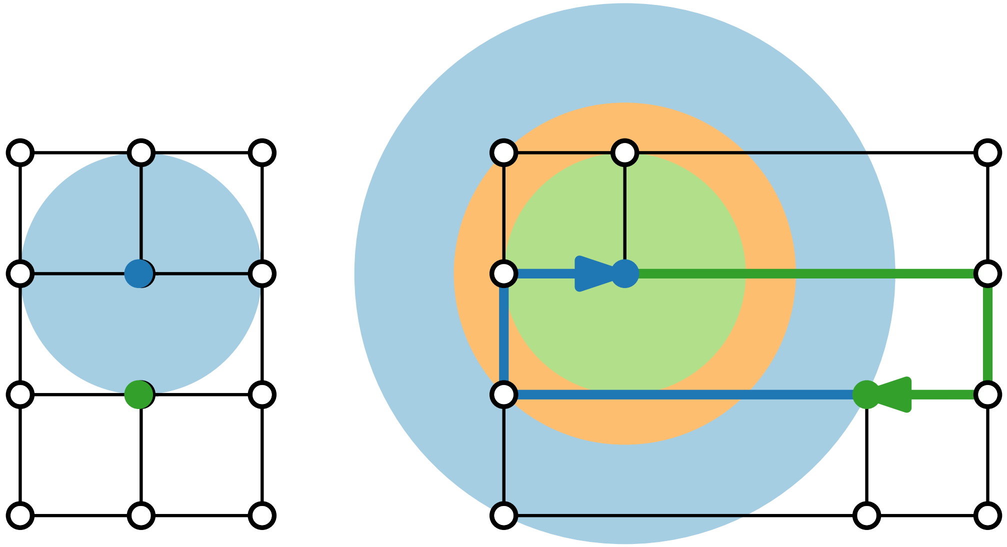
Greedy Rectilinear Drawings



[Papadimitrou, Ratajczak]
Drawing greedy ⇔ cell(v) empty $\forall v$

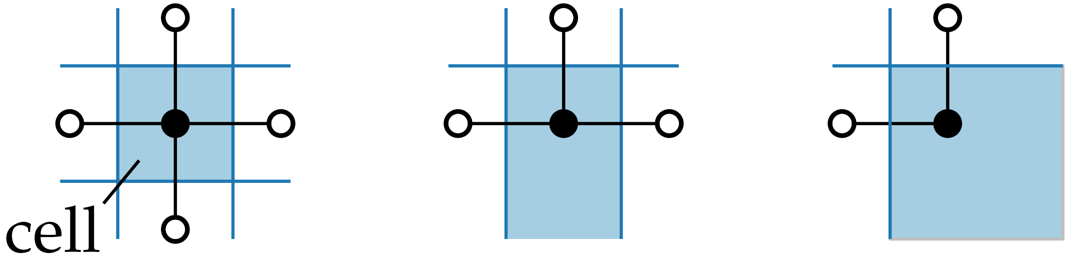


Greedy Rectilinear Drawings

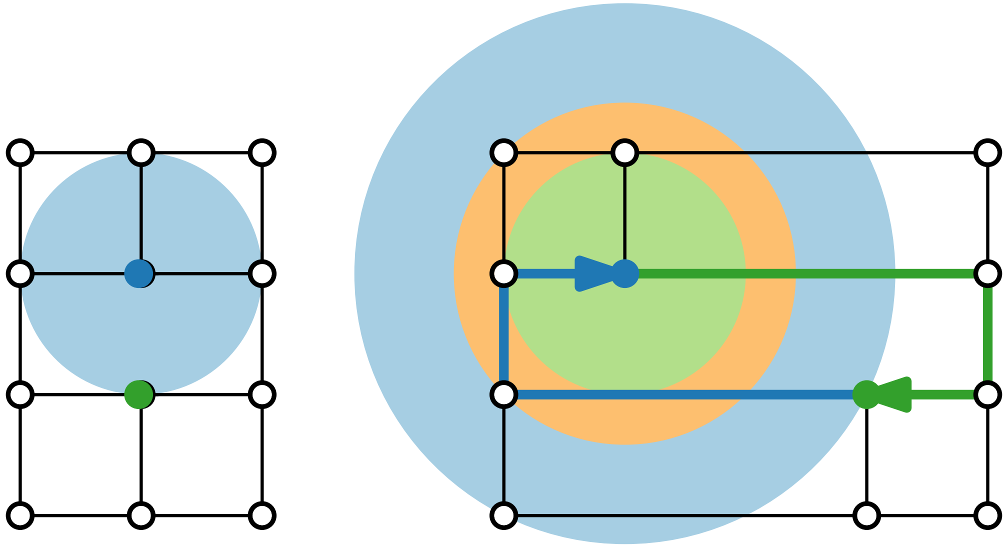


[Papadimitrou, Ratajczak]

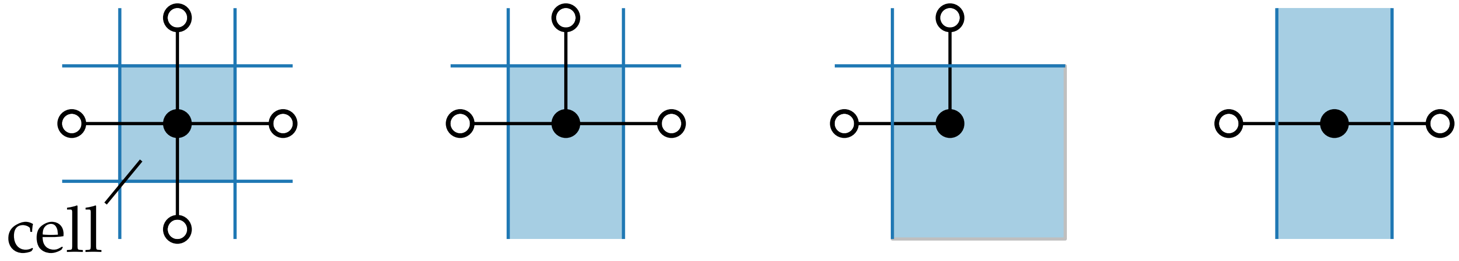
Drawing greedy ⇔ cell(v) empty $\forall v$



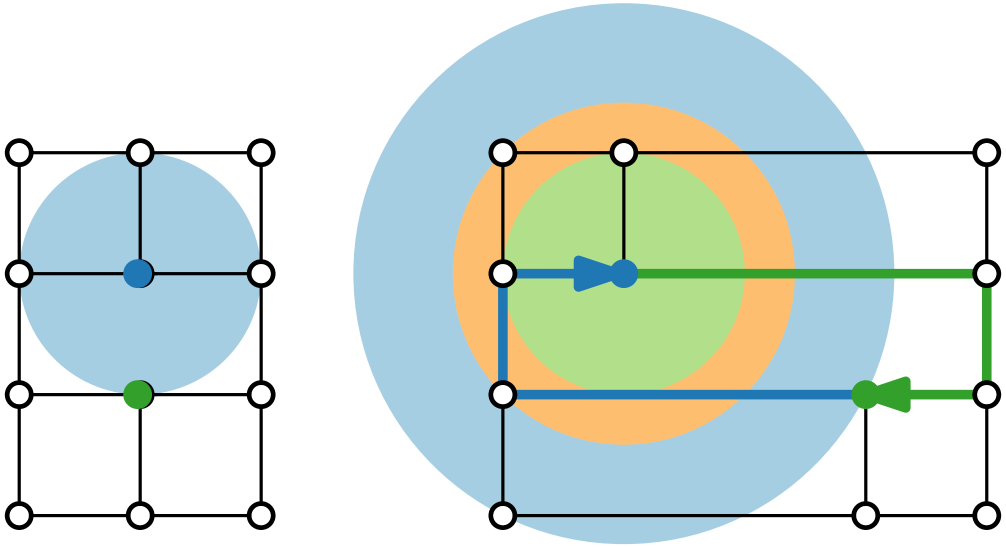
Greedy Rectilinear Drawings



[Papadimitrou, Ratajczak]
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

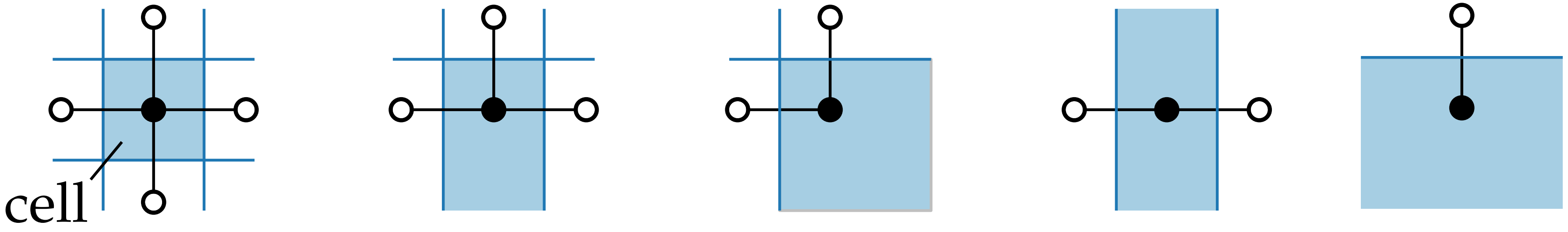


Greedy Rectilinear Drawings

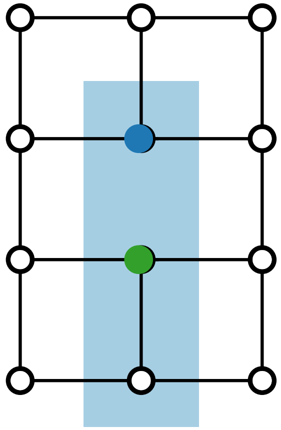


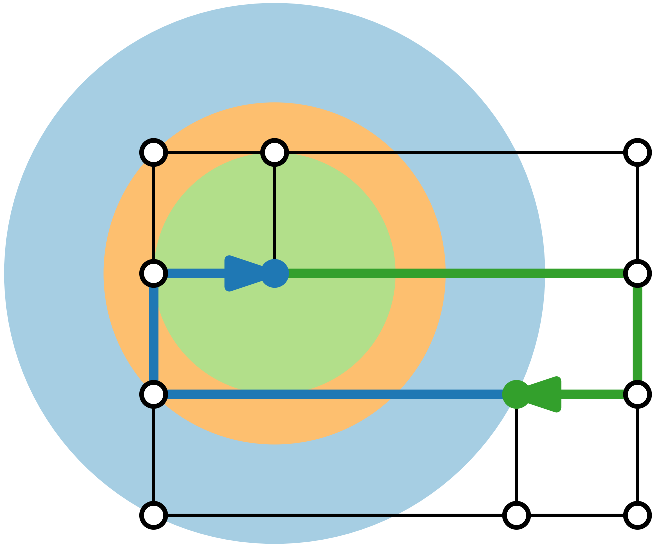
[Papadimitrou, Ratajczak]

Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



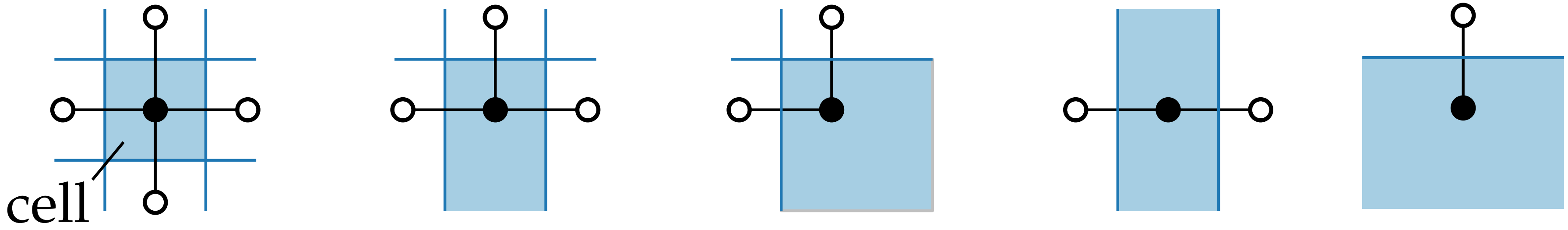
Greedy Rectilinear Drawings



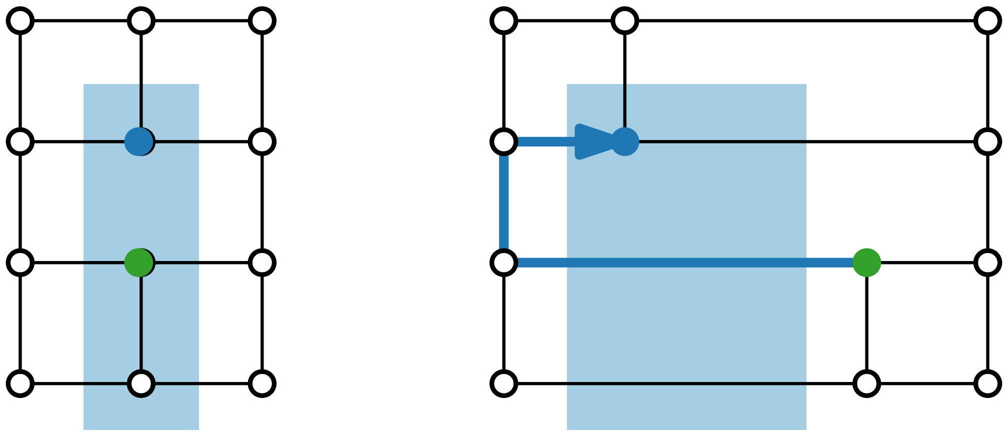


[Papadimitrou, Ratajczak]

Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

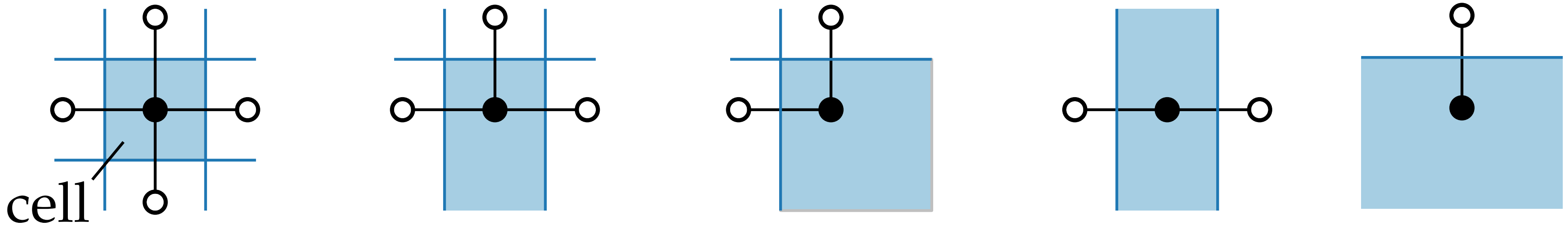


Greedy Rectilinear Drawings

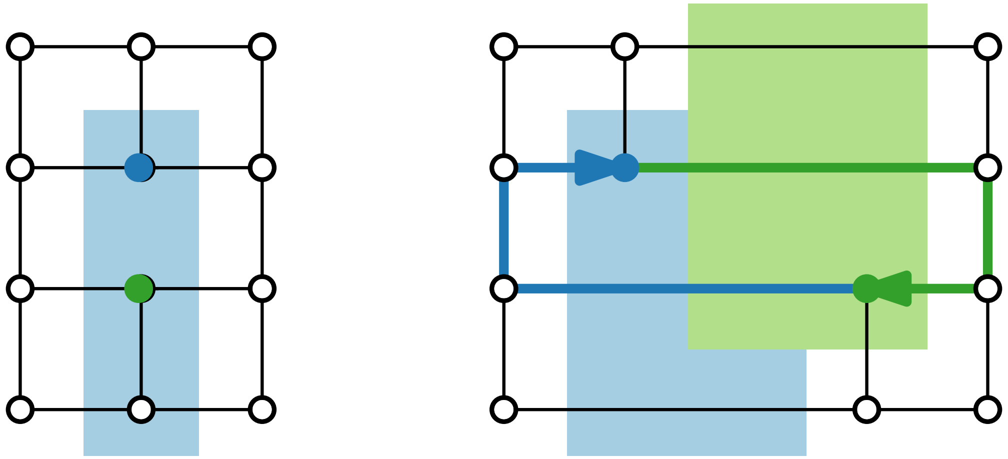


[Papadimitrou, Ratajczak]

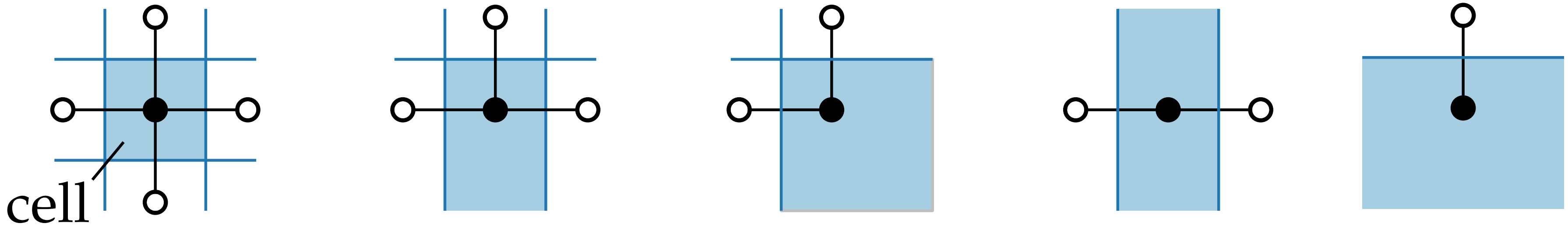
Drawing greedy ⇔ cell(v) empty $\forall v$



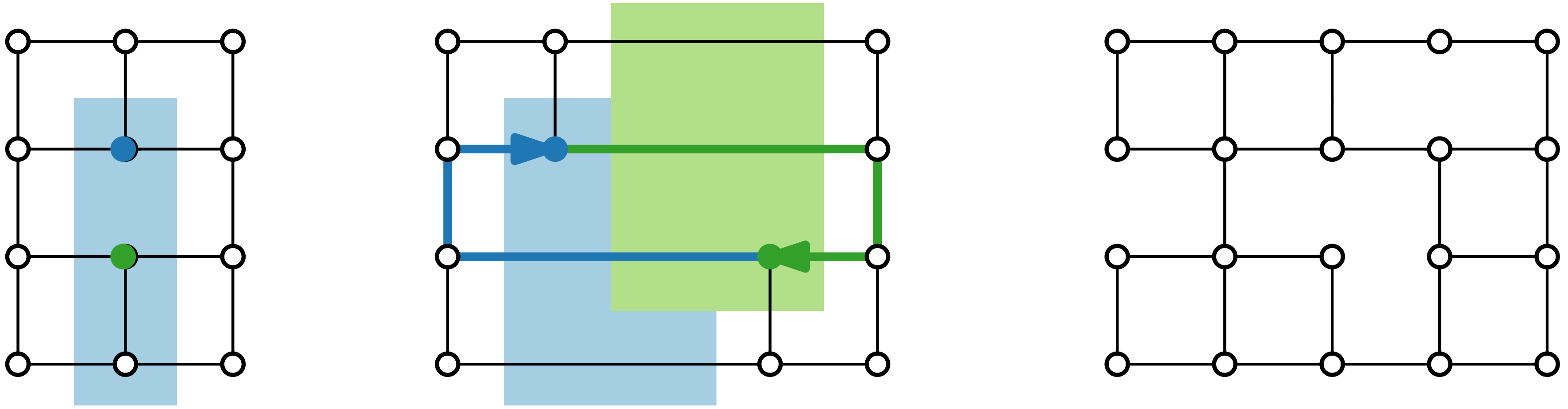
Greedy Rectilinear Drawings



[Papadimitrou, Ratajczak]
Drawing greedy ⇔ cell(v) empty $\forall v$

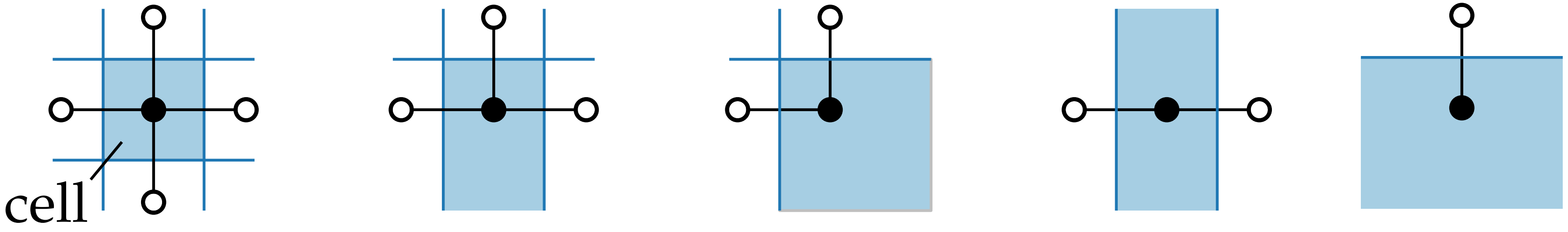


Greedy Rectilinear Drawings

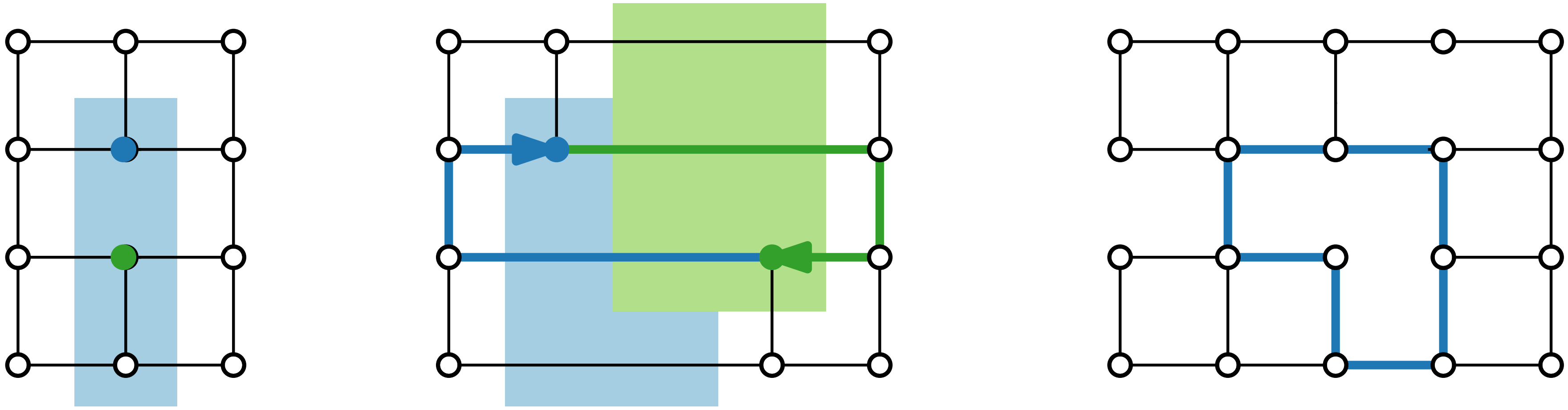


[Papadimitrou, Ratajczak]

Drawing greedy ⇔ cell(v) empty $\forall v$

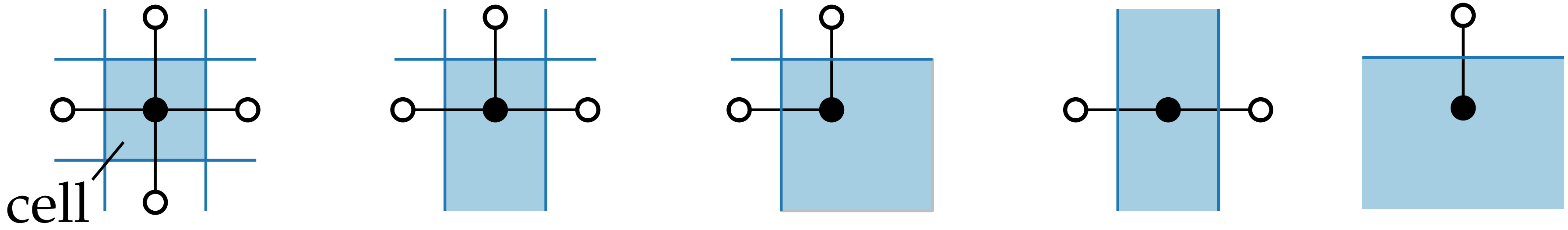


Greedy Rectilinear Drawings

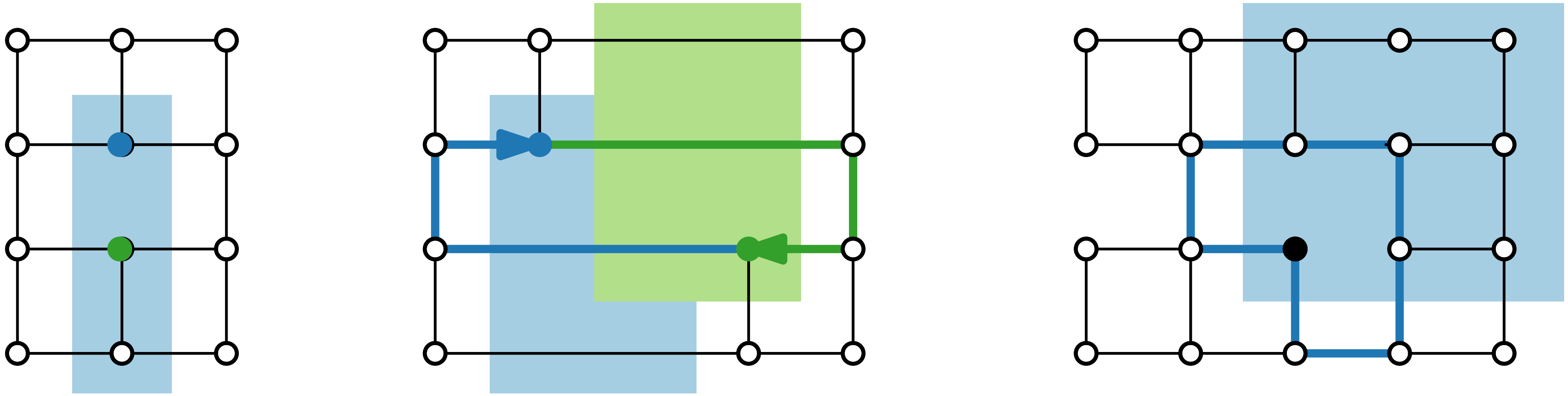


[Papadimitrou, Ratajczak]

Drawing greedy ⇔ cell(v) empty $\forall v$

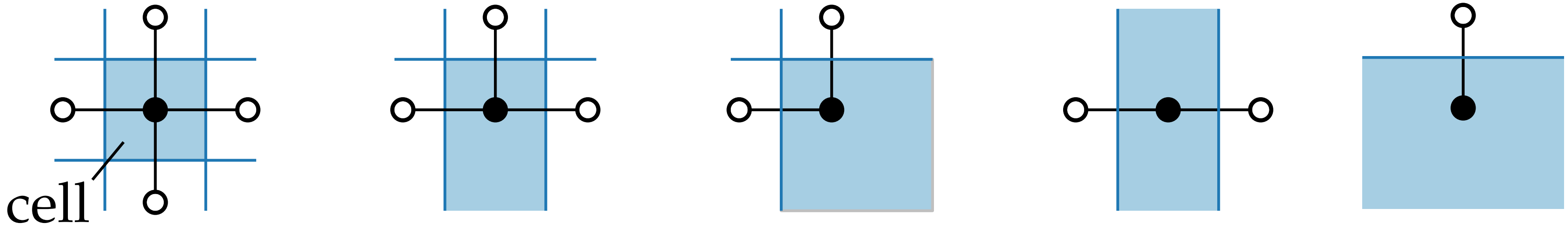


Greedy Rectilinear Drawings

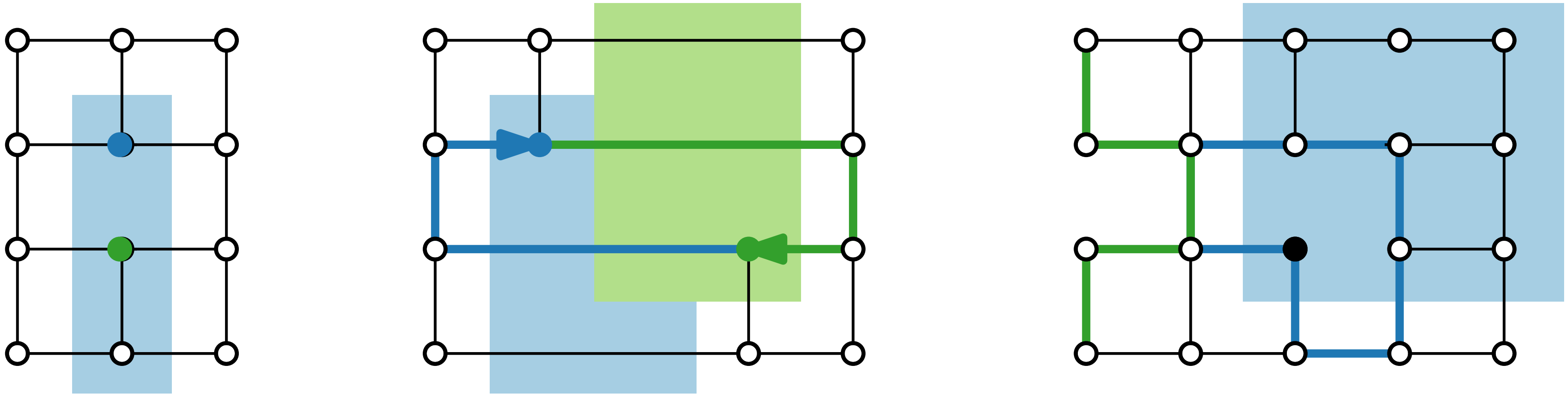


[Papadimitrou, Ratajczak]

Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

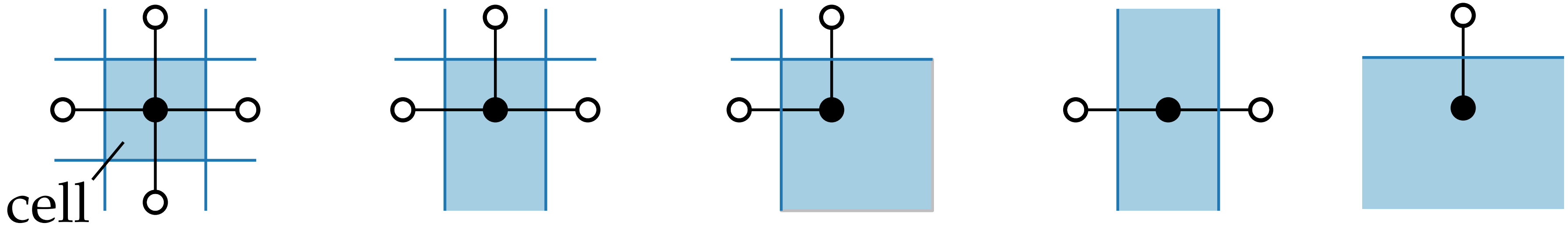


Greedy Rectilinear Drawings

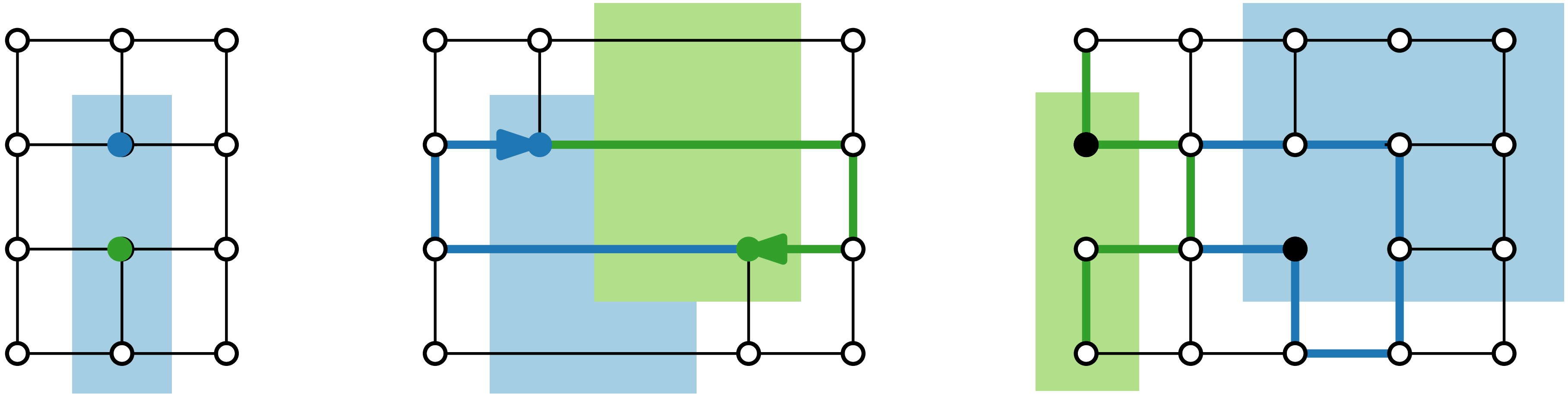


[Papadimitrou, Ratajczak]

Drawing greedy ⇔ cell(v) empty $\forall v$

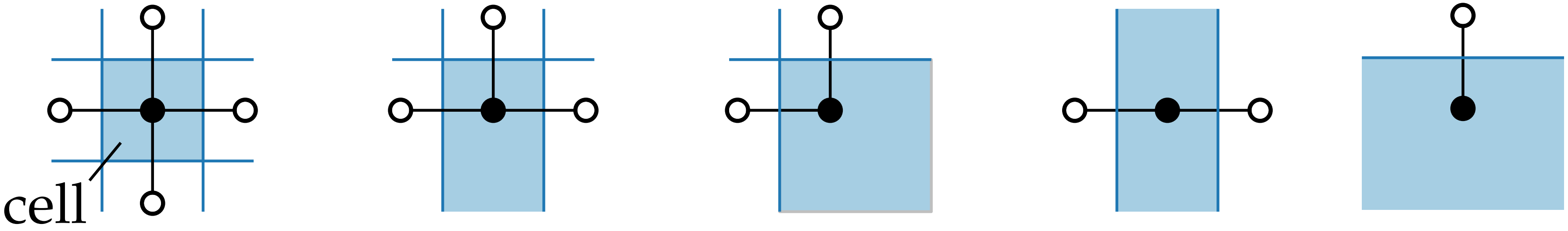


Greedy Rectilinear Drawings

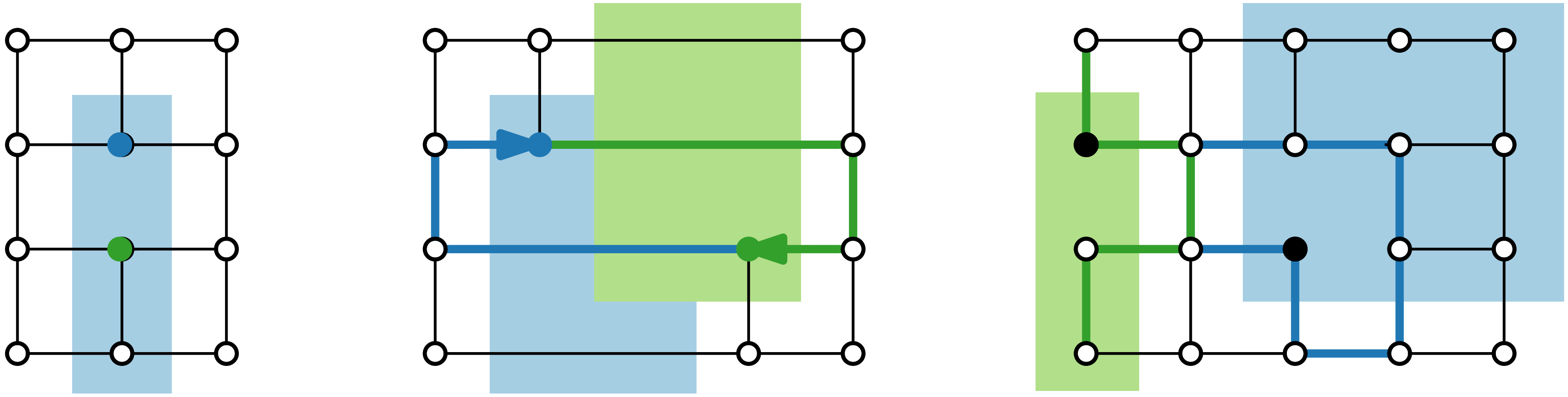


[Papadimitrou, Ratajczak]

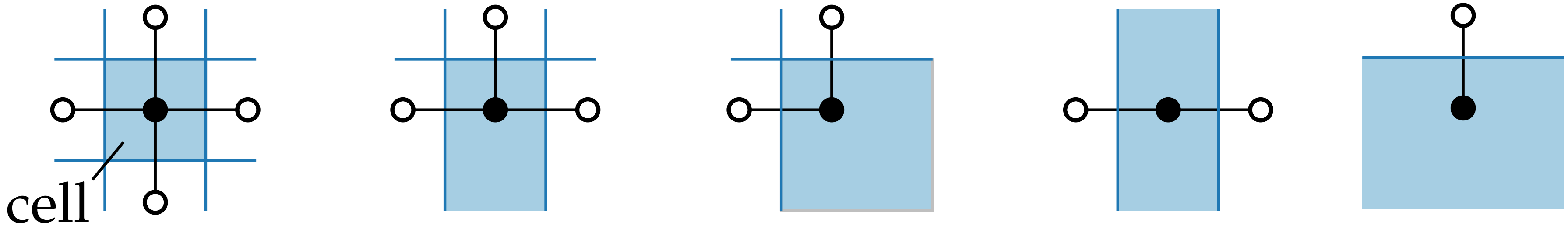
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$



Greedy Rectilinear Drawings

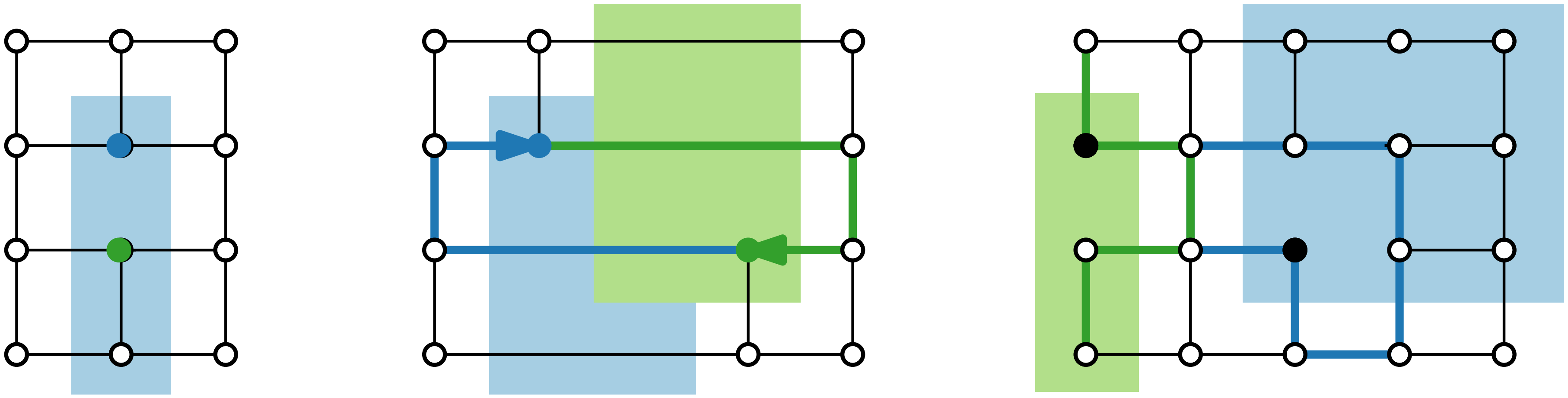


[Papadimitrou, Ratajczak]
Drawing greedy $\Leftrightarrow$ cell(v) empty $\forall v$

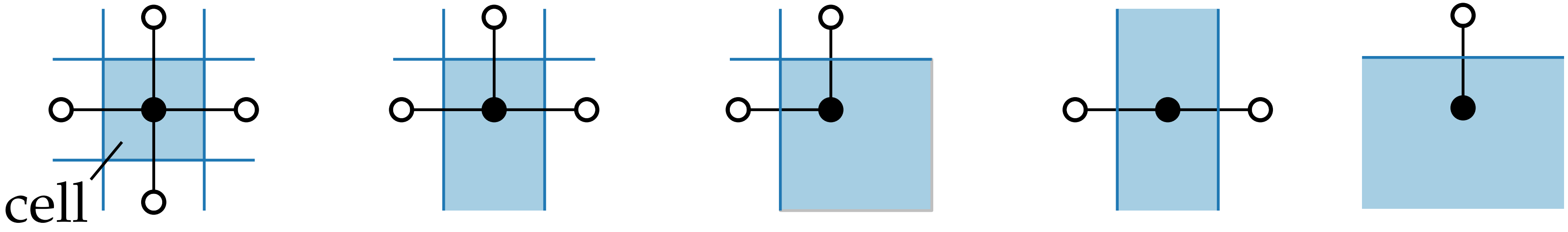


Drawing greedy $\Rightarrow$ convex

Greedy Rectilinear Drawings



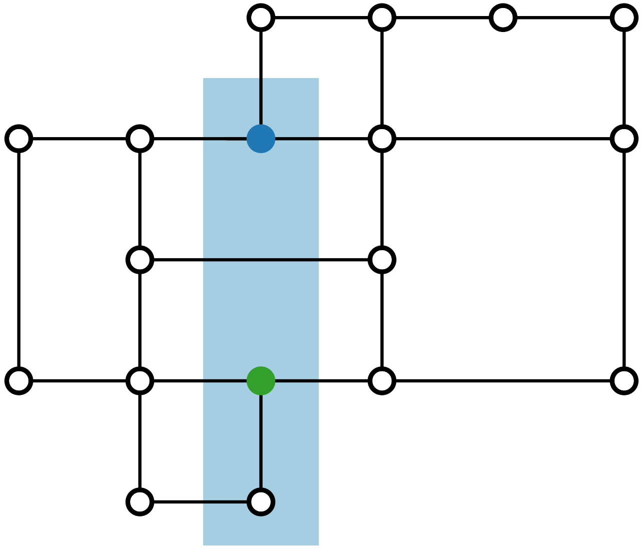
[Papadimitrou, Ratajczak]
Drawing greedy ⇔ cell(v) empty $\forall v$



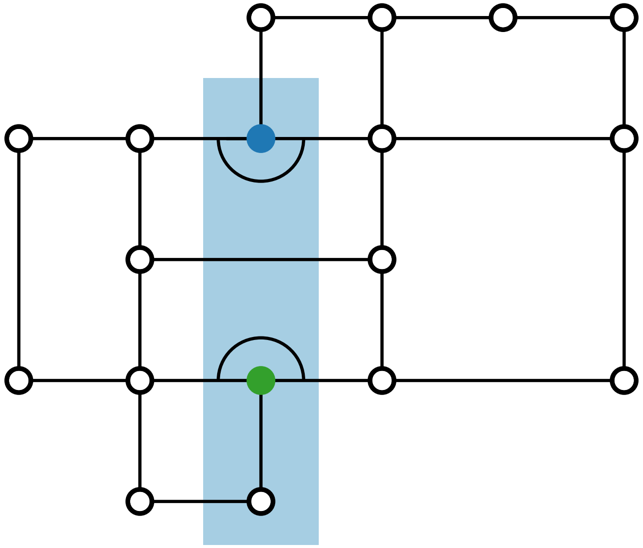
Drawing greedy ⇒ convex

Assume representation given

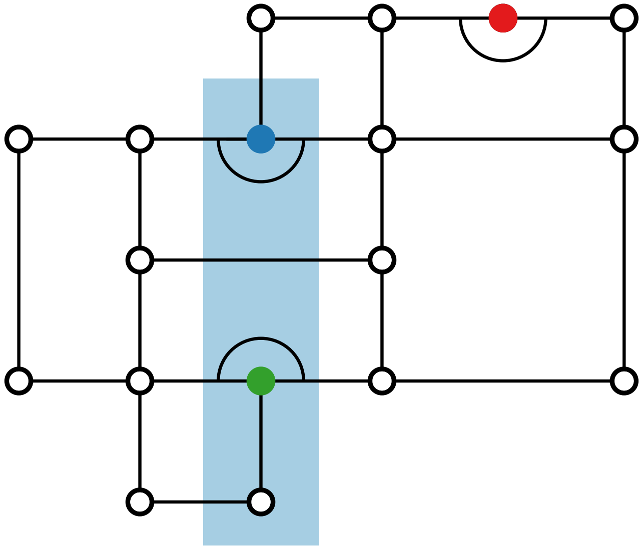
Conflicts



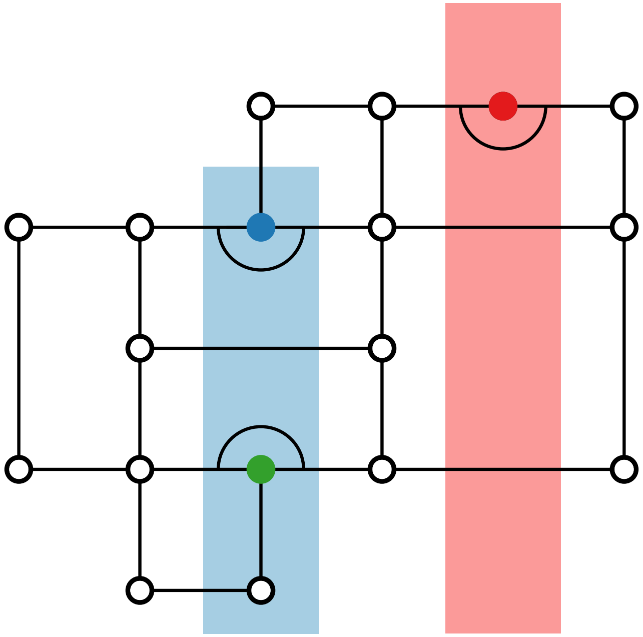
Conflicts



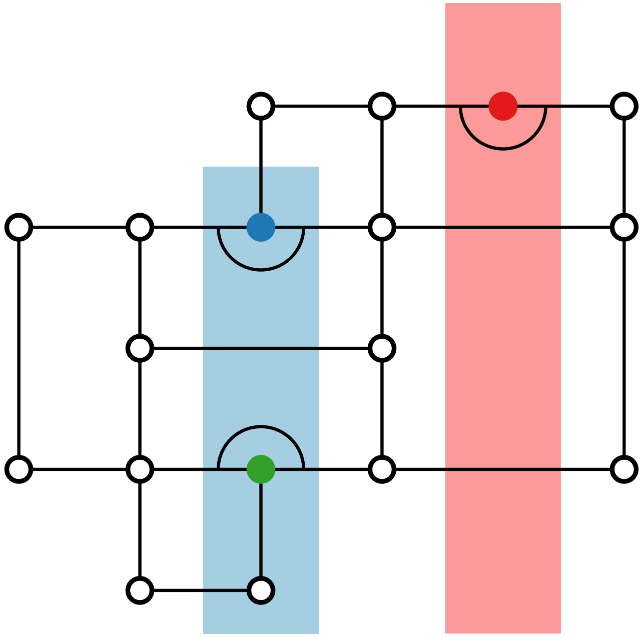
Conflicts

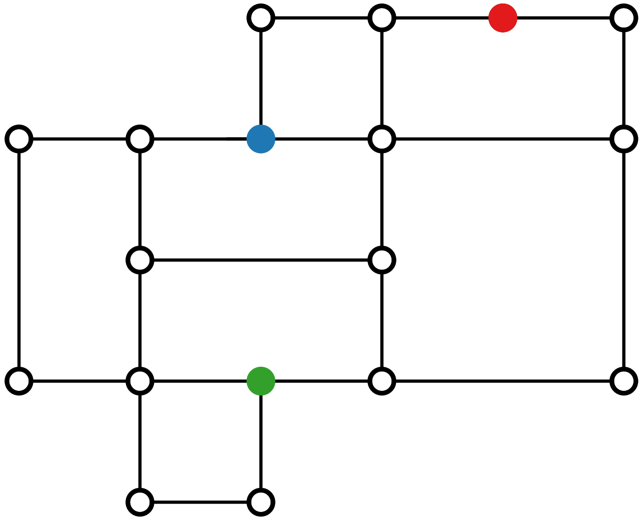


Conflicts

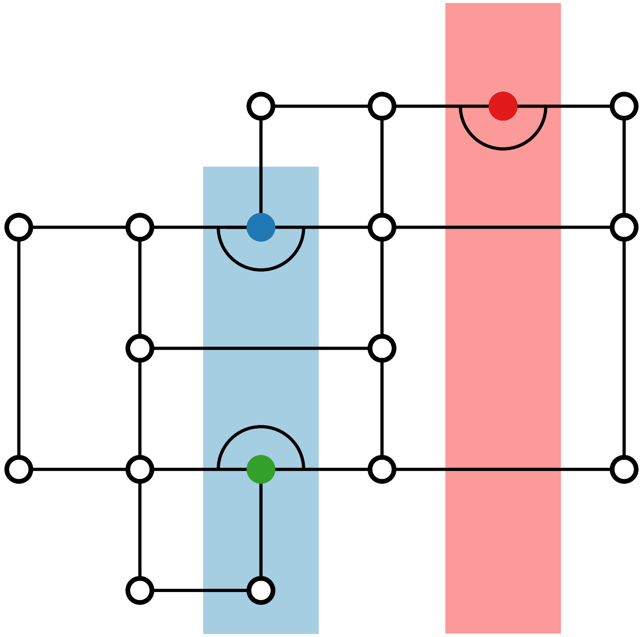


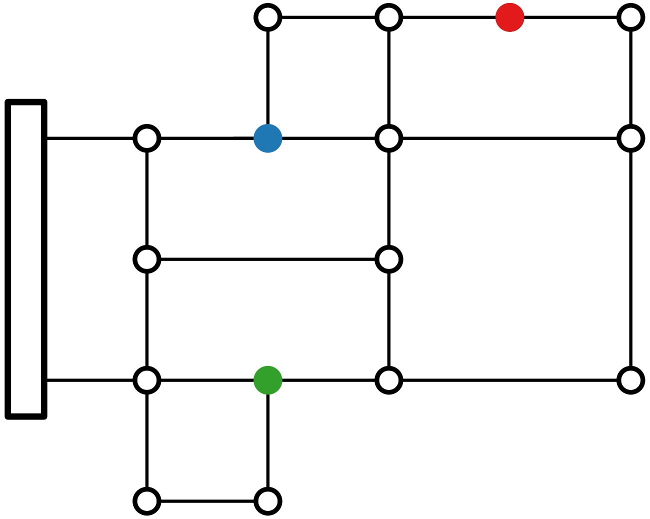
Conflicts



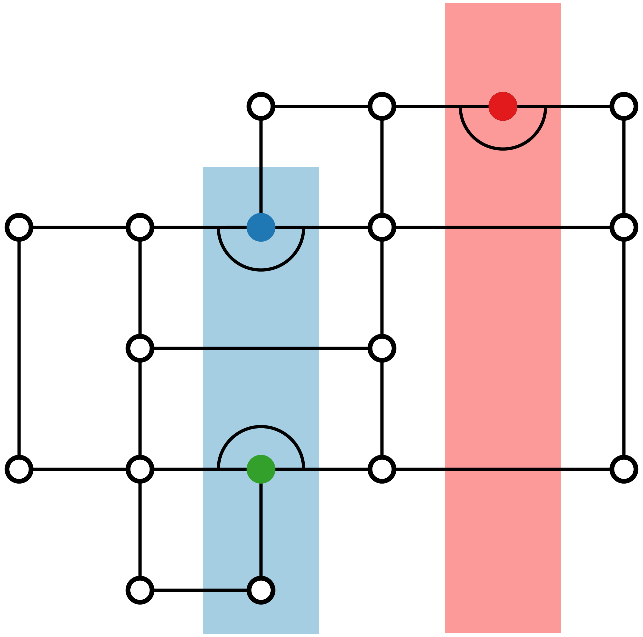


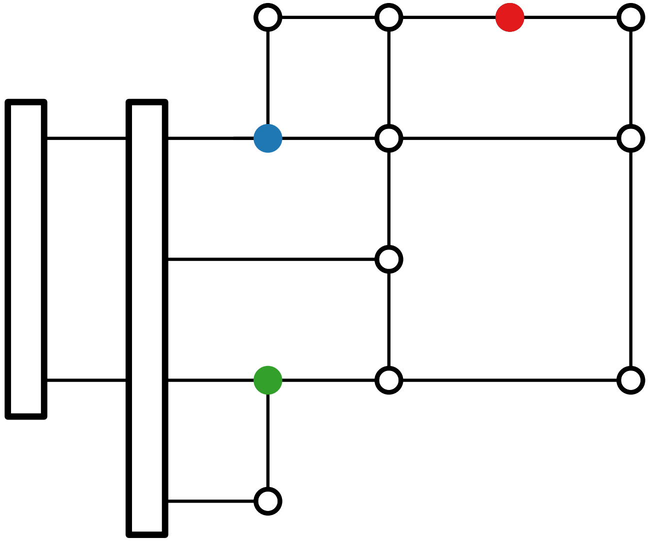
Conflicts



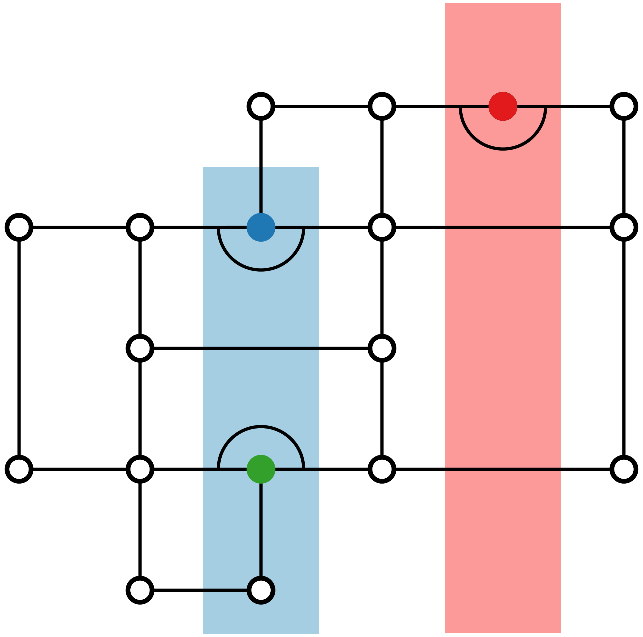


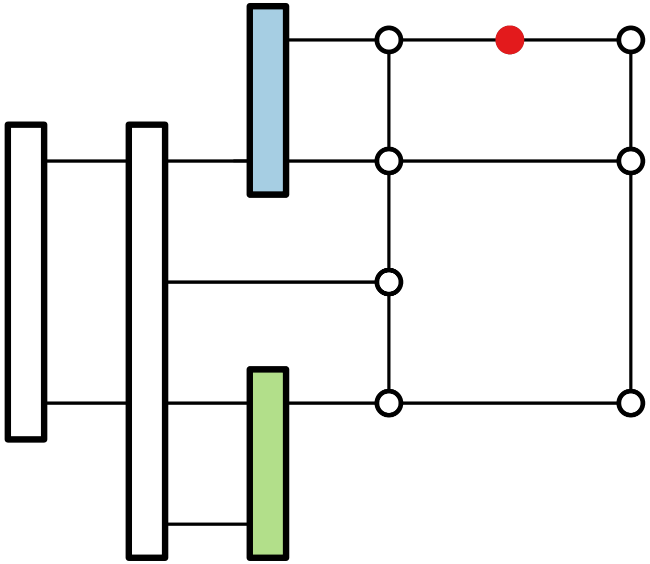
Conflicts



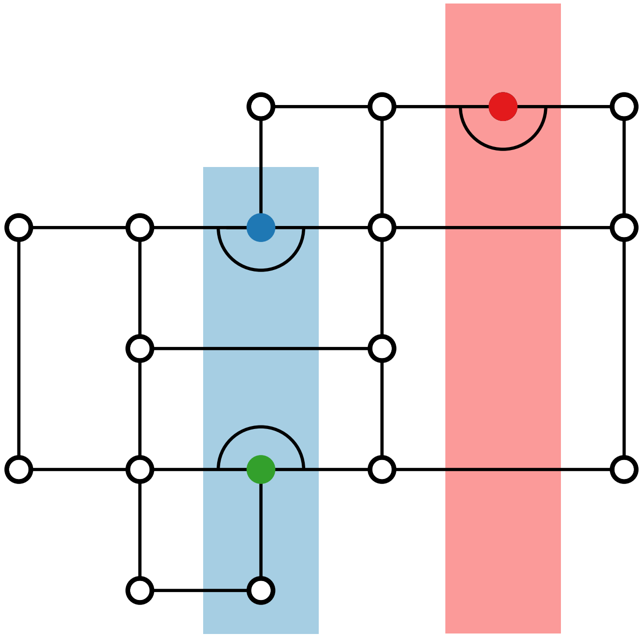


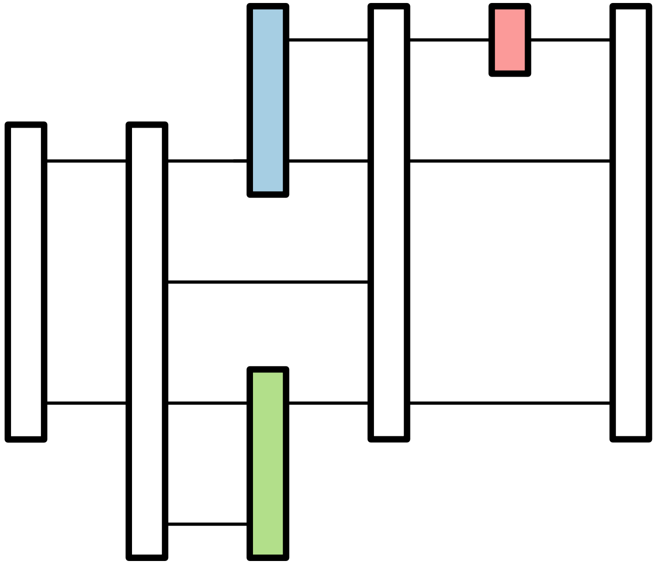
Conflicts



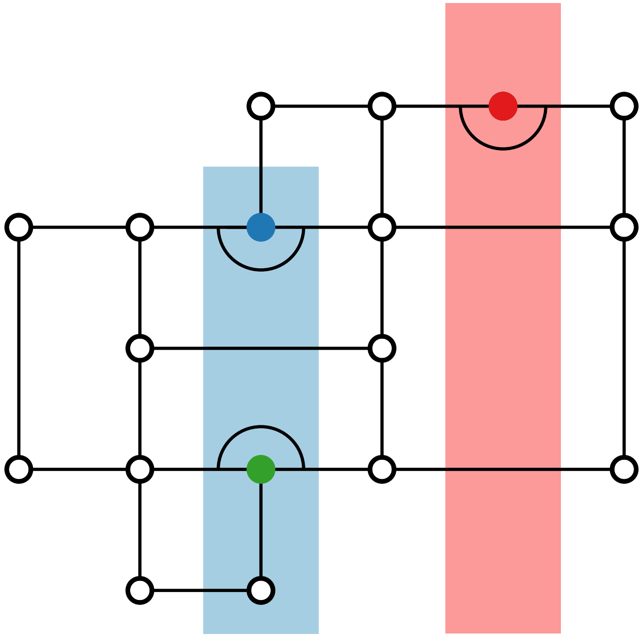


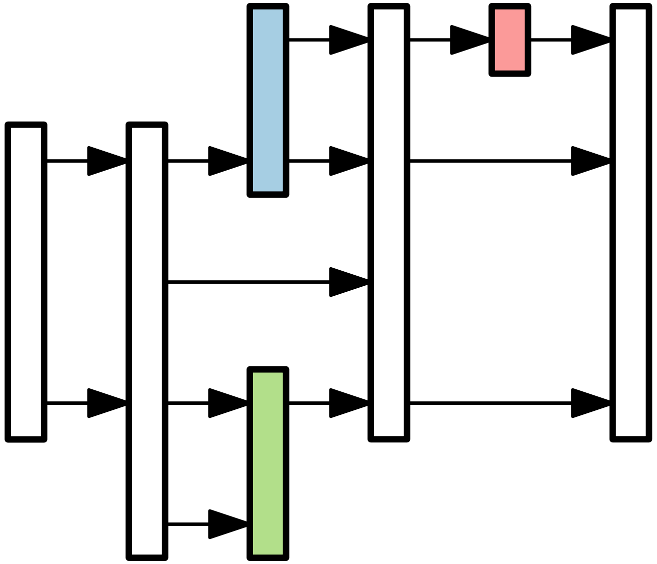
Conflicts



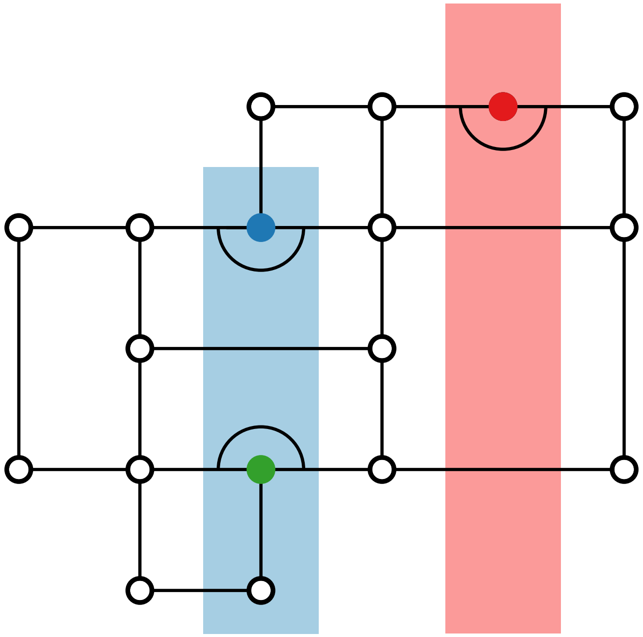


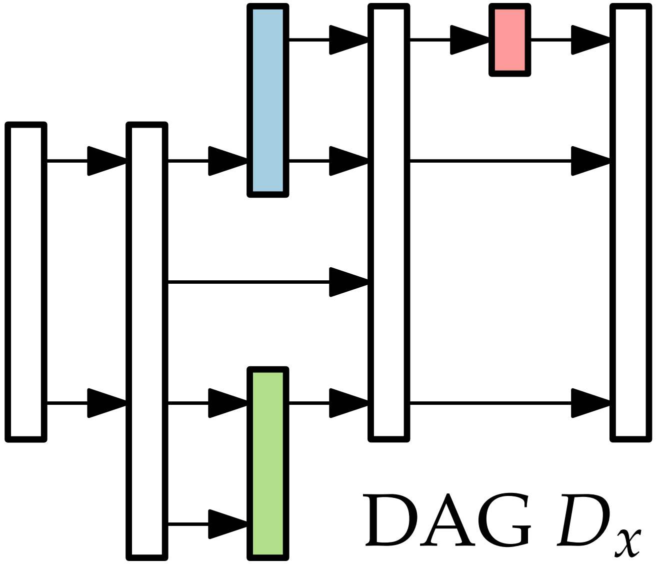
Conflicts





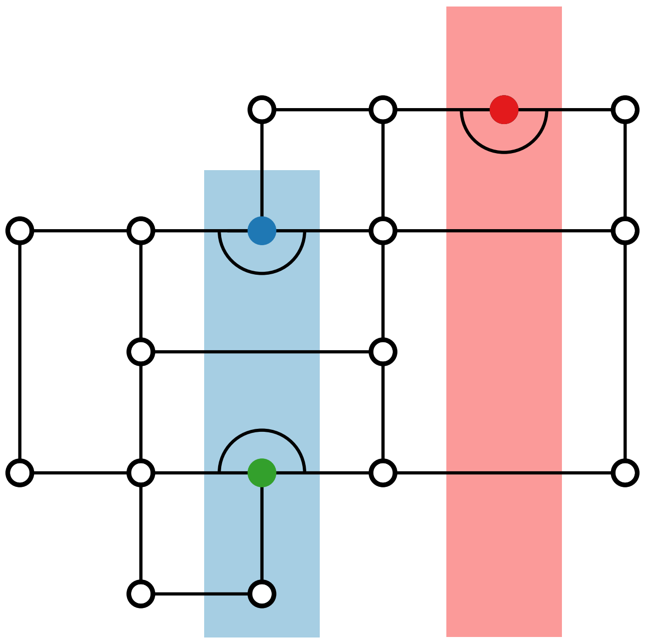
Conflicts

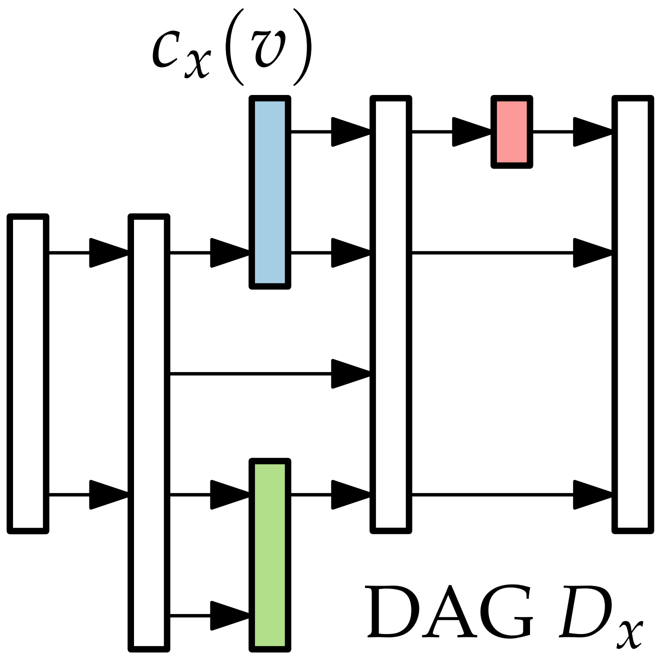




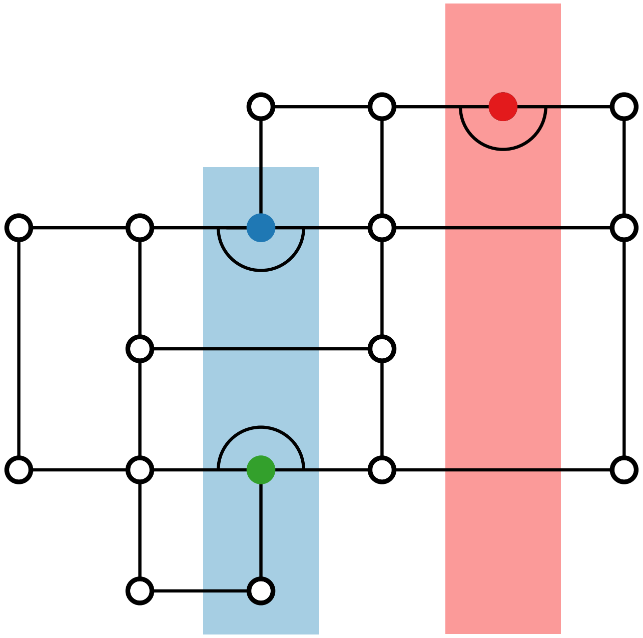
DAG D_x

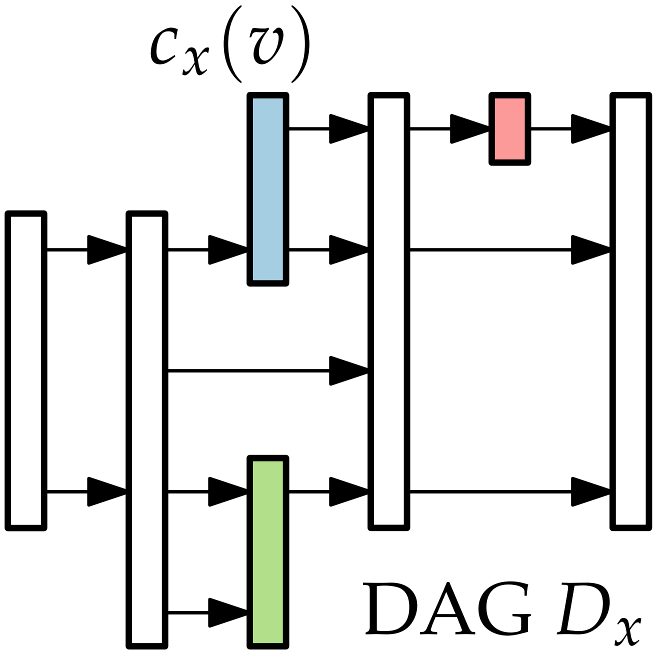
Conflicts

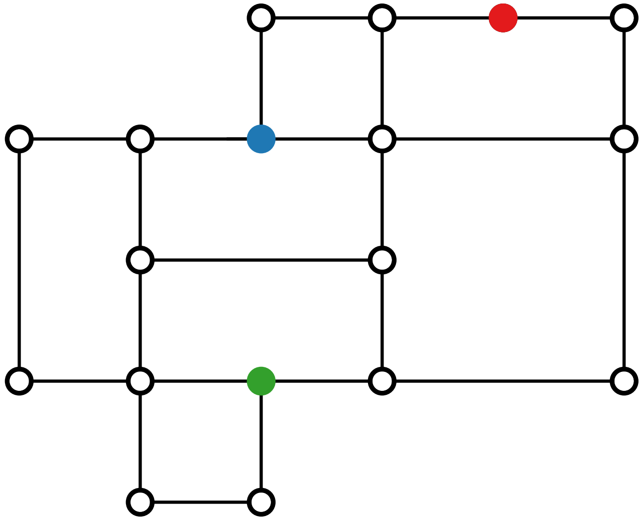




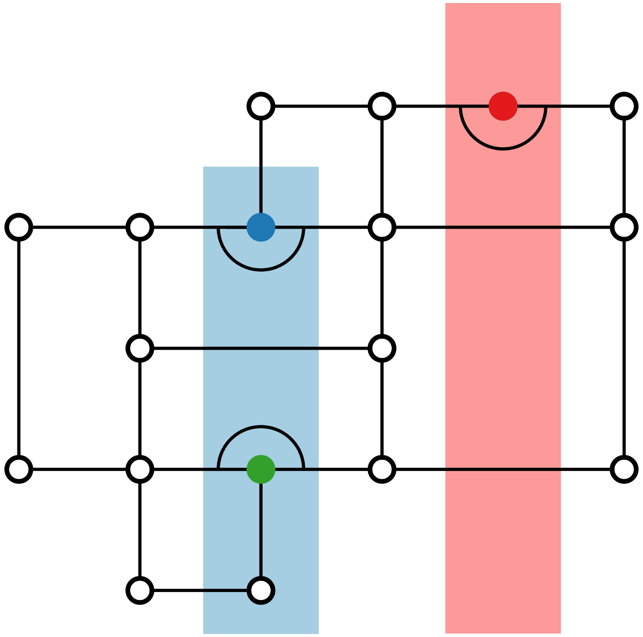
Conflicts

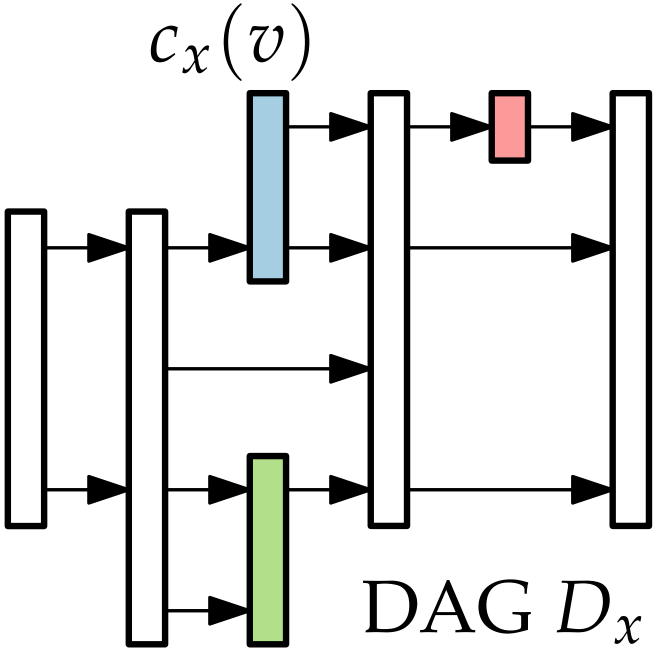


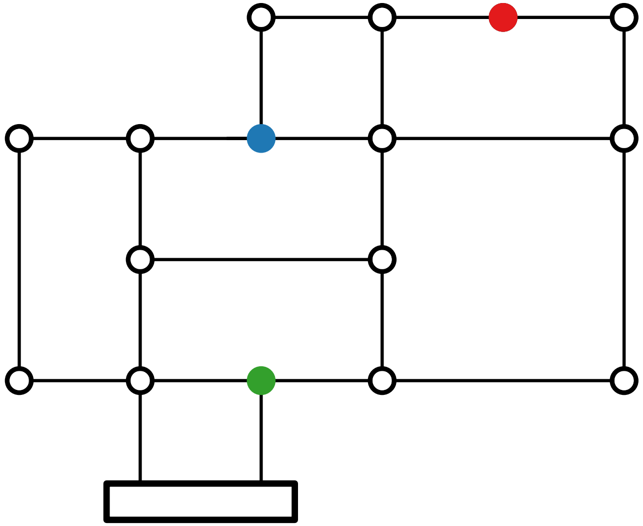




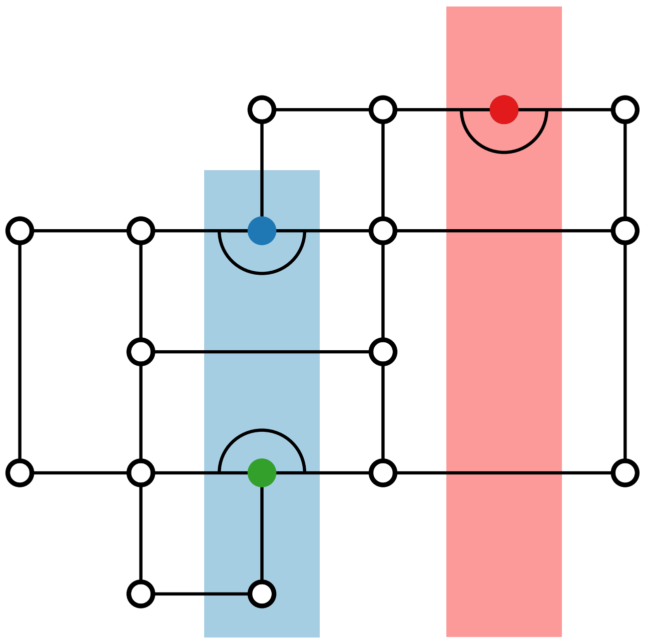
Conflicts

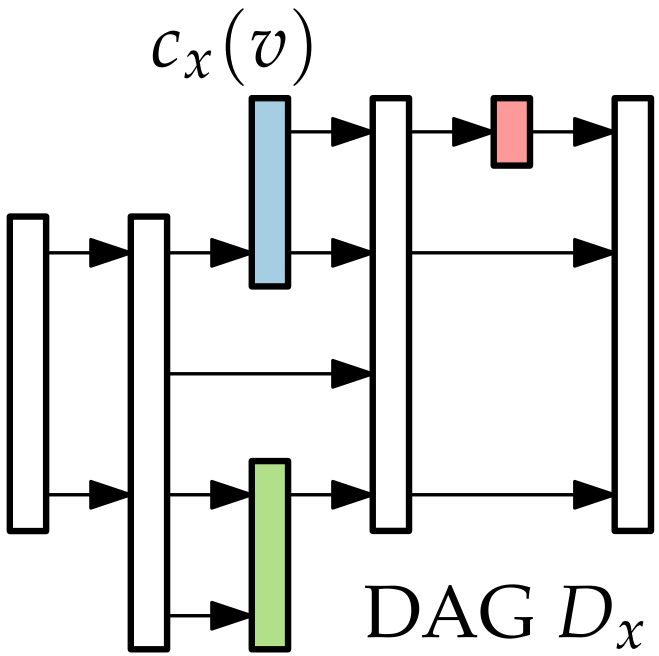


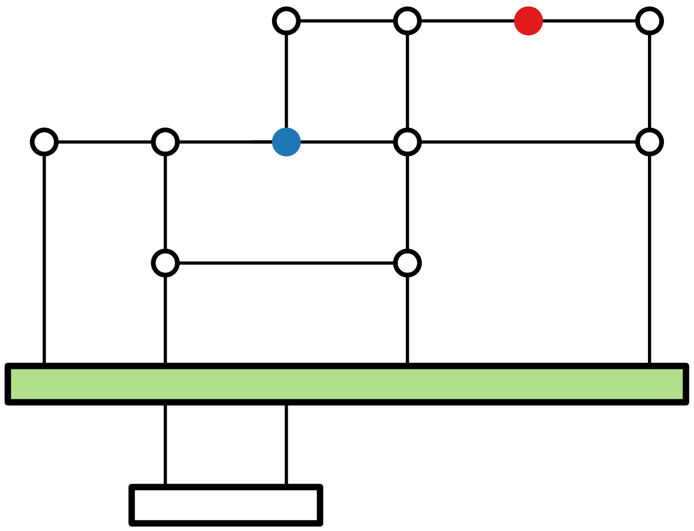




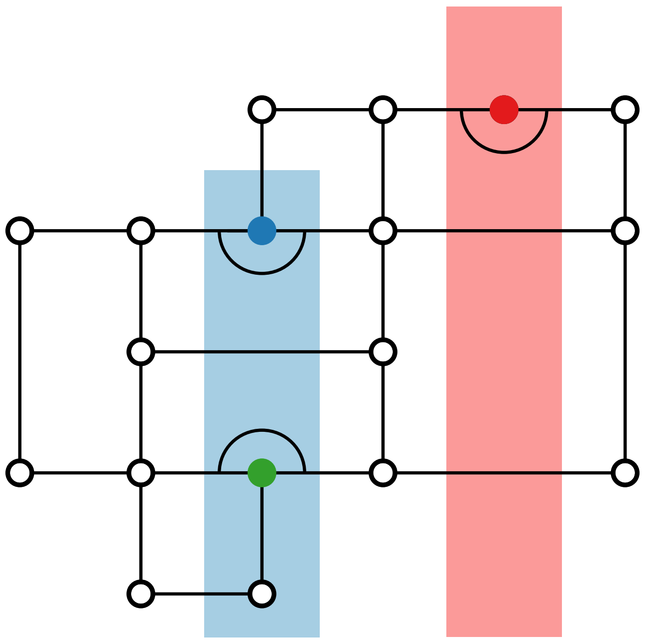
Conflicts

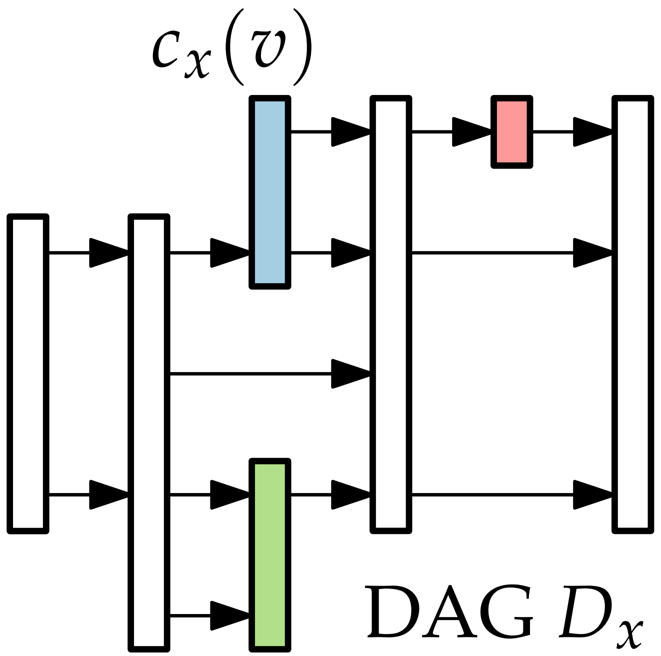


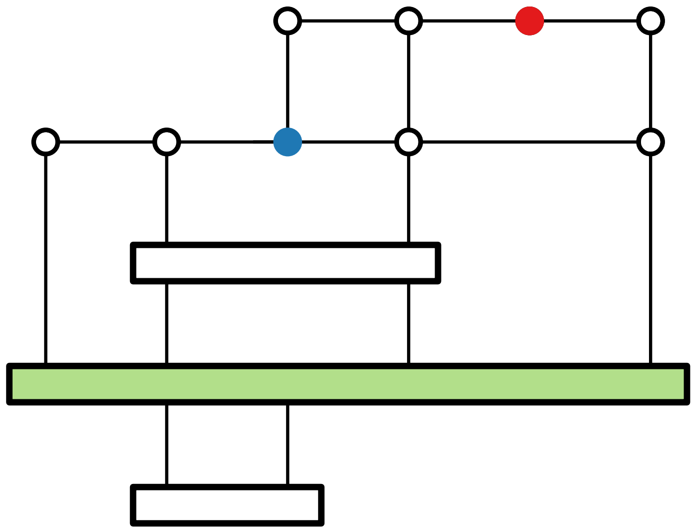




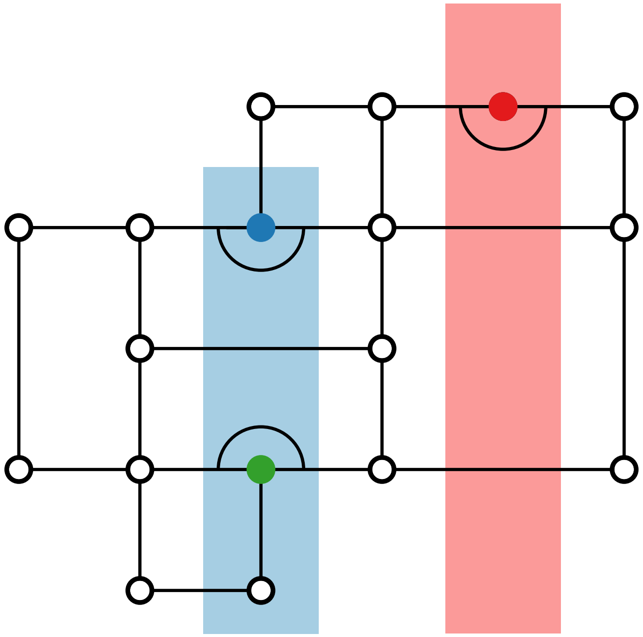
Conflicts

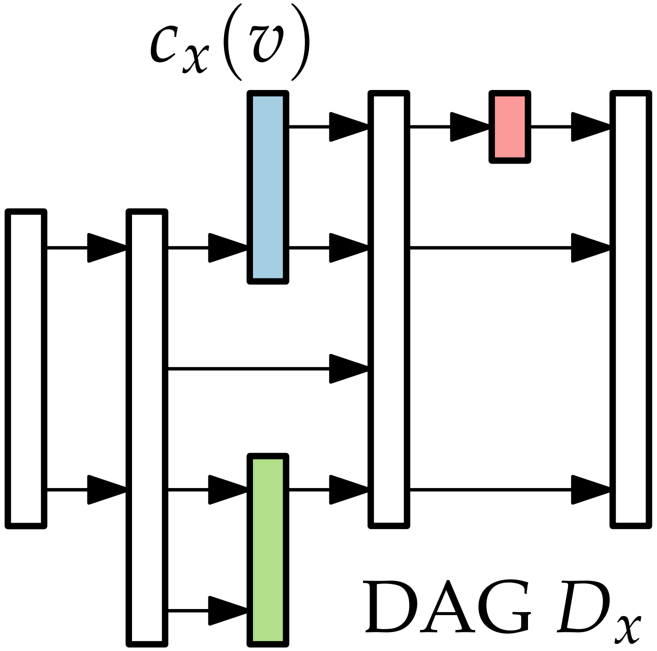


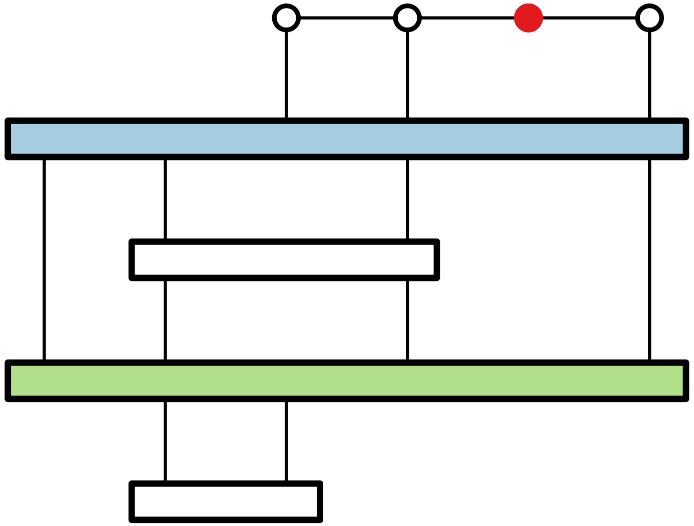




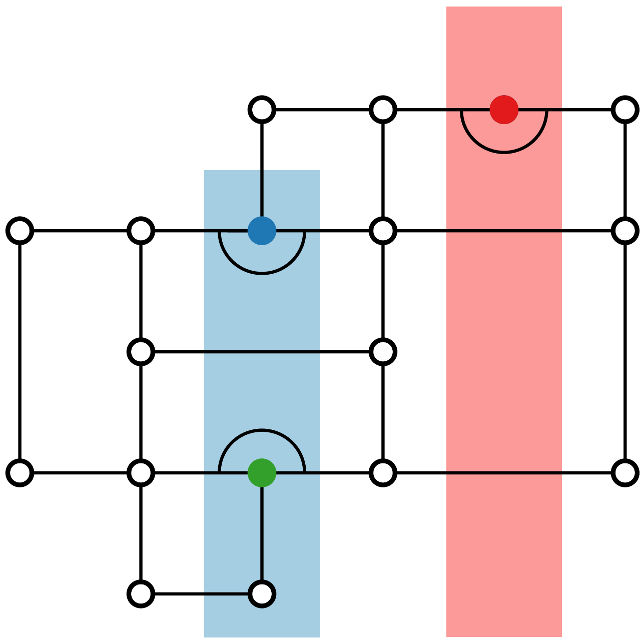
Conflicts

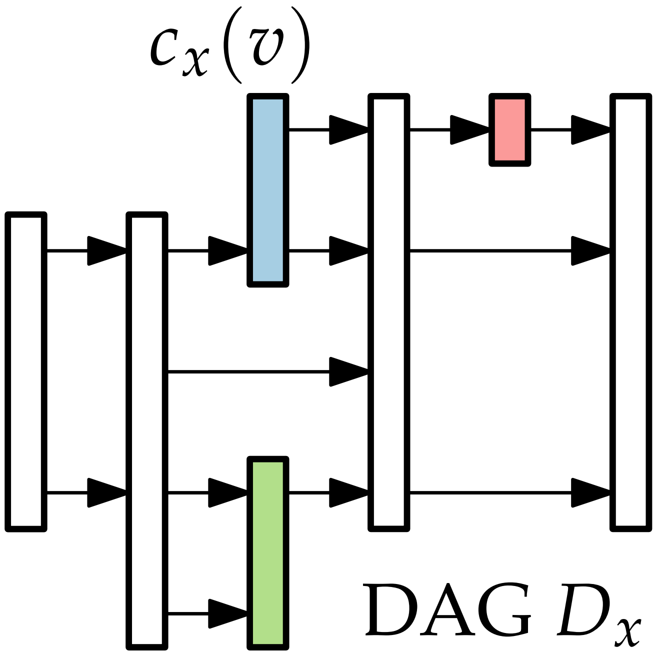


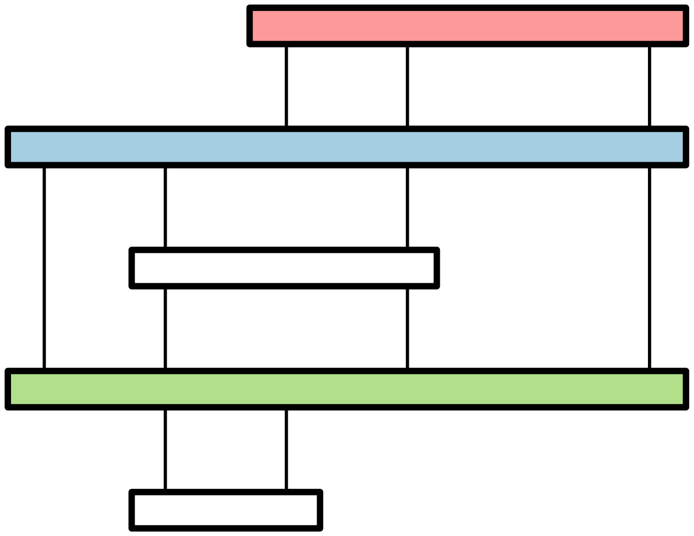




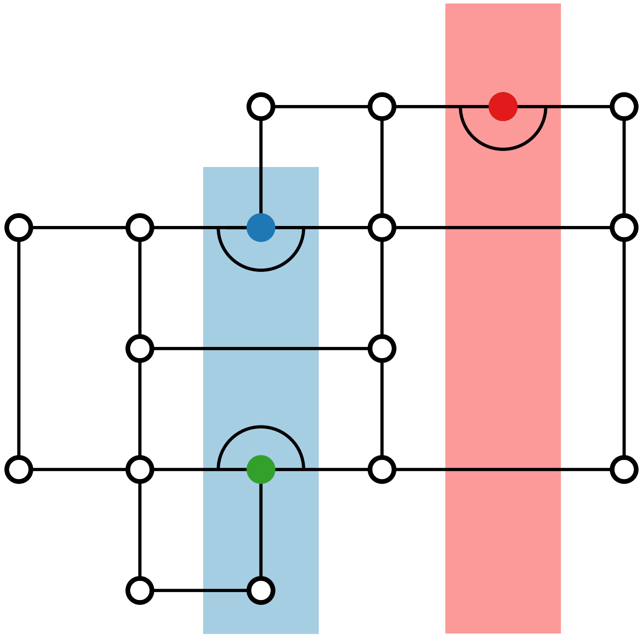
Conflicts

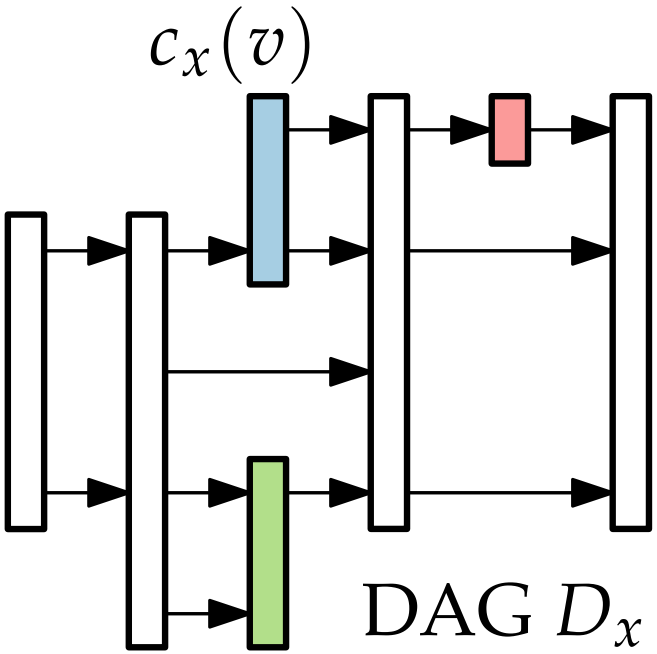


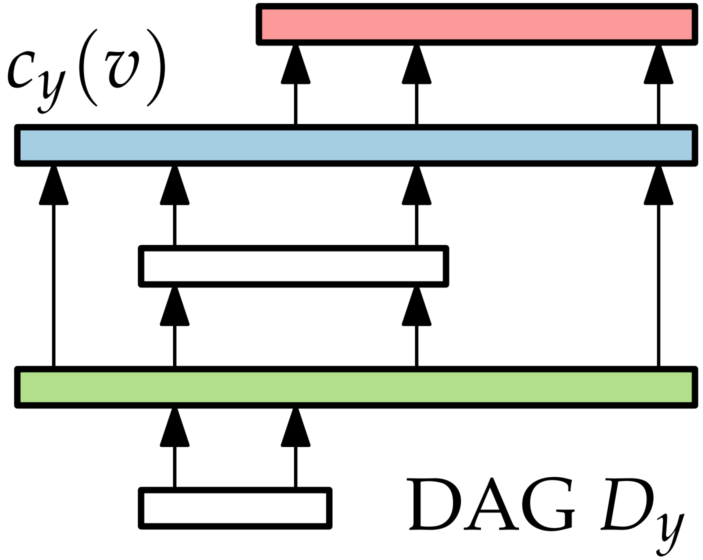




Conflicts

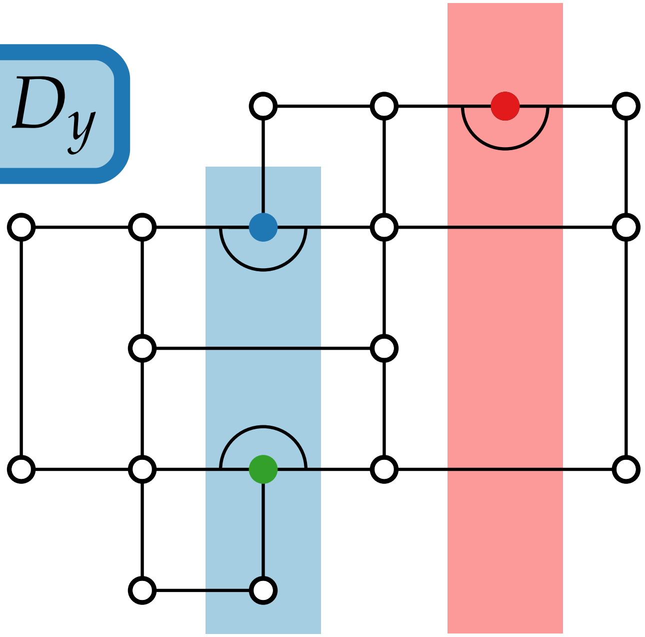


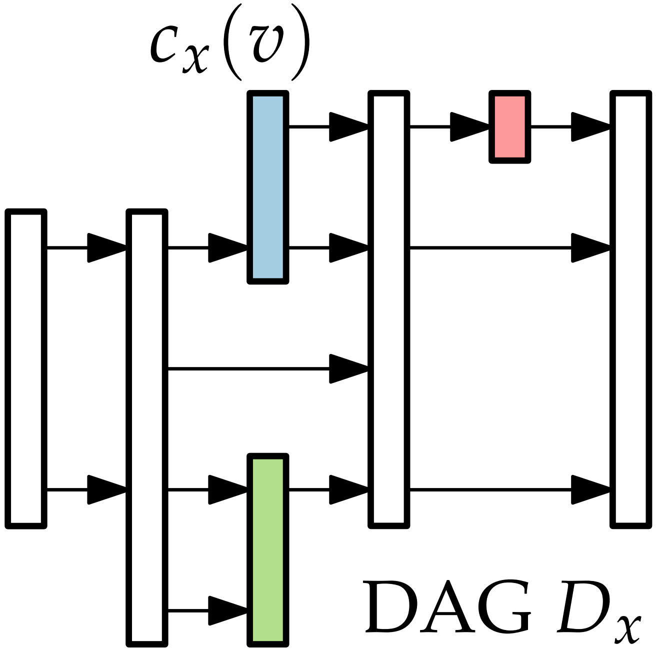


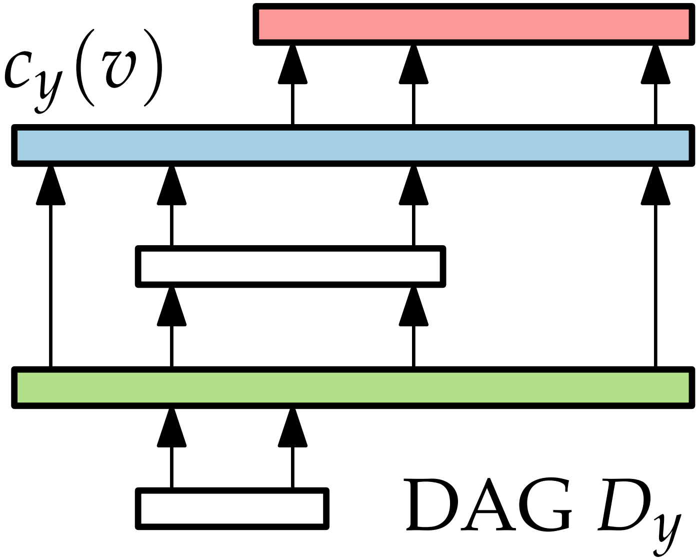


Conflicts

Conflict: no path in D_x or in D_y



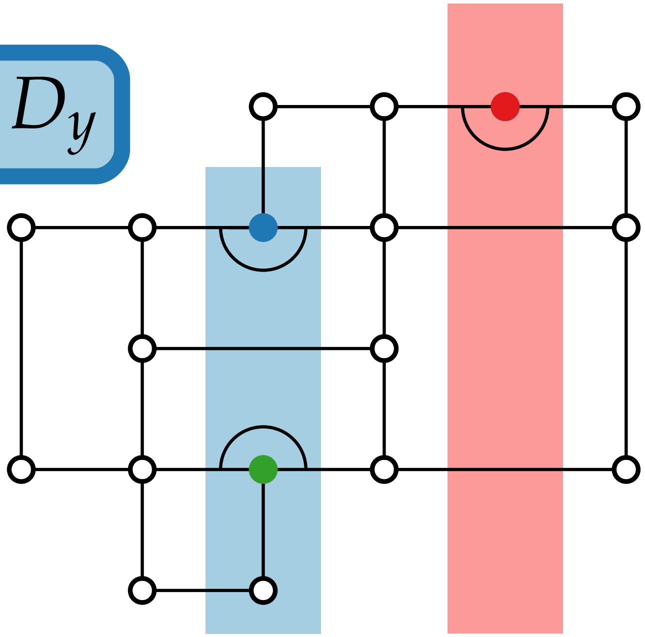


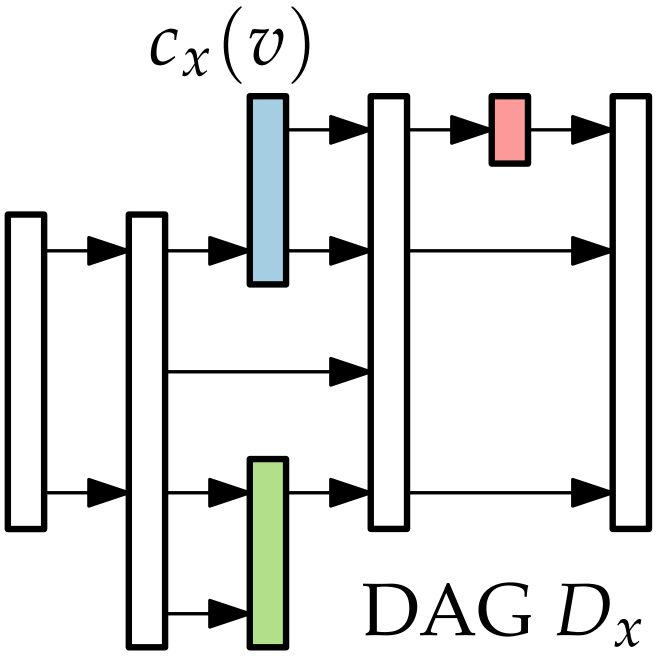


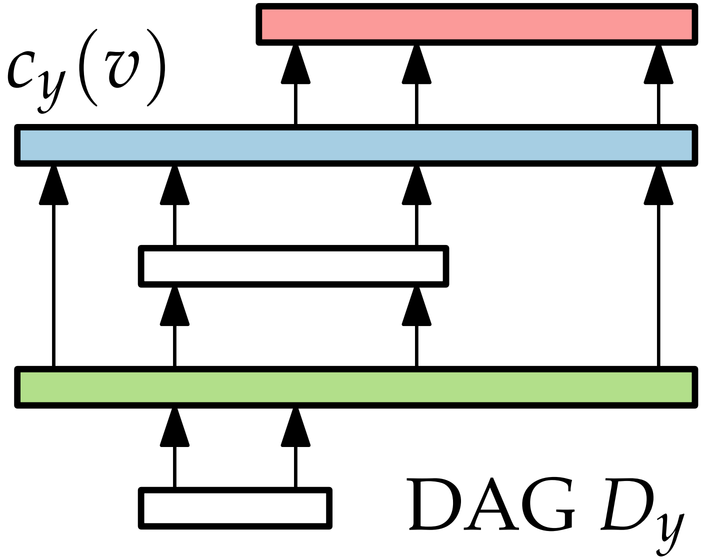
Conflicts

Conflict: no path in D_x or in D_y

either







Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

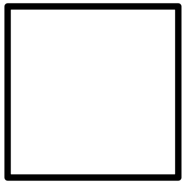
universal greedy ⇔ no conflicts

generative scheme:

Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

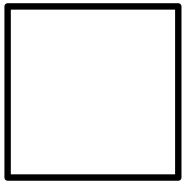
universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

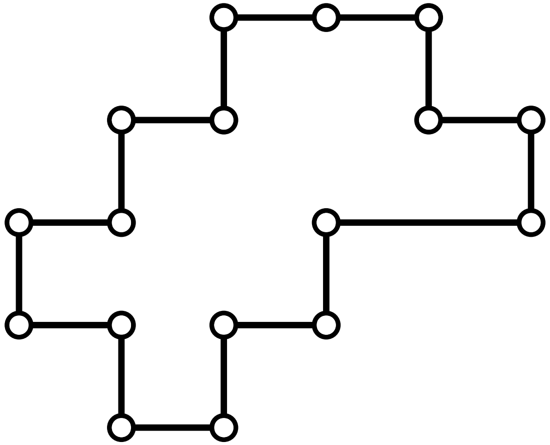
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

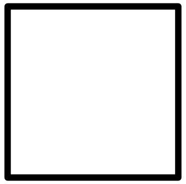
{1,2,3,4}-reflex vertex addition:



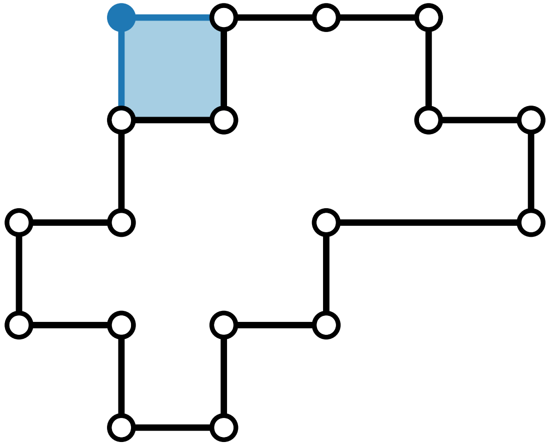
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition:

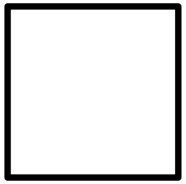


Universal Greedy Rectilinear

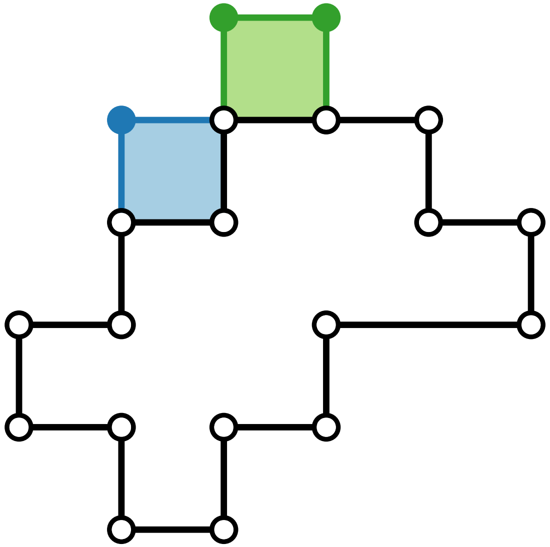
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle



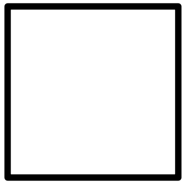
{1,2,3,4}-reflex vertex addition:



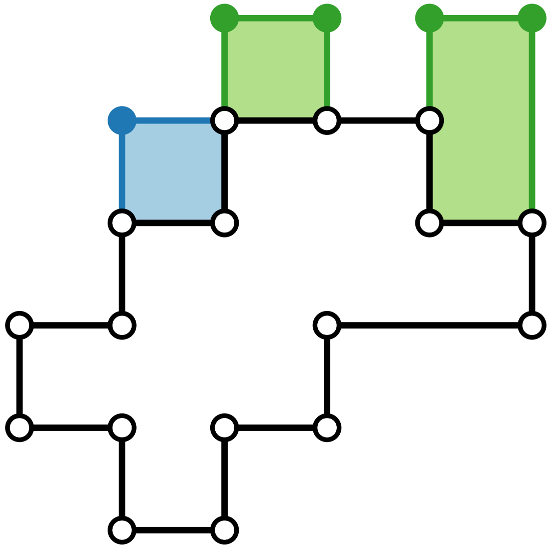
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

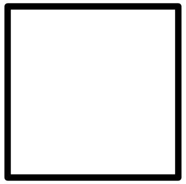
{1,2,3,4}-reflex vertex addition:



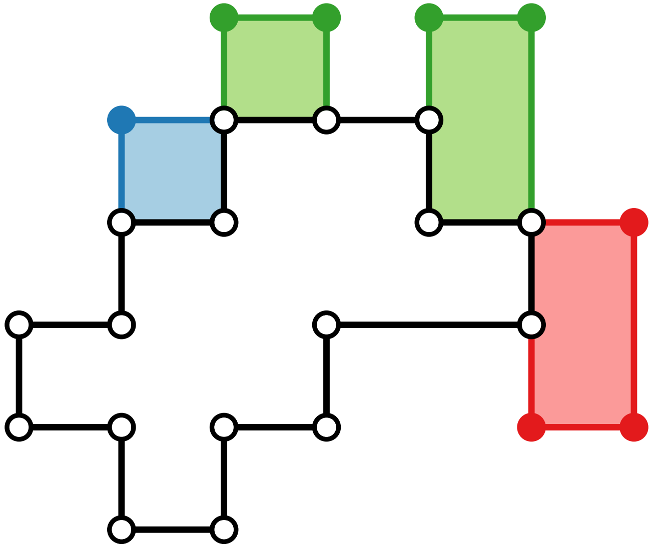
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

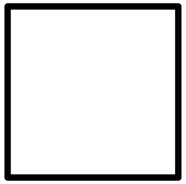
{1,2,3,4}-reflex vertex addition:



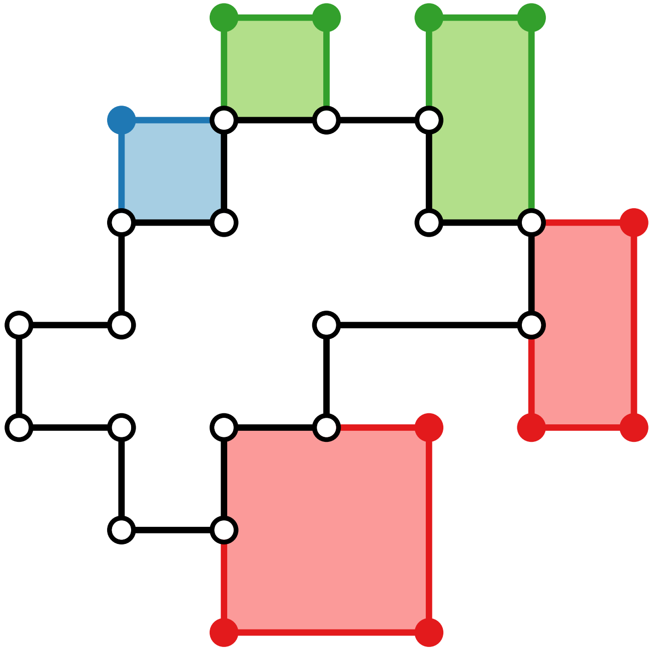
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

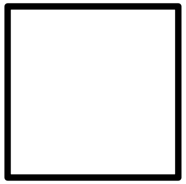
{1,2,3,4}-reflex vertex addition:



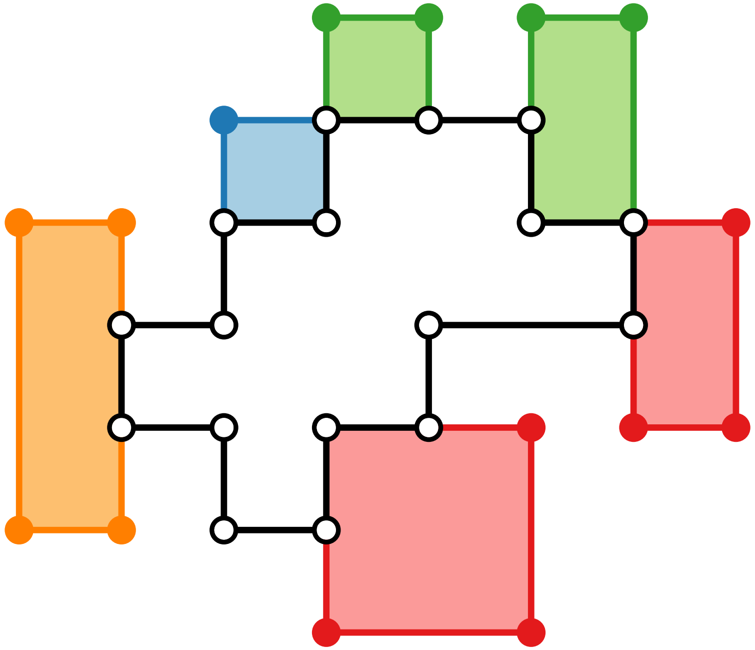
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

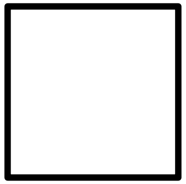
{1,2,3,4}-reflex vertex addition:



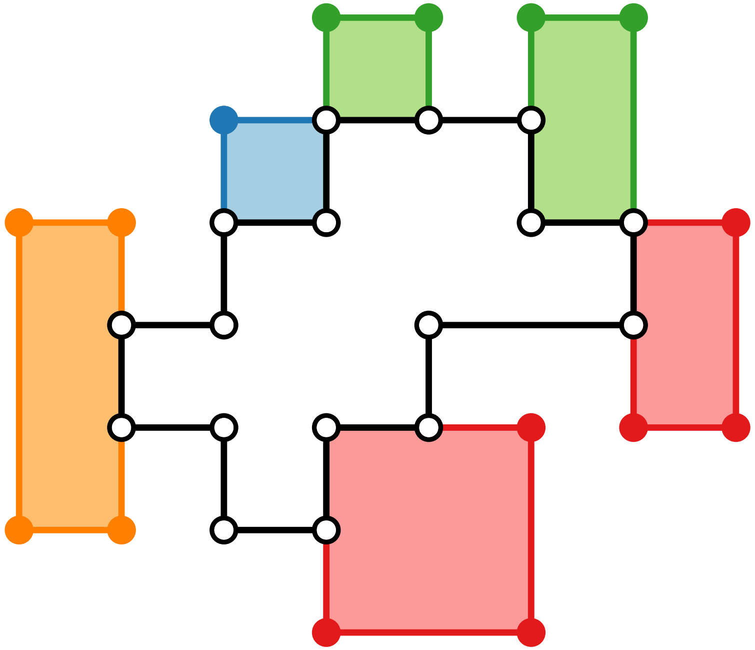
Universal Greedy Rectilinear

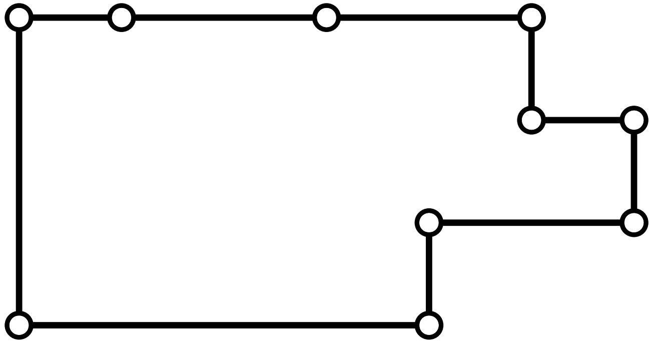
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

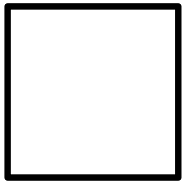




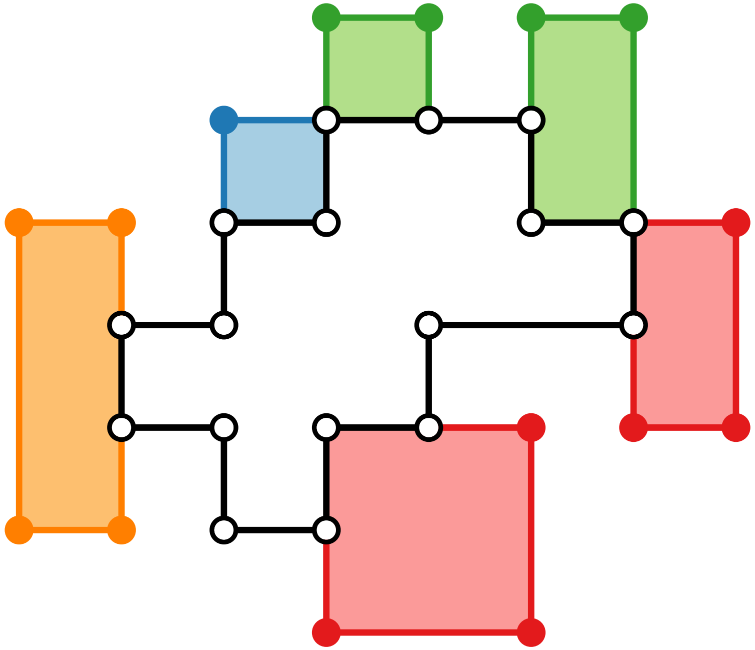
Universal Greedy Rectilinear

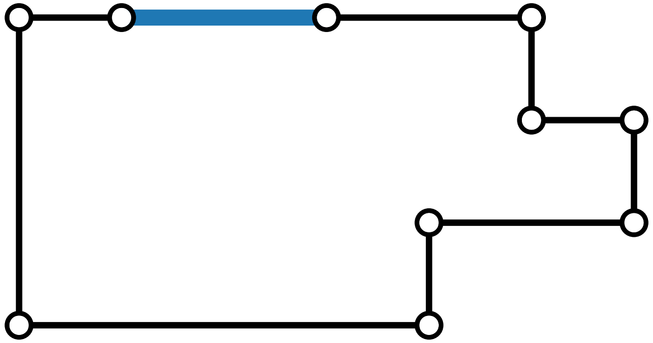
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

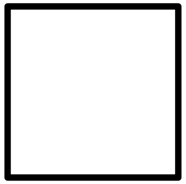




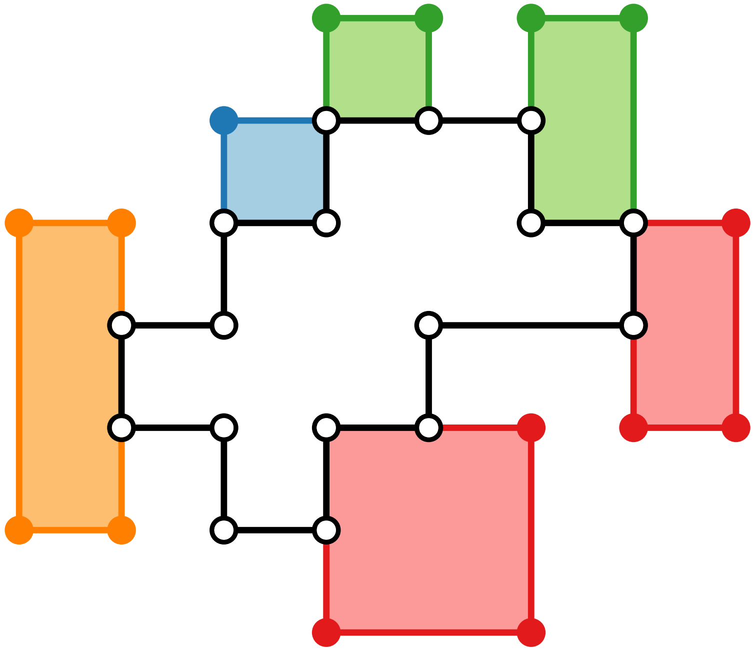
Universal Greedy Rectilinear

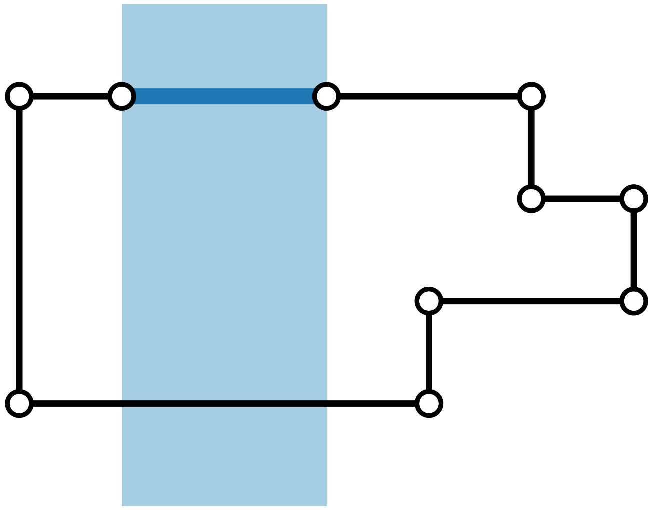
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

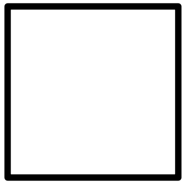




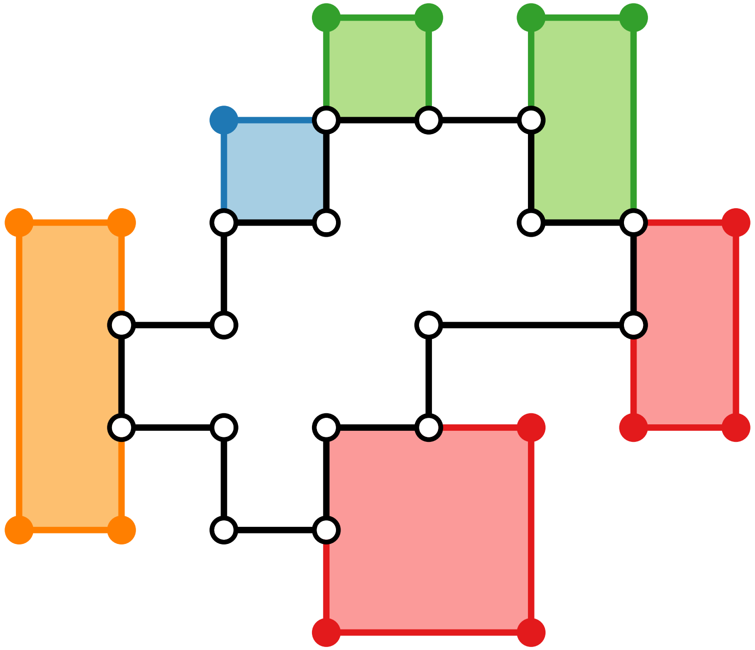
Universal Greedy Rectilinear

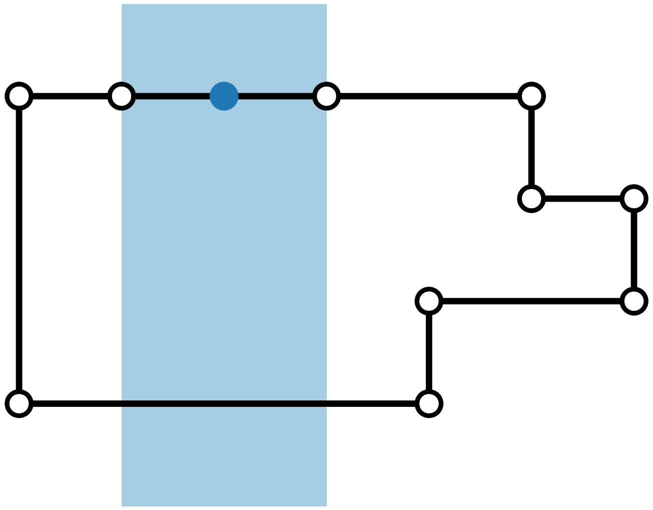
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

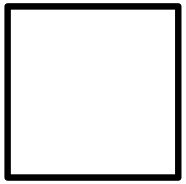




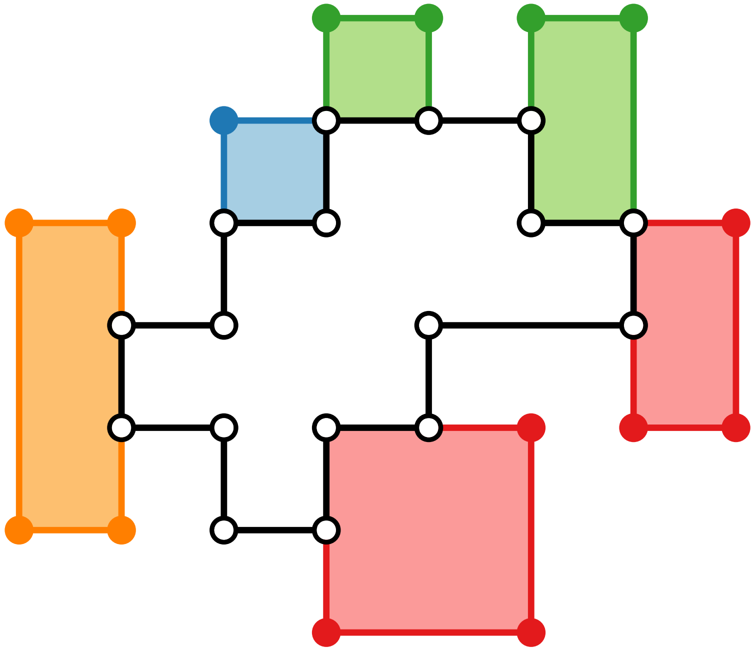
Universal Greedy Rectilinear

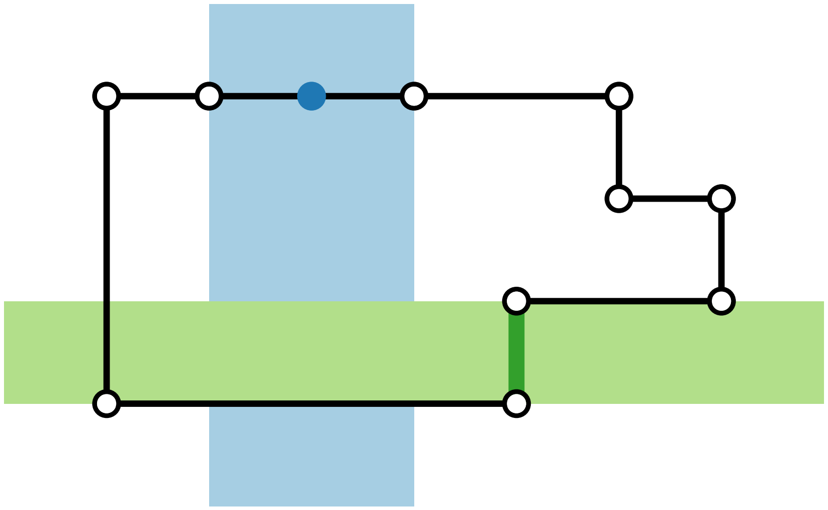
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

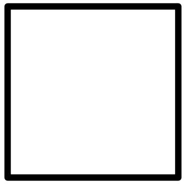




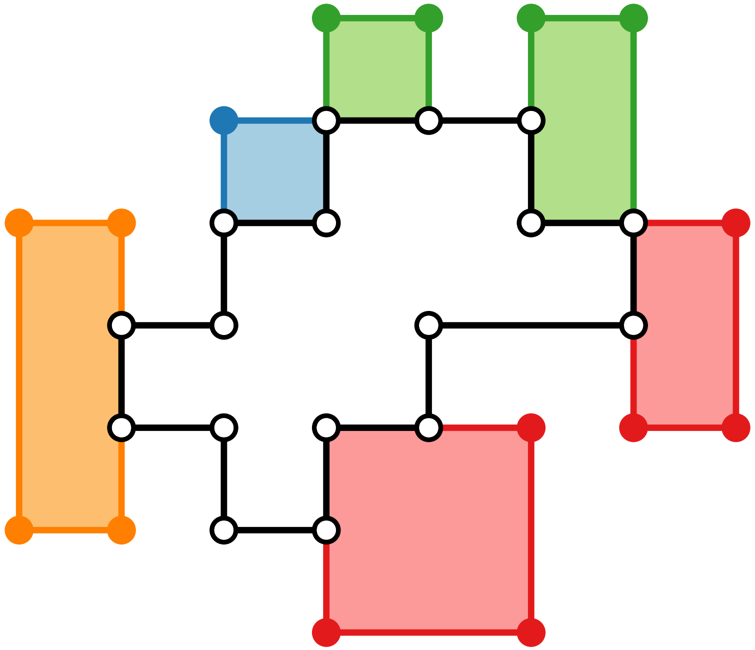
Universal Greedy Rectilinear

universal greedy: *every* drawing is greedy

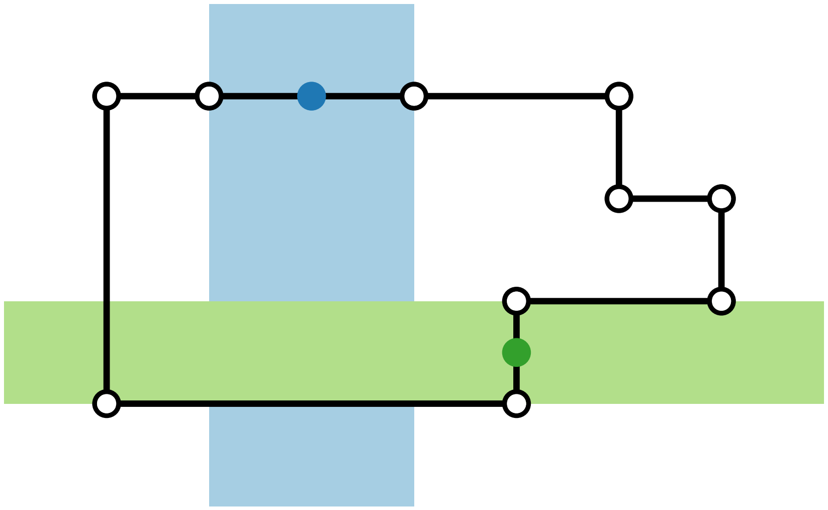
universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition:



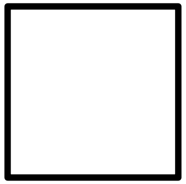
flat vertex addition:



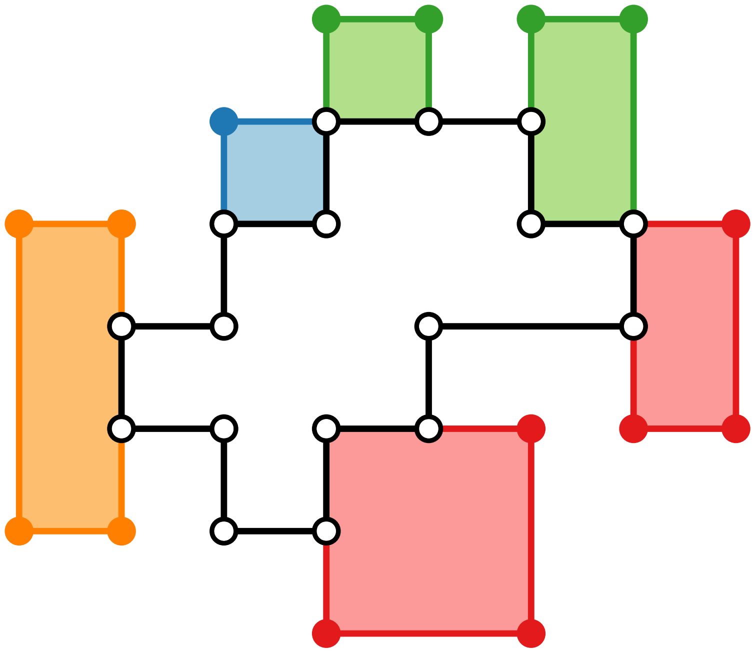
Universal Greedy Rectilinear

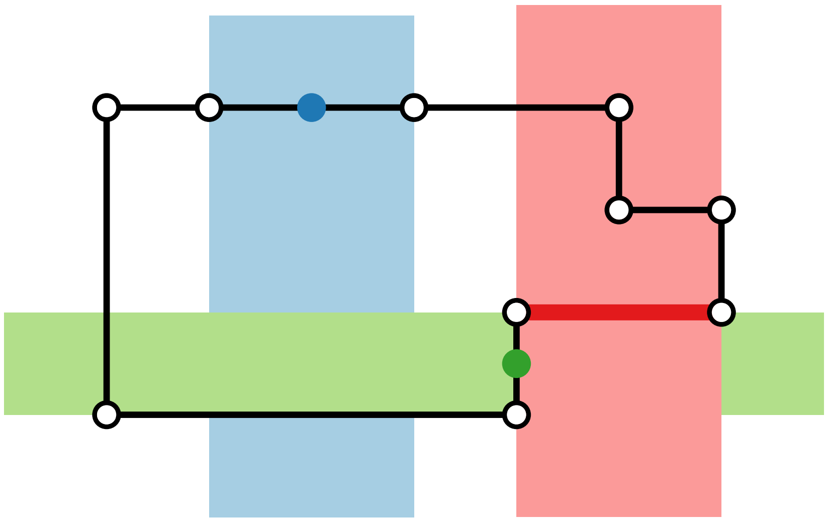
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

{1,2,3,4}-reflex vertex addition: flat vertex addition:

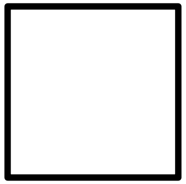




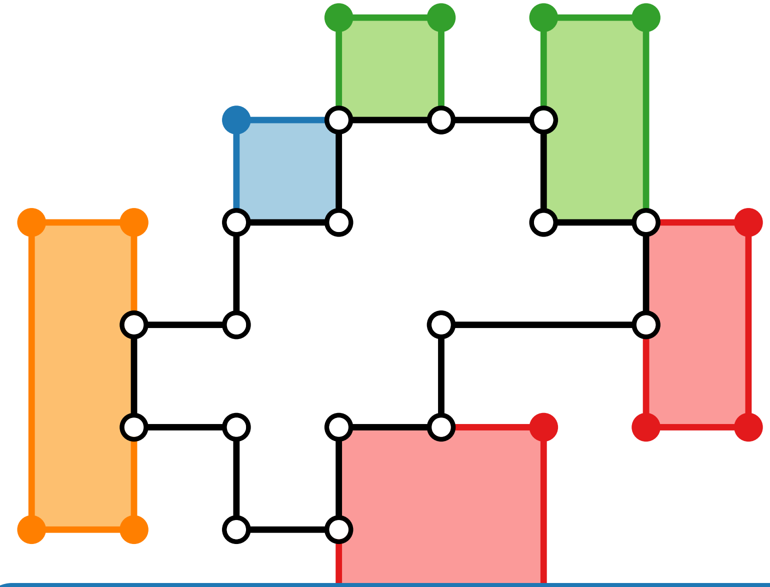
Universal Greedy Rectilinear

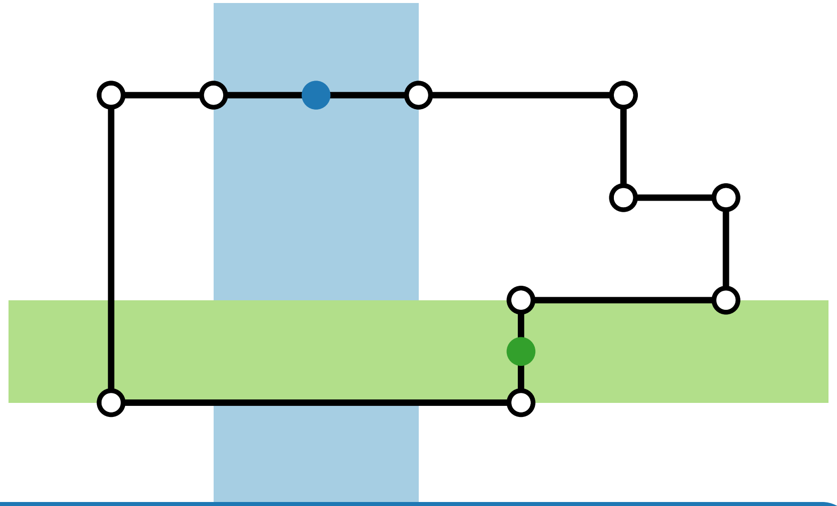
universal greedy: *every* drawing is greedy

universal greedy ⇔ no conflicts

generative scheme: start with rectangle 

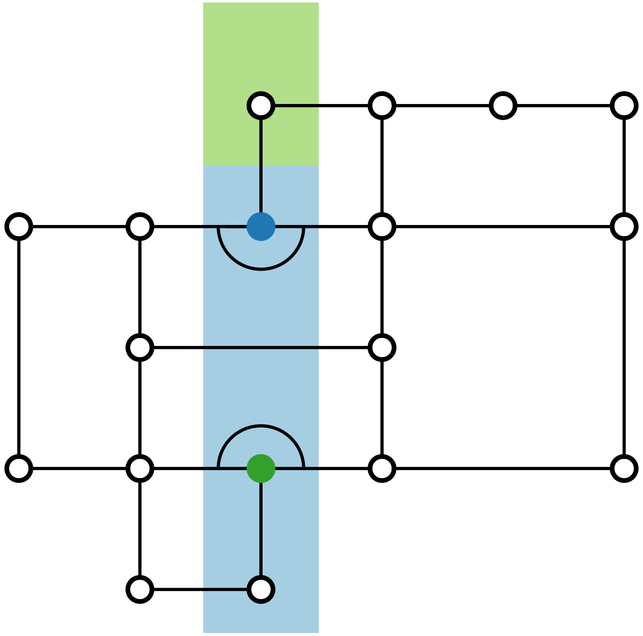
{1,2,3,4}-reflex vertex addition: flat vertex addition:



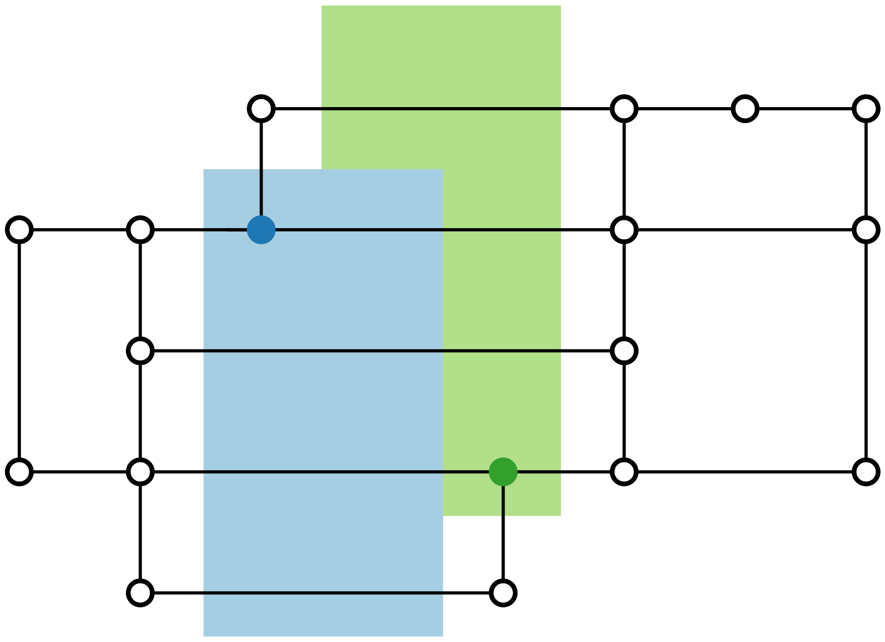


All universal greedy rectilinear graphs can be generated

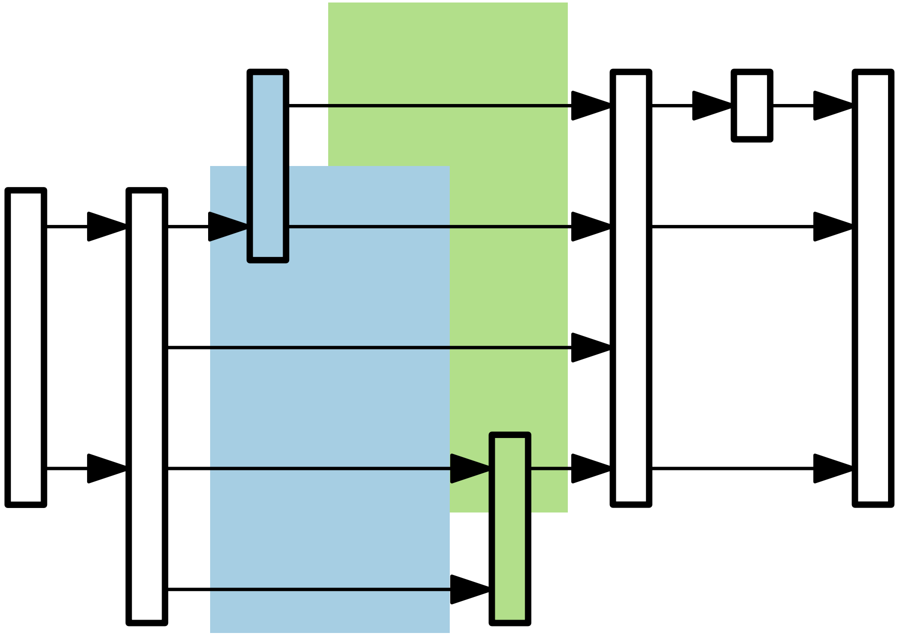
Resolving Conflicts



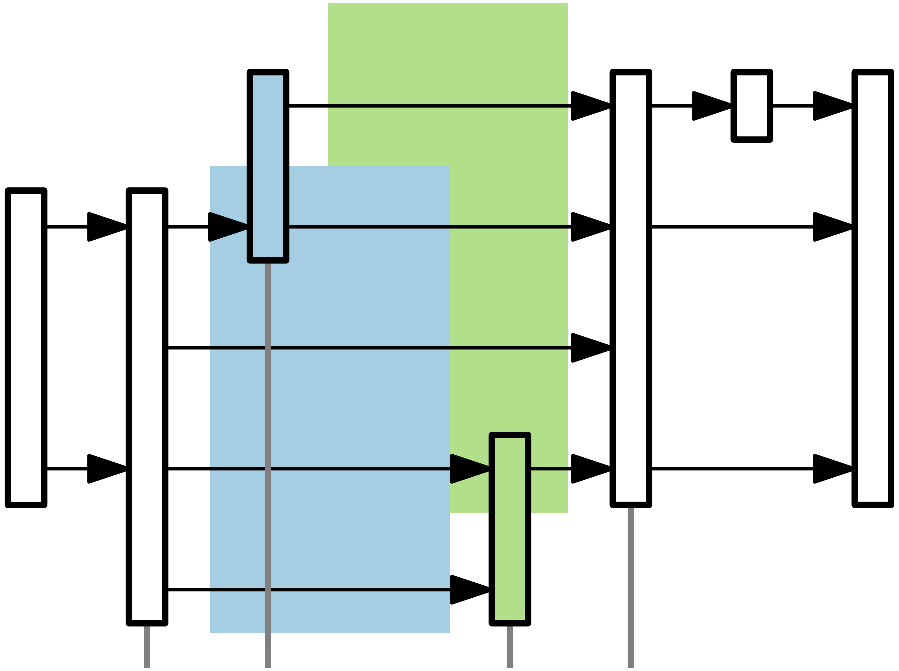
Resolving Conflicts



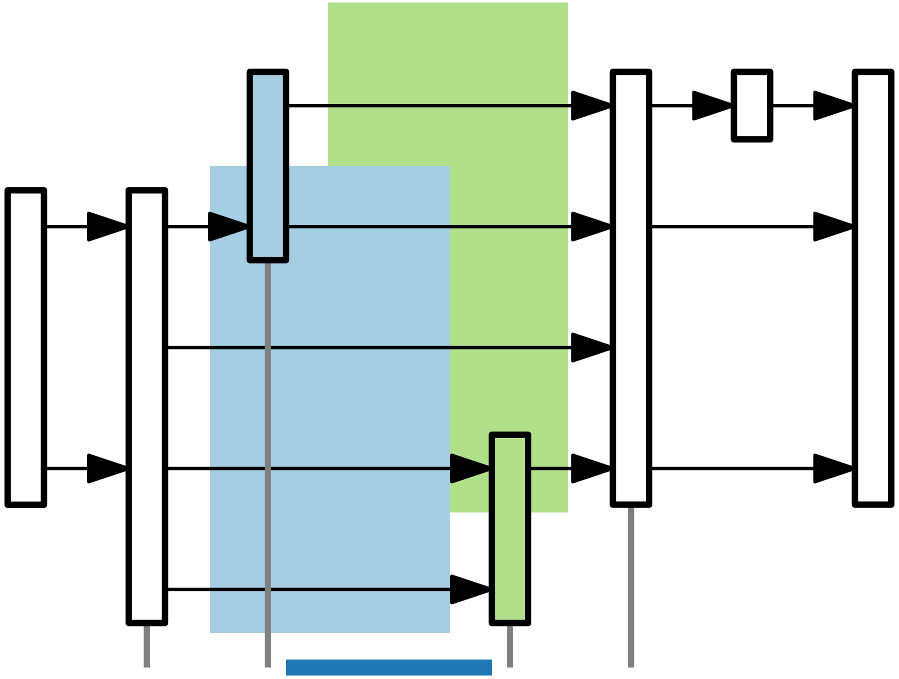
Resolving Conflicts



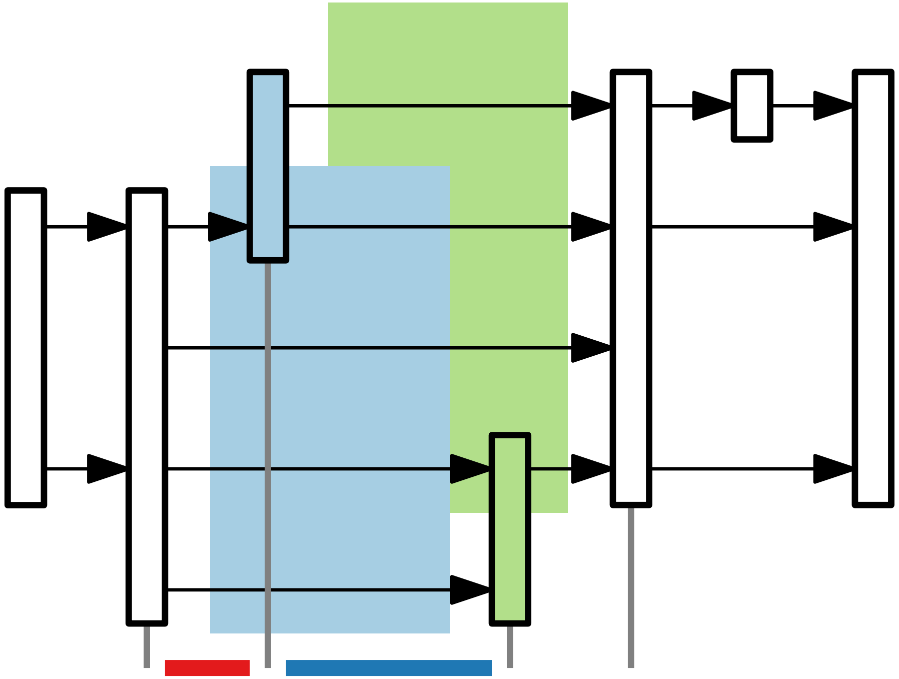
Resolving Conflicts



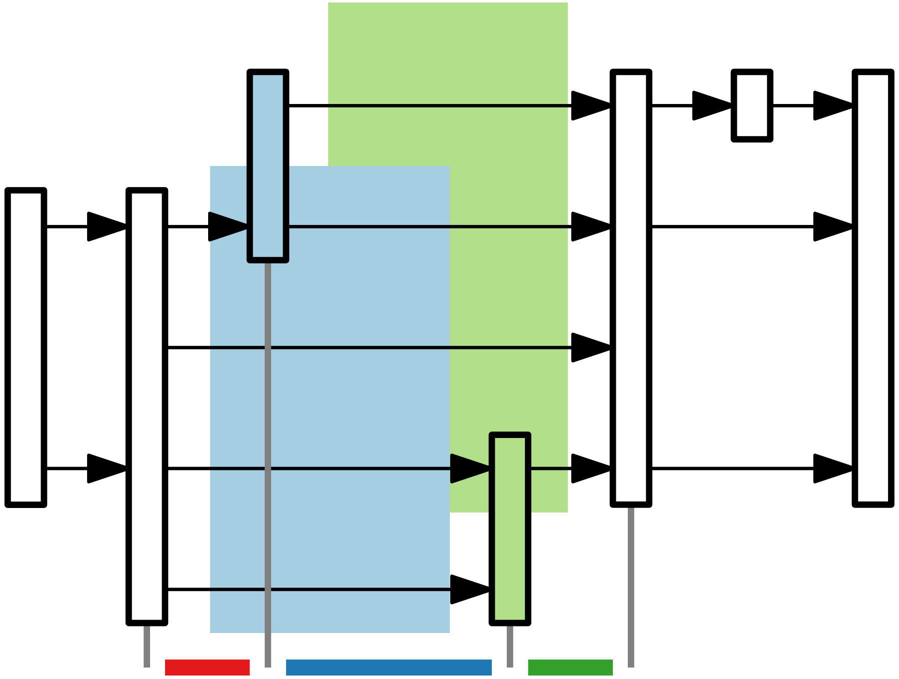
Resolving Conflicts



Resolving Conflicts

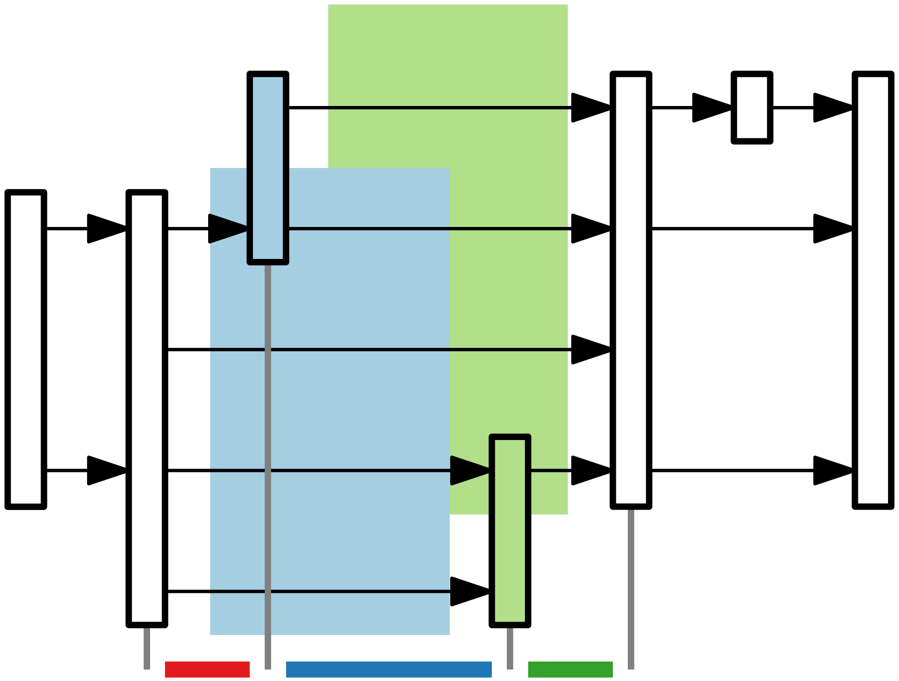


Resolving Conflicts



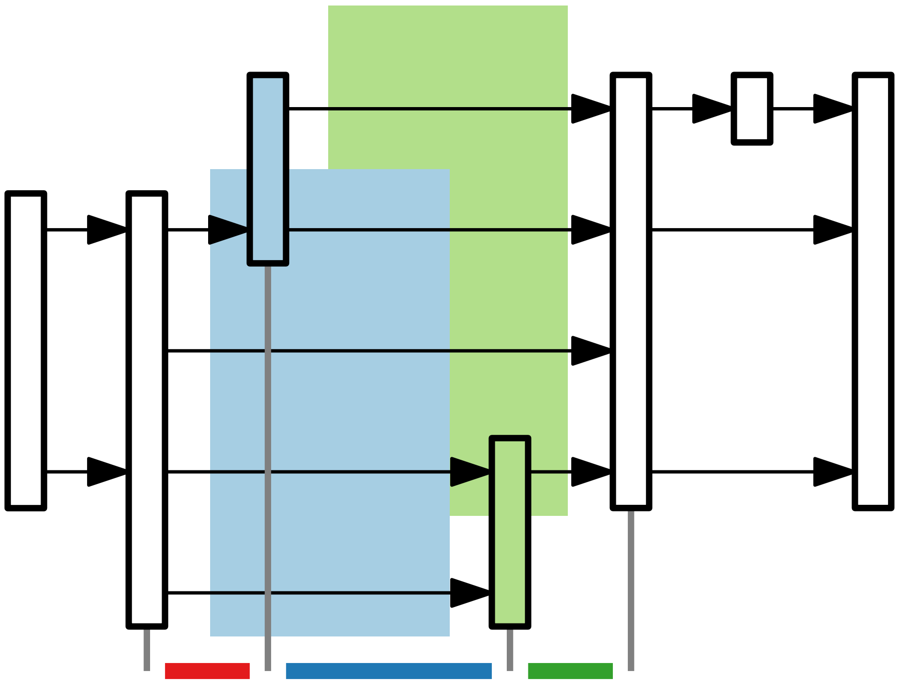
Resolving Conflicts

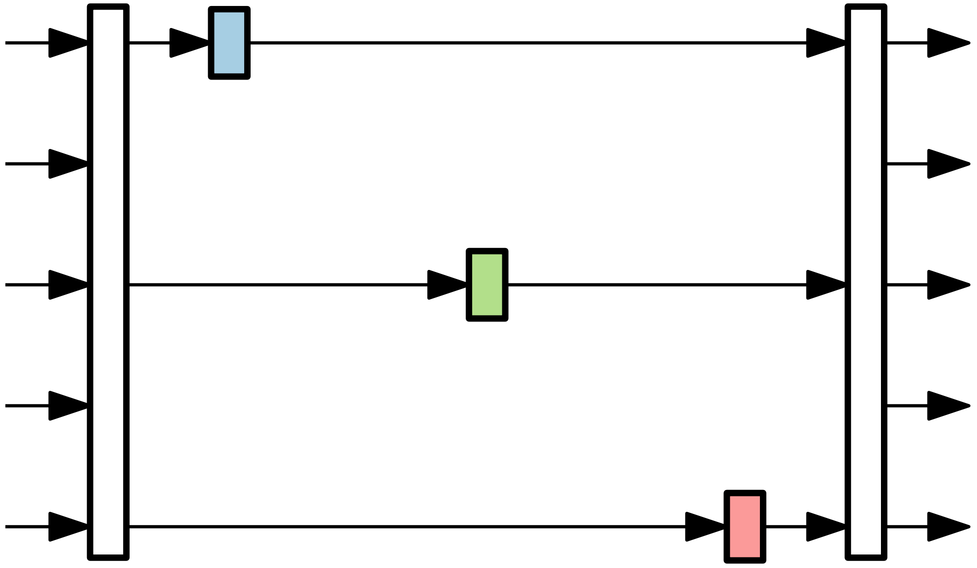
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩



Resolving Conflicts

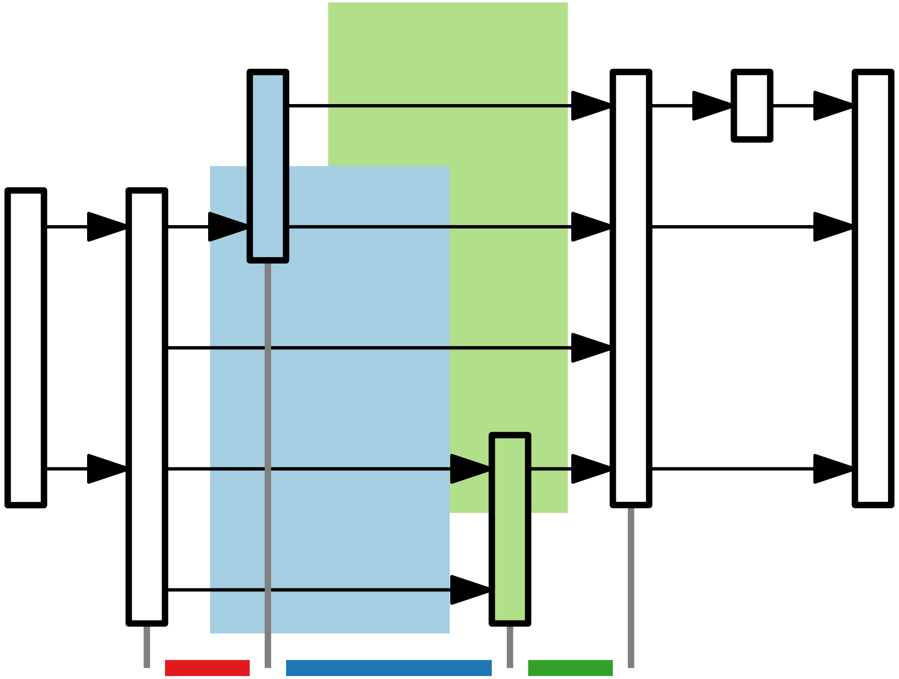
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

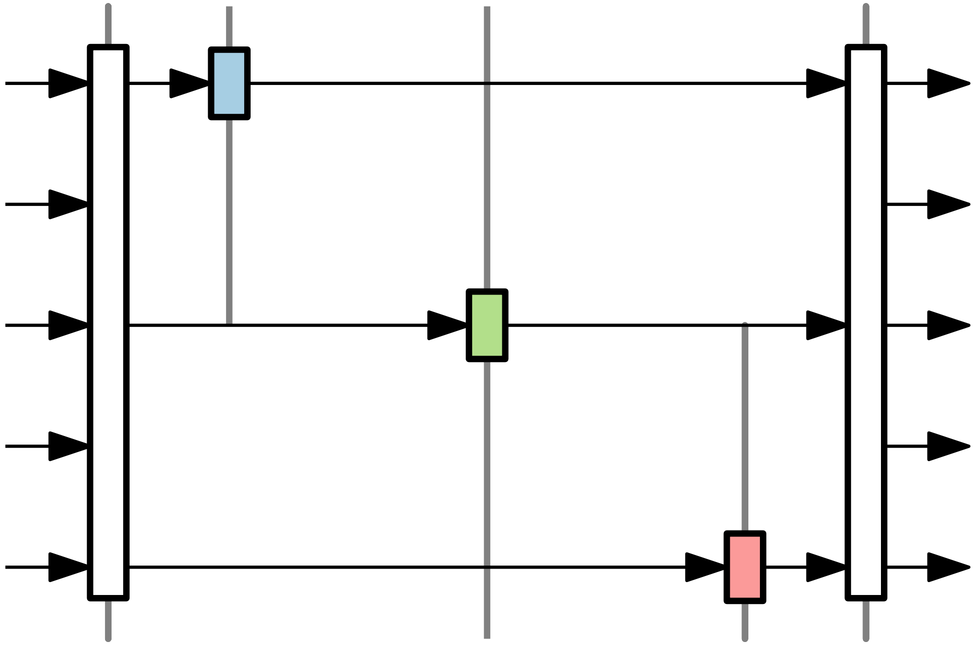




Resolving Conflicts

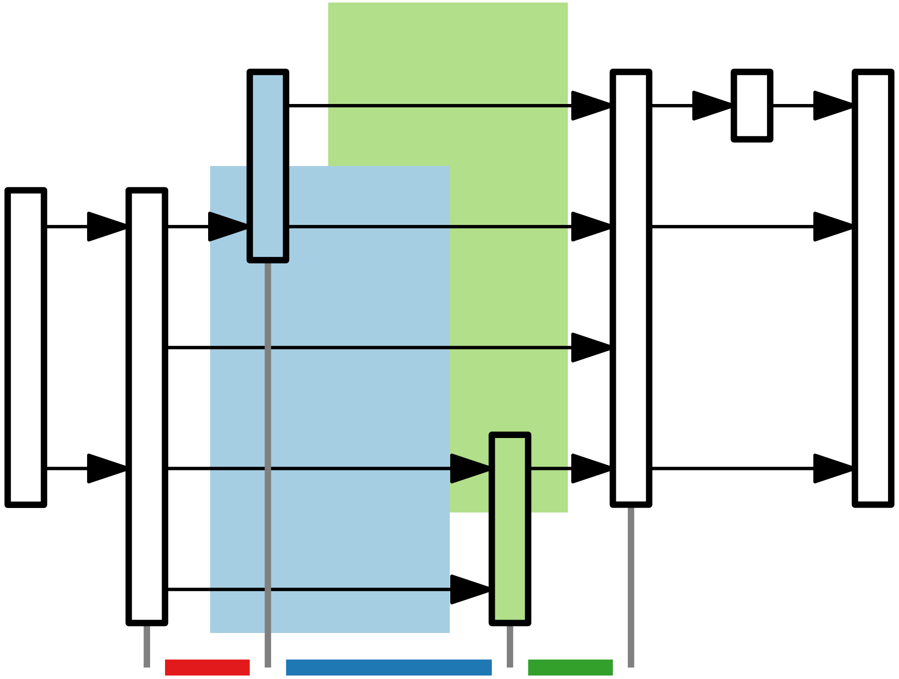
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

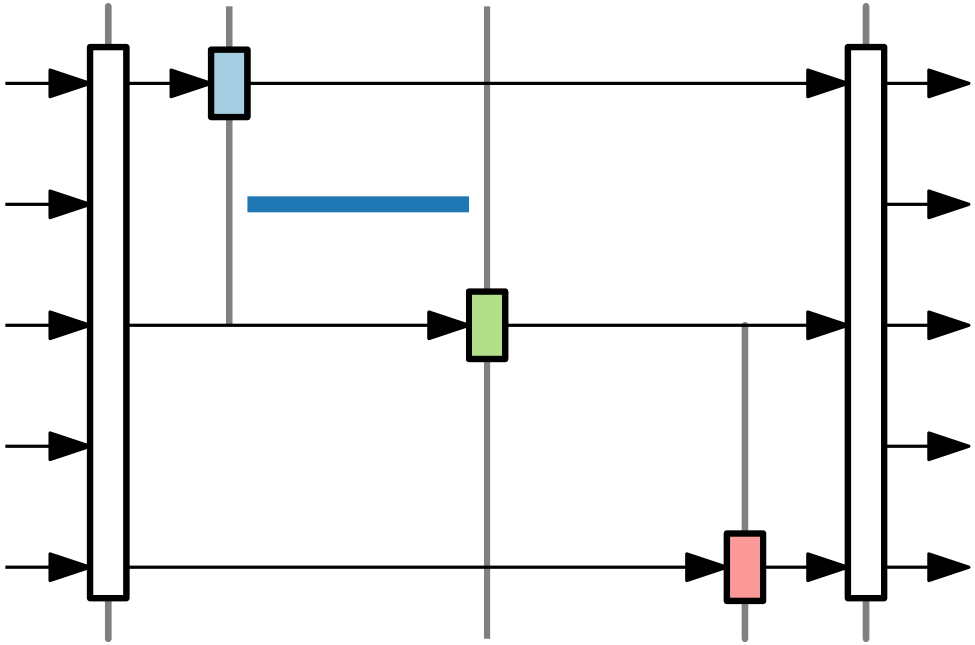




Resolving Conflicts

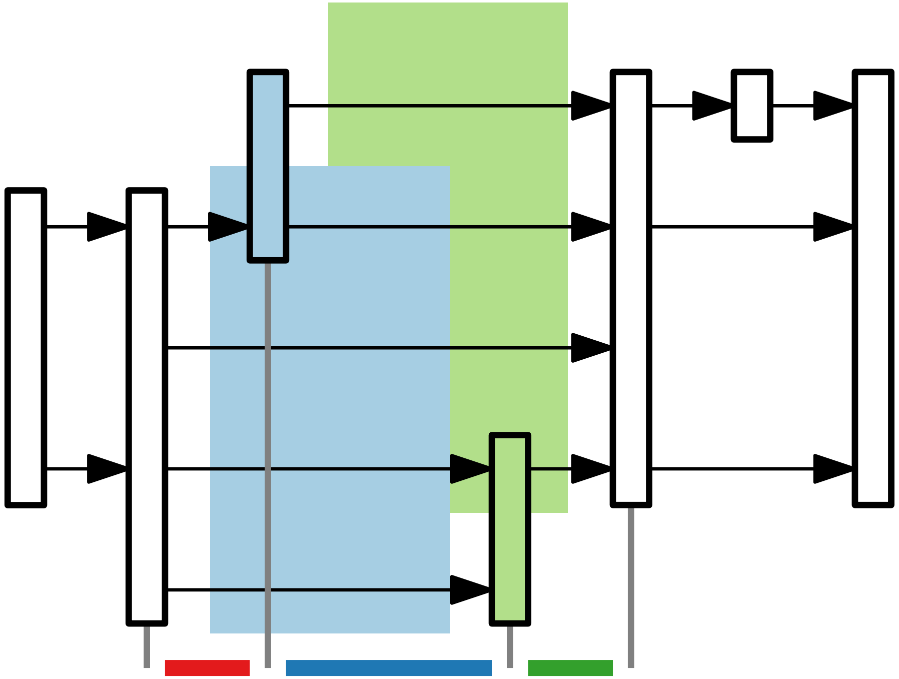
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

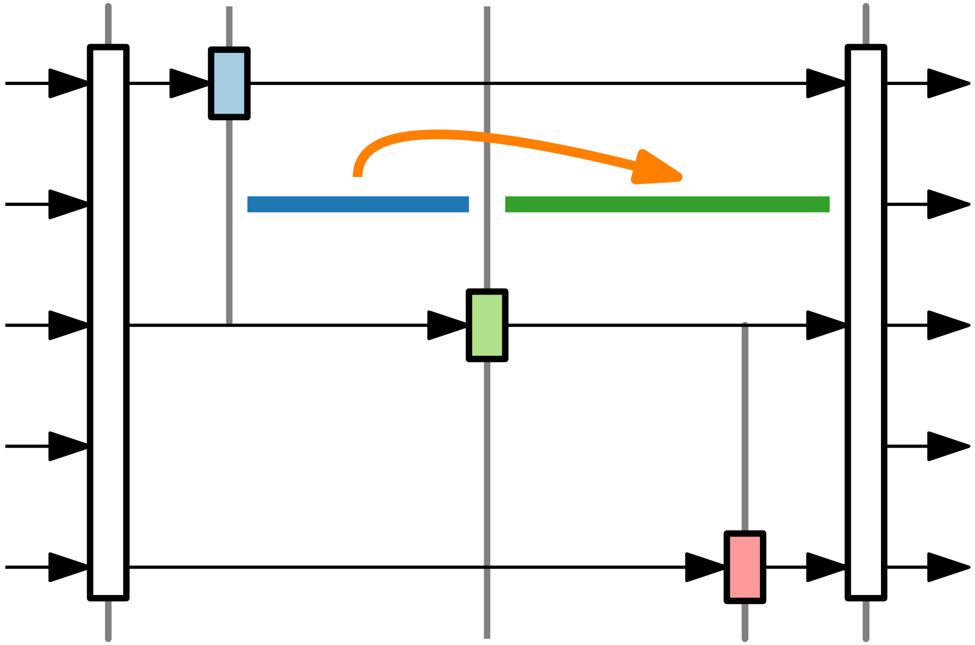




Resolving Conflicts

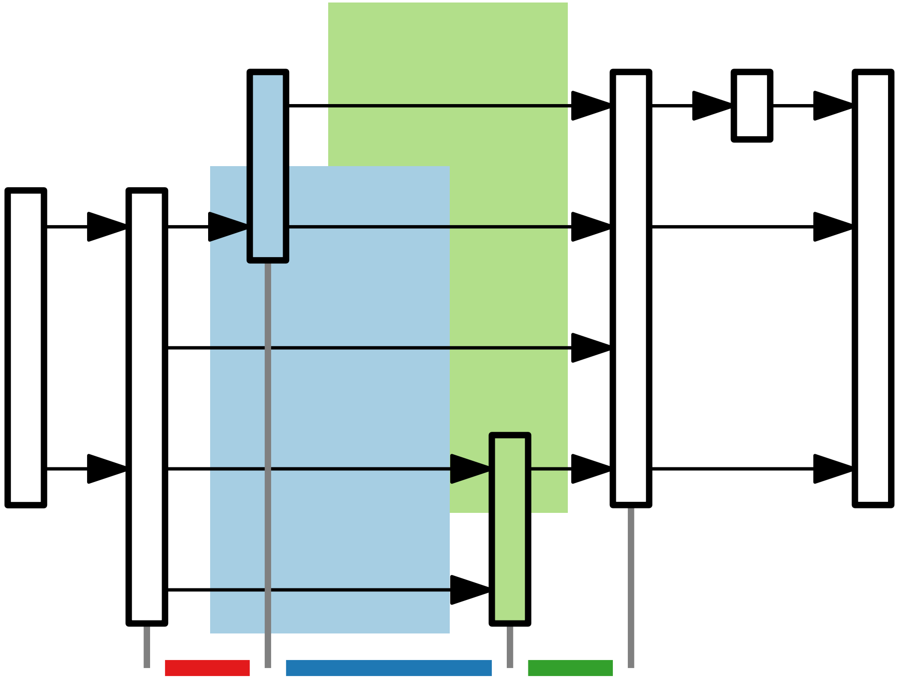
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

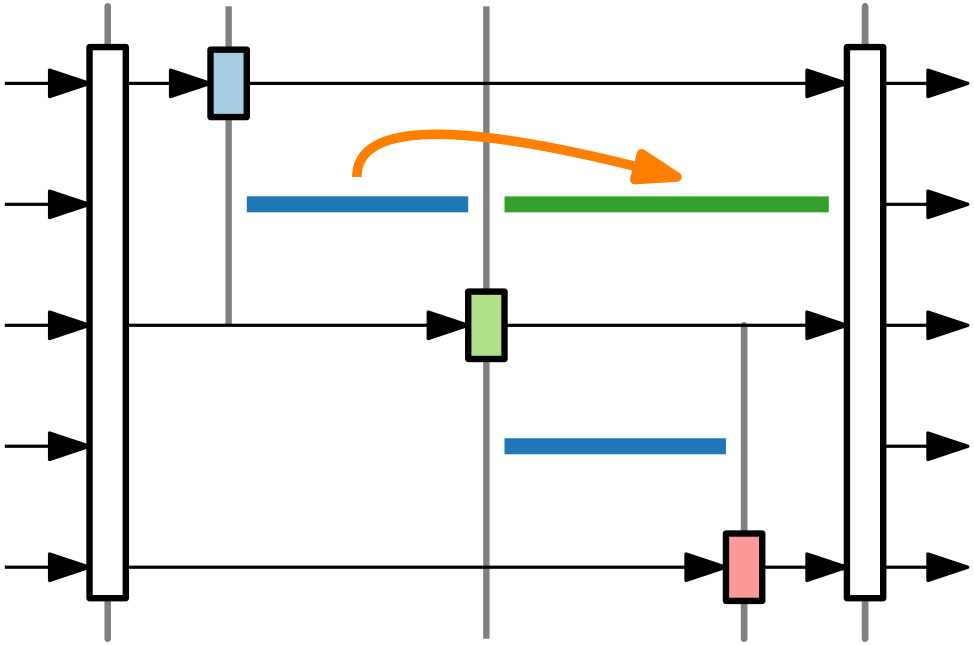




Resolving Conflicts

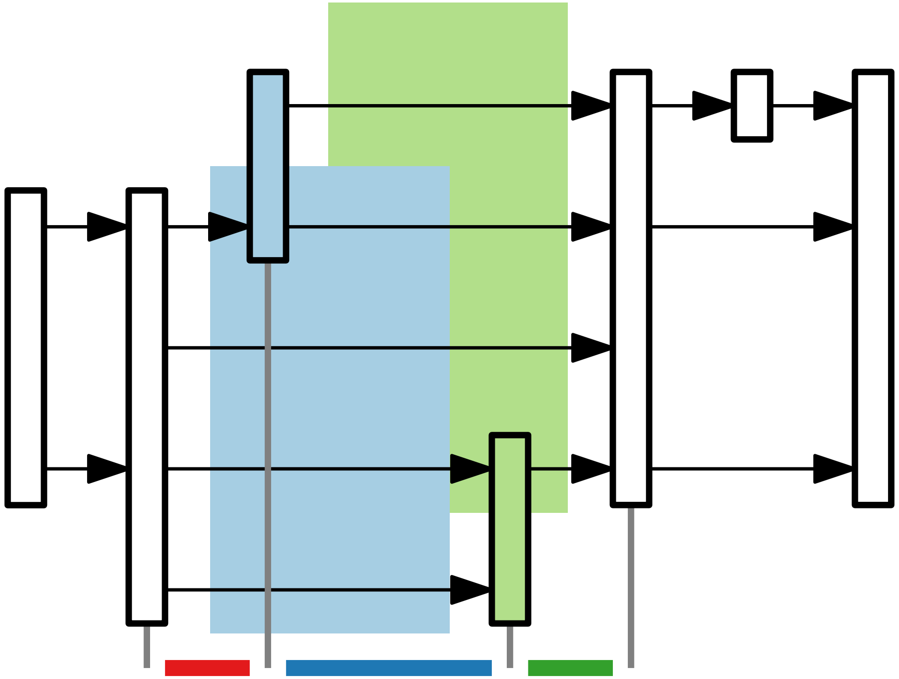
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

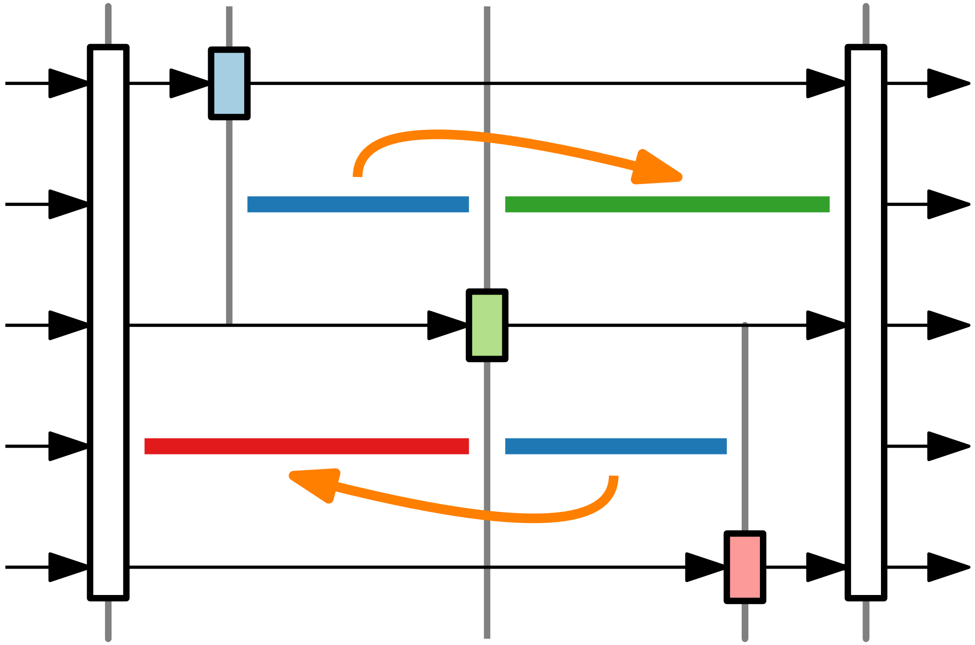




Resolving Conflicts

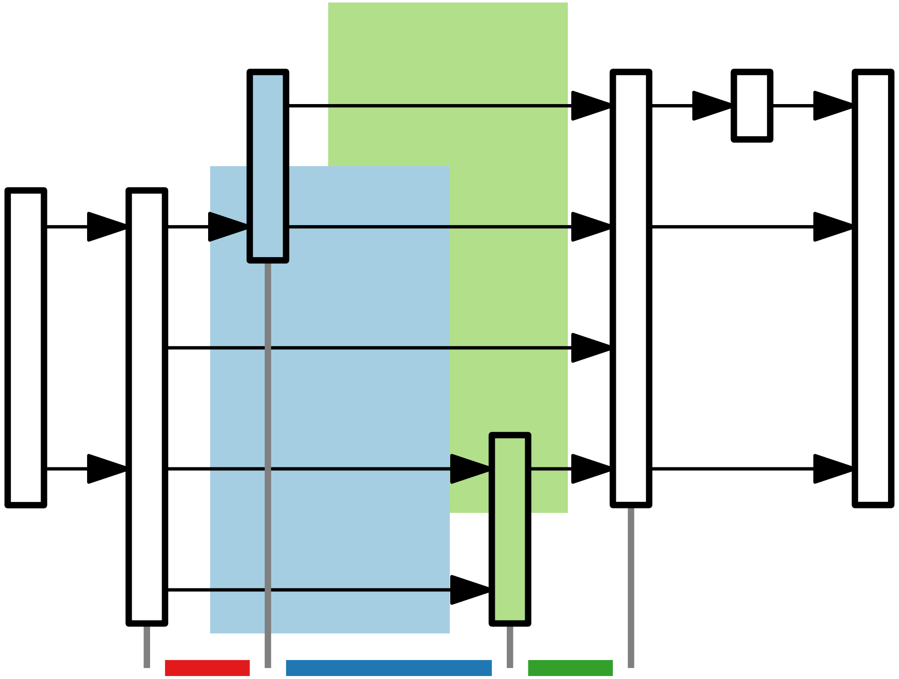
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

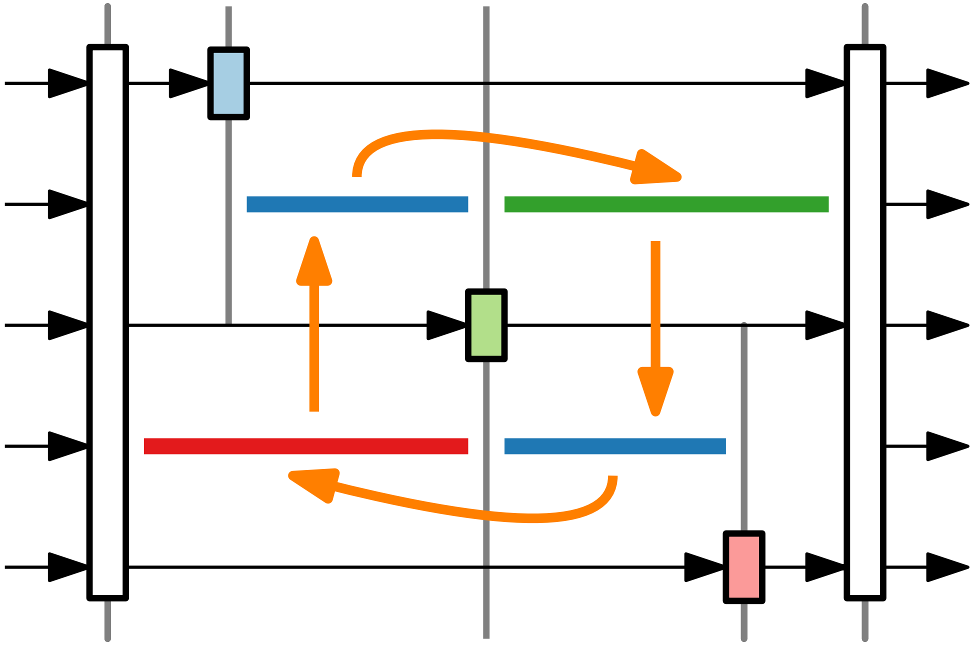




Resolving Conflicts

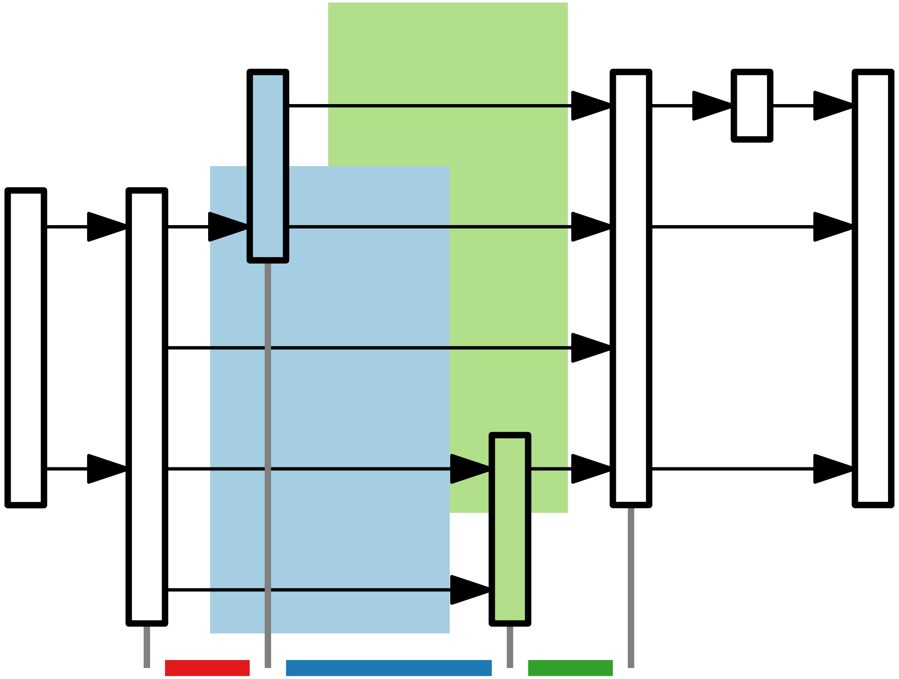
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

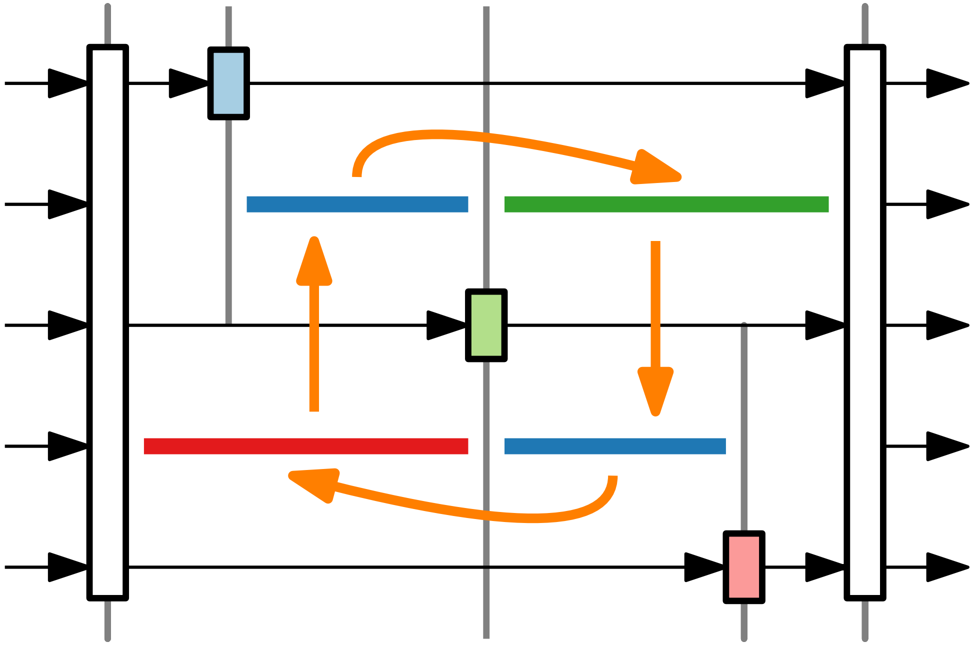


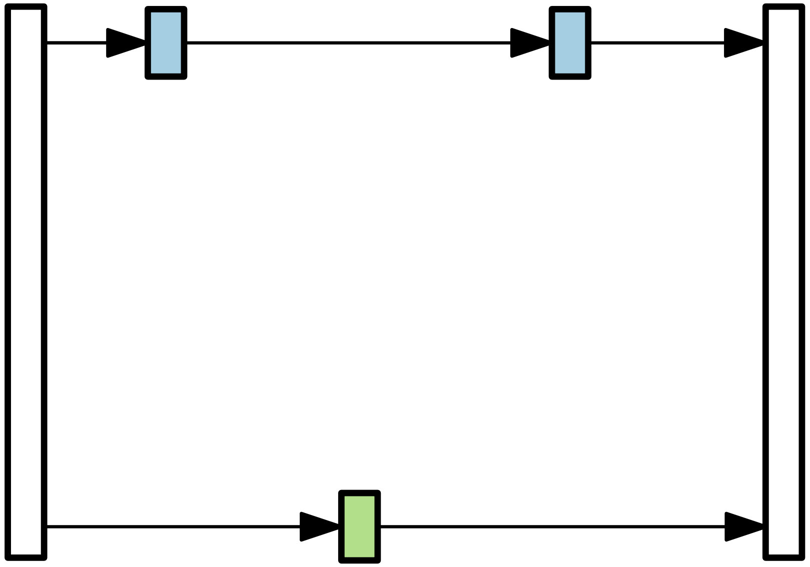


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

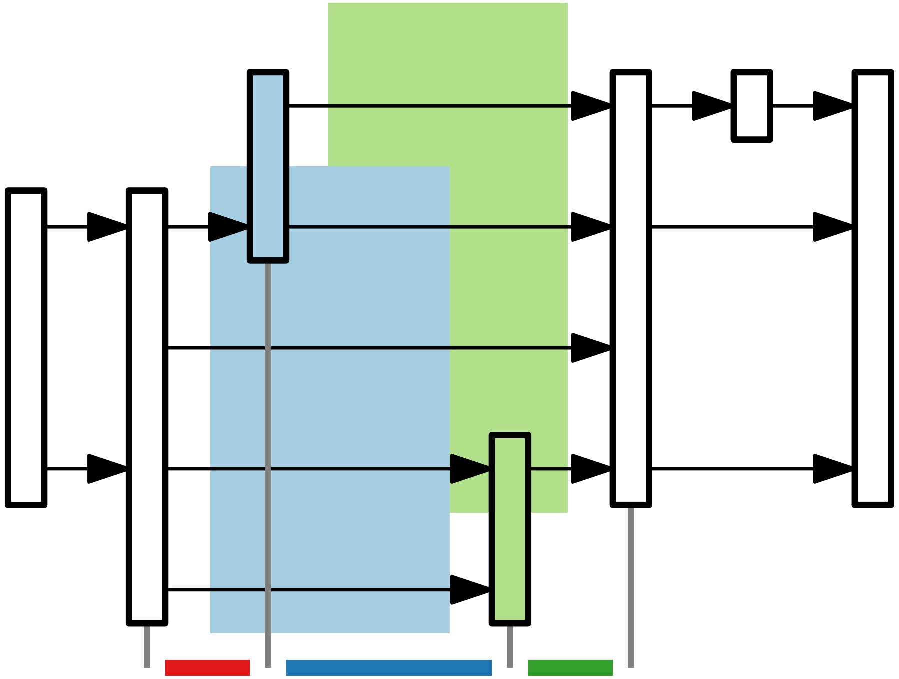


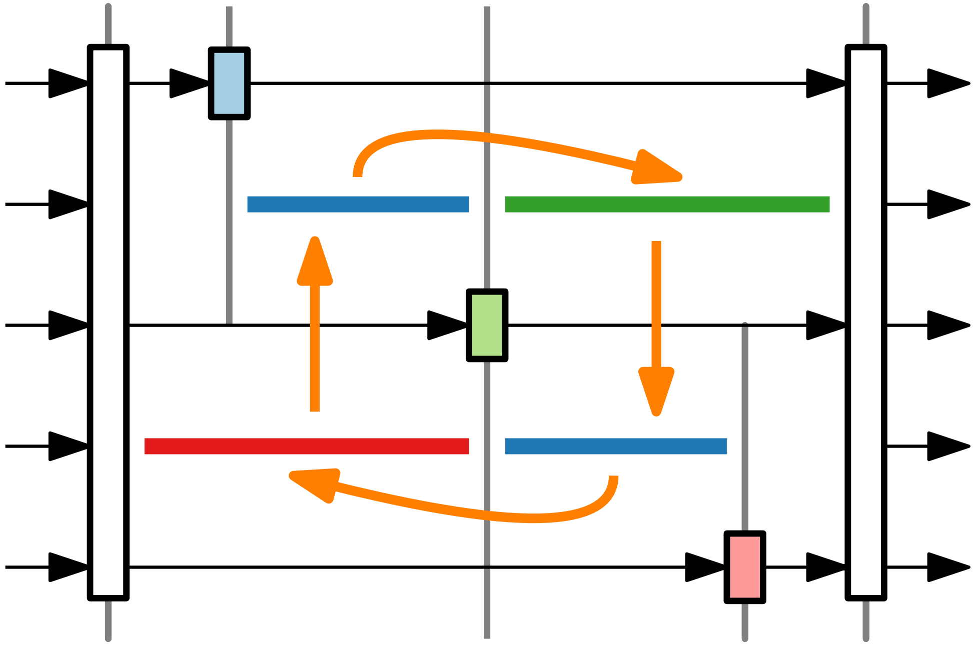


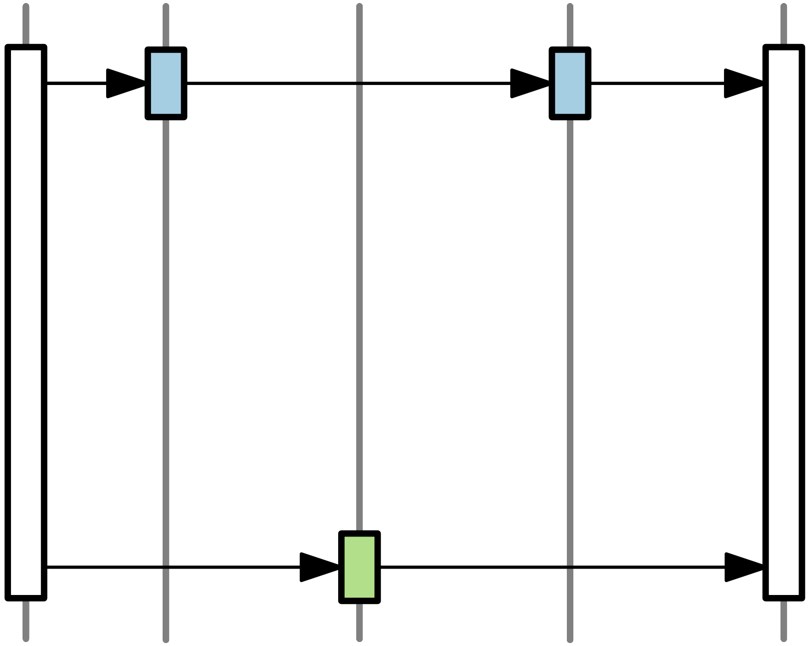


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

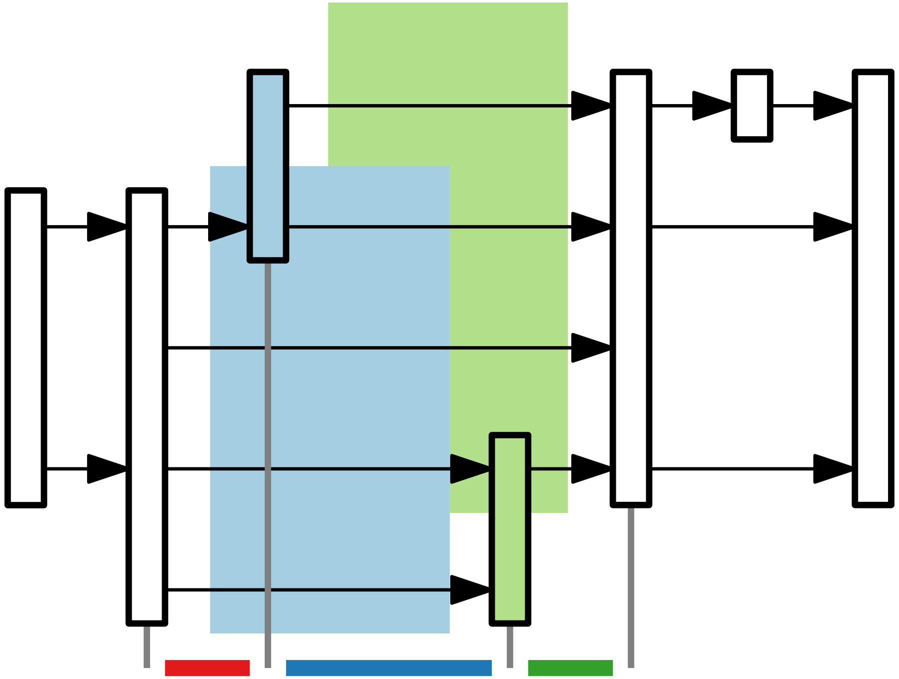


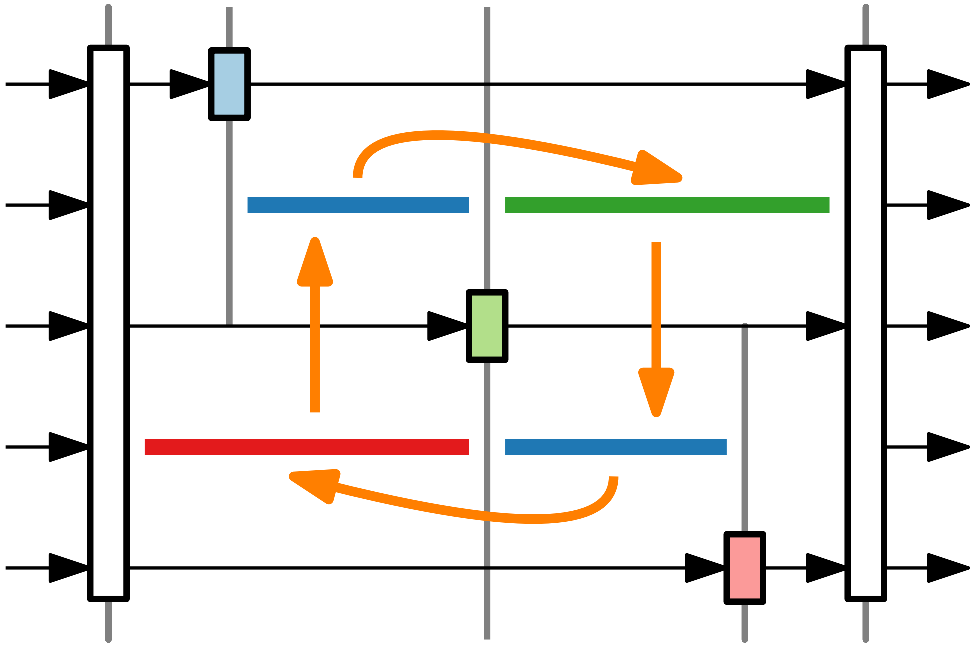


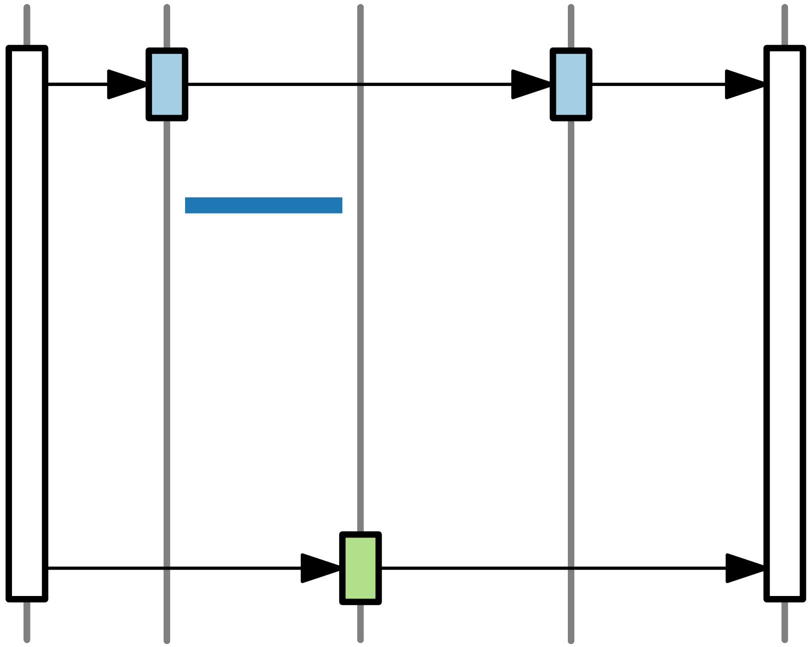


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

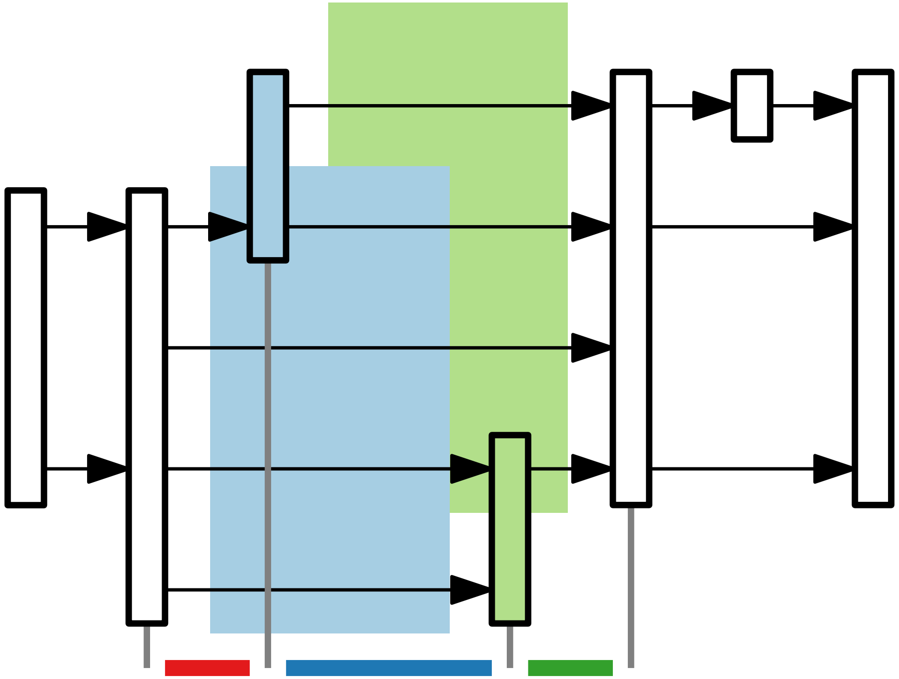


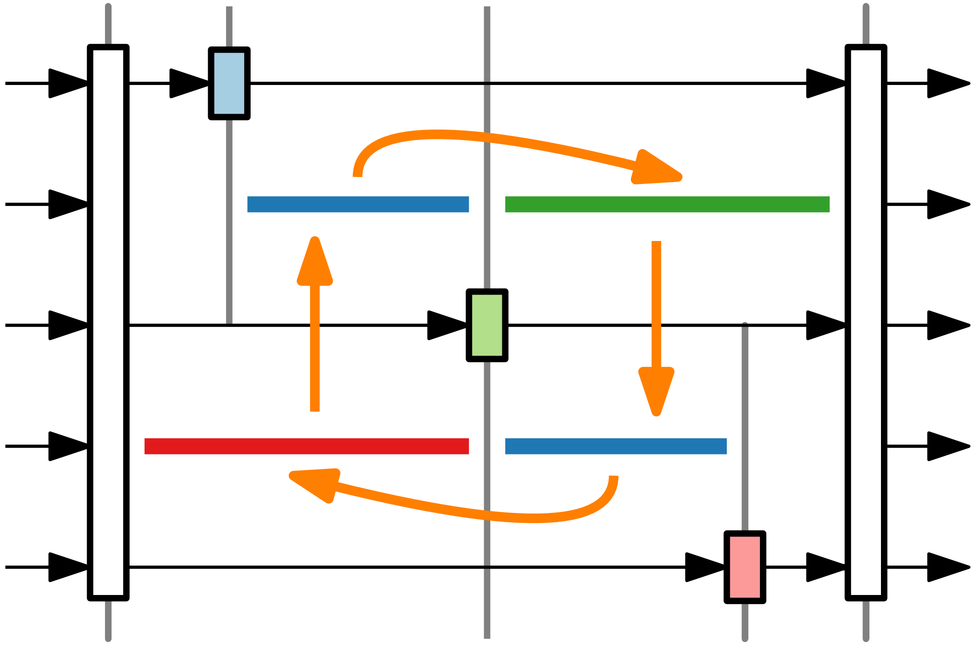


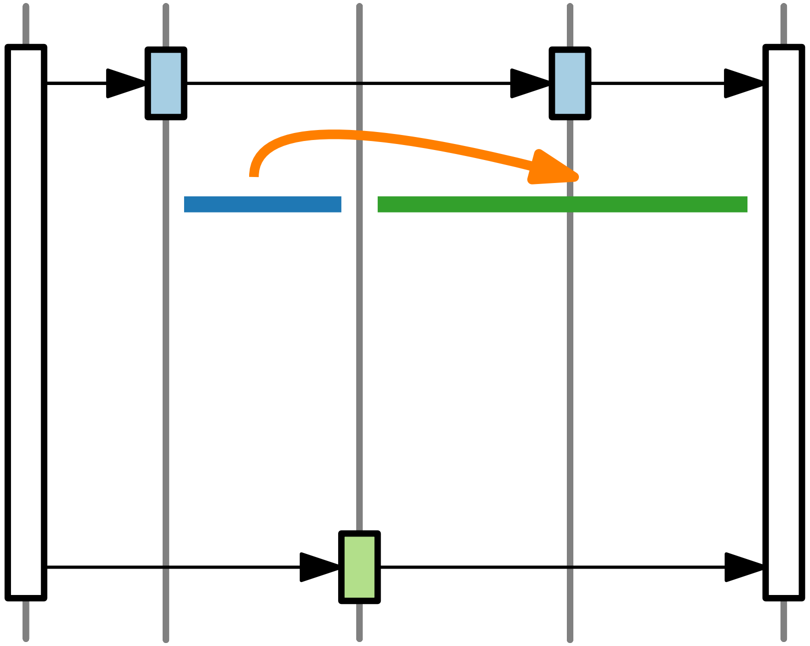


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

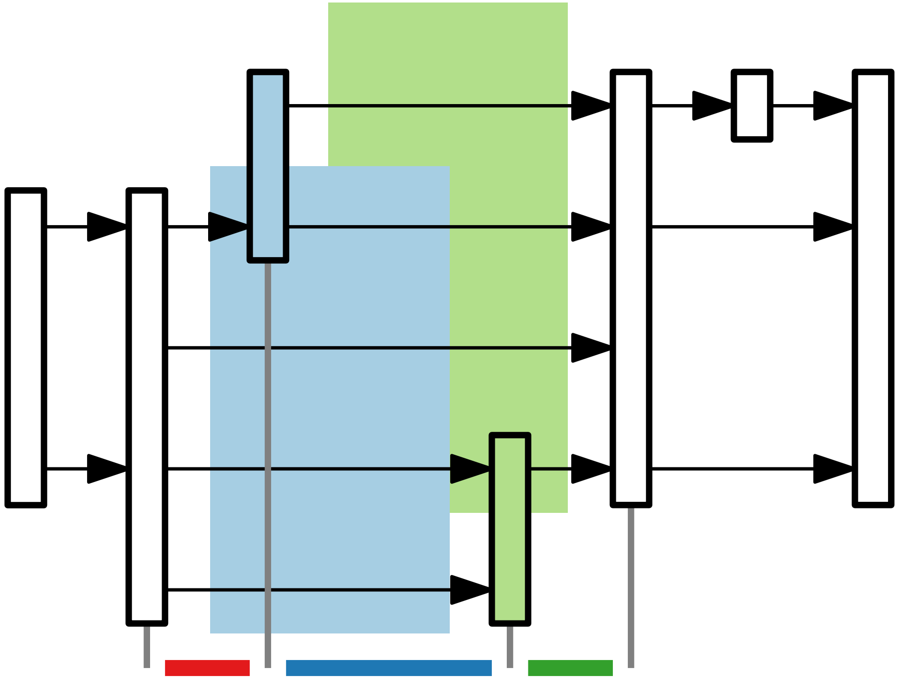


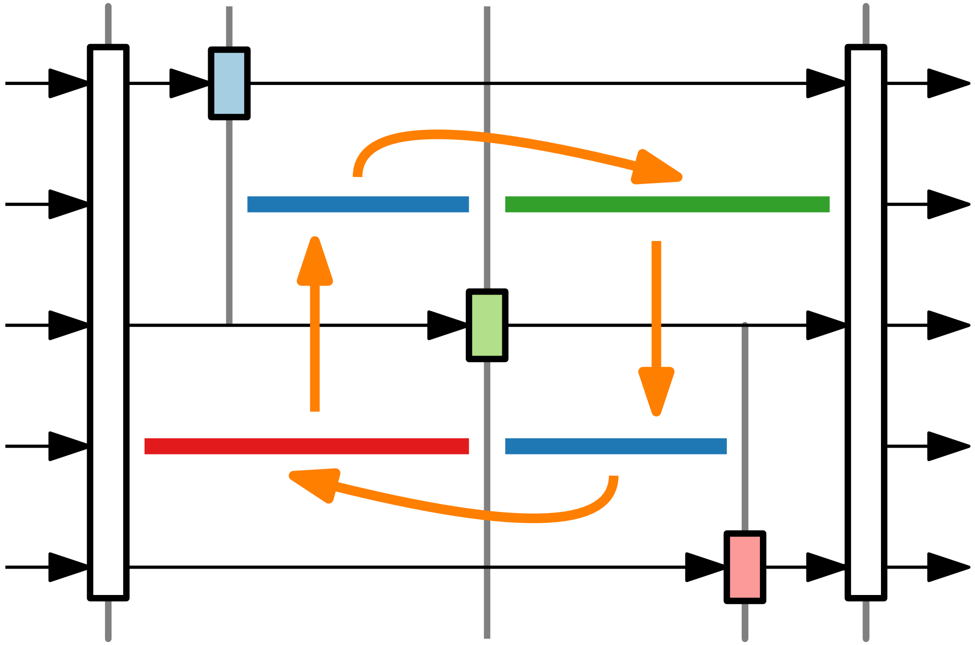


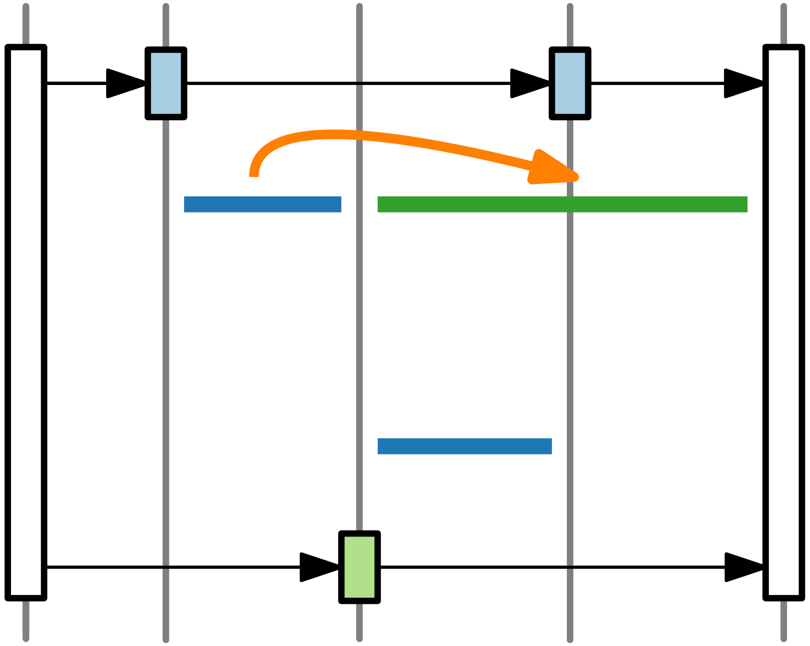


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

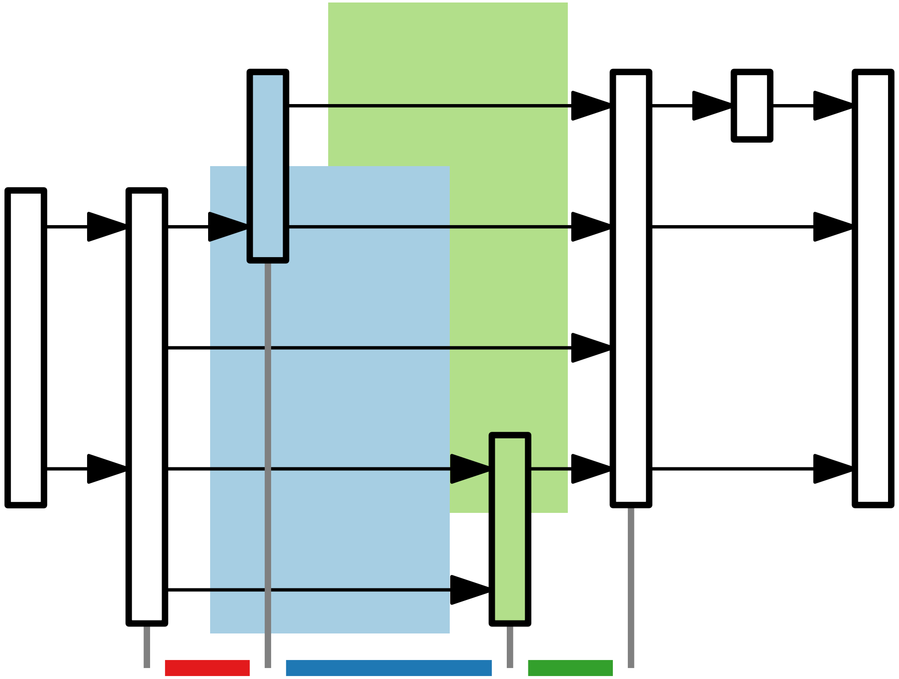


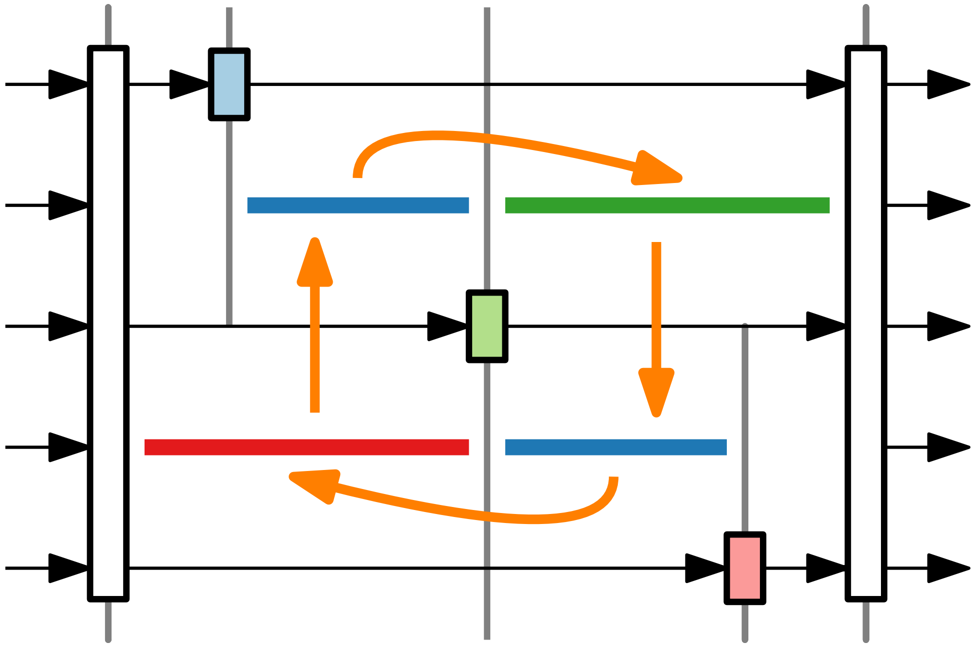


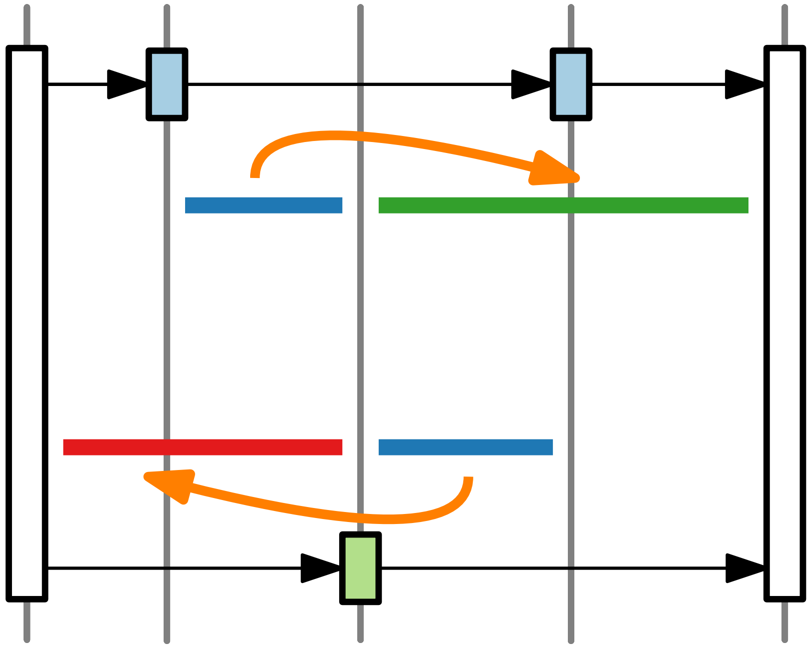


Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

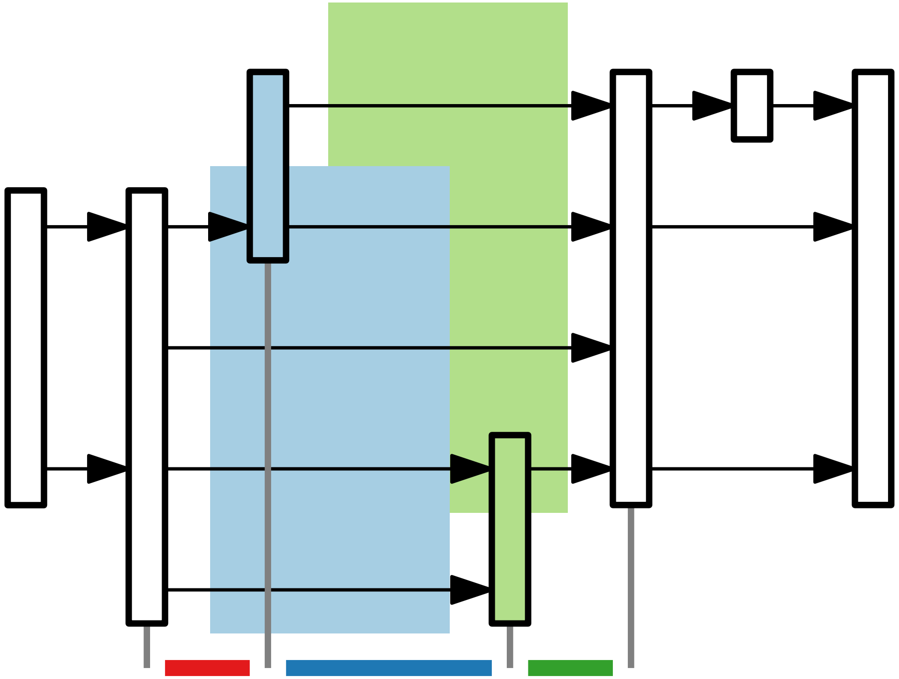


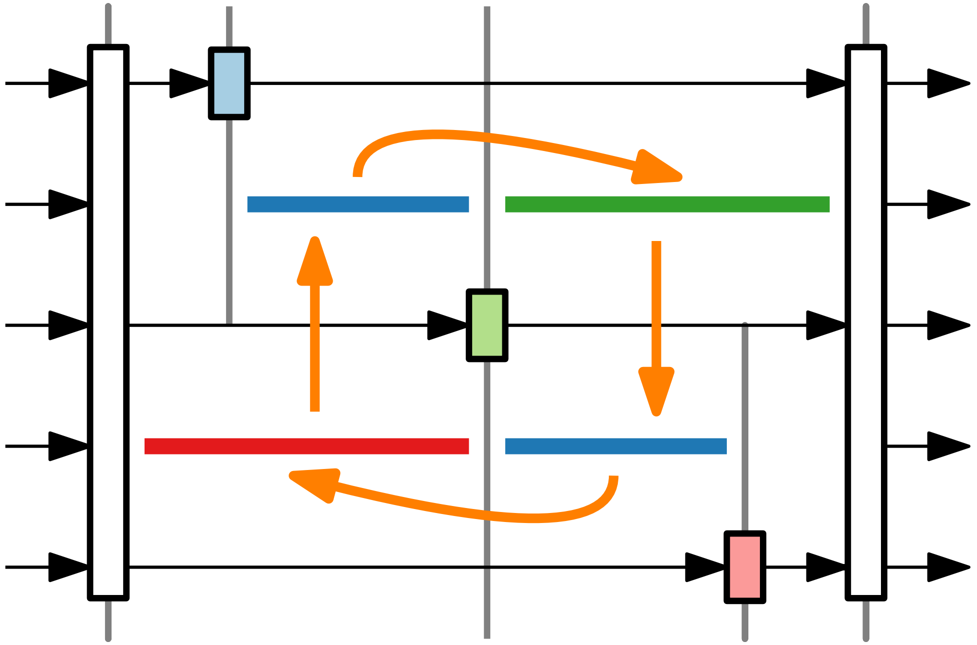


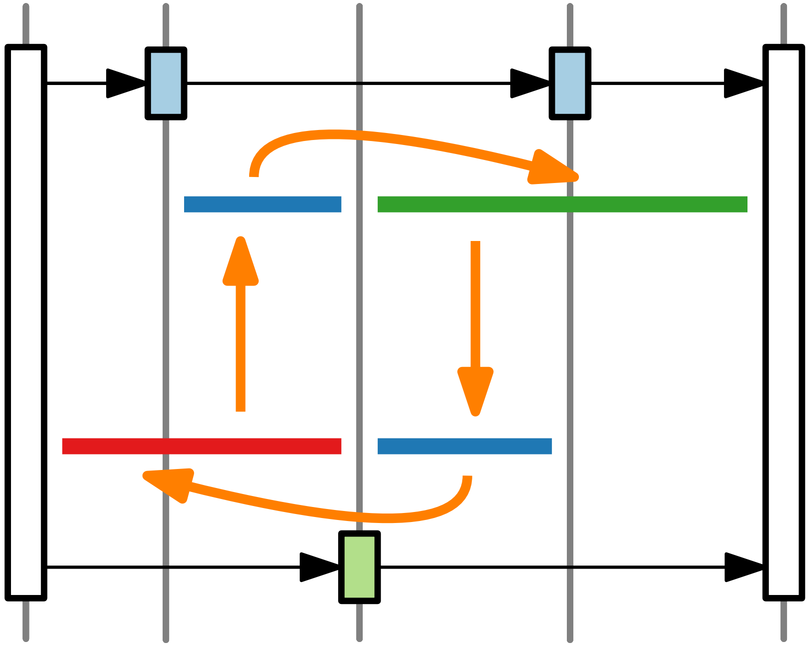


Resolving Conflicts

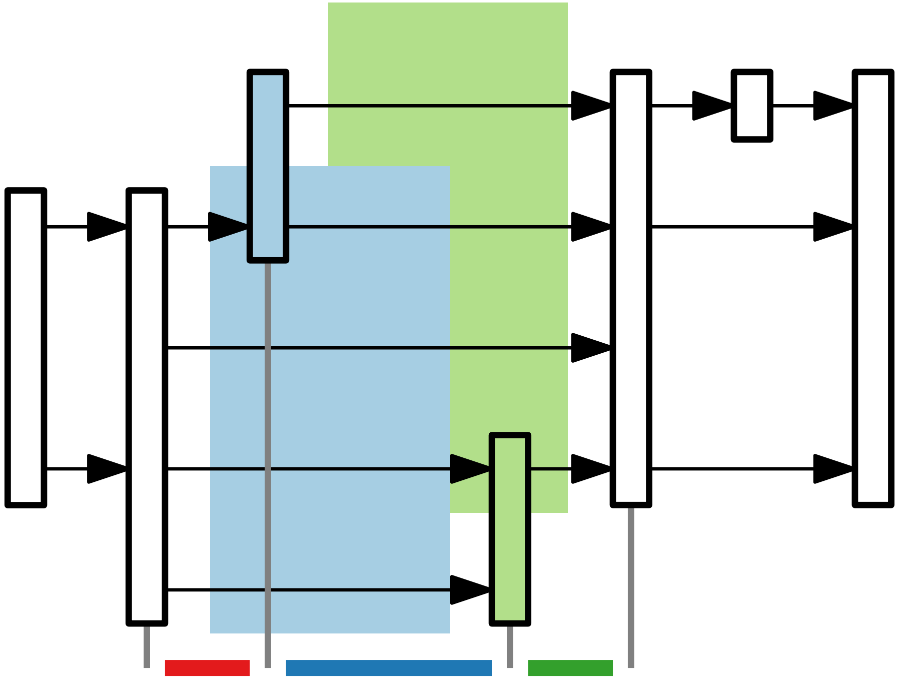
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩





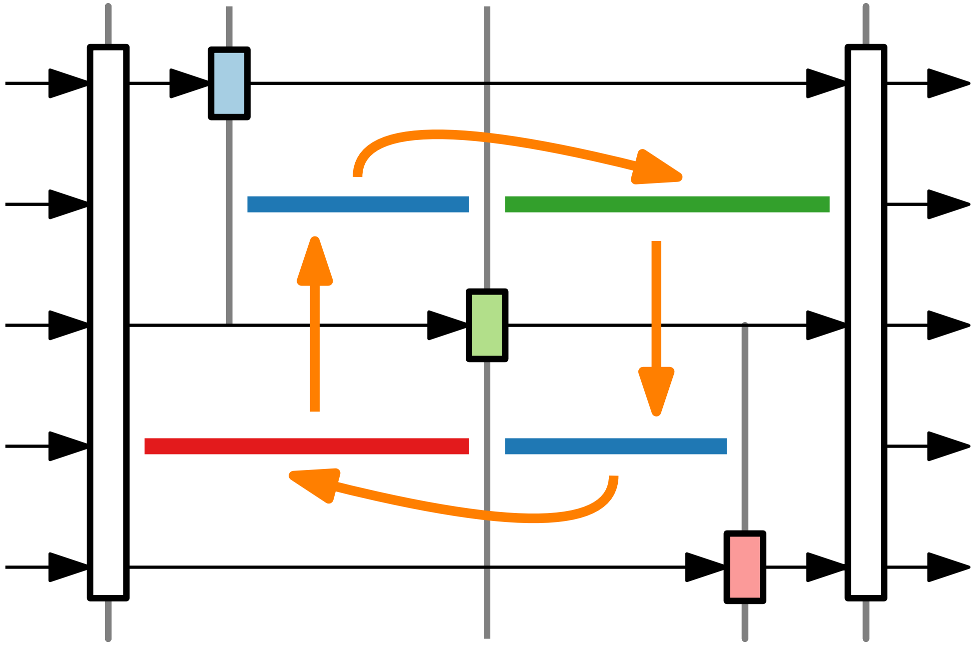


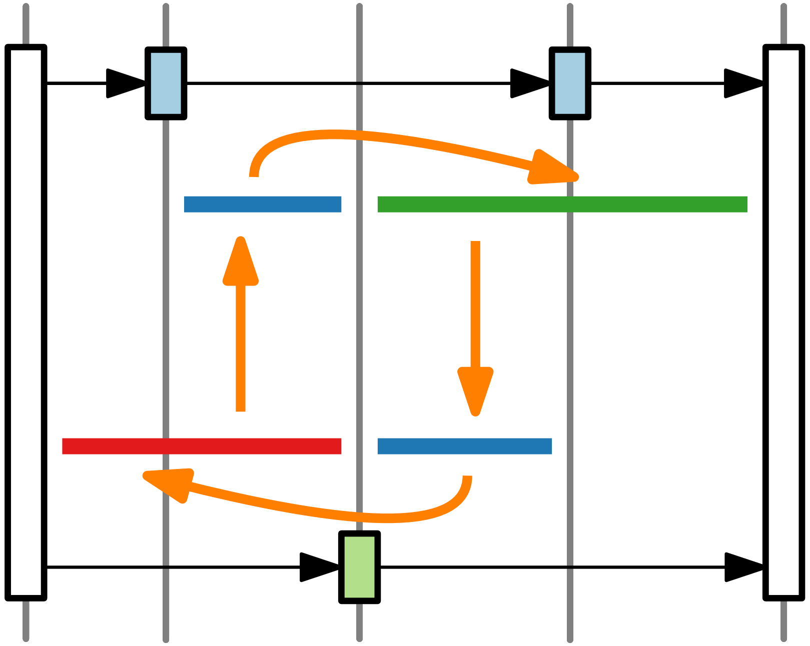
Resolving Conflicts



Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

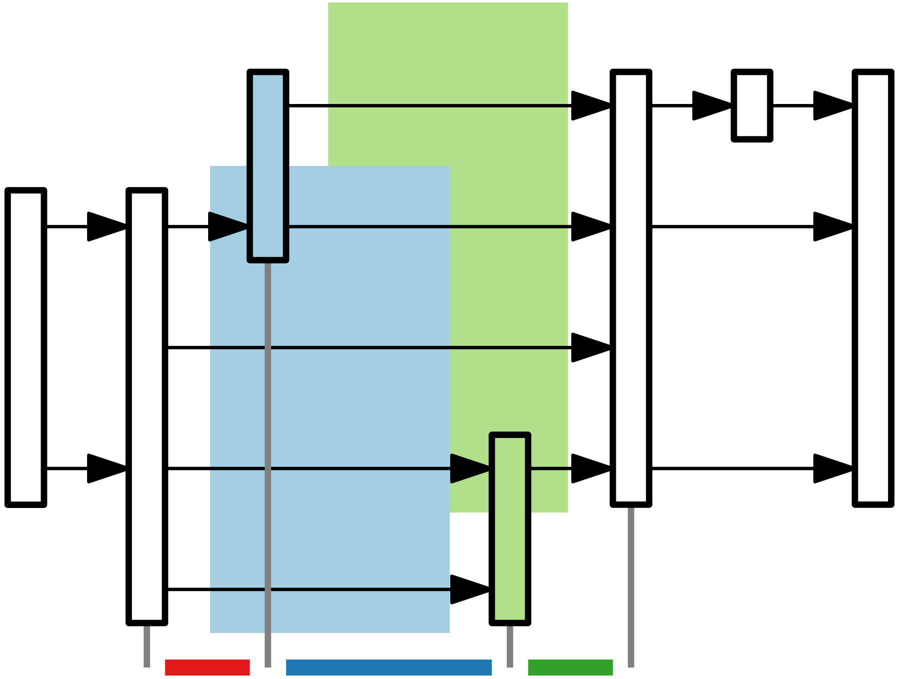
Need a *good* *st*-order:



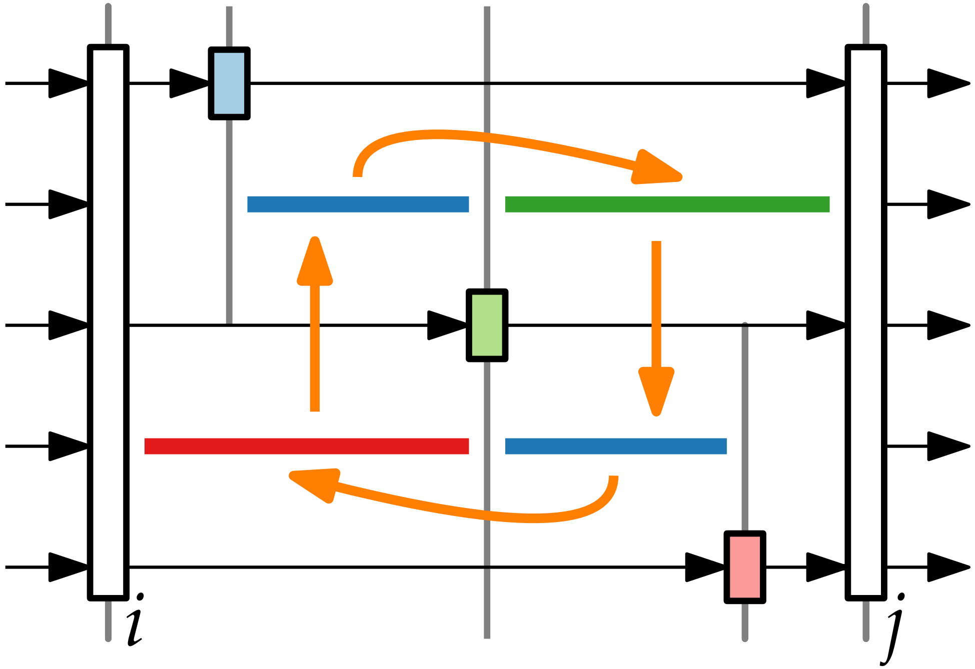


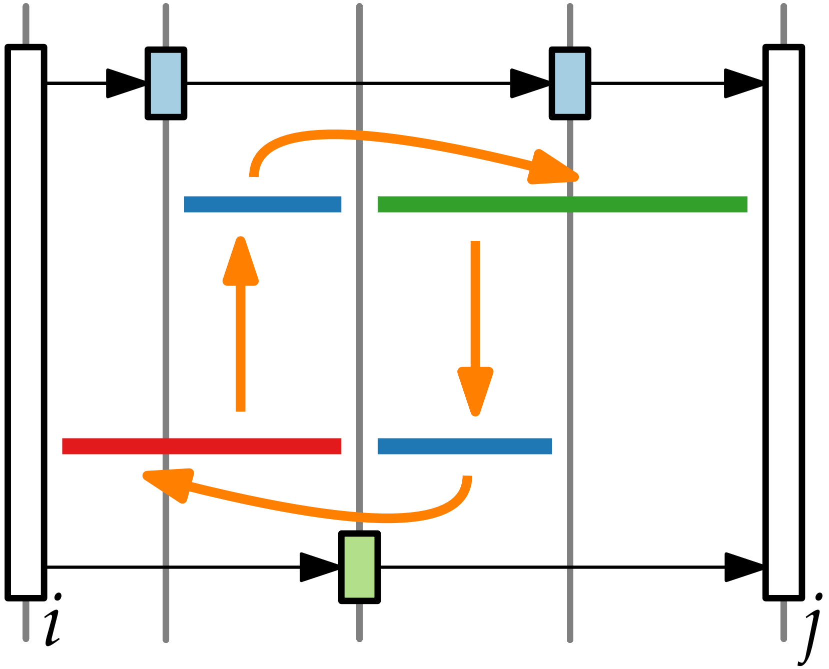
Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩



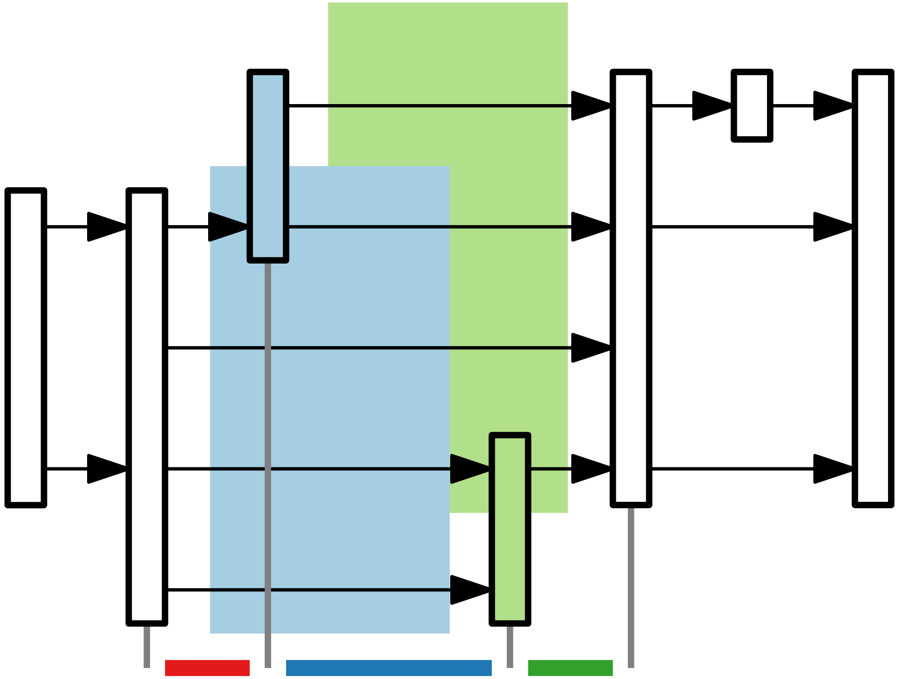
Need a *good* *st*-order:
for every interval $i, \ldots, j$:



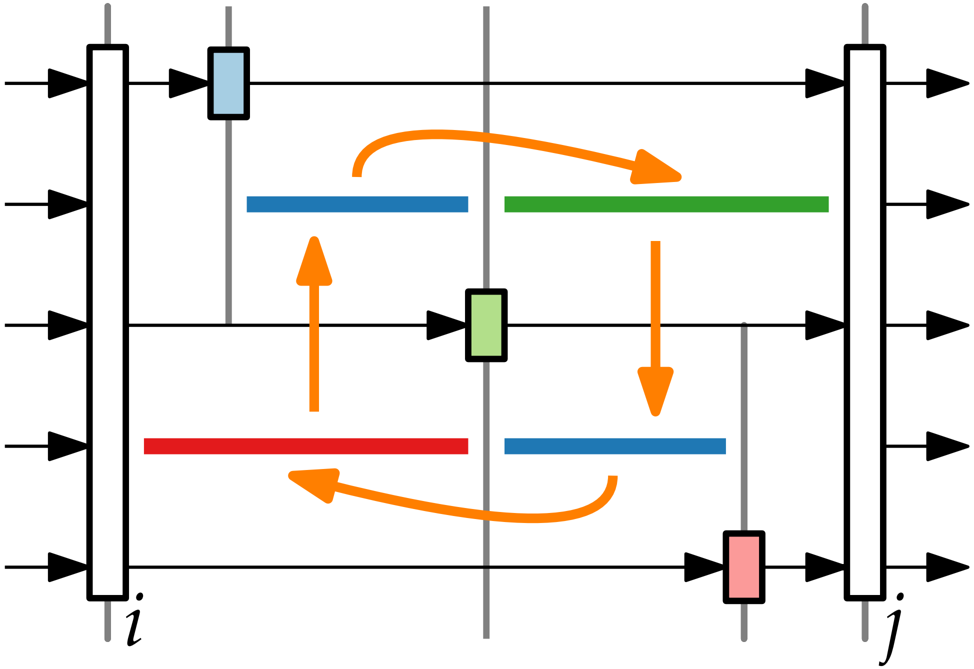


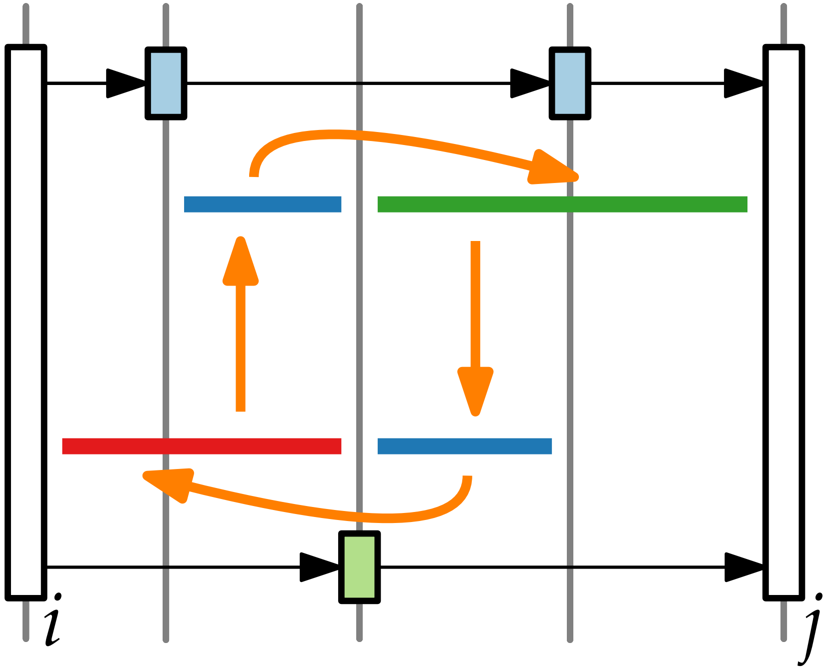
Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩



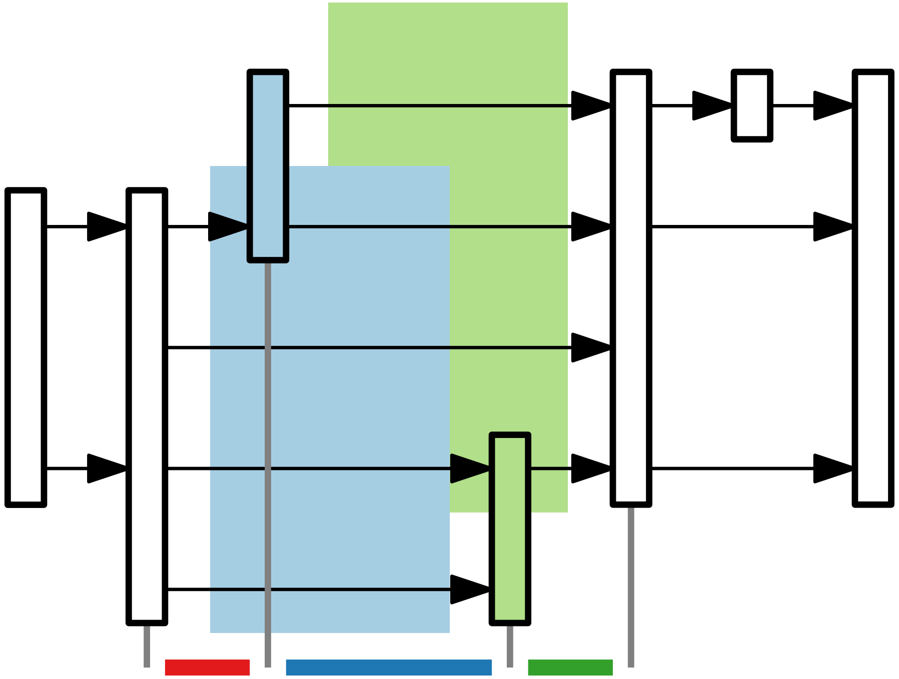
Need a *good st*-order:
for every interval $i, \ldots, j$:
– no 3 conn. comp.





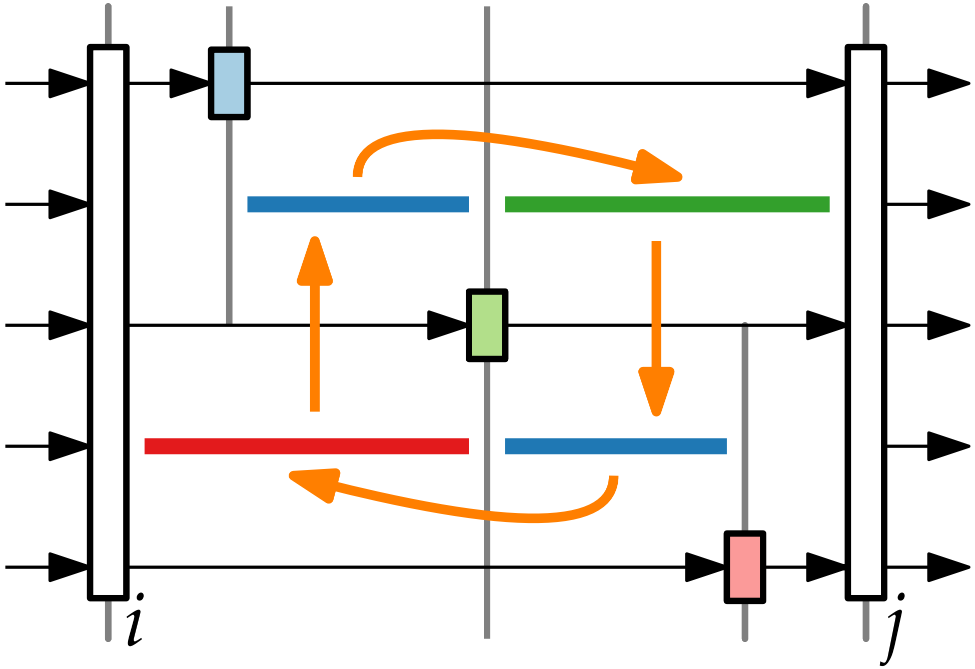
Resolving Conflicts

Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩

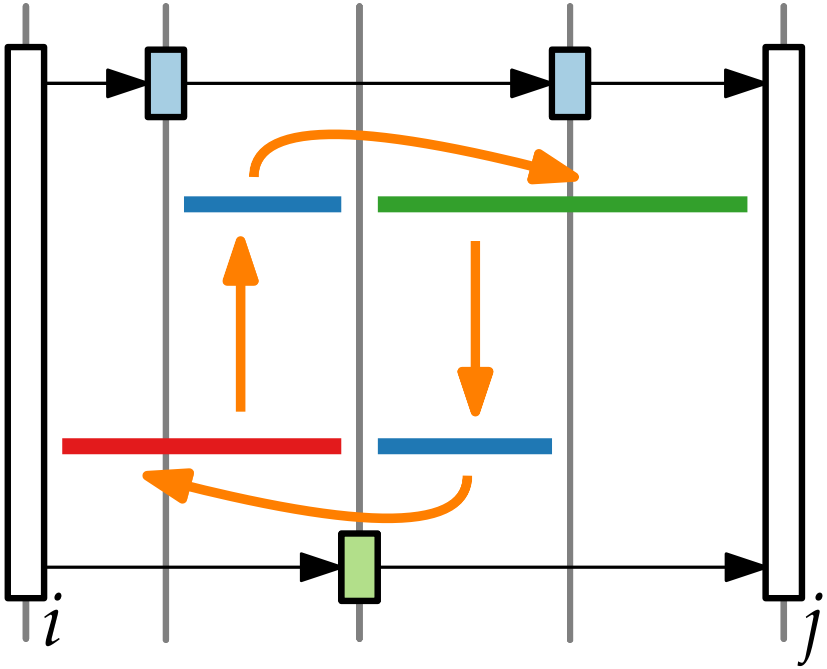


Need a *good* st-order:
for every interval $i, \ldots, j$:

– no 3 conn. comp.

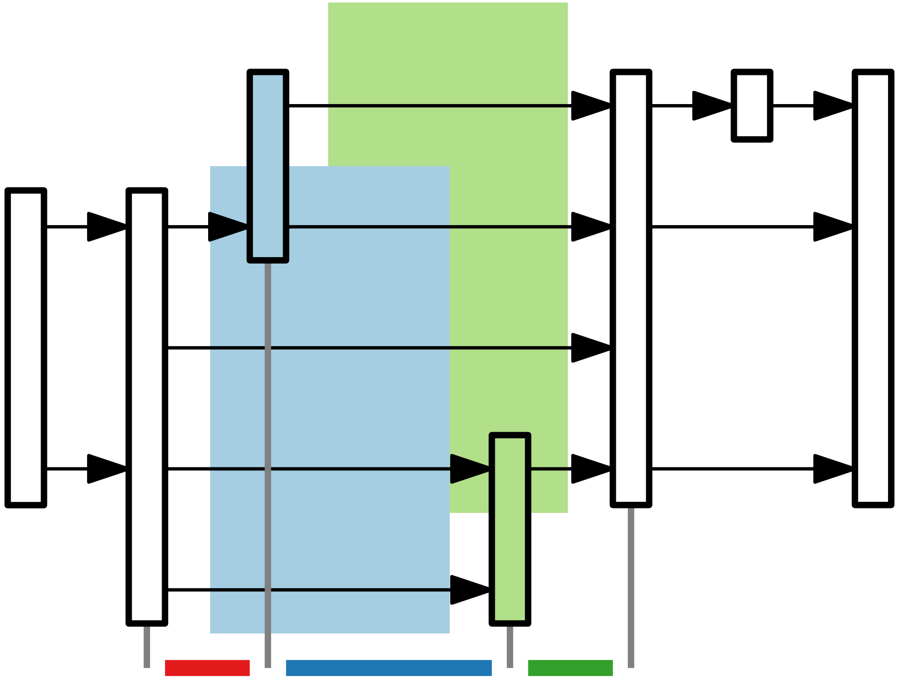


– if 2 conn. comp., then disjoint



Resolving Conflicts

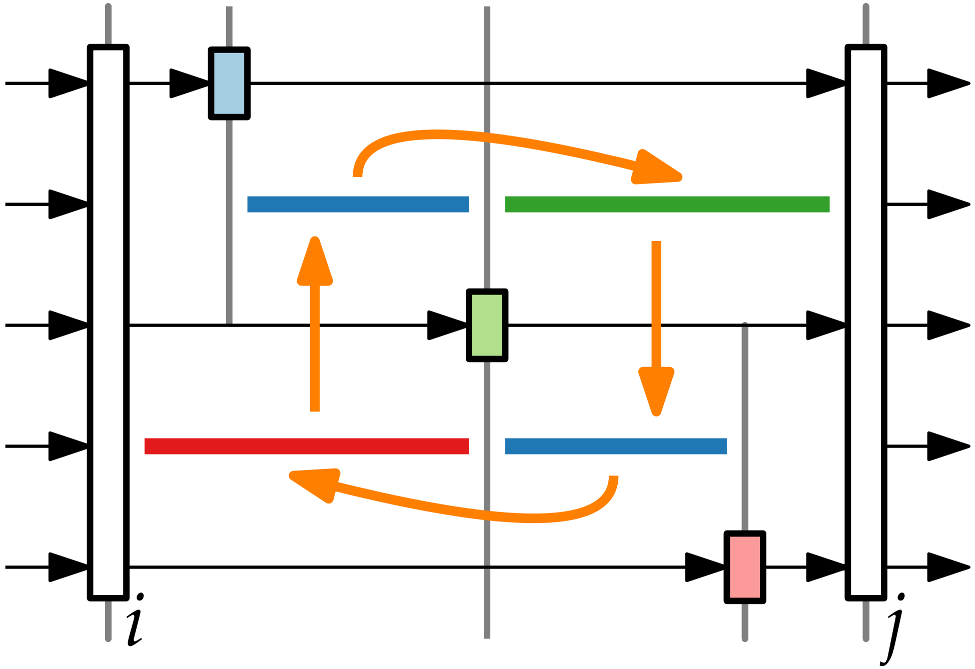
Conflict resolved: 🟦 > 🟥 and 🟦 > 🟩



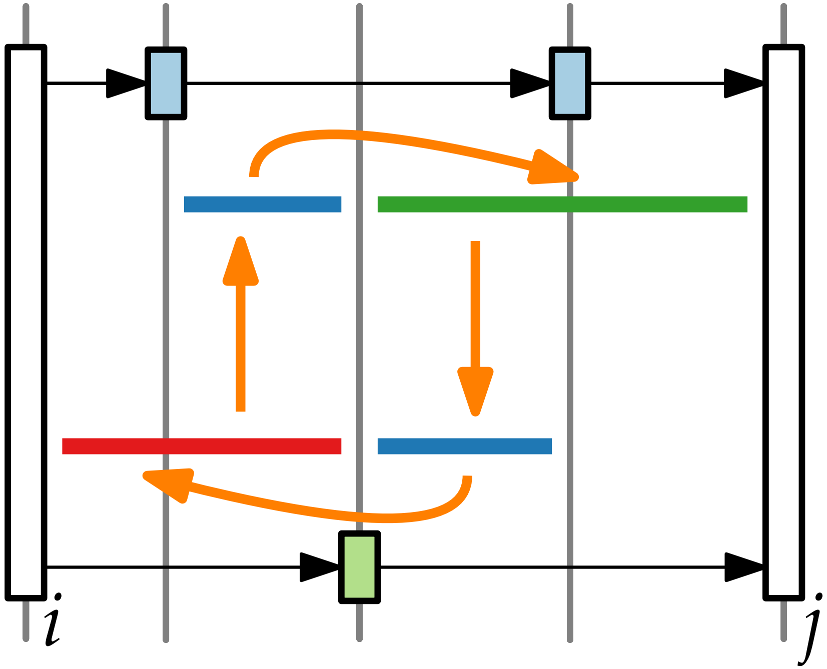
sufficient?

Need a *good st*-order:
for every interval $i, \ldots, j$:

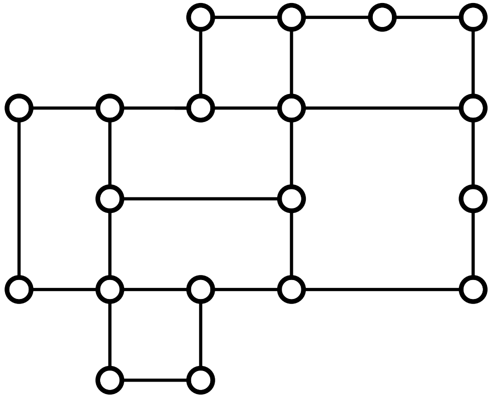
– no 3 conn. comp.



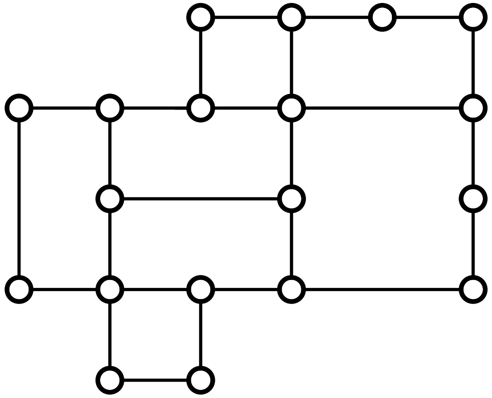
– if 2 conn. comp., then disjoint

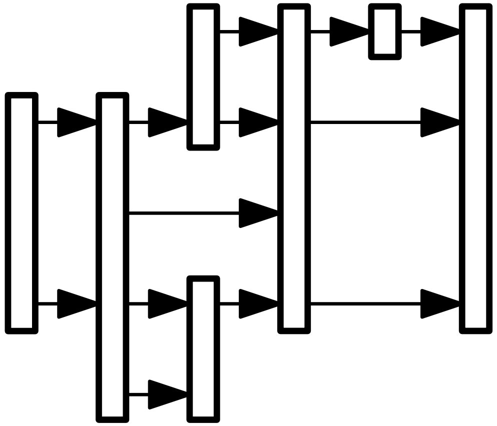


x- and *y*-conflicts independent

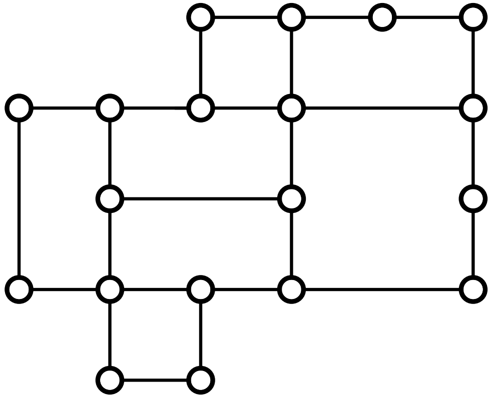


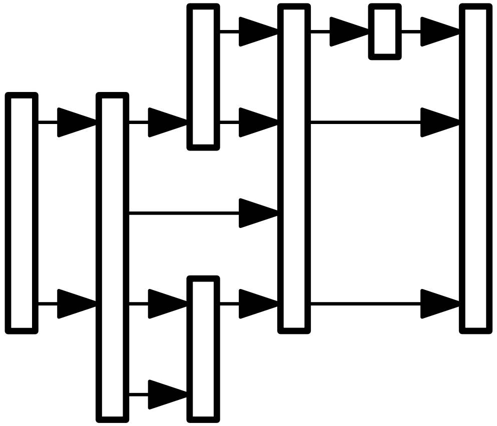
x- and *y*-conflicts independent

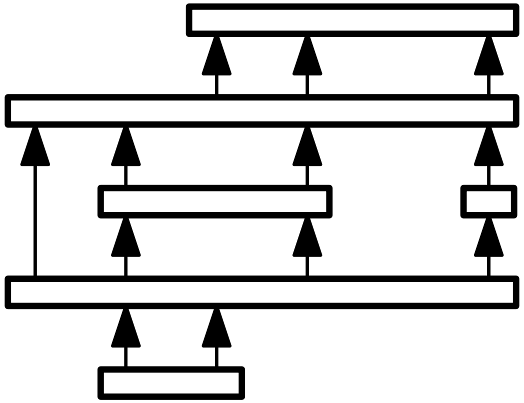




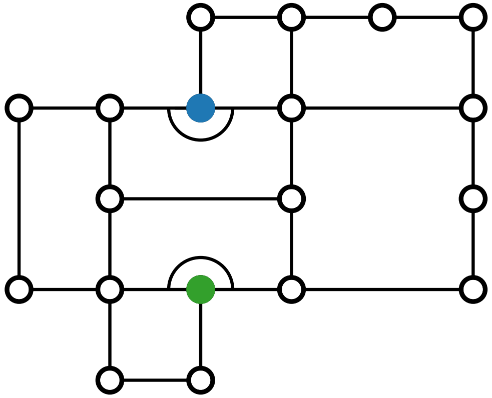
x- and *y*-conflicts independent

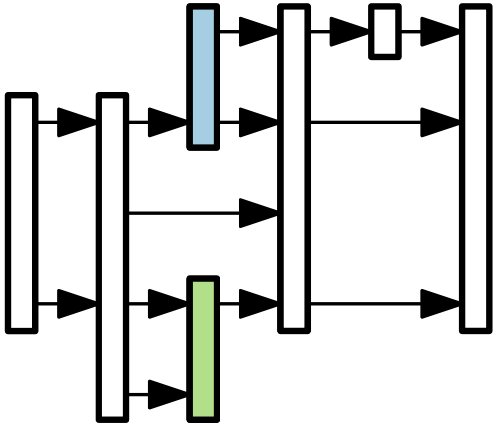


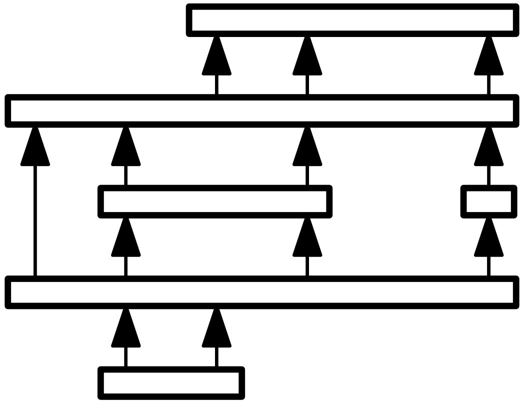




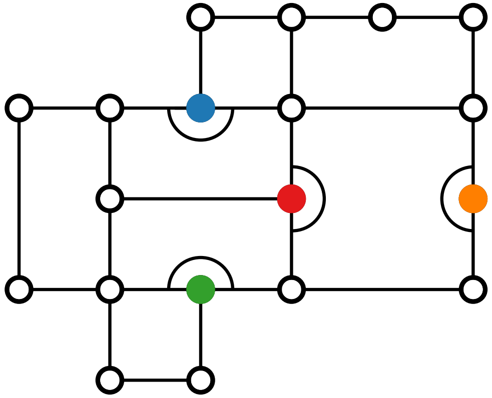
x- and *y*-conflicts independent

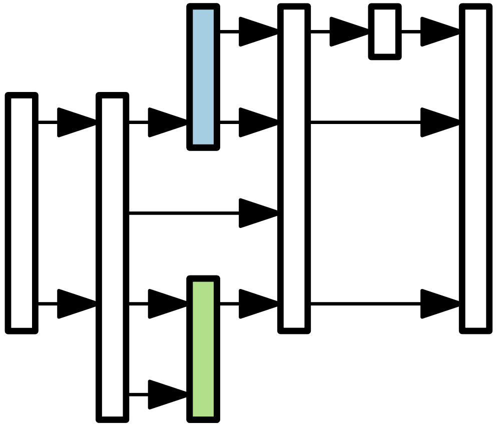


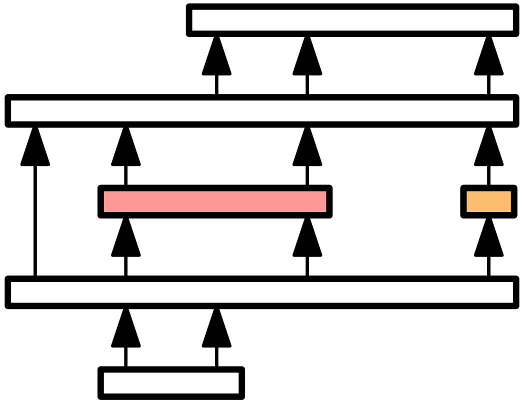




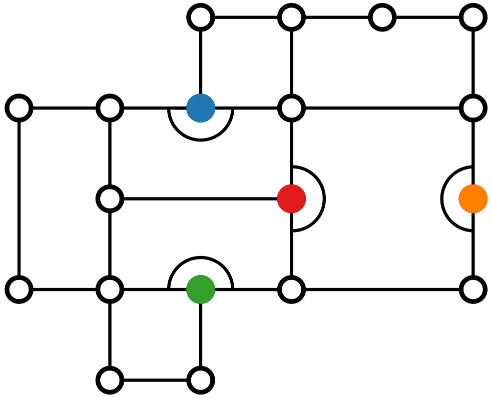
x- and *y*-conflicts independent

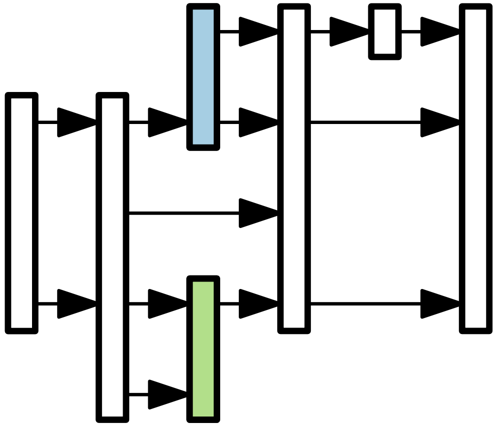




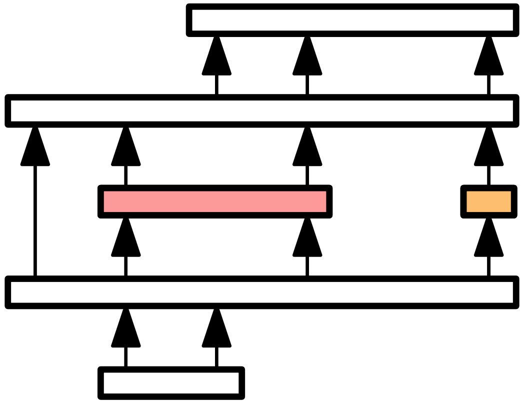


x- and *y*-conflicts independent

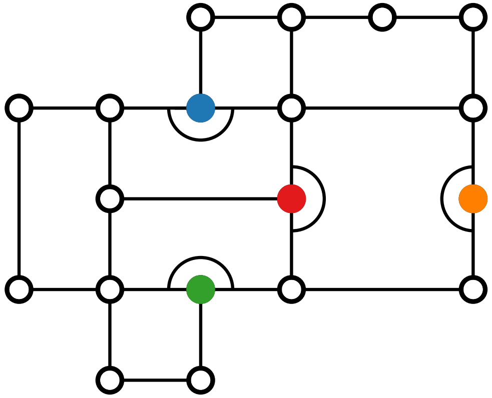


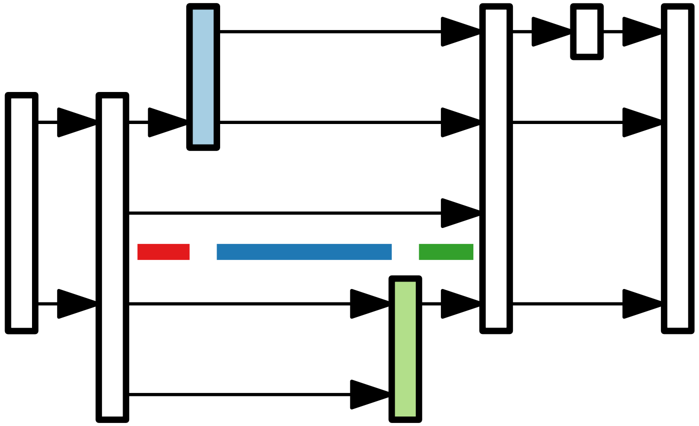


resolve *x*-conflicts

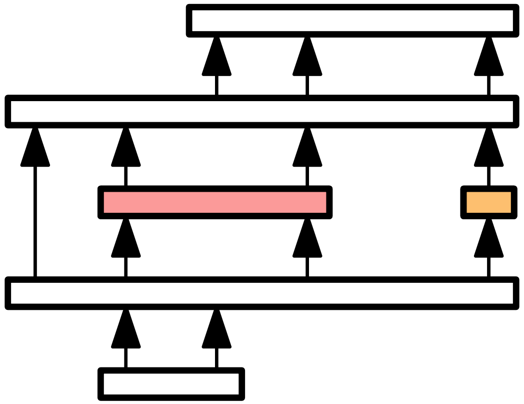


x- and *y*-conflicts independent

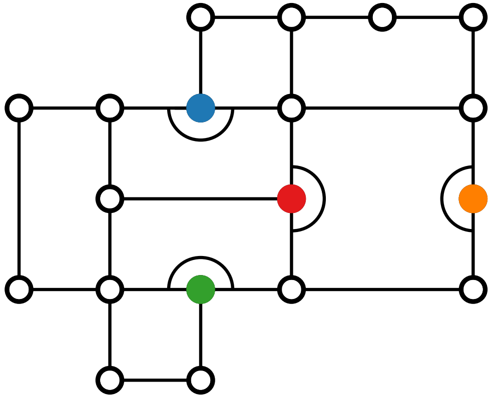


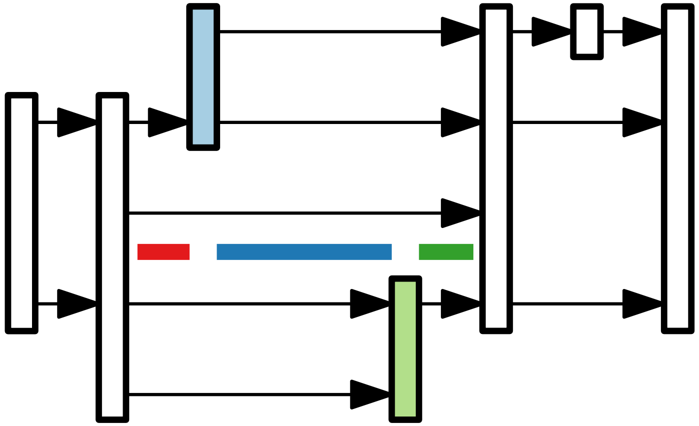


resolve *x*-conflicts

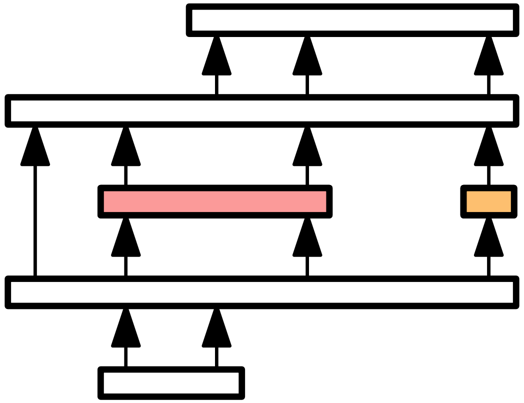


x- and y-conflicts independent



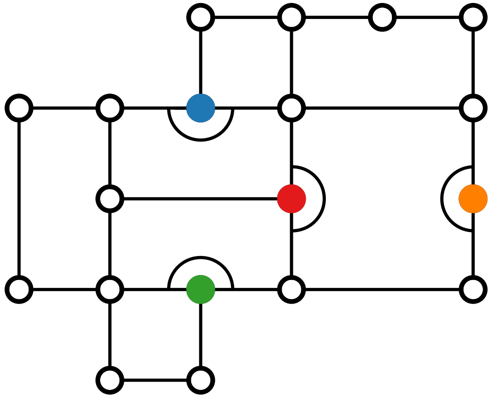


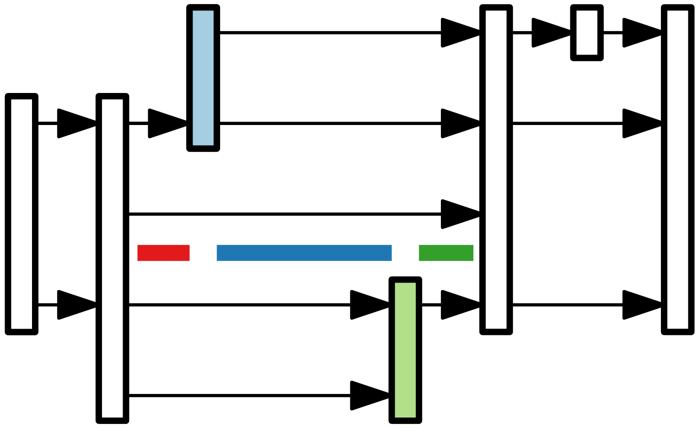
resolve x-conflicts



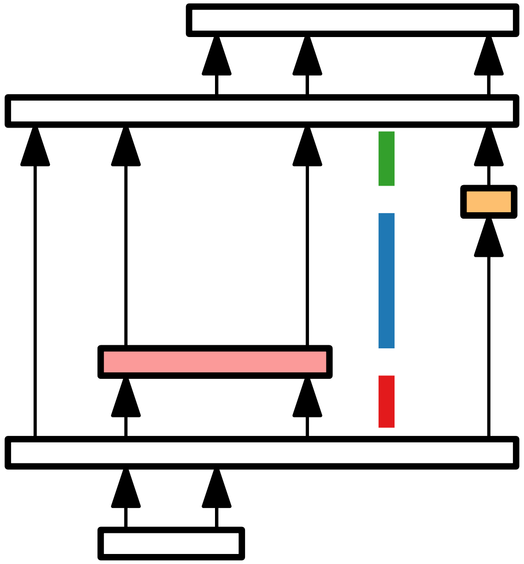
resolve y-conflicts

x- and *y*-conflicts independent



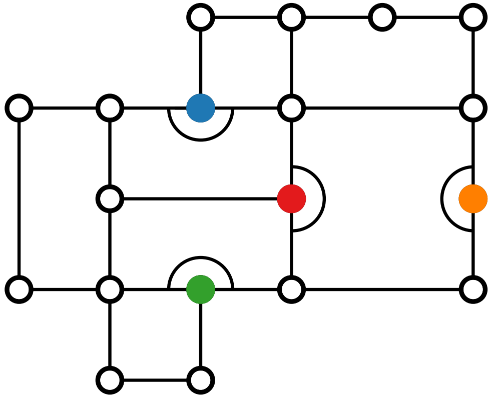


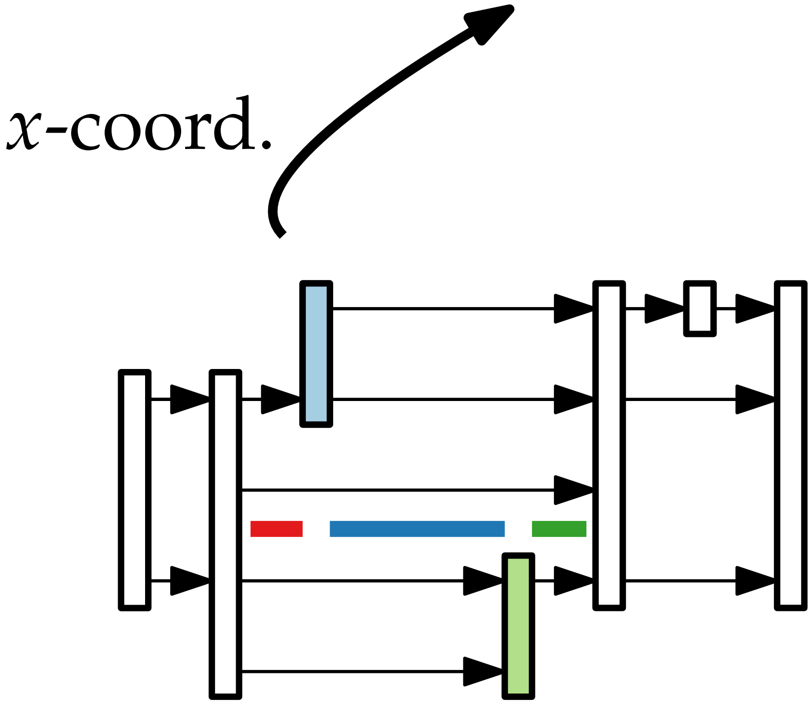
resolve *x*-conflicts



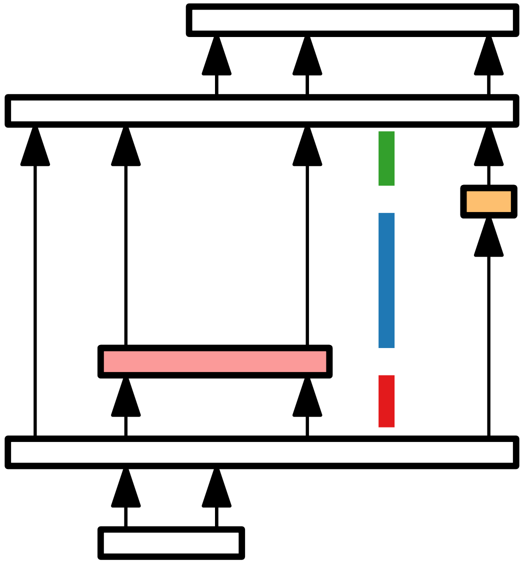
resolve *y*-conflicts

x- and y-conflicts independent



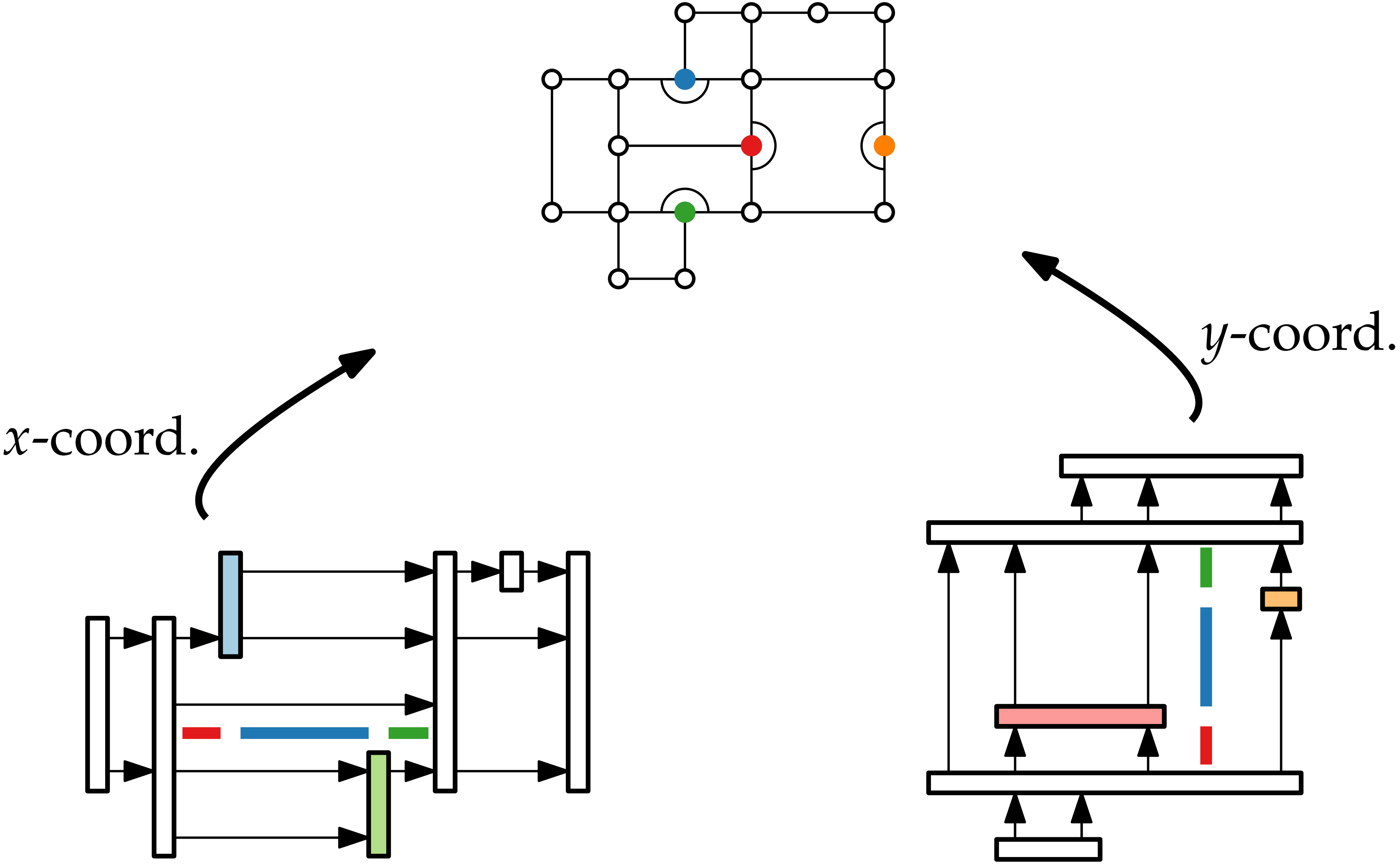


resolve x-conflicts



resolve y-conflicts

x- and *y*-conflicts independent



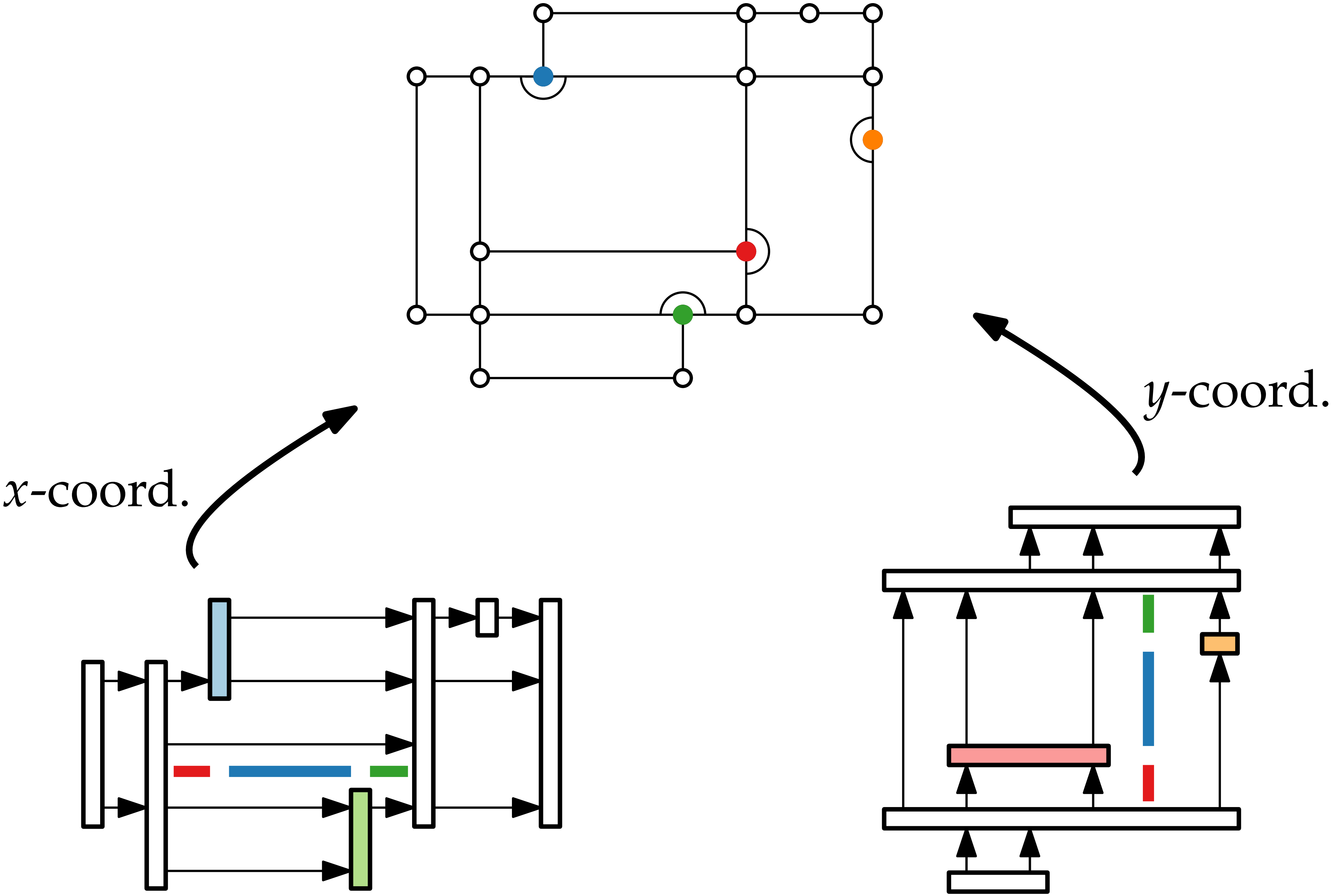
x-coord.

y-coord.

resolve *x*-conflicts

resolve *y*-conflicts

x- and *y*-conflicts independent



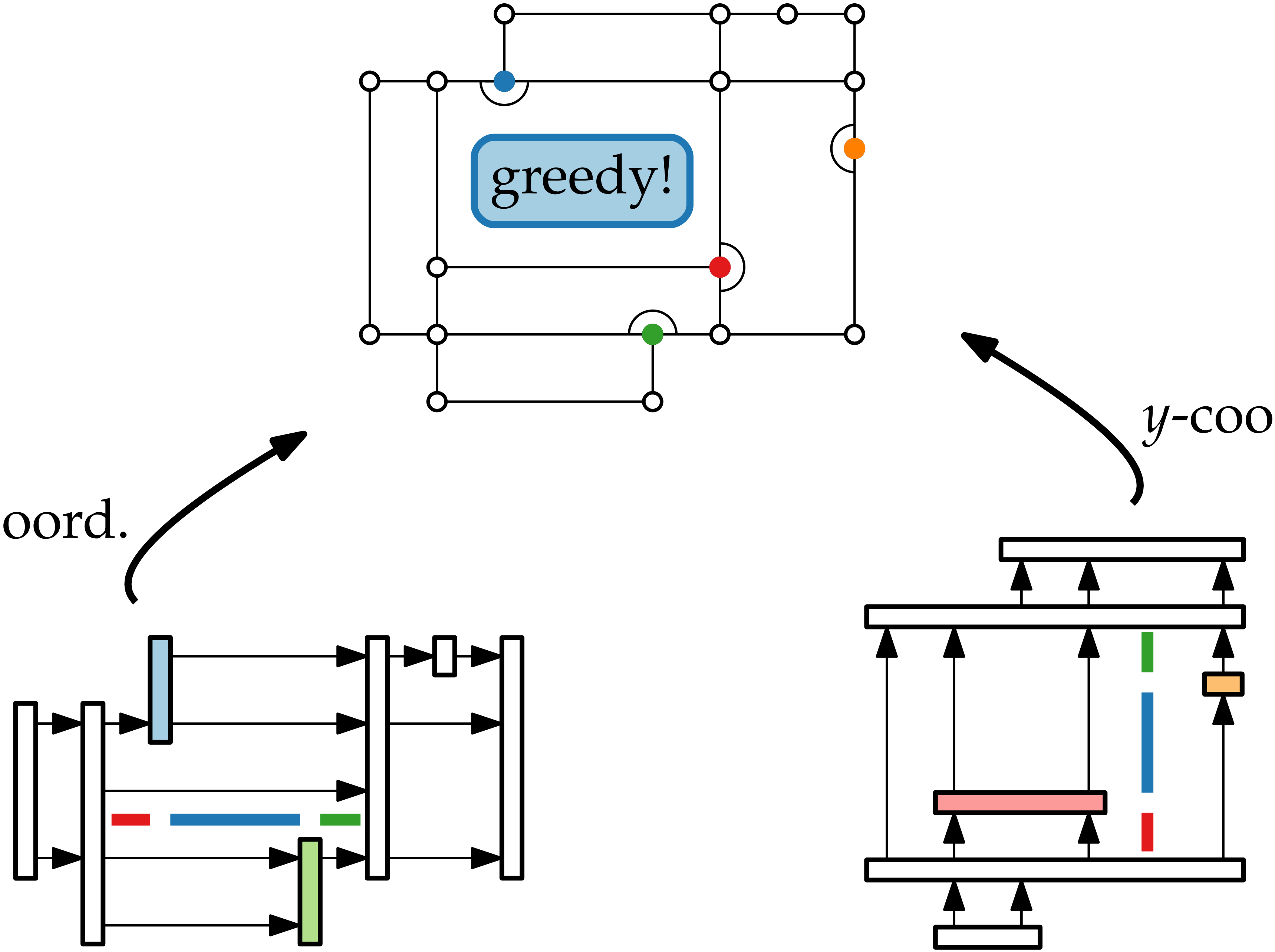
x-coord.

y-coord.

resolve *x*-conflicts

resolve *y*-conflicts

x- and y-conflicts independent



greedy!

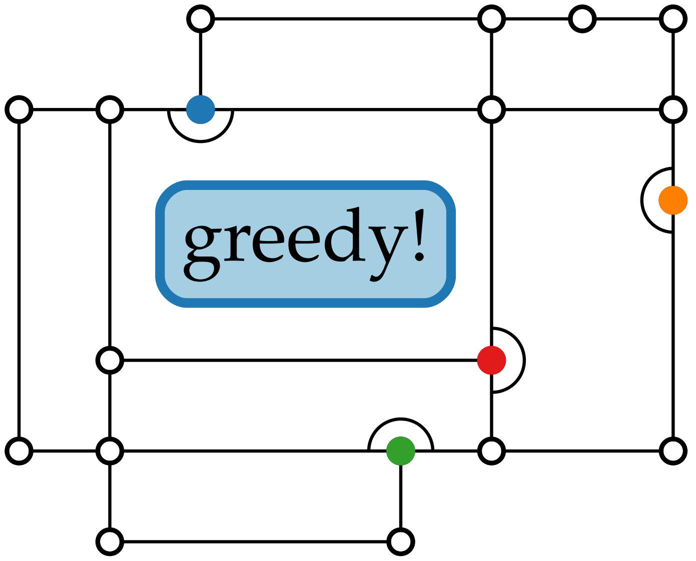
x-coord.

y-coord.

resolve x-conflicts

resolve y-conflicts

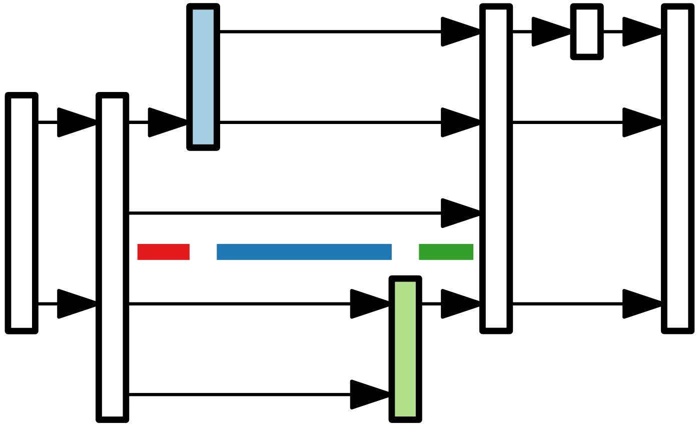
x- and y-conflicts independent



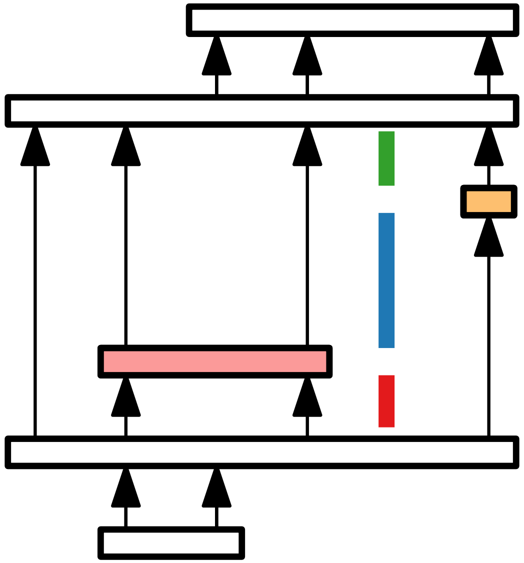
⇒ only consider x-conflicts

x-coord.

y-coord.

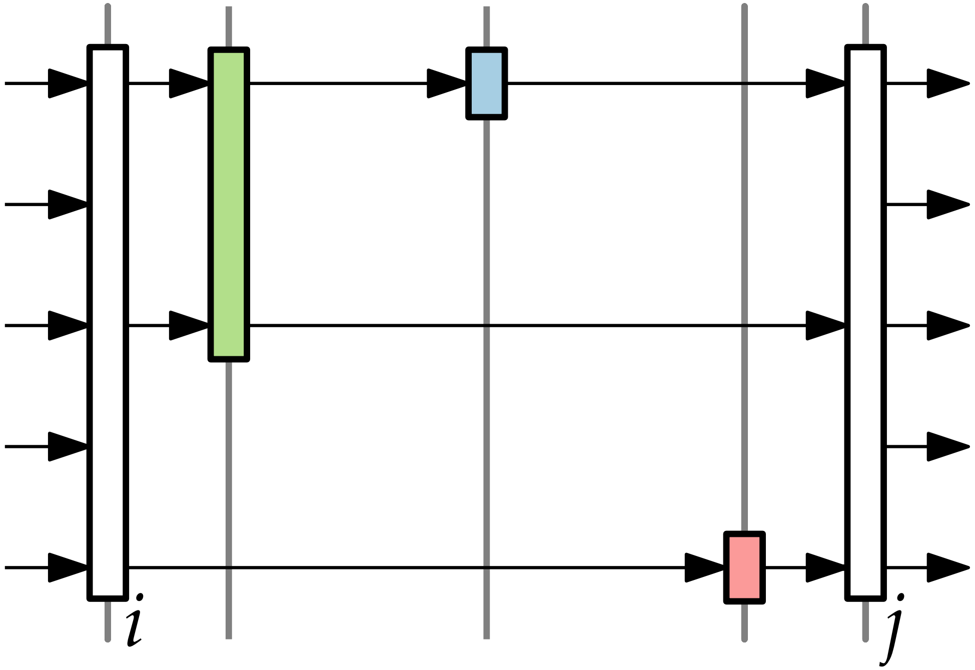


resolve x-conflicts

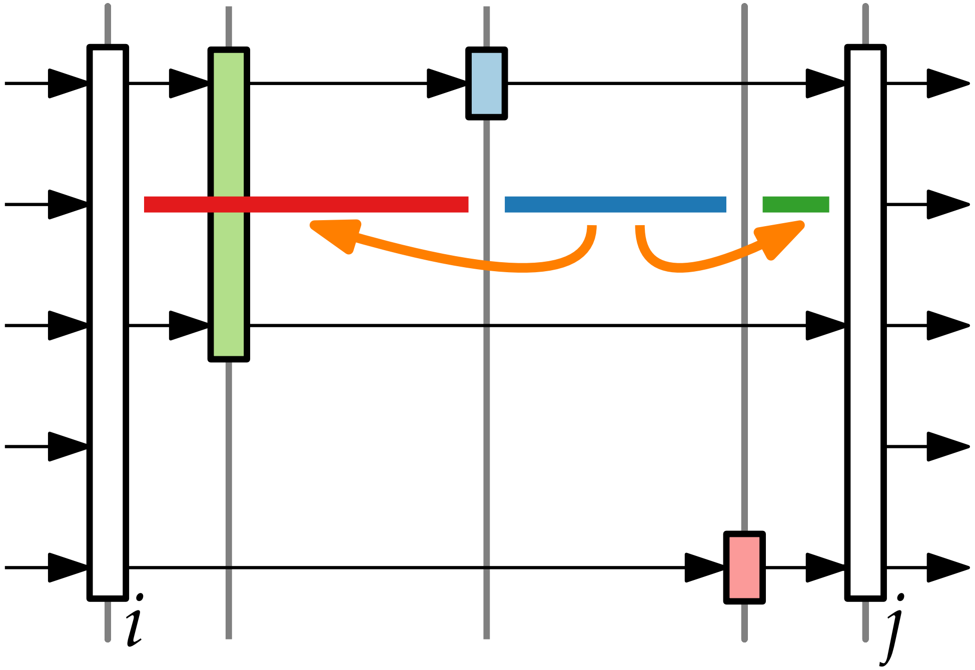


resolve y-conflicts

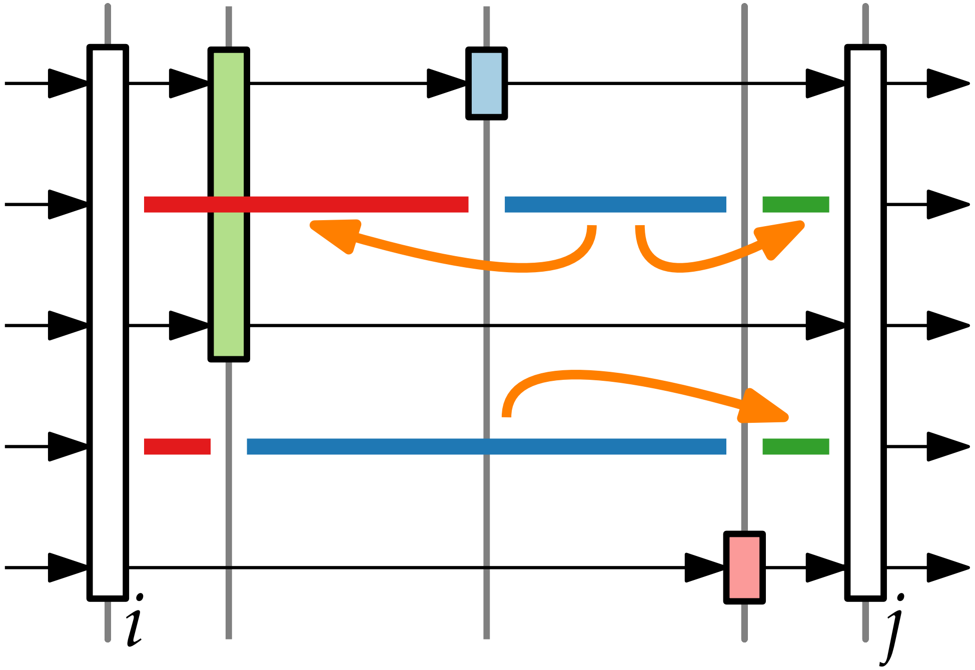
ALL conflicts?



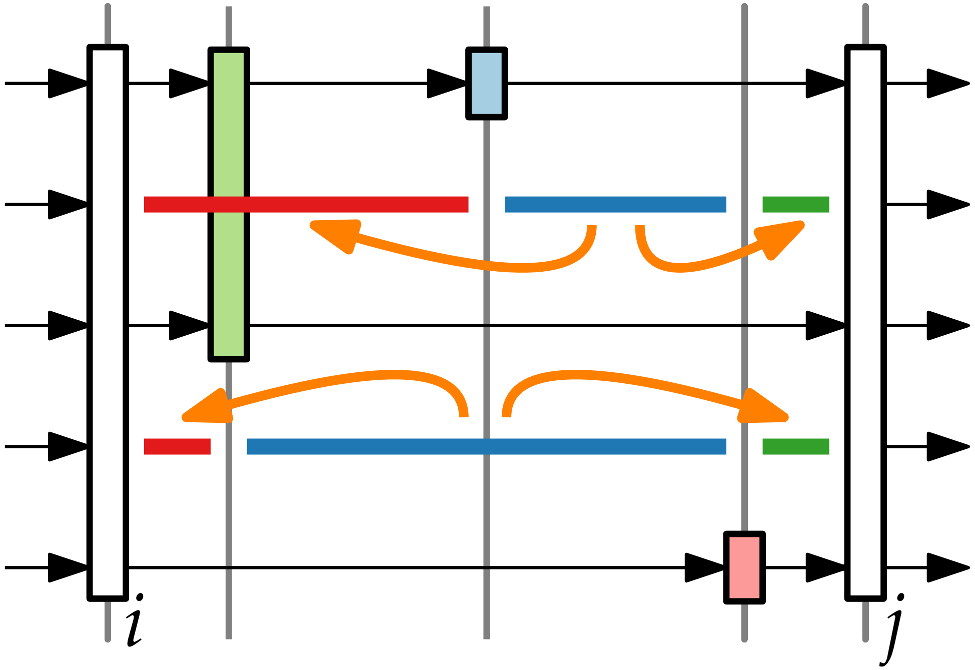
ALL conflicts?



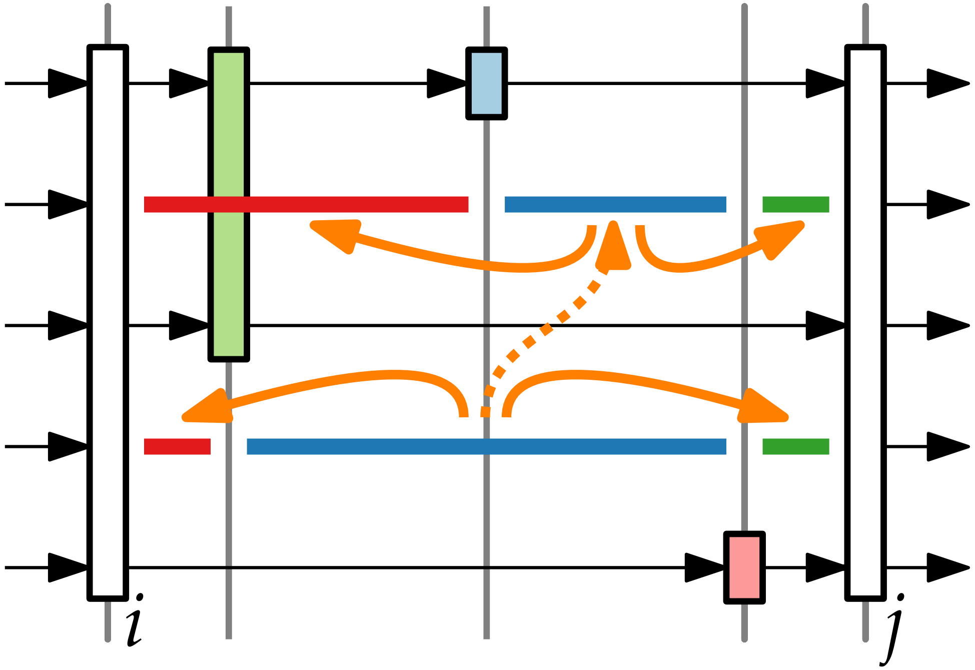
ALL conflicts?



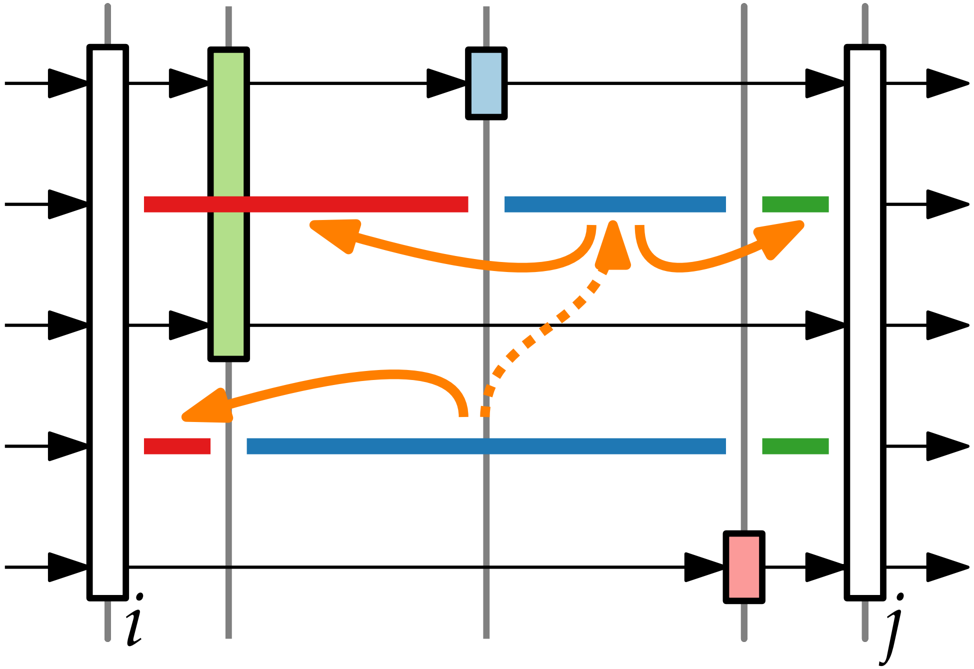
ALL conflicts?



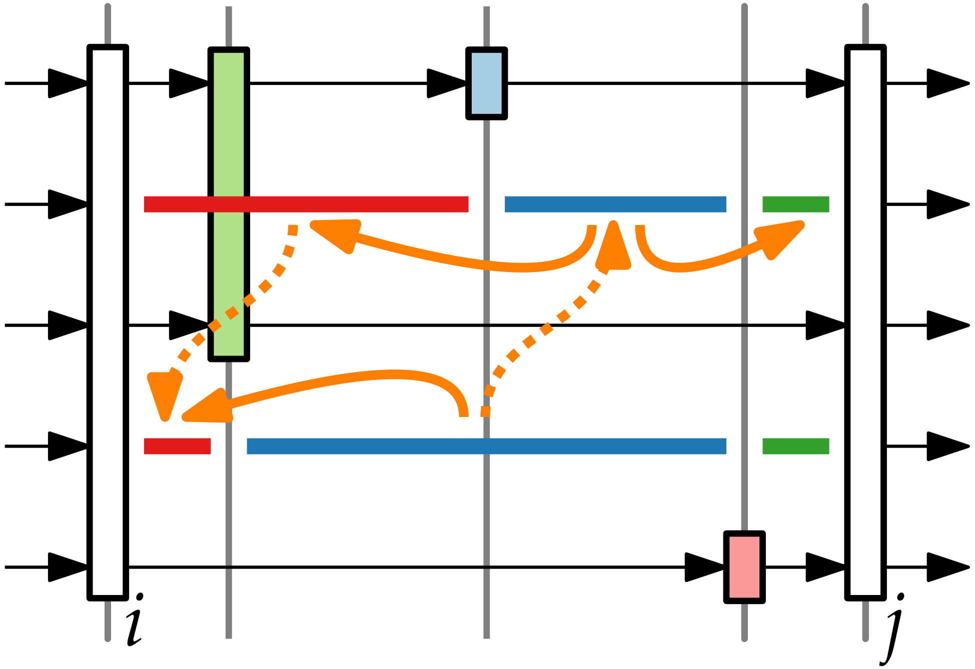
ALL conflicts?



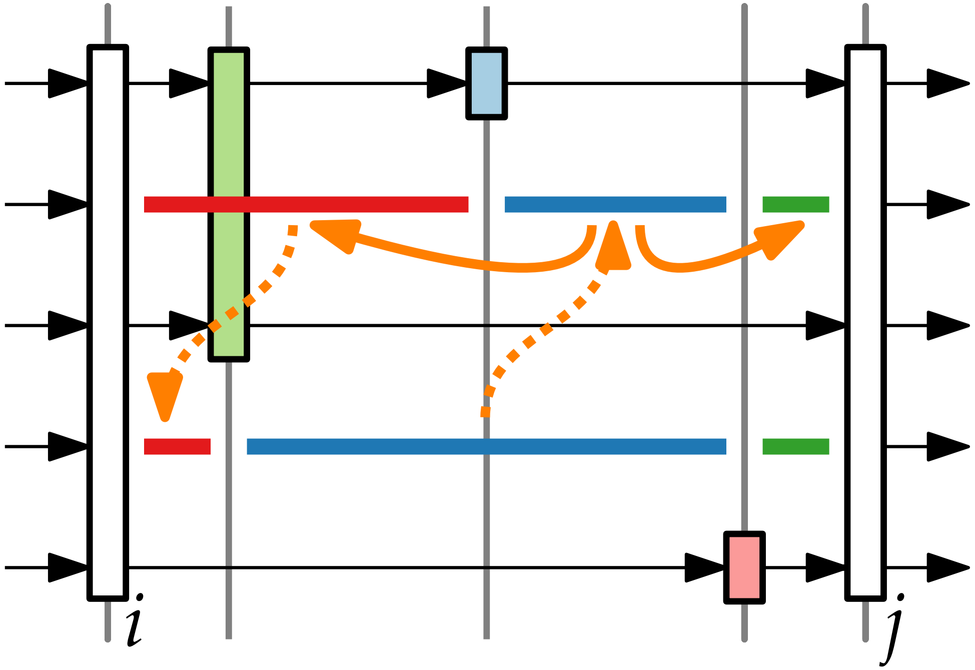
ALL conflicts?



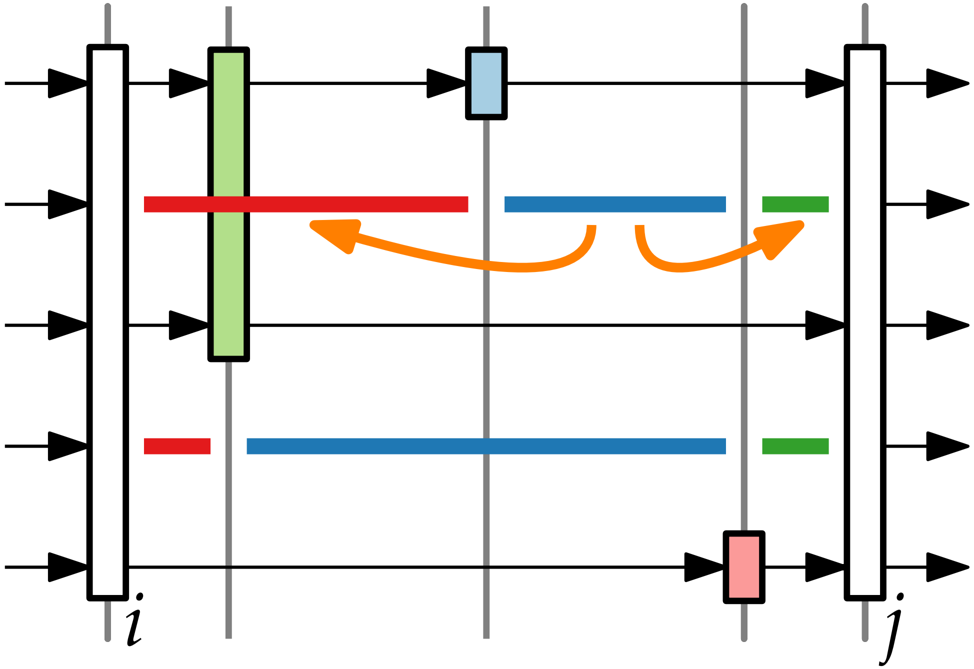
ALL conflicts?



ALL conflicts?

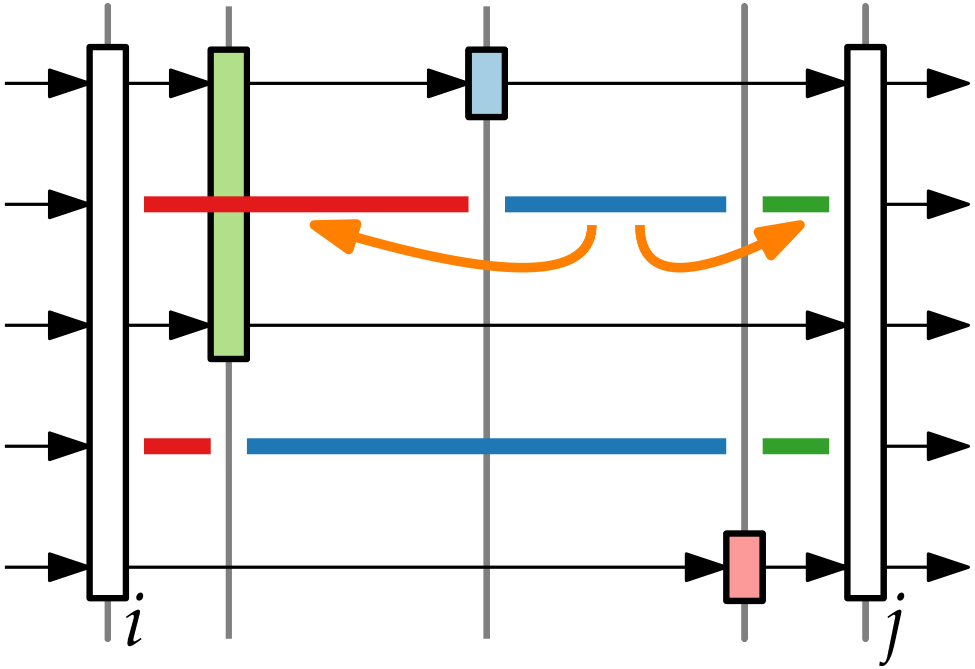


ALL conflicts?



🟦🟥 *dominates* 🟩🟥

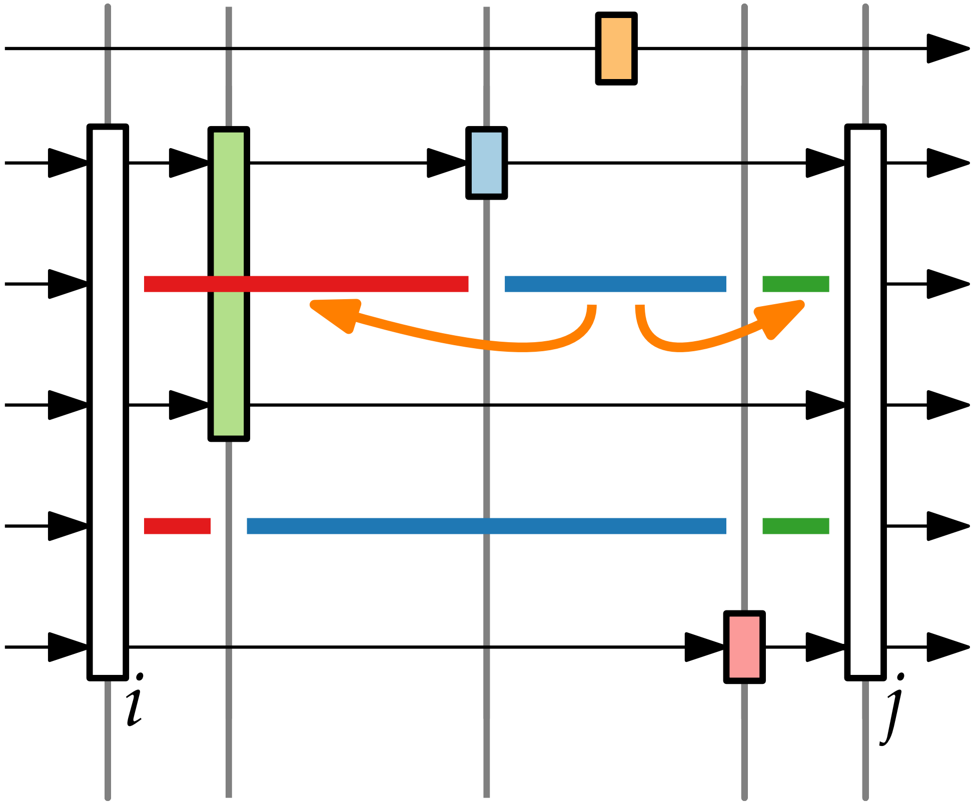
ALL conflicts?



🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

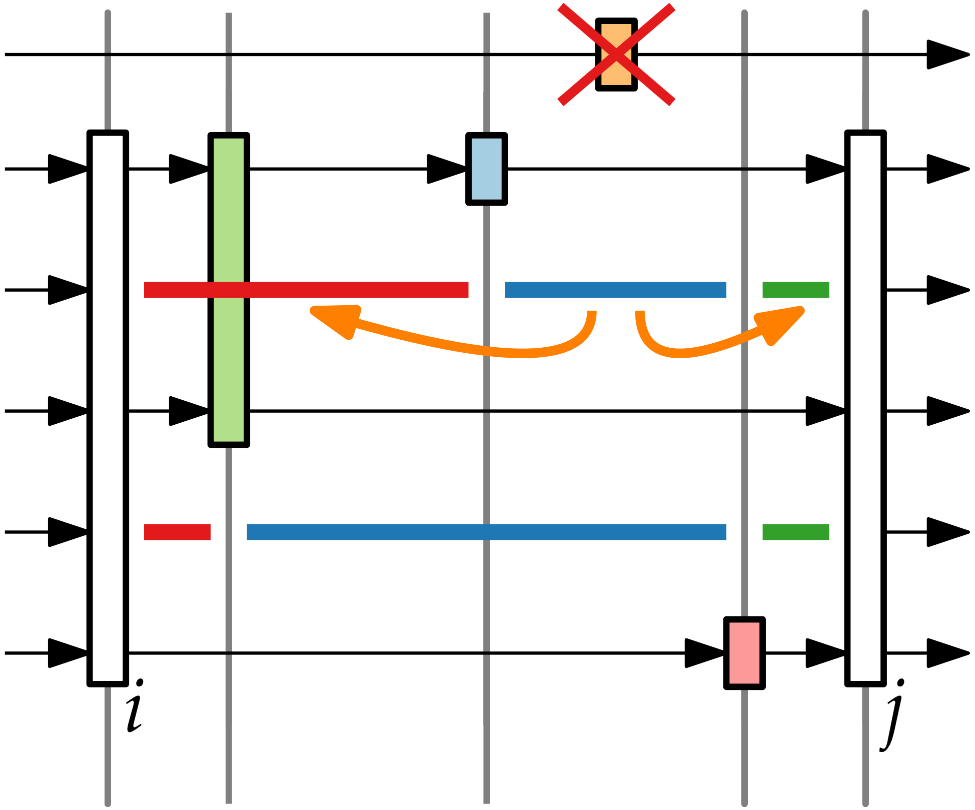
ALL conflicts?



🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

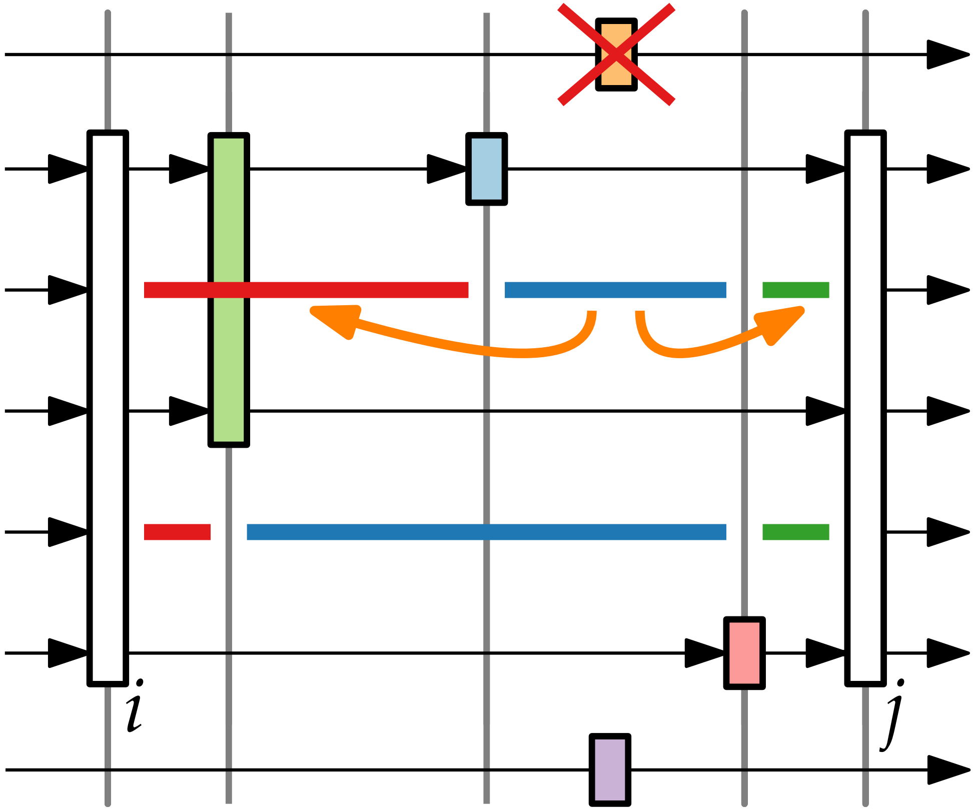
ALL conflicts?



🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

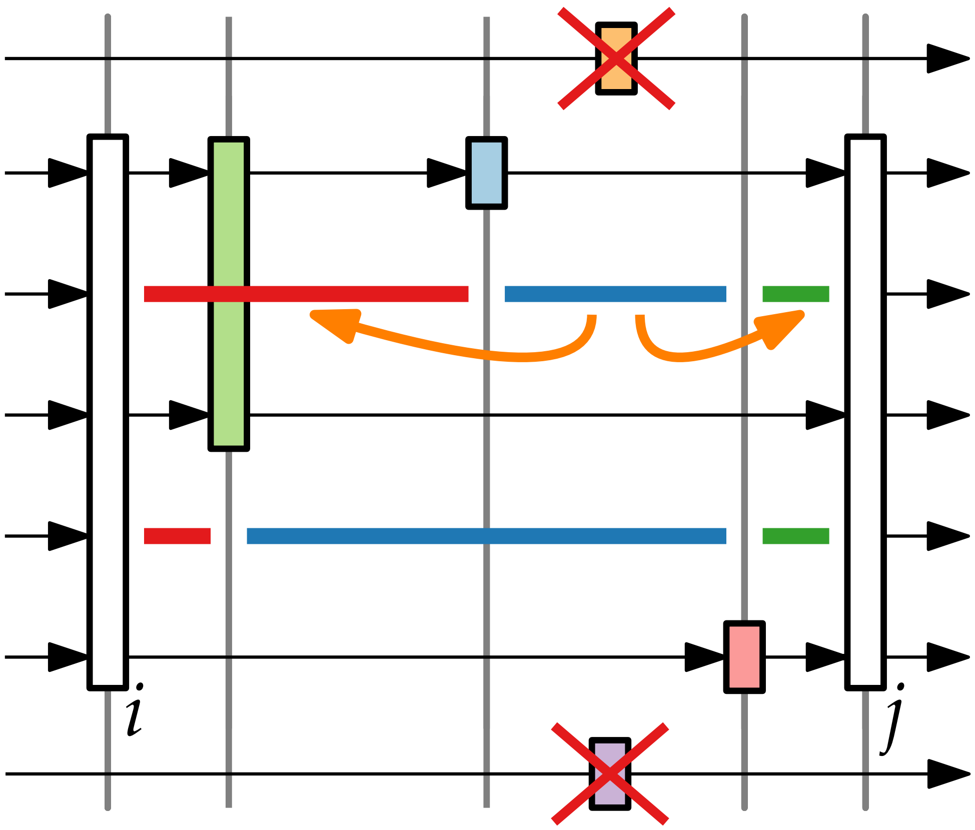
ALL conflicts?



🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

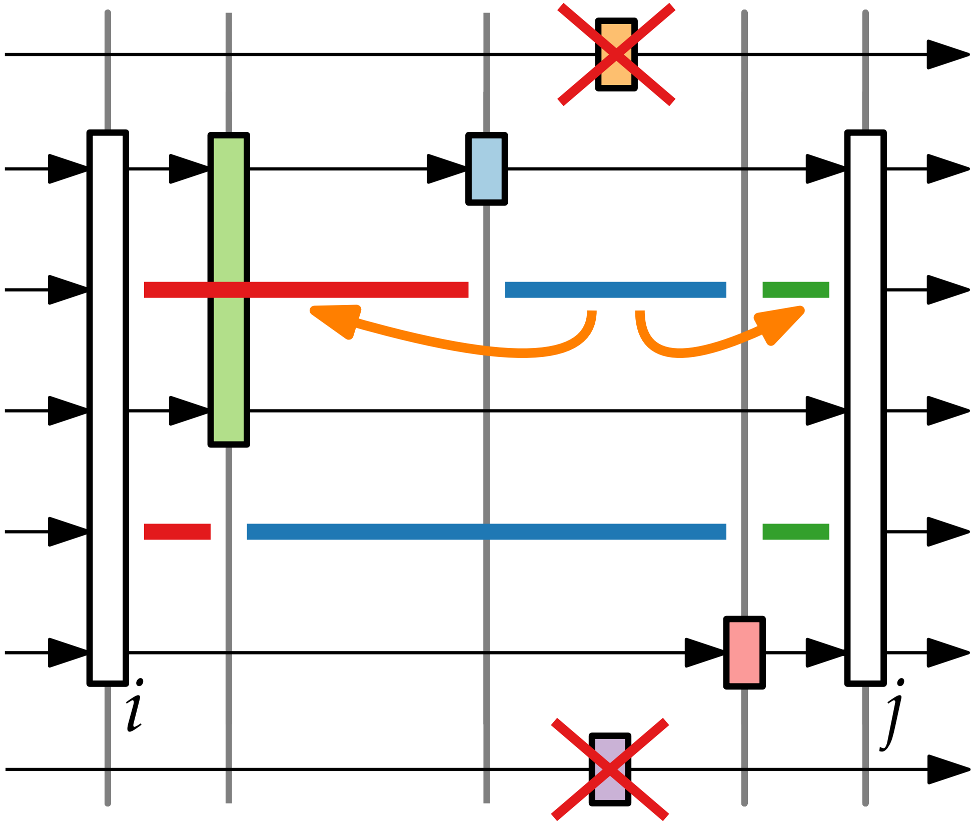
ALL conflicts?



🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

ALL conflicts?

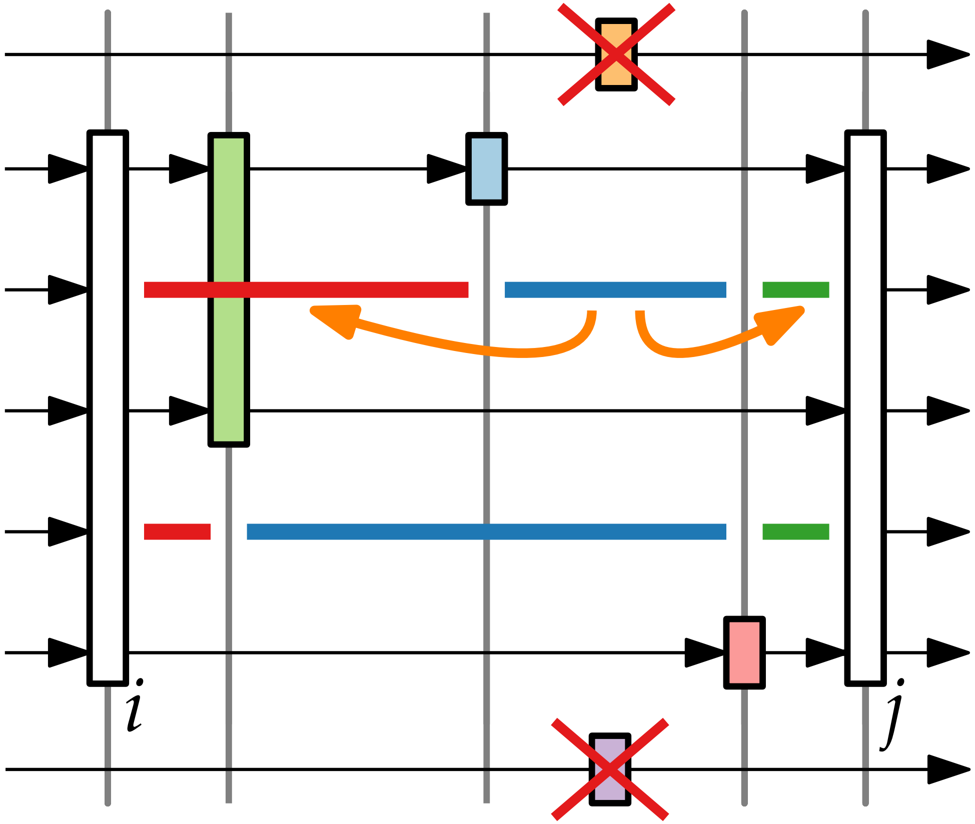


🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

Dominating conflicts are consecutive in *st*-order

ALL conflicts?



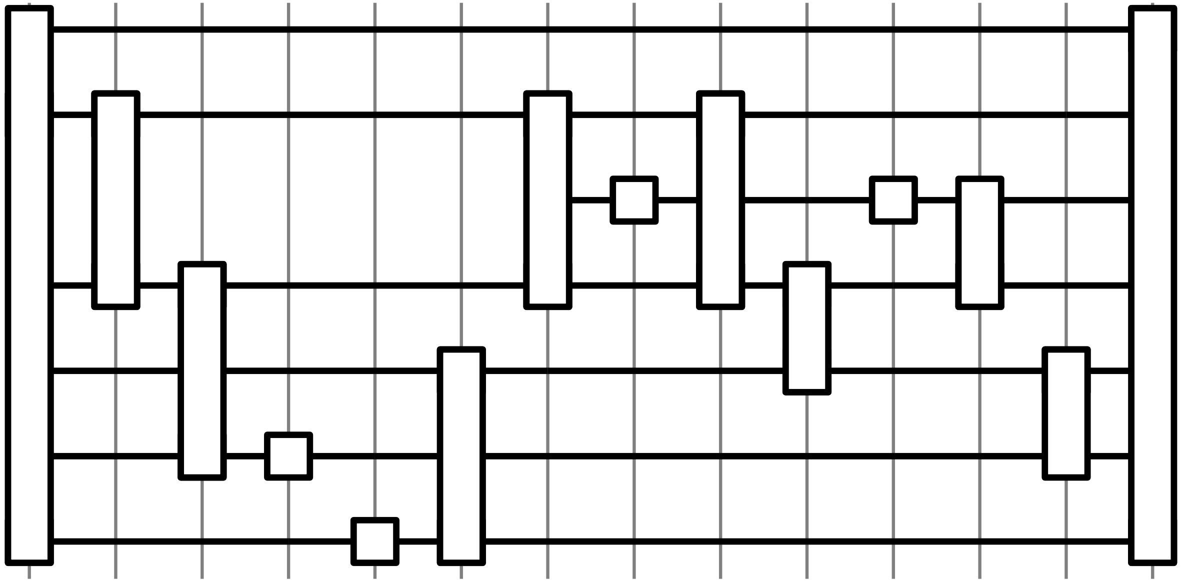
🟦🟥 *dominates* 🟩🟥

Nothing dominates 🟦🟥 ⇒ 🟦🟥 is *dominating*

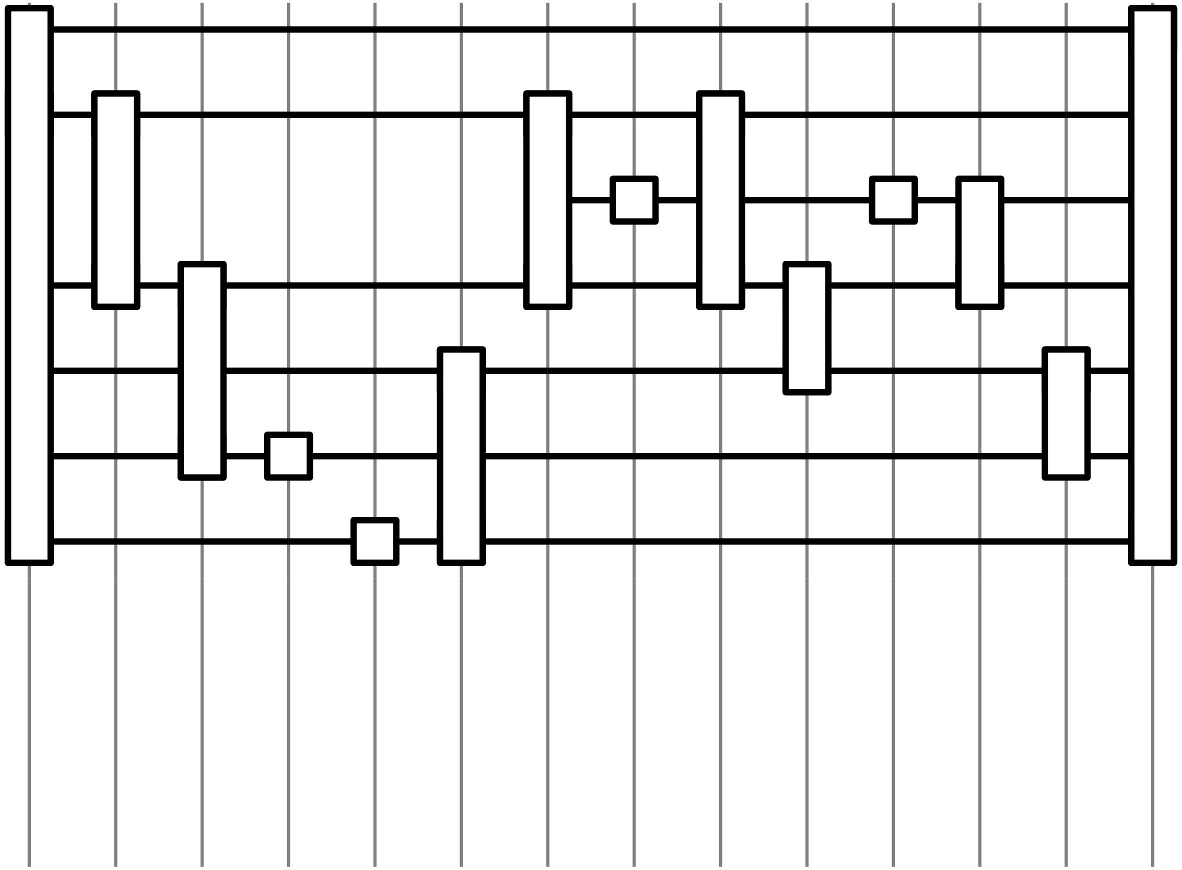
Dominating conflicts are consecutive in *st*-order

At most 1 dominating conflict per vertex

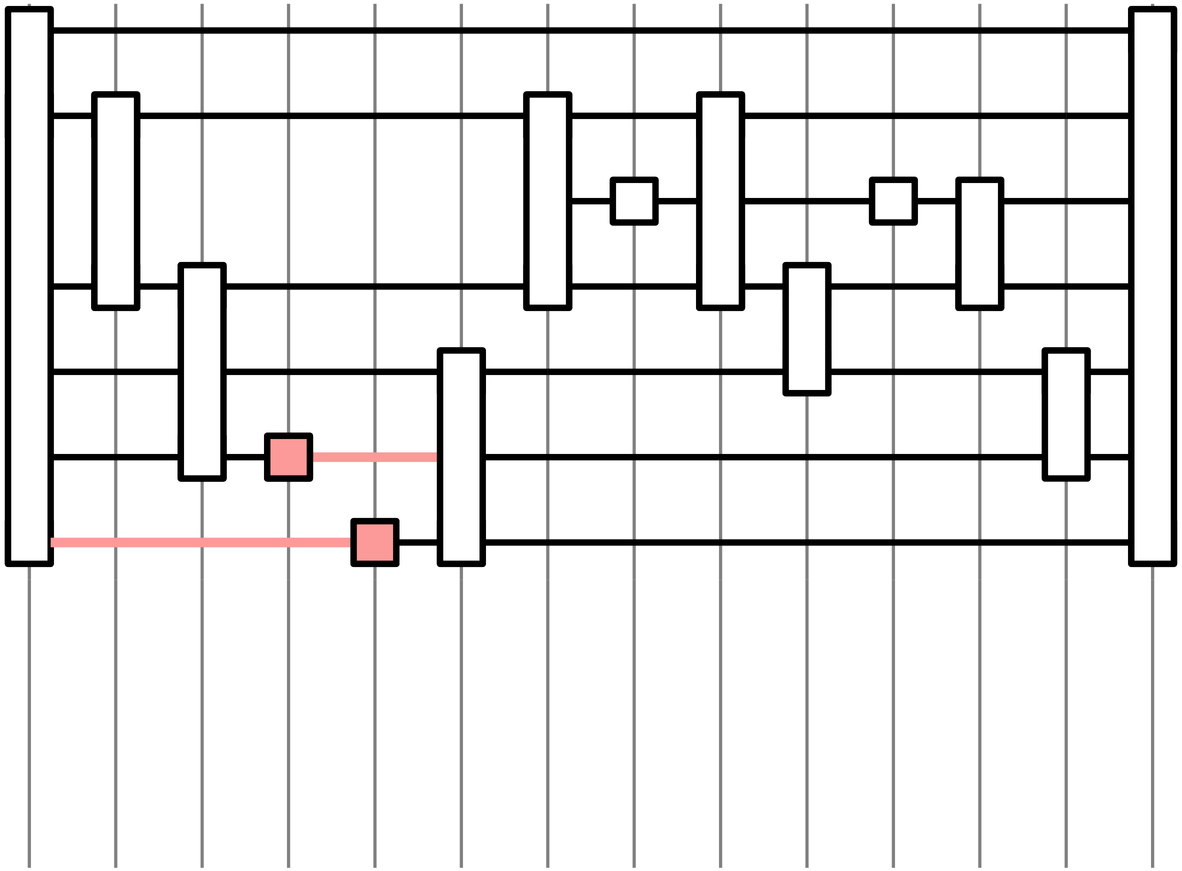
Constraints



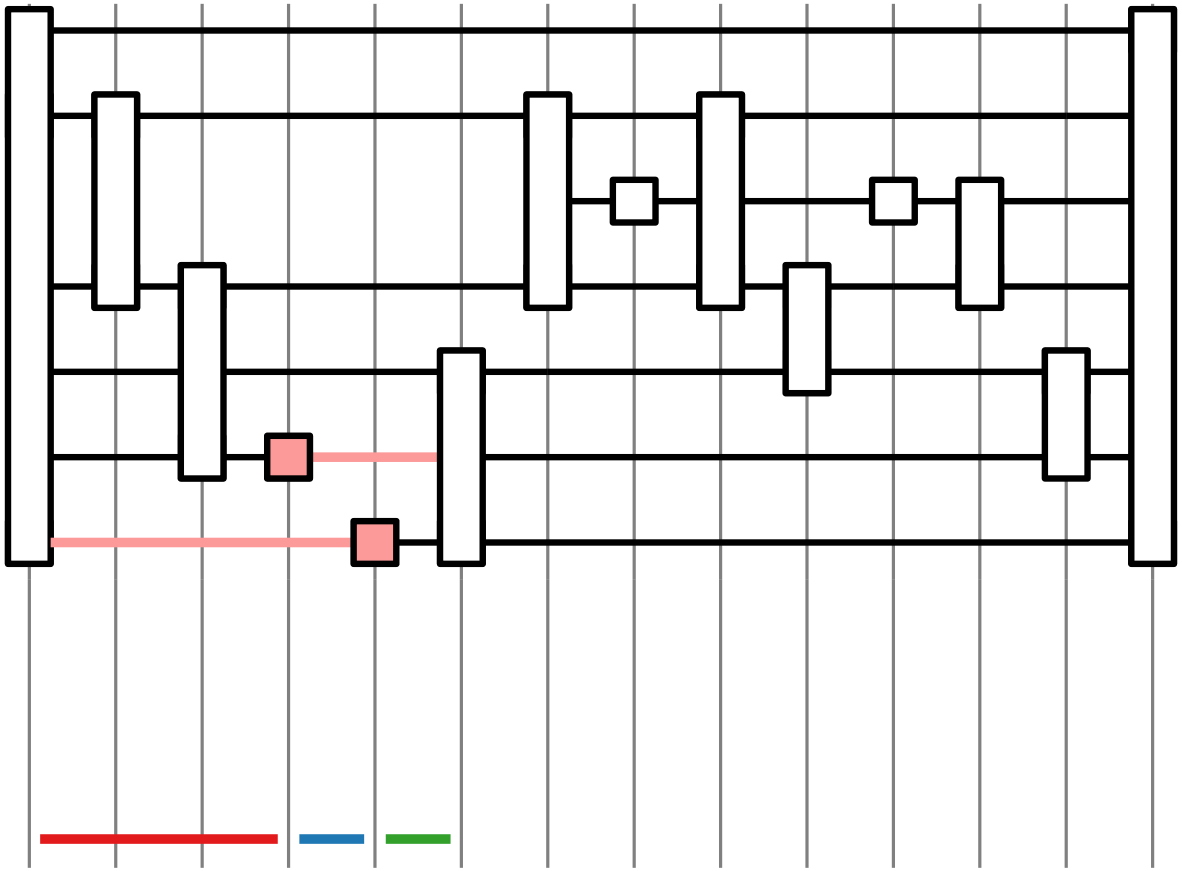
Constraints



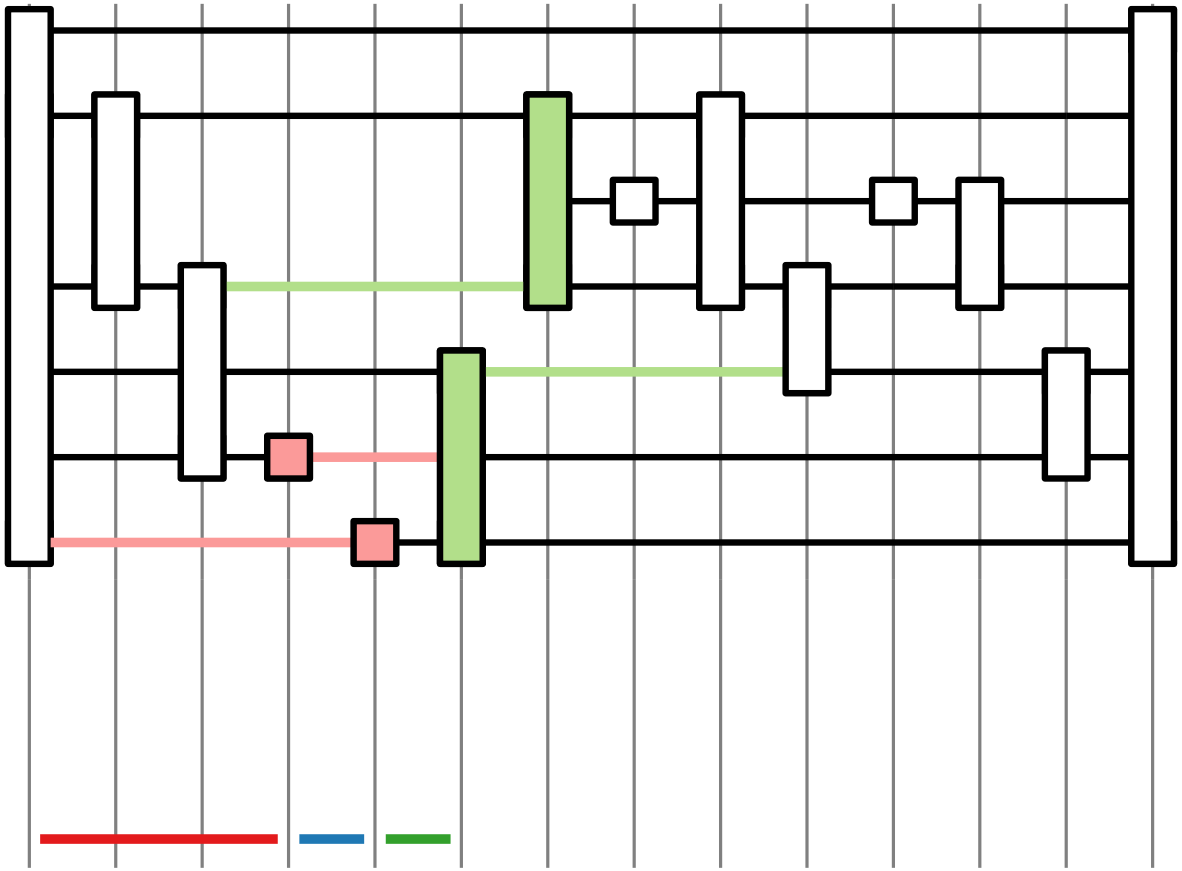
Constraints



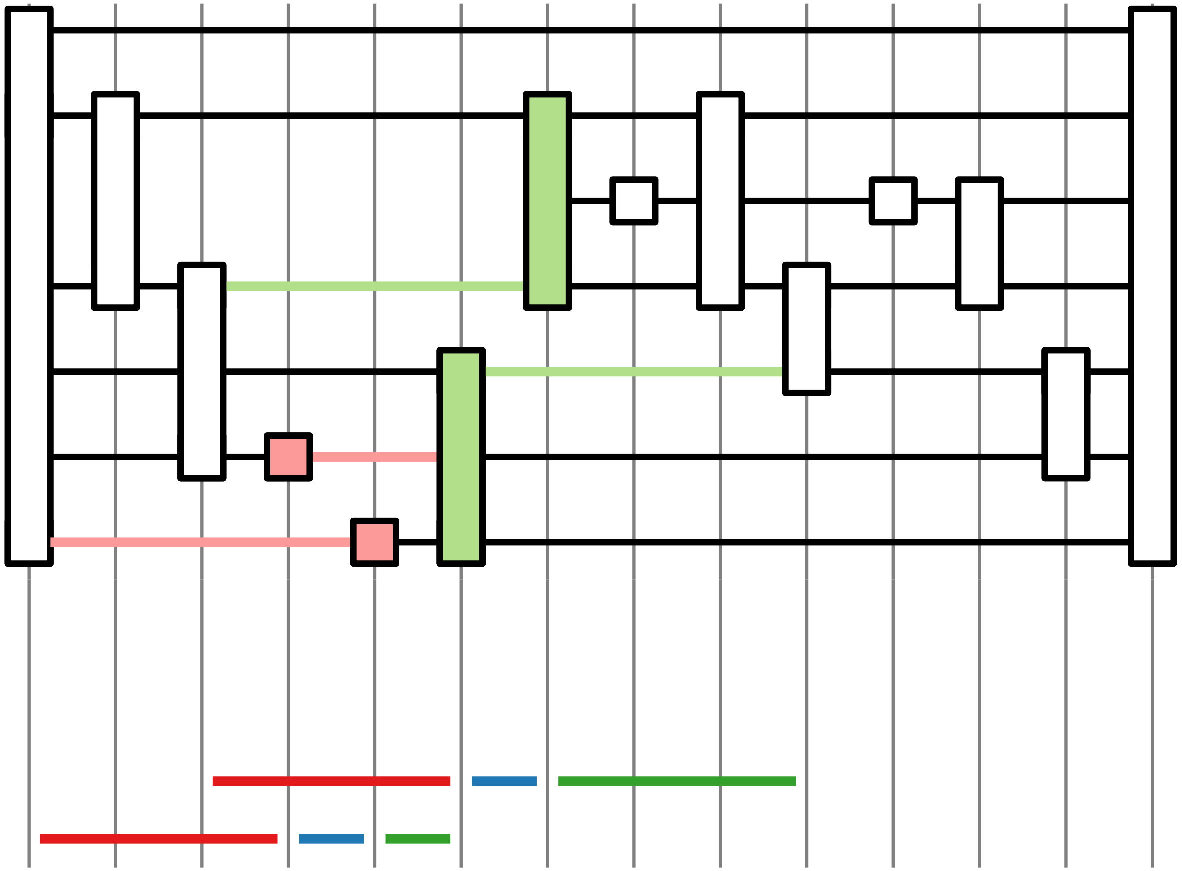
Constraints



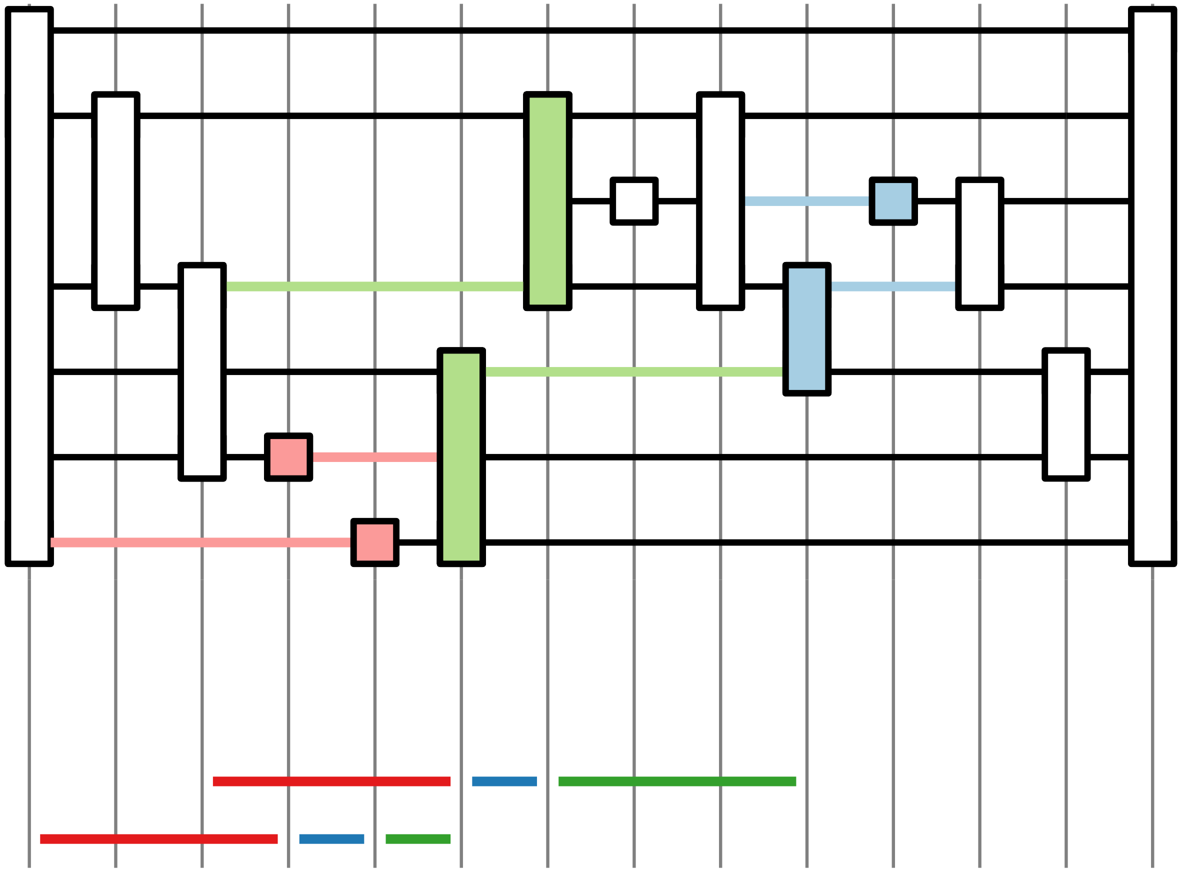
Constraints



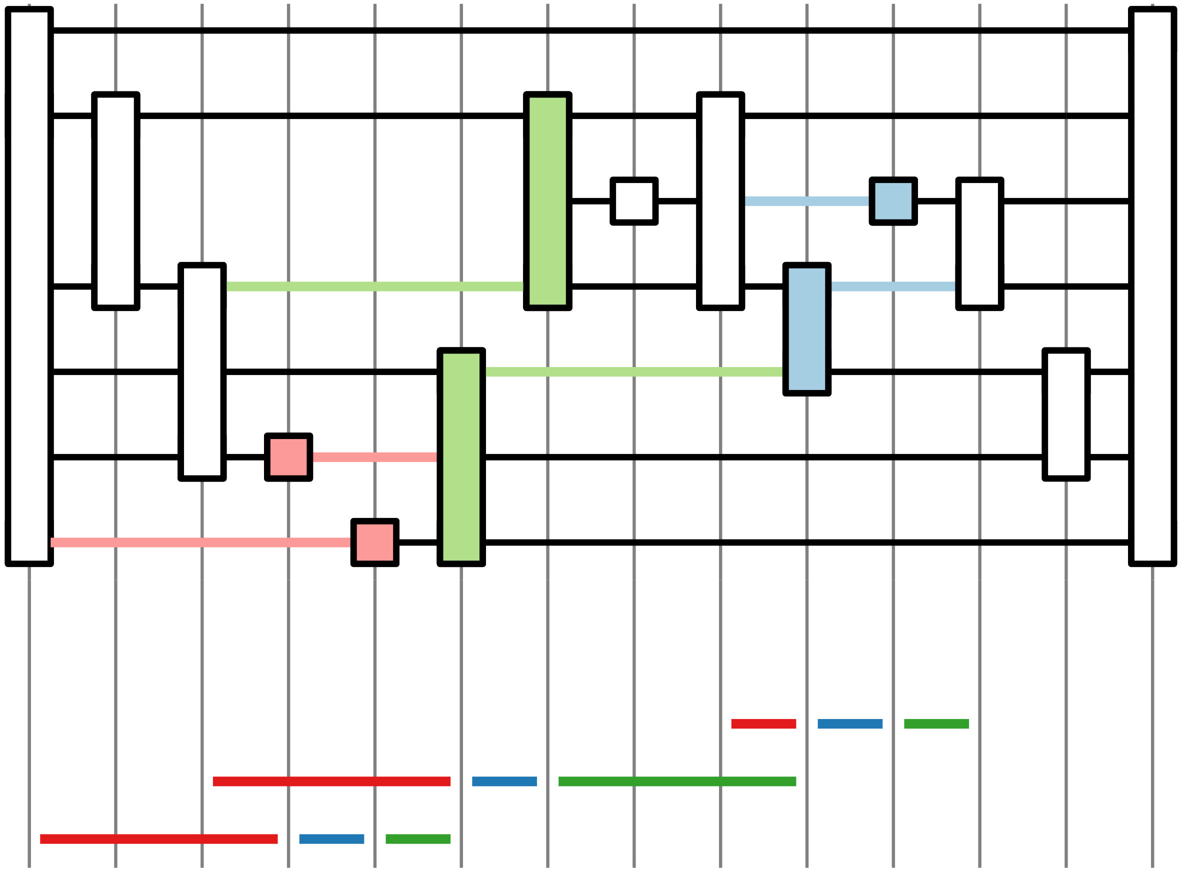
Constraints



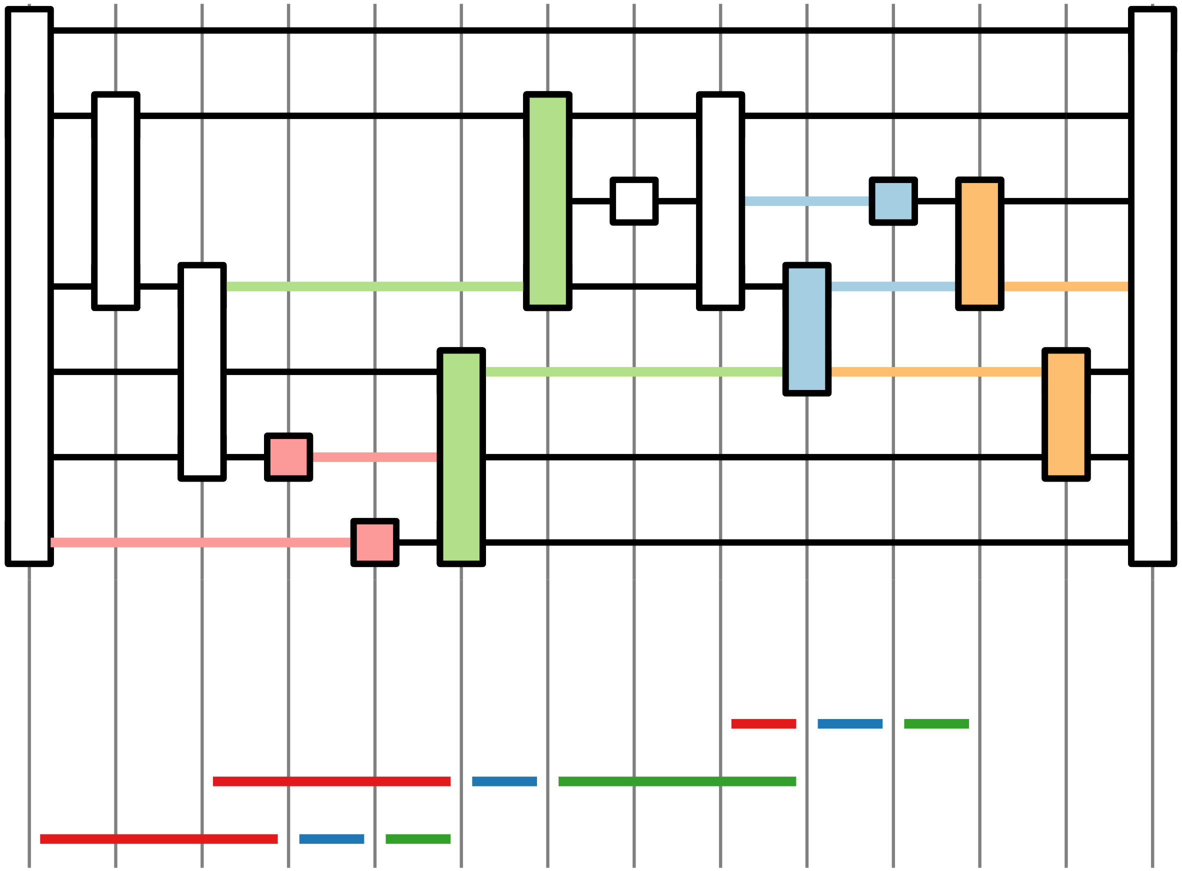
Constraints



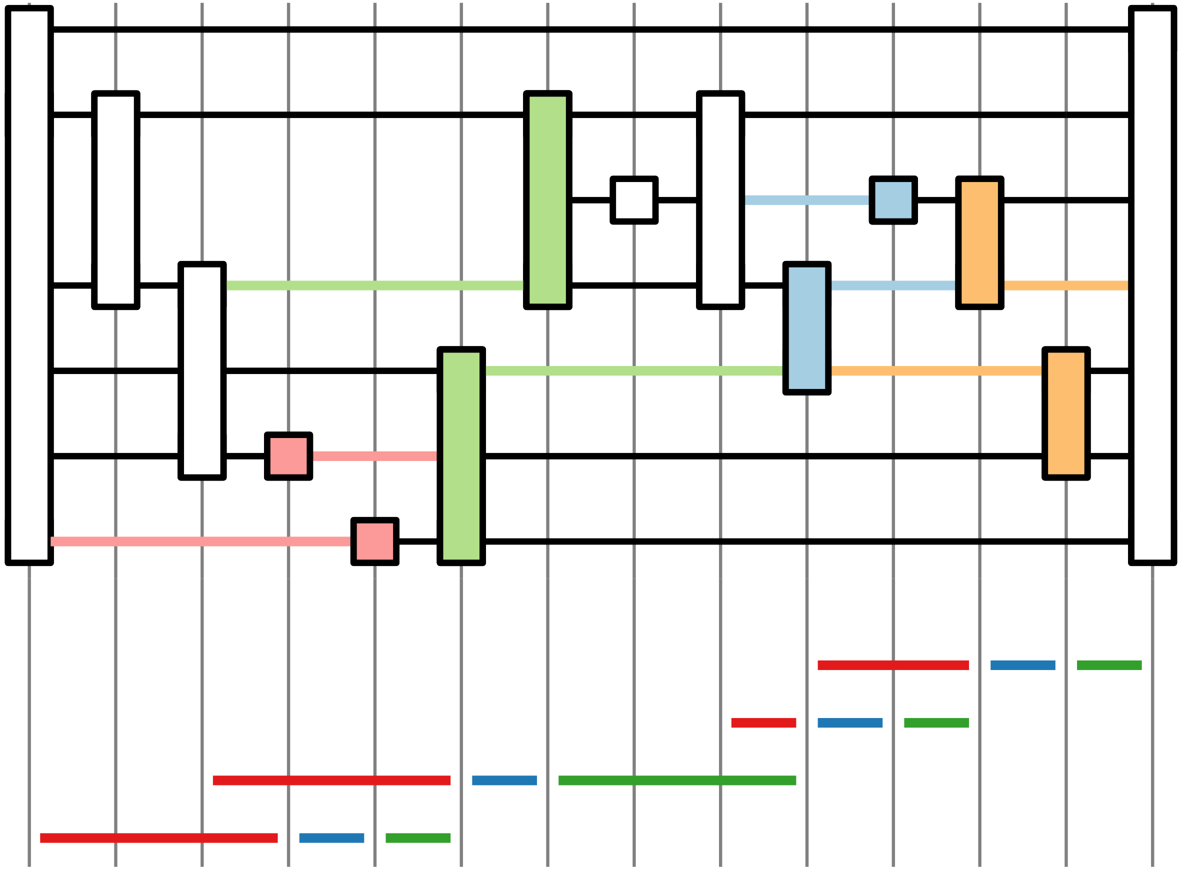
Constraints



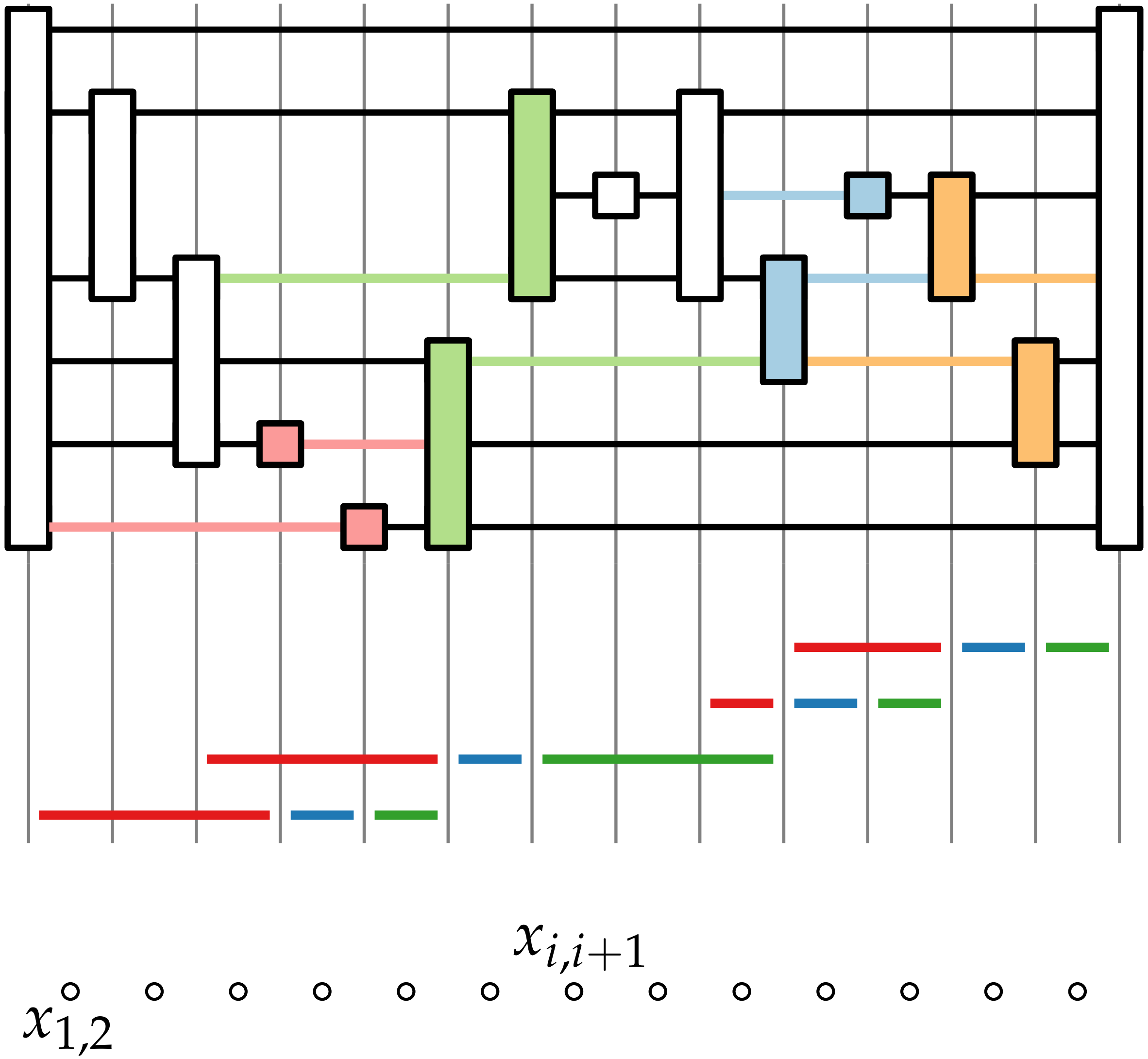
Constraints



Constraints

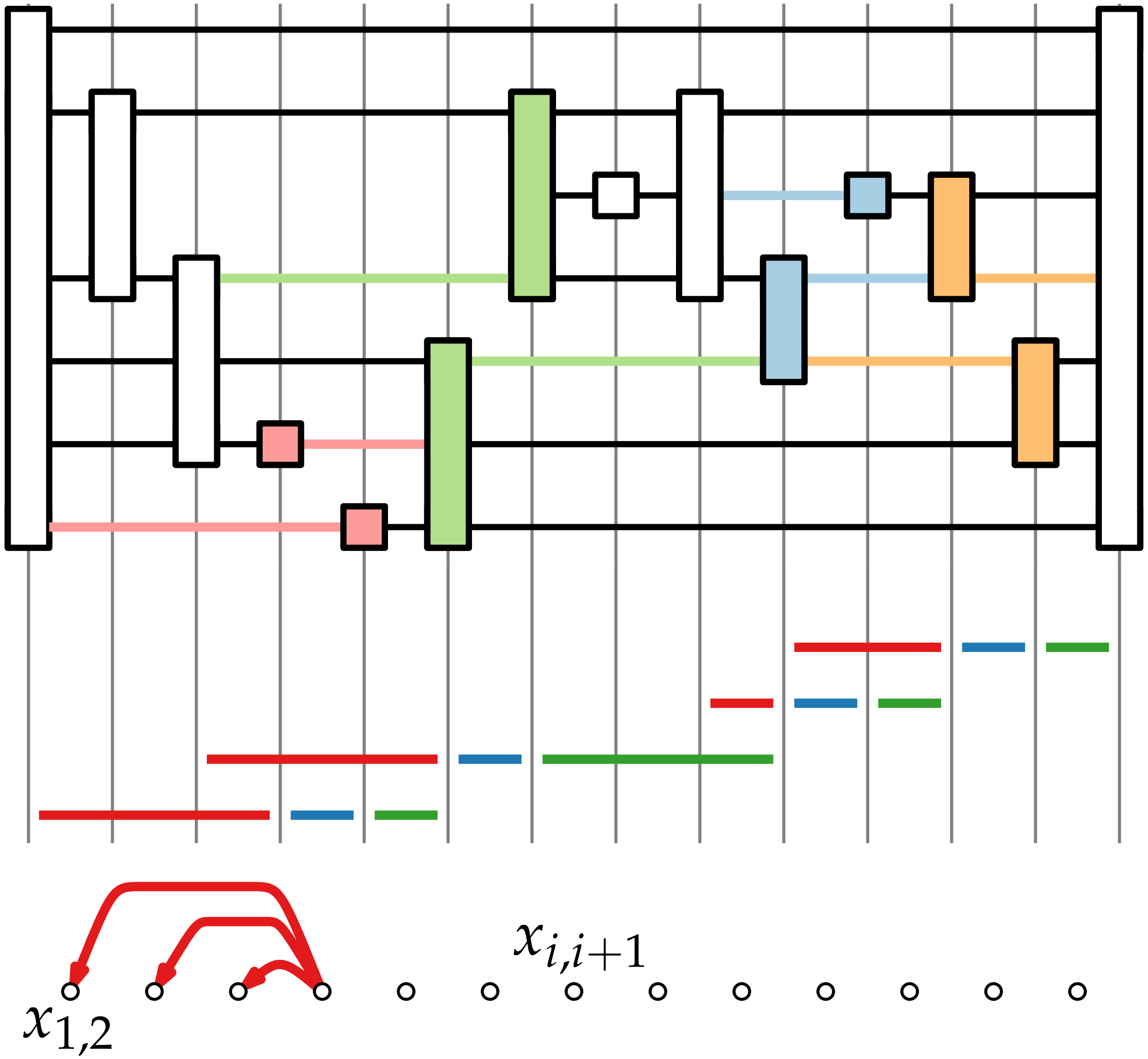


Constraints



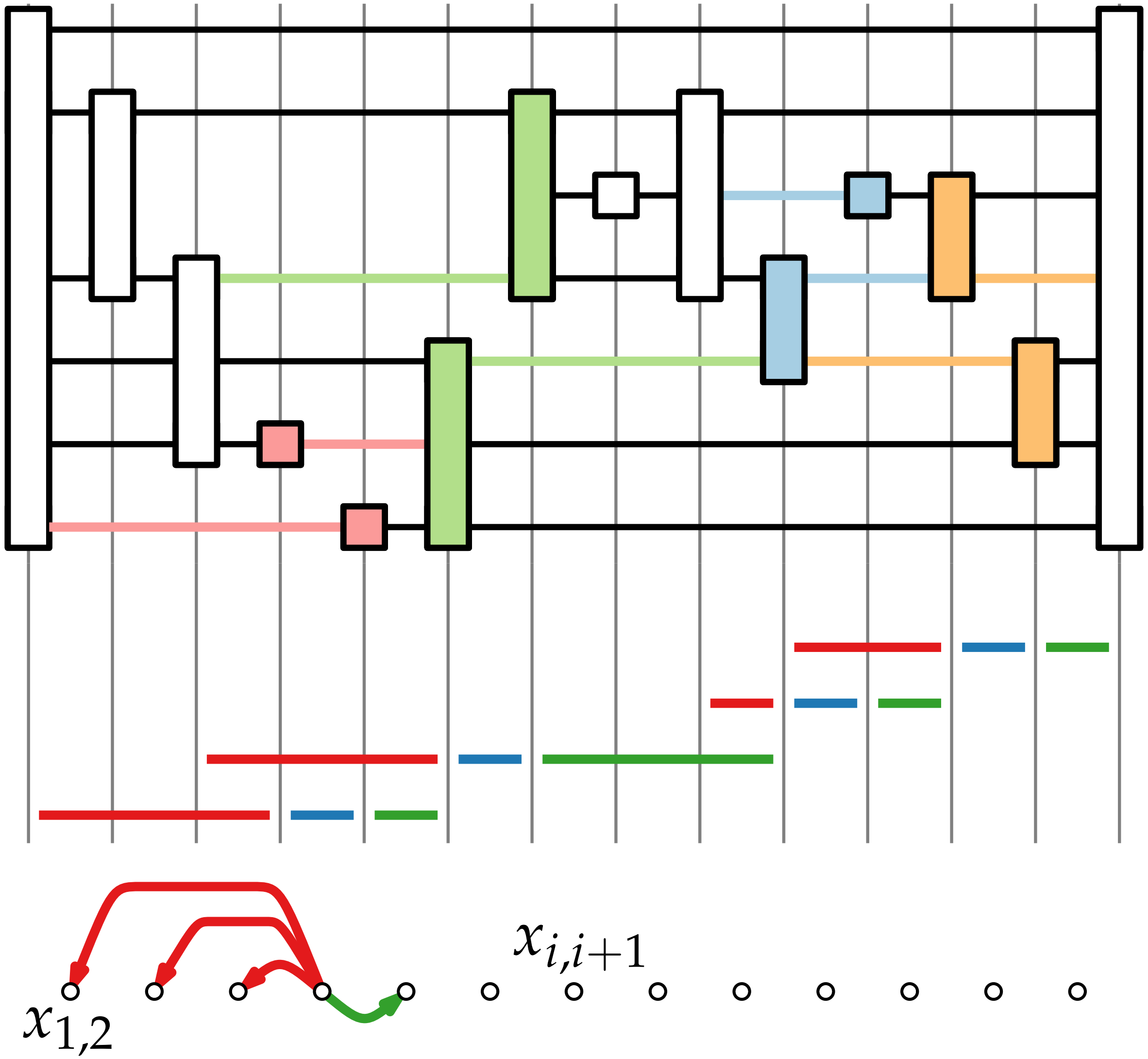
Relation Graph

Constraints



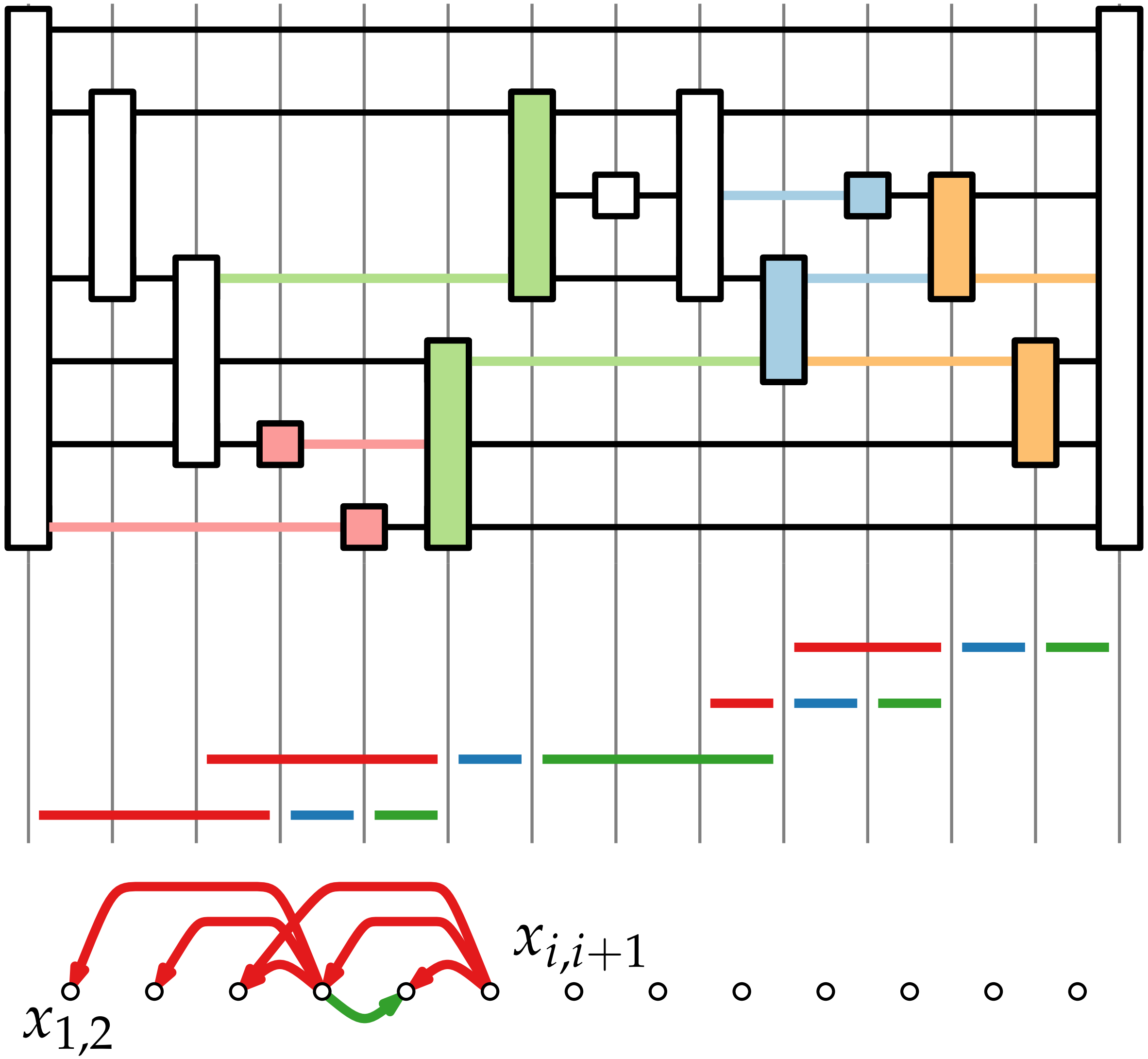
Relation Graph

Constraints



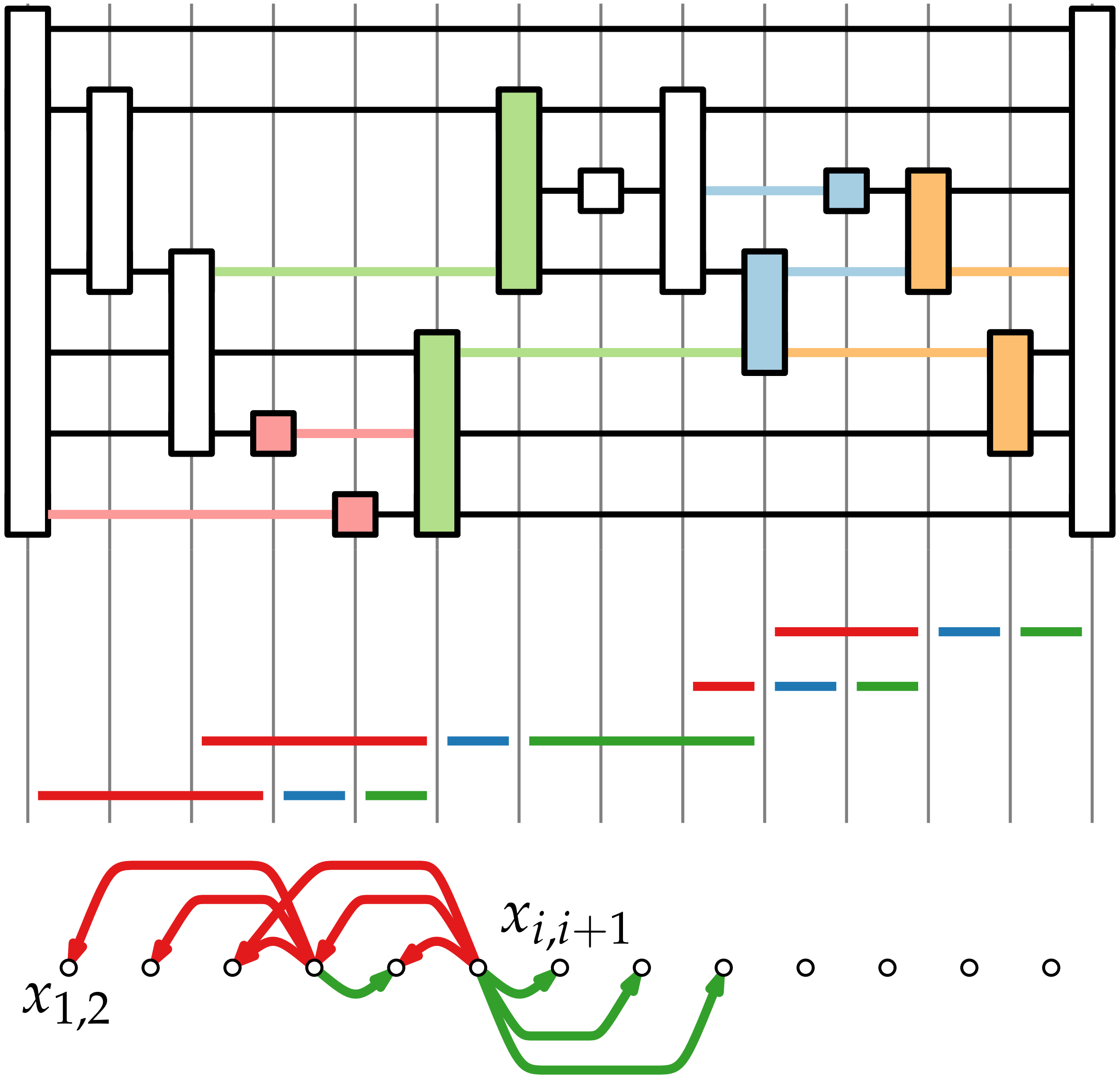
Relation Graph

Constraints



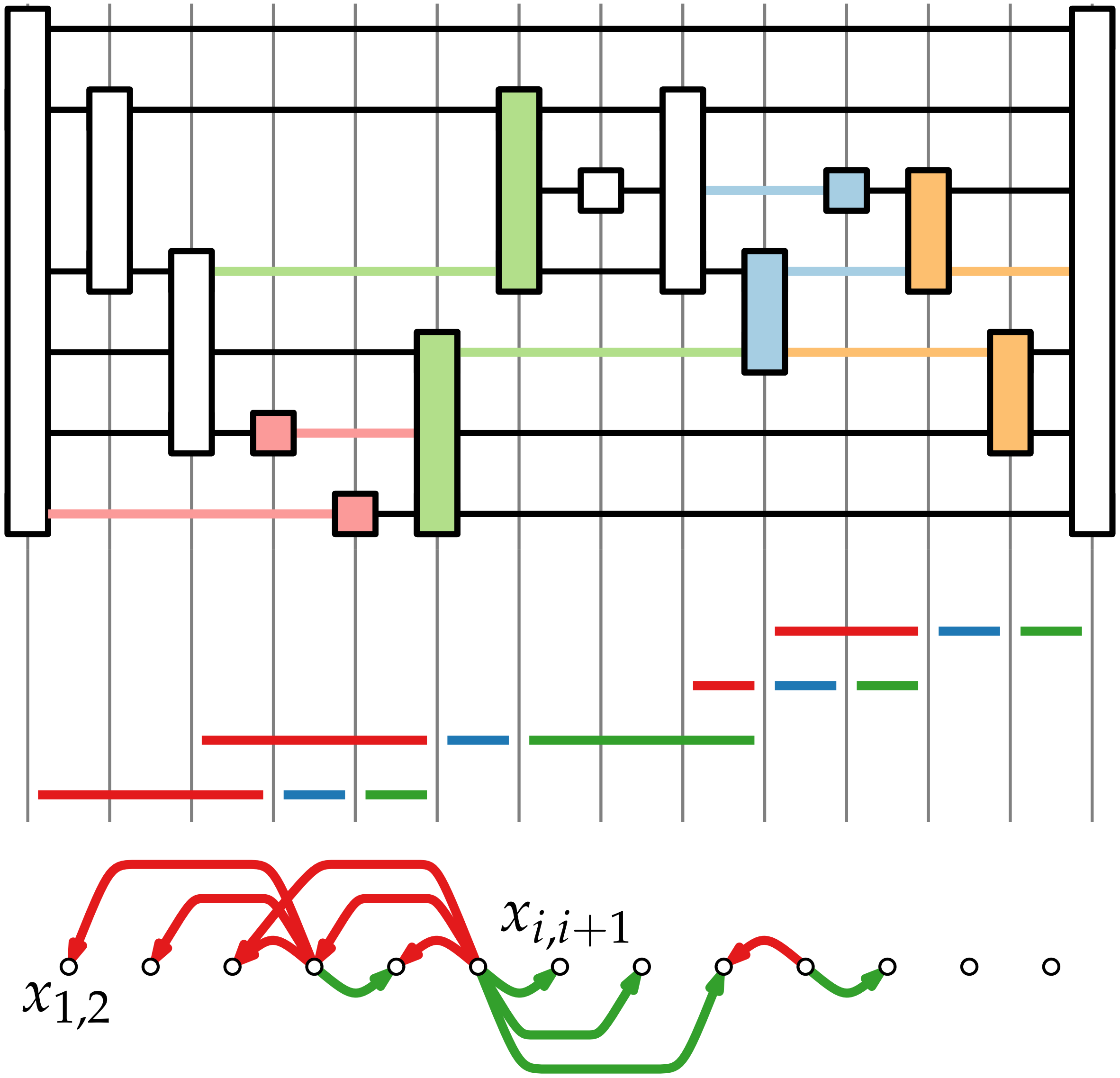
Relation Graph

Constraints



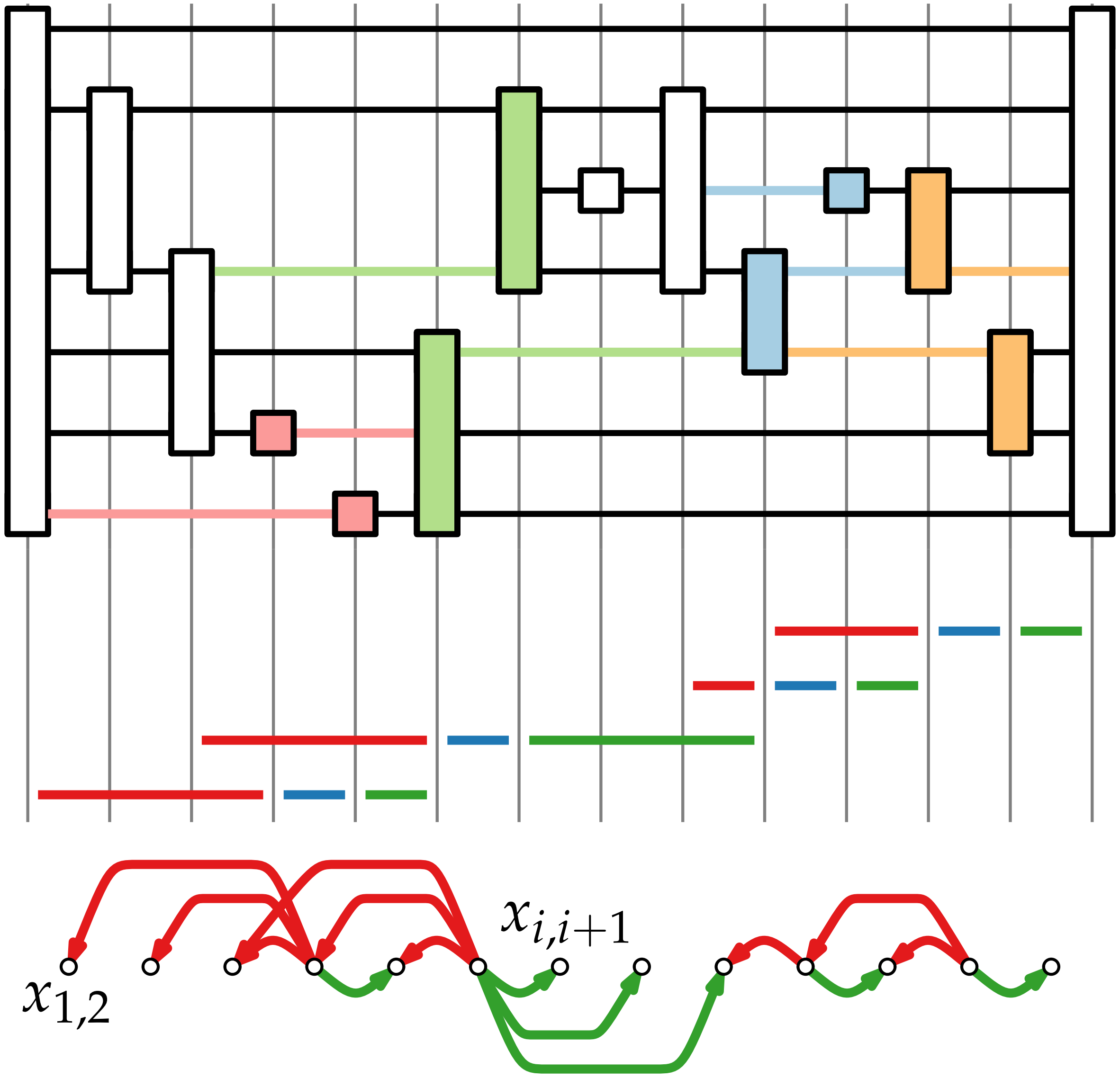
Relation Graph

Constraints



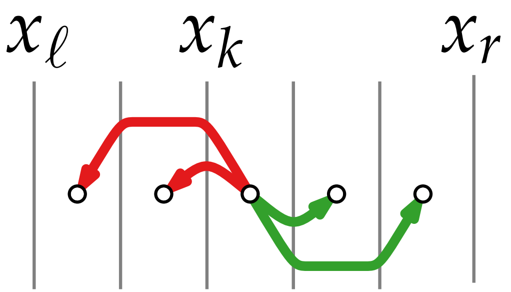
Relation Graph

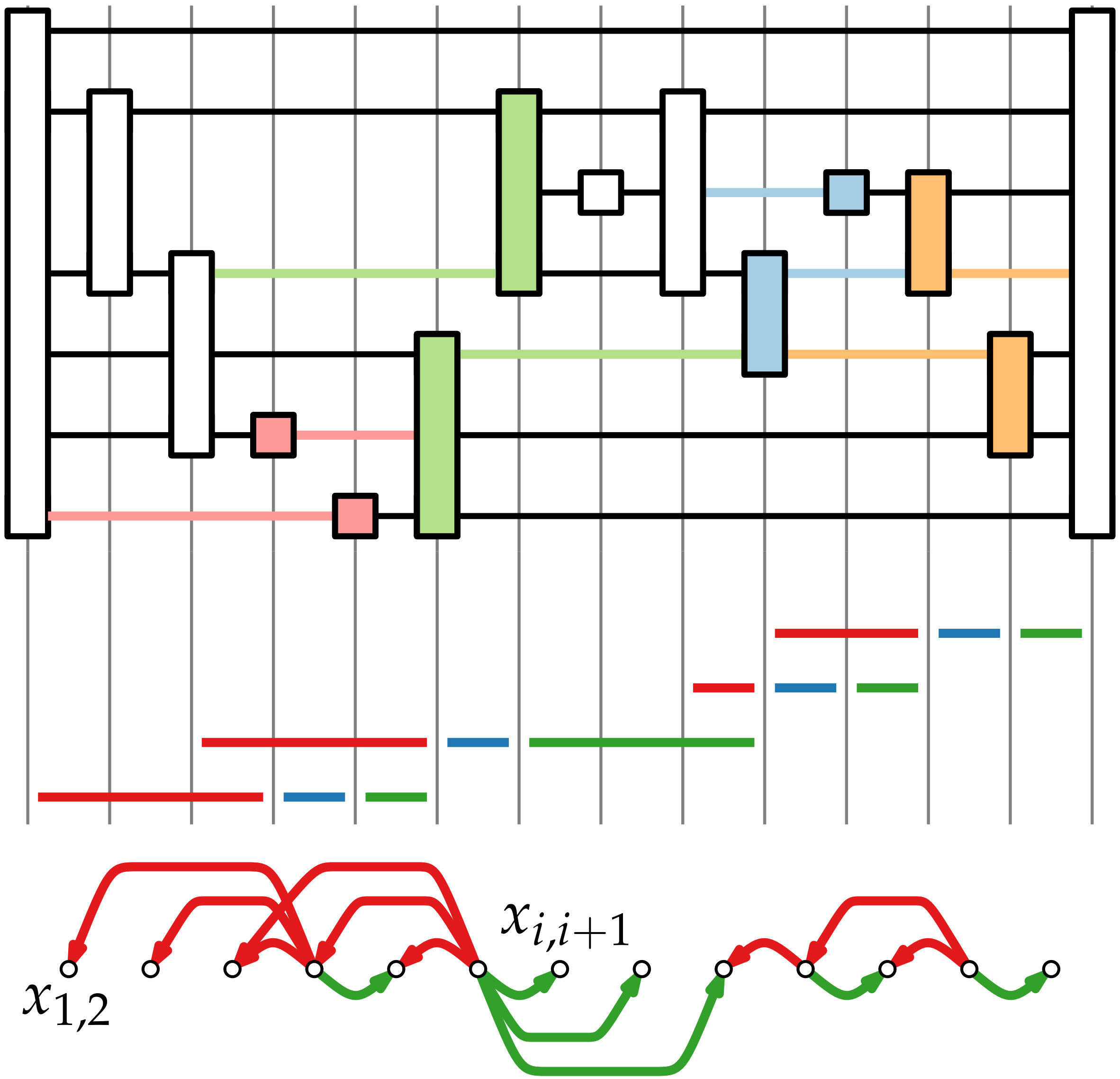
Constraints



Relation Graph

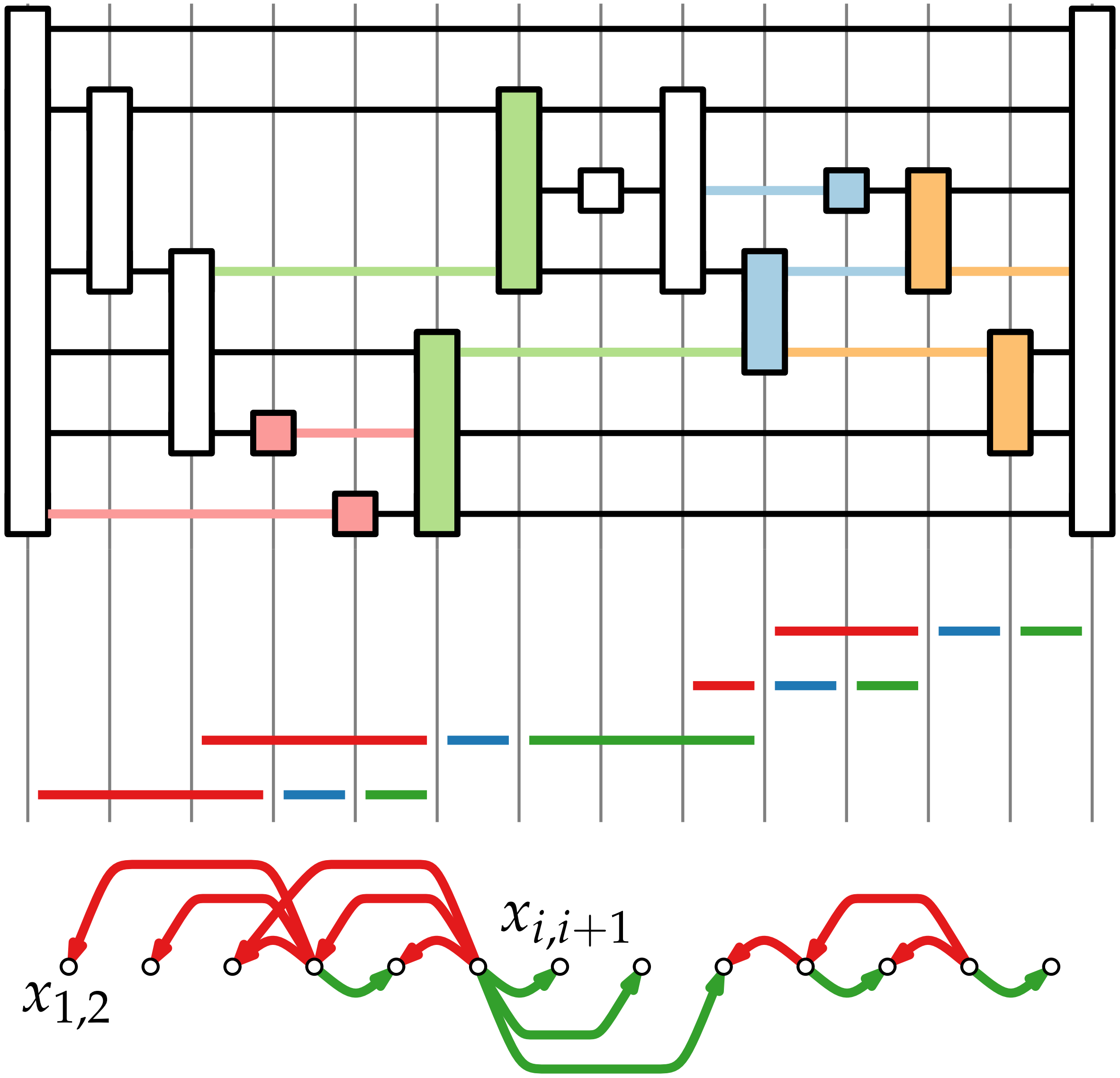
Constraints



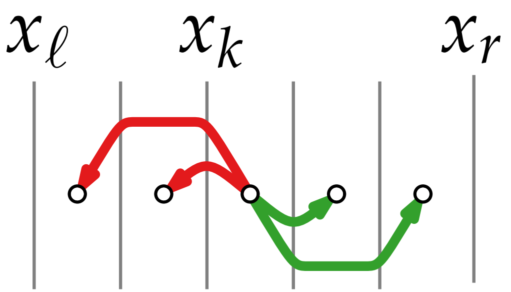


Relation Graph

Constraints

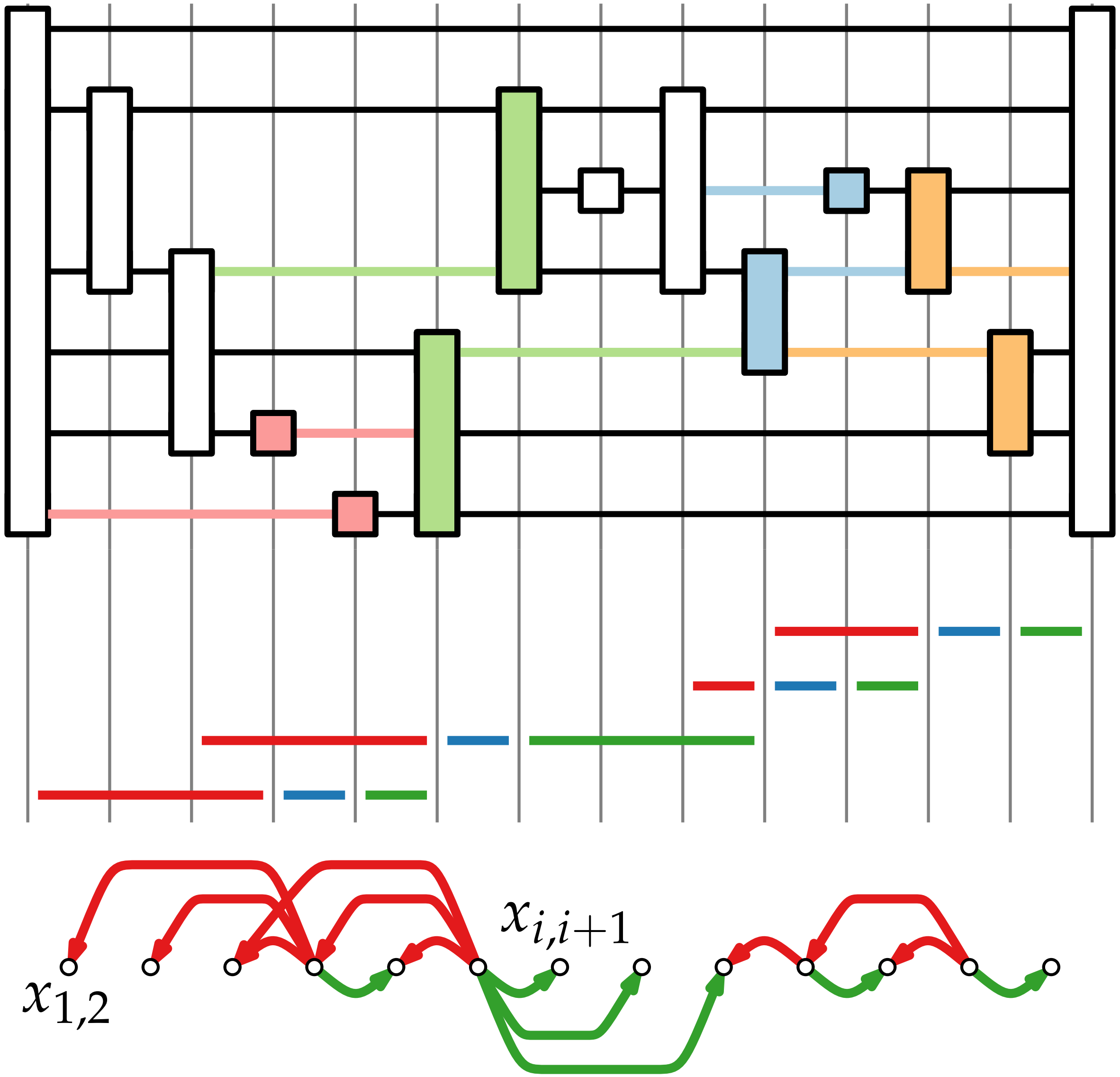


Relation Graph

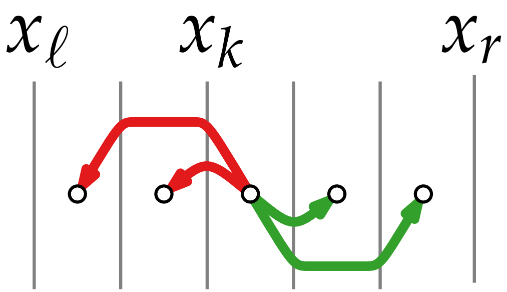


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

Constraints



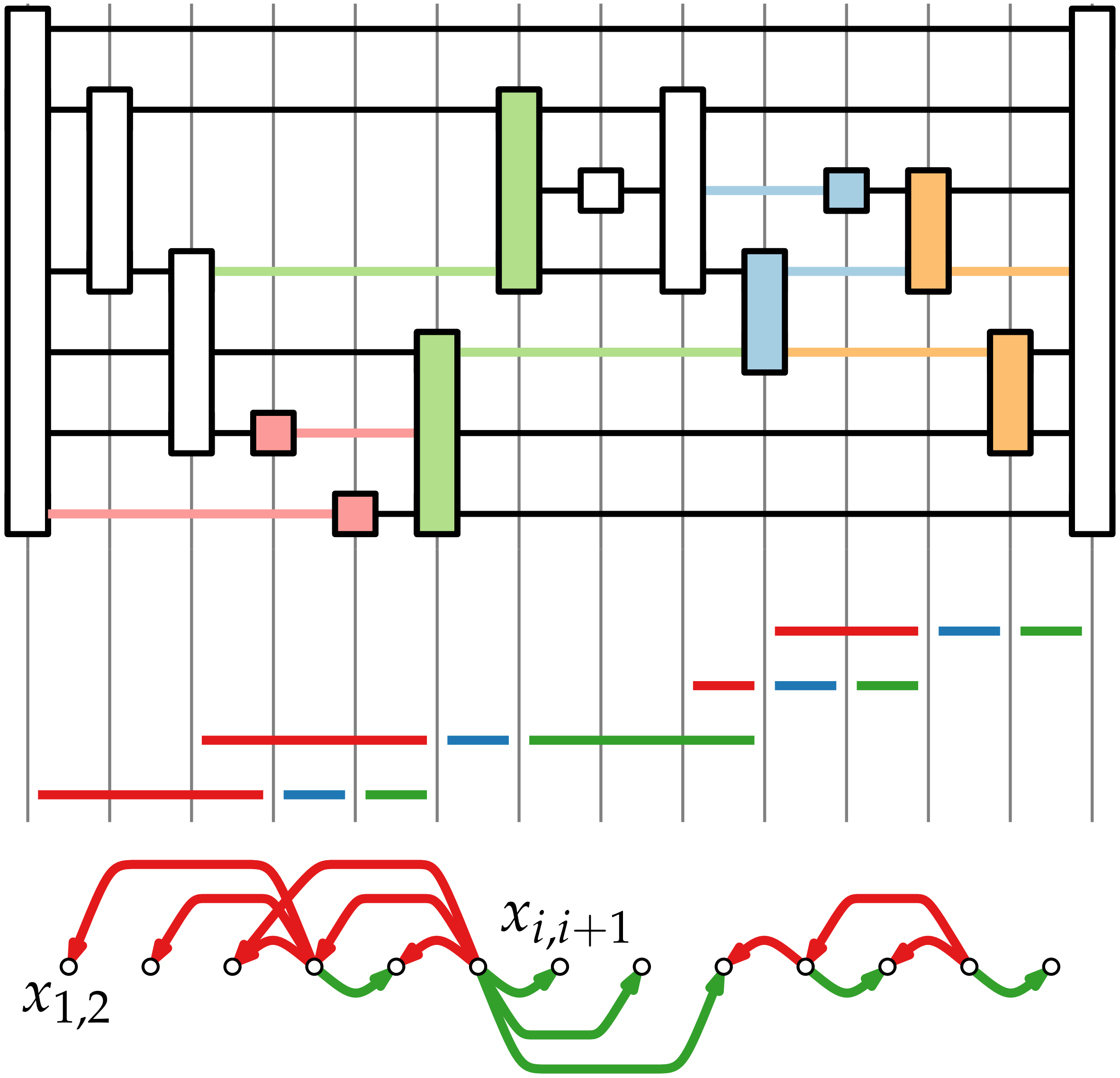
Relation Graph



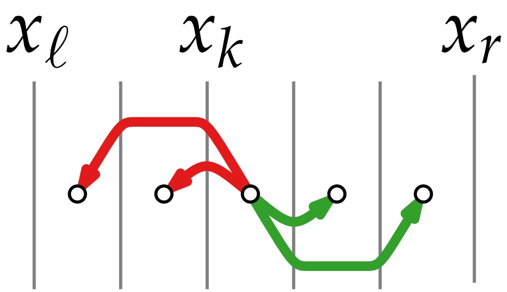
$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

Constraints



Relation Graph

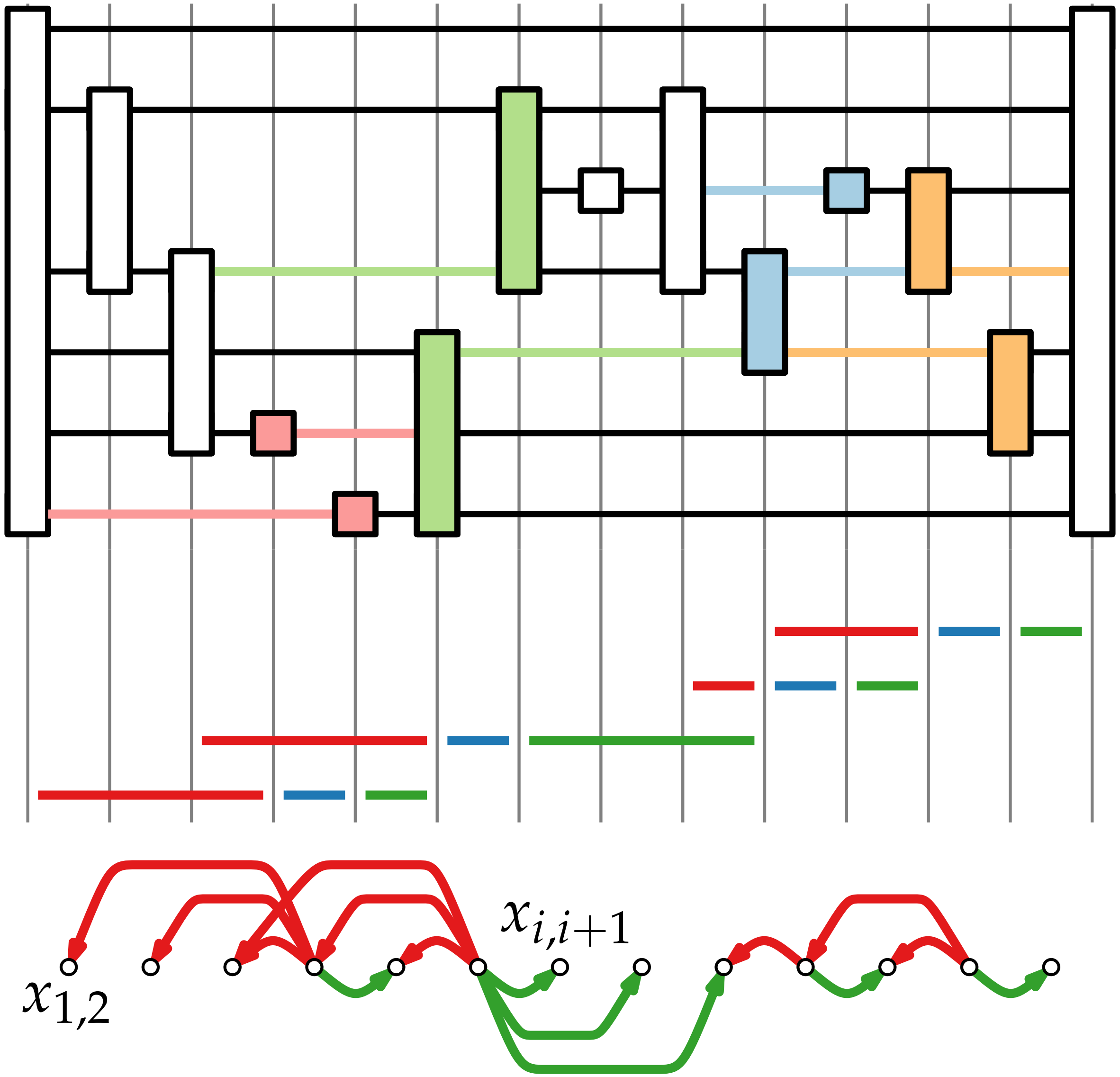


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

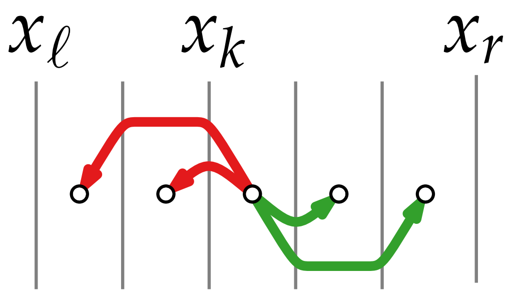
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

$\Rightarrow$ LP

Constraints



Relation Graph is acyclic

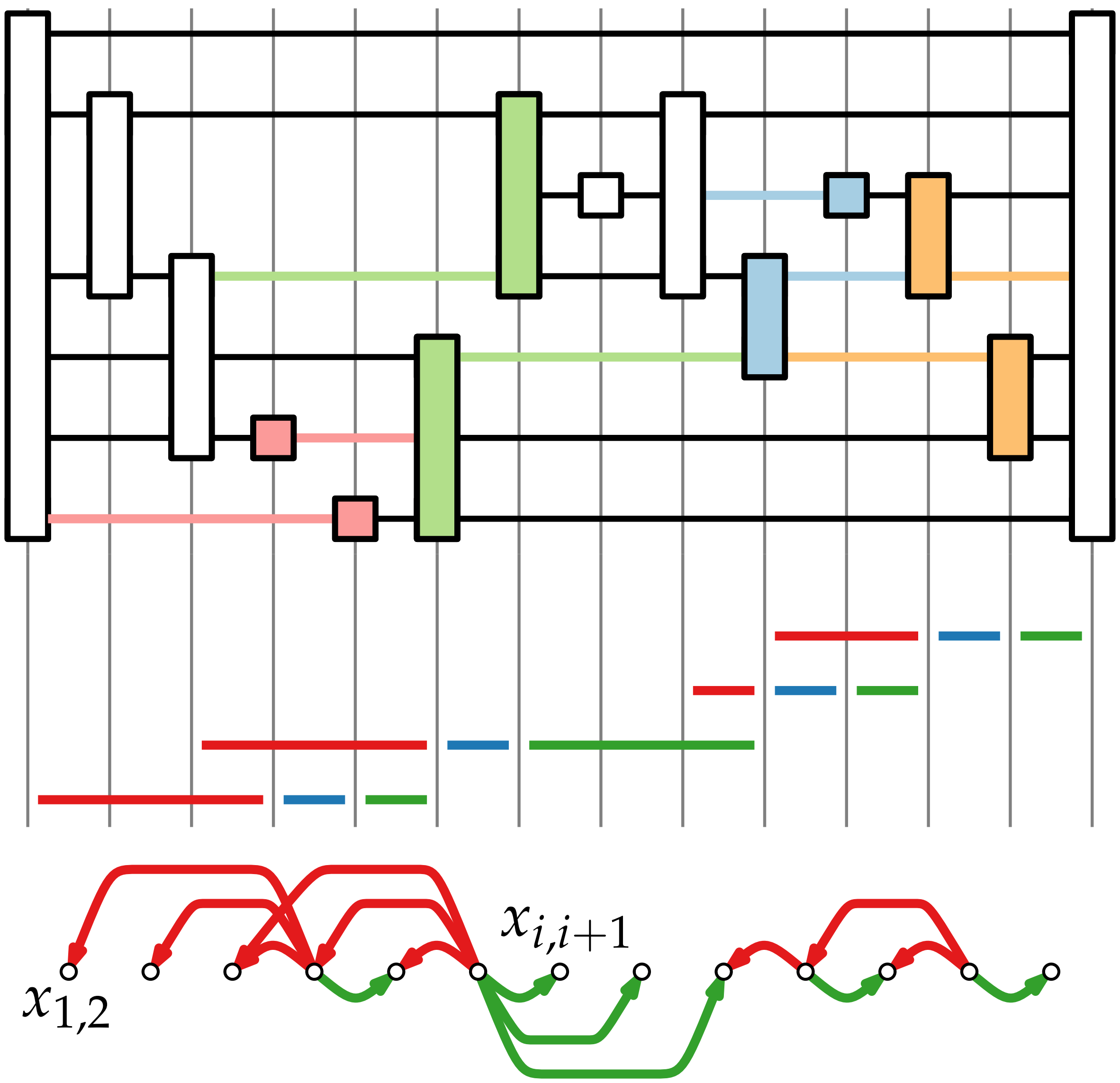


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

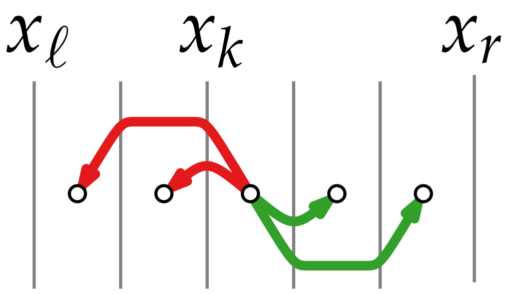
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

$\Rightarrow$ LP

Constraints



Relation Graph is acyclic

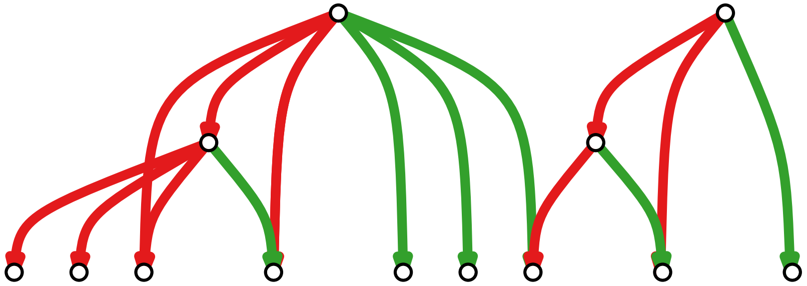


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

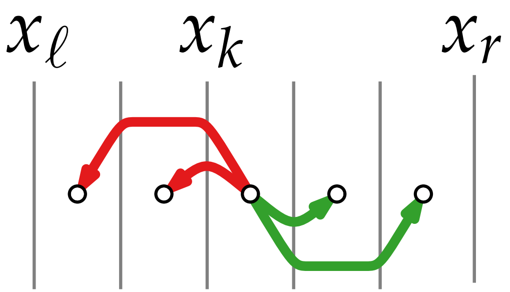
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

$\Rightarrow$ LP

Topological Sort



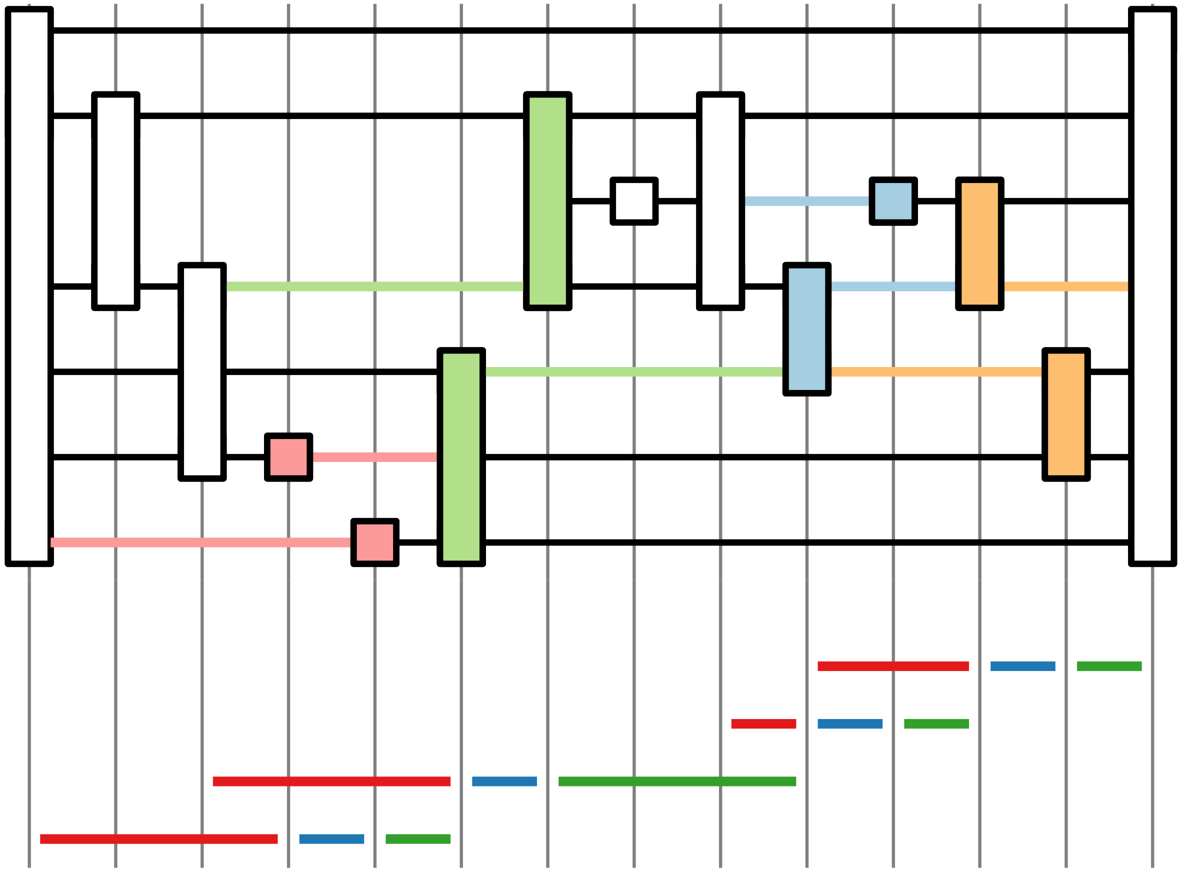
Constraints

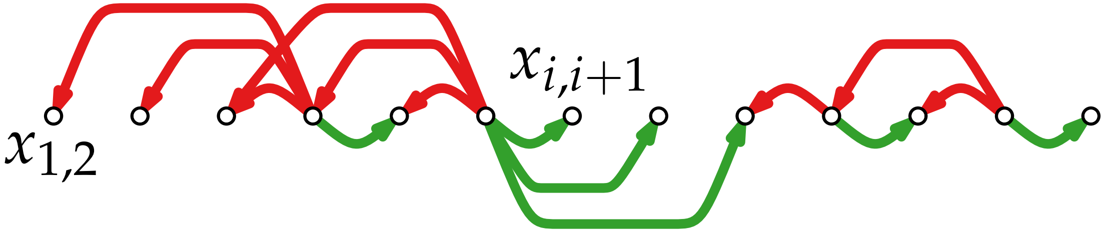


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

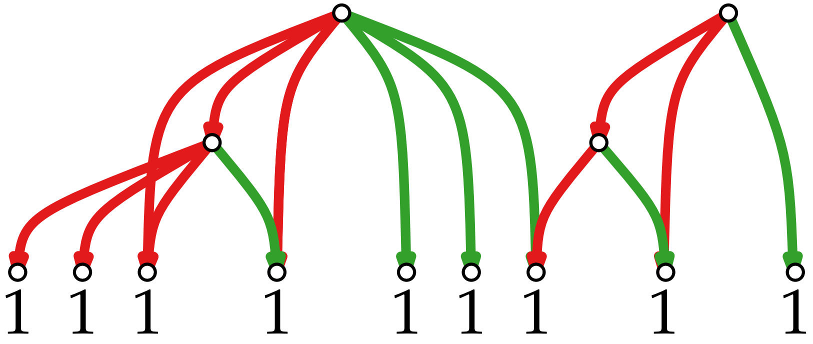
$\Rightarrow$ LP



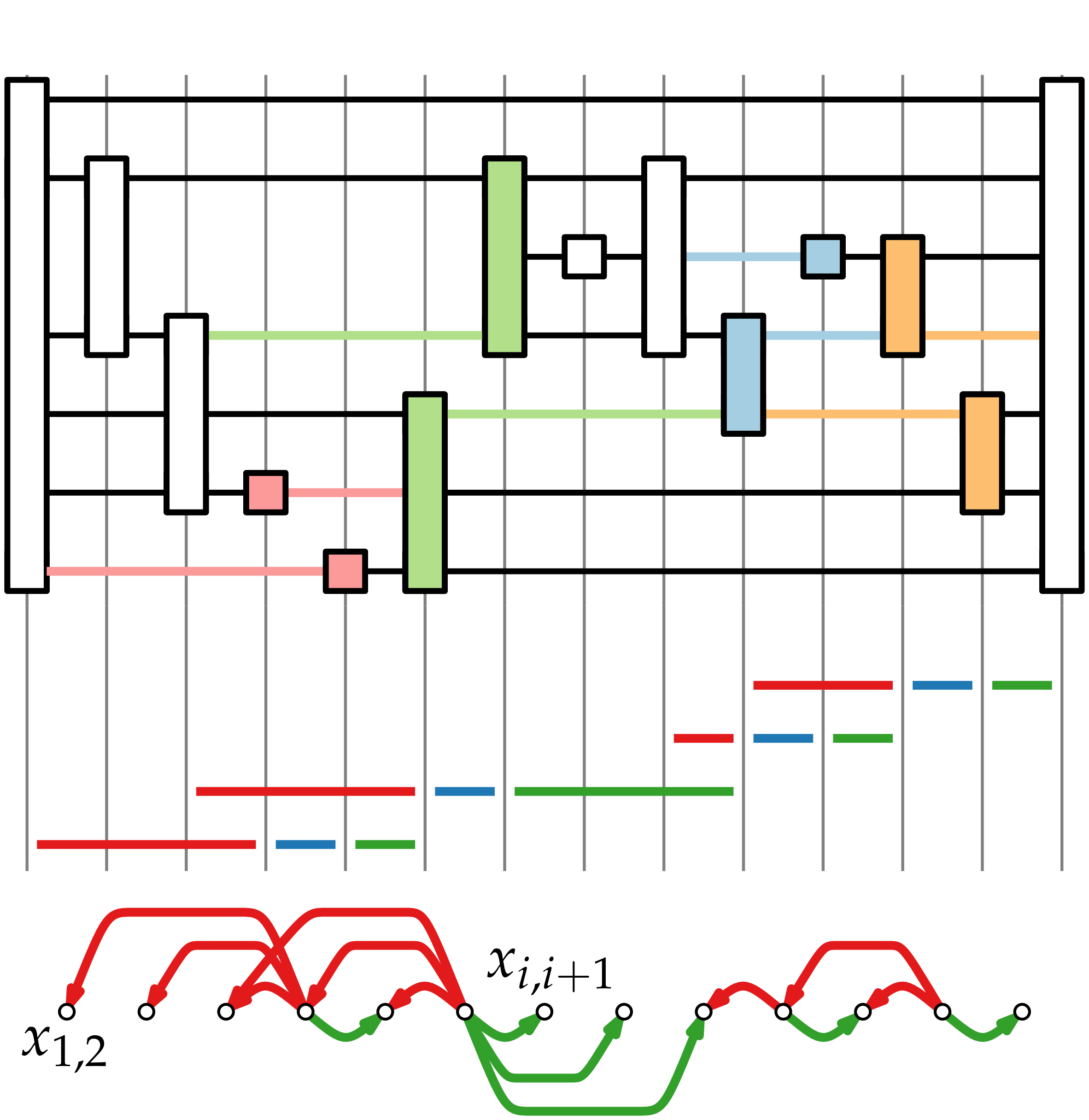


Relation Graph is acyclic

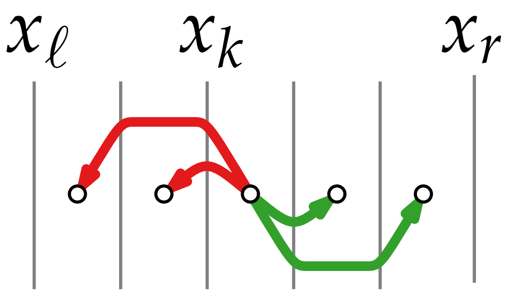
Topological Sort



Constraints



Relation Graph is acyclic

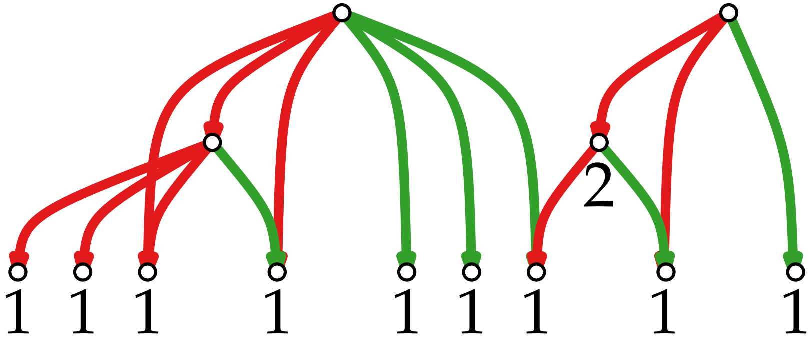


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

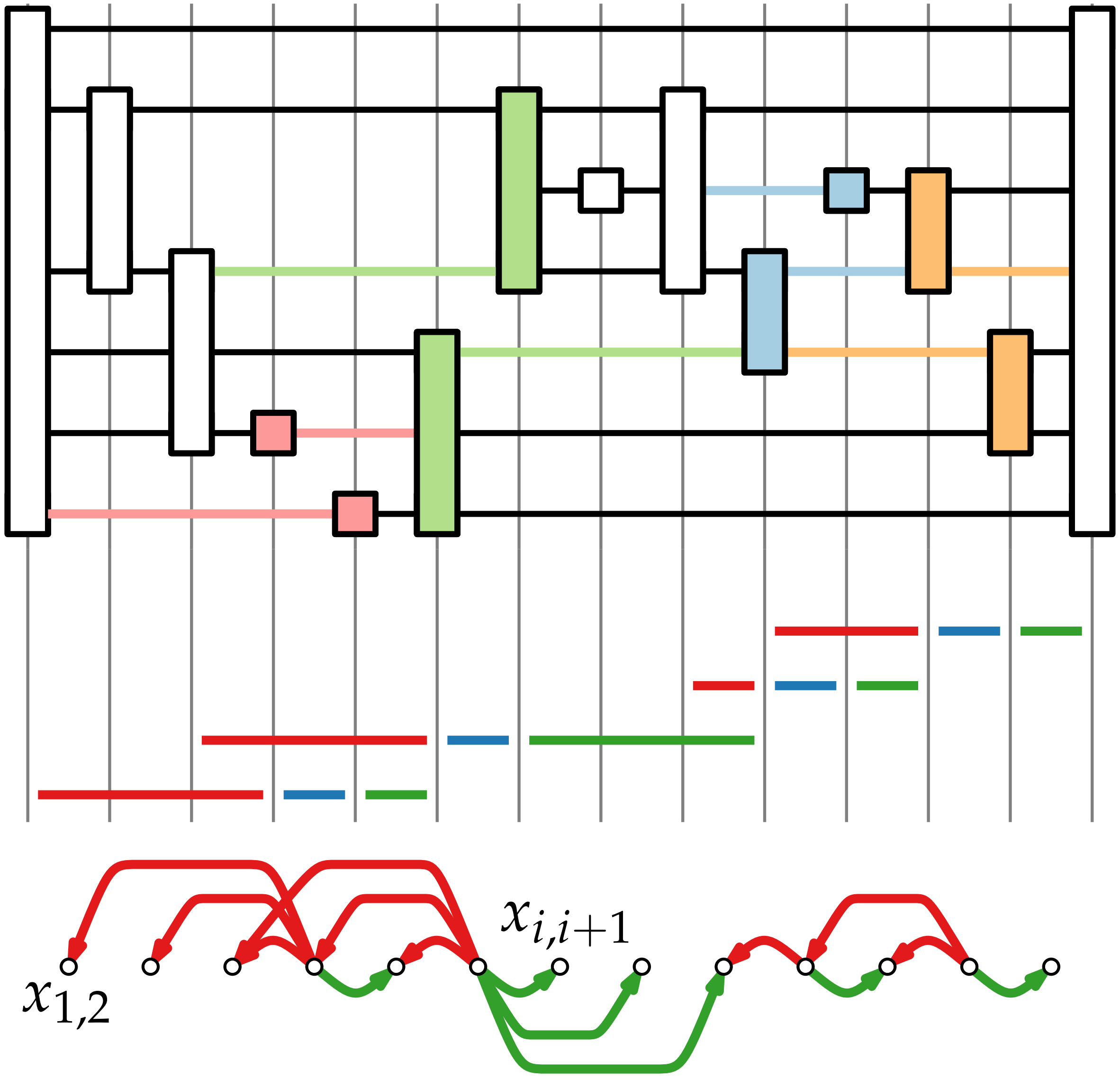
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

⇒ LP

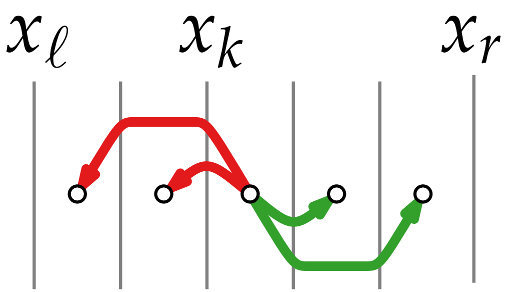
Topological Sort



Constraints



Relation Graph is acyclic

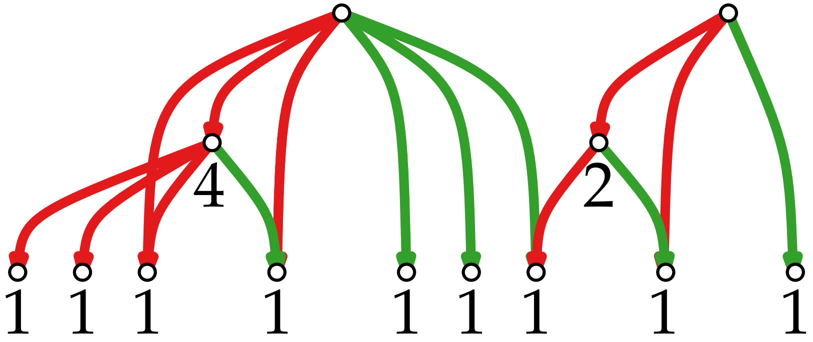


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

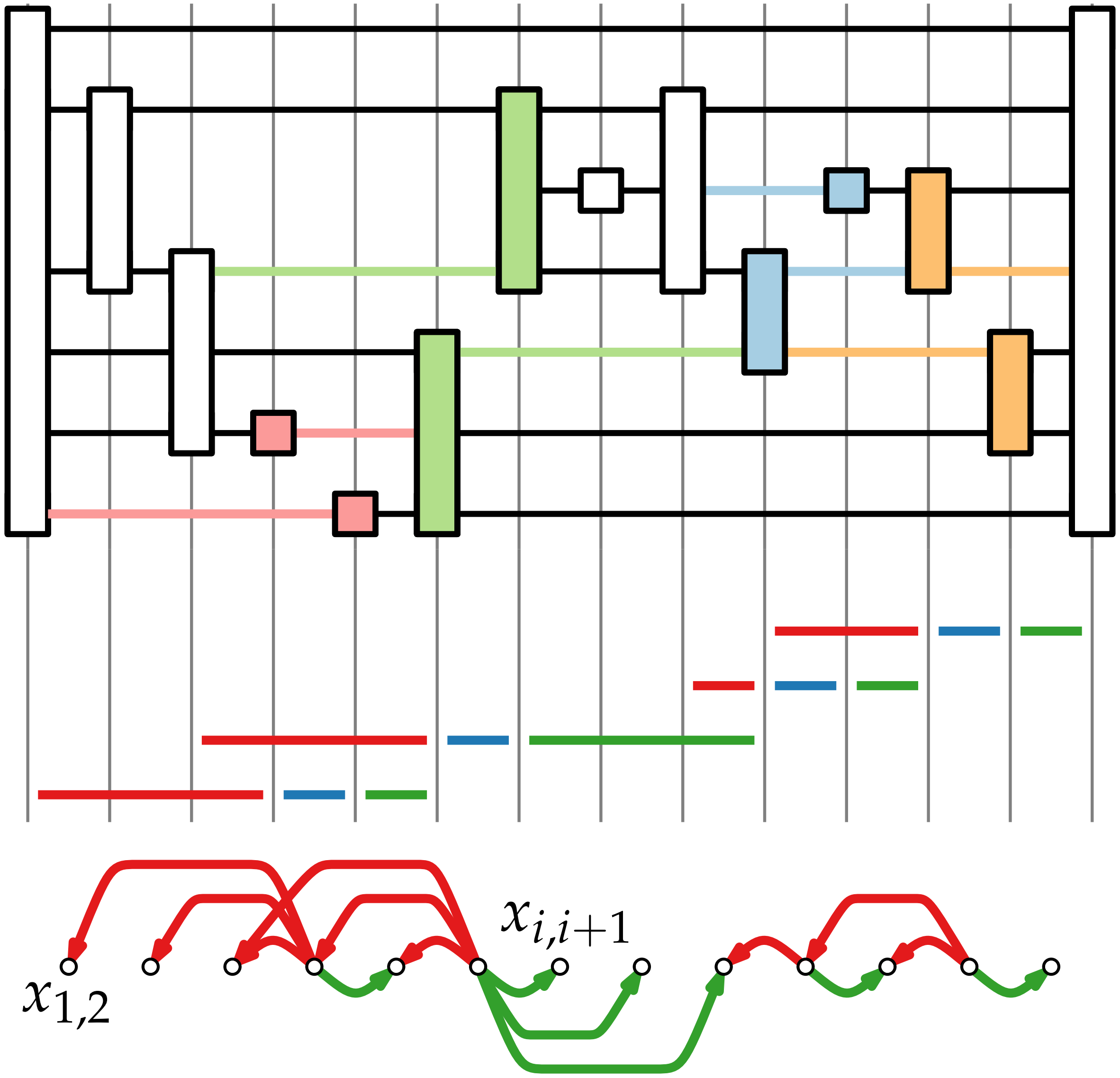
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

⇒ LP

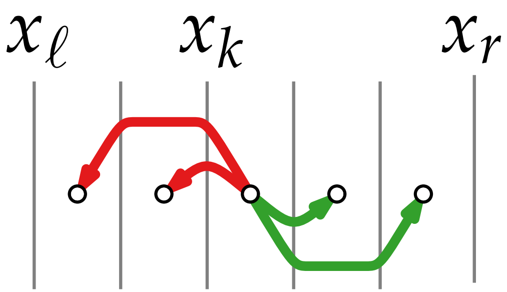
Topological Sort



Constraints



Relation Graph is acyclic

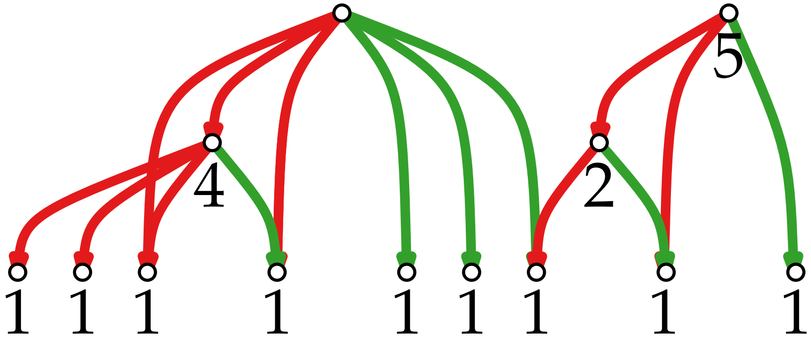


$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

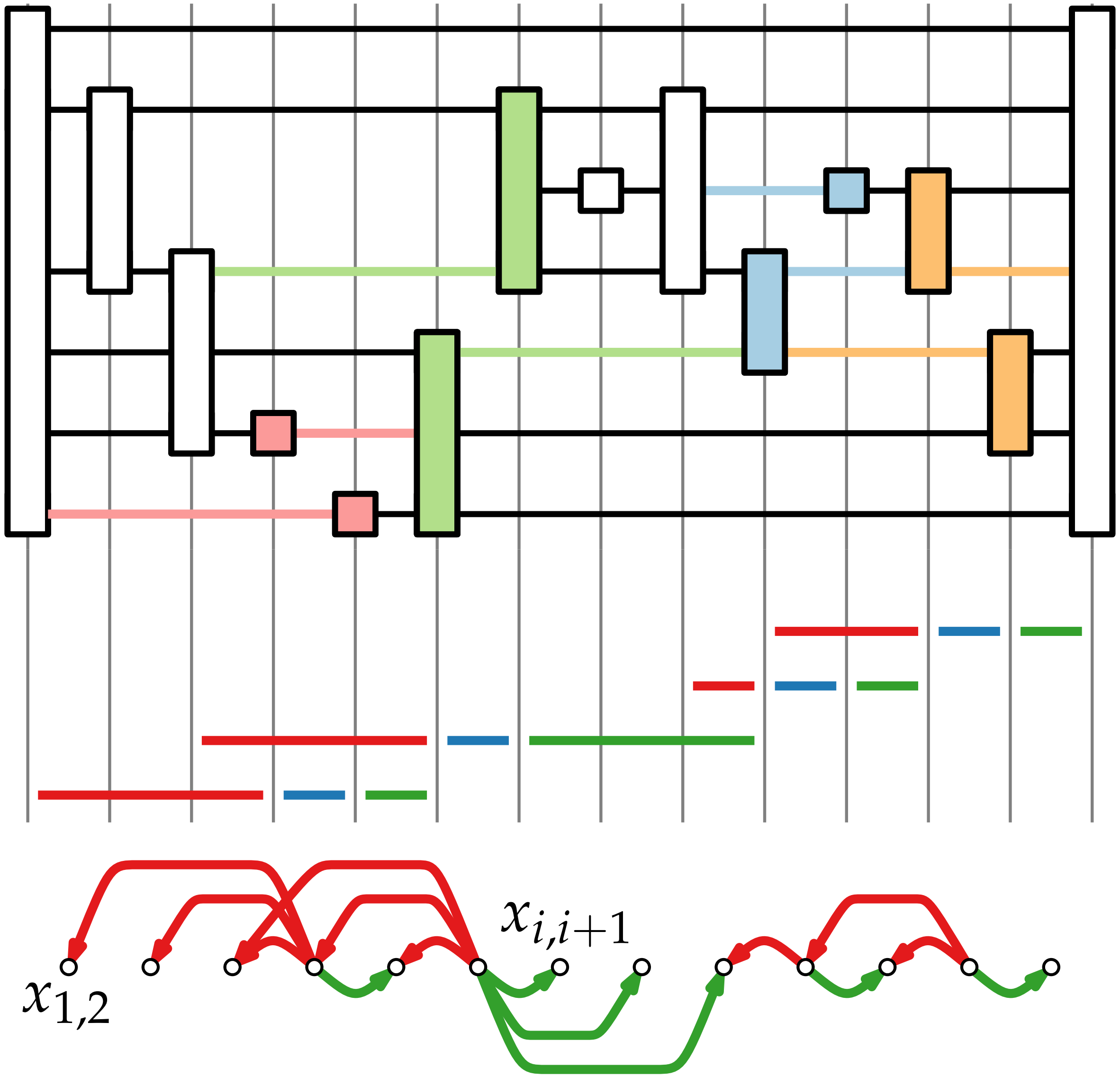
$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

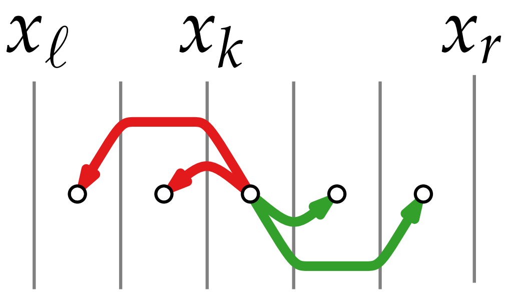
$\Rightarrow$ LP

Topological Sort



Constraints





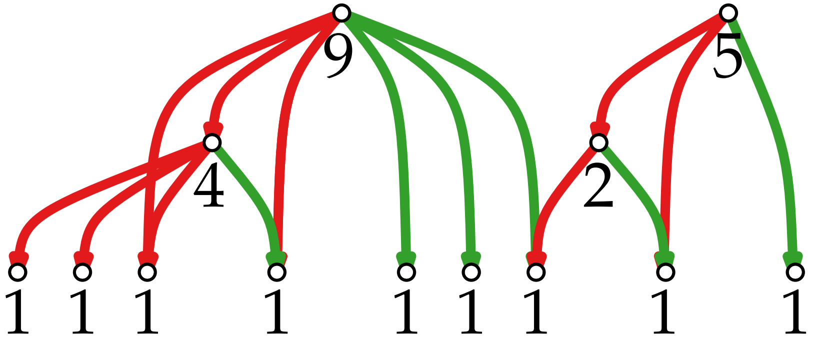
$$x_{k,k+1} > \sum_{i=\ell}^{k-1} x_{i,i+1}$$

$$x_{k,k+1} > \sum_{i=k+1}^{r-1} x_{i,i+1}$$

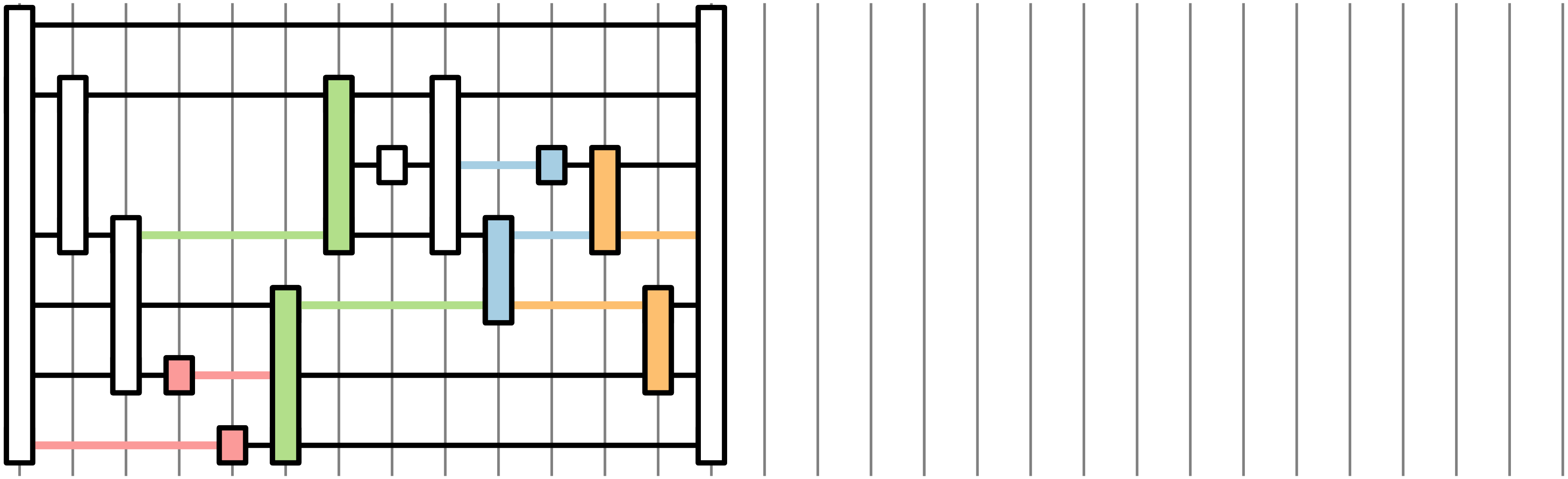
$\Rightarrow$ LP

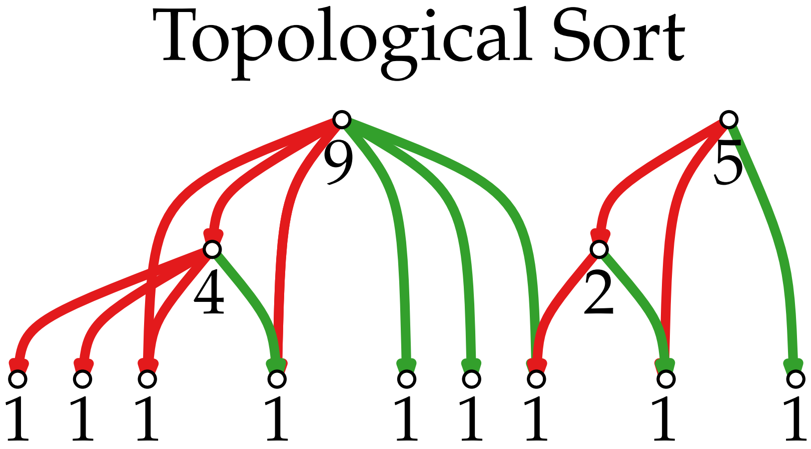
Relation Graph is acyclic

Topological Sort

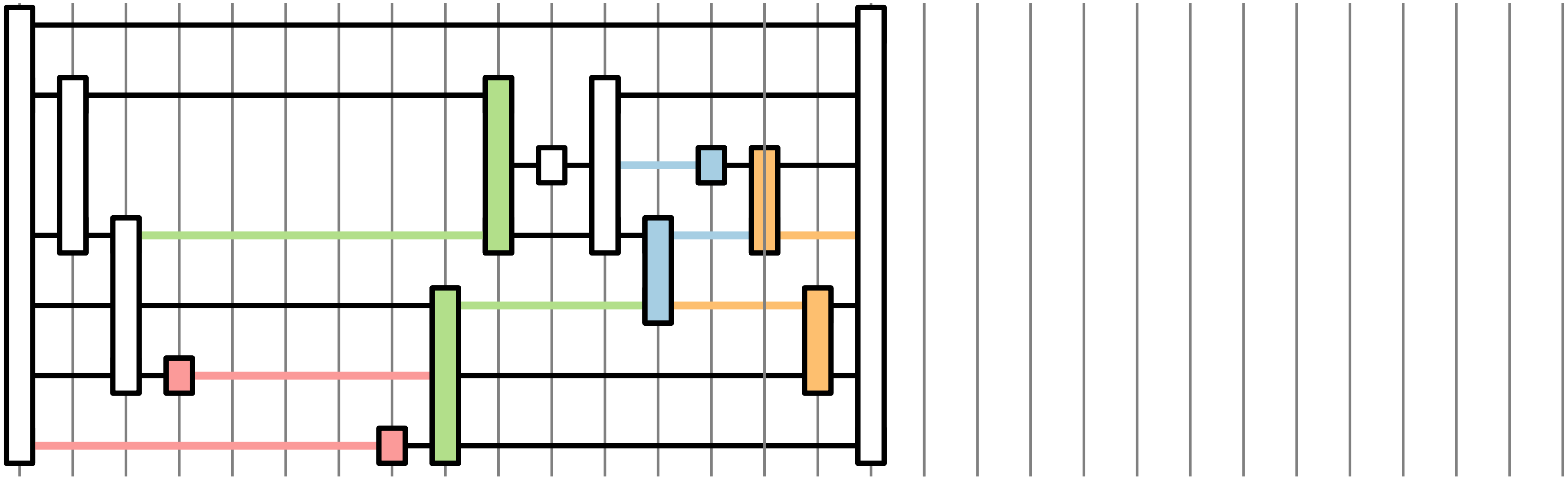


Constraints

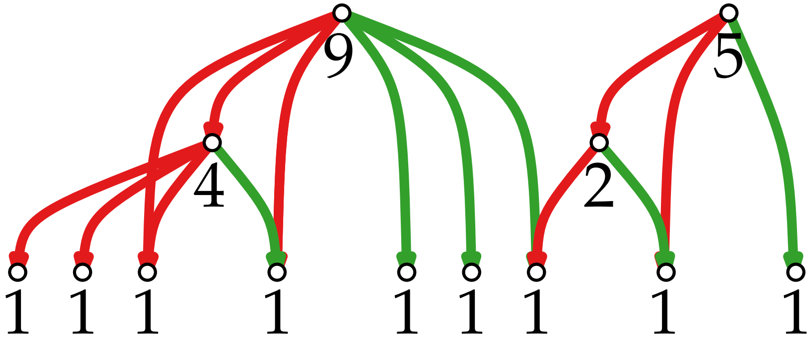




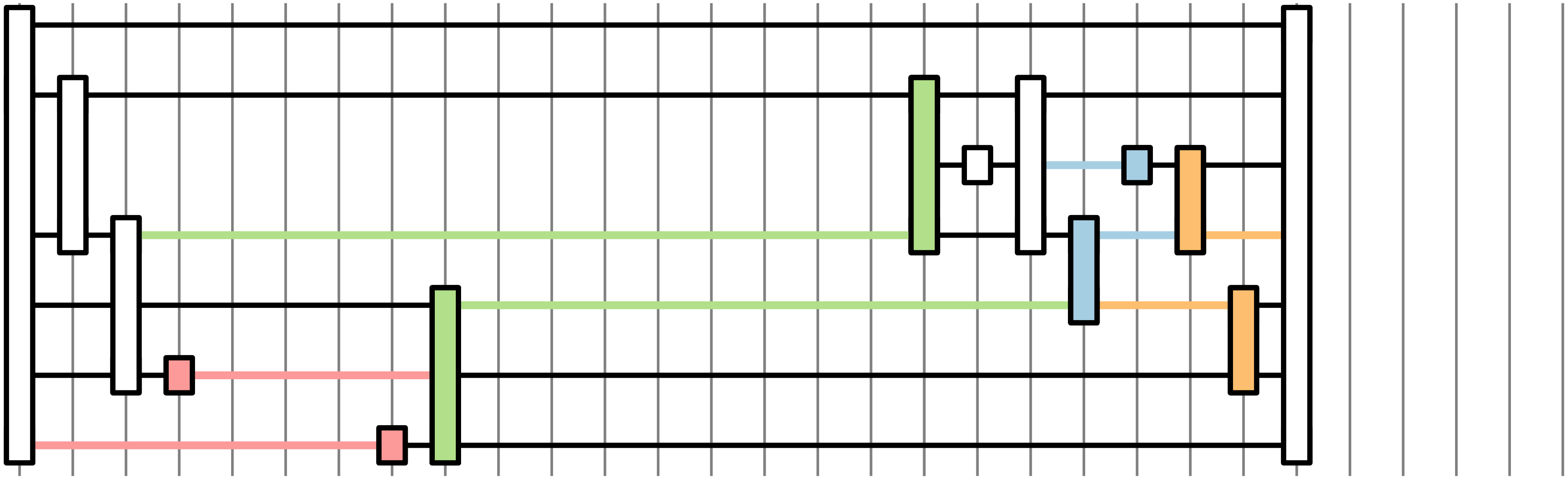
Constraints



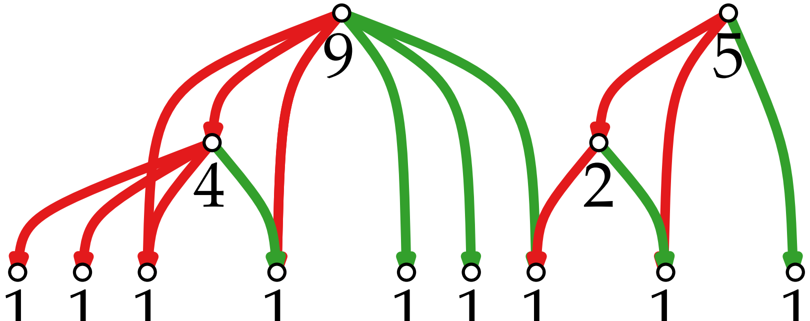
Topological Sort



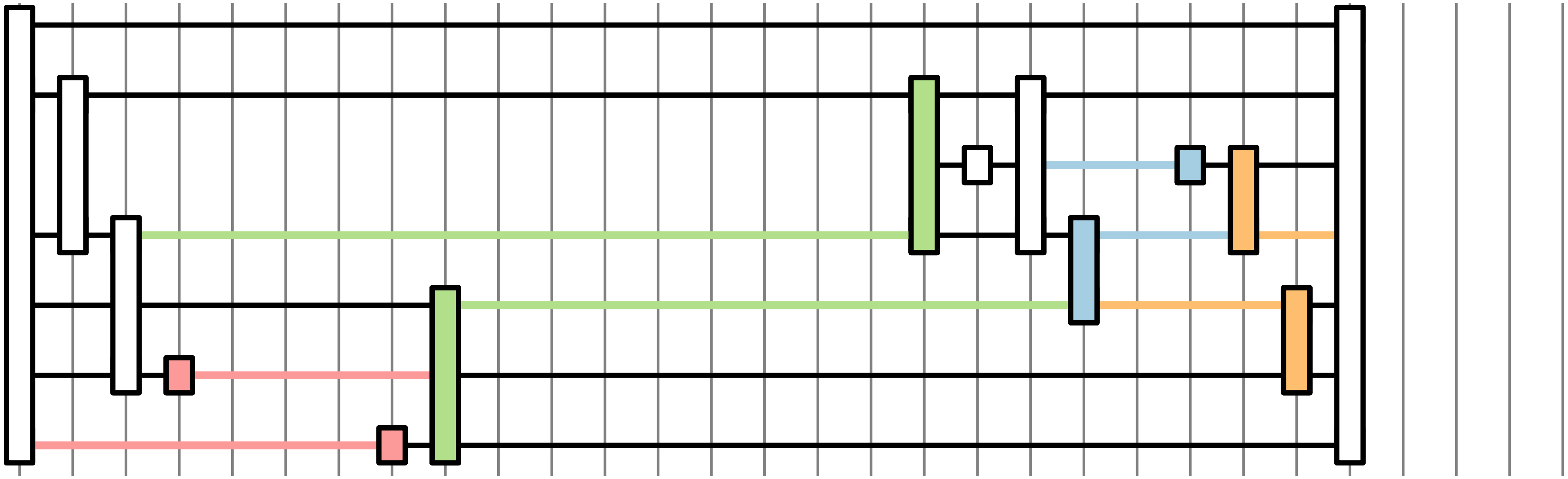
Constraints



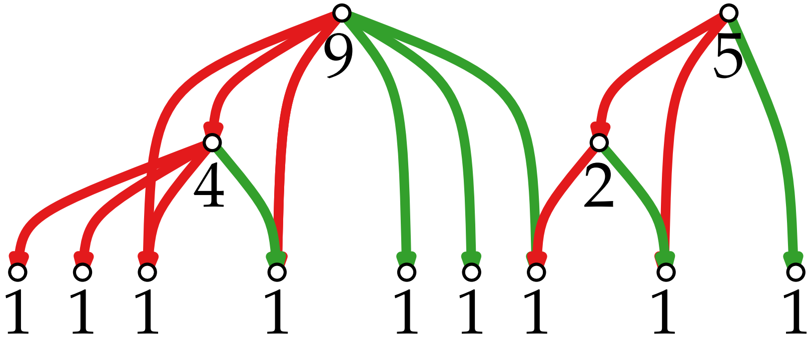
Topological Sort



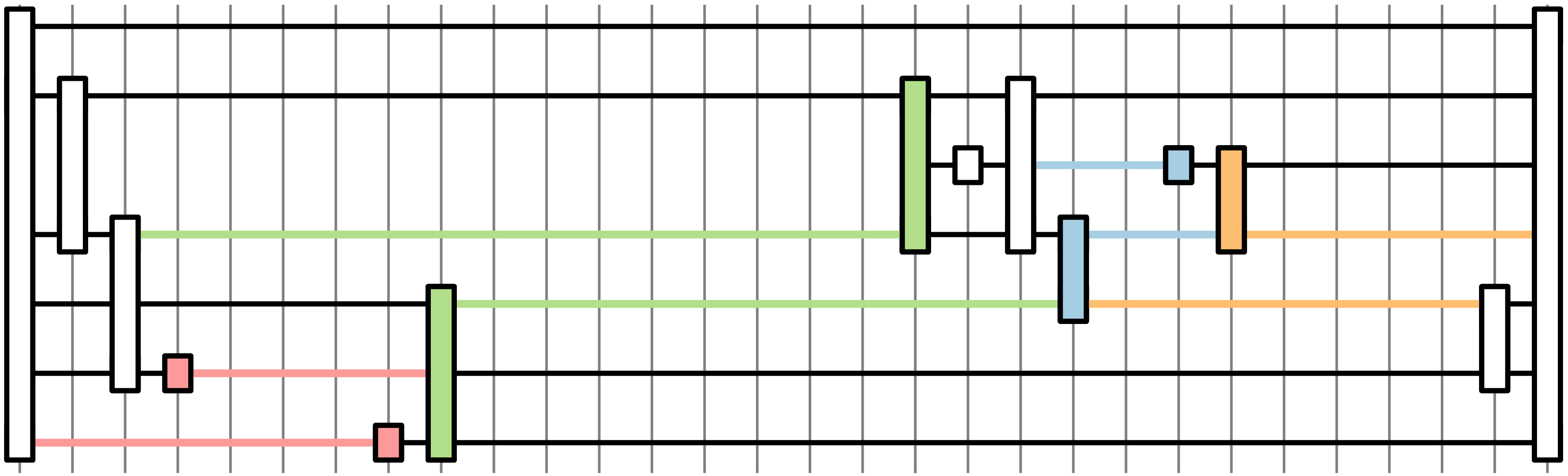
Constraints

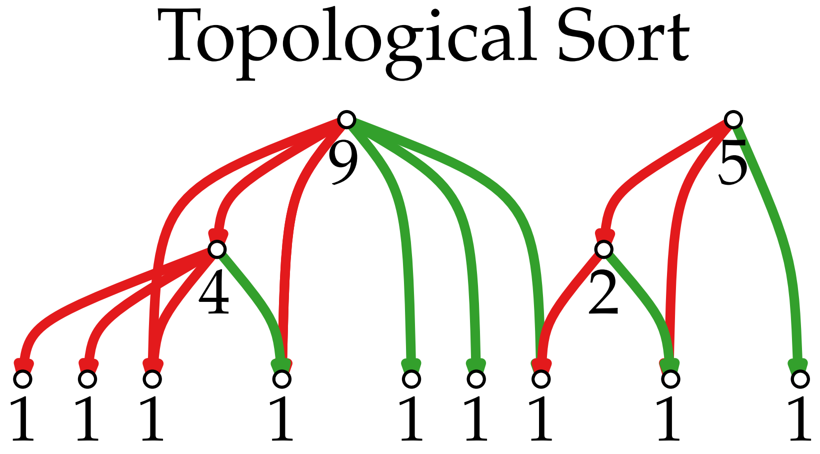


Topological Sort

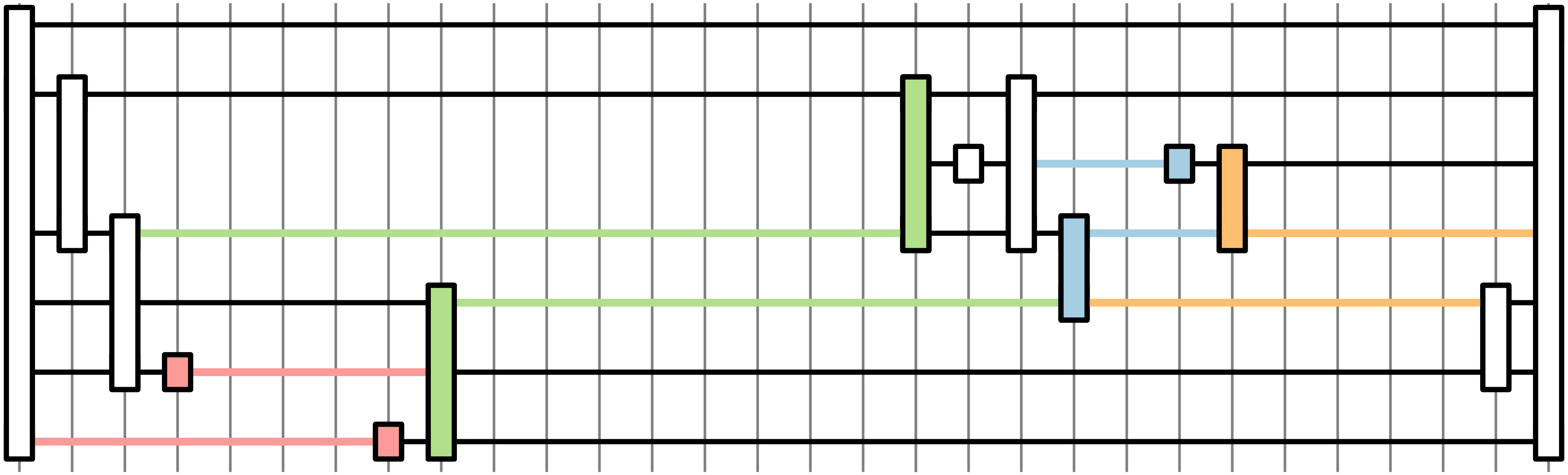


Constraints



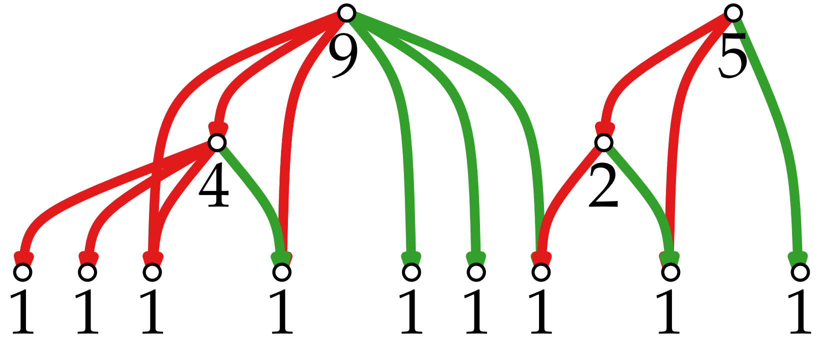


Constraints

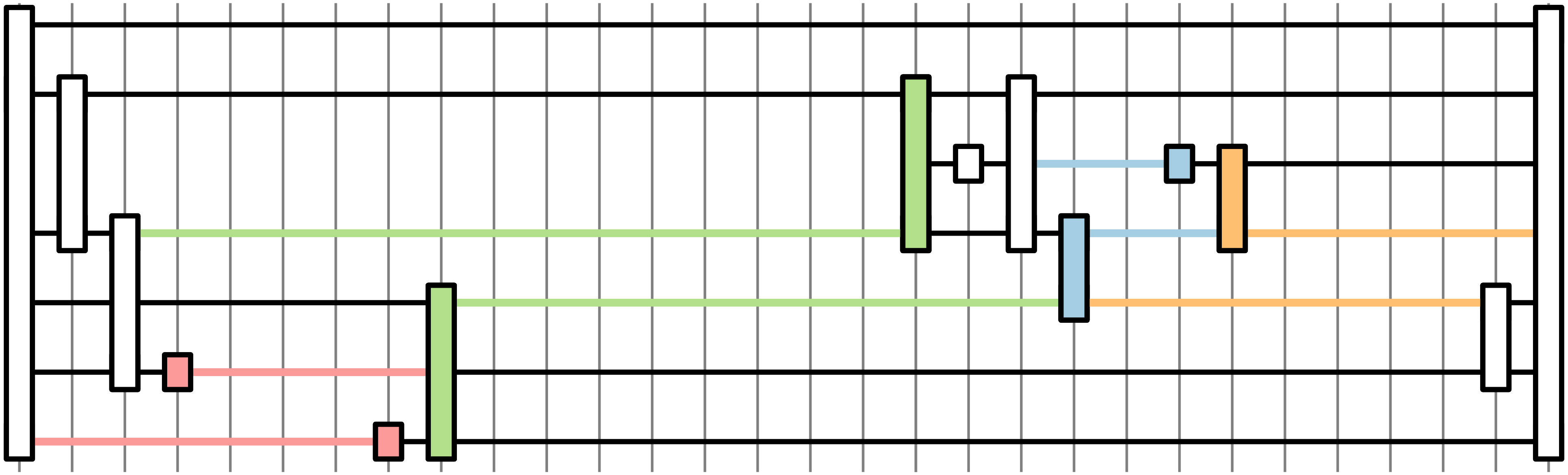


good *st*-orders ⇔ rectilinear greedy

Topological Sort



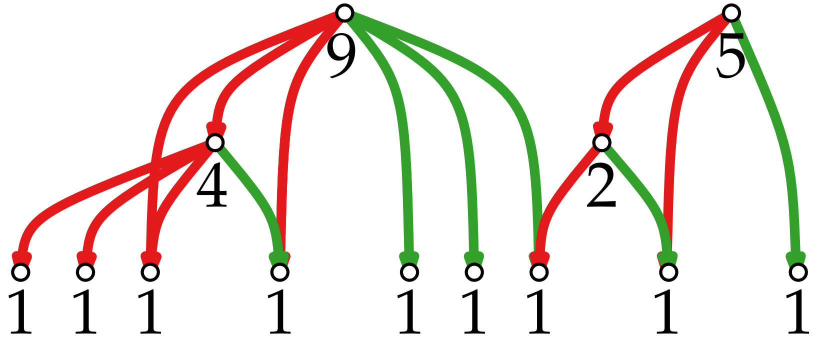
Constraints



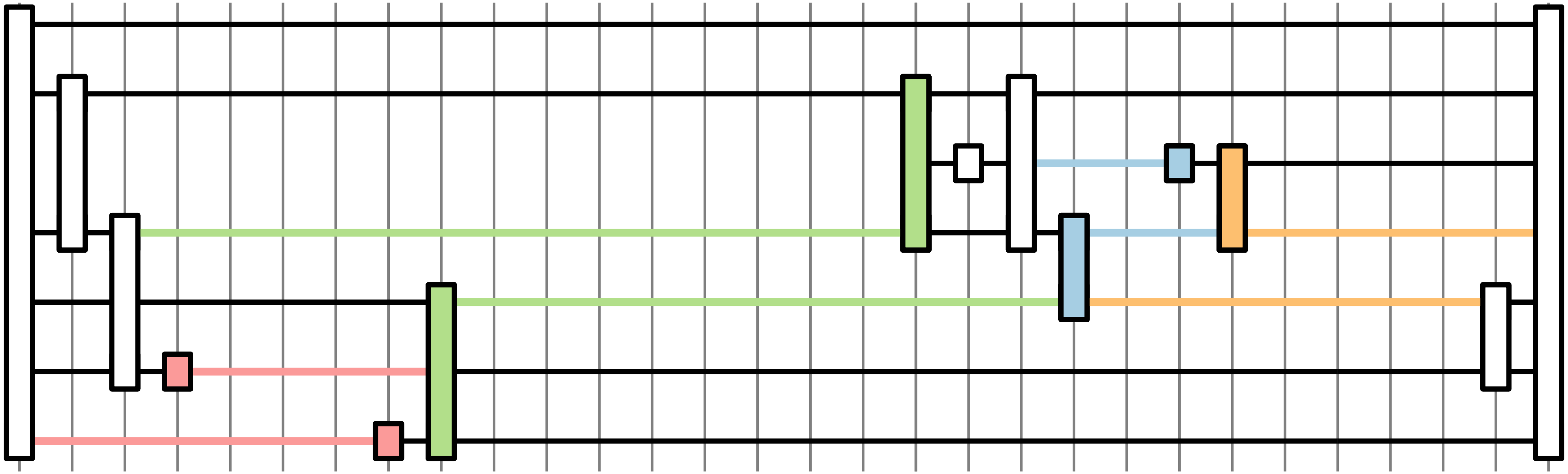
good *st*-orders ⇔ rectilinear greedy

compute drawing with min. area in $O(n^2)$ time

Topological Sort



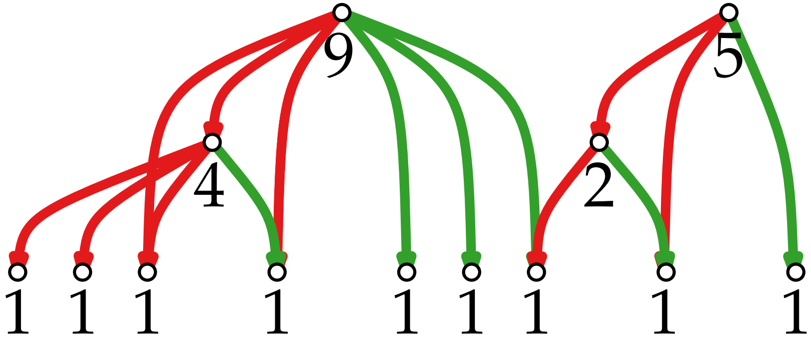
Constraints



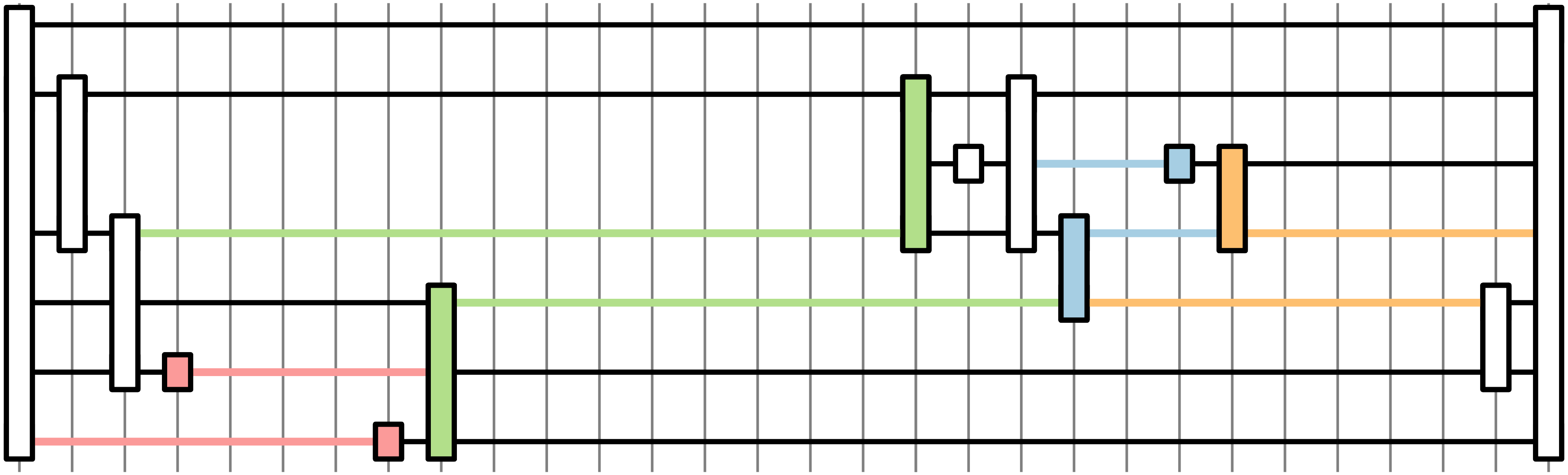
good *st*-orders ⇔ rectilinear greedy

compute drawing with min. area in $O(n^2)$ time (area can be exp.)

Topological Sort



Constraints

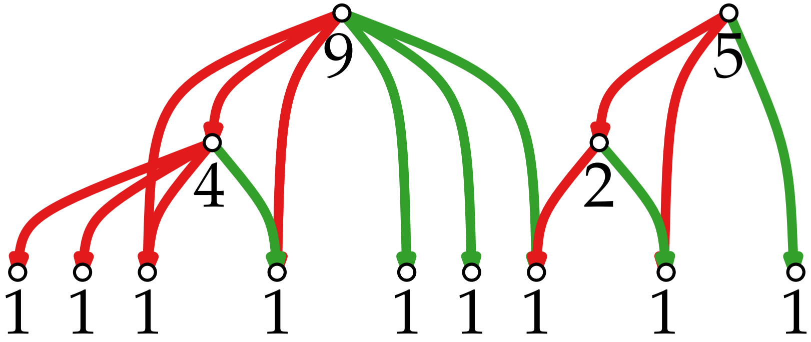


good st-orders ⇔ rectilinear greedy

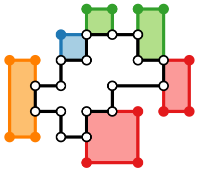
compute drawing with min. area in $O(n^2)$ time (area can be exp.)

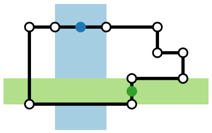
find good st-order?

Topological Sort



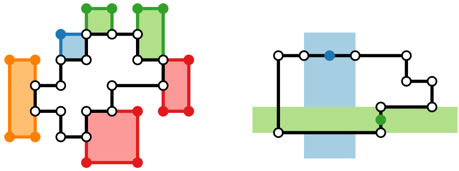
Conclusion



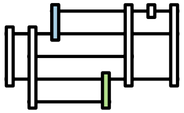


Test and gen. all universal greedy rectilinear graphs in $O(n)$ time

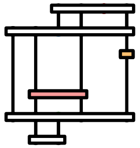
Conclusion



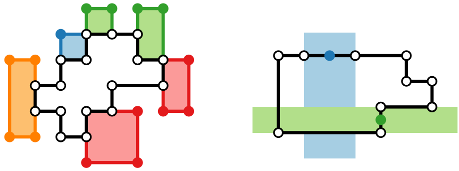
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



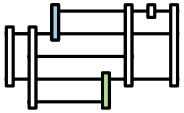
hor. and vert. *st*-digraphs



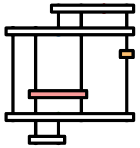
Conclusion



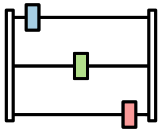
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



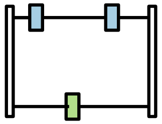
hor. and vert. *st*-digraphs



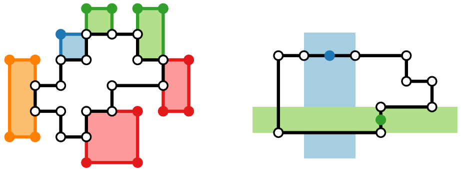
good st-order: for every interval $i, \dots, j$:



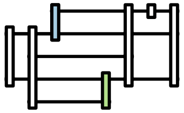
– no 3 conn. comp. – if 2 conn. comp., then disjoint



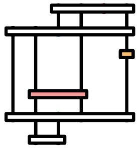
Conclusion



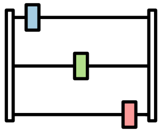
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



hor. and vert. *st*-digraphs

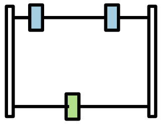


good st-order: for every interval $i, \ldots, j$:

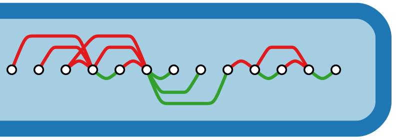


– no 3 conn. comp.

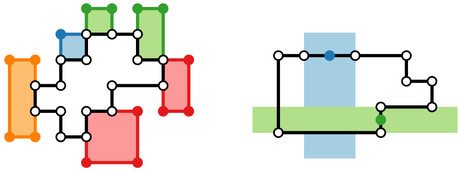
– if 2 conn. comp., then disjoint



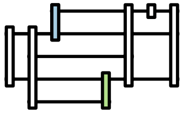
good *st*-orders ⇔ rectilinear greedy



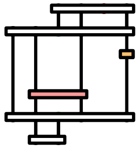
Conclusion



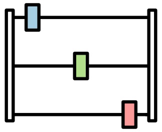
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



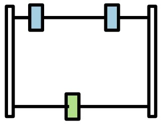
hor. and vert. *st*-digraphs



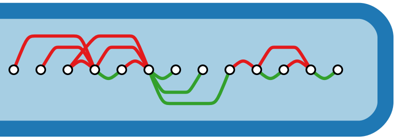
good st-order: for every interval $i, \ldots, j$:



– no 3 conn. comp. – if 2 conn. comp., then disjoint

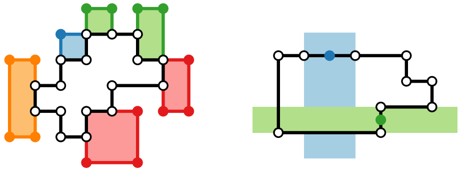


good *st*-orders ⇔ rectilinear greedy

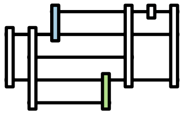


compute drawing with min. area in $O(n^2)$ time (area can be exp.)

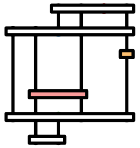
Conclusion



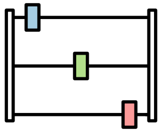
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



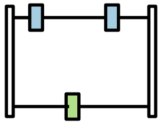
hor. and vert. *st*-digraphs



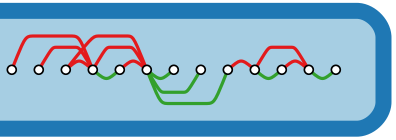
good st-order: for every interval $i, \ldots, j$:



– no 3 conn. comp. – if 2 conn. comp., then disjoint



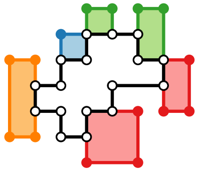
good *st*-orders ⇔ rectilinear greedy

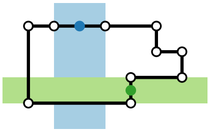


compute drawing with min. area in $O(n^2)$ time (area can be exp.)

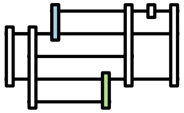
find good *st*-order?

Conclusion

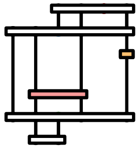




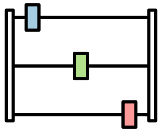
Test and gen. all universal greedy rectilinear graphs in $O(n)$ time



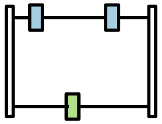
hor. and vert. *st*-digraphs



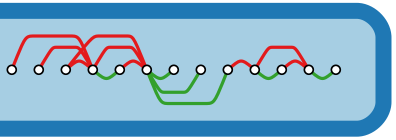
good st-order: for every interval $i, \ldots, j$:



– no 3 conn. comp. – if 2 conn. comp., then disjoint



good *st*-orders ⇔ rectilinear greedy



compute drawing with min. area in $O(n^2)$ time (area can be exp.)

find good *st*-order?

st-digraph series-parallel ⇒ find good *st*-order in $O(n)$ time