APPLYING BLOCKCHAIN TECHNOLOGY FOR HYPERCONNECTED LOGISTICS

Wout Hofman, Jacco Spek, Christopher Brewster

Fourth International Physical Internet Conference, Graz, 2017

TNO innovation for life

OBJECTIVE

- Can Blockchain Technology contribute to realizing the Physical Internet
  - Is the TRL sufficient?
  - How can it be applied?

- Starting from an example – container trans-shipment in a port

PHYSICAL INTERNET – LAYERING AND DATA

TNO innovation for life

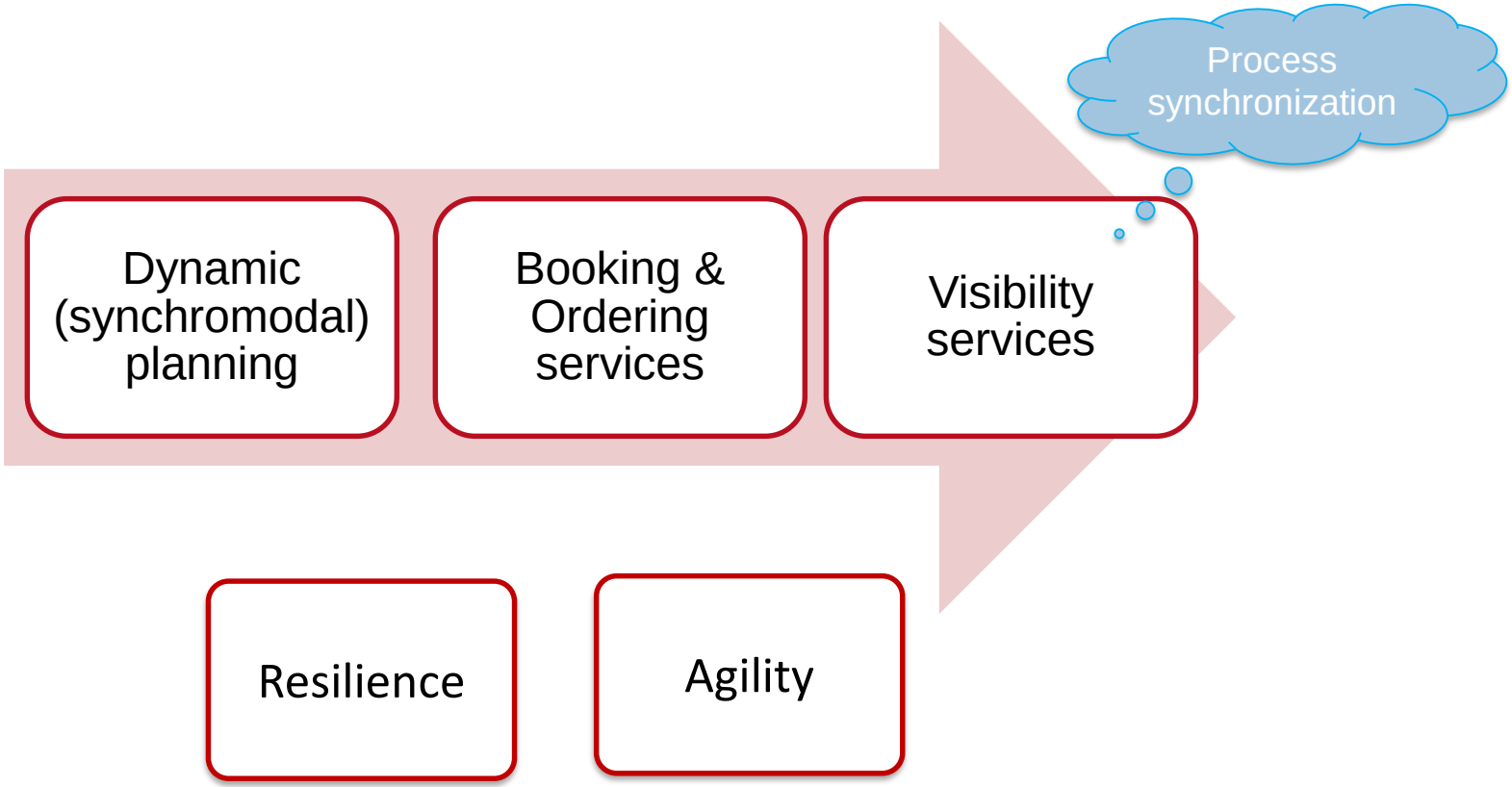
Internet layers

- Application layer
- Transport layer
- Network layer
- Link layer
- Physical layer

Physical Internet layers

Customer layer	Supply, demand, and replenishment of products with a Quality of Service (QoS) requirement
л -box	
Shipment layer	End-to-end shipment of cargo over the logistics network
л -box	
Logistic network layer	Dynamical routing of packages over a logistics network of hubs and corridors between those hubs
Autonomous operation Corridor/transport layer	Multi modal transport along corridors between hubs utilizing an underlying QoS
Physical Infrastructure layer	Phsyical infrastructure (road, rail, inland waterways, sea, air, pipeline) and hubs with a particular QoS based on maintenance and incidents, closing hours, etc.

DATA REQUIREMENTS



BLOCKCHAIN TECHNOLOGY

- Open blockchain
  - Ethereum with Smart Contracts
  - Independent foundation
- Permissioned blockchain (open source)
  - Hyperledger Fabric with smart contracts deployed as chain code
  - Others like Blockchain Explorer as visualisation
  - Premier members like IBM, Accenture, Airbus, Intel, JP Morgan, Fuijitsu, and many others (general members)
- Multichain – developed by TUDelft
- BigChainDB – underlying scalable data storage solution for Ethereum and Hyperledger; foundation with members serving as peers for data distribution

BLOCKCHAIN TECHNOLOGY – GENERAL OBSERVATIONS

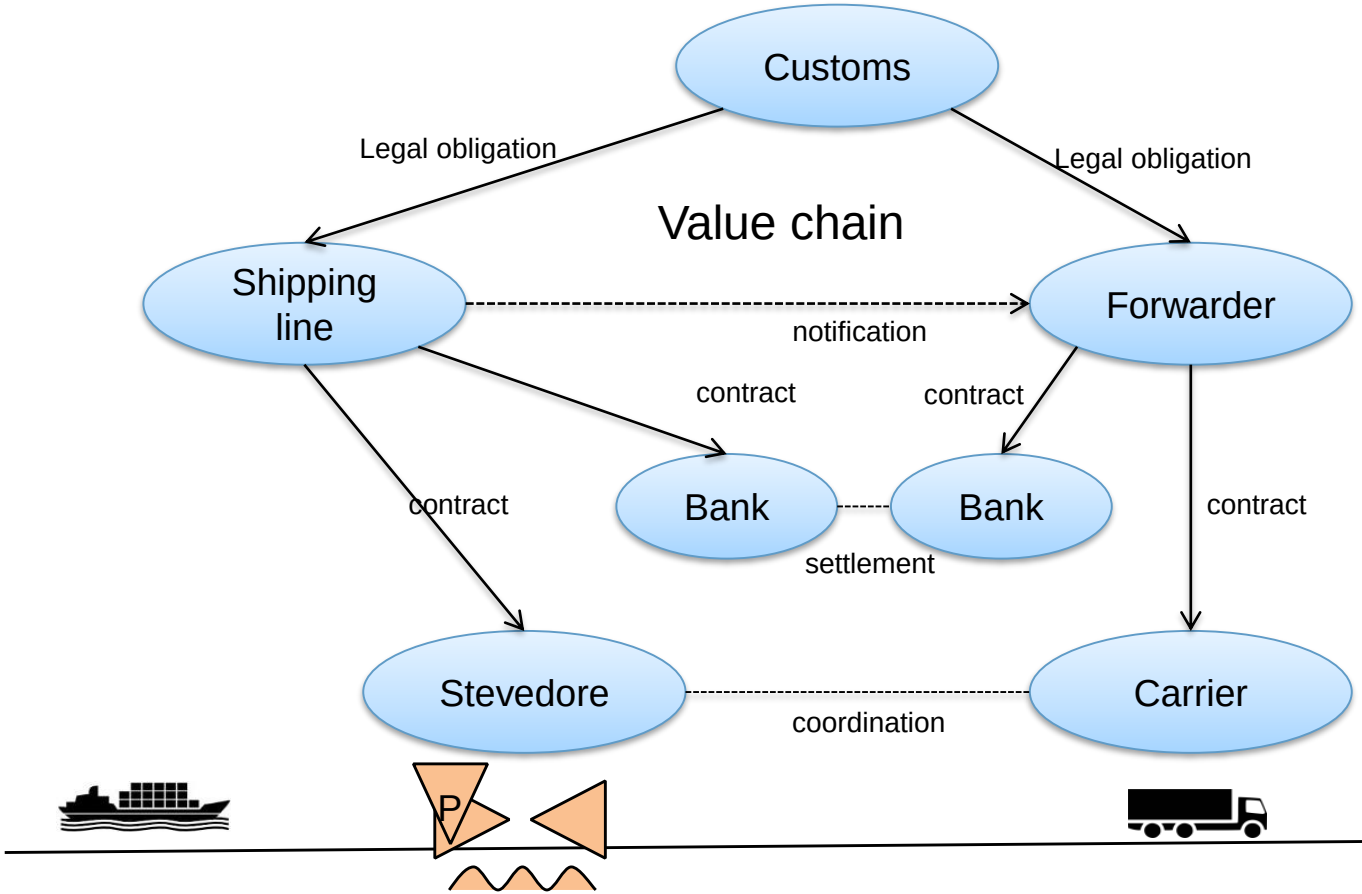
- Hundreds of projects ….
  - http://www.prnewswire.com/news-releases/maersk-and-ibm-unveil-first-industry-wide-cross-border-supply-chain-solution-on-blockchain-300418039.html
  - Ripple – major financial institutions
  - Everledger – diamonds
  - …, and some 200 (or more) digital currencies (very volatile)
- Most on creating a trusted environment for electronic document sharing
  - IBM GTD – trusted document sharing and events (container trip)

- Data management and – governance
  - What is stored in a blockchain and what not? Linked data versus open data
  - Access control – blockchain data or in back office systems
  - Not for streaming data
- Scalability and performance
  - Block size (data management, archiving)
  - Mining – trust and consensus models

WHY USE BLOCKCHAIN

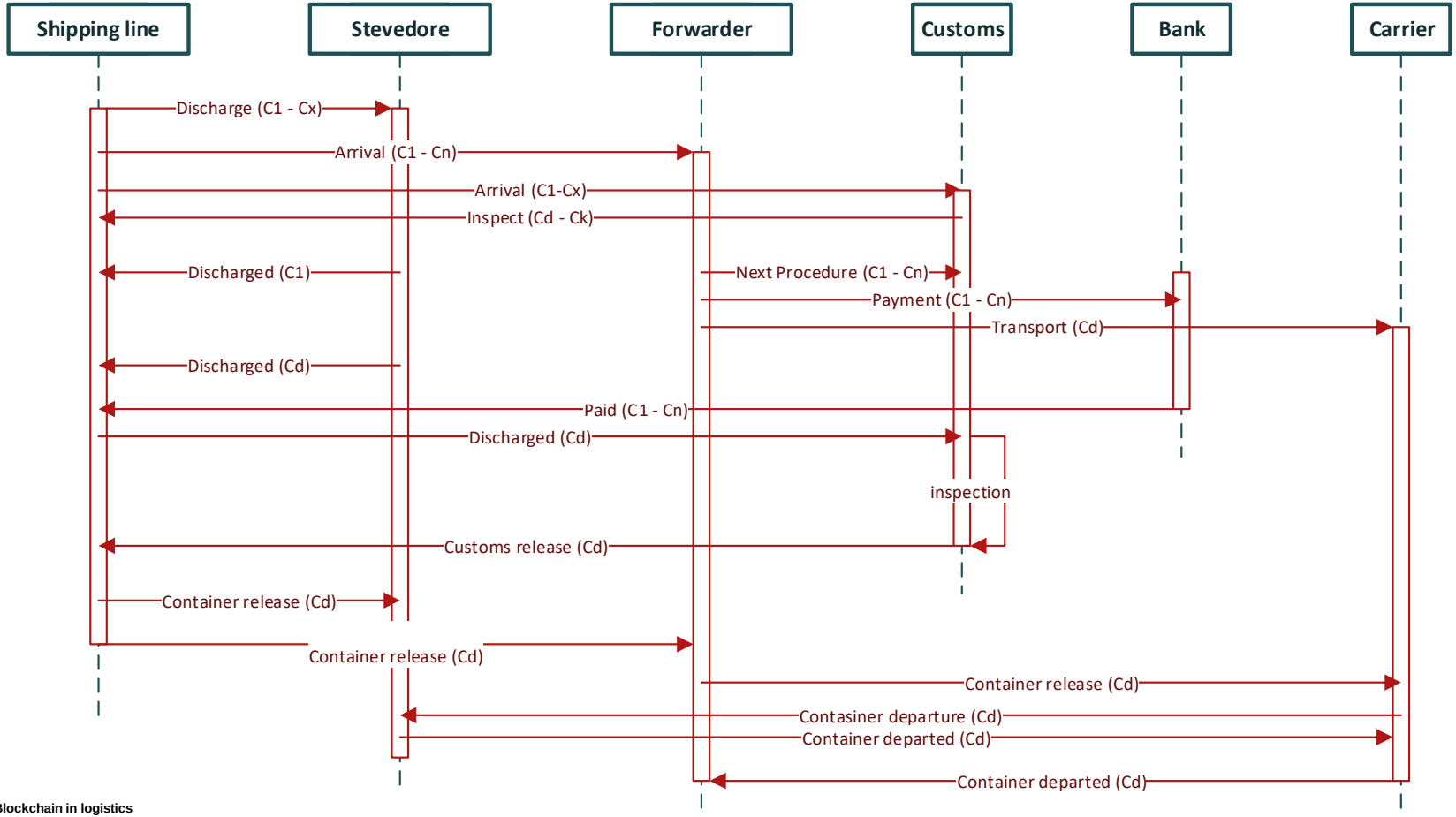
- Trust (and disintermediation)
  - Validation algorithms?
  - Bitcoin
    - one bitcoin can only be spend once
    - The majority provides the truth
- Non-repudiation
  - Immutable data storage
- Interoperability
  - Data stored in a blockchain
  - Data (event ledger) is distributed to all stakeholders (or cloud storage service providers)
  - Smart Contracts/Chain code - operations and data semantics

AN EXAMPLE – CONTAINER TRANS-SHIPMENT IN A PORT



Issues
- Commercial release
- Customs release
- Physical availability

CURRENT SITUATION

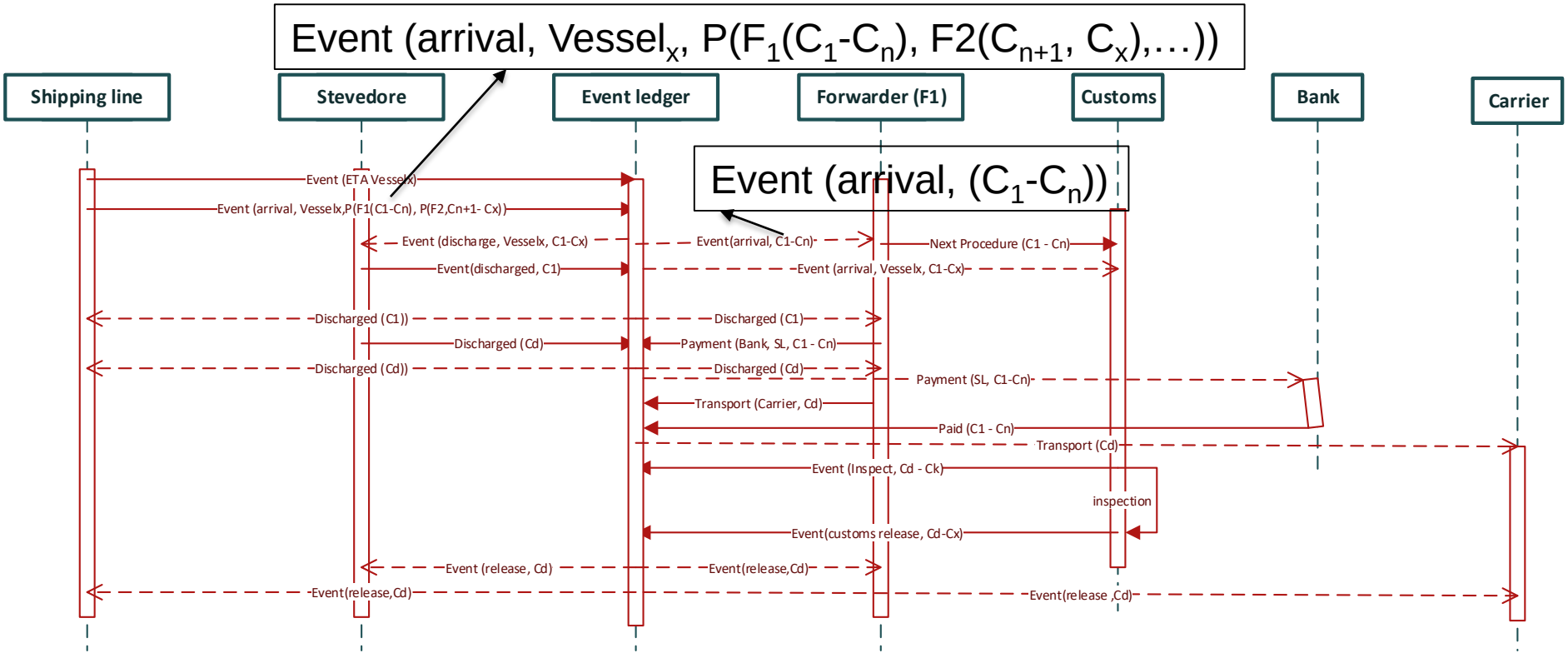


FEATURES

- A community (or commercial) system can be used to distribute data and address various challenges like
  - Many stakeholders – one point of entry
  - Many stakeholders need to share the same data – data distribution, data re-use
  - Different implementation guides of open standards of various stakeholders – data transformations

- Data quality
  - Duplication of data to many stakeholders – errors due to process changes and (no) propagation
  - Status changes are shared too late (e.g. discharge data available after vessel departure) – delays in the process, increased lead times

REAL TIME STATUS SHARING WITH BLOCKCHAIN



Chaincode for all roles, simple data structure (>3.000 lines of code)

‘PERMISSION’ IS THE CORE OF THE SOLUTION

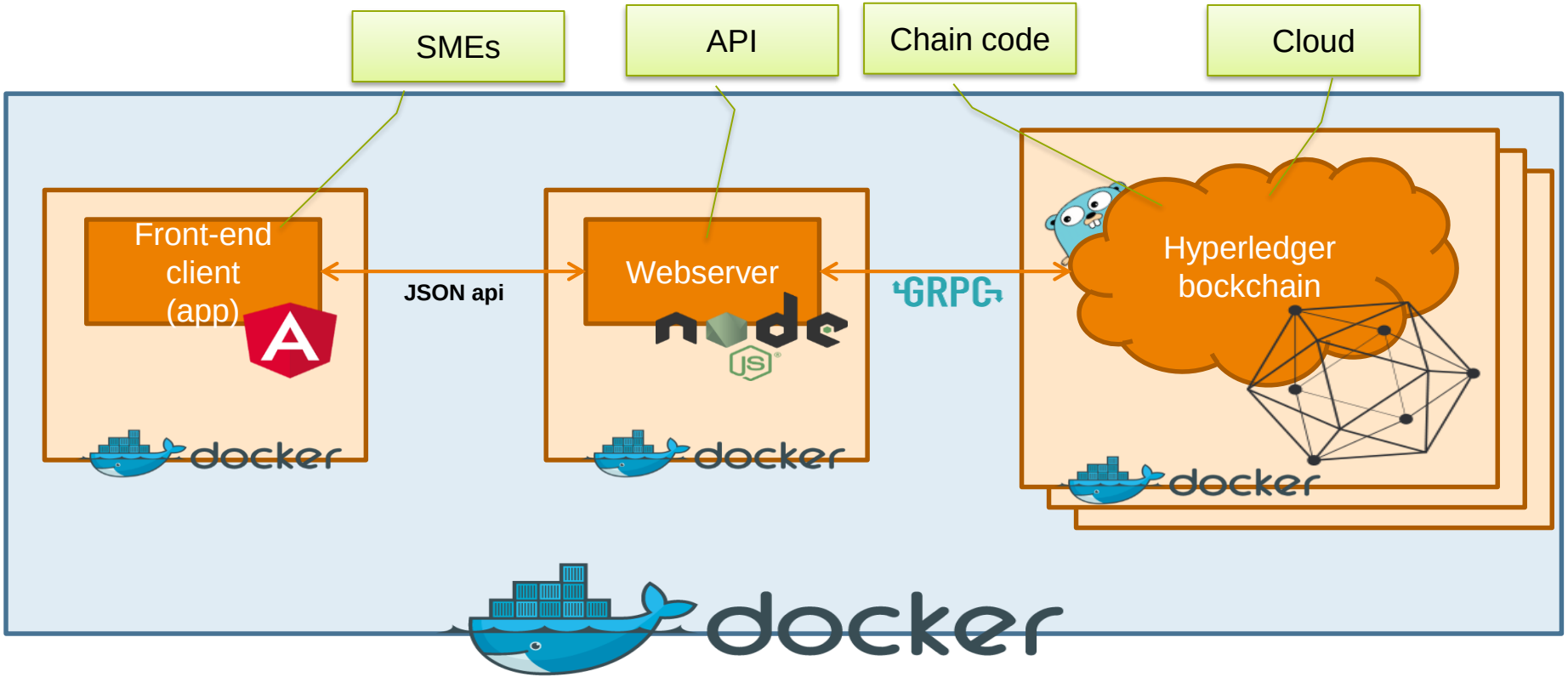
- **Chain code** (compiled smart contracts) defines the behaviour of a role
  - Any type of data can be stored in the blockchain
    - Orders, payments, status, delays
    - Current and future status
  - Data manipulation (view and update)
  - Semantics of data?

- **Permissions** to access data are shared
  - Orders – there is a transactional relation
  - Actively grant permission (Event (arrival, $Vessel_x$, $P(F_1(C_1\text{-}C_n), F2(C_{n+1}, C_x),\ldots)))$

- ‘Separation of concern’ and reputation are the basis for **consensus** (when to add a block to a chain)
  - There are always different roles involved
  - Non-repudiation – all transactions are always available
  - Transactional relations reflect trust and reputation (‘good’ behaviour)
  - Adding sensors (IoT) introduces an extra validation (‘a container can only be at one location at a time’)

- **Register** and **connect** once via the chain code to the blockchain

DEPLOYMENT ON ANY ENVIRONMENT



THE SAME APPROACH CAN BE APPLIED IN OTHER CONSTELLATIONS

- Per modality, e.g. road, rail, inland waterways
- Per hub, e.g. terminal/port, airport, inland terminal, …
- On corridors, like identified in TEN-T
- Cargo types, like dry/liquid bulk, eCommerce, containers
- Supply chains like flowers, electronics, automotive spare parts, ….

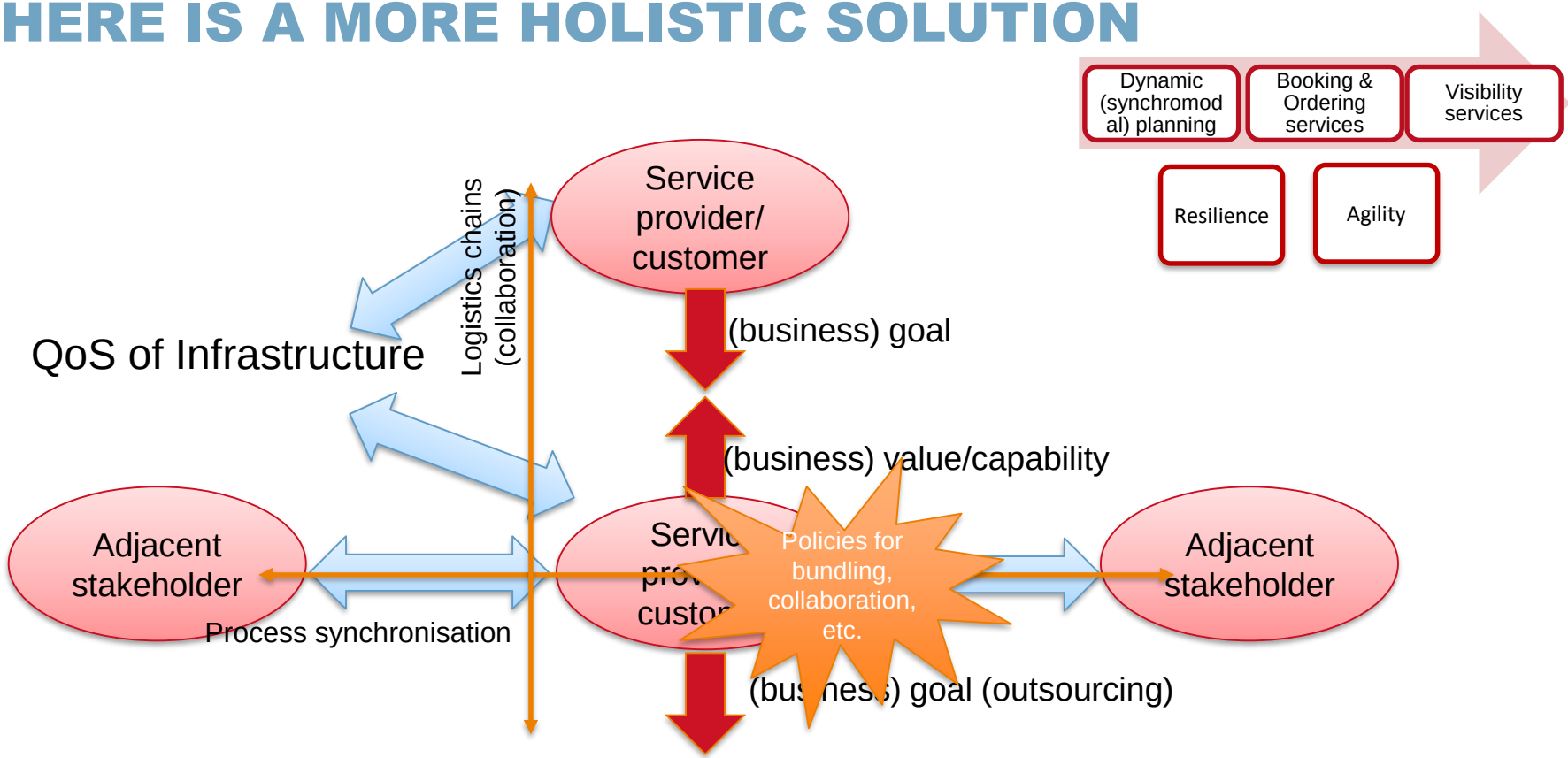
The Hyperledger Ordening service and channels potentially supports this approach

DO WE LET 1000 FLOWERS BLOSSOM ....

Semantic differences.
Differences in functionality.

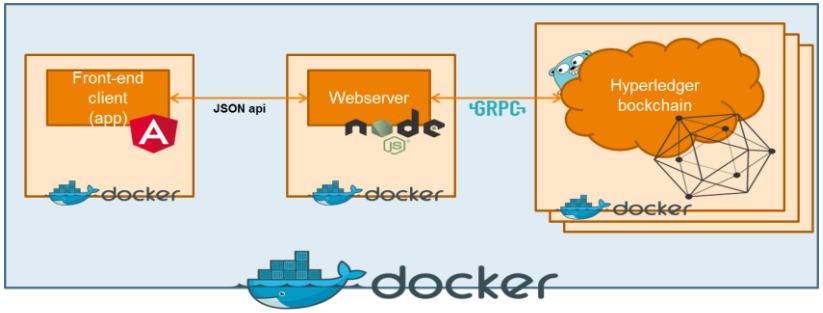
Legacy of the future!

THERE IS A MORE HOLISTIC SOLUTION



CAN WE CREATE A SMART CONTRACT STORE?

- API Registry with supporting Smart Contracts
  - Based on business services for transactions
  - Supporting process synchronisation
  - QoS of infrastructure combined with resilience APIs
- Front-end applications distributed via Appstores
- Gateways to back office systems
  - ontology matching for data transformation (NWO PhD research)
  - BPM orchestration (BPEL) for API integration
- Requires conceptual standards
  - From business services to (existing) open standards/ontologies
  - Interaction choreography for transaction support (rules)
  - Developed for EC DG Move in DTLF



- Value chain and business models for a global logistics blockchain?
- Requirements to event ledger storage (trust and non-repudiation of the store)?
- Which blockchain technology?
- Software engineering and development concepts (data, process, human interface, decomposition)?


THANK YOU FOR YOUR ATTENTION
TNO innovation for life

Towards the Physical Internet with Coloured Petri Nets

Angela Di Febbraro, **Davide Giglio**, and **Nicola Sacco**

Department of Mechanical, Energy, Management, and Transportation Engineering
University of Genova, Italy

IPIC 2017 – 4th International Physical Internet Conference

July 4-6, 2017
Graz University of Technology, Austria

Outline

Introduction

Motivations

- The road to the Physical Internet will change drastically most of the production and logistic processes that actually characterize the supply chain
- Need of effective modelling tools which allow analysing the performance of such kind of interoperating systems

Goal of this work

- Investigate the applicability of the Petri net formalism to the Physical Internet paradigm

source
PHYSICAL INTERNET INITIATIVE

source
PETRI NET WORLD

Introduction

Physical Internet $\leftrightarrow$ Complex DES $\leftrightarrow$ **Petri Nets**

Petri nets have been proven to be a valuable and powerful tool for design, analysis, and control of discrete-event systems

Petri nets have been successfully applied to:

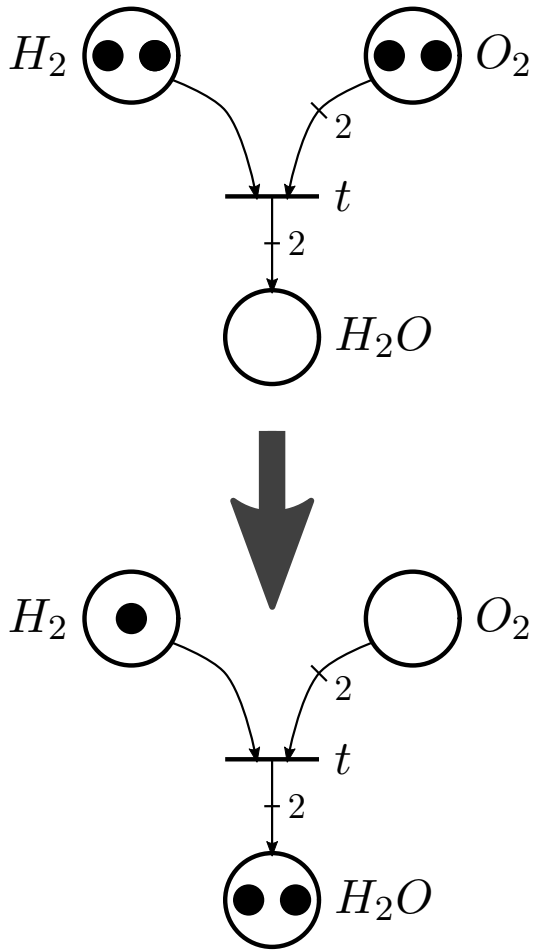
- manufacturing and production systems
- communication protocols
- indoor transportation and outdoor traffic systems
- supply chains and logistic systems
- ...

Basics of Petri Nets

Definition

A Petri net is a 5-tuple $\boldsymbol{PN} = \{P, T, F, W, M_0\}$ where:

- $\boldsymbol{P} = \{p_1, p_2, \ldots, p_n\}$ is a finite set of **places**
- $\boldsymbol{T} = \{t_1, t_2, \ldots, t_m\}$ is a finite set of **transitions**
- $\boldsymbol{F} \subseteq (P \times T) \cup (T \times P)$ is a set of **arcs**
- $\boldsymbol{W} : F \to \{1, 2, 3, \ldots\}$ is a weight function
- $\boldsymbol{M_0} : P \to \{0, 1, 2, 3, \ldots\}$ is the initial **marking**



source
MURATA (1989)

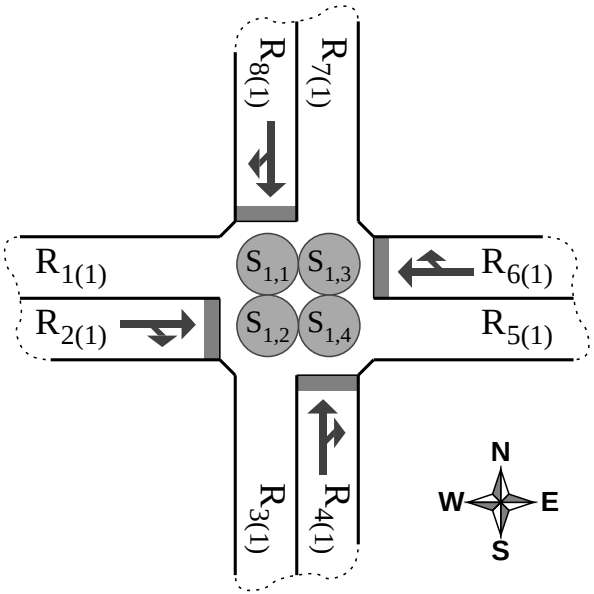
Basics of Petri Nets

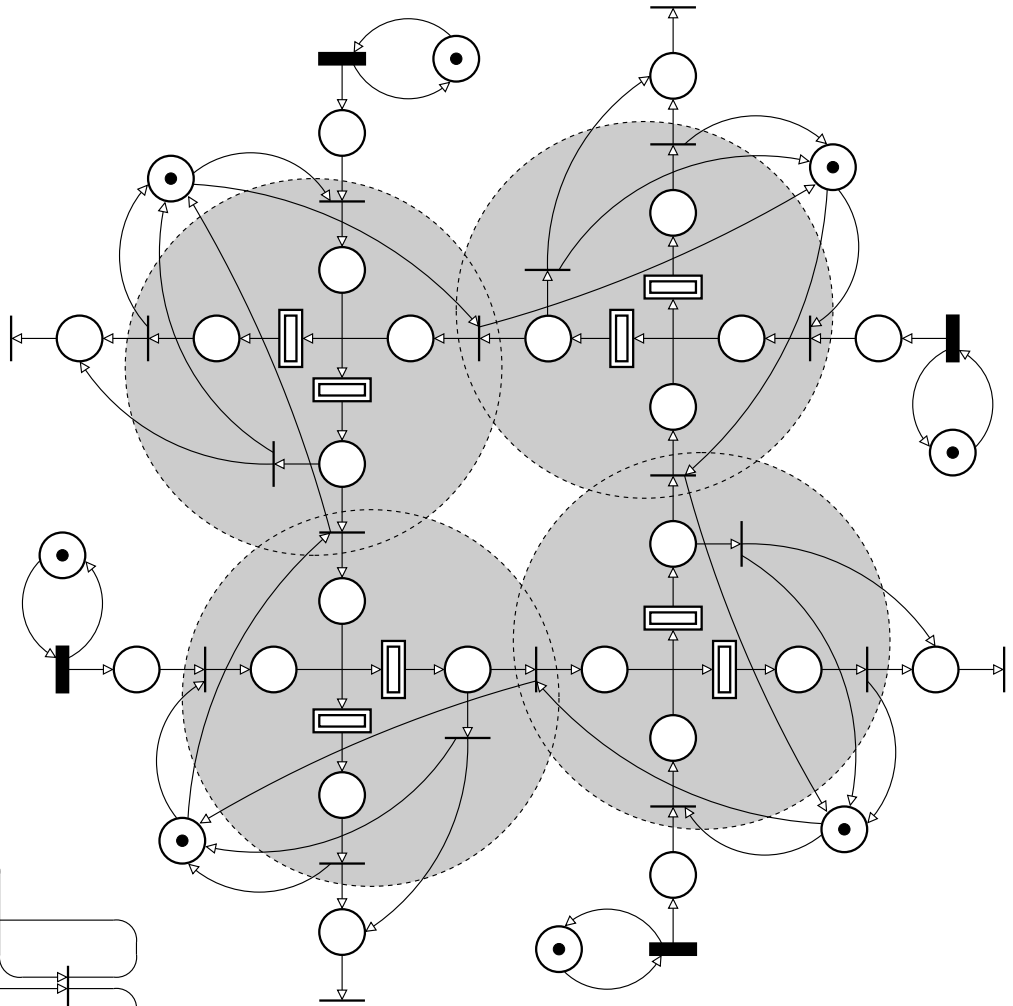
Once defined the Petri net model of a system, it is possible:

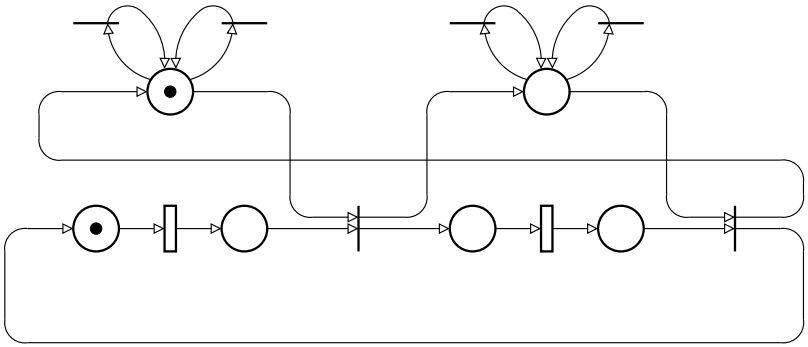
- to analyse the **structural properties** of the system (boundedness of places, liveness of transitions, possible presence of deadlock, reversibility)
- to compute some **performance indexes** of the system, such as throughput, average number of items, etc.
- to **simulate** the system by using the PN as an intermediate model between real world and simulation tools
- to **optimise** some parameters of the system by using the Petri net within an optimisation problem
- to define and implement **supervisory control strategies** aimed at avoiding dangerous states, forbidden states, and deadlock states

Some applications of Petri Nets

Urban traffic system

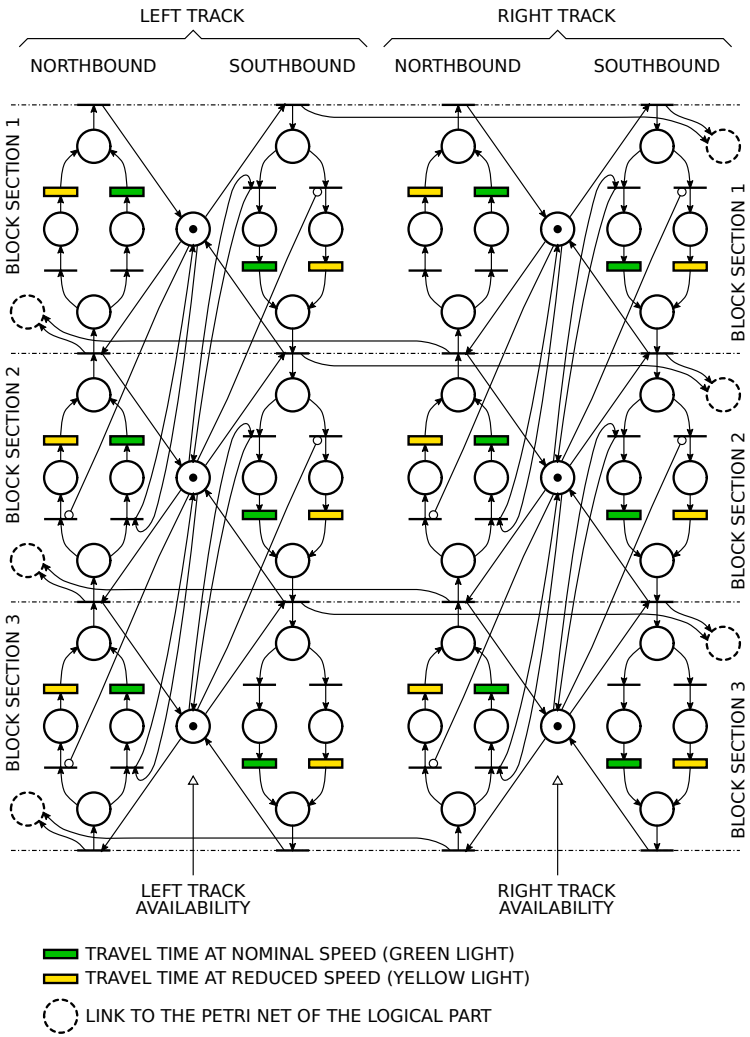


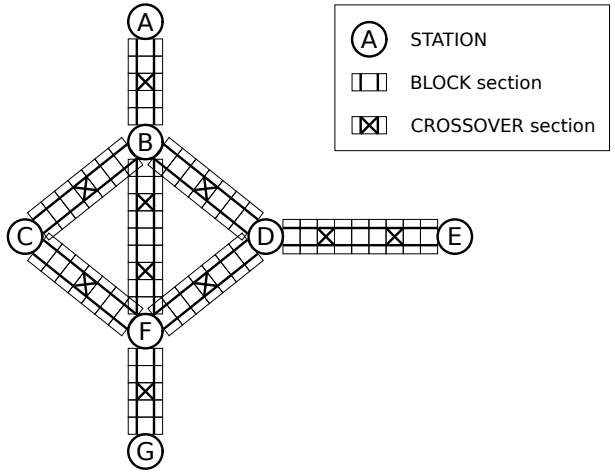


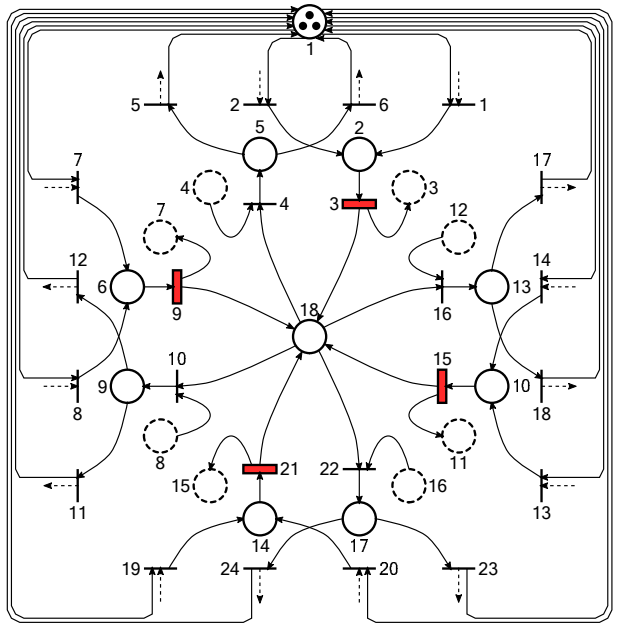


Some applications of Petri Nets

Railway network

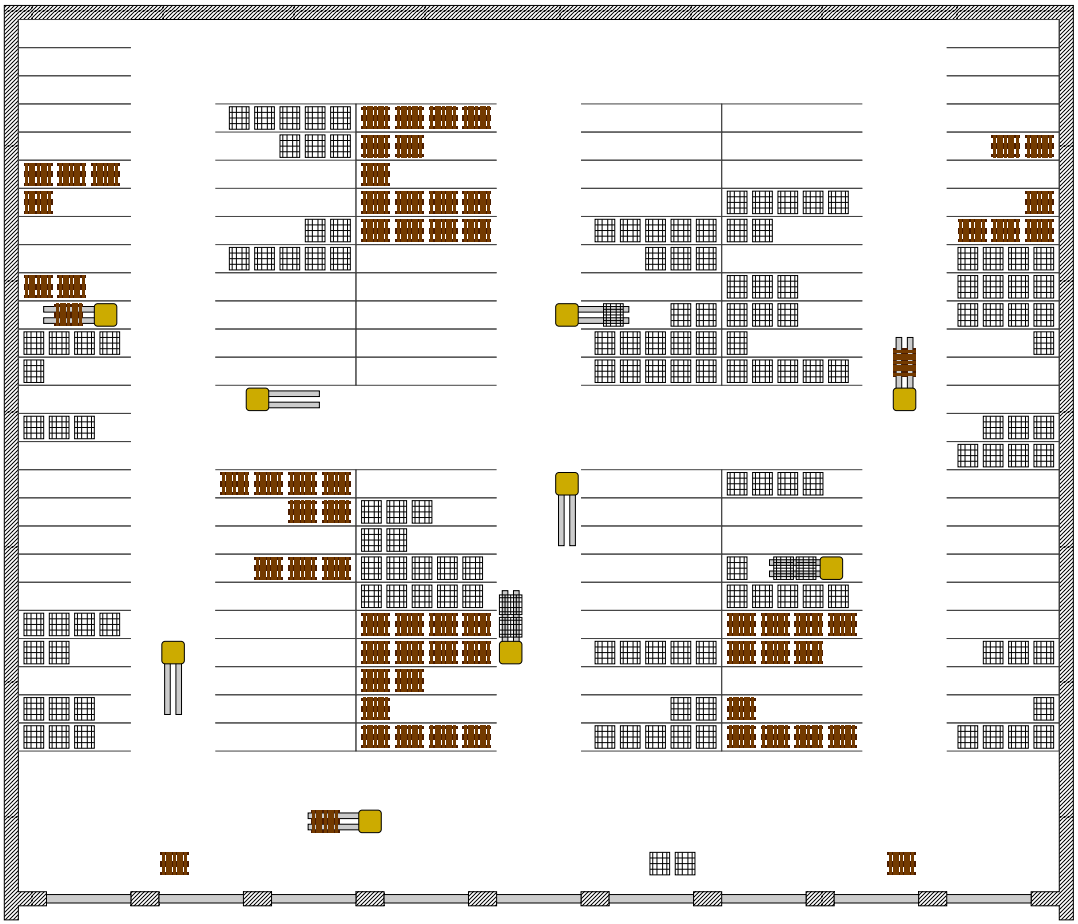


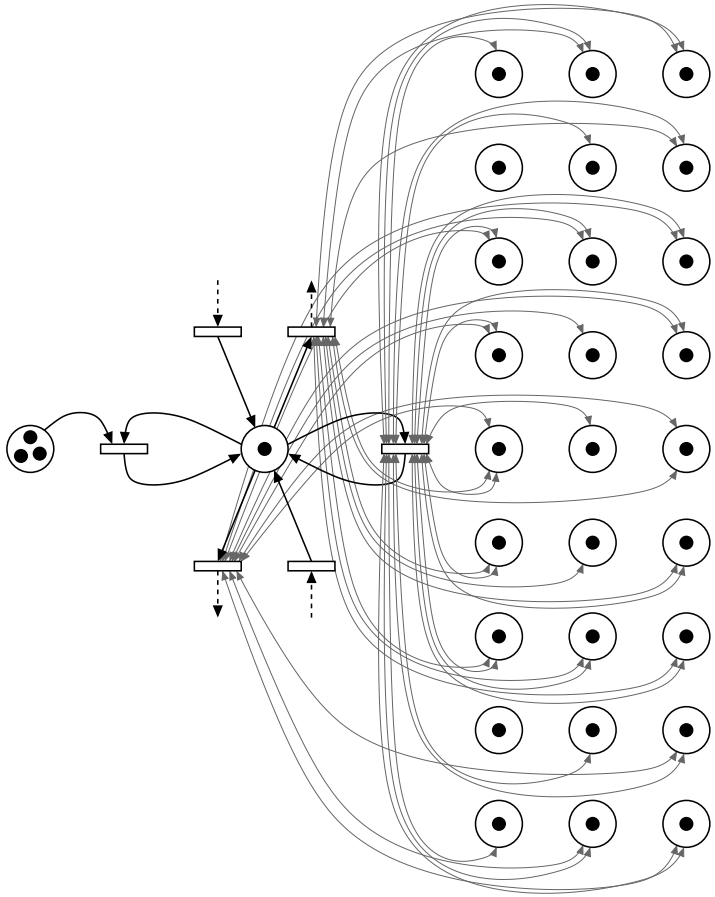




Some applications of Petri Nets

Indoor distribution warehouse





Petri nets

In the **Physical Internet**

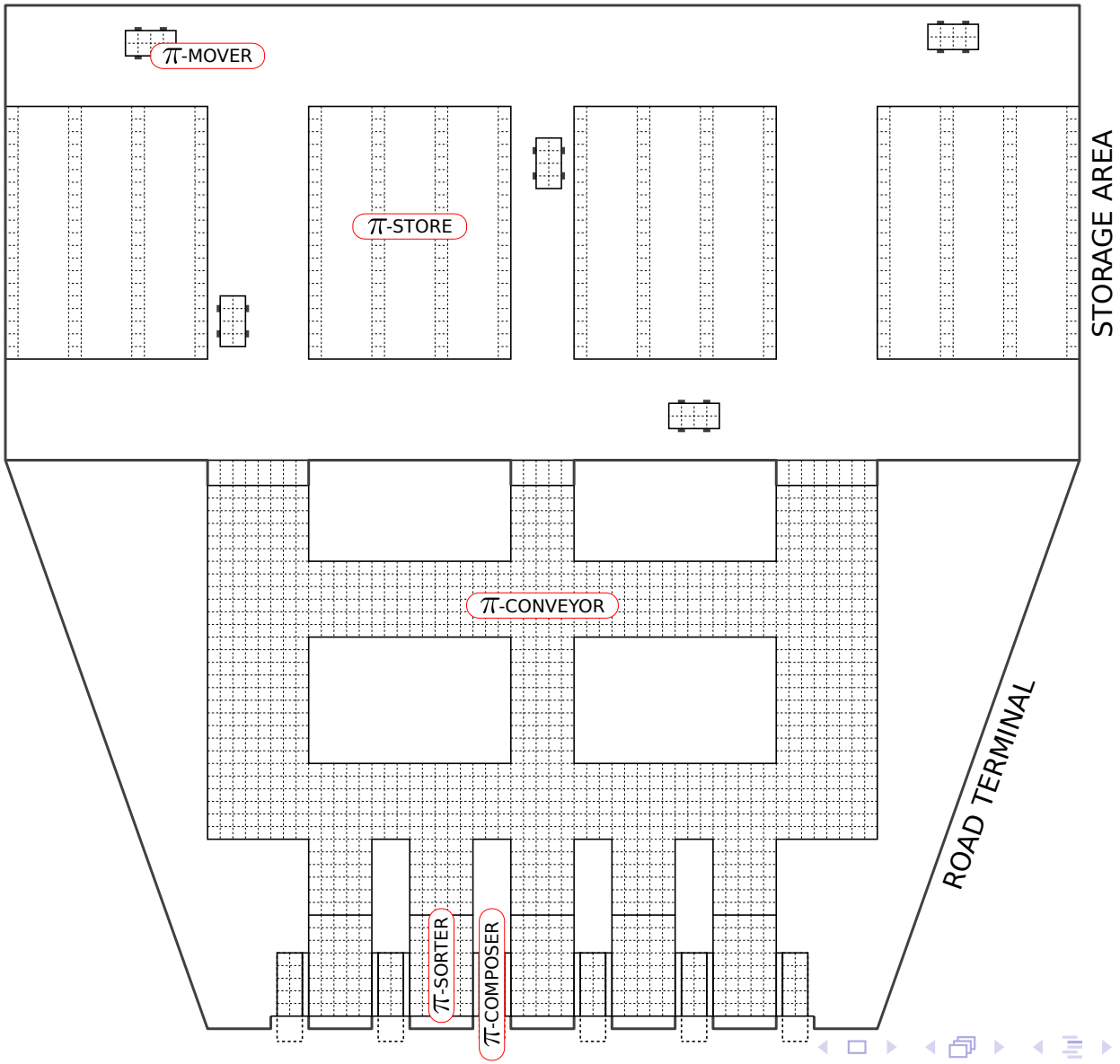
- Petri nets seem particularly appropriate to represent the dynamics of π-containers within the PI
  - they are able to capture precedence relations and interactions among events which characterize facilities and infrastructures (multimodal logistics centres and hubs, transit centres, roads and railways) through which π-containers are delivered
- Coloured Petri Nets (CPN) are especially suitable to model the different kinds of π-containers and the operations required by them and carried out by π-movers, π-conveyors, π-stores, ...

Contribution of the paper

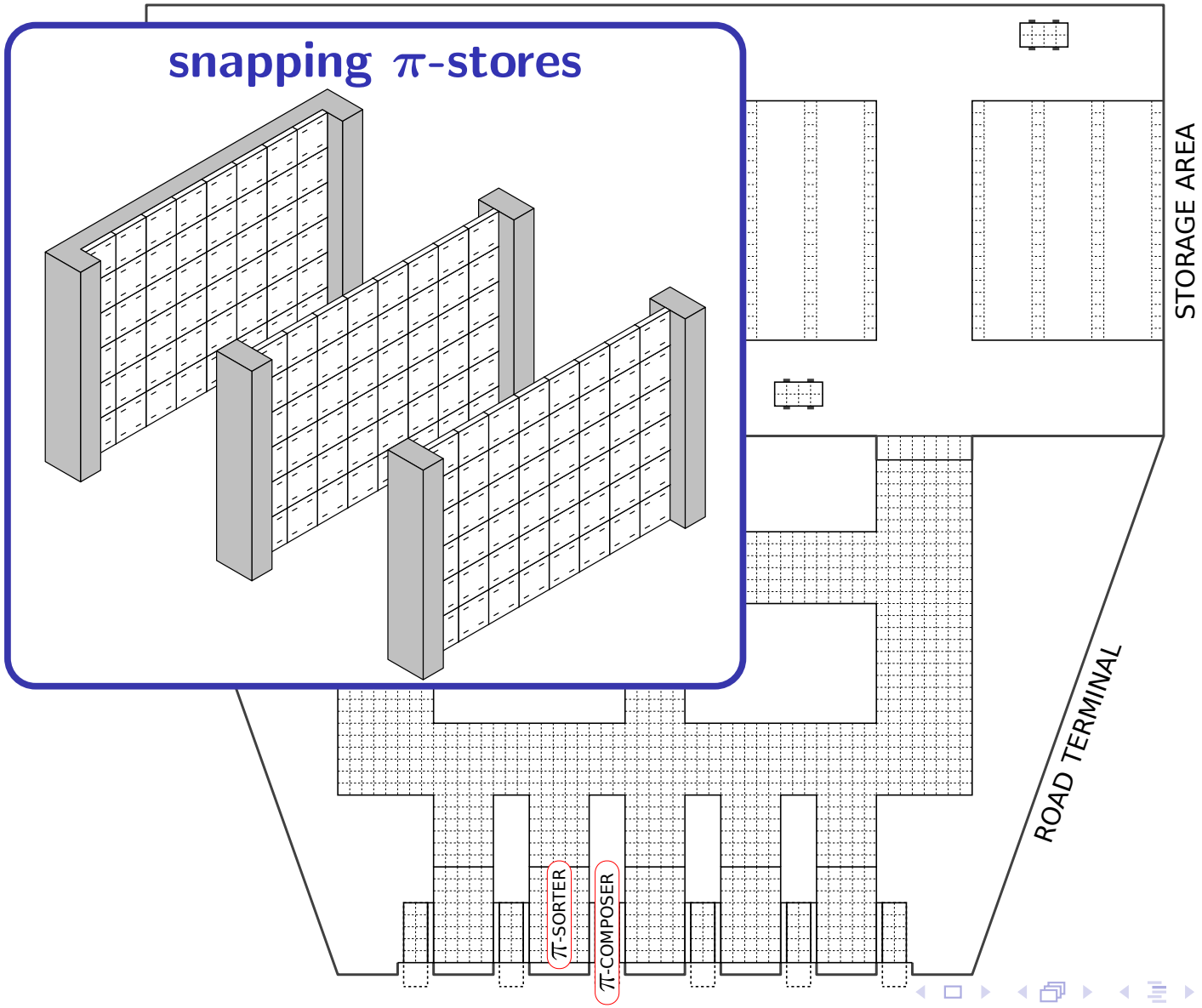
- CPN model of an example of π-conveyor
- CPN model of an example of π-sorter/π-composer
- Formalization of "π-core"

1. Introduction
   - Basics of Petri Nets and some applications
2. **The model of the multimodal hub**
   - π-core
   - π-containers (composed)
3. The coloured Petri net (CPN)
   - π-conveyor
   - π-sorter/π-composer
4. Conclusions
   - Further research directions

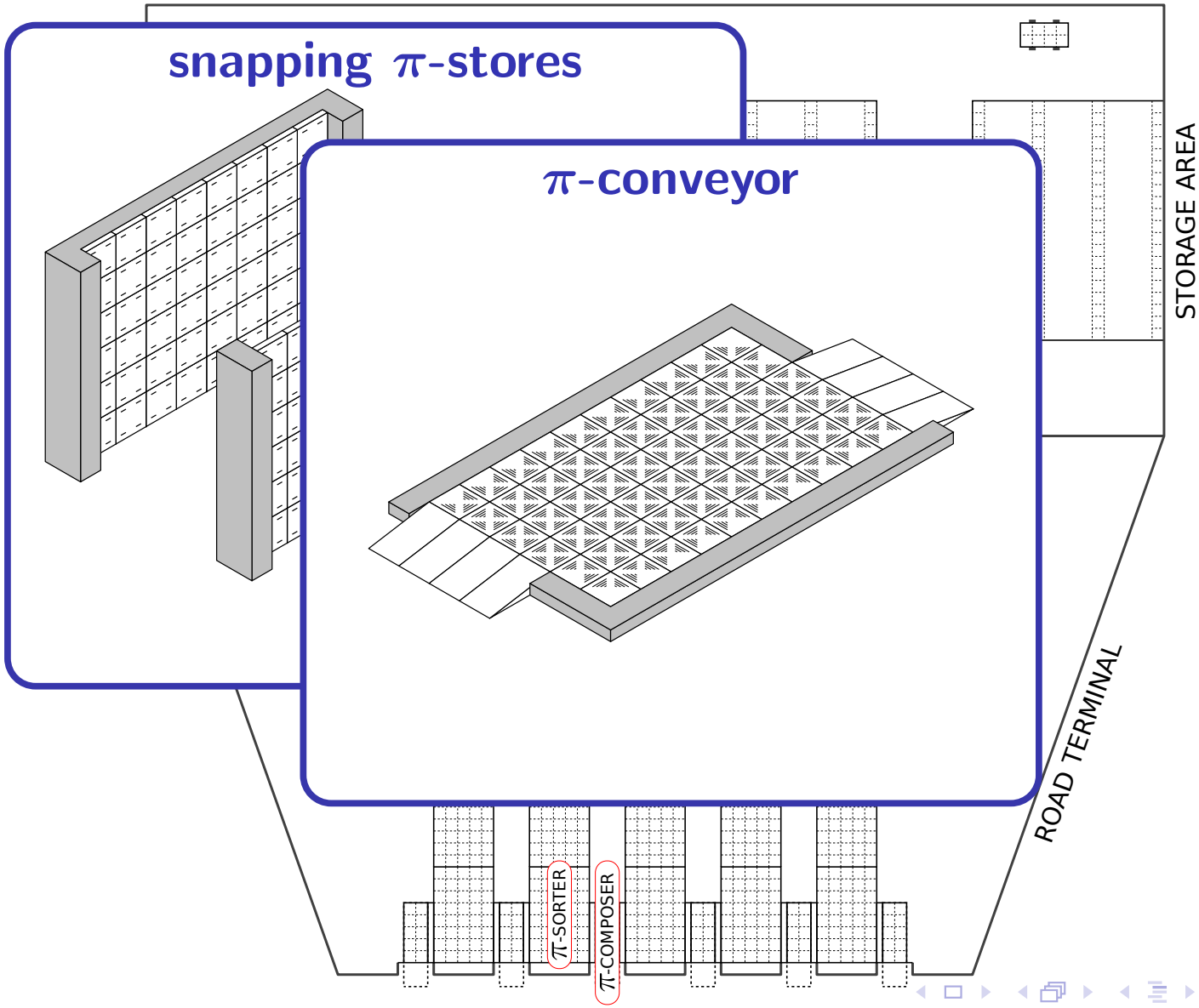
The model of the multimodal hub



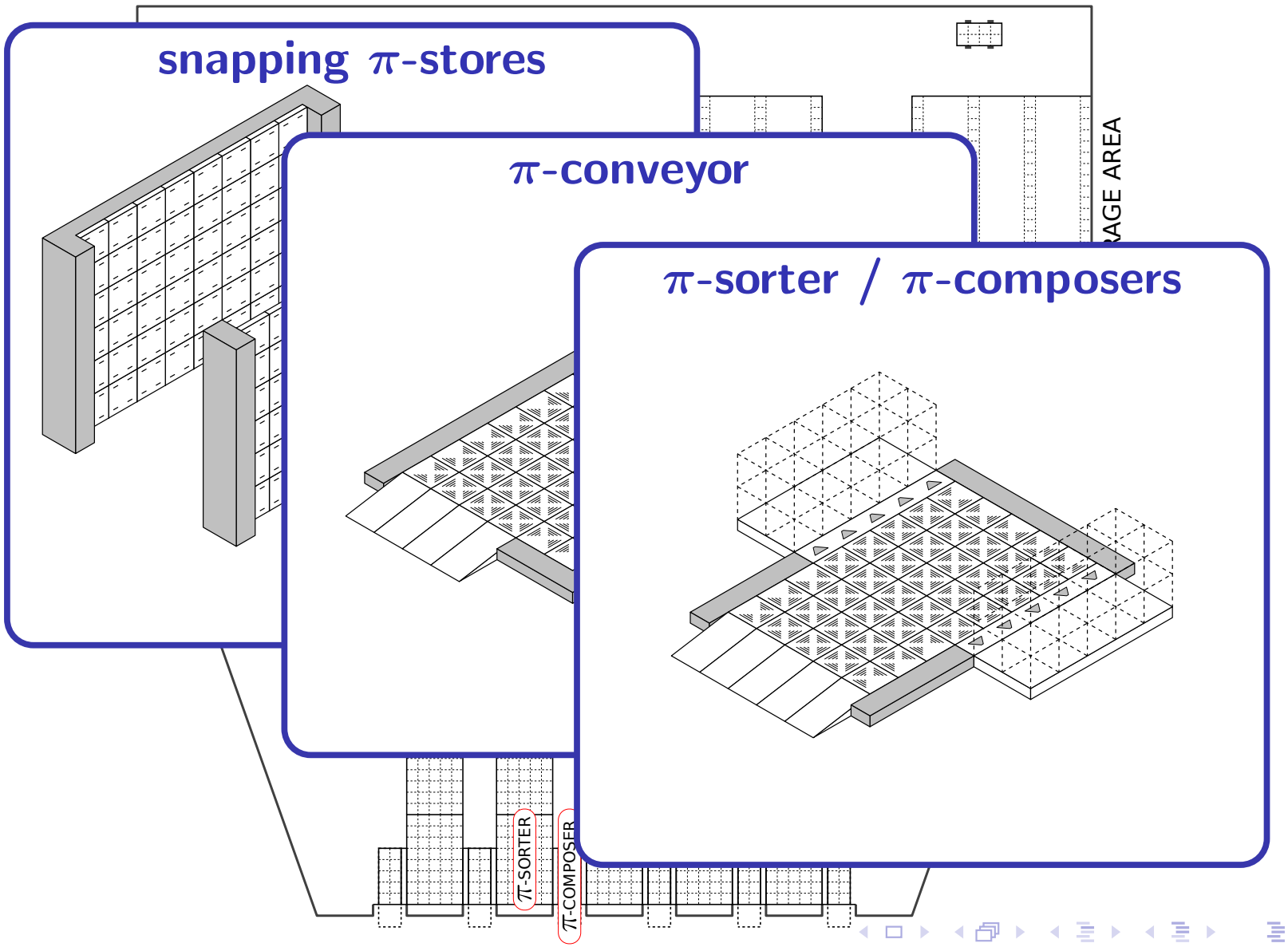
The model of the multimodal hub



The model of the multimodal hub

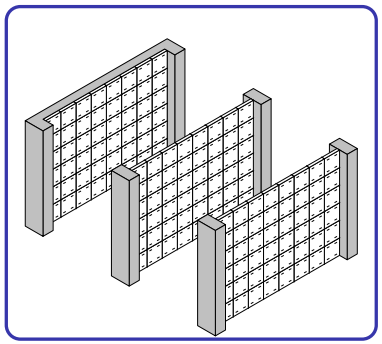
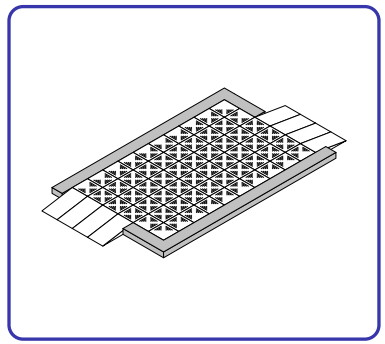
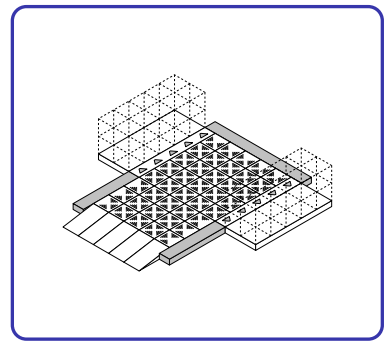


The model of the multimodal hub

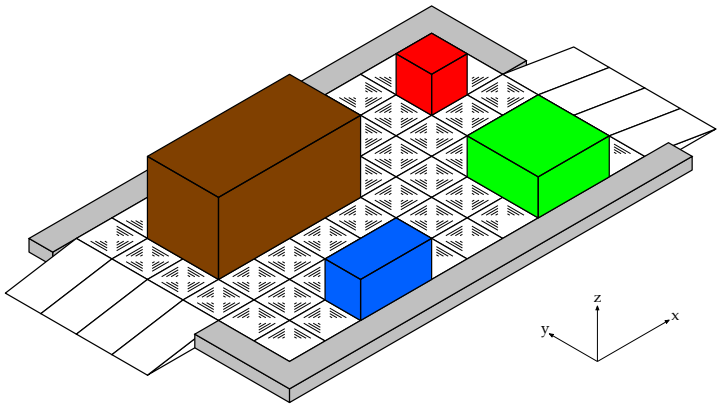


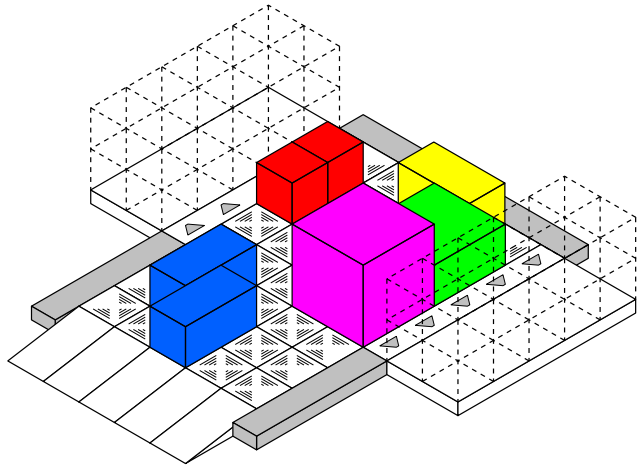
π-core (of π-containers)

In the new logistics facilities and material handling systems that are compatible with the PI paradigm, many π-nodes and π-movers will be built exploiting the concept of **π-cell**

π-containers occupy one or more cells



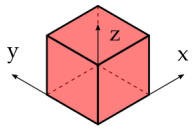


π-core (of π-containers)

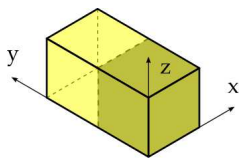
To keep the model simple, it is essential to represent, in the CPN, a π-container with a single coloured token

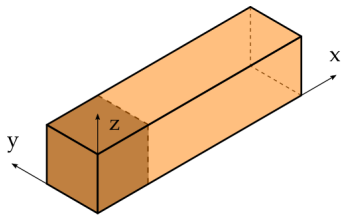
We solved this issue by explicitly representing the **π-core** of a π-container (with the coloured token) and implicitly considering the rest of the π-container

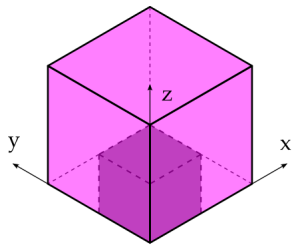
- the size of the π-cells determine the π-container of unitary size



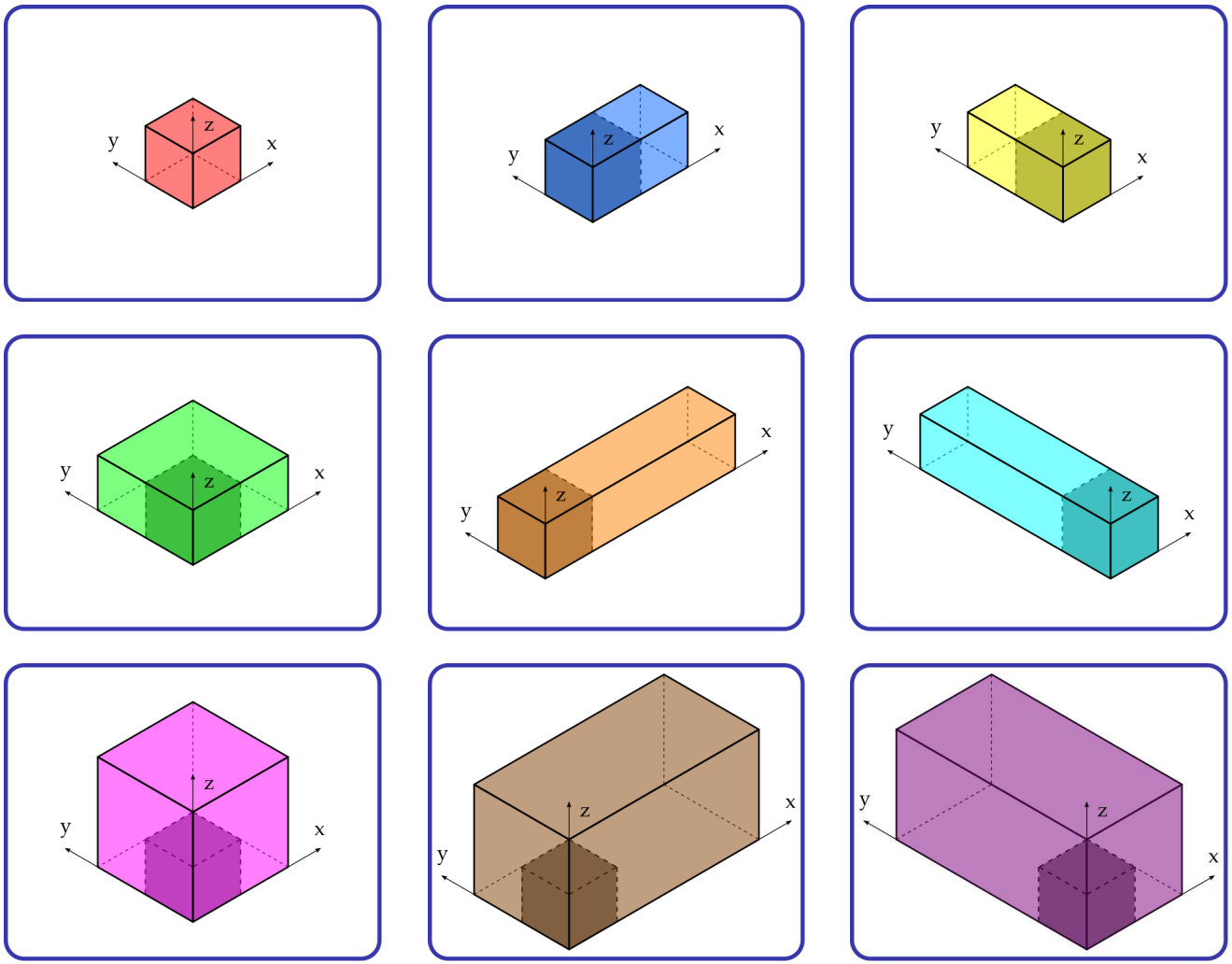
- the π-core is the "cube" of unitary size inside the π-container, which is conventionally located at the origin of the 3d axes







π-core (of π-containers)

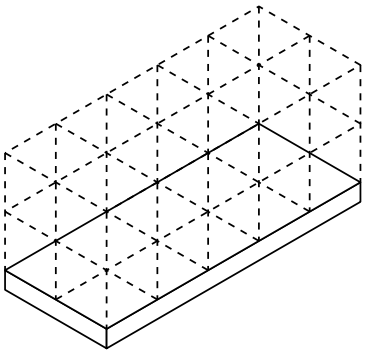


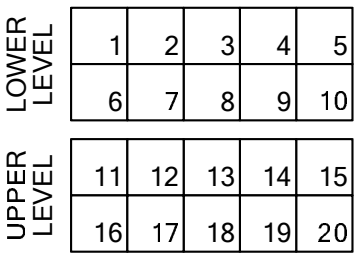
- size 111 — 2.4×2.4×2.4 m²
- size 211 — 4.8×2.4×2.4 m²
- size 121 — 2.4×4.8×2.4 m²
- size 221 — 4.8×4.8×2.4 m²
- size 411 — 9.6×2.4×2.4 m²
- size 141 — 2.4×9.6×2.4 m²
- size 222 — 4.8×4.8×4.8 m²
- size 422 — 9.6×4.8×4.8 m²
- size 242 — 4.8×9.6×4.8 m²

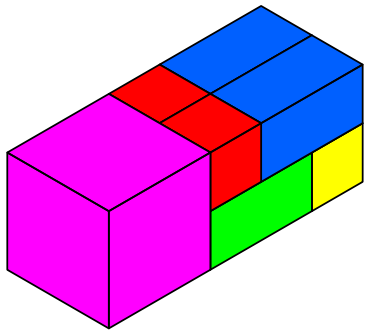
Each colour specifies a size of π-containers
(a coloured token represents a π-container of certain size)

π-containers (composed)

Another fundamental aspect in the Physical Internet paradigm is the possibility of **composing containers** of standard size by suitably attaching π-containers of different sizes

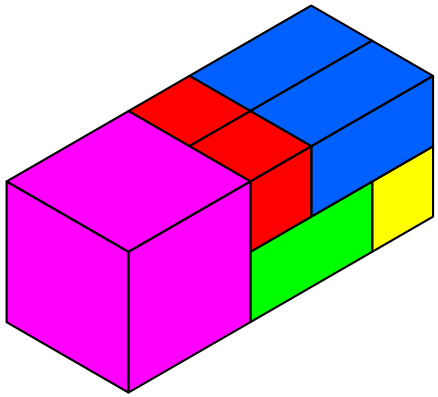






In the proposed CPN representation, the composed container will be modelled with a single coloured token whose colour is a vector of elements describing the way the container is composed

π-containers (composed)



This composed container is represented by the vector
(0,0,0,0,0,222,0,221,0,121,0,0,111,211,0,0,0,111,211,0)
that corresponds to the scheme

0	0	0	0	0
222	0	221	0	121

LOWER LEVEL (cells #1÷#10)

0	0	111	211	0
0	0	111	211	0

UPPER LEVEL (cells #11÷#20)

A single token with structured color is able to represent the whole composed container!

π-containers (composed)

The **sequence of loads** of the single π-containers (carried out by the π-composer) is necessary to define the **guard functions** of some transitions in the CPN which models the π-composer

It is defined as an ordered sequence of integer numbers in the range [1,20]

- a number i in the j-th position of the sequence means that the π-container whose π-core has to be placed in the i-th cell of the composed container must be the j-th in the loading sequence
- previous example: sequence (6,8,10,18,13,19,14)

All possible loading sequences can be a-priori defined on the basis of the allowed structures of composed containers

1. Introduction
   - Basics of Petri Nets and some applications
2. The model of the multimodal hub
   - π-core
   - π-containers (composed)
3. **The coloured Petri net (CPN)**
   - π-conveyor
   - π-sorter/π-composer
4. Conclusions
   - Further research directions

Coloured Petri nets

Definition

A Coloured Petri Net is a 9-tuple $\{P, T, \mathcal{A}, \Sigma, \mathcal{V}, \mathcal{C}, G, E, I\}$ where:

- $\boldsymbol{P}$ is a finite set of **places**
- $\boldsymbol{T}$ is a finite set of **transitions**
- $\boldsymbol{\mathcal{A}} \subseteq (P \times T) \cup (T \times P)$ is a set of directed **arcs**
- $\boldsymbol{\Sigma}$ is a finite set of non-empty **colour sets**
- $\boldsymbol{\mathcal{V}}$ is a finite set of **typed variables**
- $\boldsymbol{\mathcal{C}} : P \to \Sigma$ is the **colour set function**
- $\boldsymbol{G} : T \to \text{Expr}_{\mathcal{V}}$ is the **guard function**
- $\boldsymbol{E} : A \to \text{Expr}_{\mathcal{V}}$ is the **arc expression function**
- $\boldsymbol{I} : P \to \text{Expr}_{\emptyset}$ is the **initialization function**

Coloured Petri nets

In the proposed CPN model

- **Colour sets**

$$\sigma = \{\text{BSIZE}, \text{BSIZE0}, \text{STRUCT}\}$$

with:

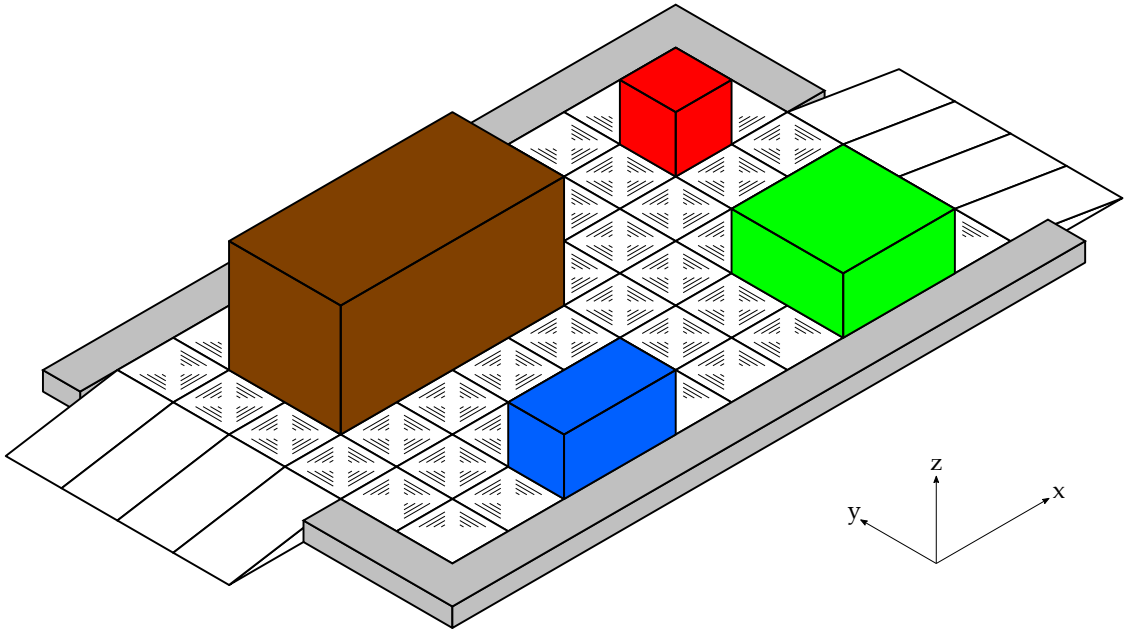
$$\text{BSIZE} = \{111, 211, 121, 221, 411, 141, 222, 422, 242\}$$
$$\text{BSIZE0} = \text{BSIZE} \cup \{0\}$$
$$\text{STRUCT} = \text{BSIZE0}^{20} = \text{BSIZE0} \times \text{BSIZE0} \times \ldots \times \text{BSIZE0 (20 times)}$$

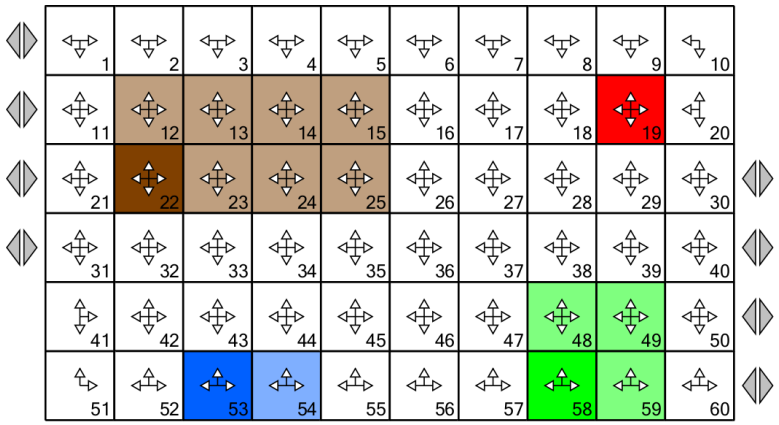
- **Guard functions**

Defined to prevent physical conflicts between π-containers

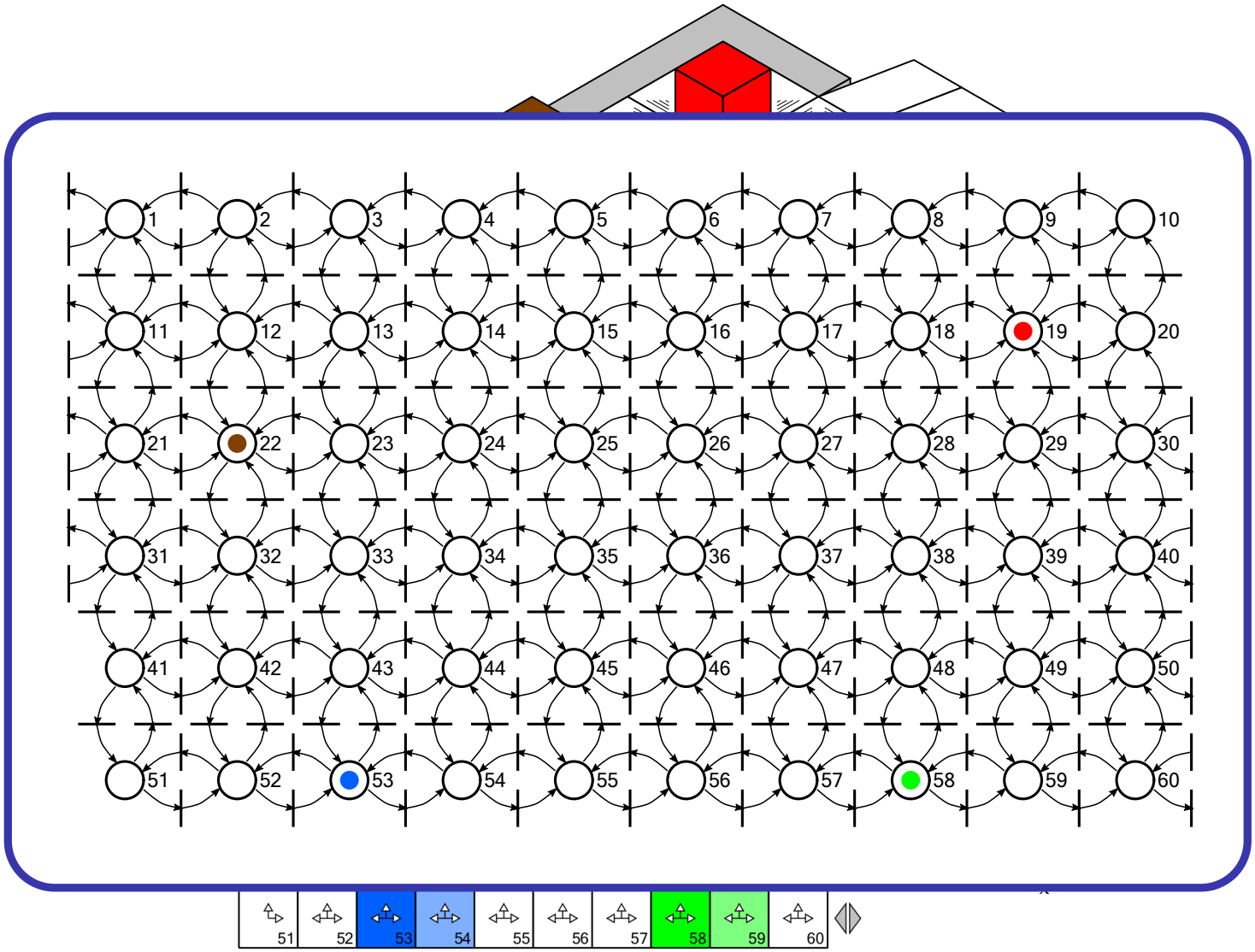
$\Rightarrow$ the movements of coloured tokens within the CPN actually correspond to feasible movements of π-containers

π-conveyor

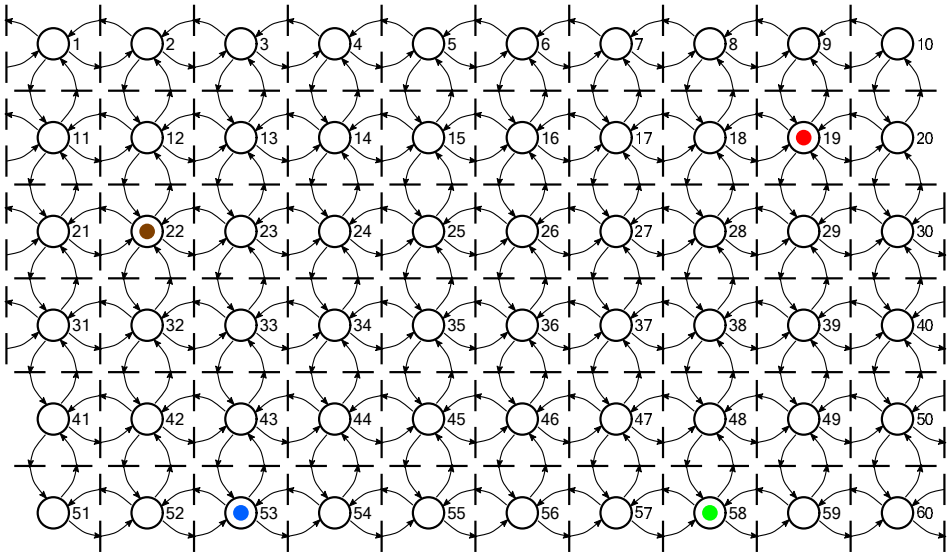

z
x
y


1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
y
x

π-conveyor


1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60

π-conveyor



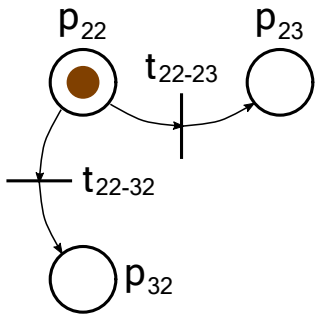
$$\mathcal{C}(p_h) = \mathsf{BSIZE}, \forall h = 1, \ldots, 60 \qquad \mathcal{V} = \{b_h : \mathsf{BSIZE}; h = 1, \ldots, 60\}$$

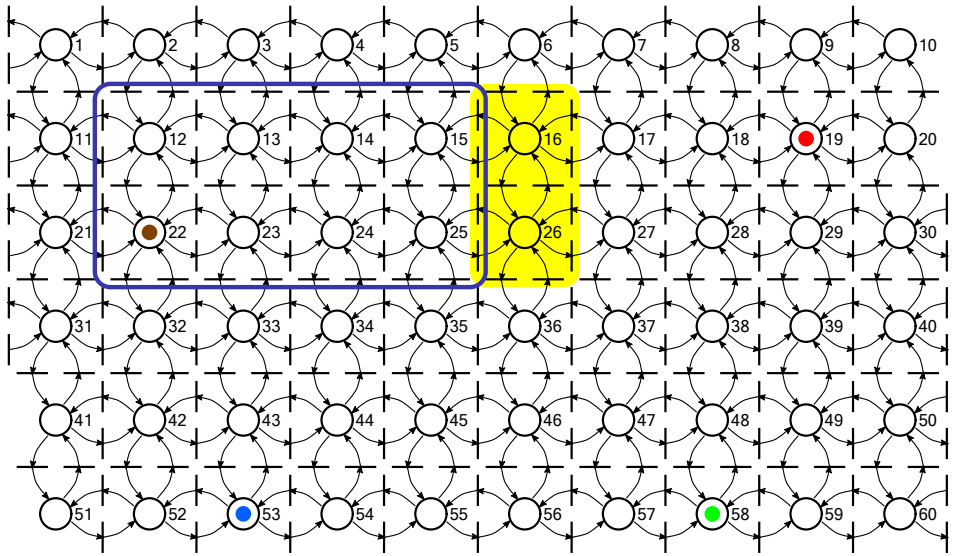
$$E(p_h, t_{h-k}) = 1'b_h, \forall h, k = 1, \ldots, 60 \qquad E(t_{h-k}, p_k) = 1'b_h, \forall h, k = 1, \ldots, 60$$

$$G(t_{h-k}) = [(b_h = 111) \wedge (C_{h-k-111})] \vee [(b_h = 211) \wedge (C_{h-k-211})] \vee [(b_h = 121) \wedge (C_{h-k-121})] \vee [(b_h = 221) \wedge (C_{h-k-221})] \vee [(b_h = 411) \wedge (C_{h-k-411})] \vee [(b_h = 141) \wedge (C_{h-k-141})] \vee [(b_h = 222) \wedge (C_{h-k-222})] \vee [(b_h = 422) \wedge (C_{h-k-422})] \vee [(b_h = 242) \wedge (C_{h-k-242})], \forall h, k = 1, \ldots, 60$$

π-conveyor

Brown π-container (size 422) which moves eastbound

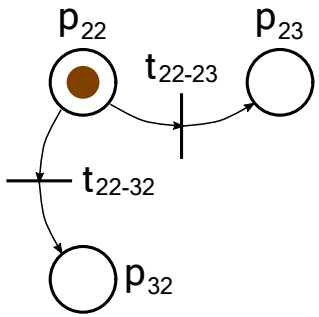


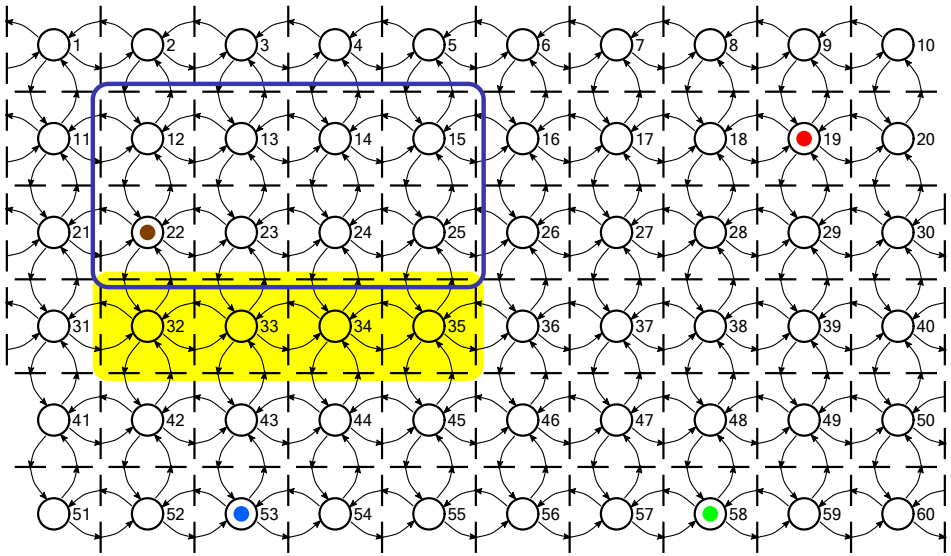


$$C_{22-23-422} = [(b_{16} \neq 111) \vee (b_{16} \neq 121) \vee (b_{16} \neq 211) \vee (b_{16} \neq 221) \vee (b_{16} \neq 411) \vee (b_{16} \neq 222) \vee (b_{16} \neq 422)] \wedge [(b_{26} \neq 111) \vee (b_{26} \neq 121) \vee (b_{26} \neq 211) \vee (b_{26} \neq 221) \vee (b_{26} \neq 411) \vee (b_{26} \neq 222) \vee (b_{26} \neq 422)] \wedge [(b_{36} \neq 121) \vee (b_{36} \neq 221) \vee (b_{36} \neq 141) \vee (b_{36} \neq 222) \vee (b_{36} \neq 422) \vee (b_{36} \neq 242)] \wedge [(b_{46} \neq 141) \vee (b_{46} \neq 242)] \wedge [(b_{56} \neq 141) \vee (b_{56} \neq 242)]$$

π-conveyor

Brown π-container (size 422) which moves southtbound

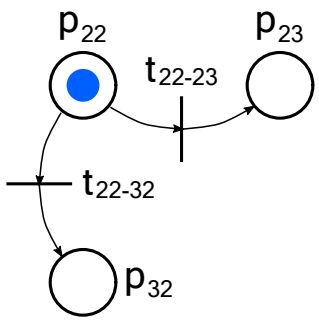


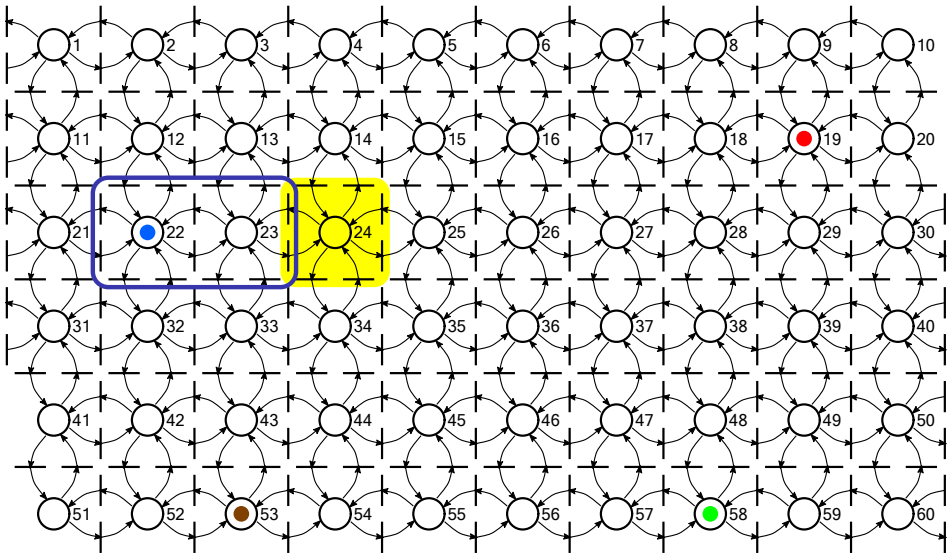


$$C_{22-32-422} = [(b_{31} \neq 211) \vee (b_{31} \neq 411)] \wedge [(b_{32} \neq 111) \vee (b_{32} \neq 211) \vee (b_{32} \neq 411)] \wedge [(b_{33} \neq 111) \vee (b_{33} \neq 211) \vee (b_{33} \neq 411)] \wedge [(b_{34} \neq 111) \vee (b_{34} \neq 211) \vee (b_{34} \neq 411)] \wedge [(b_{35} \neq 111) \vee (b_{35} \neq 211) \vee (b_{35} \neq 411)] \wedge [(b_{41} \neq 221) \vee (b_{41} \neq 222) \vee (b_{41} \neq 422)] \wedge [(b_{42} \neq 121) \vee (b_{42} \neq 221) \vee (b_{42} \neq 222) \vee (b_{42} \neq 422)] \wedge [(b_{43} \neq 121) \vee (b_{43} \neq 221) \vee (b_{43} \neq 222) \vee (b_{43} \neq 422)] \wedge [(b_{44} \neq 121) \vee (b_{44} \neq 221) \vee (b_{44} \neq 222) \vee (b_{44} \neq 422)] \wedge [(b_{45} \neq 121) \vee (b_{45} \neq 221) \vee (b_{45} \neq 222) \vee (b_{45} \neq 422)]$$

π-conveyor

Blue π-container (size 211) which moves eastbound

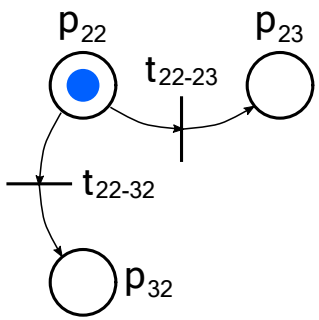


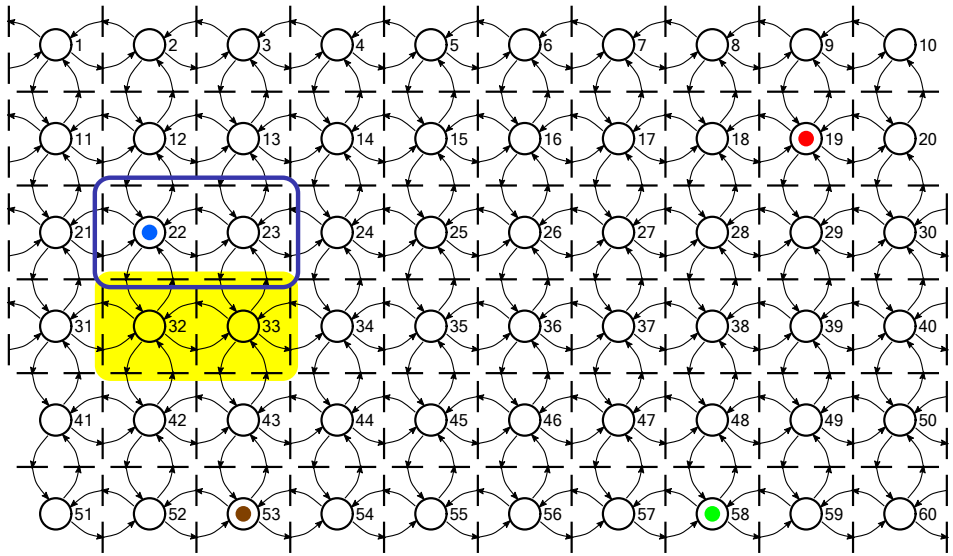


$$C_{22-23-211} = [(b_{24} \neq 111) \vee (b_{24} \neq 121) \vee (b_{24} \neq 211) \vee (b_{24} \neq 221) \vee (b_{24} \neq 411) \vee (b_{24} \neq 222) \vee (b_{24} \neq 422)] \wedge [(b_{34} \neq 121) \vee (b_{34} \neq 221) \vee (b_{34} \neq 141) \vee (b_{34} \neq 222) \vee (b_{34} \neq 422) \vee (b_{34} \neq 242)] \wedge [(b_{44} \neq 141) \vee (b_{44} \neq 242)] \wedge [(b_{54} \neq 141) \vee (b_{54} \neq 242)]$$

π-conveyor

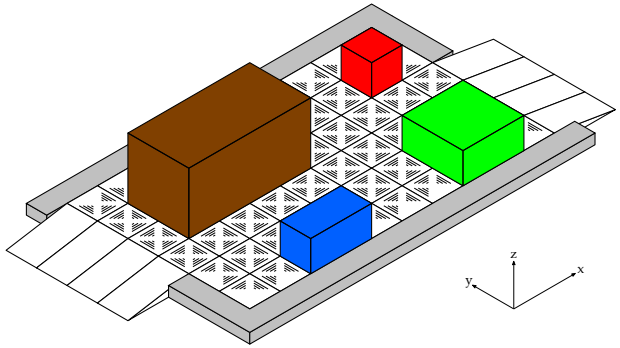
Blue π-container (size 211) which moves southbound

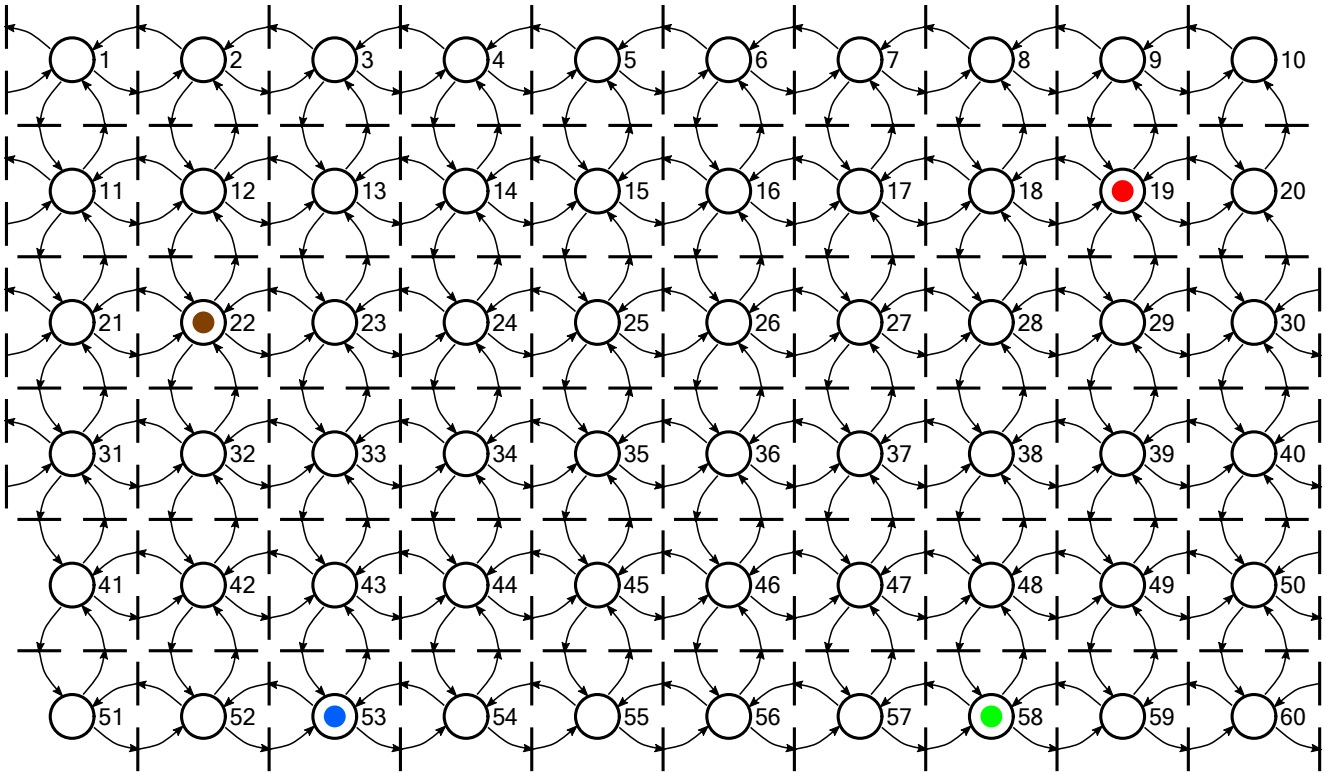




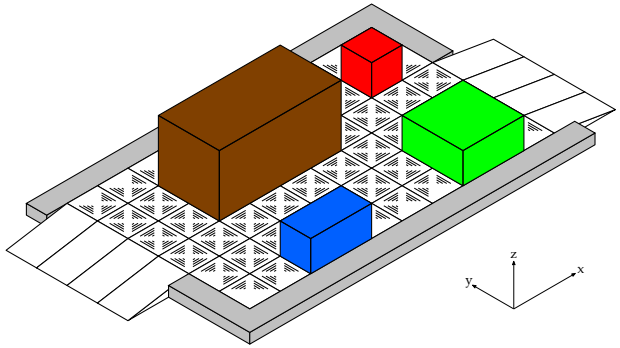
$$C_{22-32-211} = [(b_{31} \neq 211) \vee (b_{31} \neq 411)] \wedge [(b_{32} \neq 111) \vee (b_{32} \neq 211) \vee (b_{32} \neq 411)] \wedge [(b_{33} \neq 111) \vee (b_{33} \neq 211) \vee (b_{33} \neq 411)] \wedge [(b_{41} \neq 221) \vee (b_{41} \neq 222) \vee (b_{41} \neq 422)] \wedge [(b_{42} \neq 121) \vee (b_{42} \neq 221) \vee (b_{42} \neq 222) \vee (b_{42} \neq 422)] \wedge [(b_{43} \neq 121) \vee (b_{43} \neq 221) \vee (b_{43} \neq 222) \vee (b_{43} \neq 422)]$$

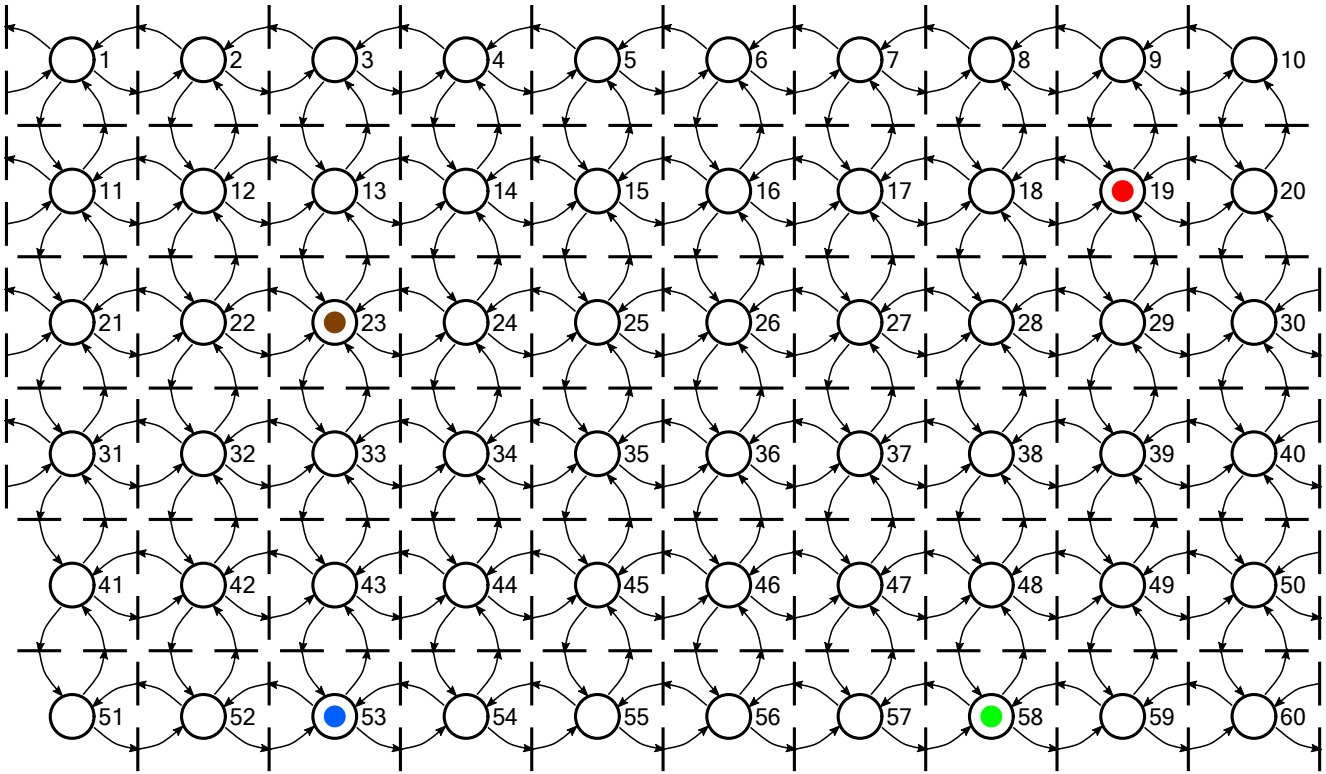
π-conveyor


z
y
x

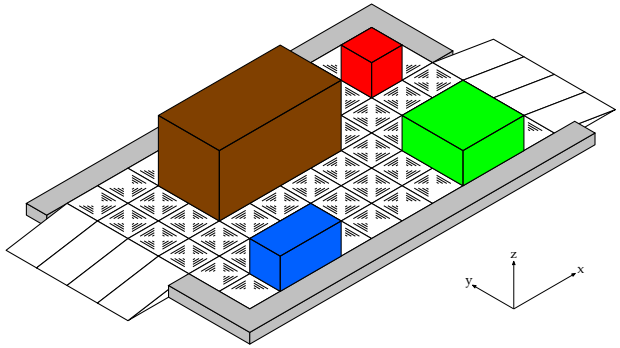

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

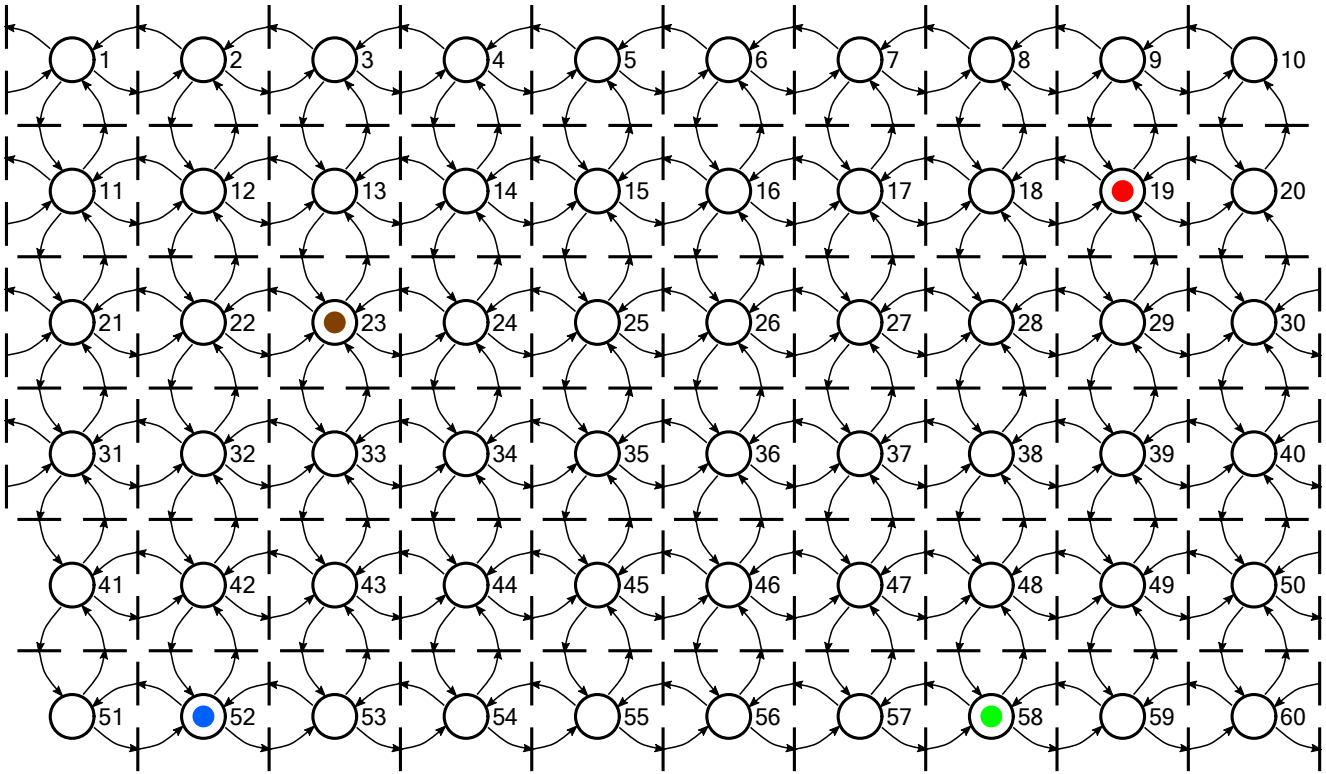
π-conveyor


z
y
x

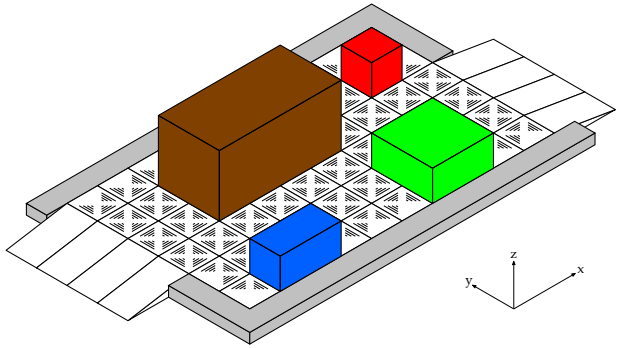

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

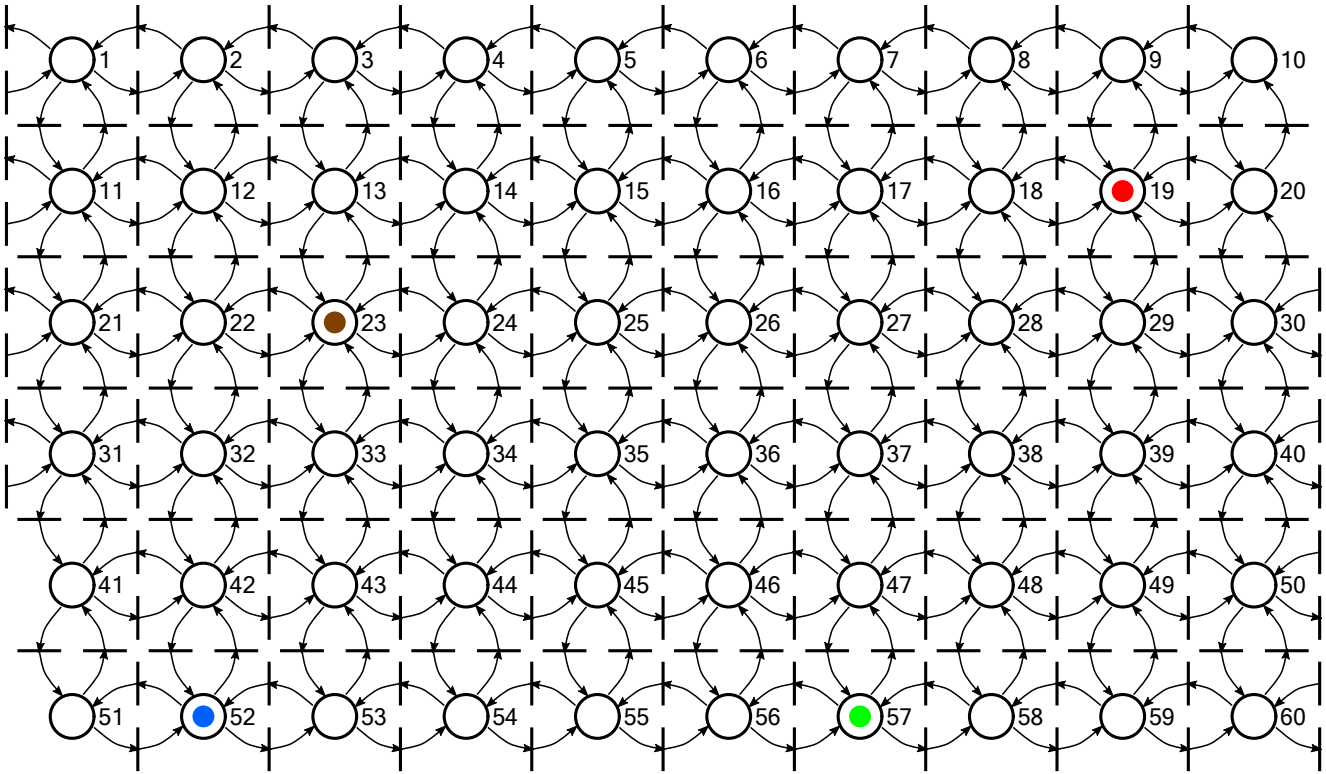
π-conveyor



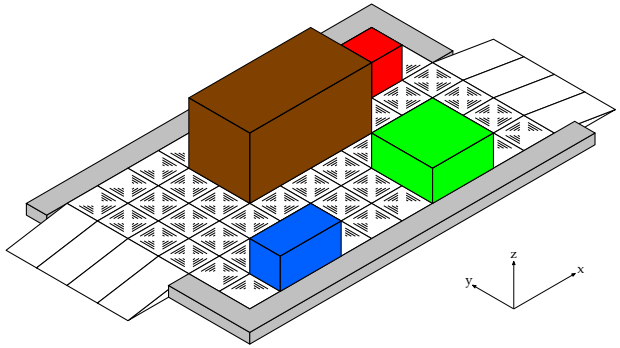

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

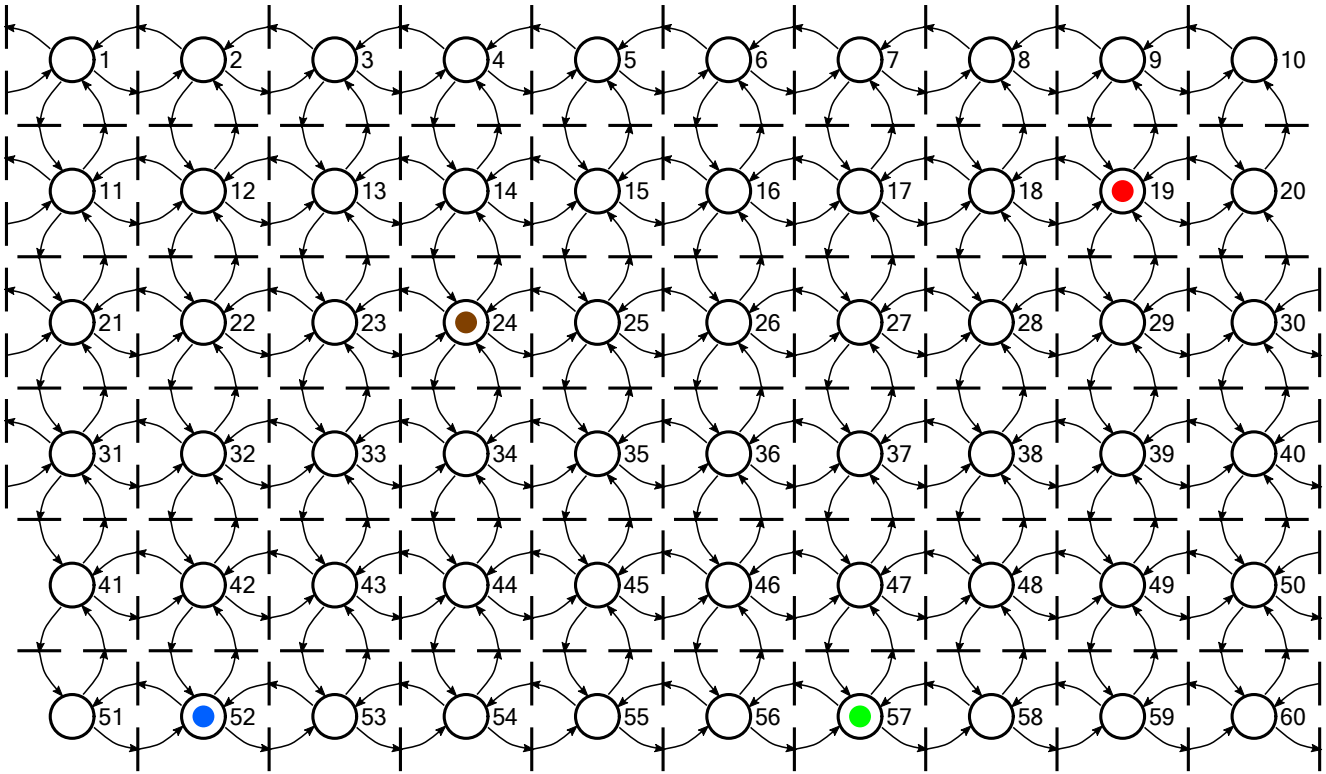
π-conveyor


z
y
x

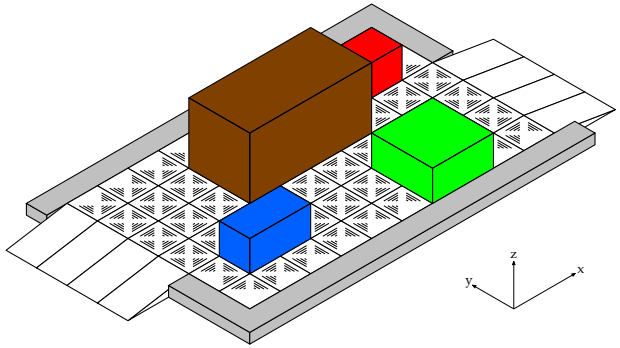

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

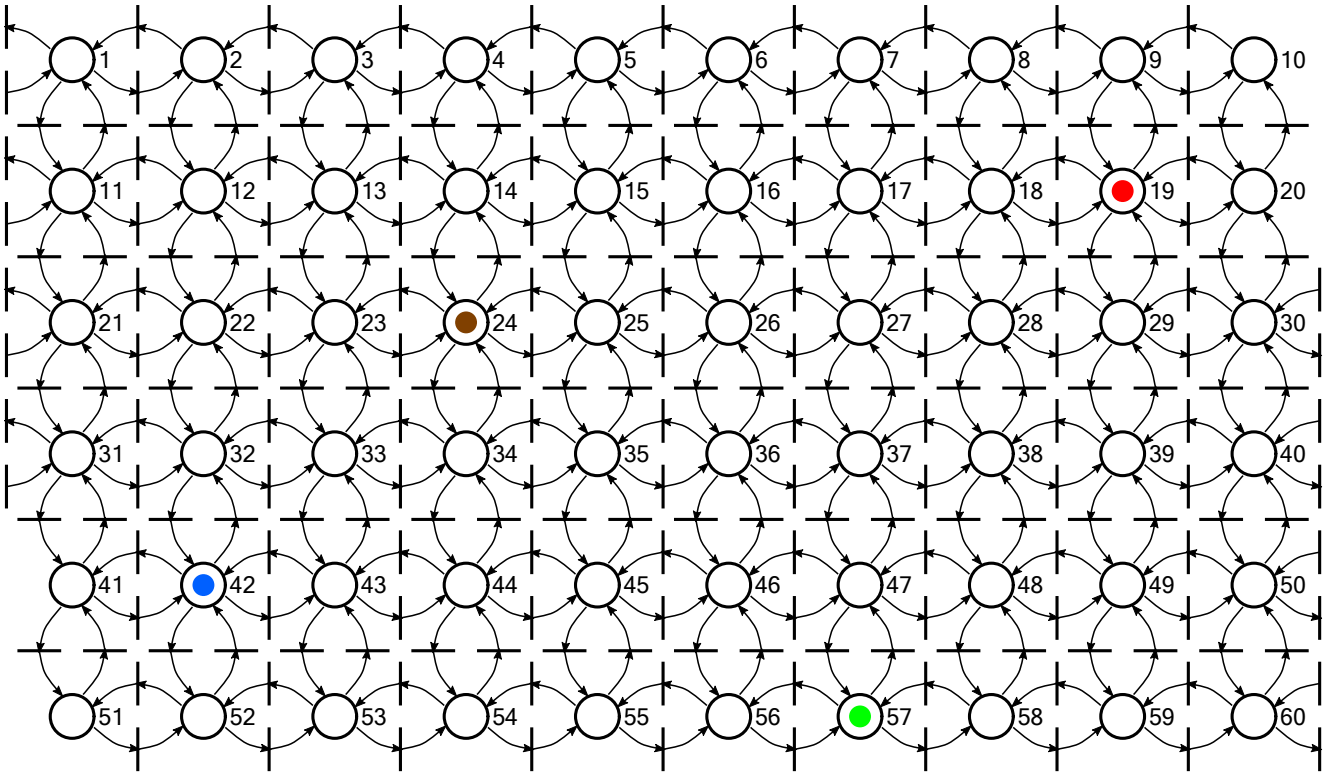
π-conveyor


z
y
x

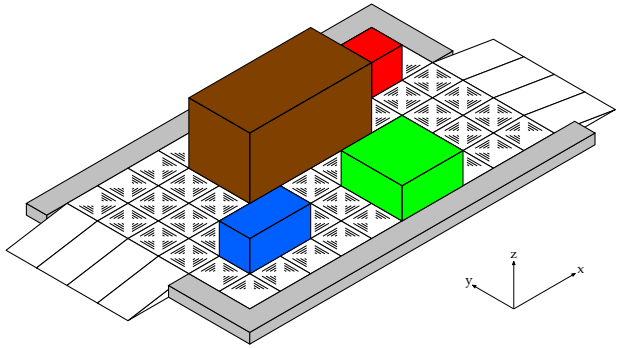

1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60

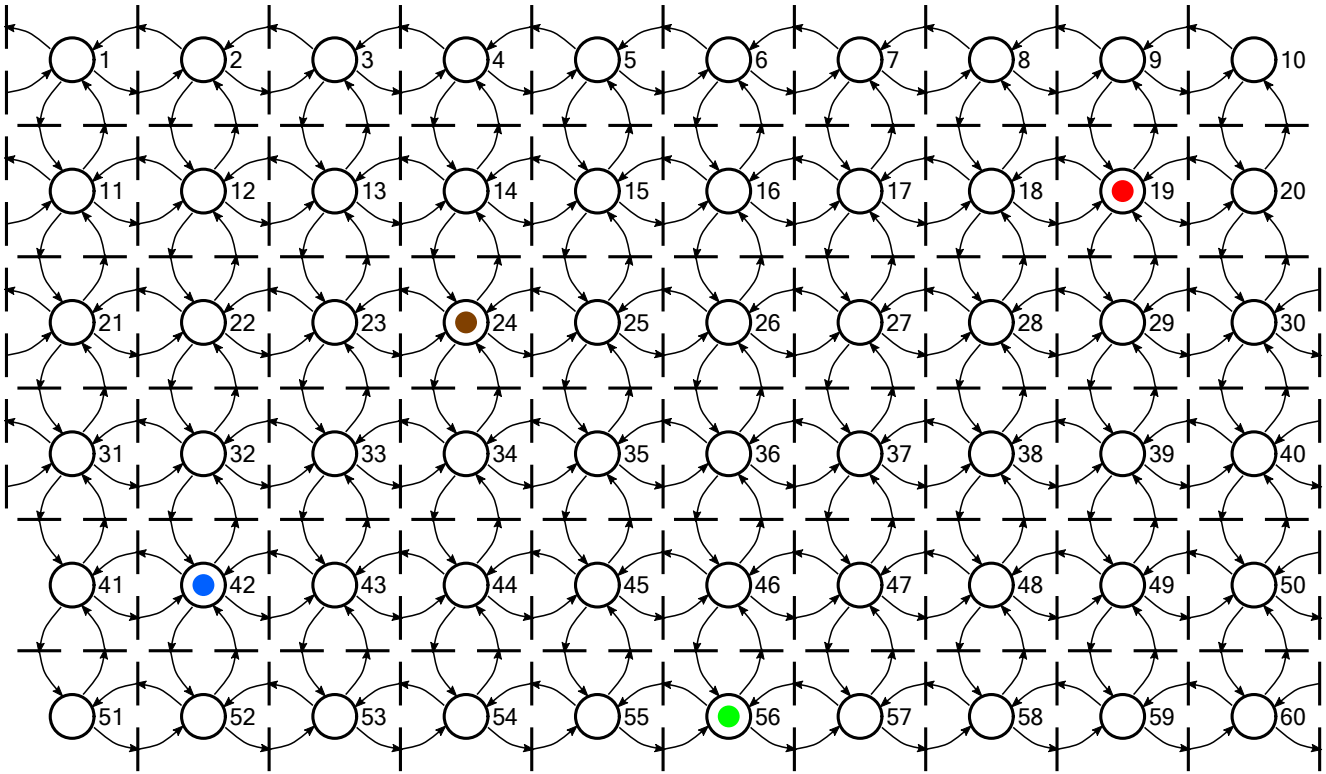
π-conveyor


z
y
x

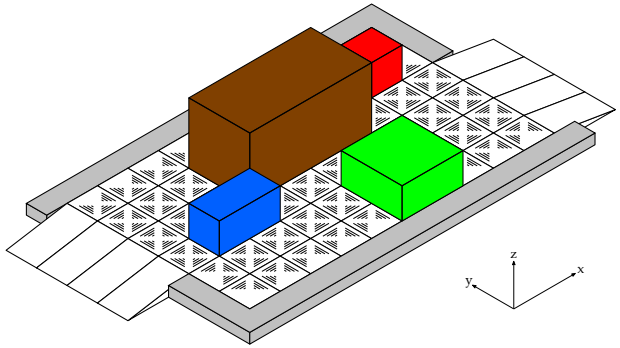

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

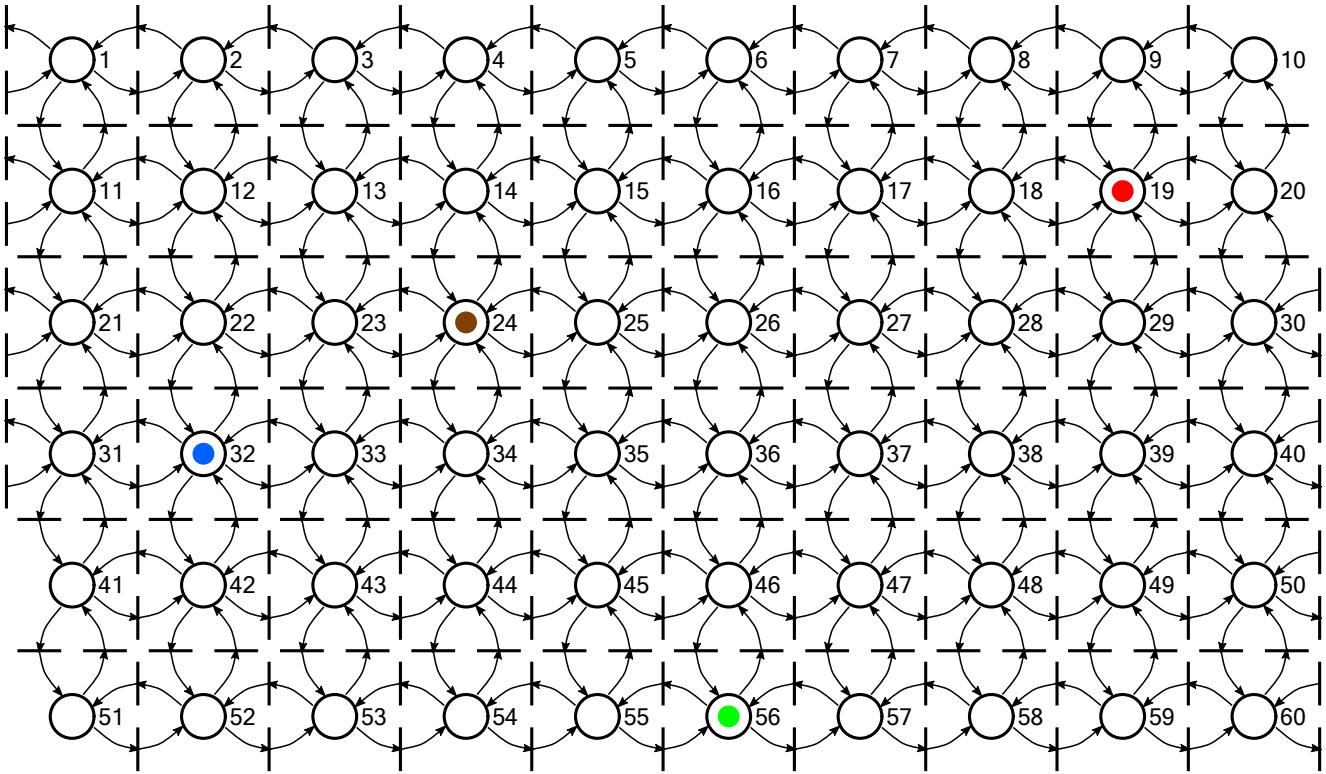
π-conveyor


z
y
x

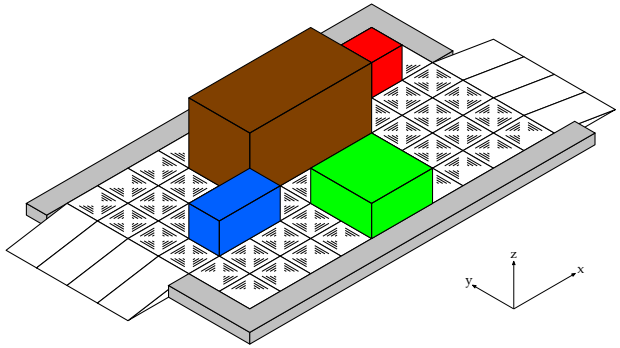

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

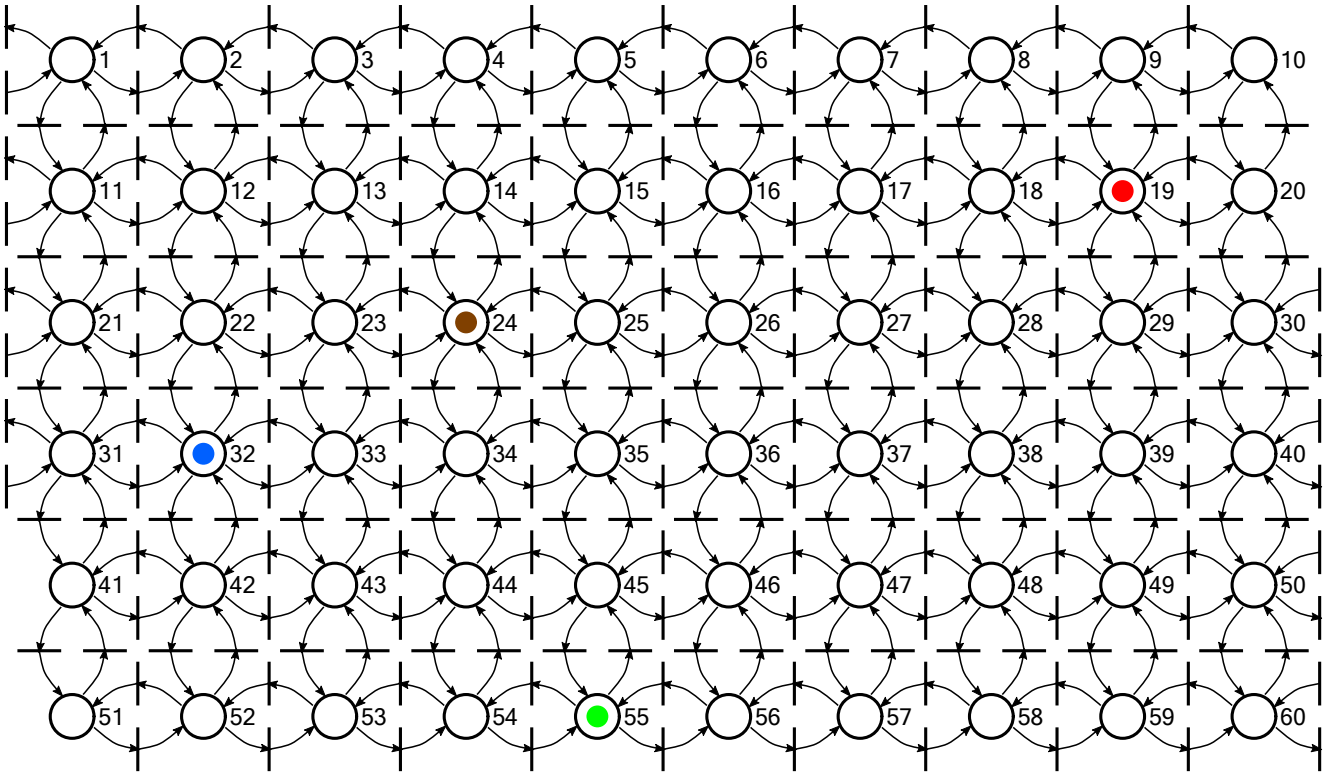
π-conveyor


z
y
x

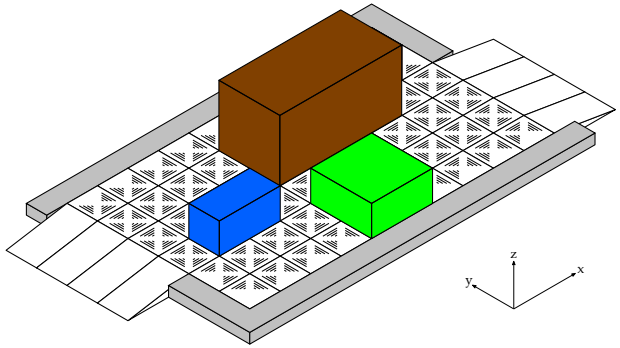

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

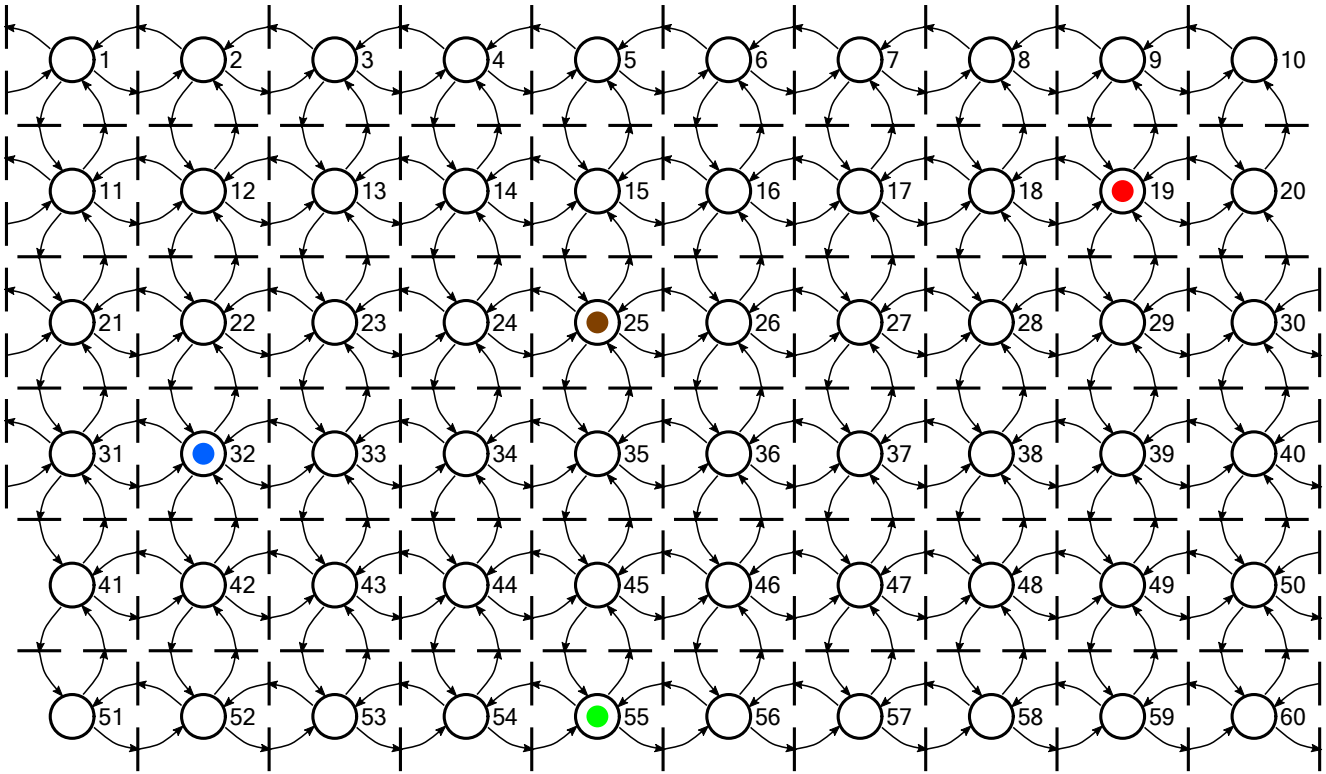
π-conveyor


z
y
x

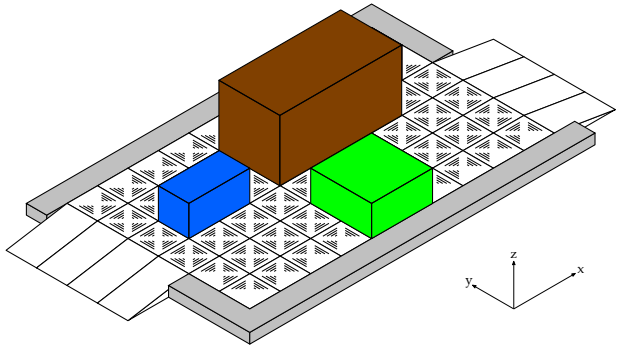

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

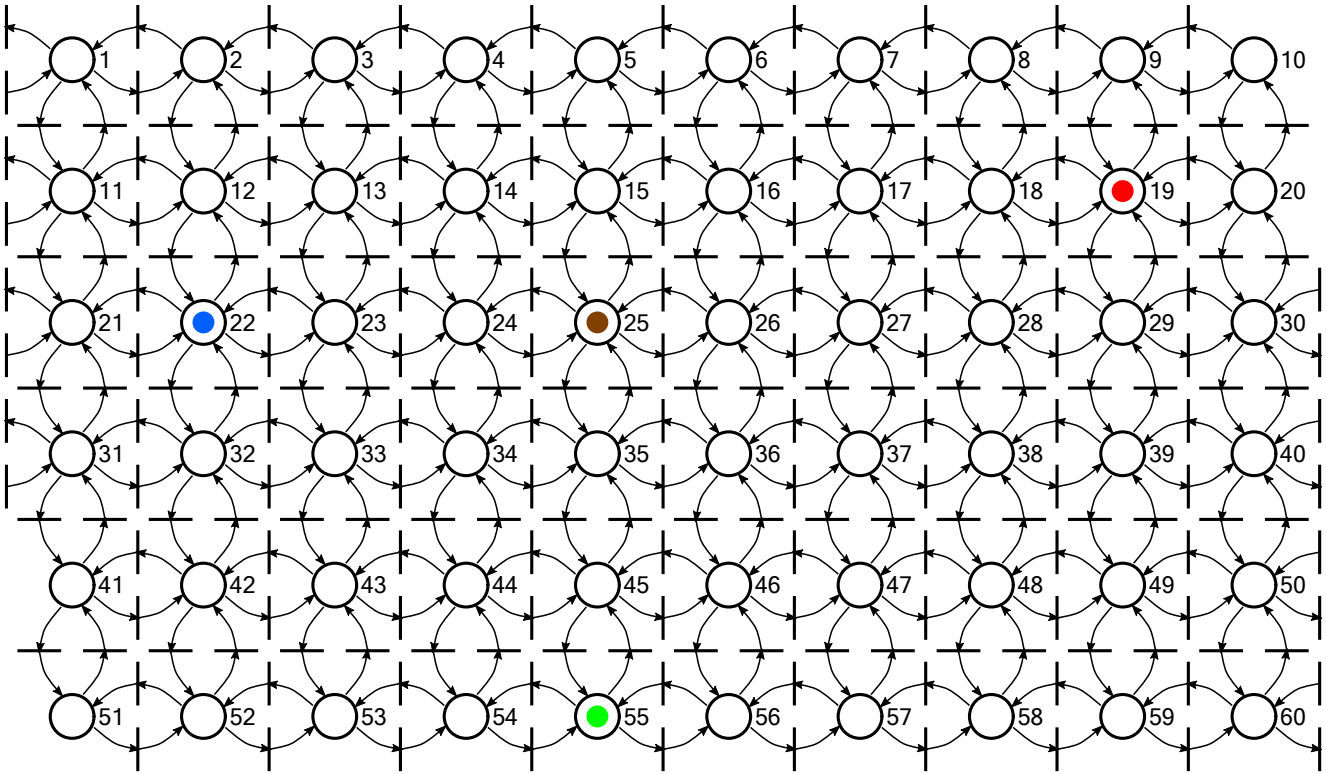
π-conveyor


z
y
x

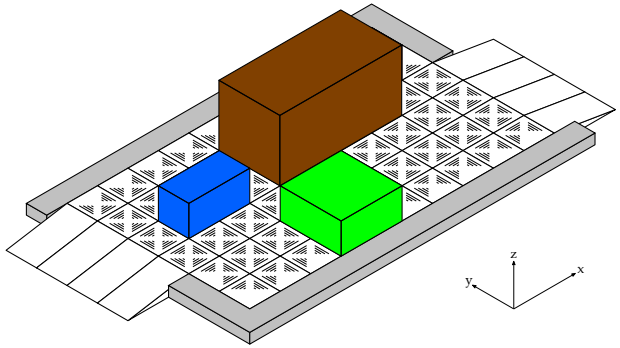

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

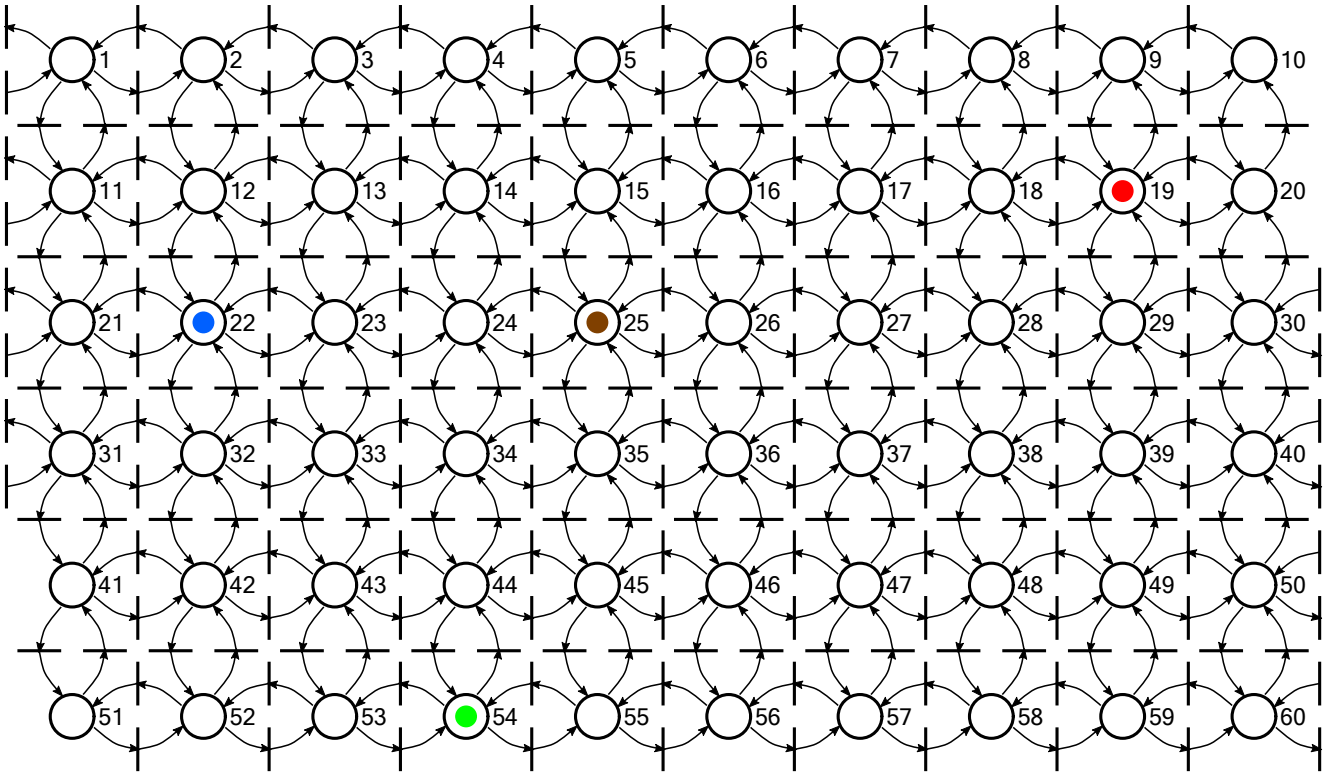
π-conveyor



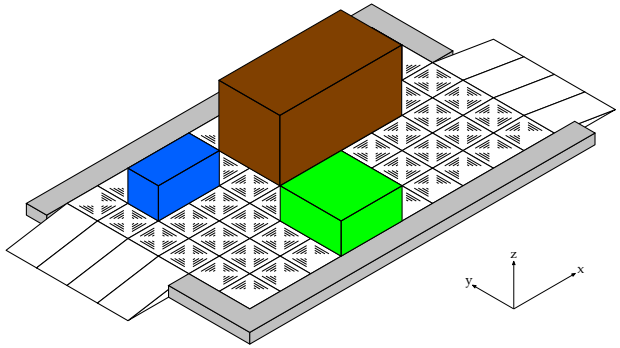

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

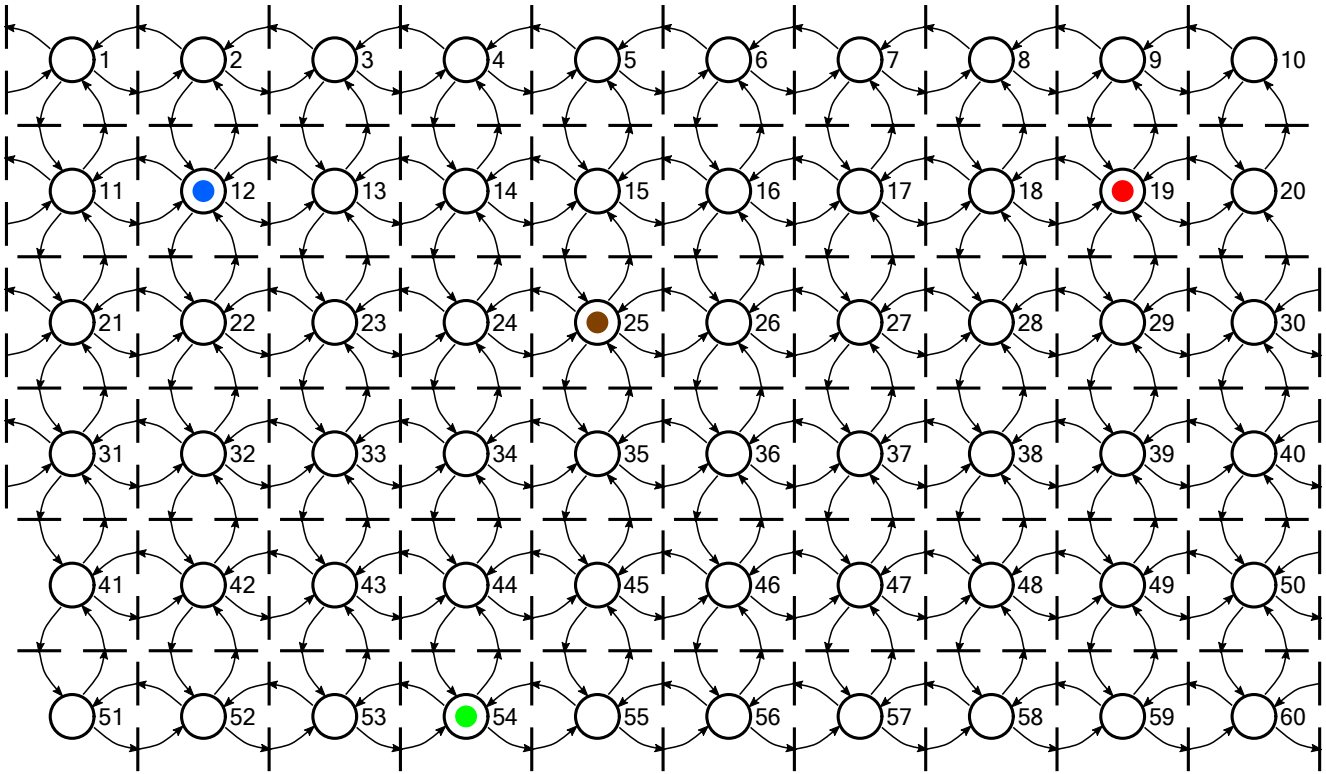
π-conveyor


z
y
x

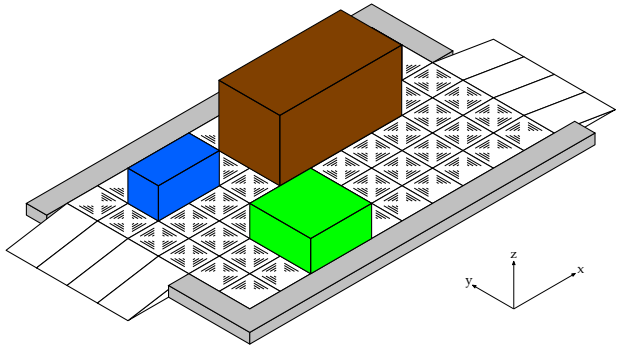

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

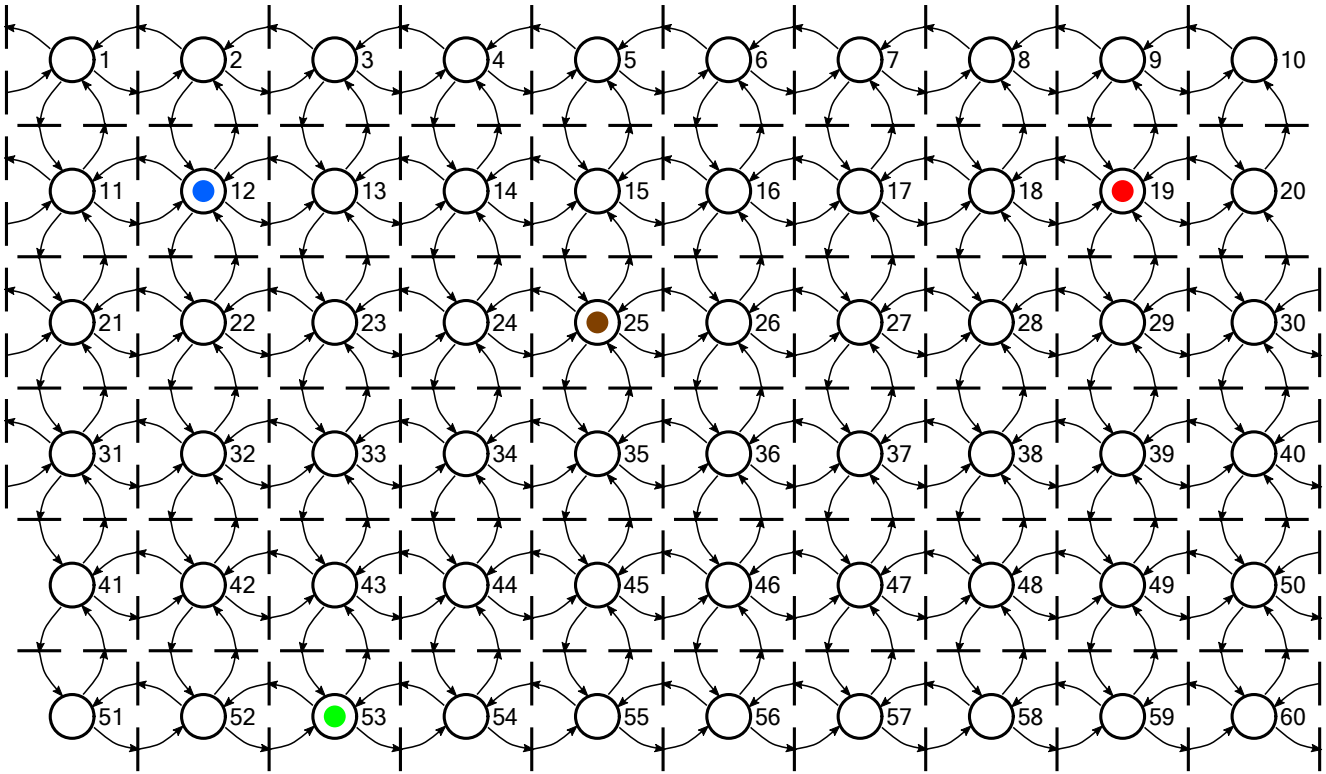
π-conveyor


z
y
x

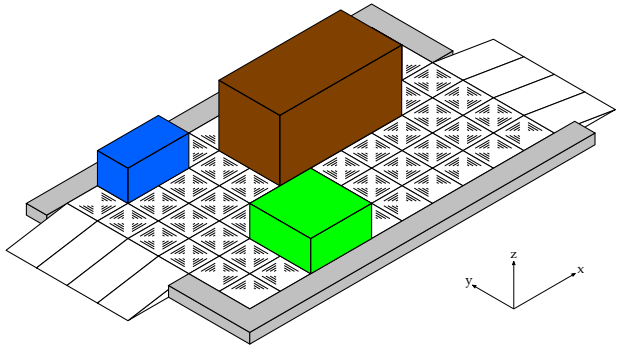

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

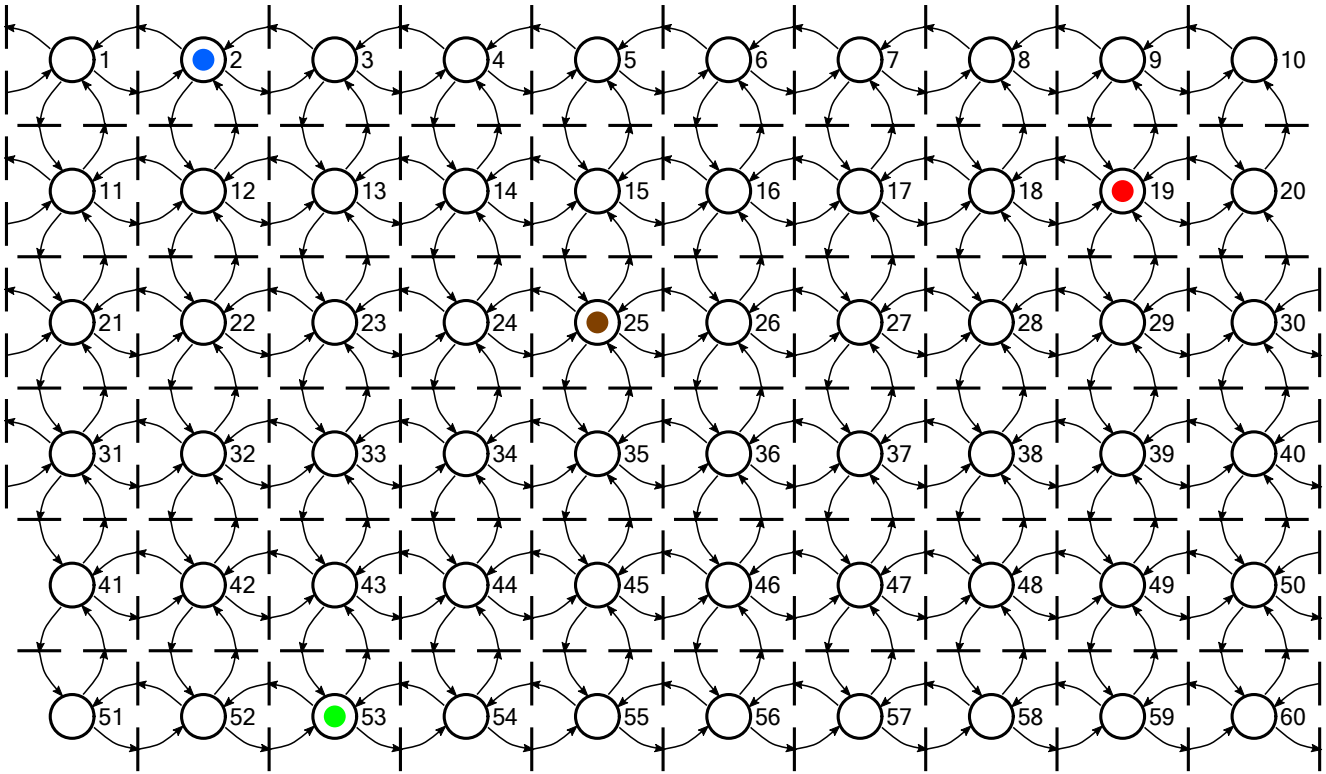
π-conveyor



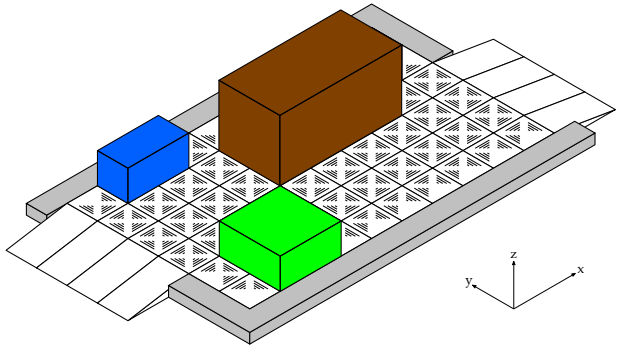

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

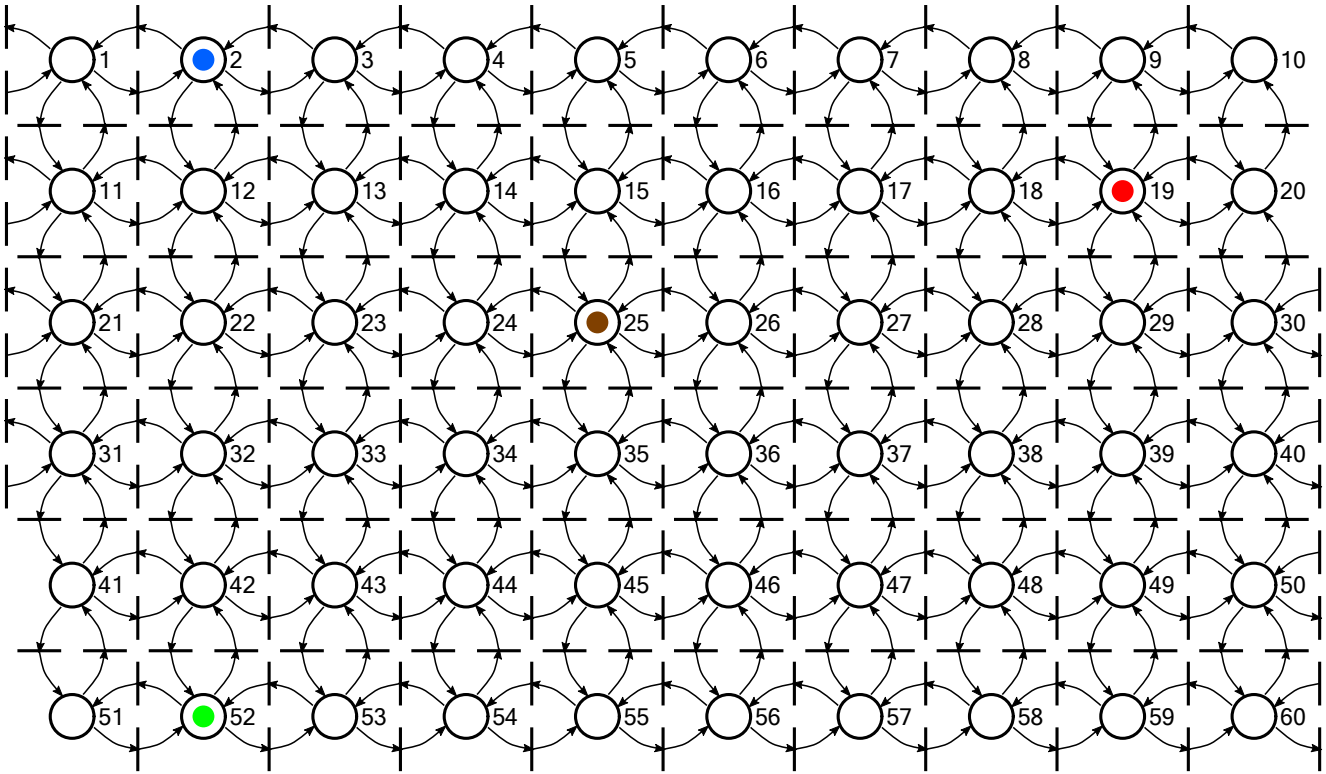
π-conveyor



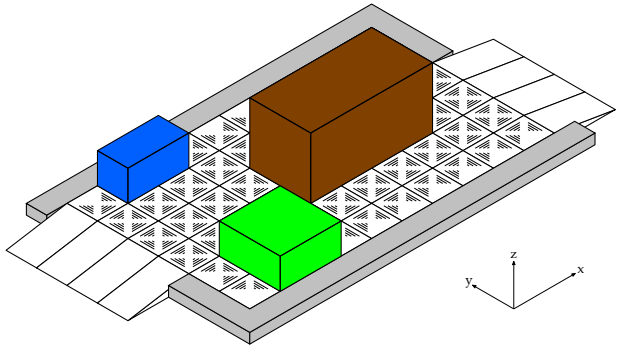

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

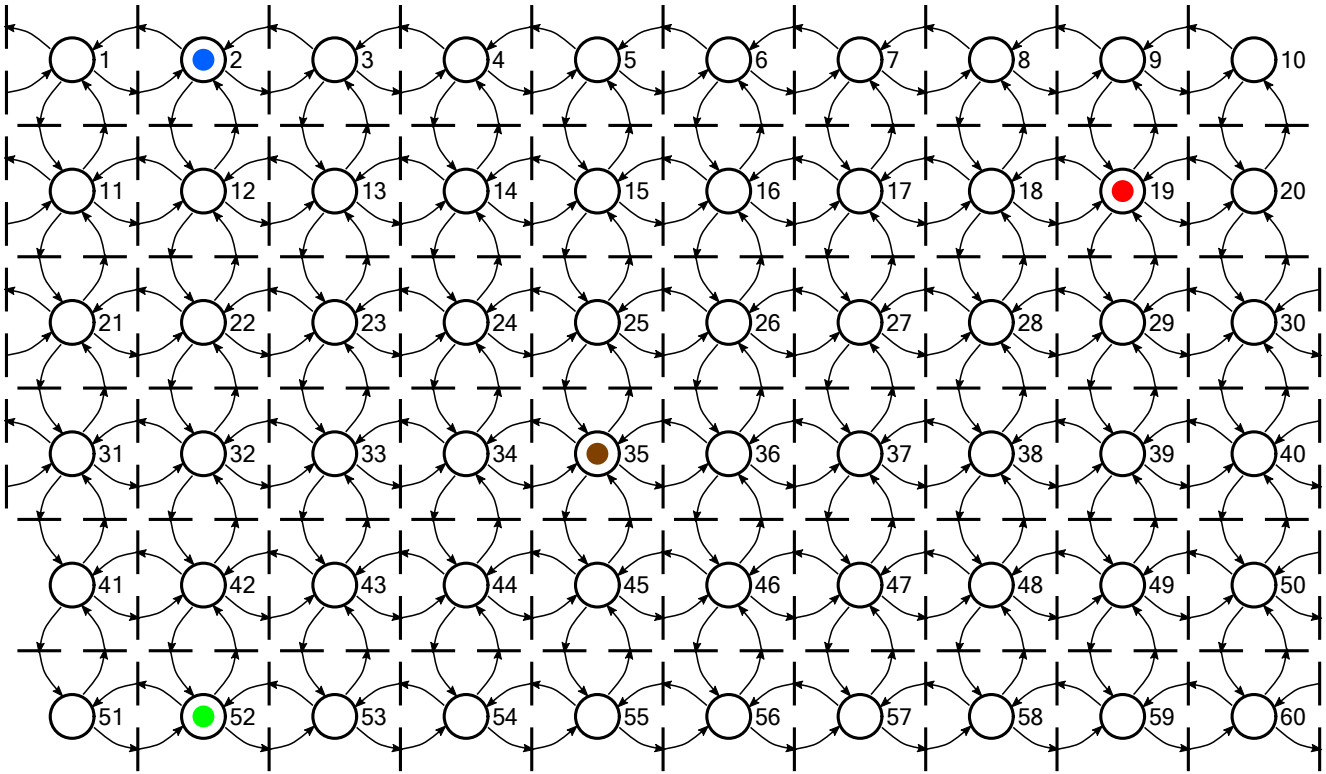
π-conveyor



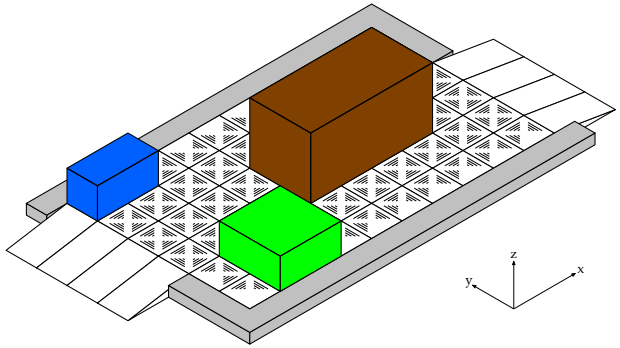

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

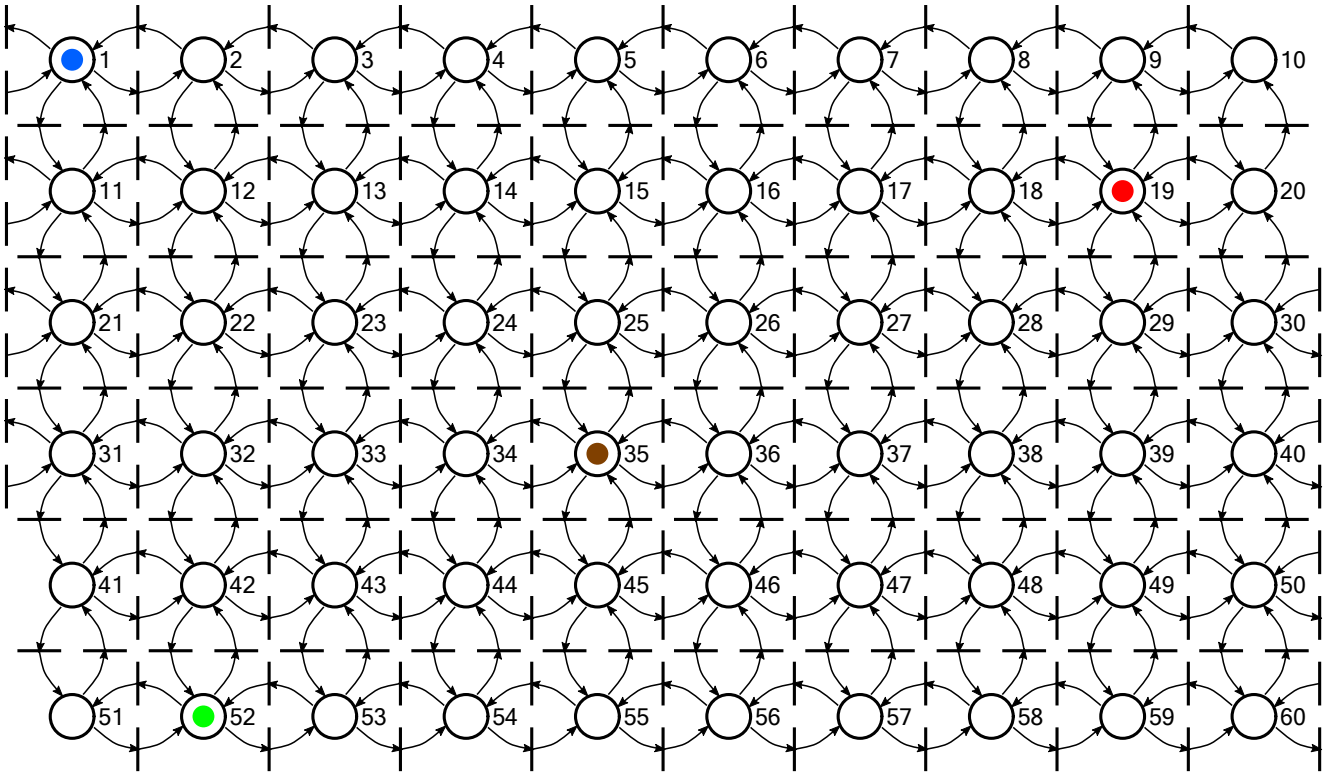
π-conveyor


z
y
x

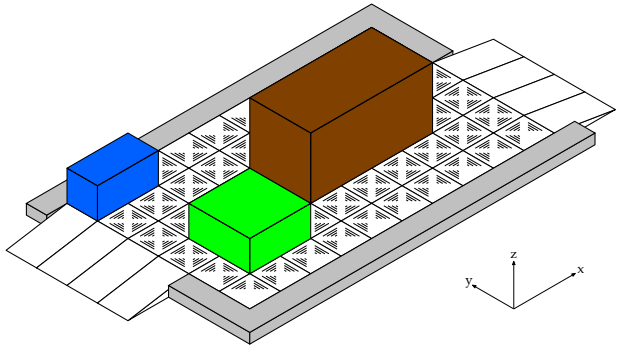

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

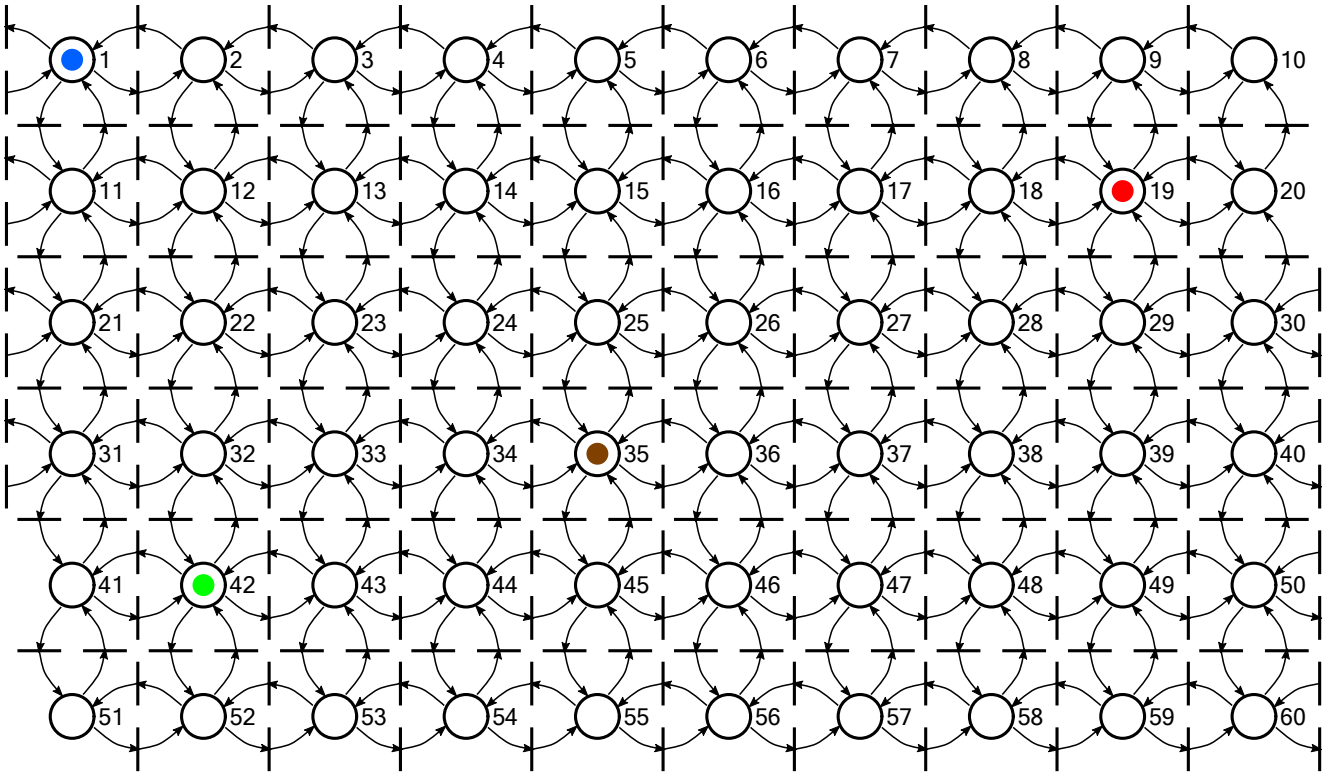
π-conveyor



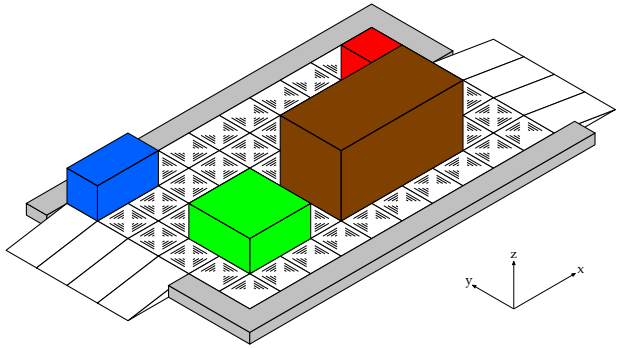

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

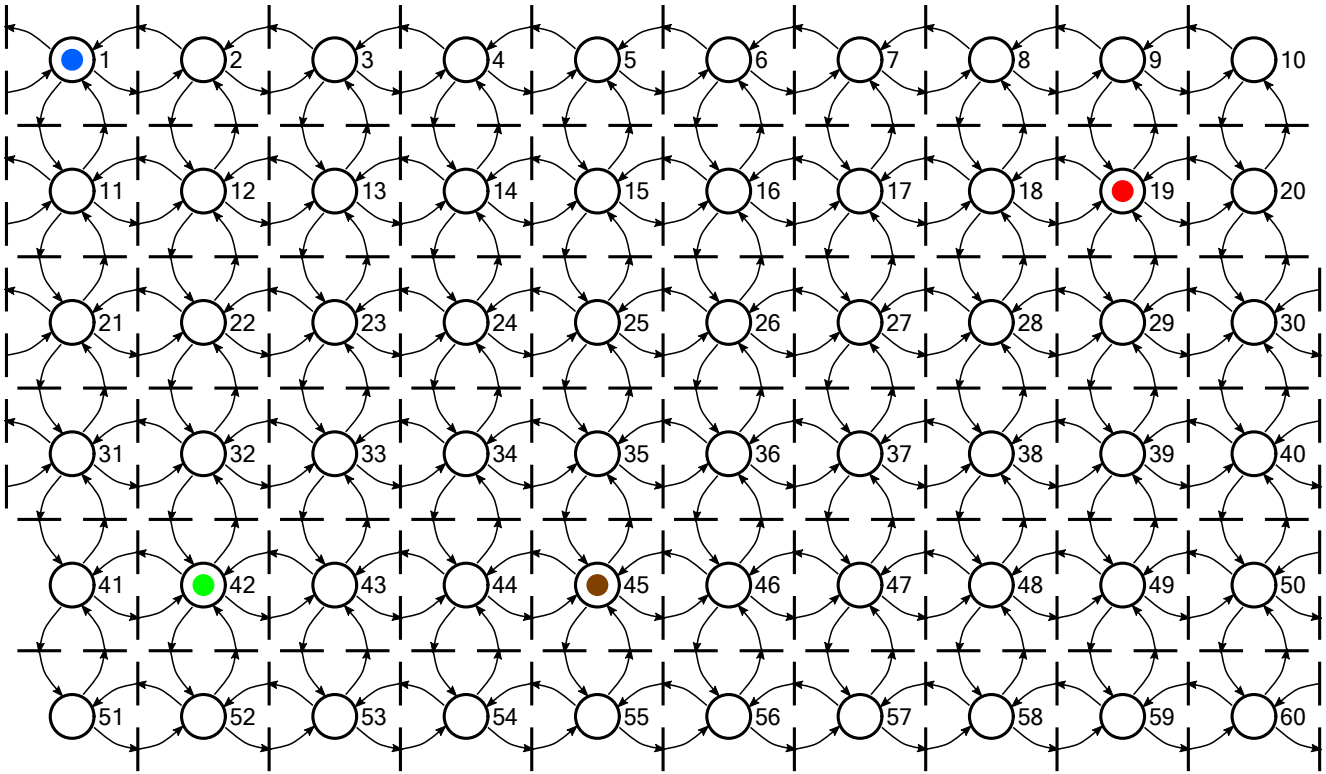
π-conveyor


z
y
x

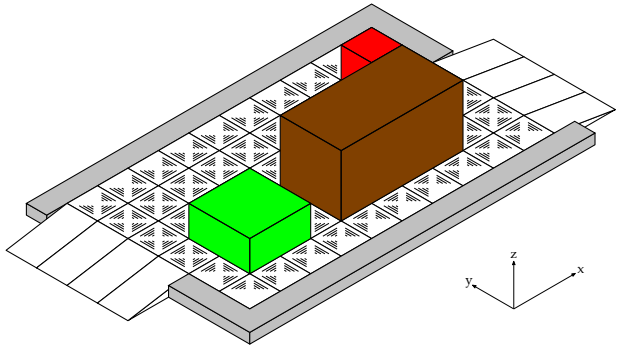

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

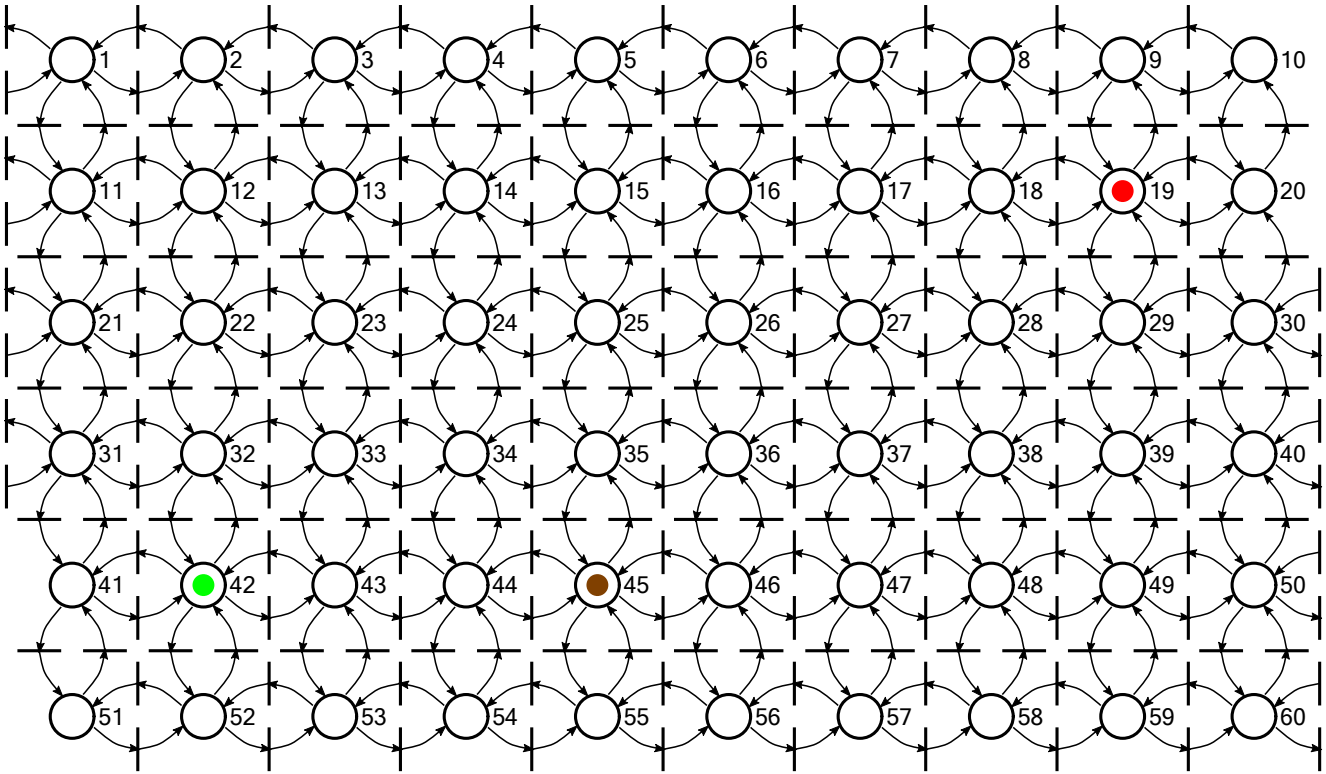
π-conveyor


z
y
x

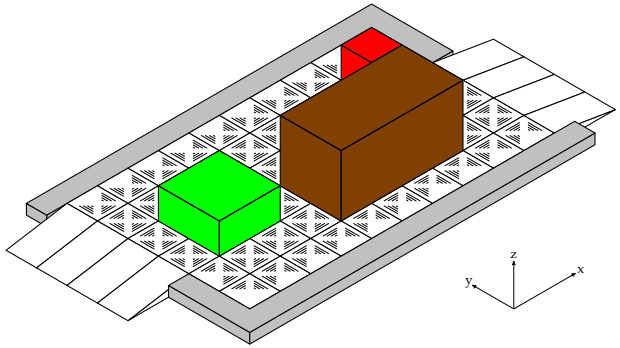

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

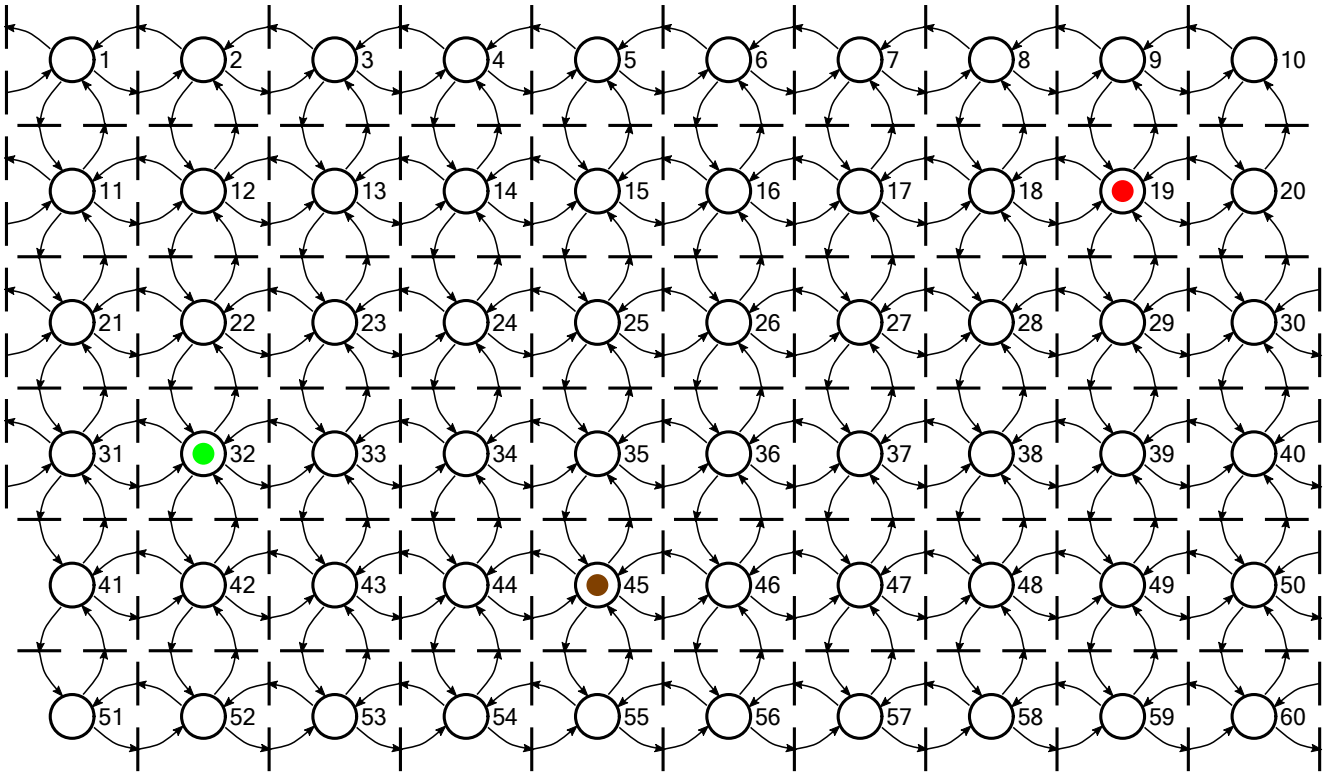
π-conveyor


z
y
x

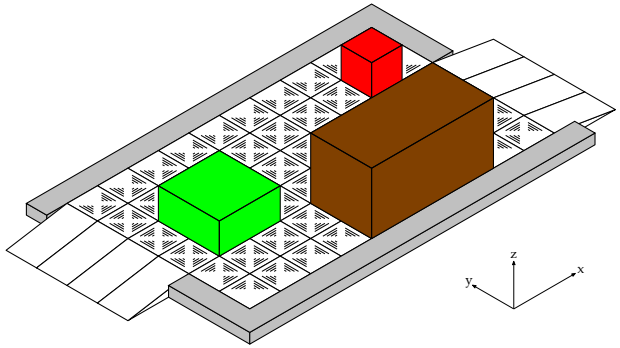

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

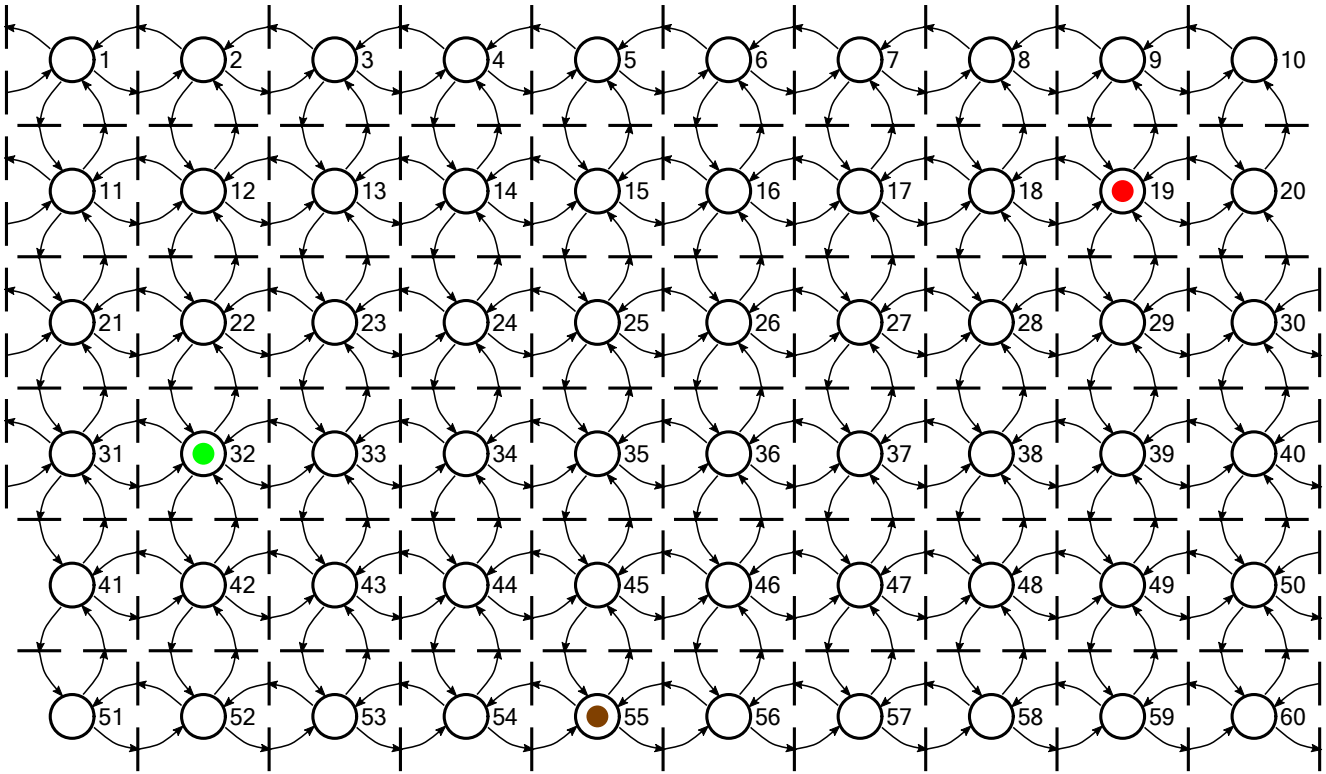
π-conveyor


z
y
x

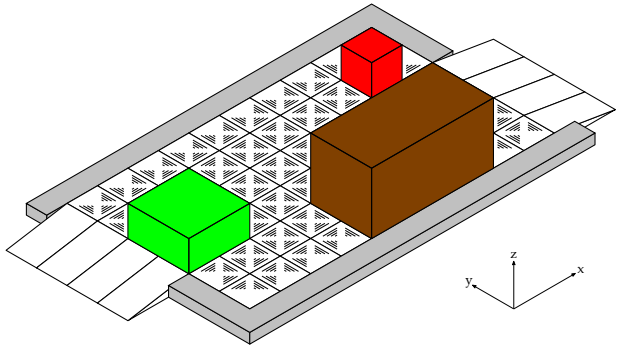

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

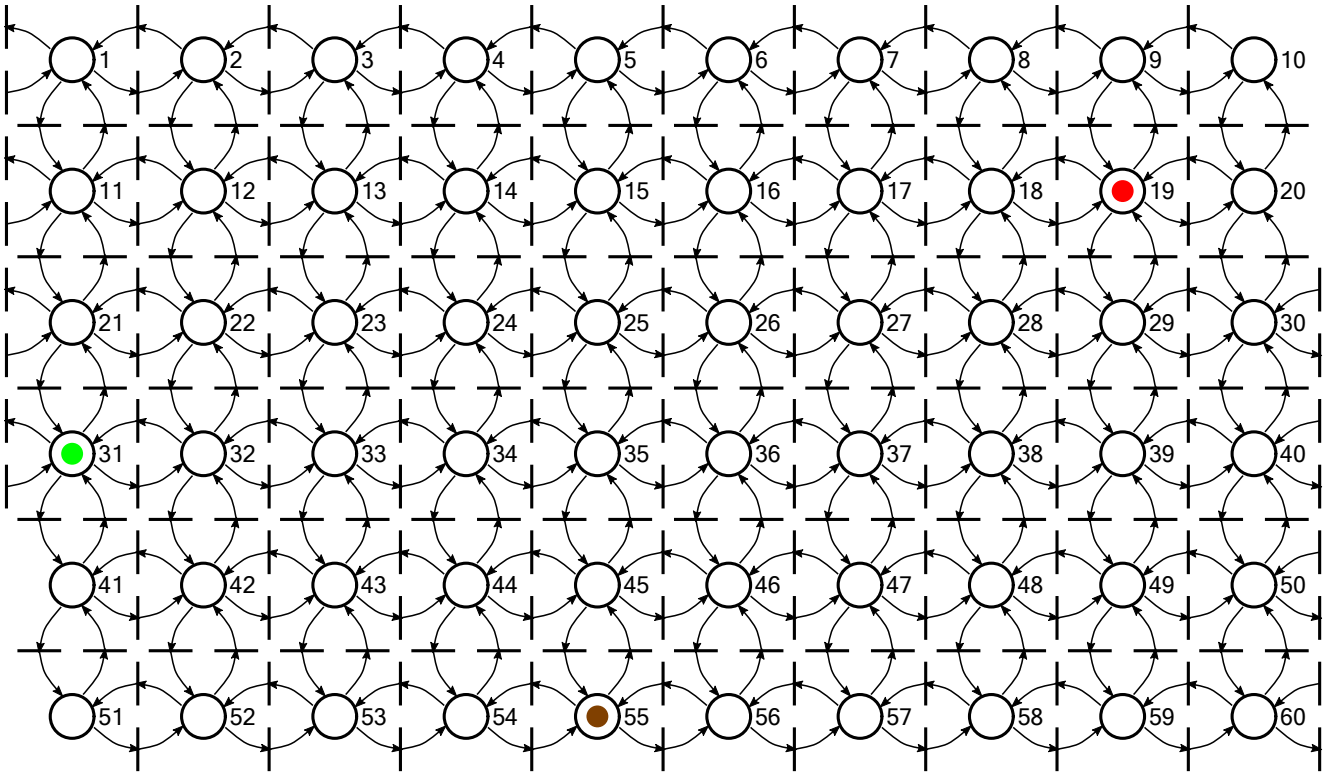
π-conveyor


z
y
x

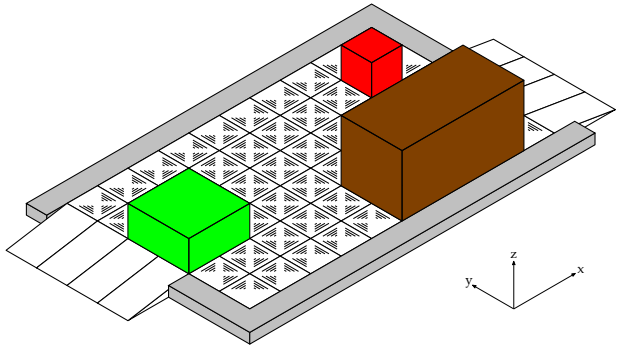

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

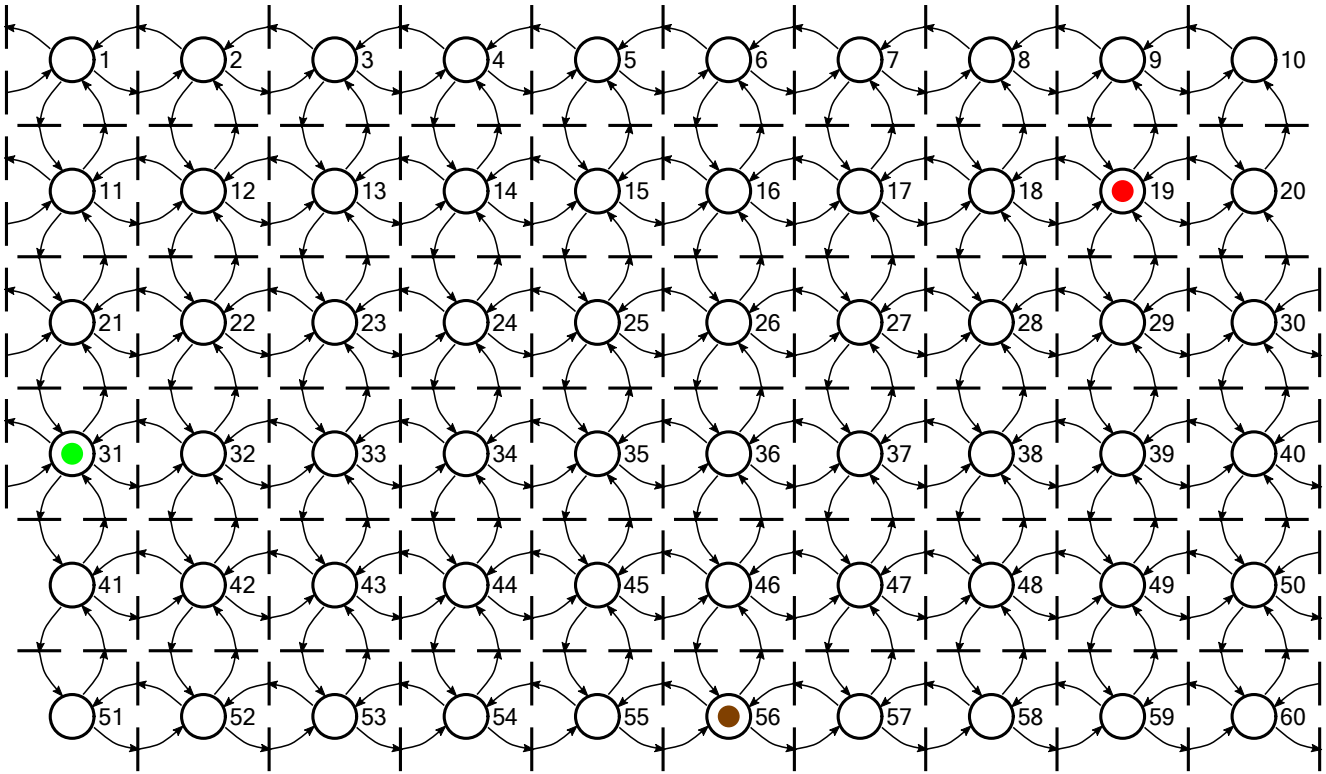
π-conveyor


z
y
x

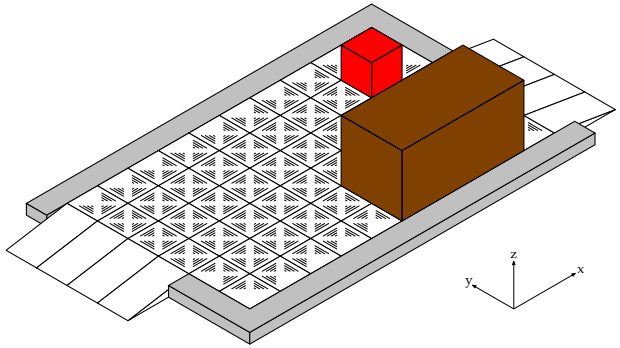

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

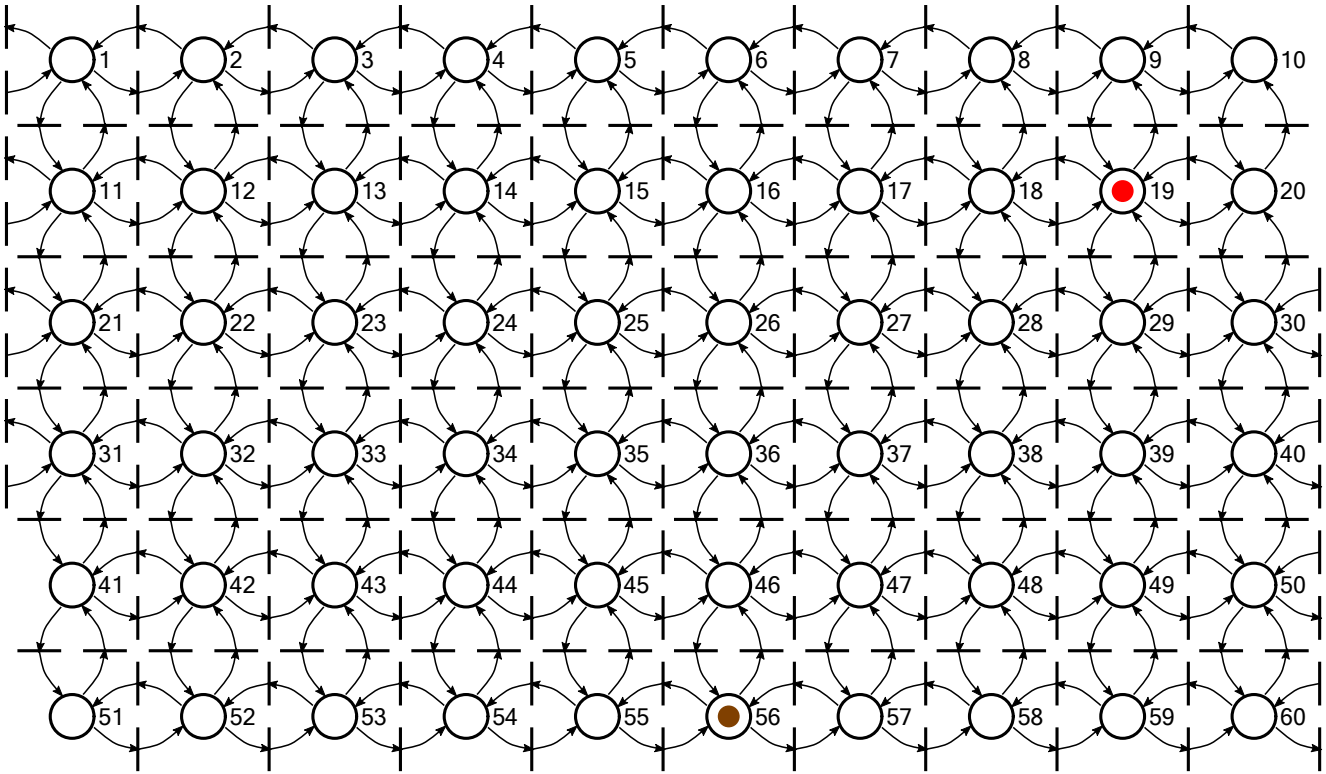
π-conveyor


z
y
x

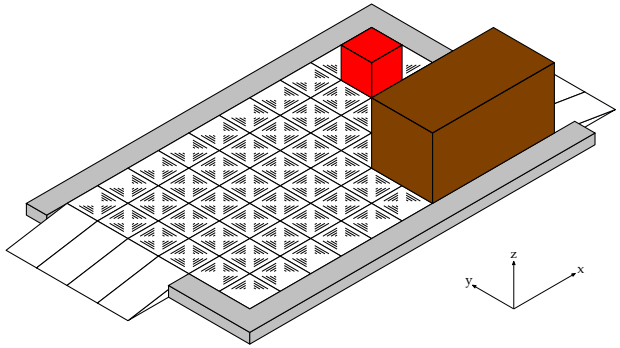

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

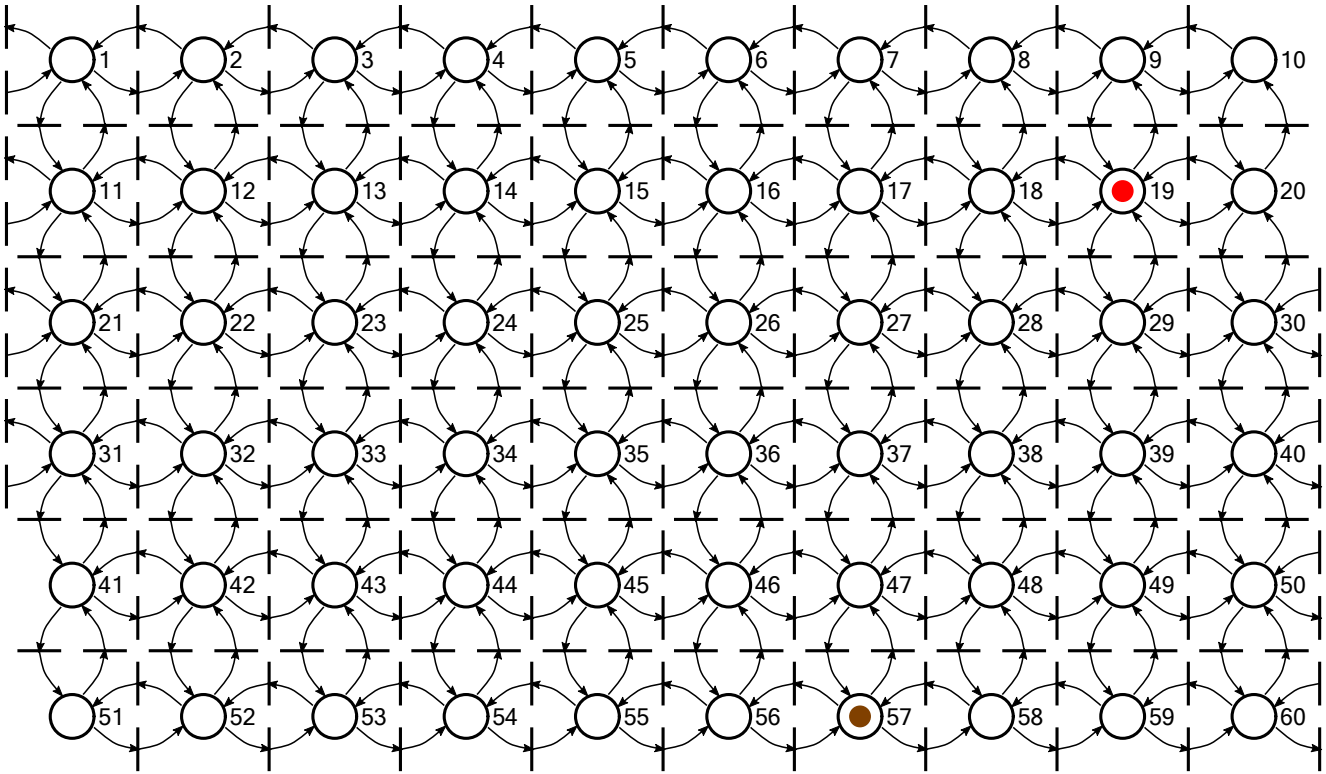
π-conveyor


z
y
x

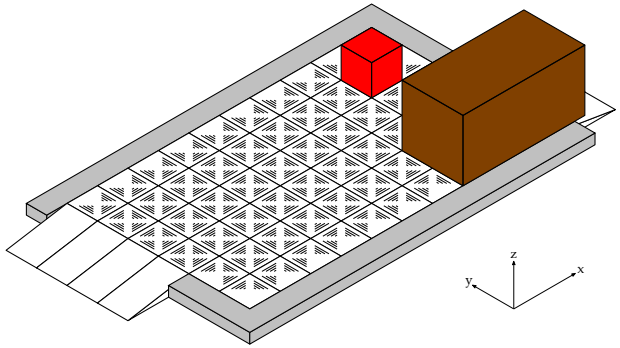

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

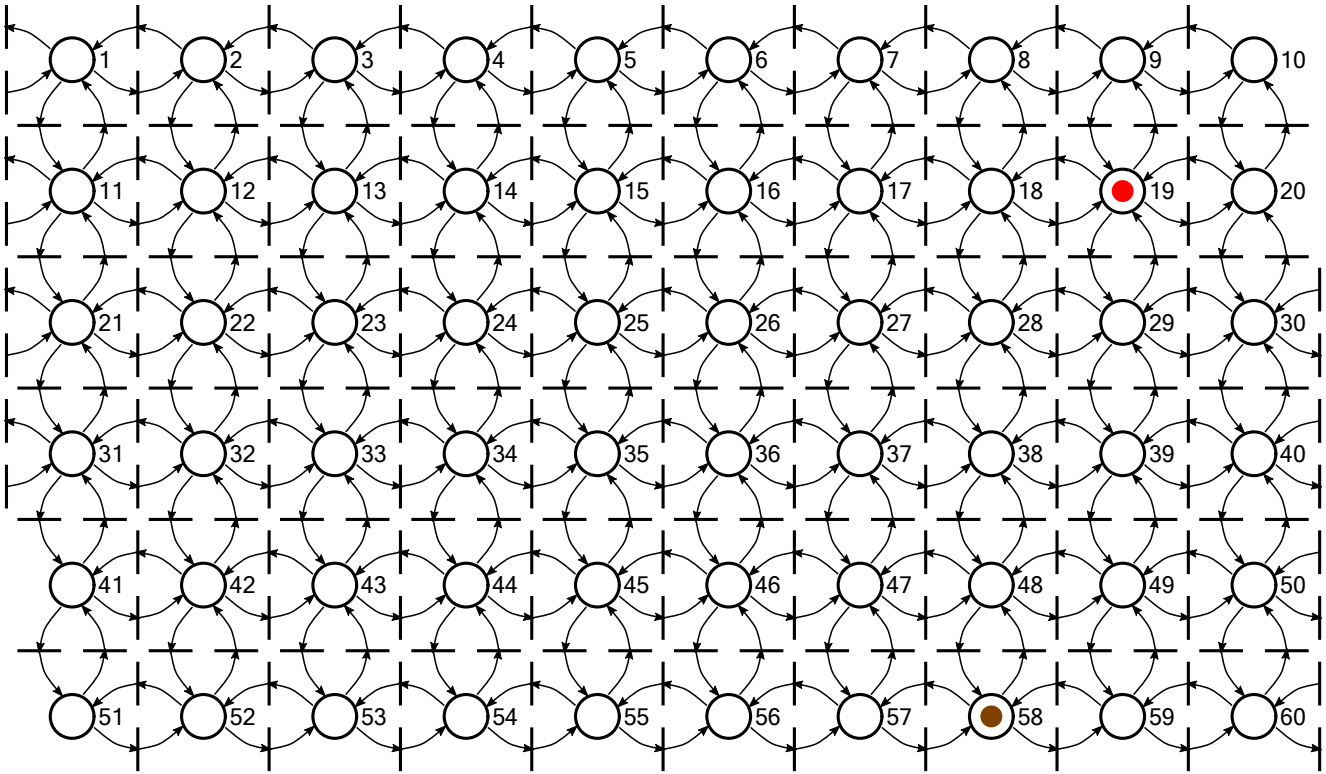
π-conveyor


z
y
x

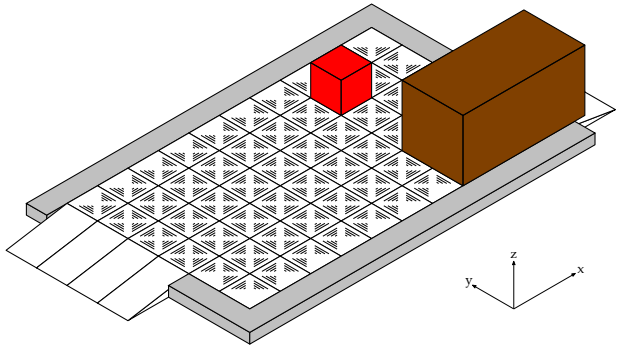

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

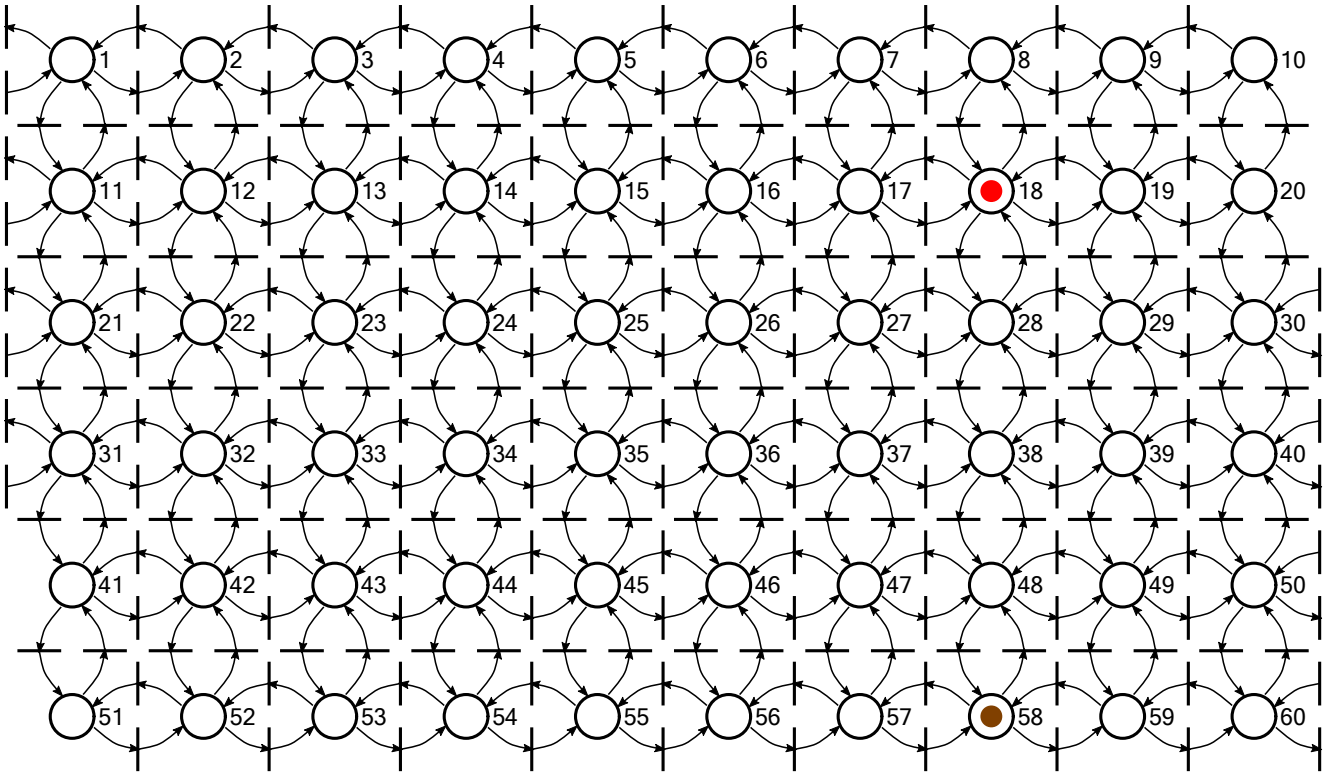
π-conveyor


z
y
x

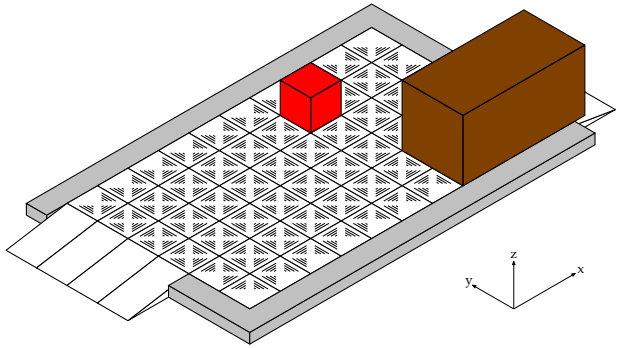

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

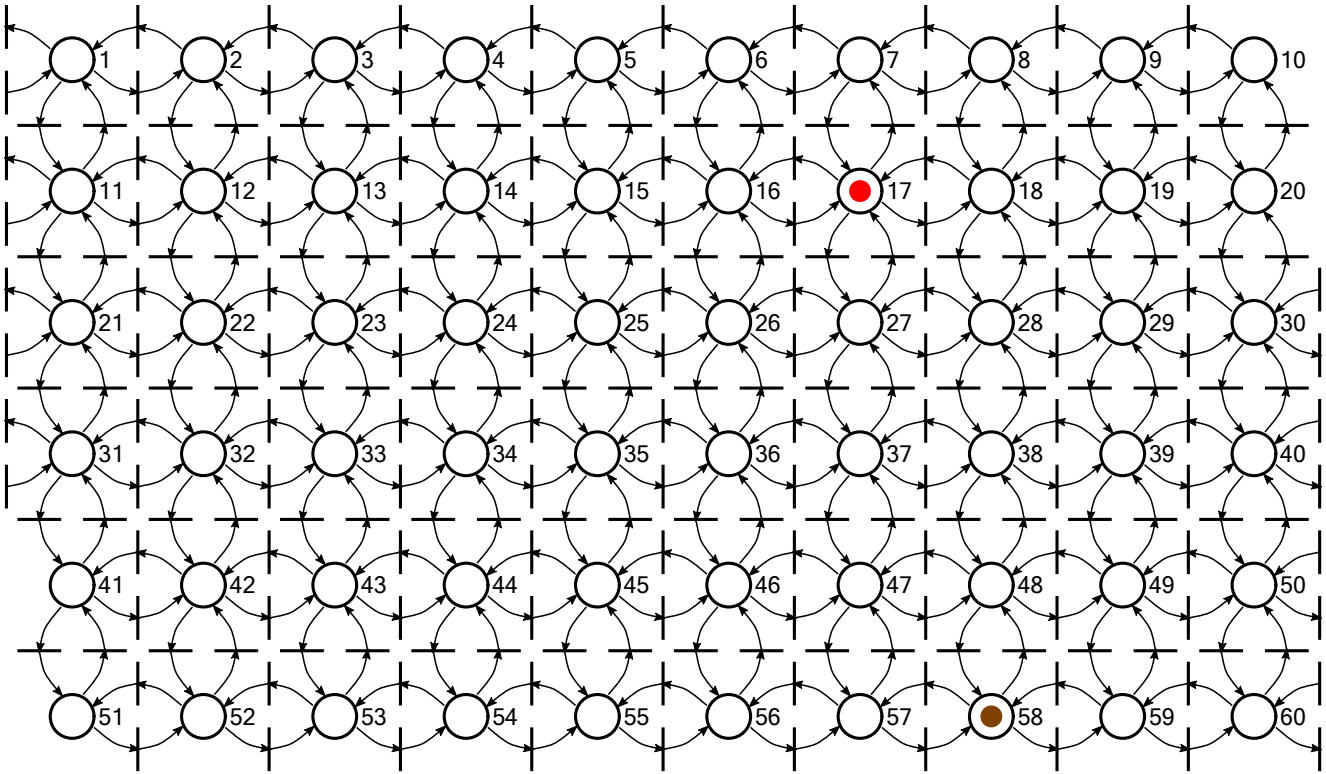
π-conveyor


z
y
x

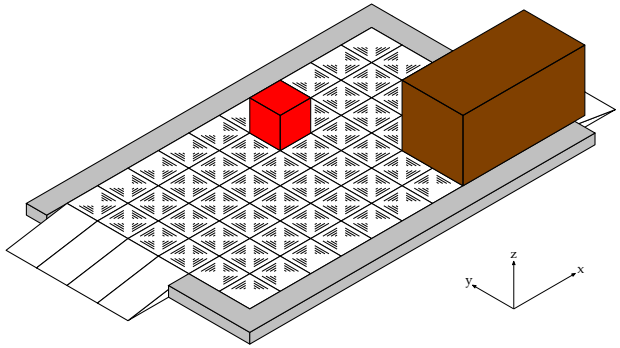

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

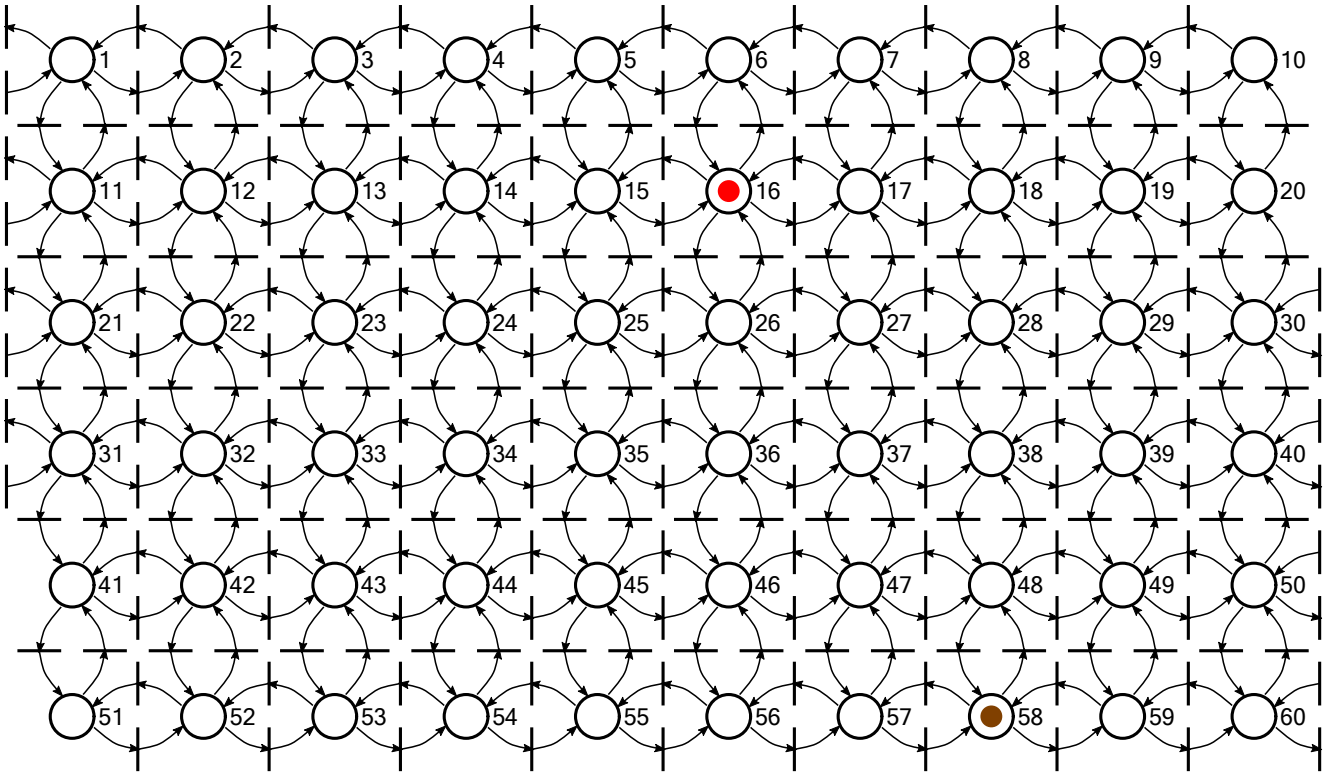
π-conveyor


z
y
x

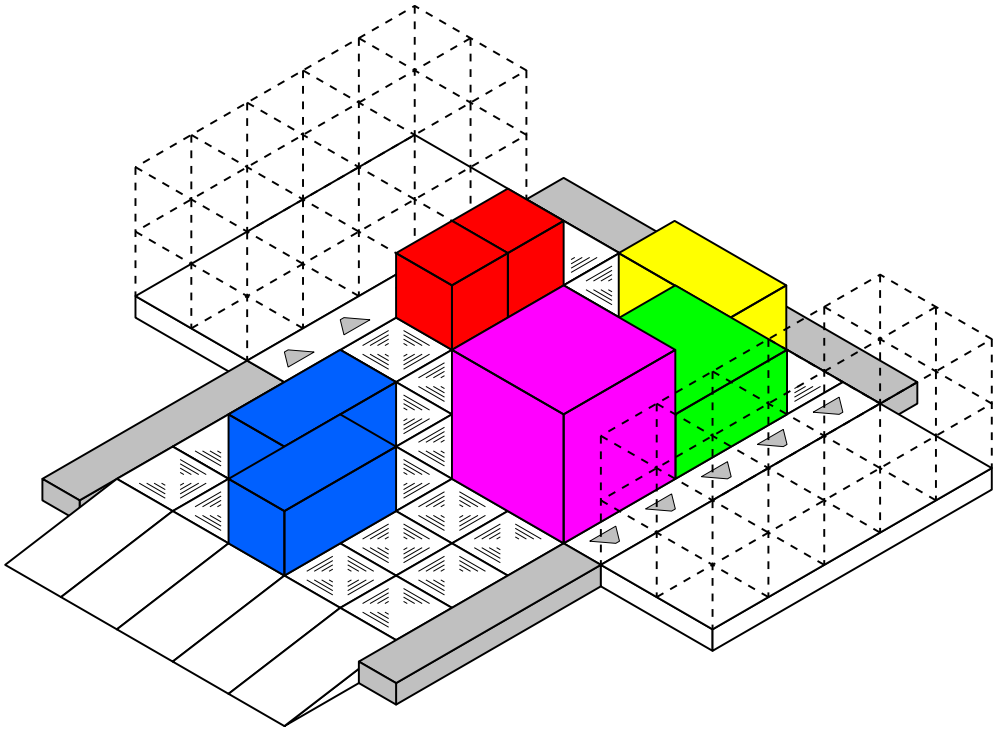

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

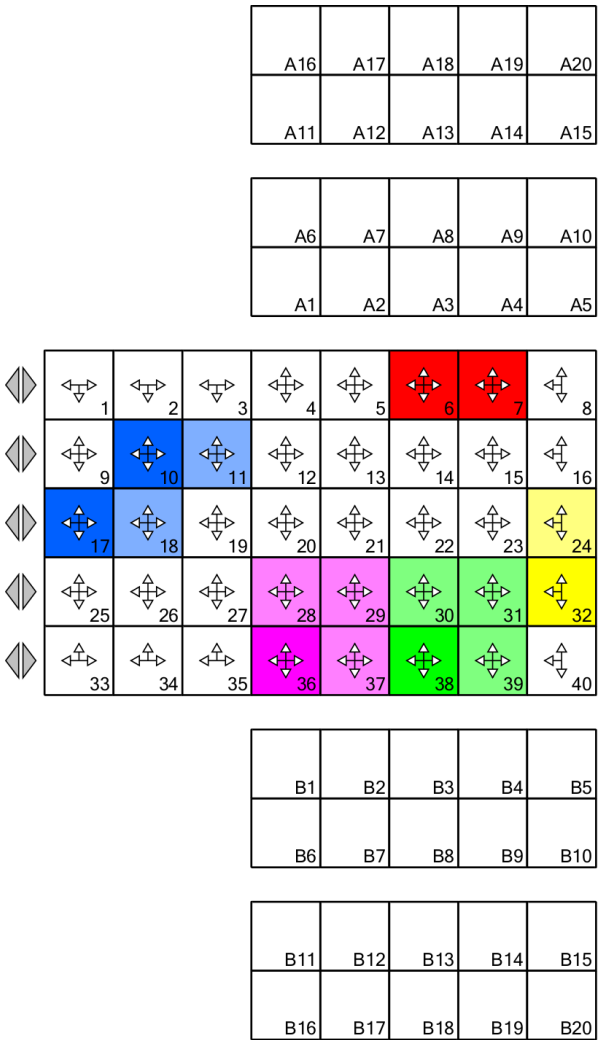
π-conveyor


z
y
x

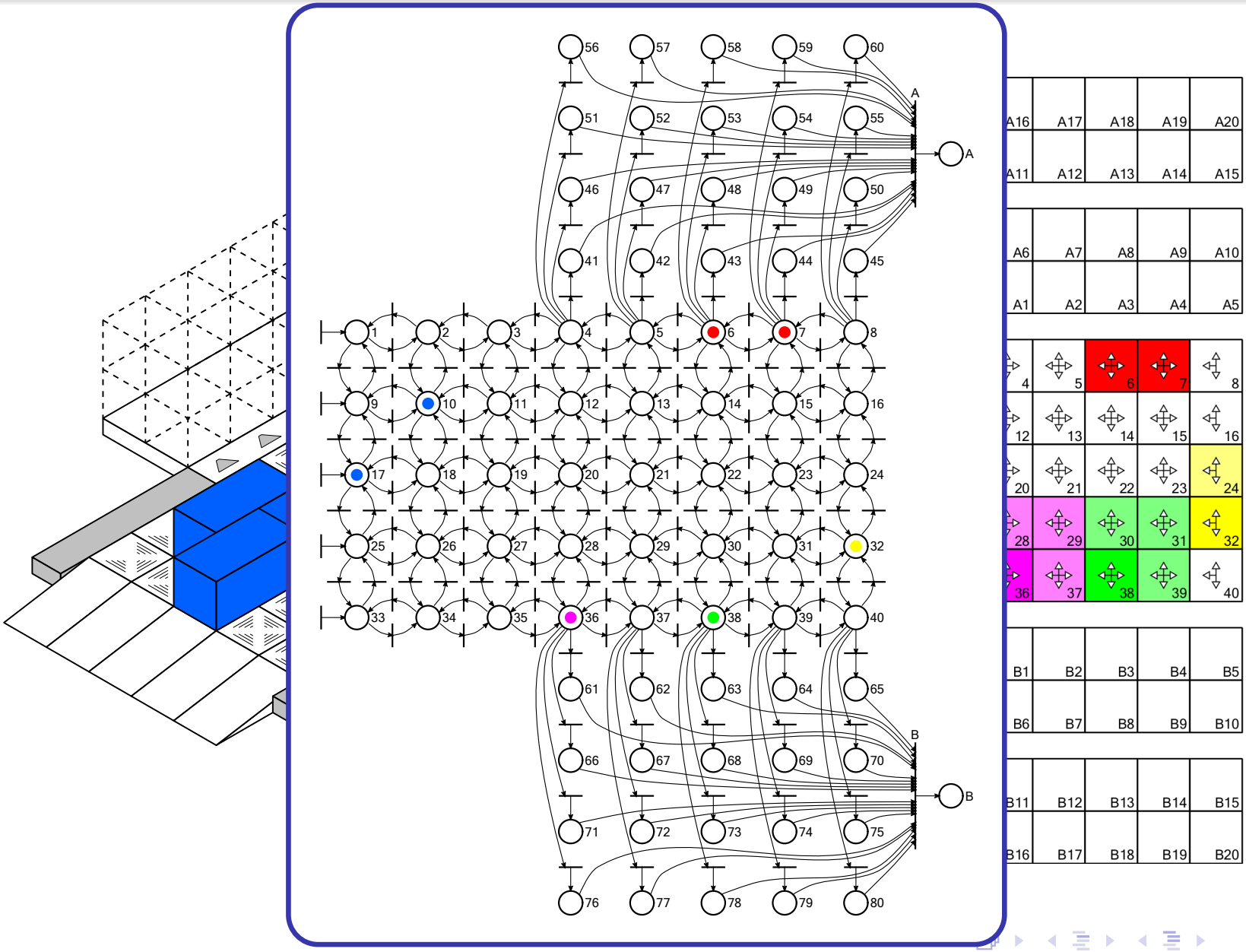

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

π-sorter/π-composer

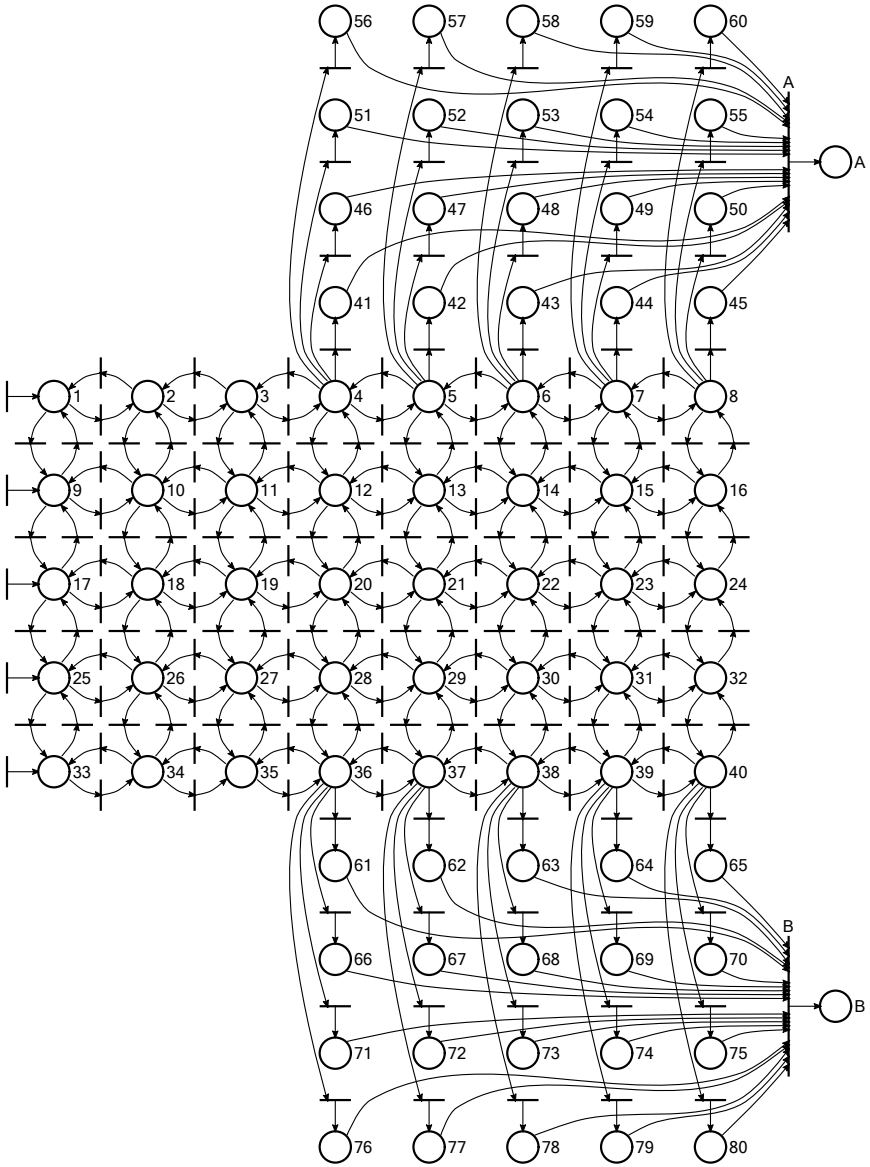




π-sorter/π-composer



π-sorter/π-composer



Colour sets and Guard functions

$\mathcal{C}(p_h) = \mathsf{BSIZE}, \forall h = 1, \ldots, 80$

$\mathcal{C}(p_l) = \mathsf{STRUCT}, \forall l = \mathsf{A}, \mathsf{B}$

$\mathcal{V} = \{b_h : \mathsf{BSIZE}; h = 1, \ldots, 80\}$

$E(t_\mathsf{A}, p_\mathsf{A}) = 1'(b_{41}, \ldots, b_{60})$

$E(t_\mathsf{B}, p_\mathsf{B}) = 1'(b_{61}, \ldots, b_{80})$

(all the other arc expr. as before: $1'b_h$)

$G(t_{h-k}) = \ldots$ (as before)

$G(t_{\mathsf{A}k}) = f_{\mathsf{LS}}(b_4, \ldots, b_8, b_{41}, \ldots, b_{60})$

$G(t_{\mathsf{B}k}) = f_{\mathsf{LS}}(b_{36}, \ldots, b_{40}, b_{61}, \ldots, b_{80})$

$G(t_\mathsf{A}) = g_{\mathsf{LS}}(b_{41}, \ldots, b_{60})$

$G(t_\mathsf{B}) = g_{\mathsf{LS}}(b_{61}, \ldots, b_{80})$

π-sorter/π-composer

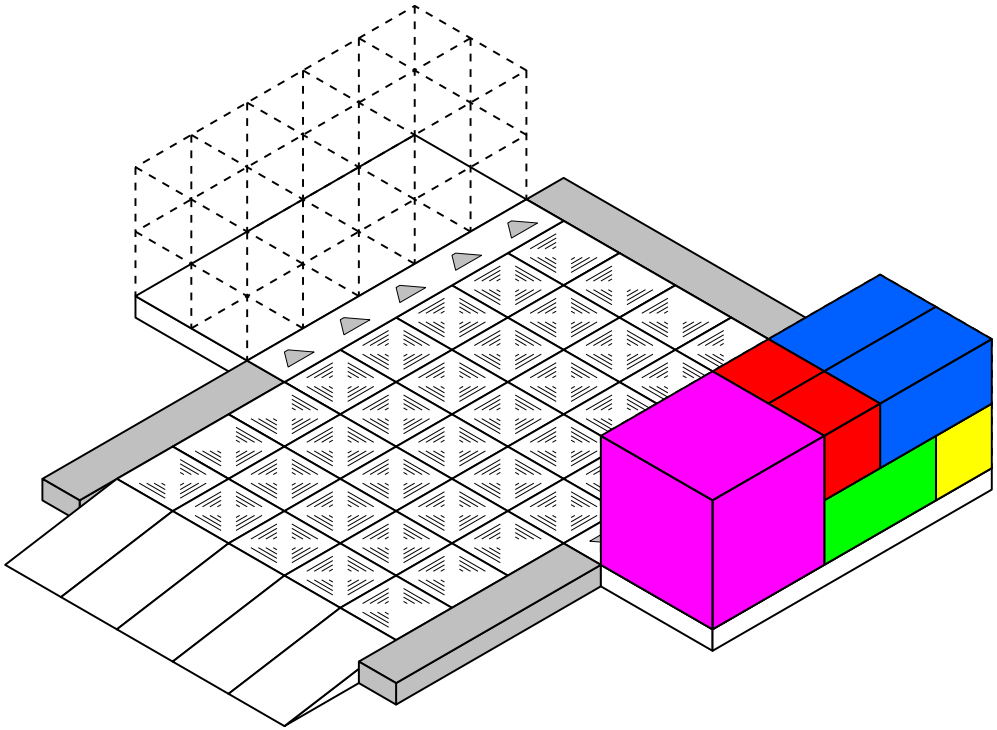
Loading sequence

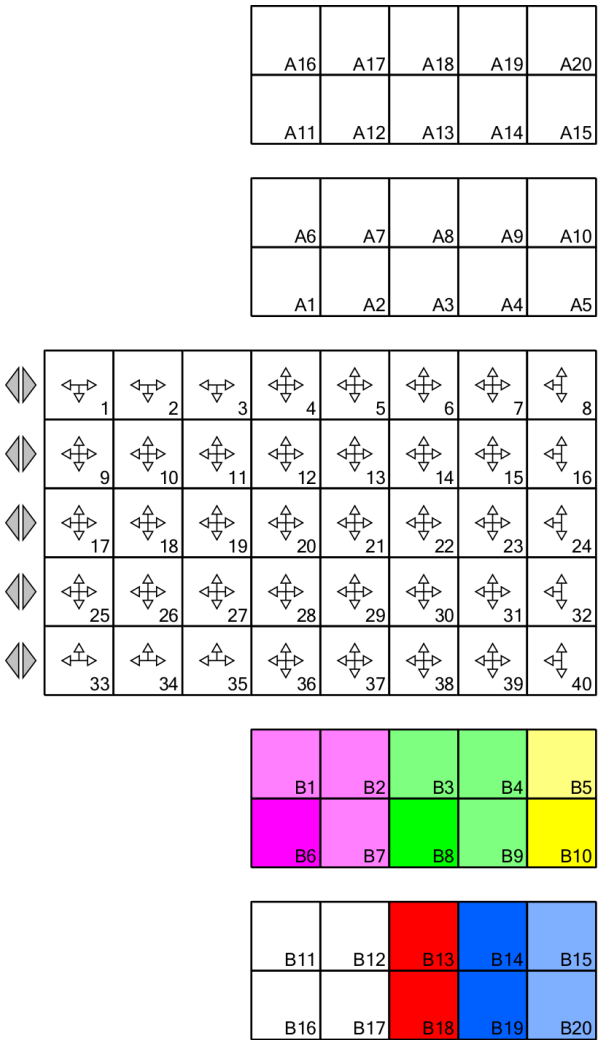
Sequence	π-container	π-core at start	π-core at end
1st	222 (magenta)	cell 36	cell B6
2nd	221 (green)	cell 38	cell B8
3rd	121 (yellow)	cell 40	cell B10
4th	111 (red)	cell 38	cell B18
5th	111 (red)	cell 38	cell B13
6th	211 (blue)	cell 39	cell B19
7th	211 (blue)	cell 39	cell B14

Guard functions

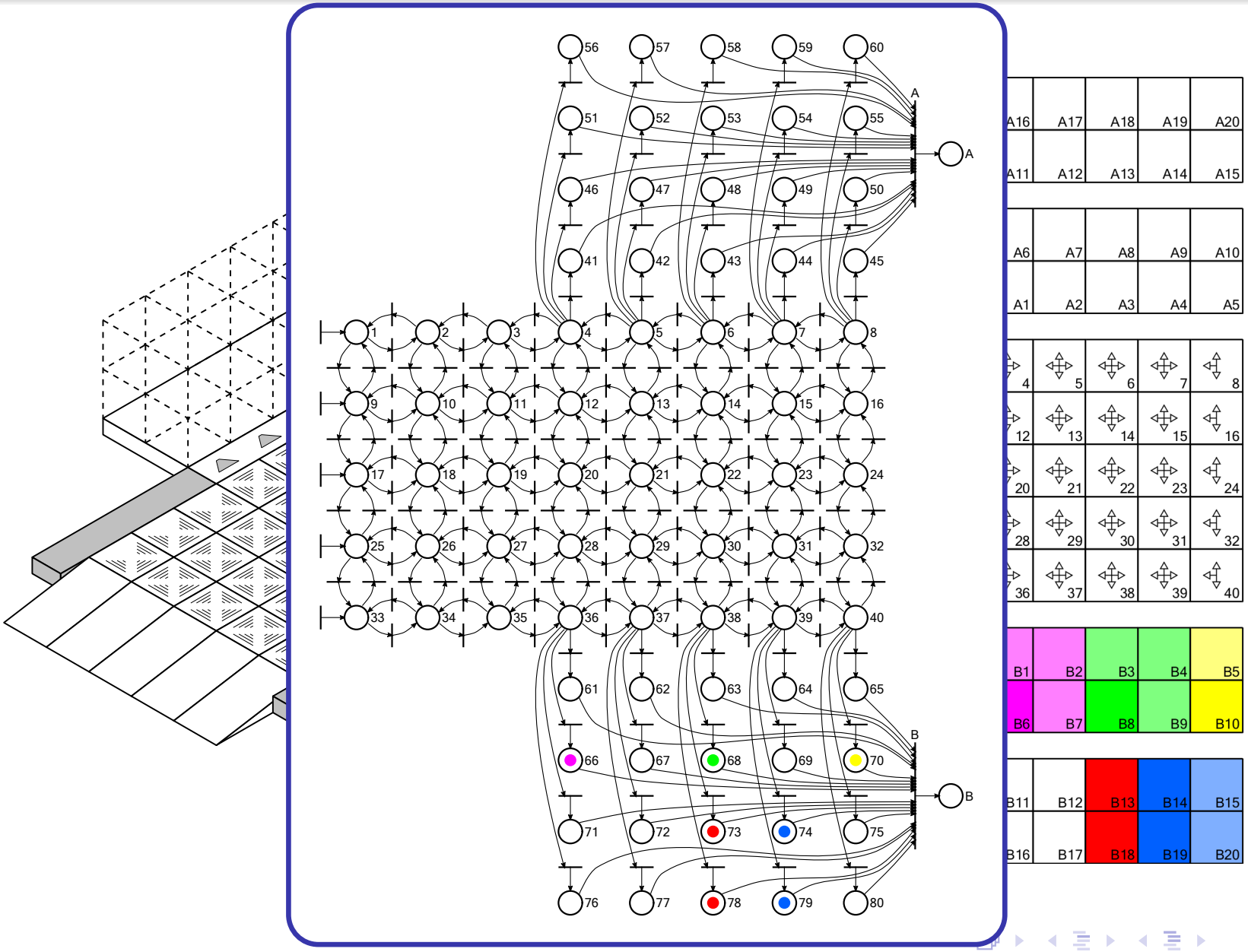
$$G(t_{\mathrm{B66}}) = (b_{36} = 222)$$
$$G(t_{\mathrm{B68}}) = (b_{38} = 221) \wedge (b_{66} = 222)$$
$$G(t_{\mathrm{B70}}) = (b_{40} = 121) \wedge (b_{68} = 221)$$
$$G(t_{\mathrm{B78}}) = (b_{38} = 111) \wedge (b_{70} = 121)$$
$$G(t_{\mathrm{B73}}) = (b_{38} = 111) \wedge (b_{78} = 111)$$
$$G(t_{\mathrm{B79}}) = (b_{39} = 211) \wedge (b_{73} = 111)$$
$$G(t_{\mathrm{B74}}) = (b_{39} = 211) \wedge (b_{79} = 211)$$
$$G(t_{\mathrm{B}k}) = 0, k = 61, 62, 63, 64, 65, 67, 69, 71, 72, 75, 76, 77, 80$$

π-sorter/π-composer

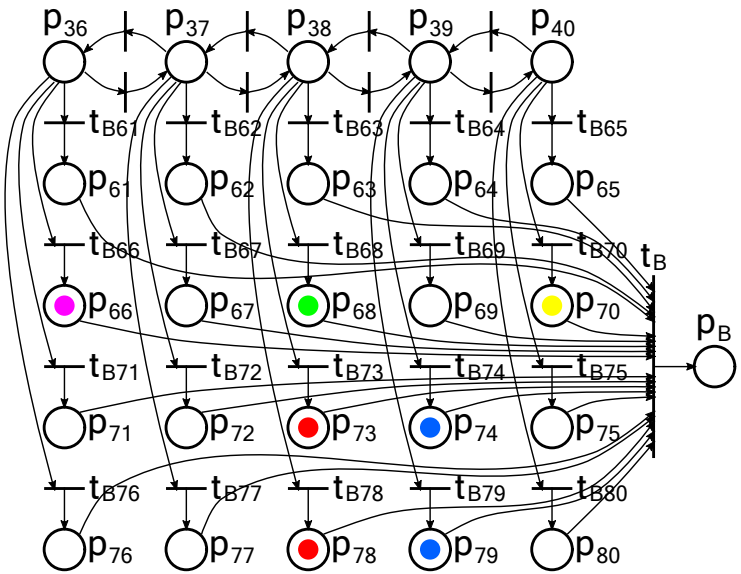


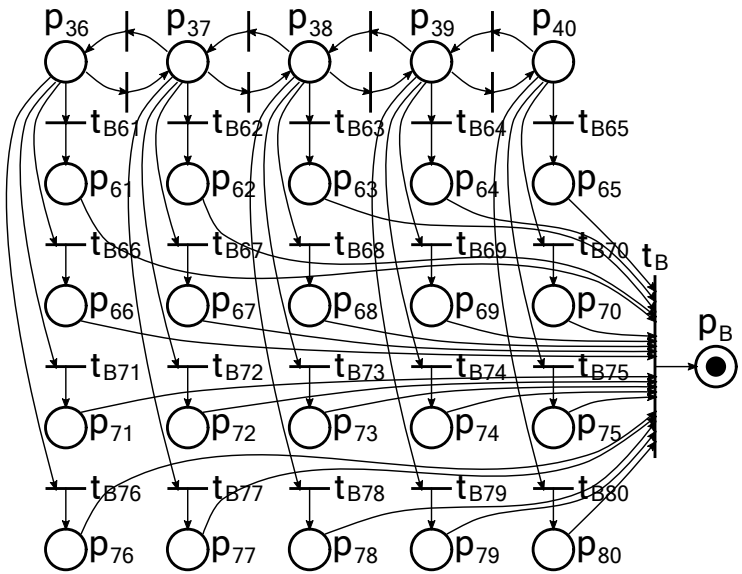


π-sorter/π-composer



π-sorter/π-composer

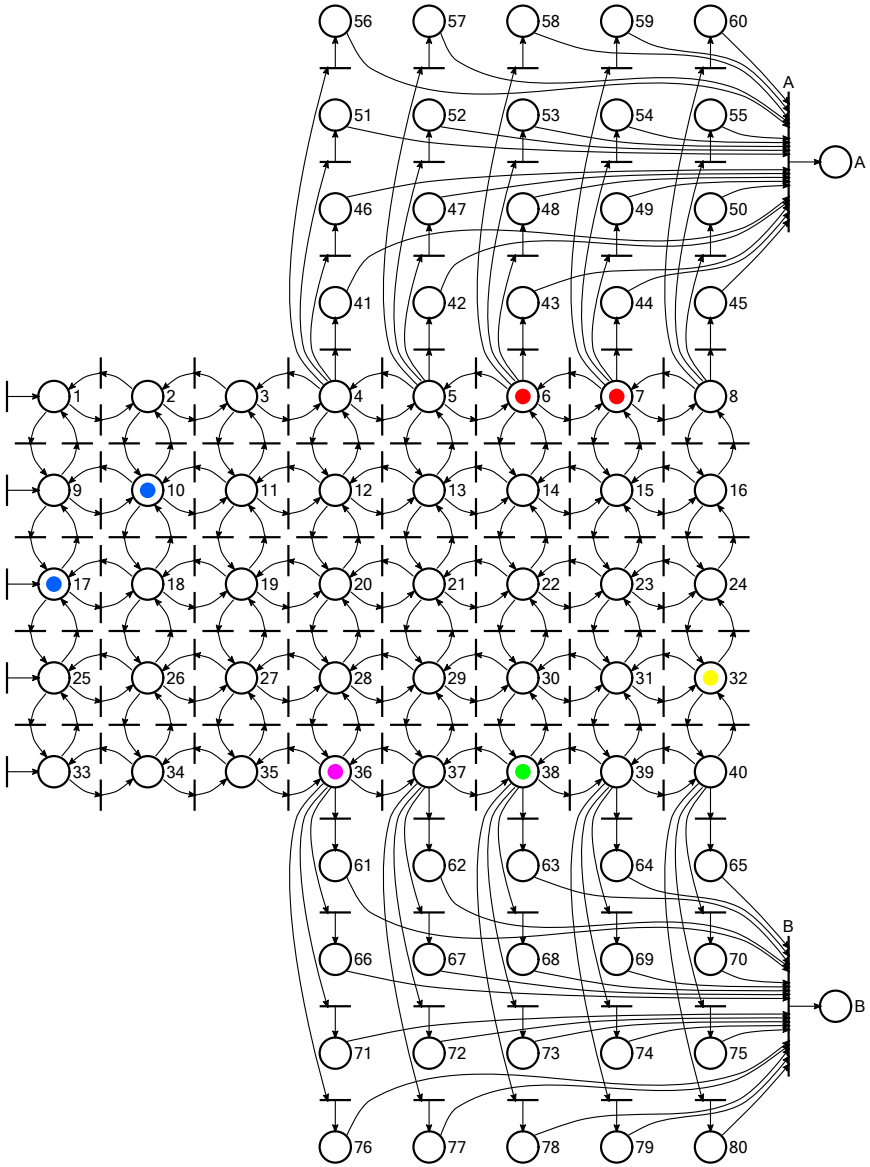


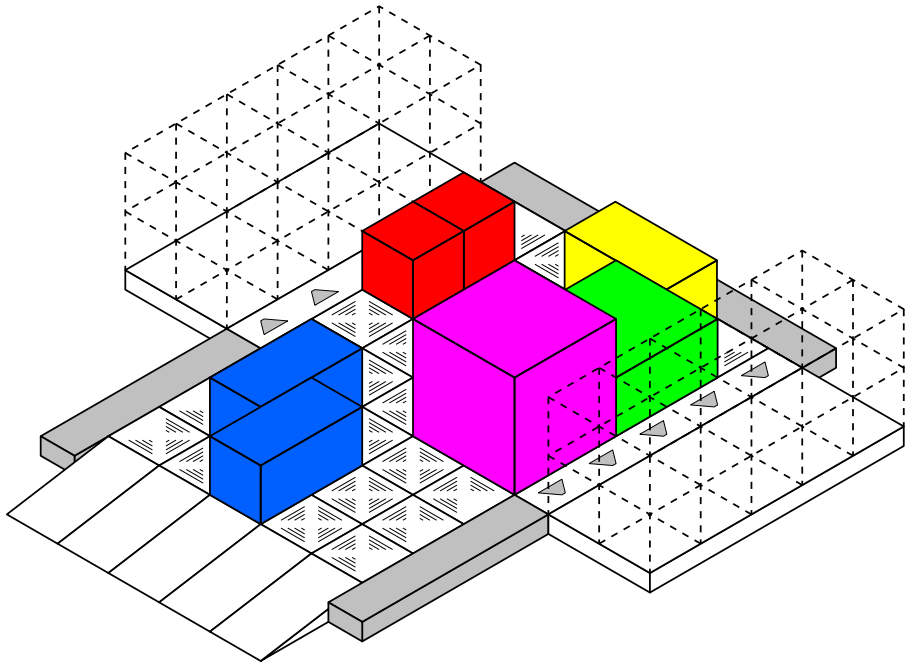


Guard function

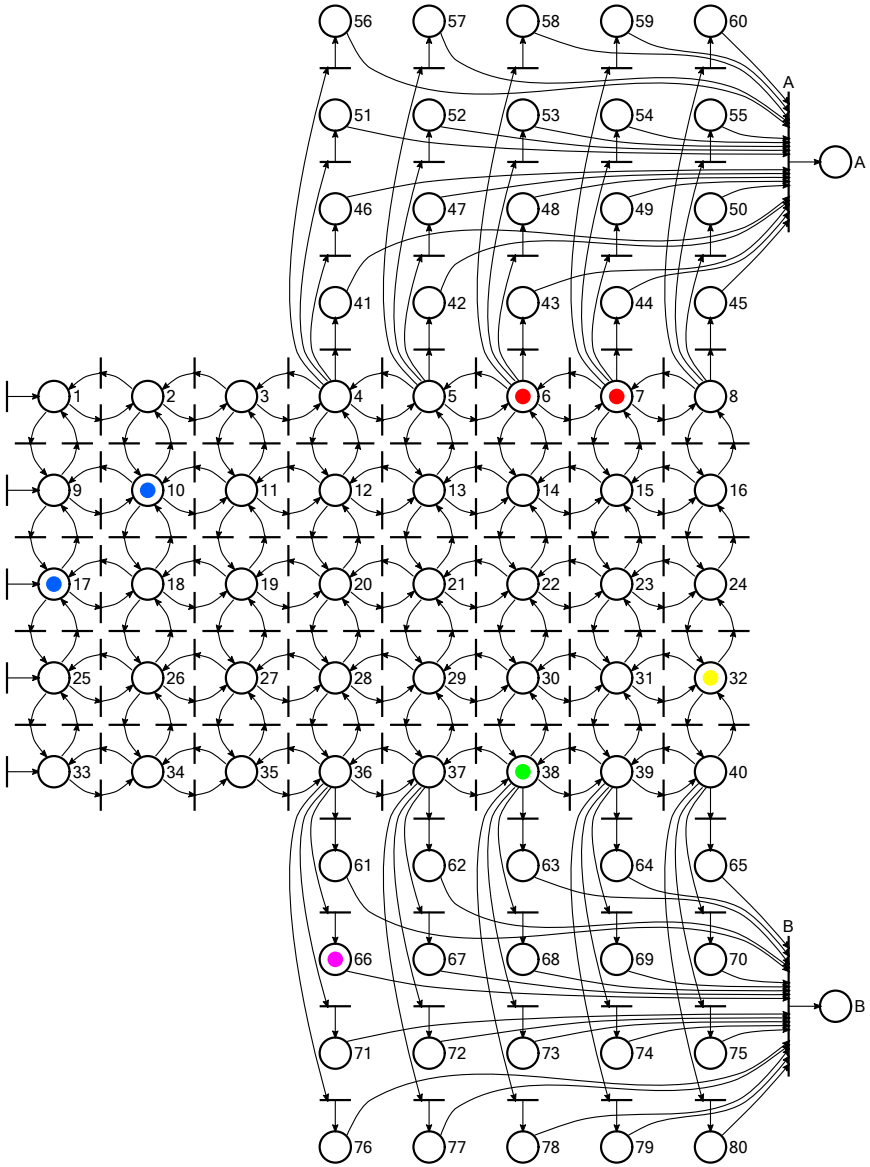
$$G(t_B) = (b_{61} = 0) \wedge (b_{62} = 0) \wedge (b_{63} = 0) \wedge (b_{64} = 0) \wedge (b_{65} = 0) \wedge (b_{66} = 222) \wedge (b_{67} = 0) \wedge (b_{68} = 221) \wedge (b_{69} = 0) \wedge (b_{70} = 121) \wedge (b_{71} = 0) \wedge (b_{72} = 0) \wedge (b_{73} = 111) \wedge (b_{74} = 211) \wedge (b_{75} = 0) \wedge (b_{76} = 0) \wedge (b_{77} = 0) \wedge (b_{78} = 111) \wedge (b_{79} = 211) \wedge (b_{80} = 0)$$

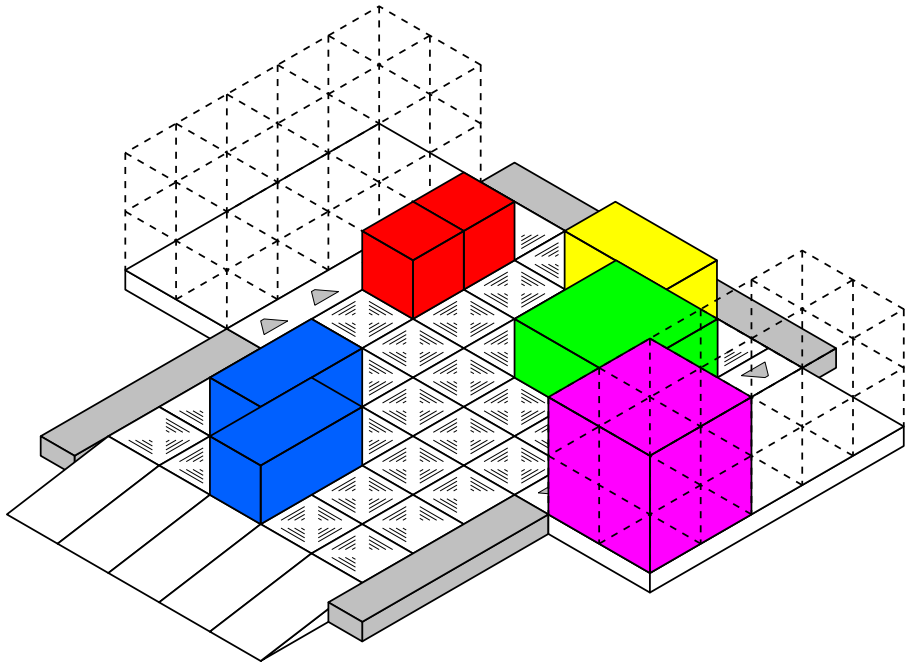
π-sorter/π-composer



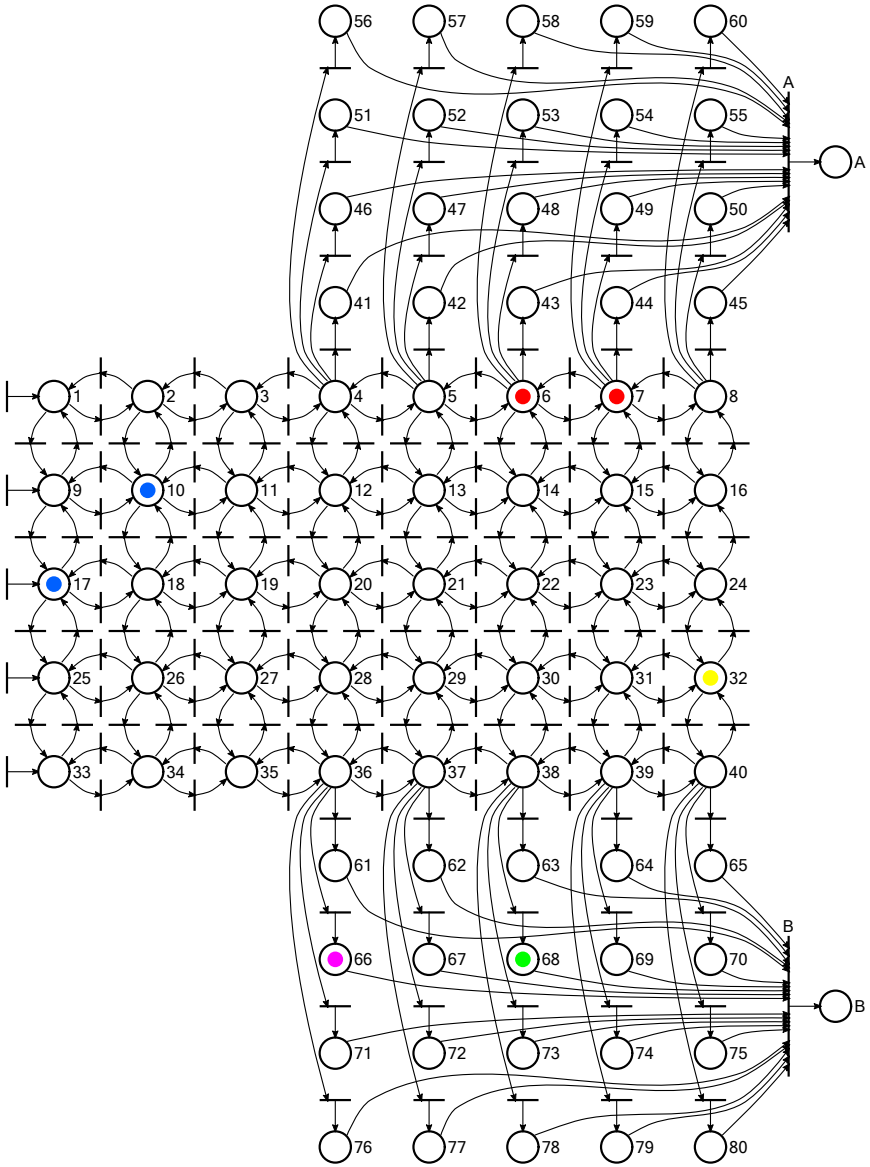


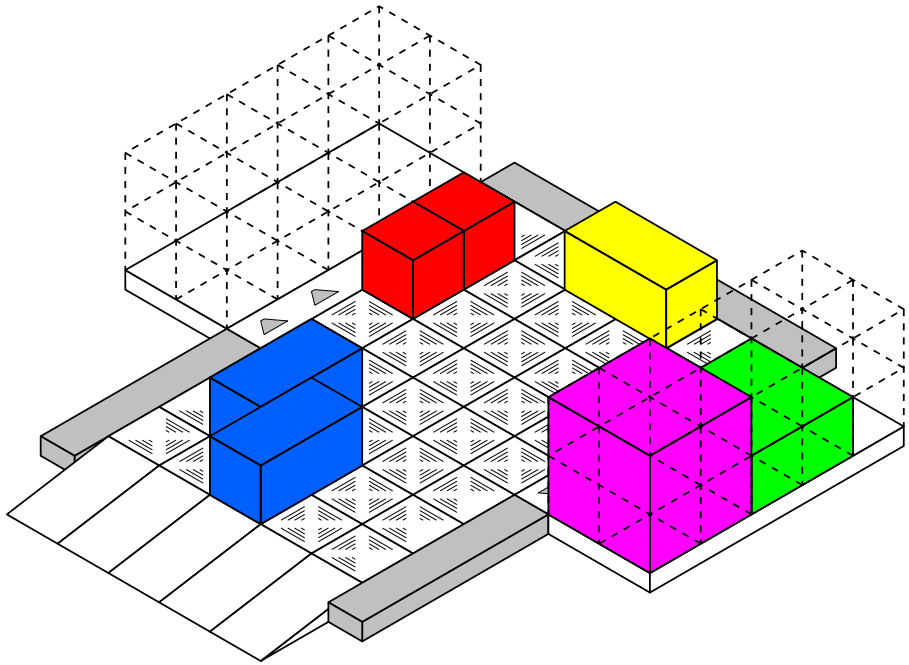
π-sorter/π-composer



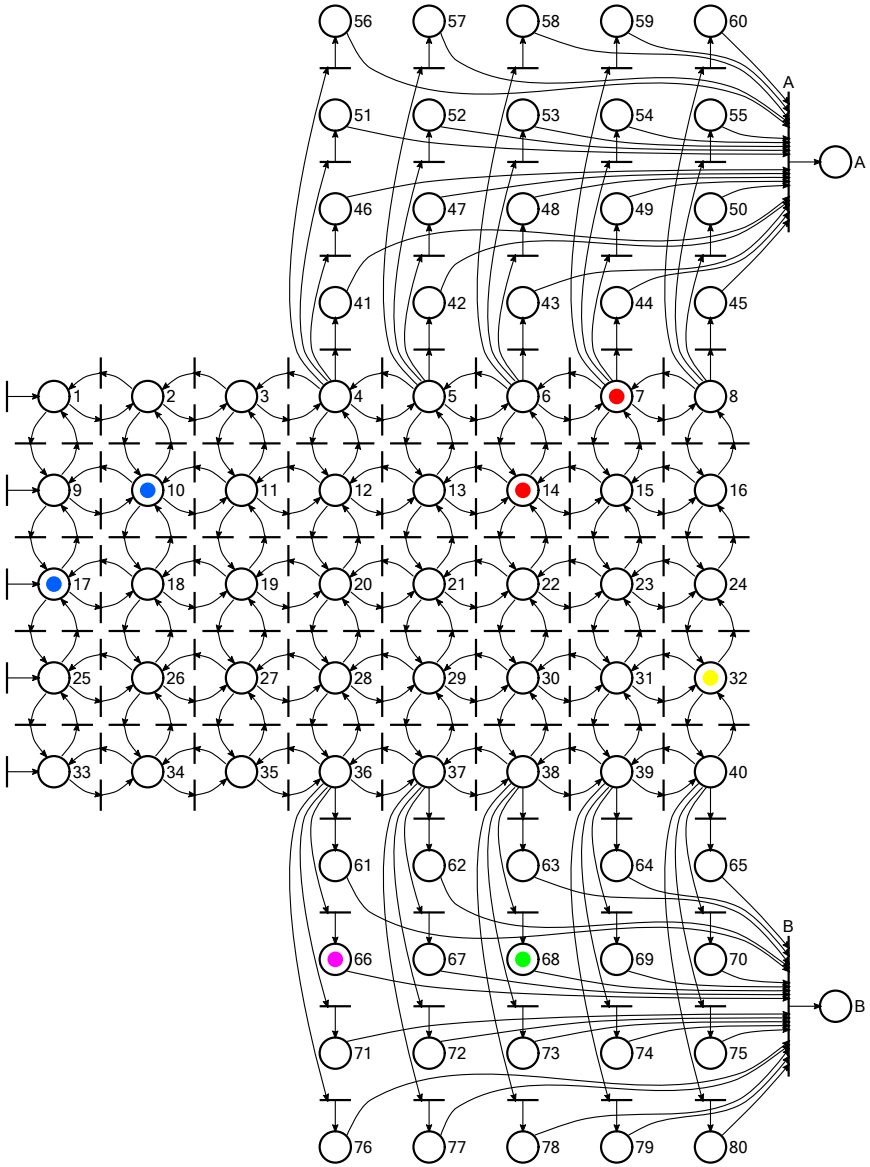


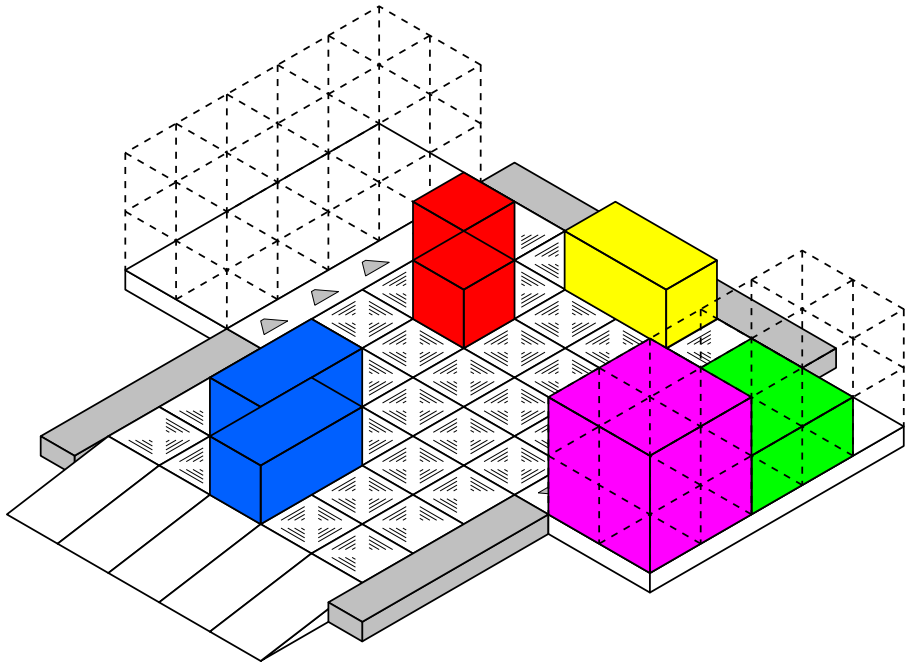
π-sorter/π-composer



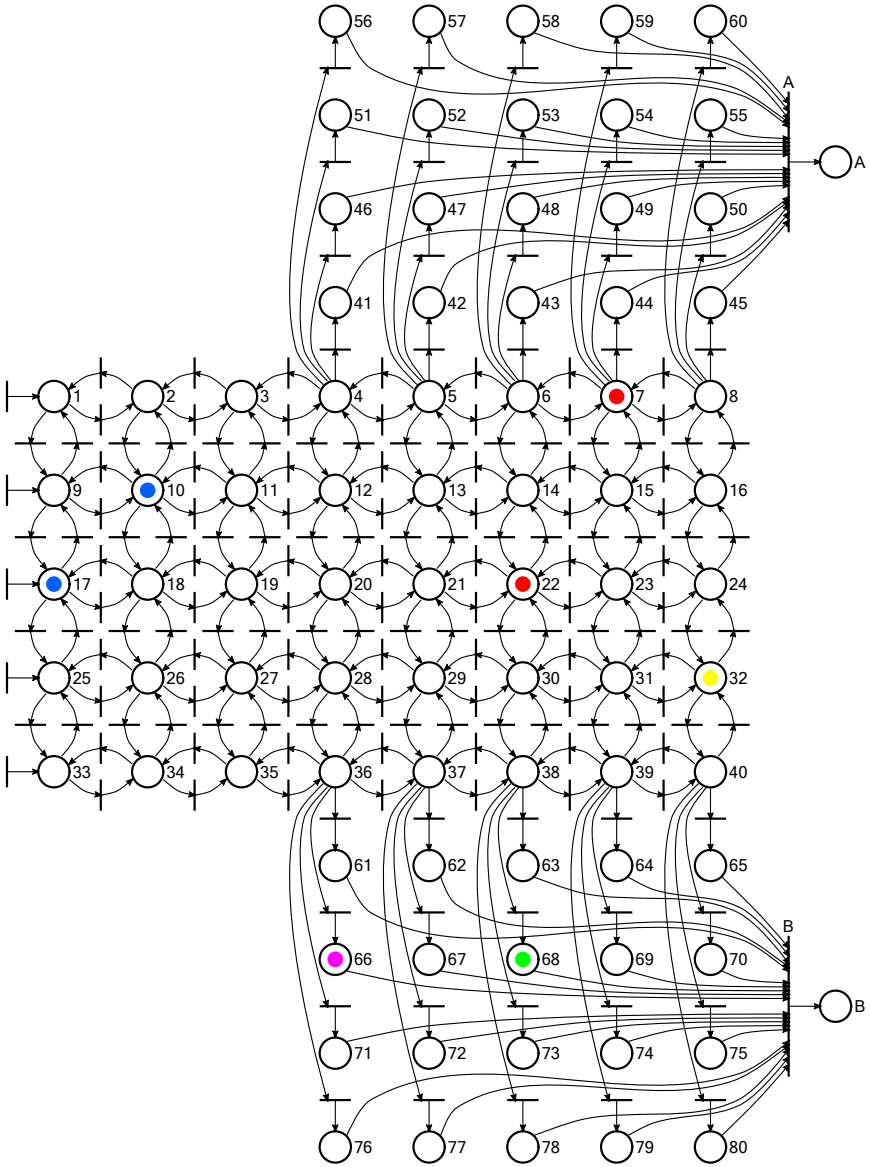


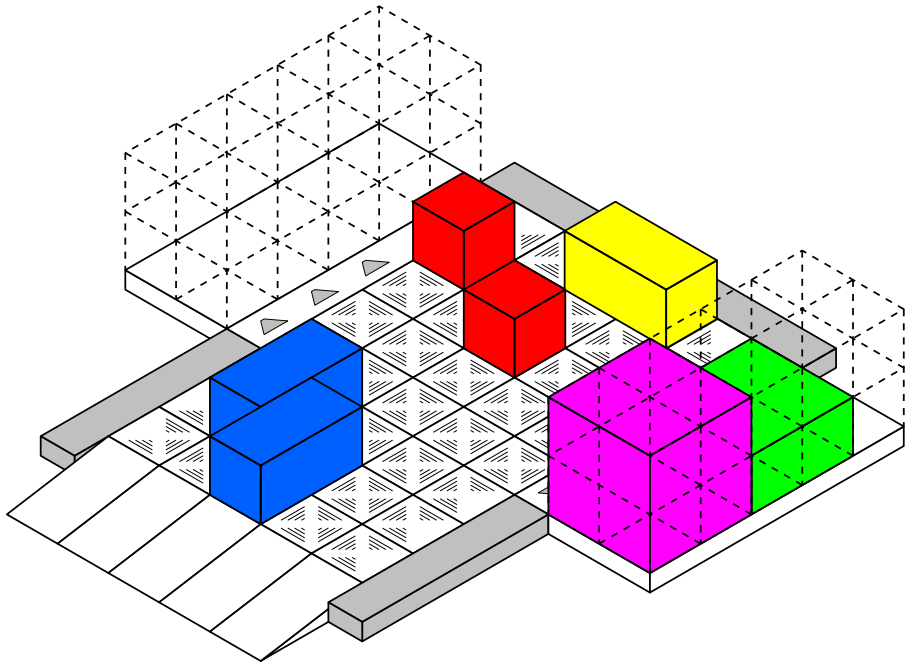
π-sorter/π-composer



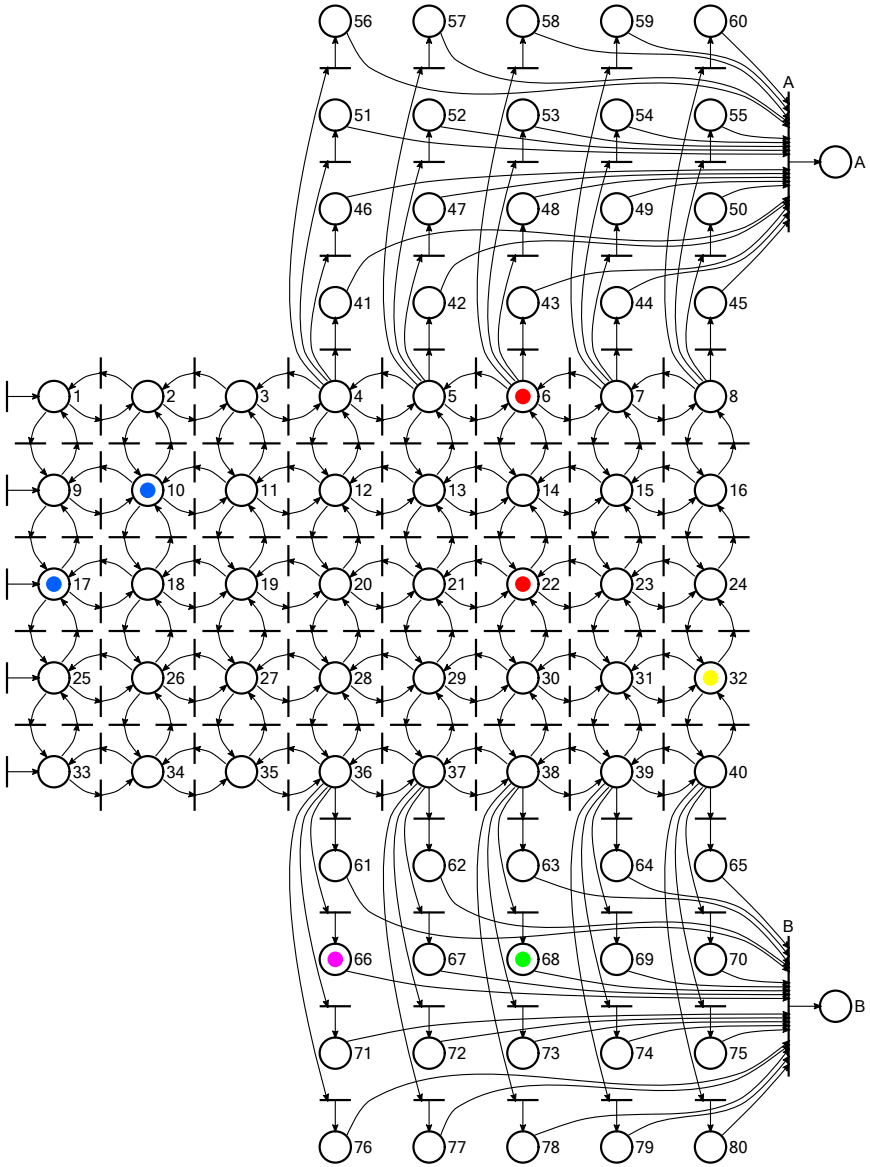


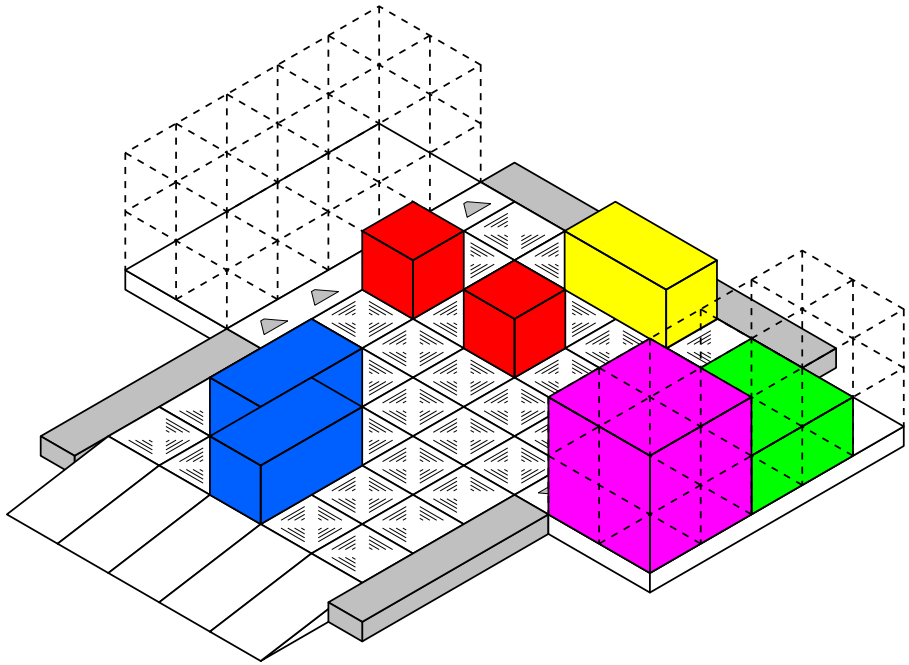
π-sorter/π-composer



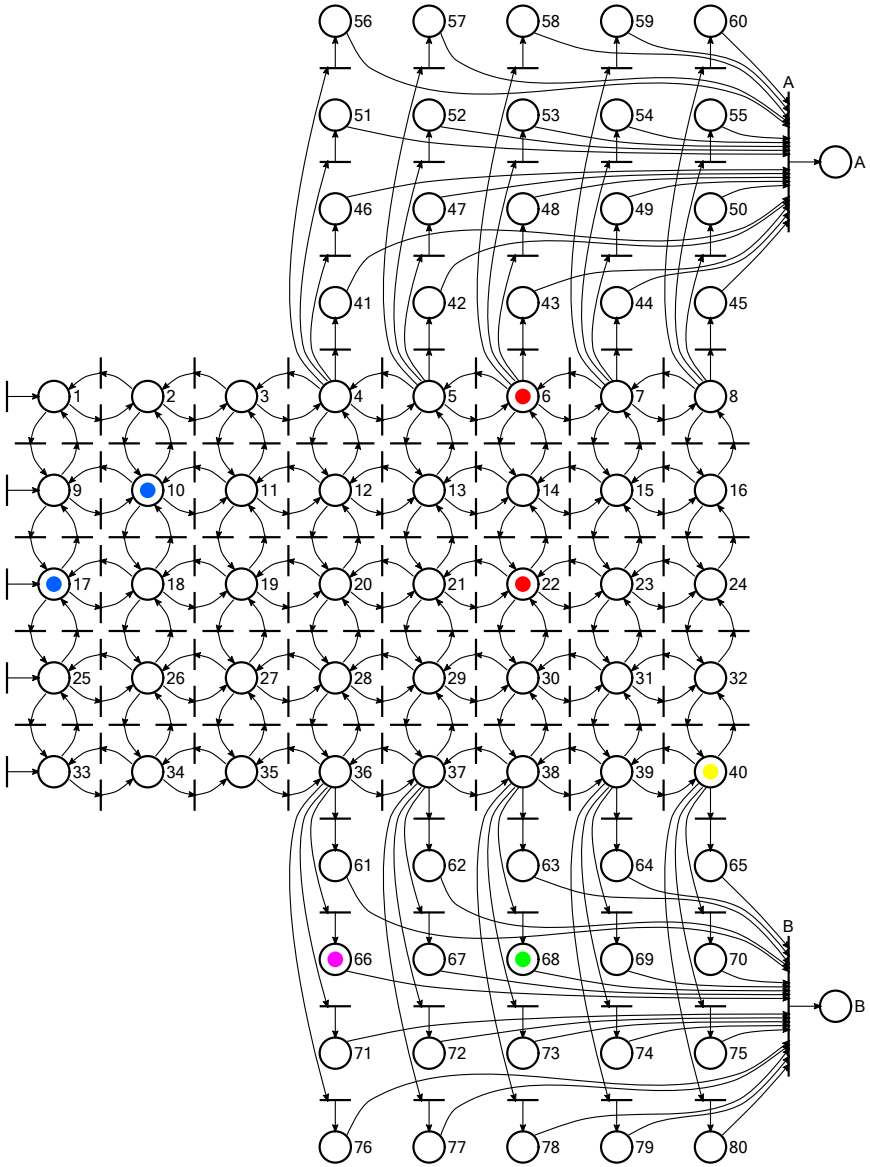


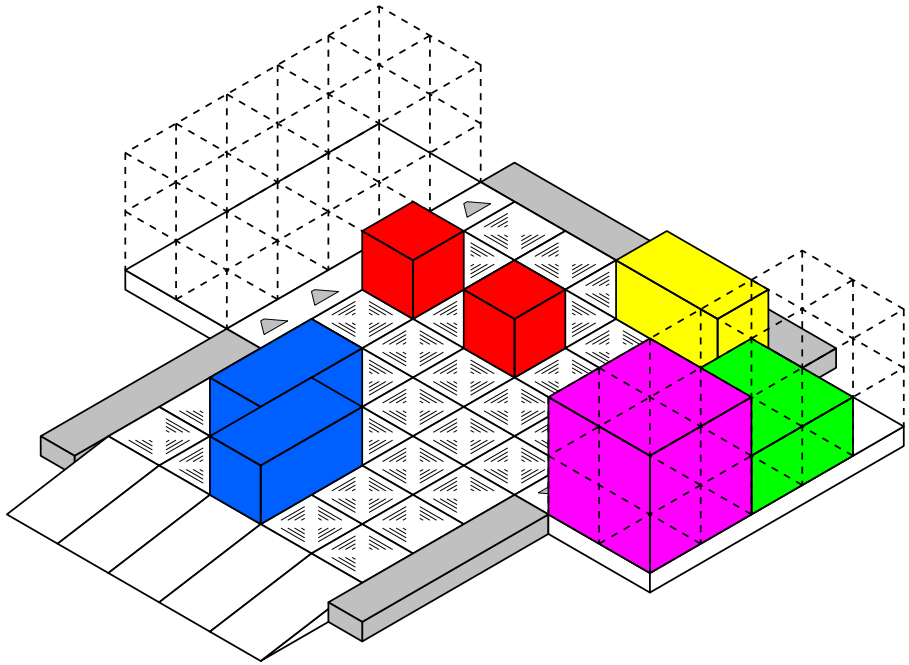
π-sorter/π-composer



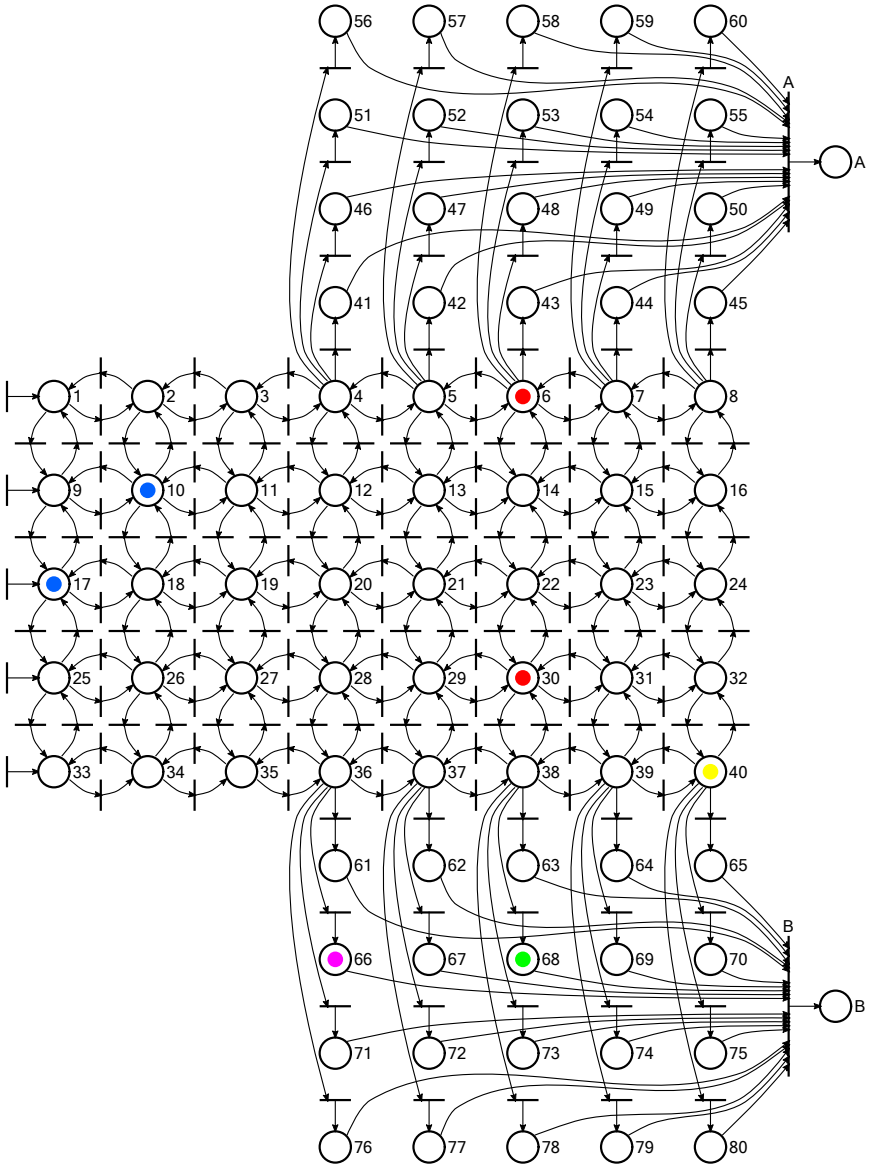


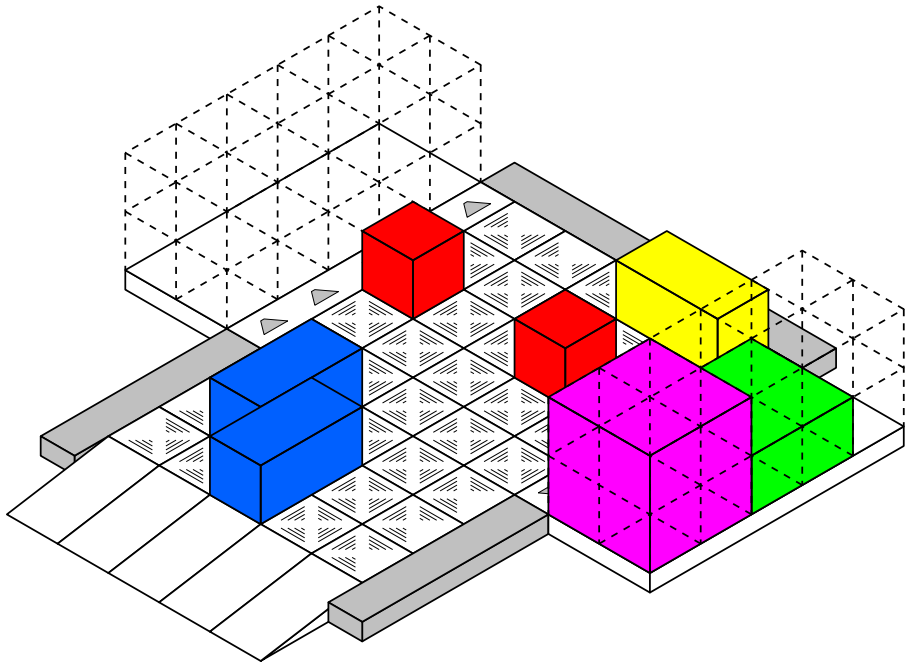
π-sorter/π-composer



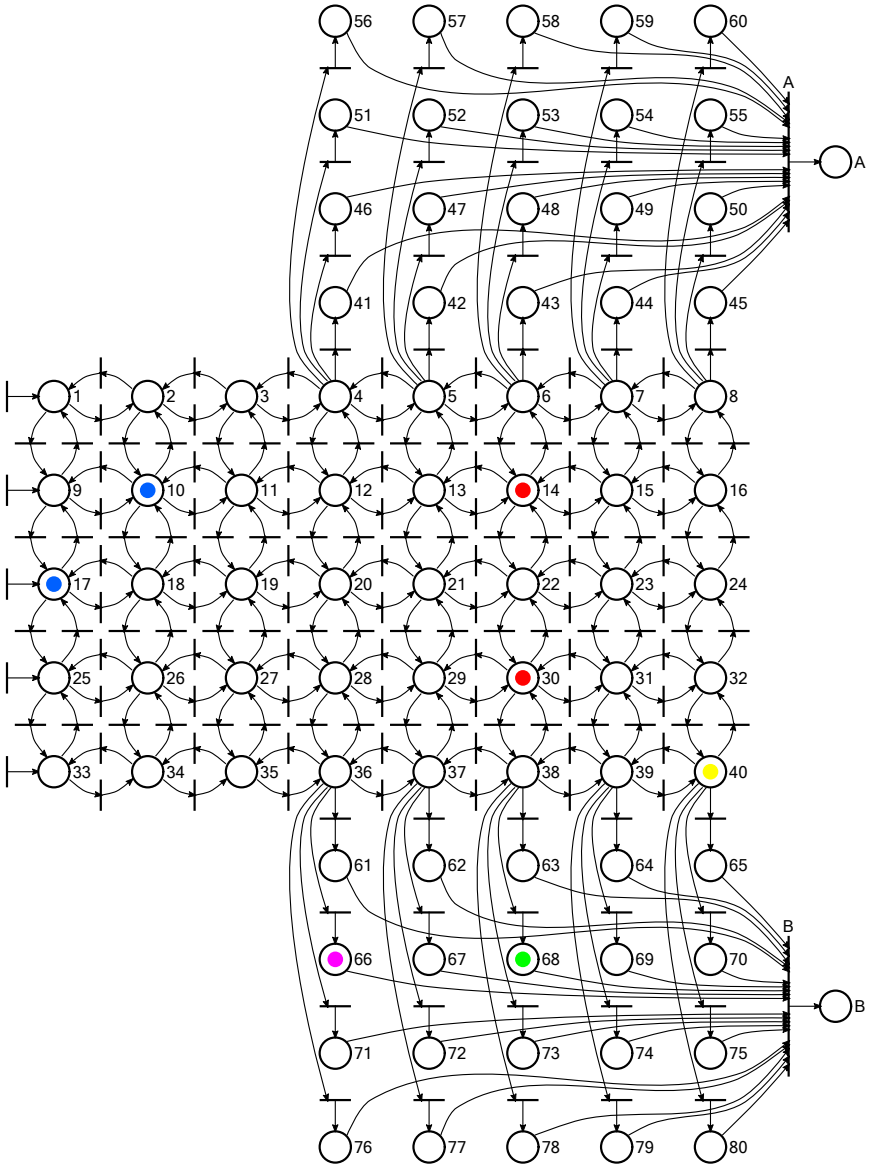


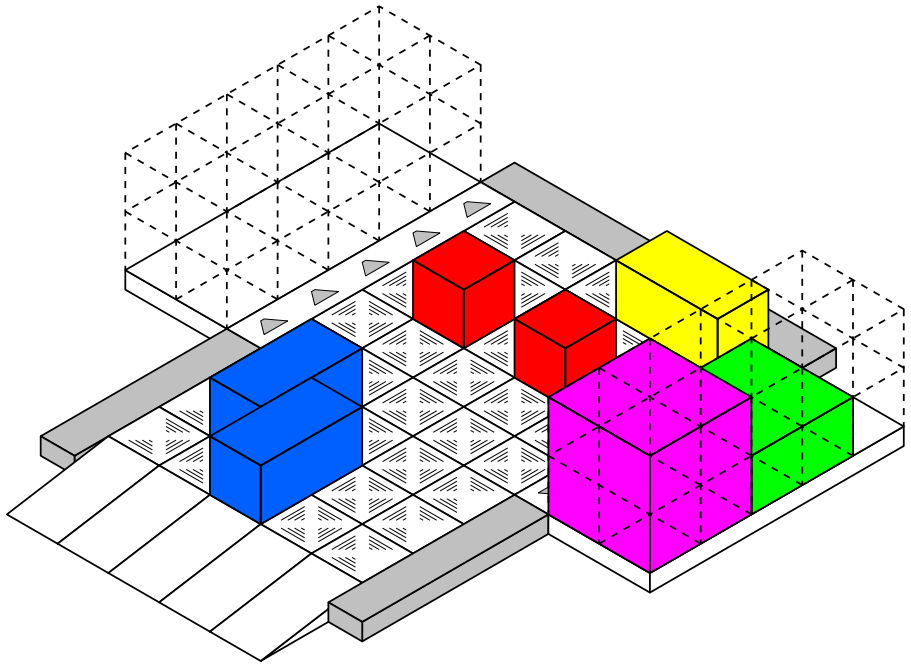
π-sorter/π-composer



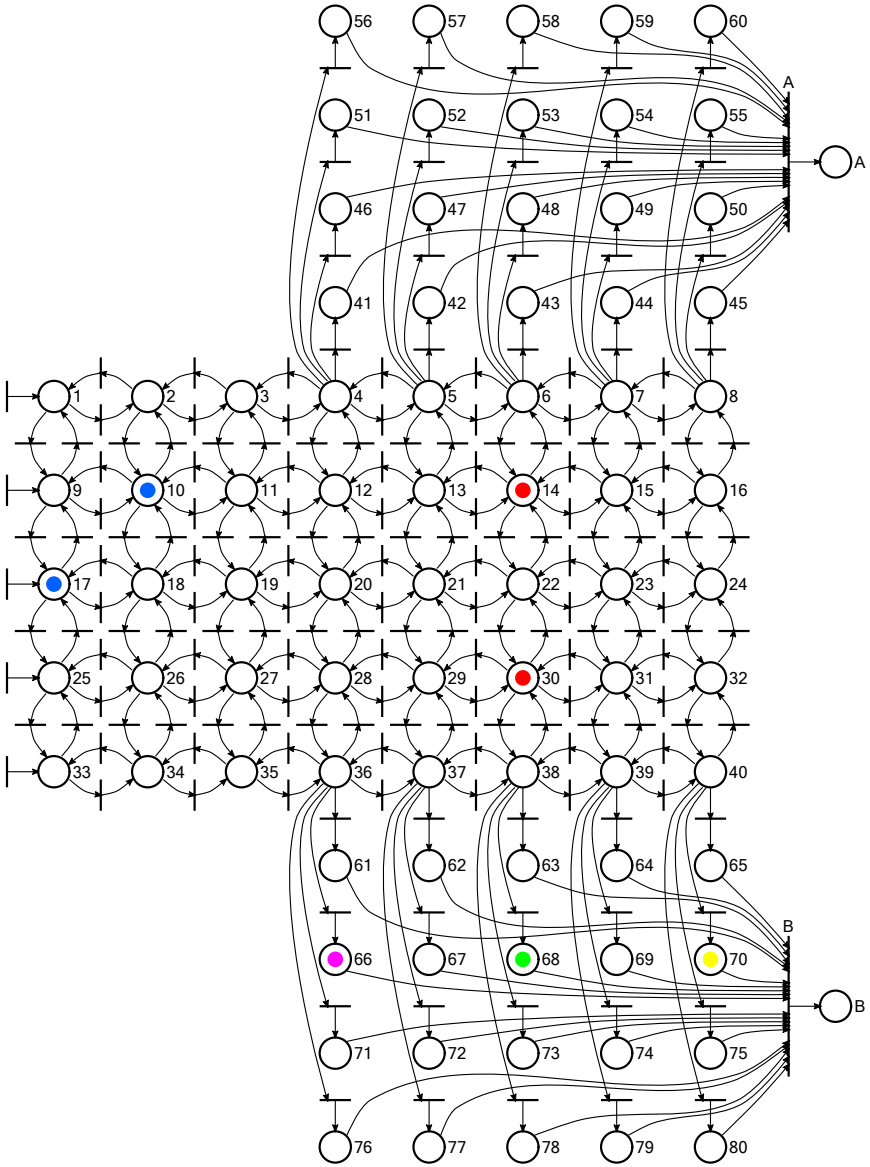


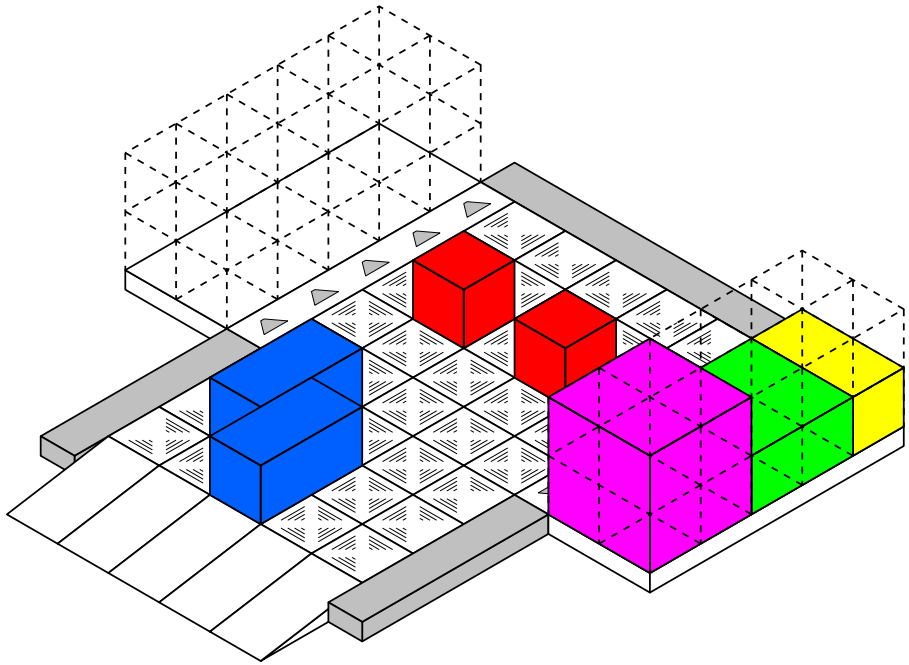
π-sorter/π-composer



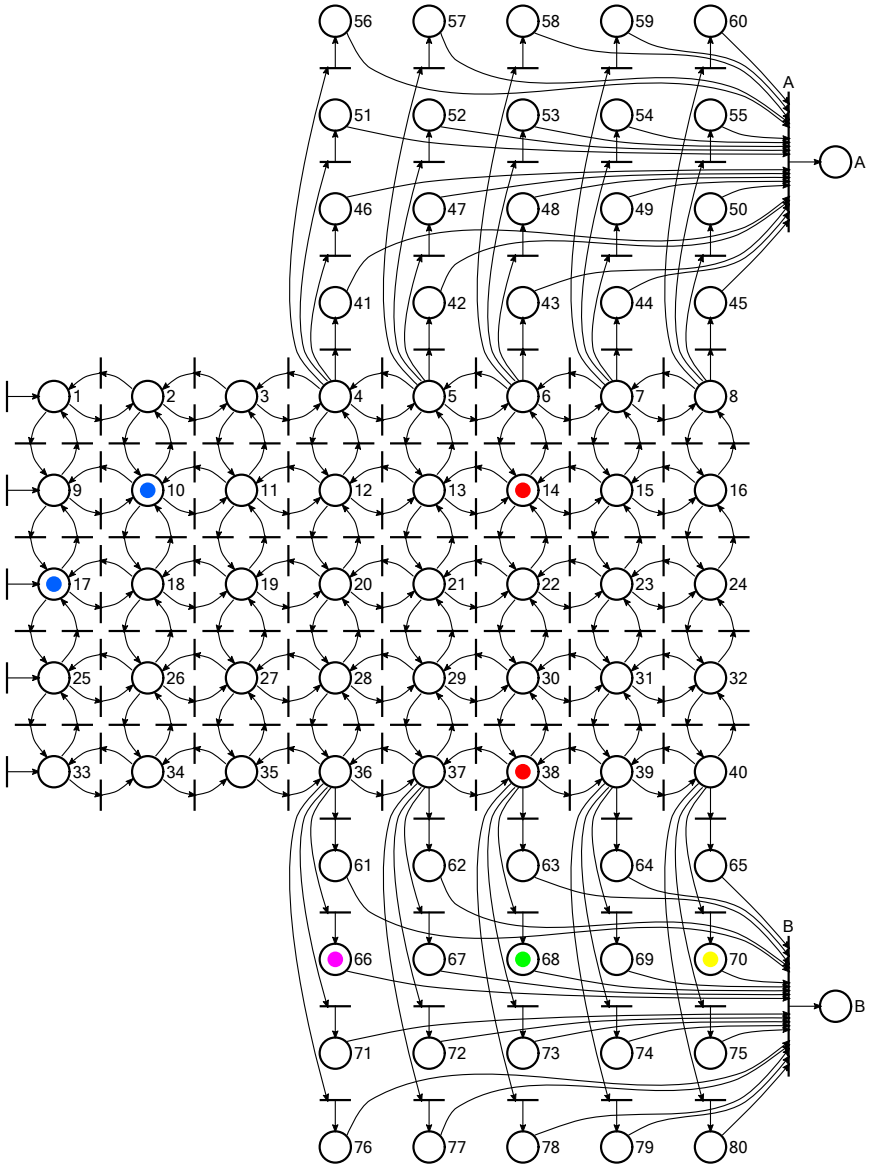


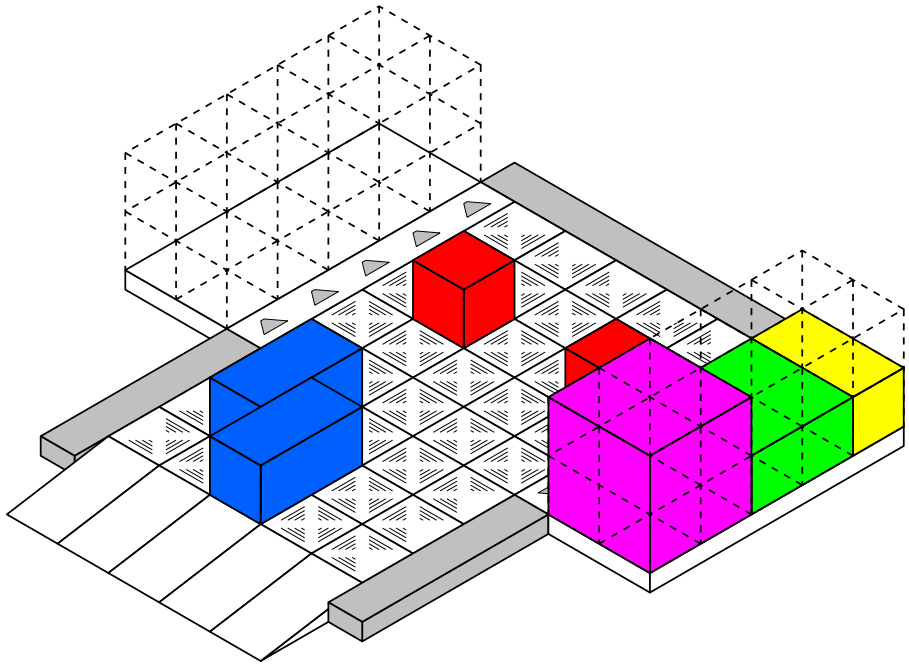
π-sorter/π-composer



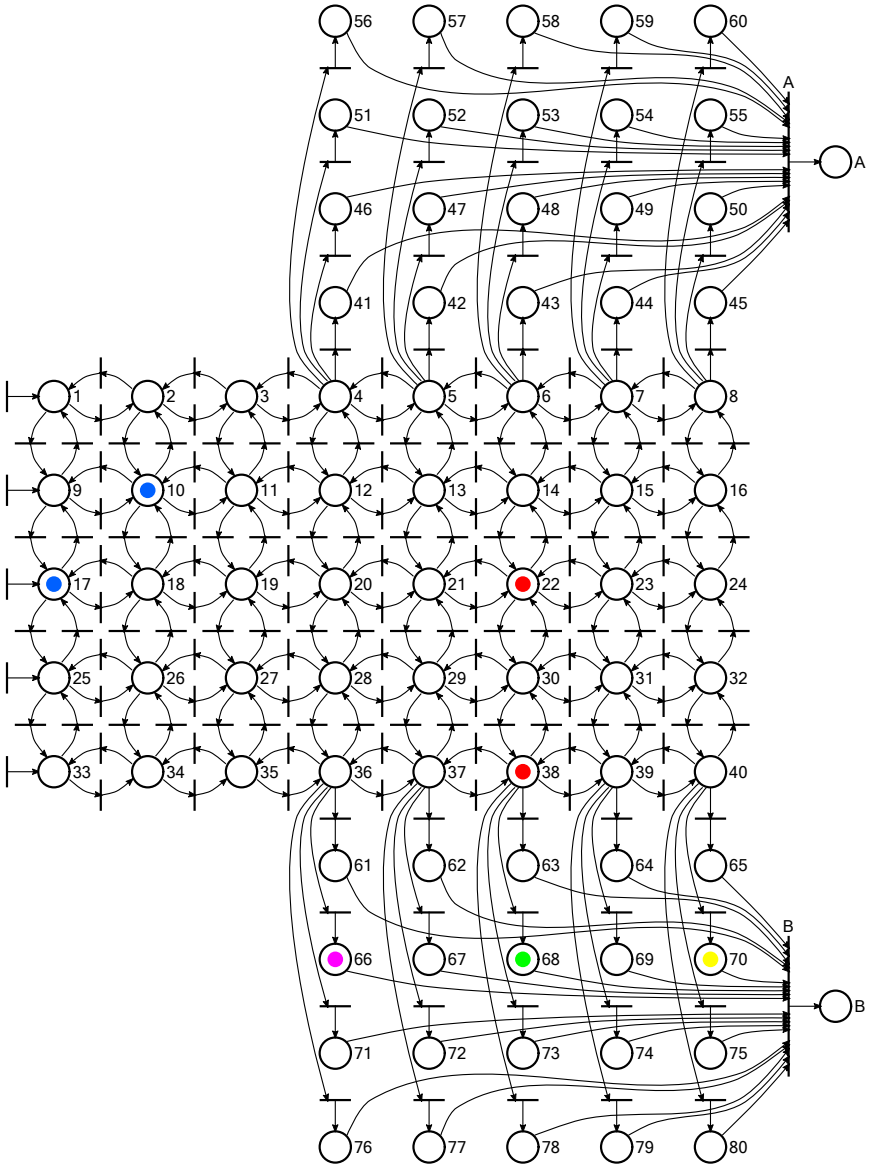


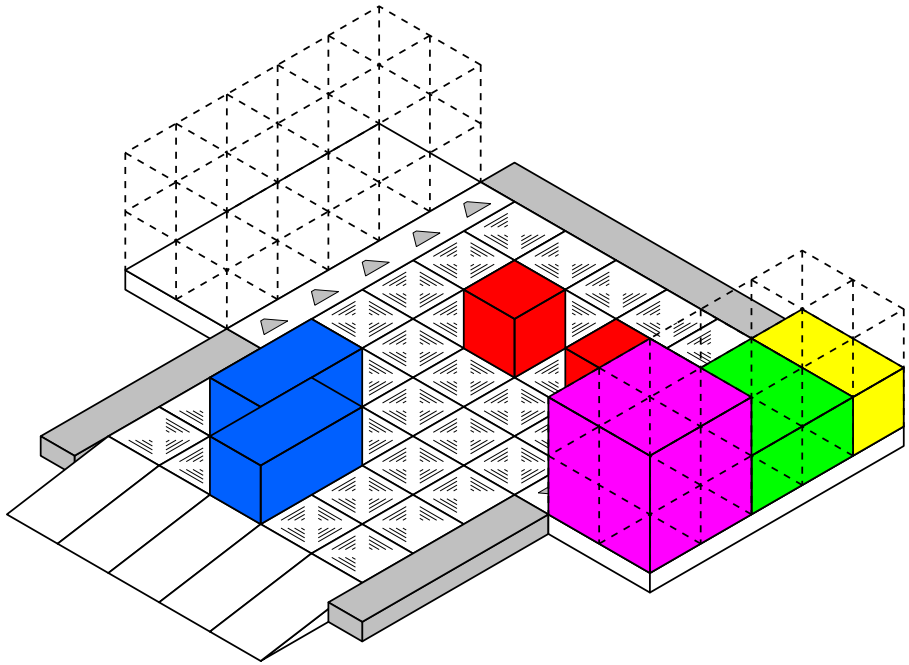
π-sorter/π-composer



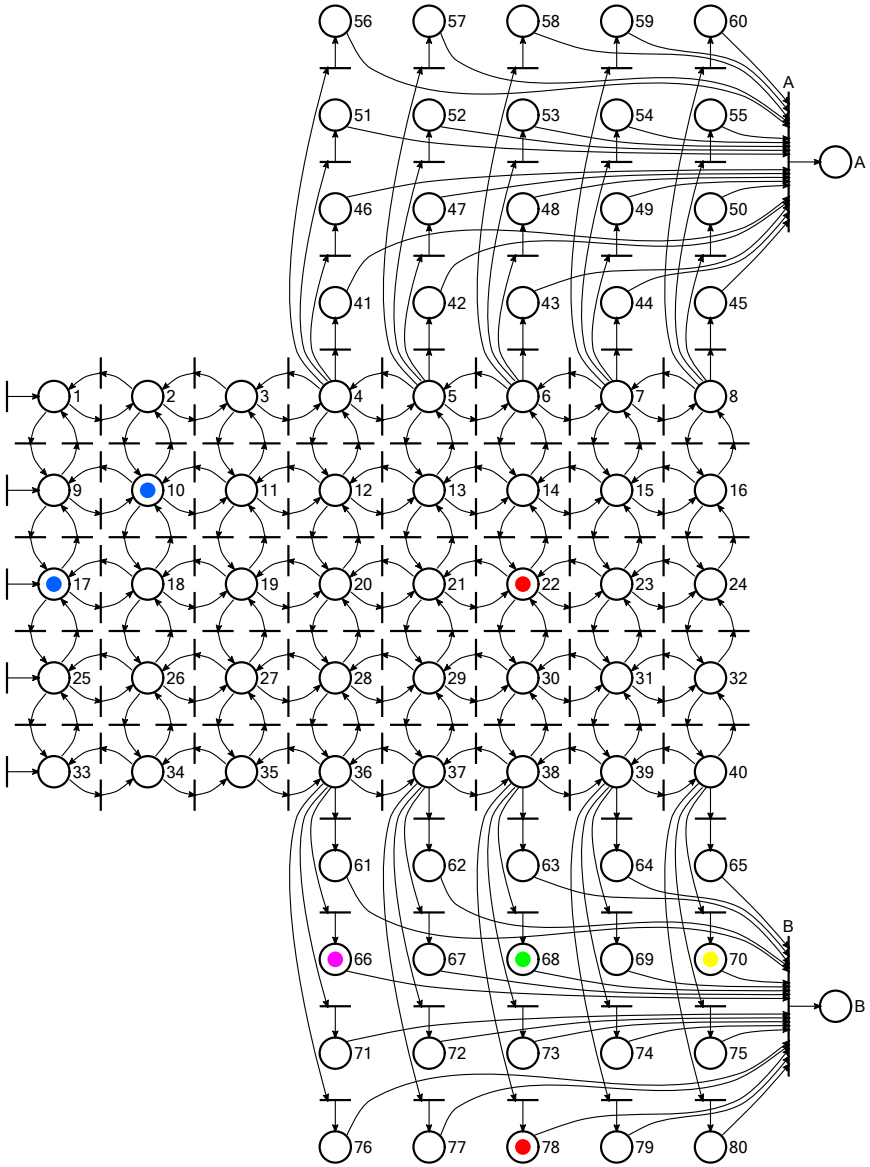


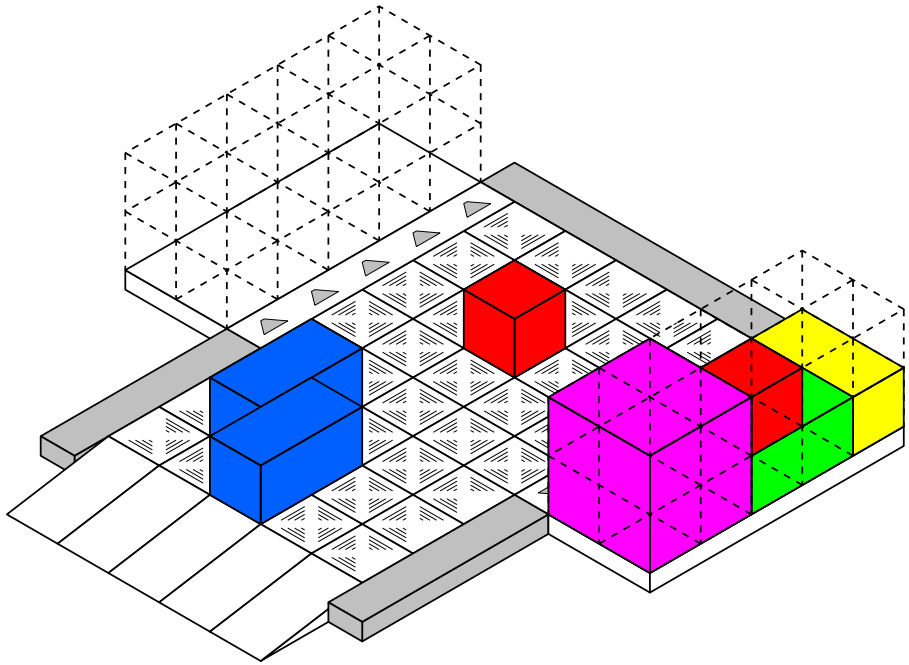
π-sorter/π-composer



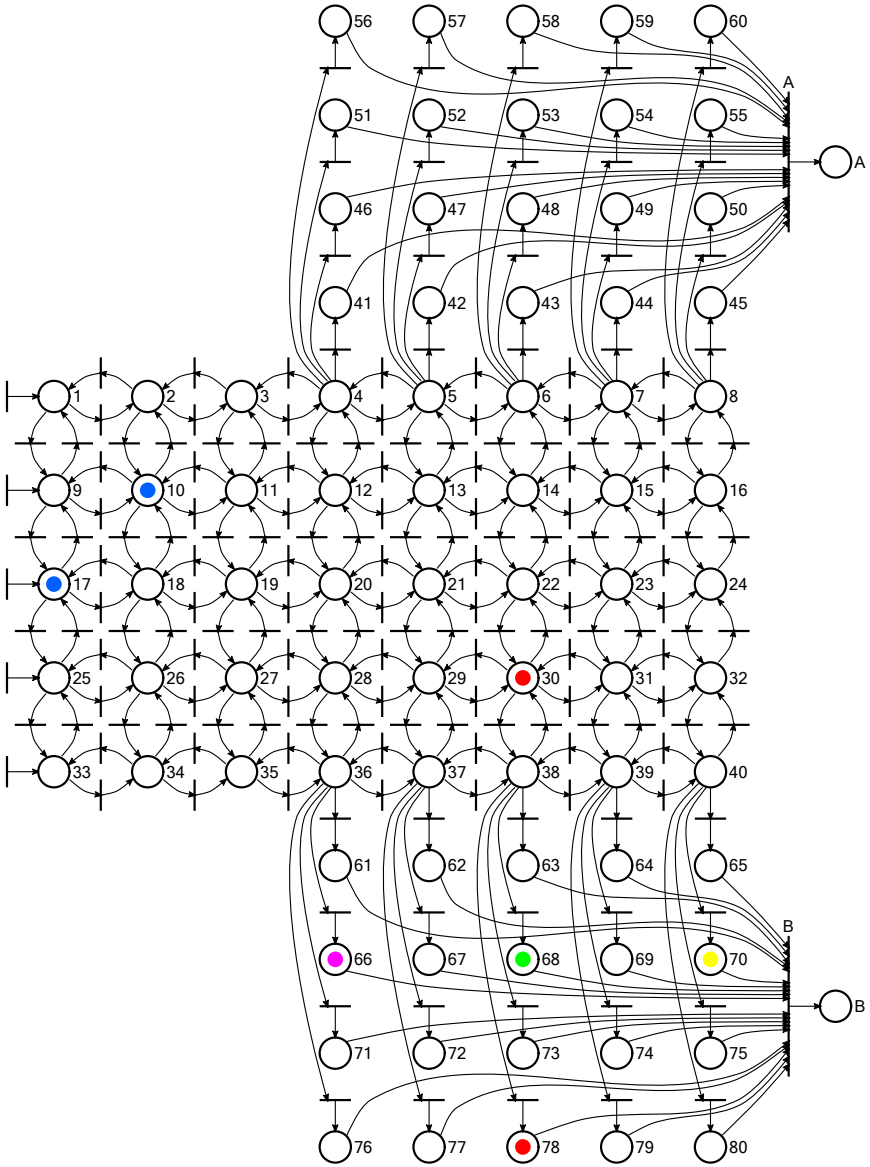


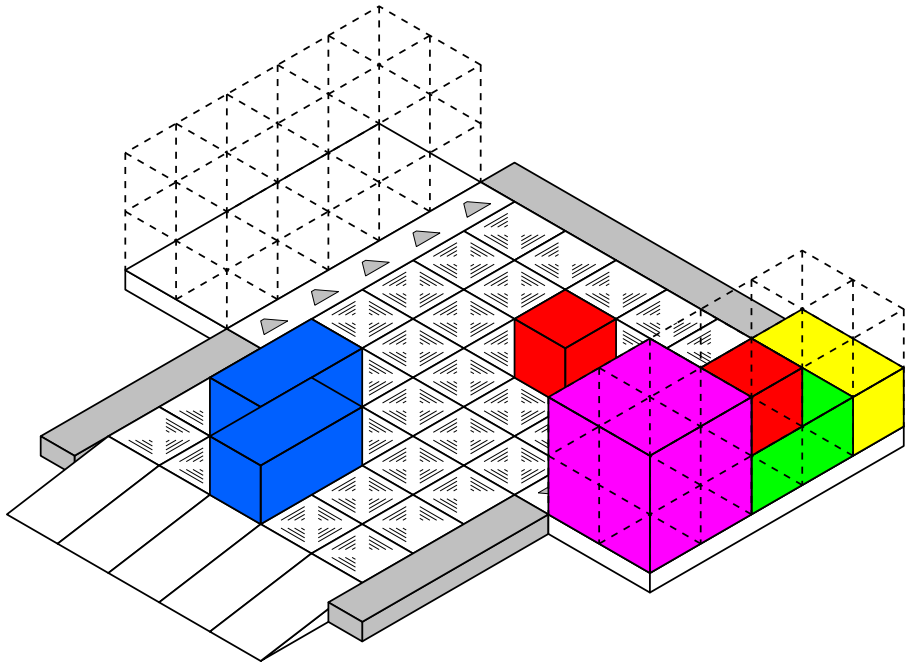
π-sorter/π-composer



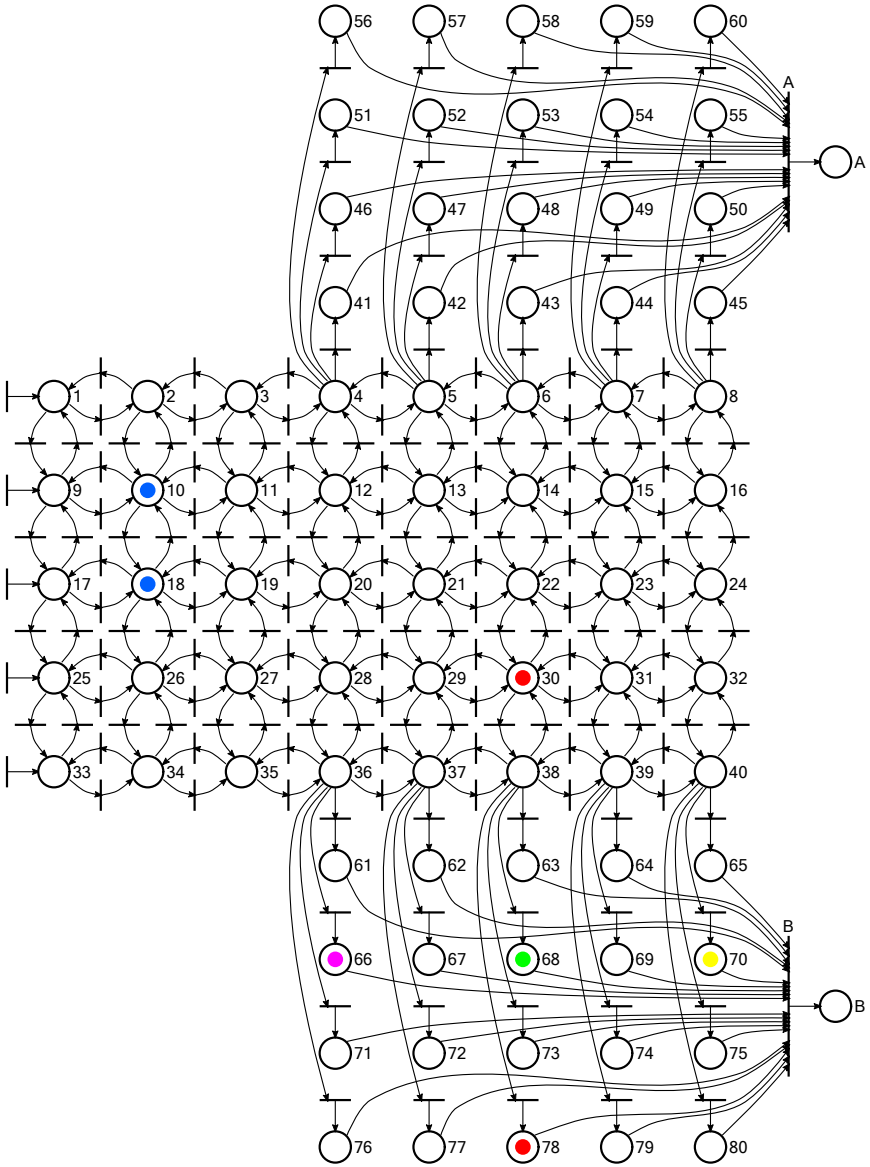


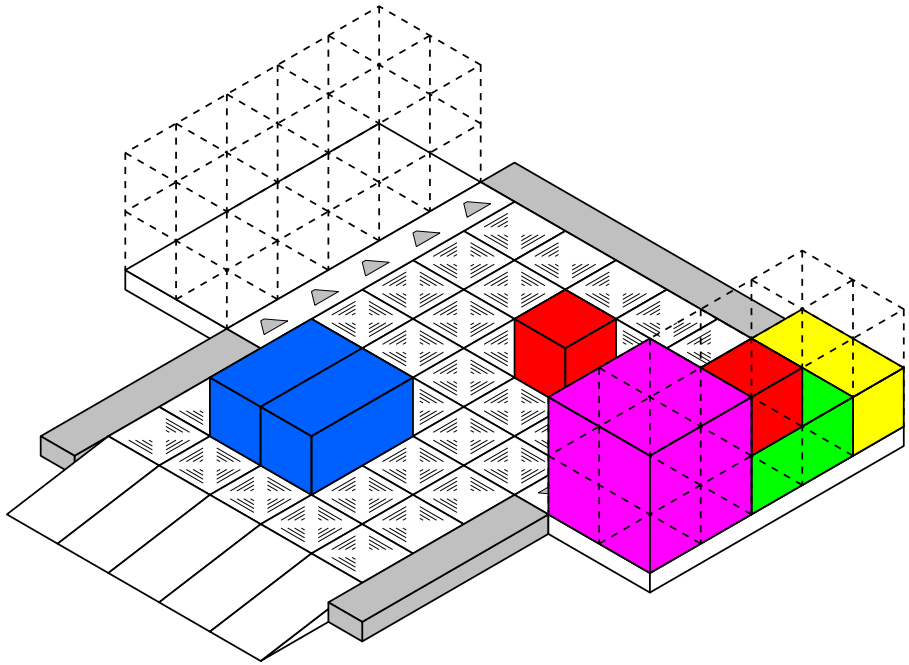
π-sorter/π-composer



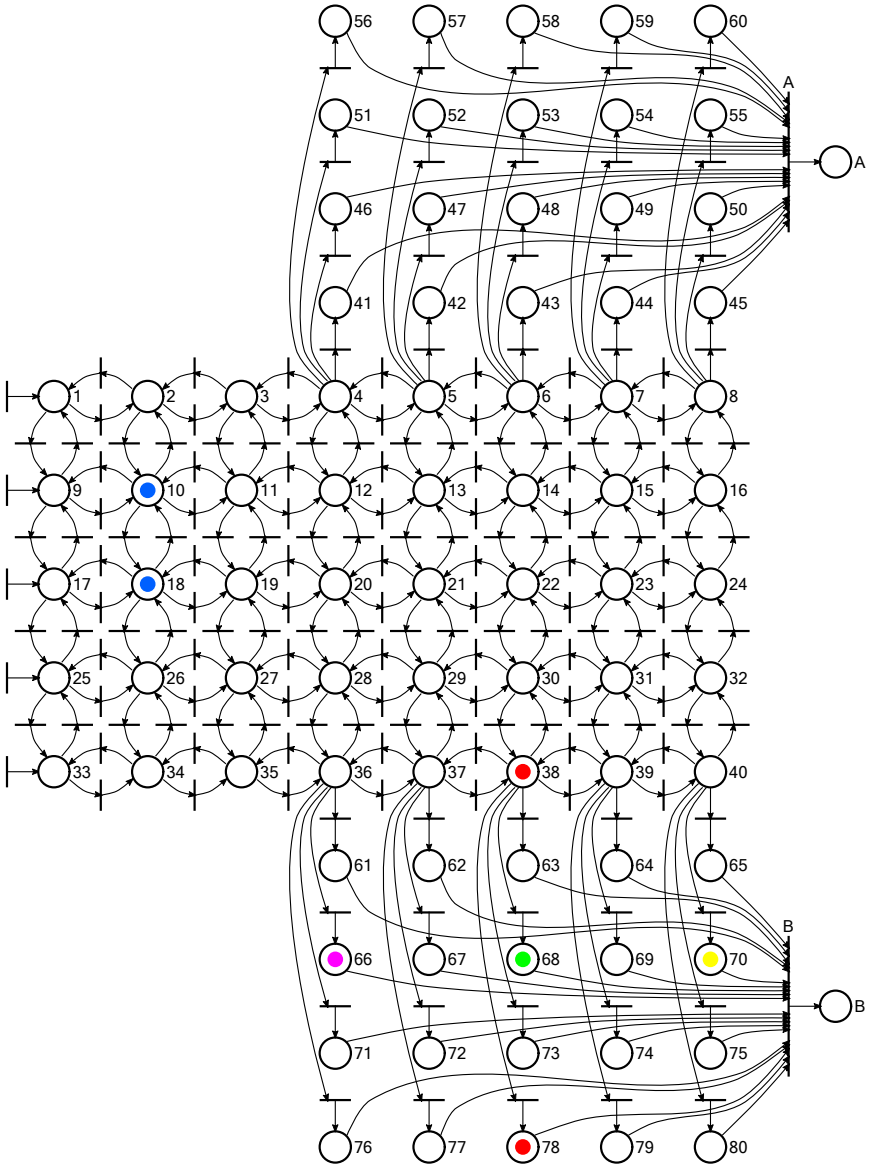


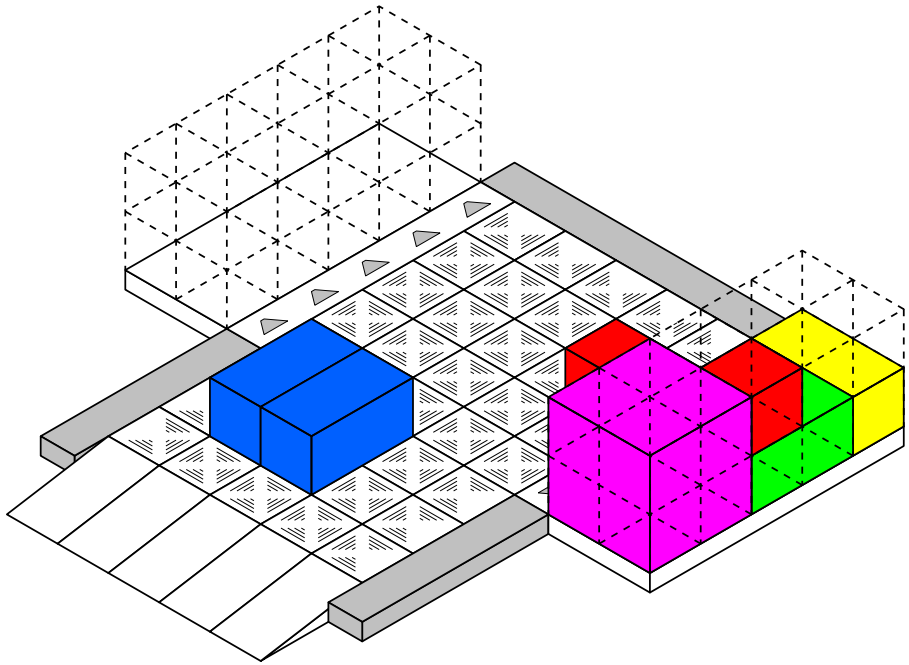
π-sorter/π-composer



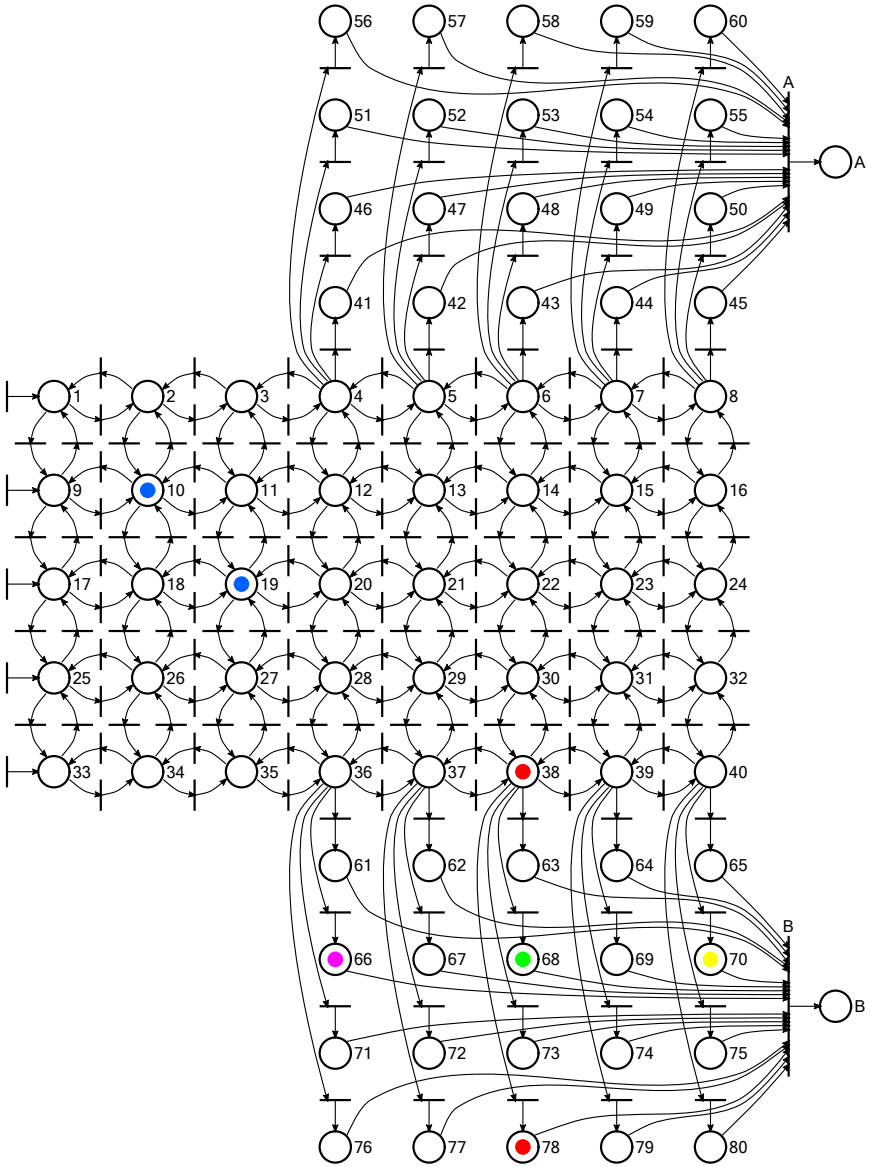


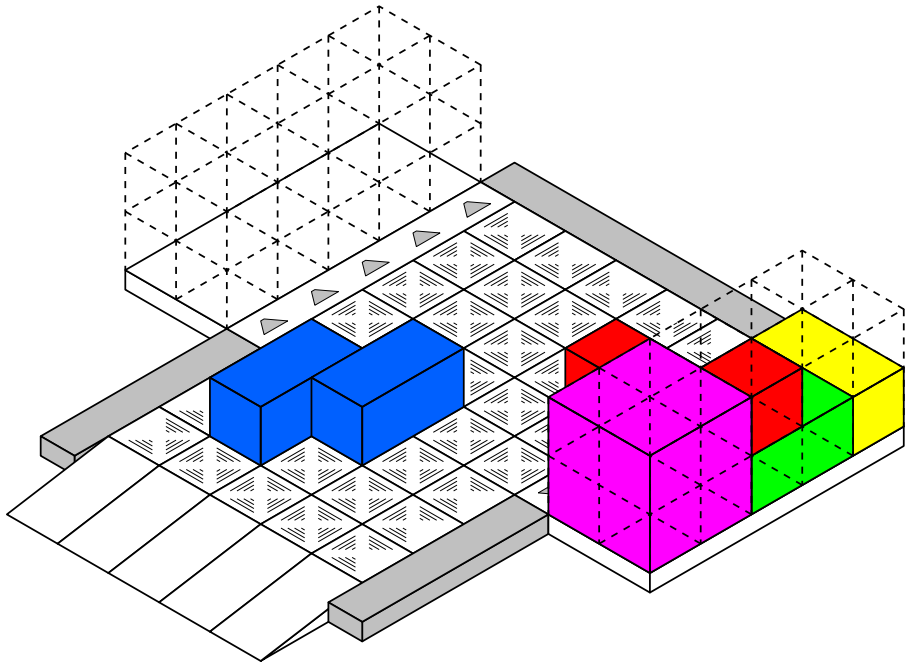
π-sorter/π-composer



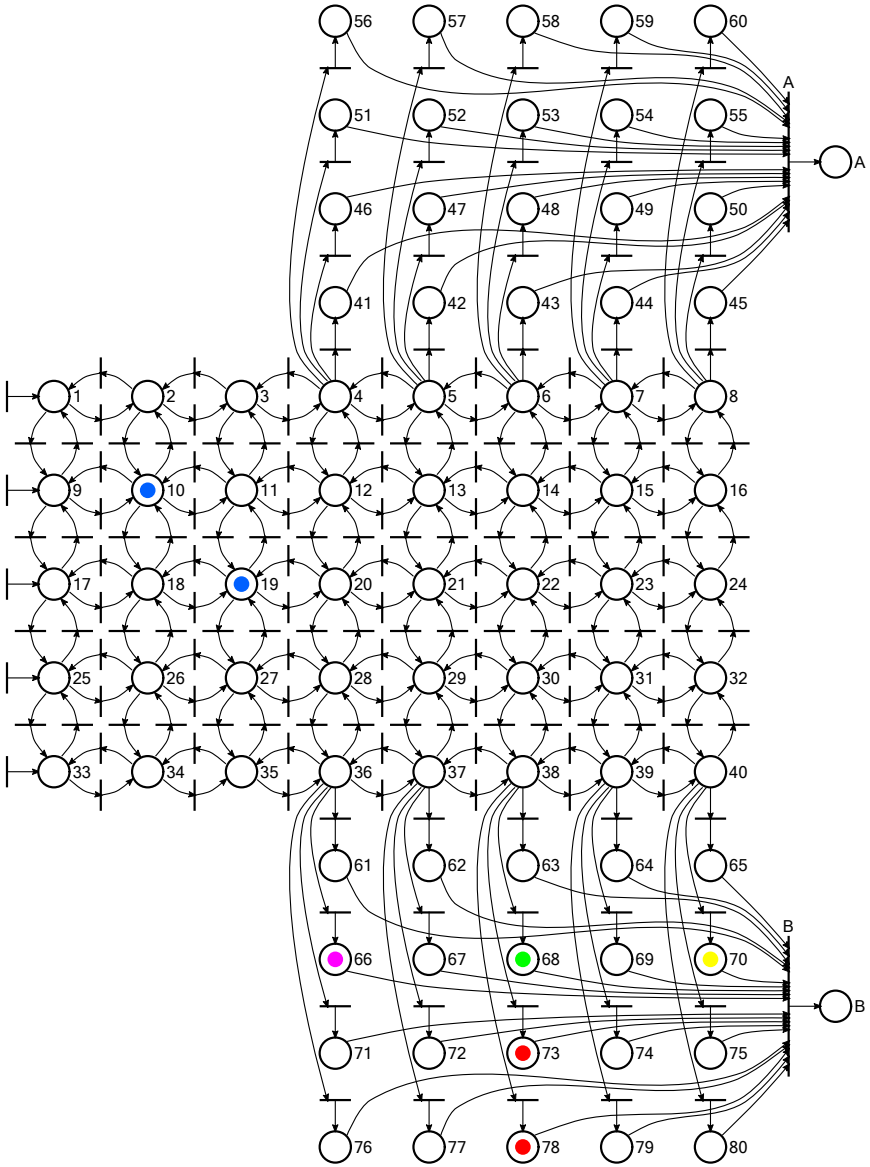


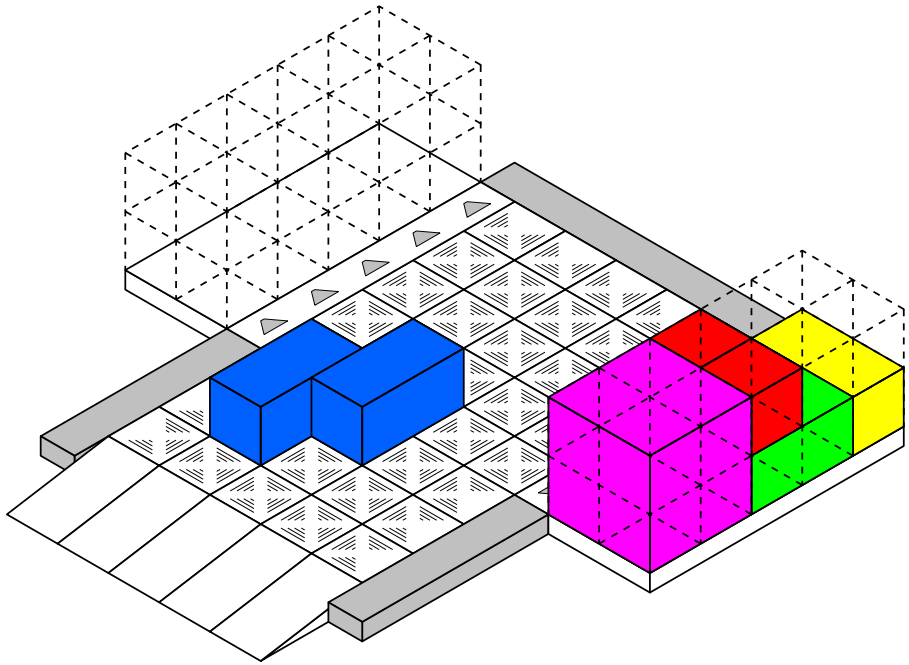
π-sorter/π-composer



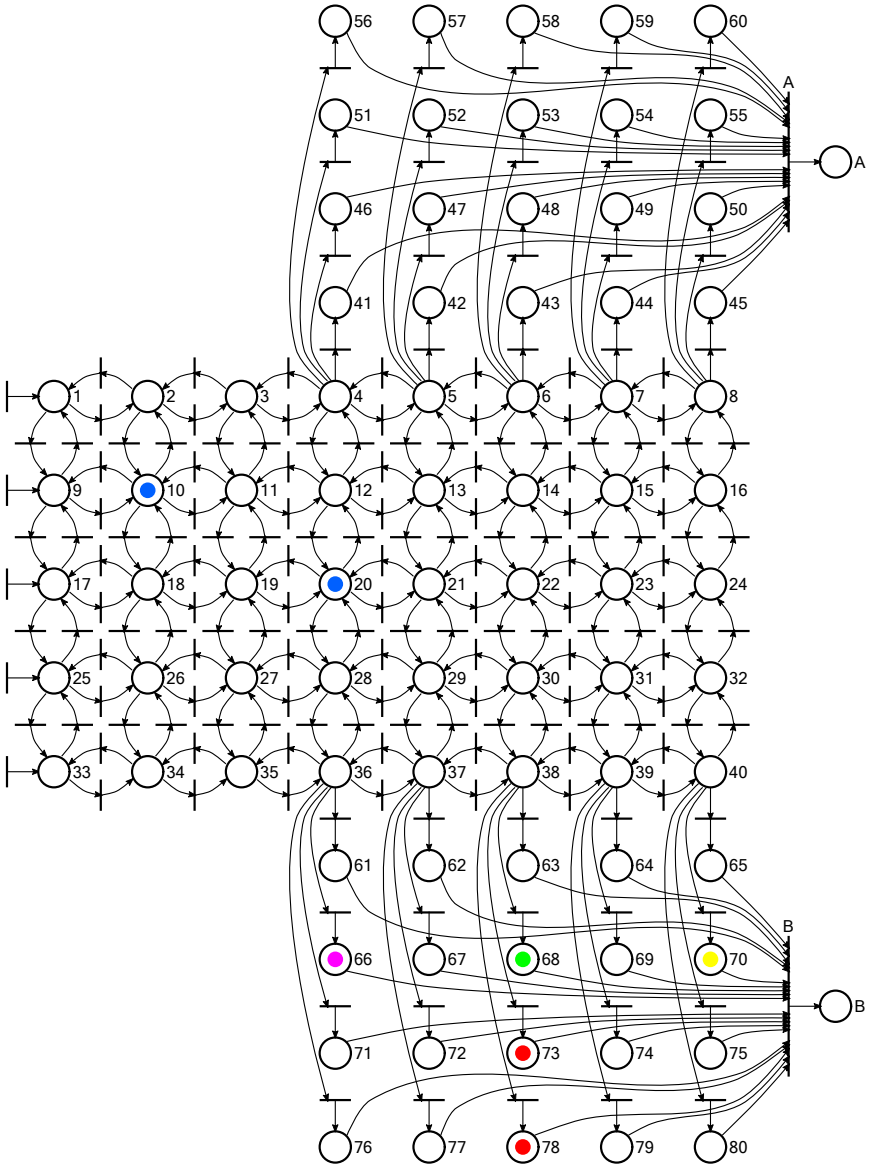


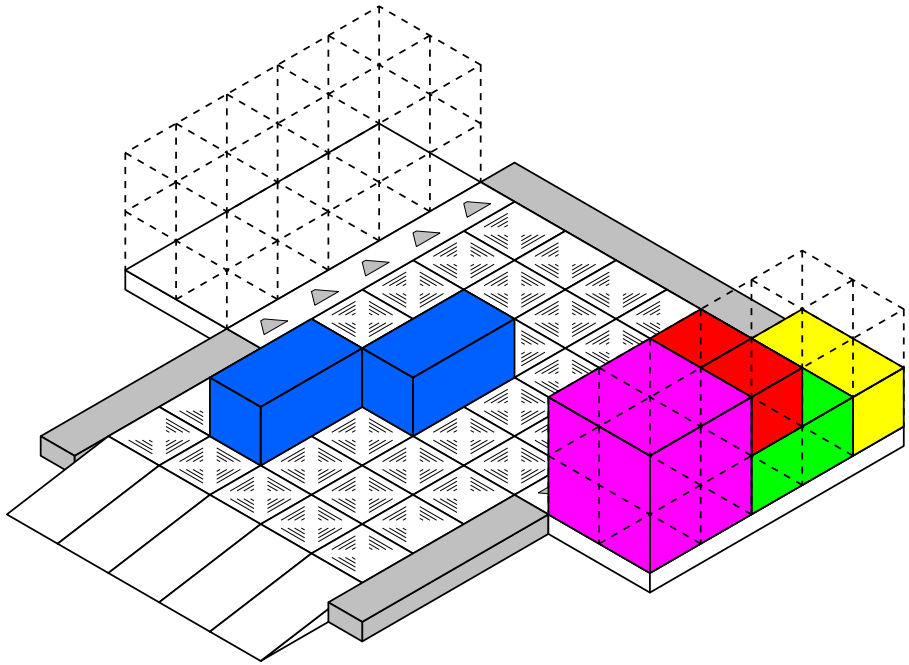
π-sorter/π-composer



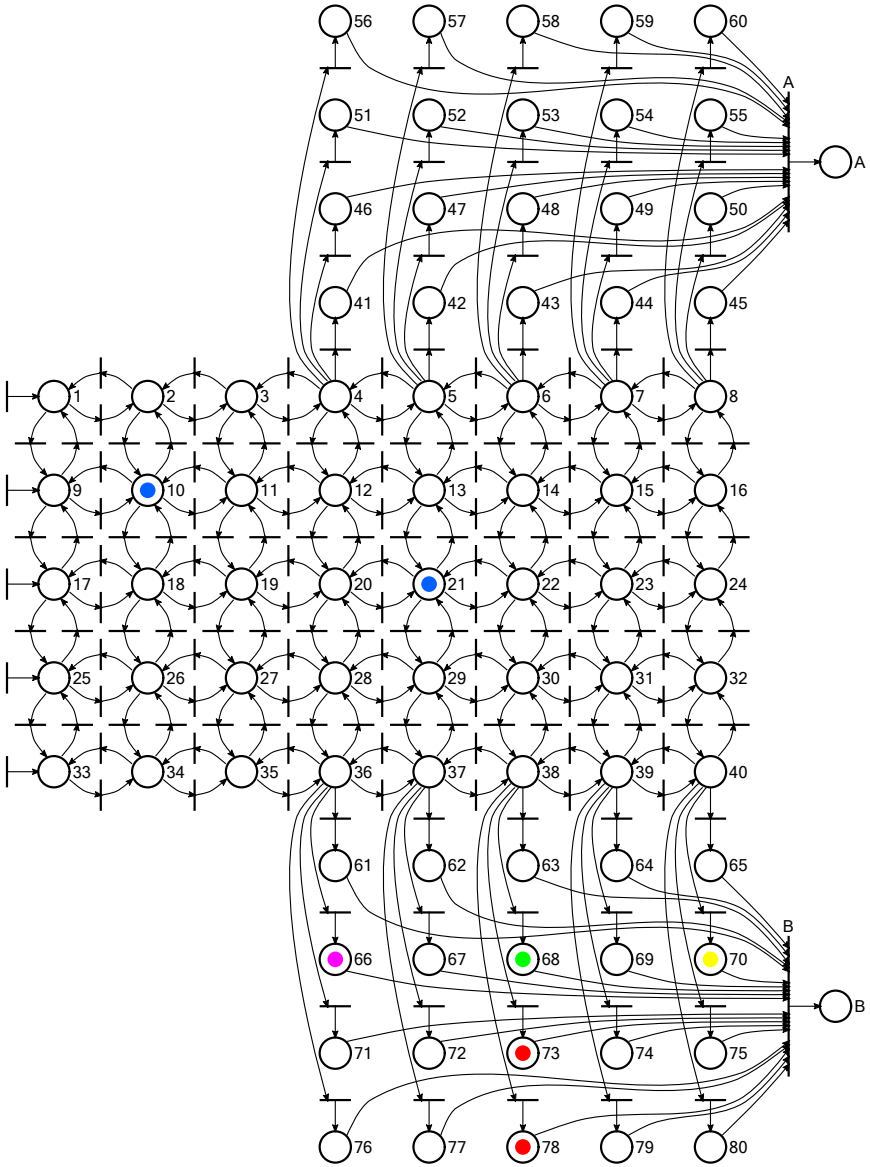


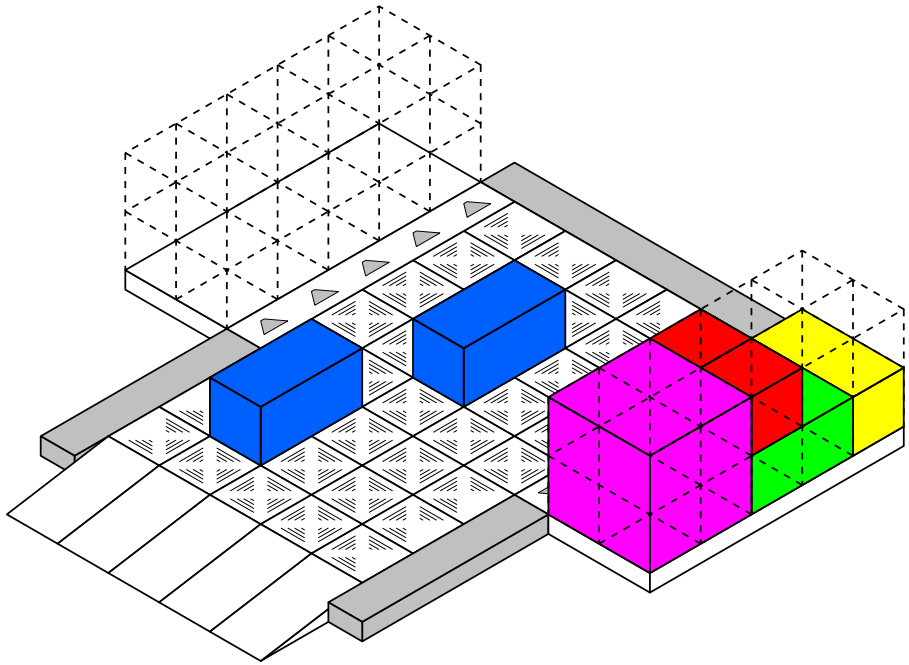
π-sorter/π-composer



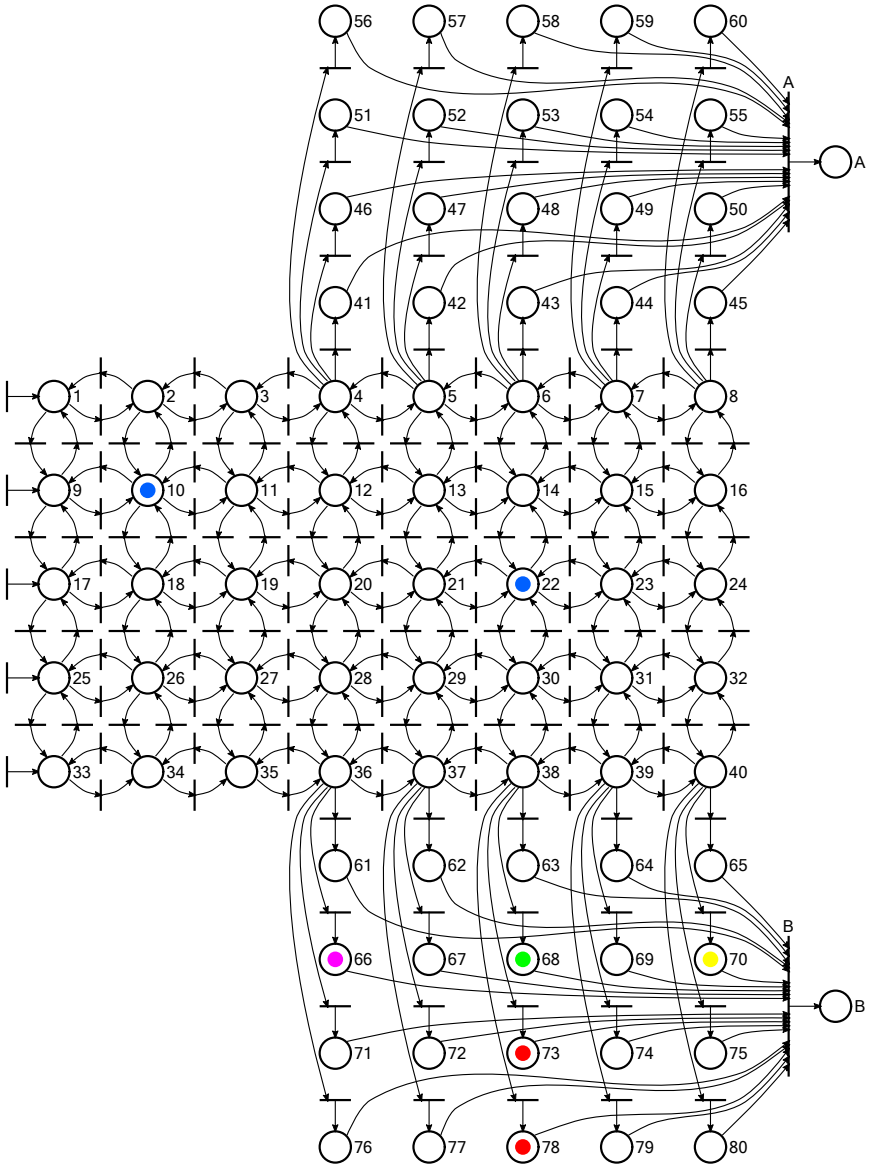


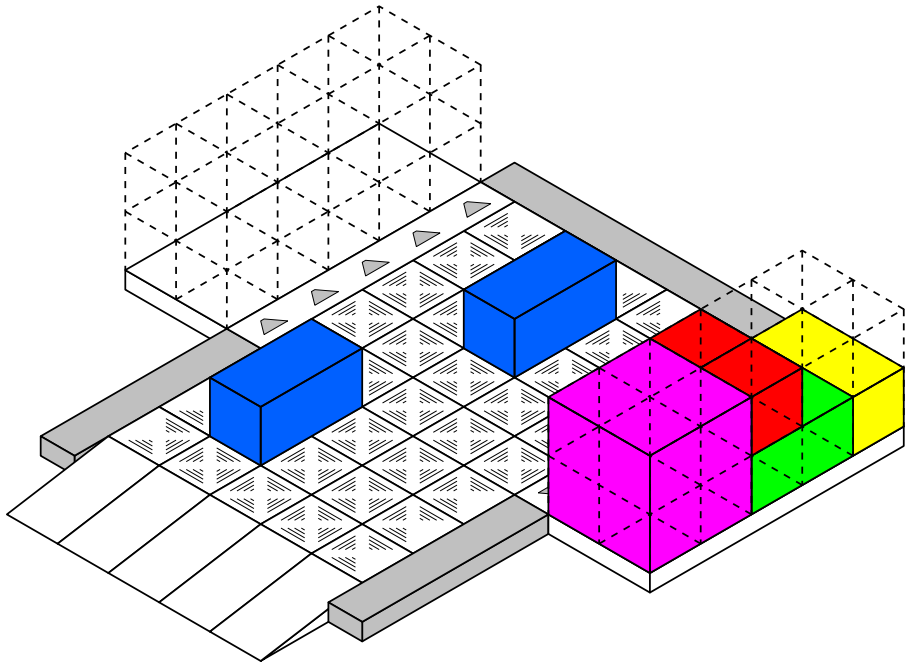
π-sorter/π-composer



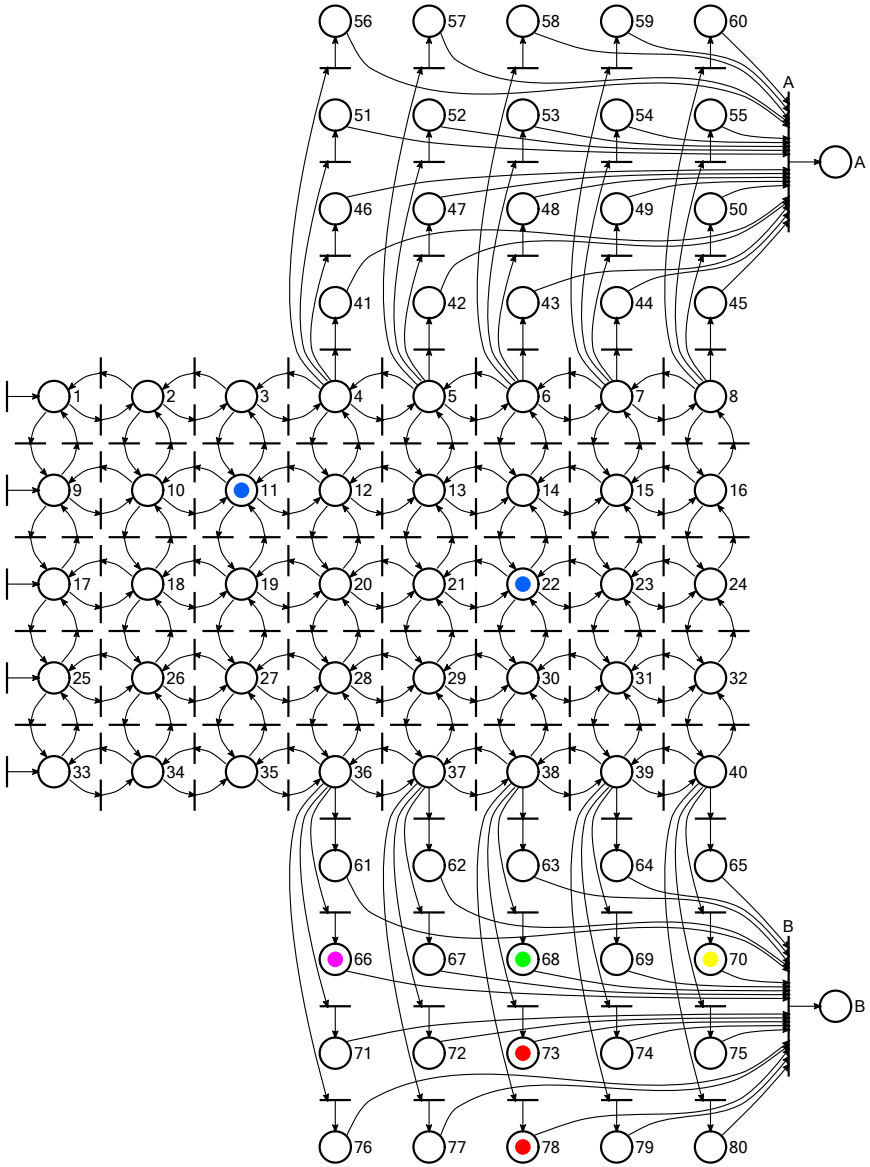


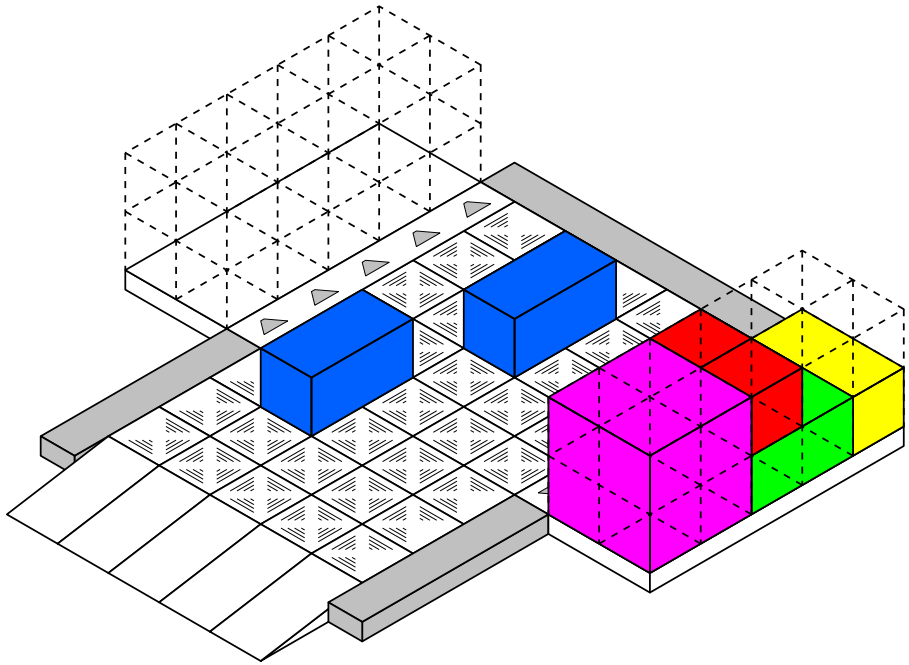
π-sorter/π-composer



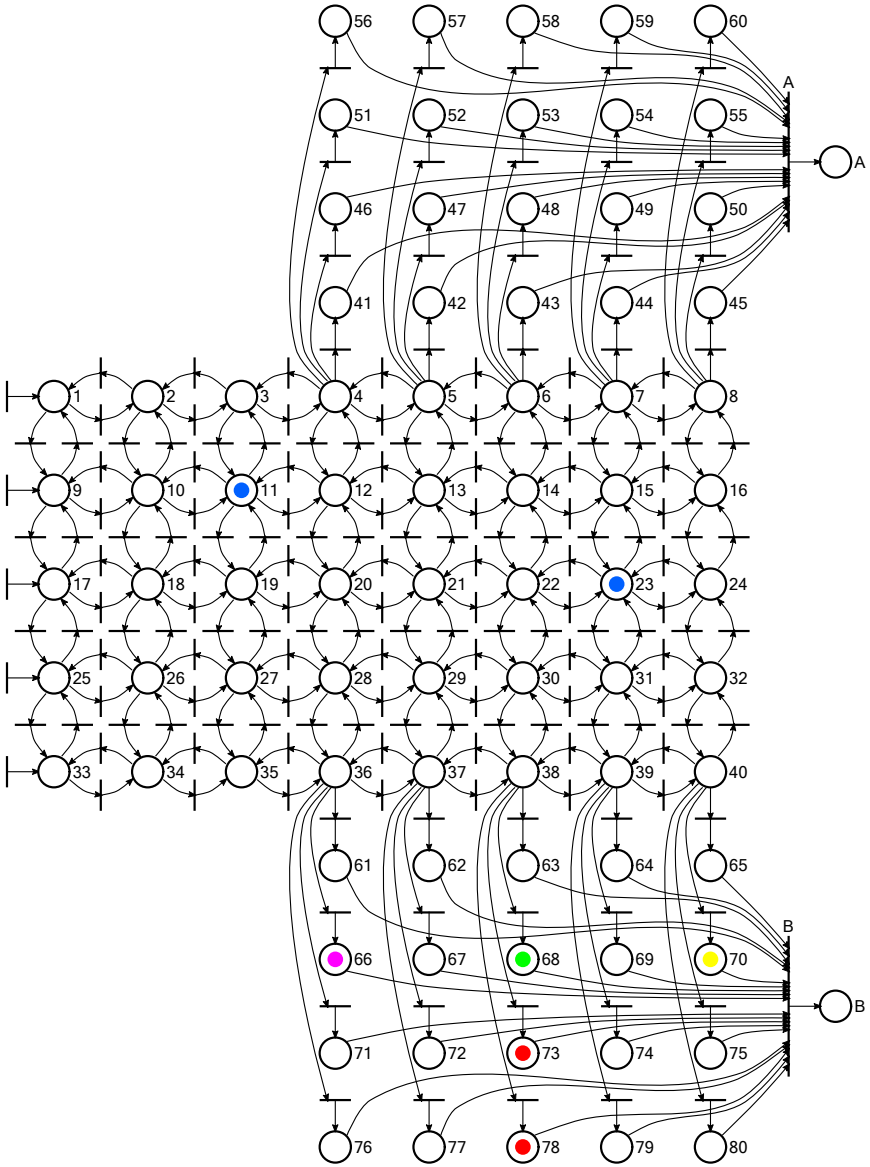


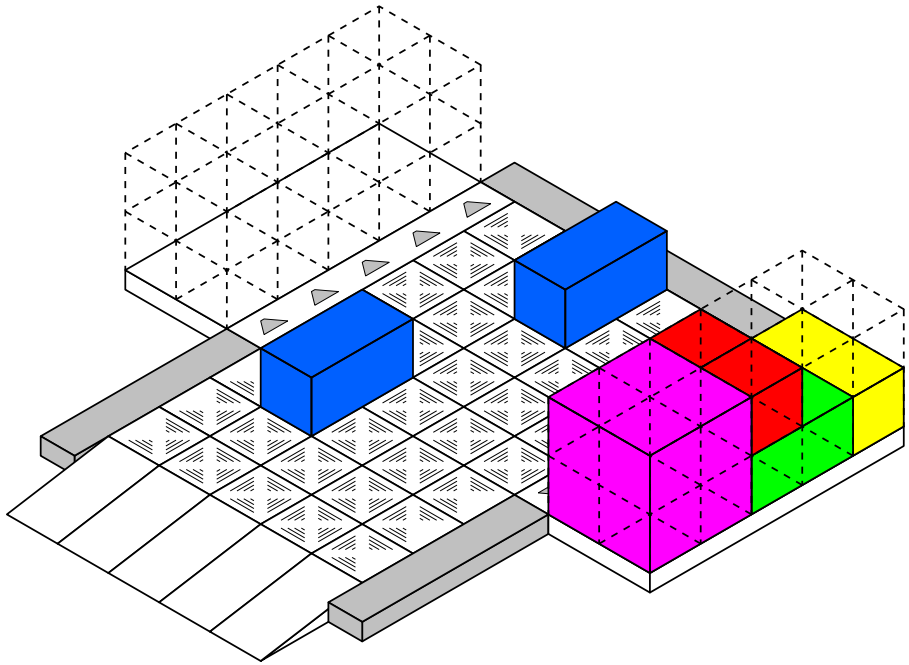
π-sorter/π-composer



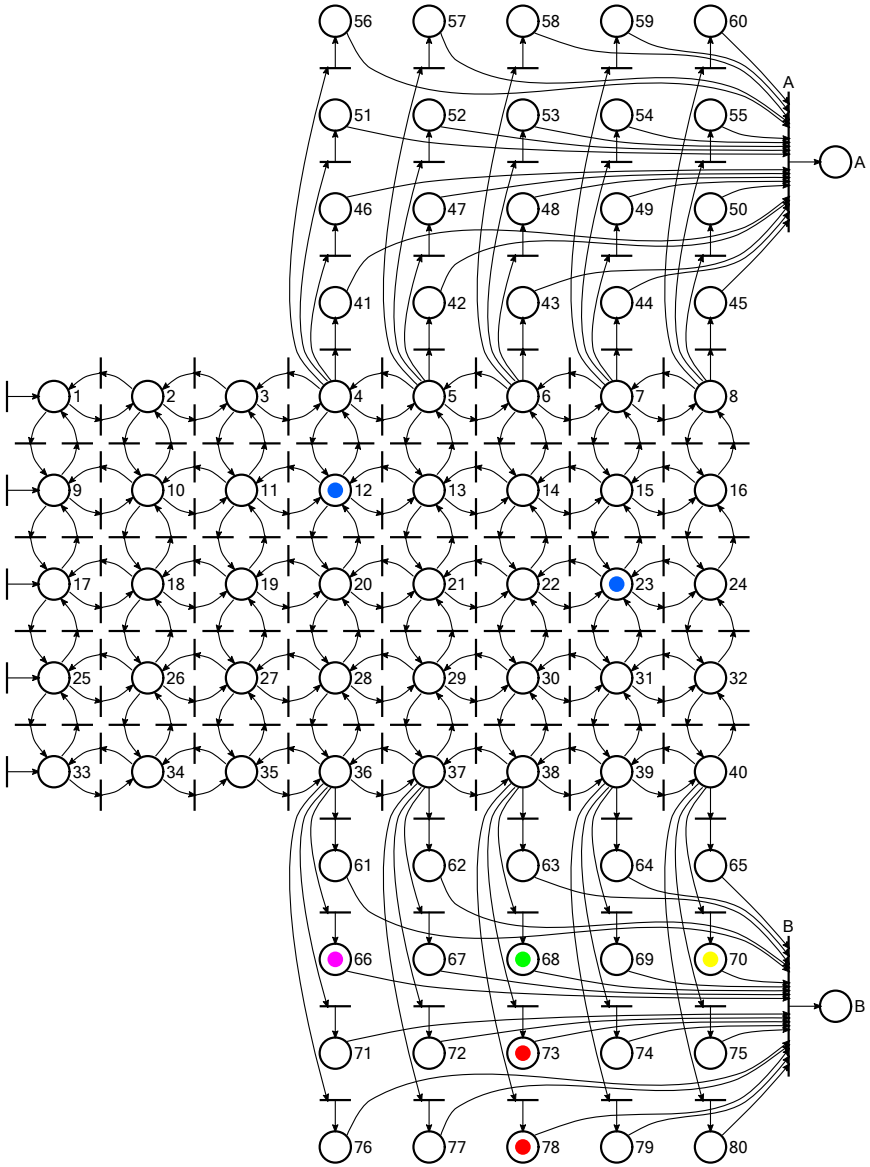


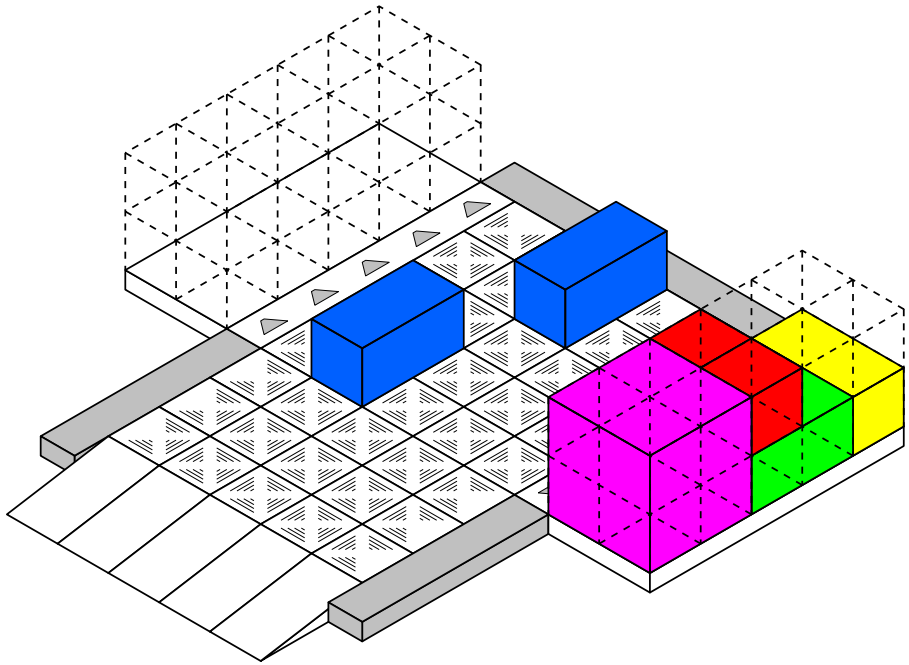
π-sorter/π-composer



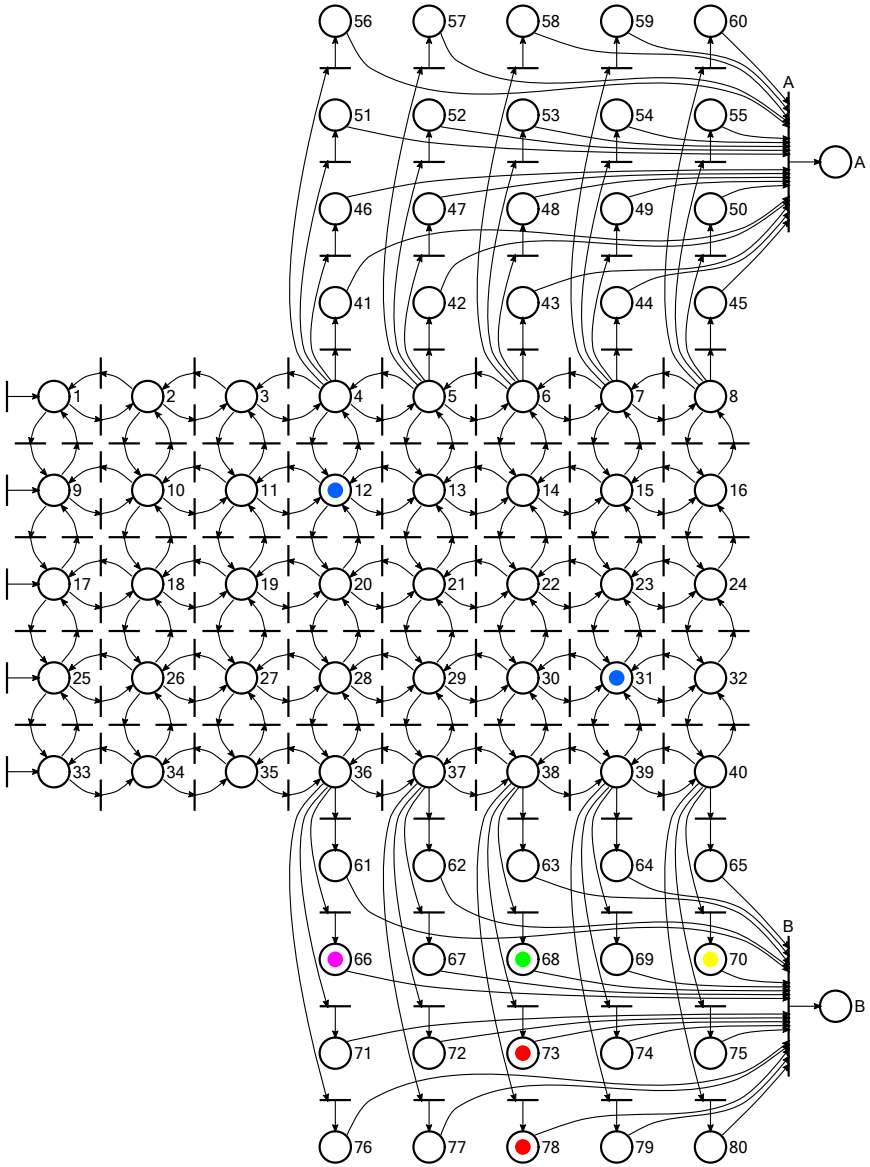


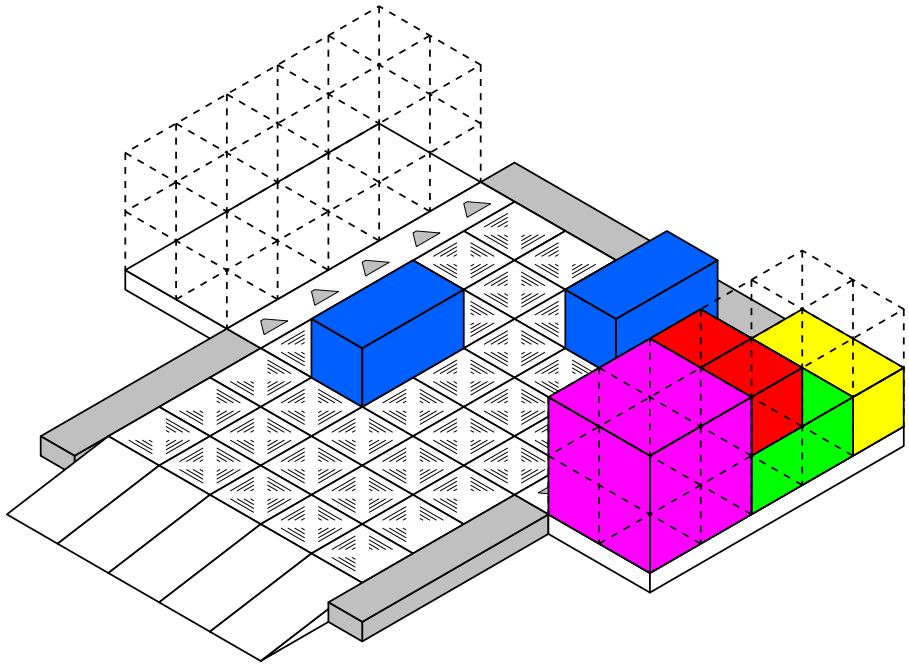
π-sorter/π-composer



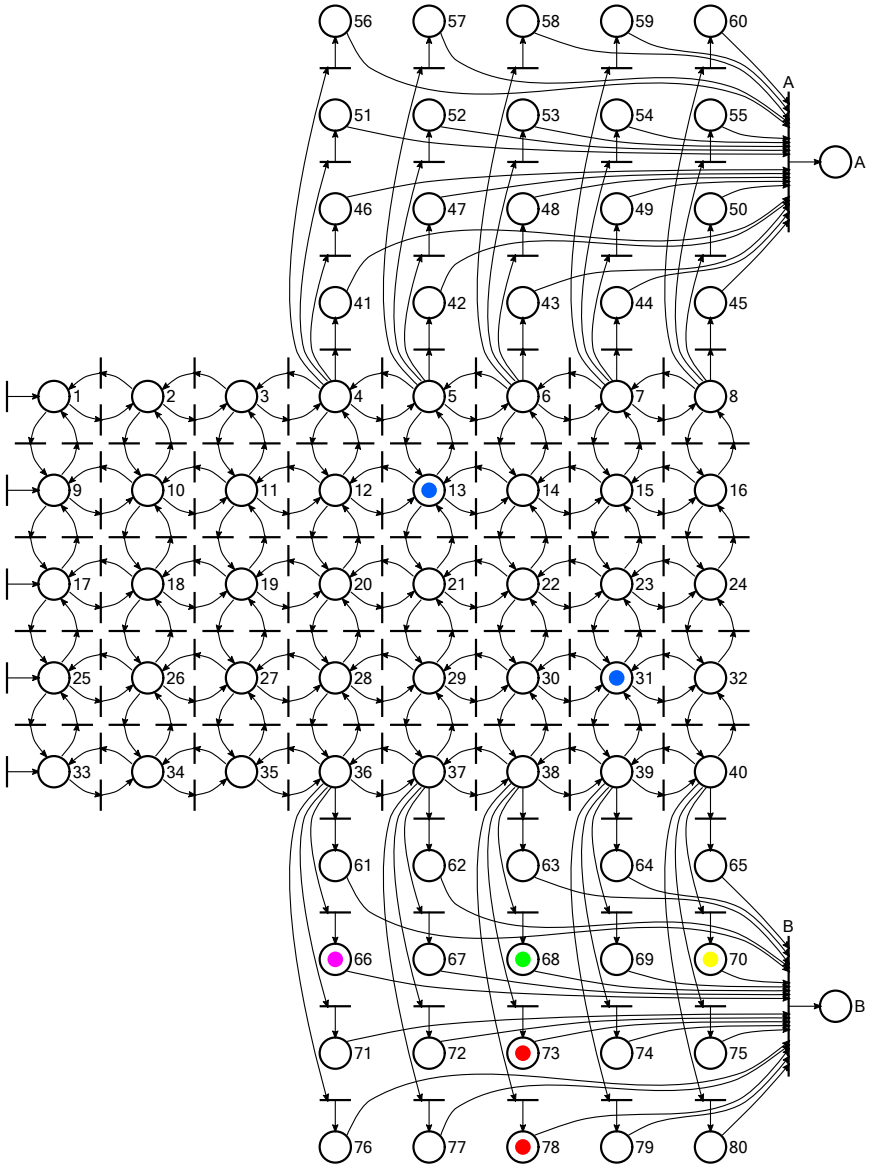


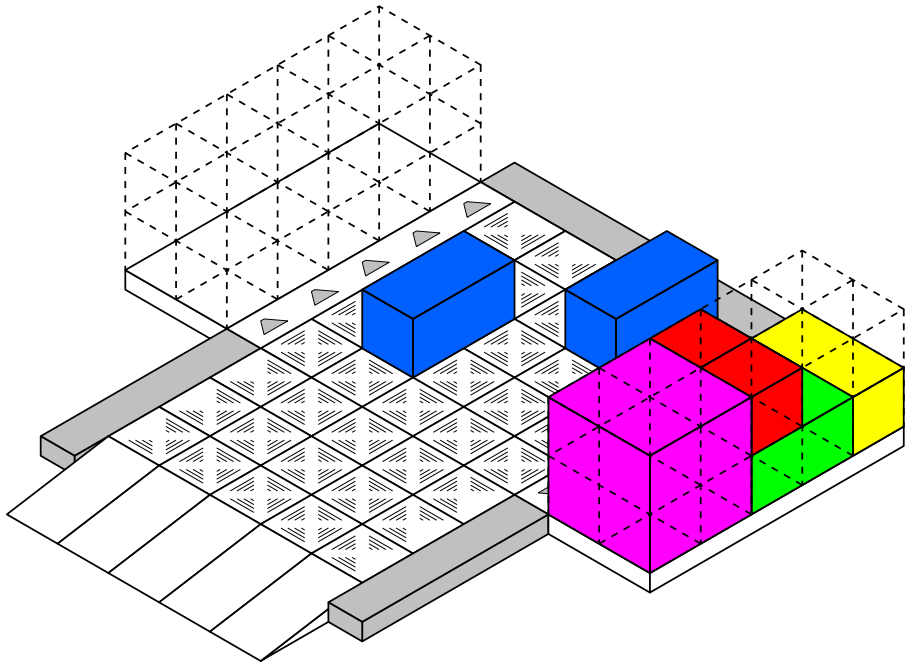
π-sorter/π-composer



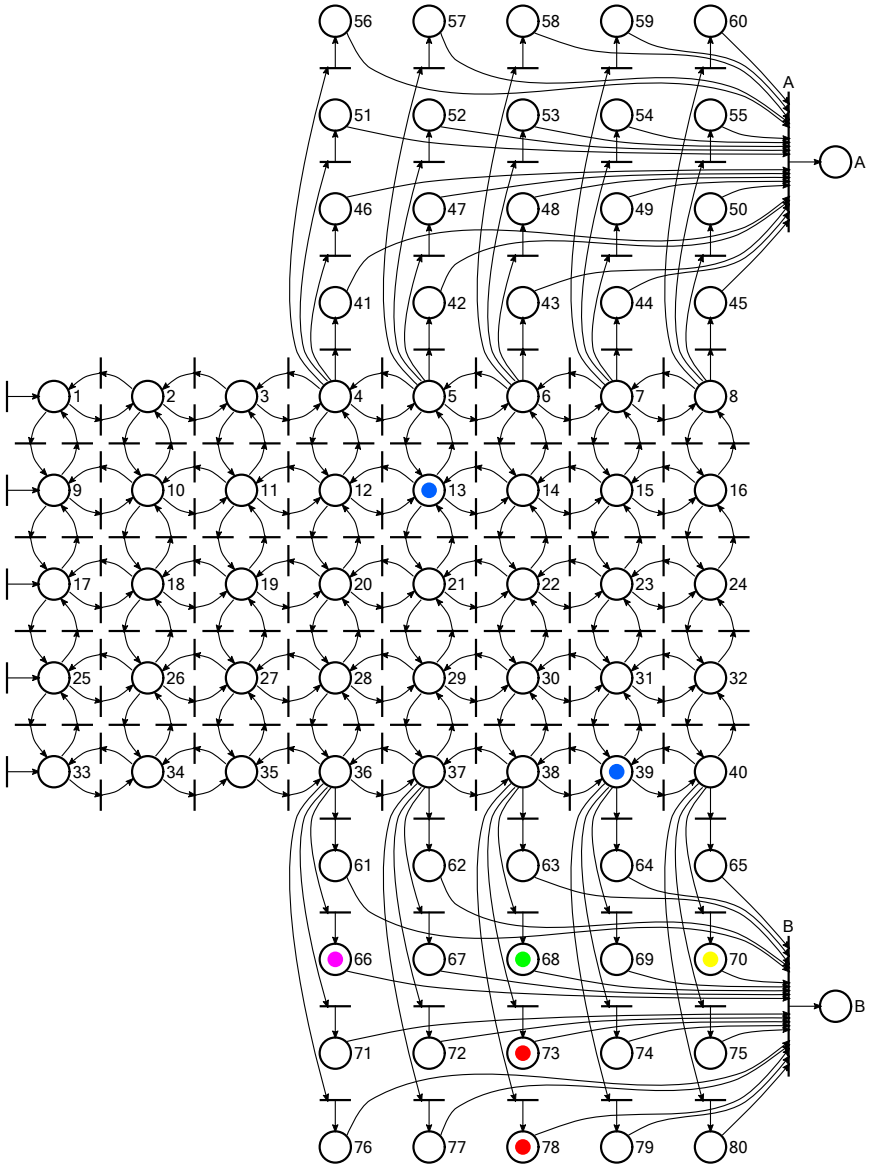


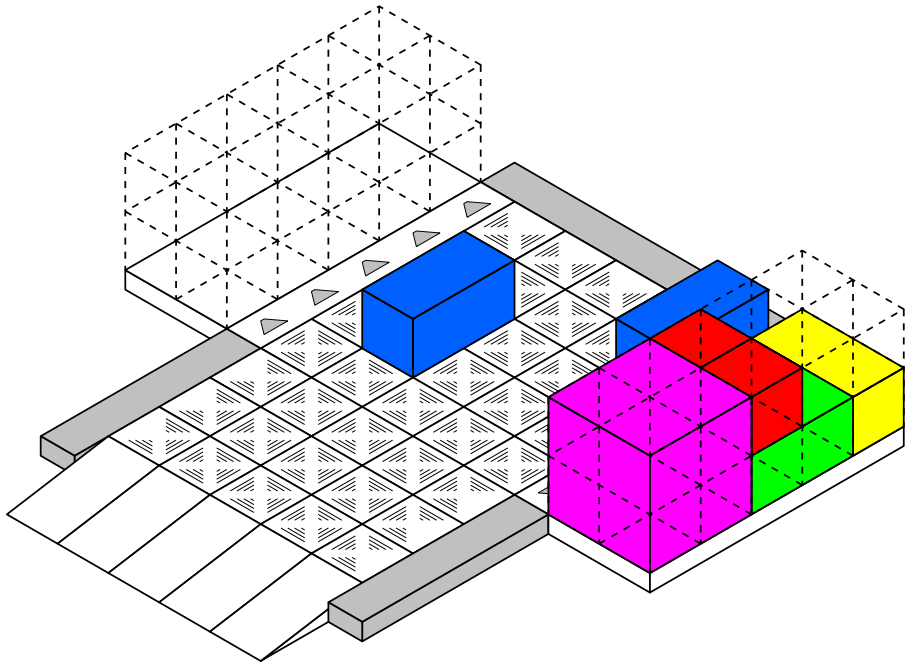
π-sorter/π-composer



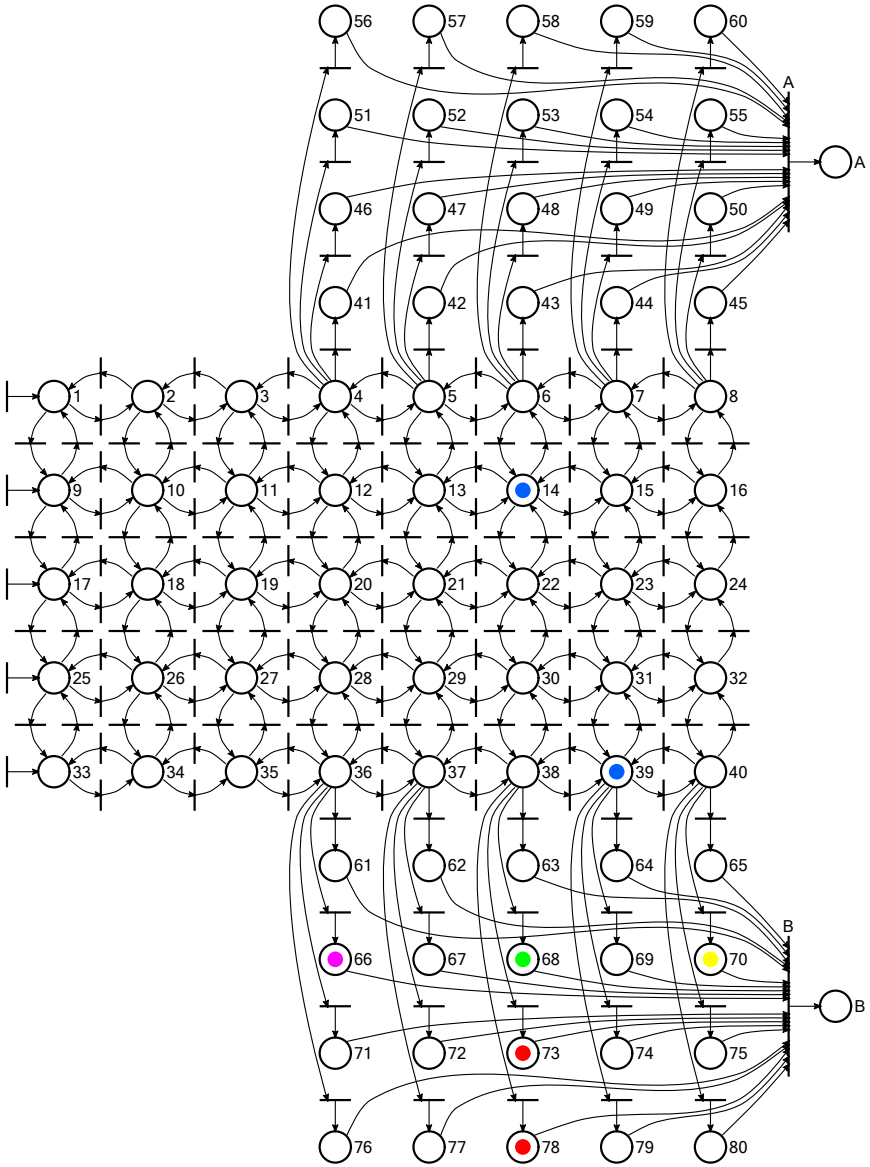


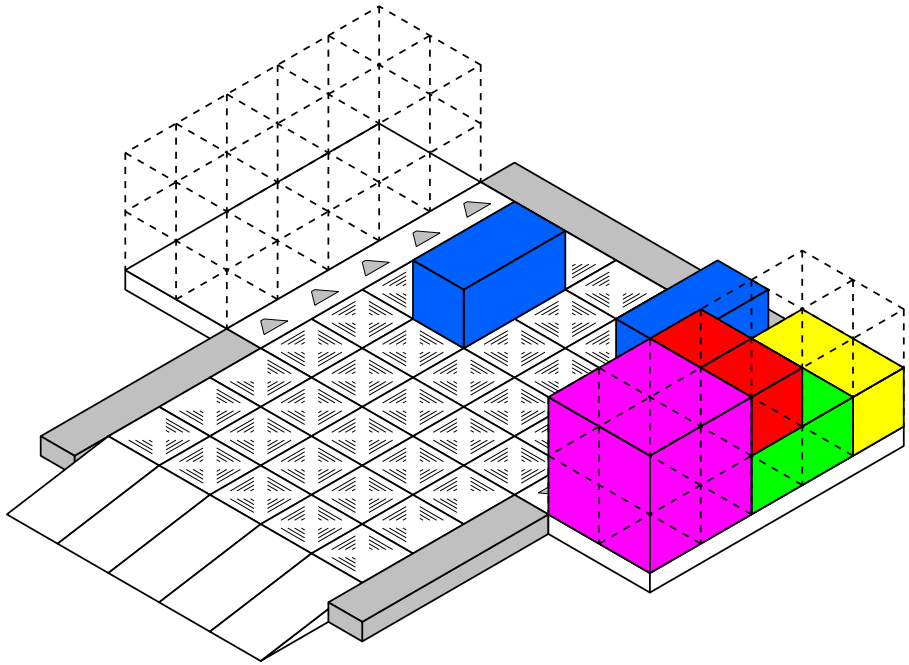
π-sorter/π-composer



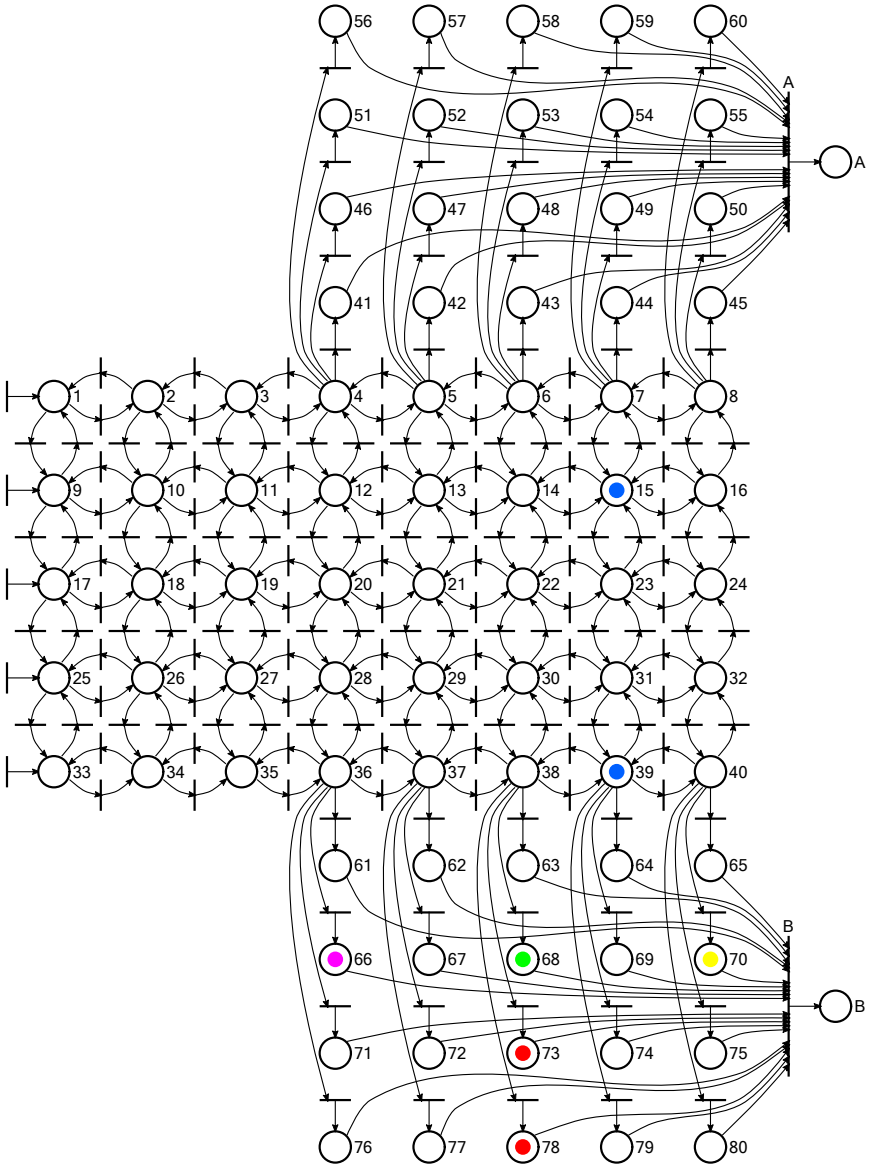


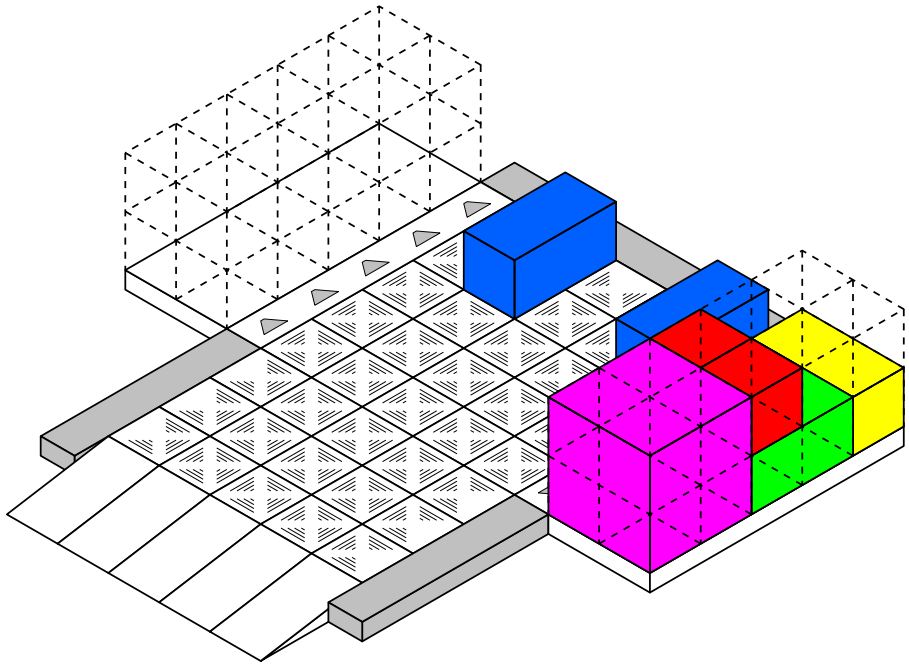
π-sorter/π-composer



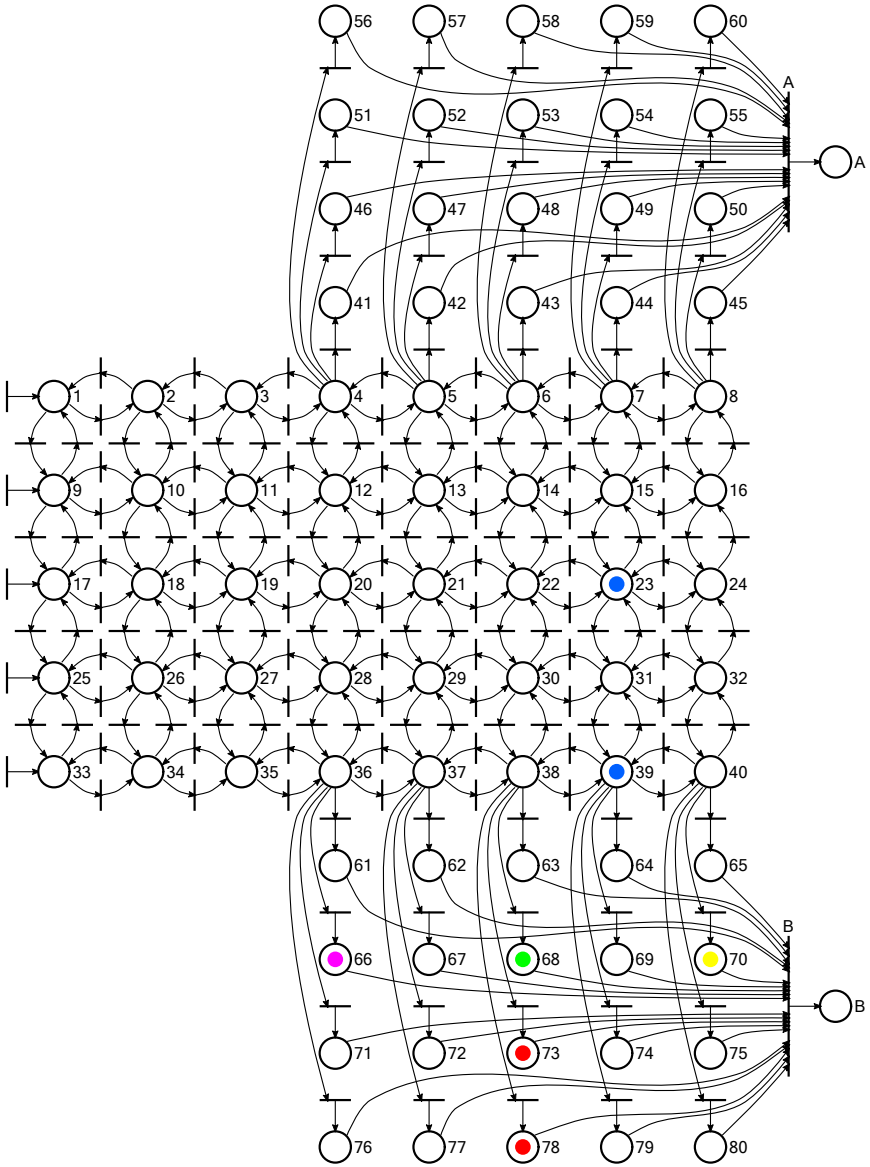


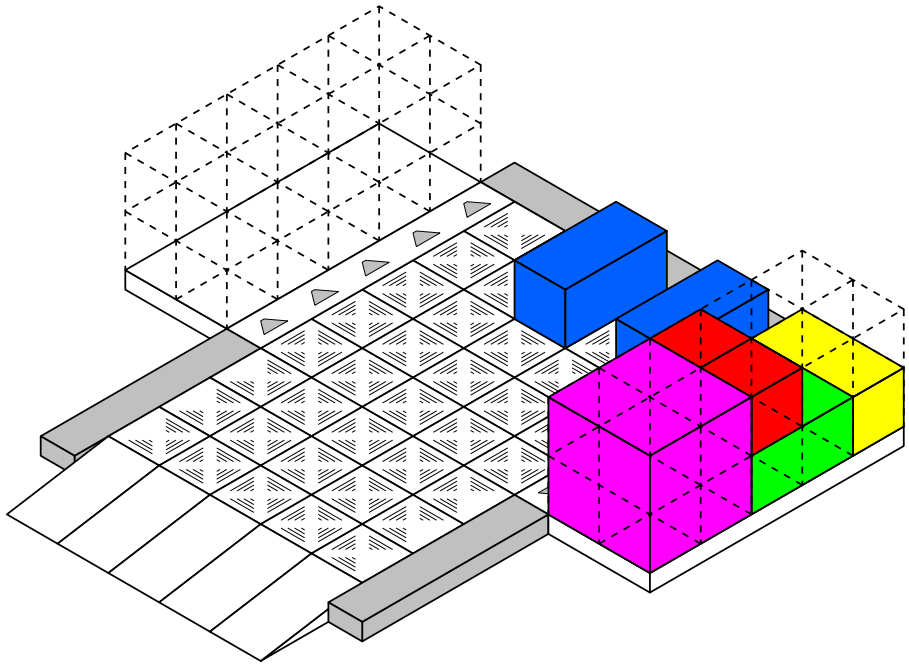
π-sorter/π-composer



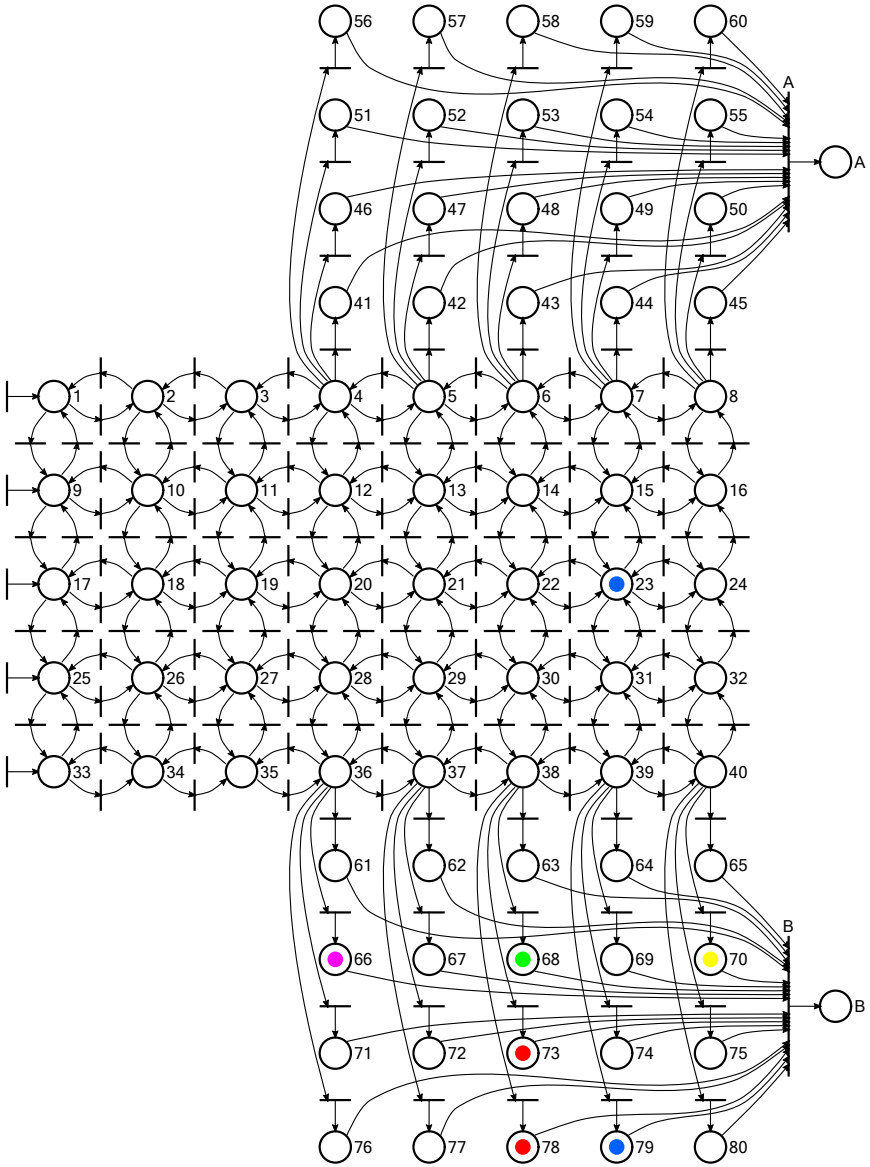


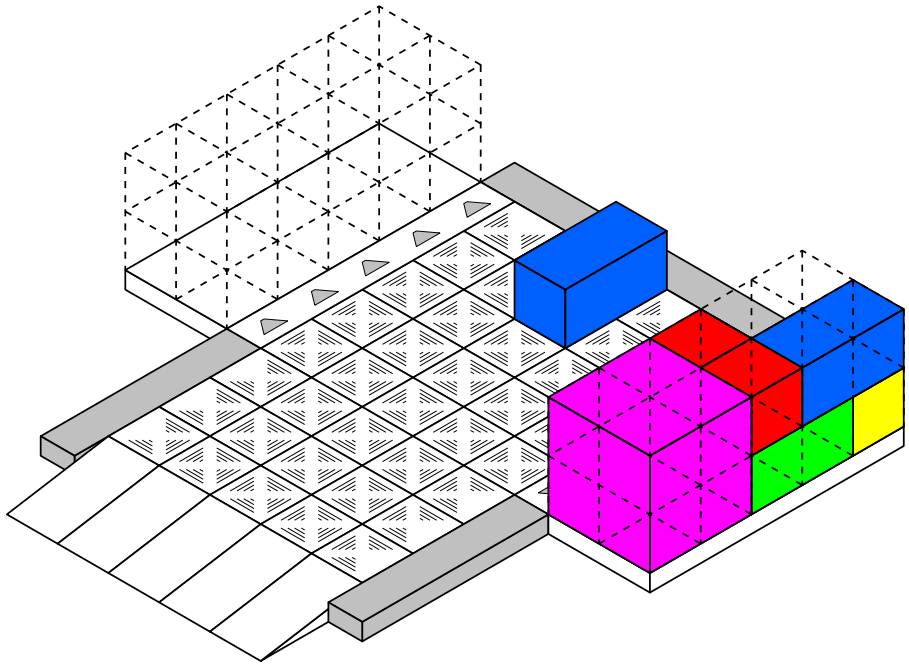
π-sorter/π-composer



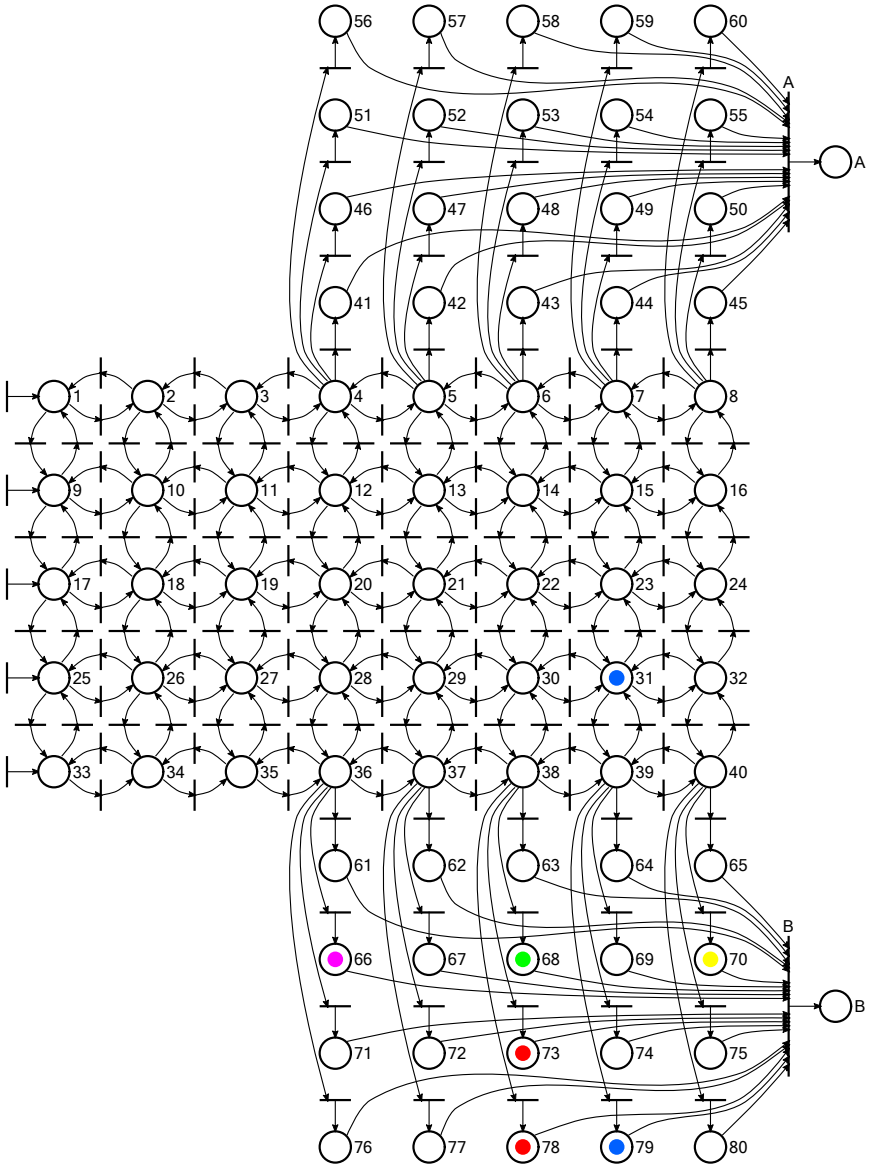


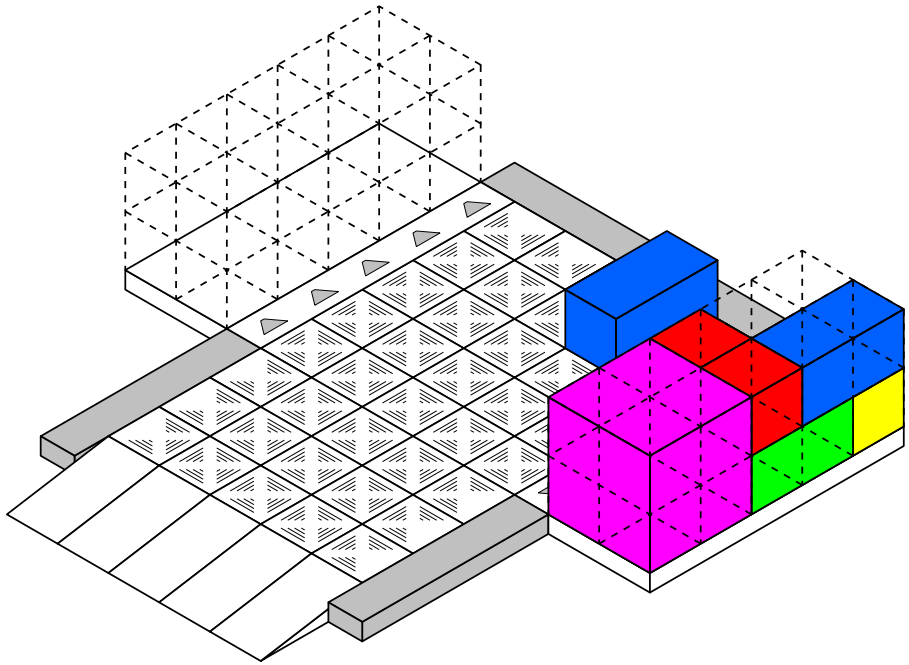
π-sorter/π-composer



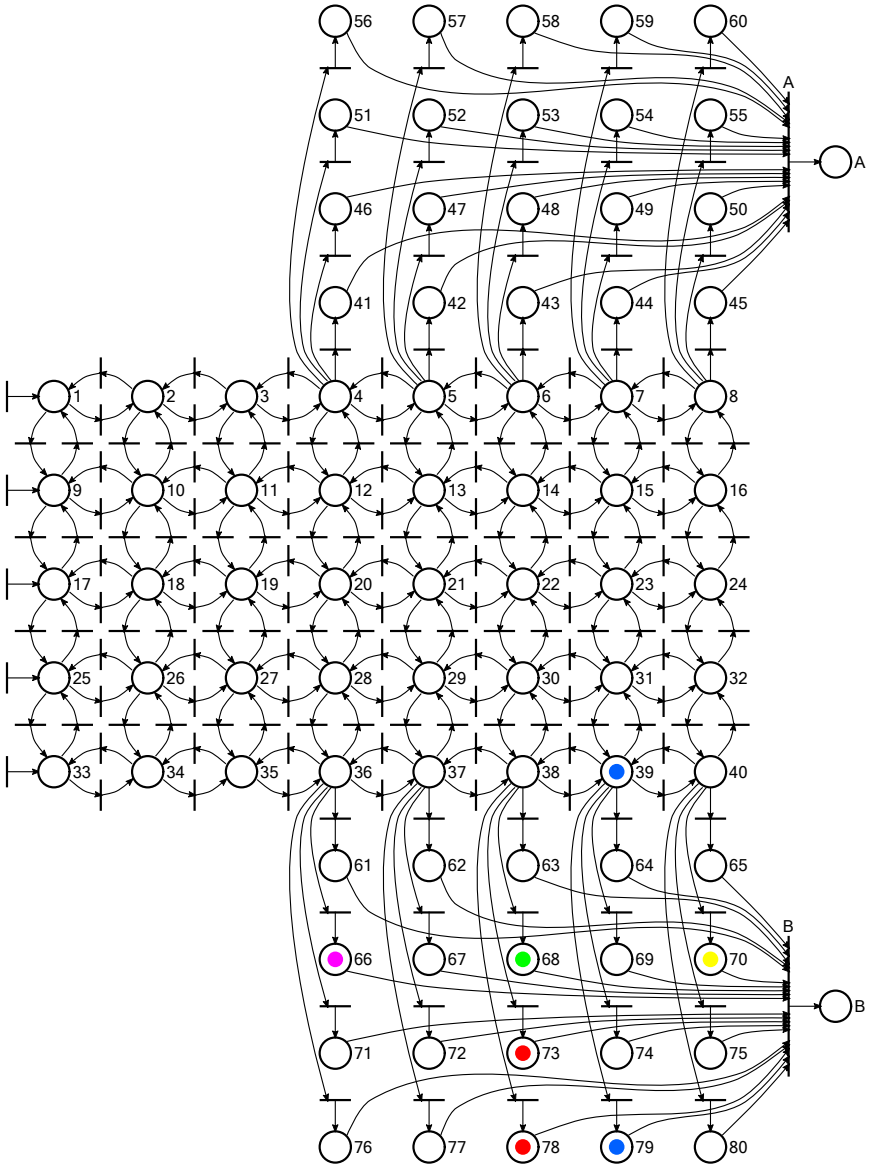


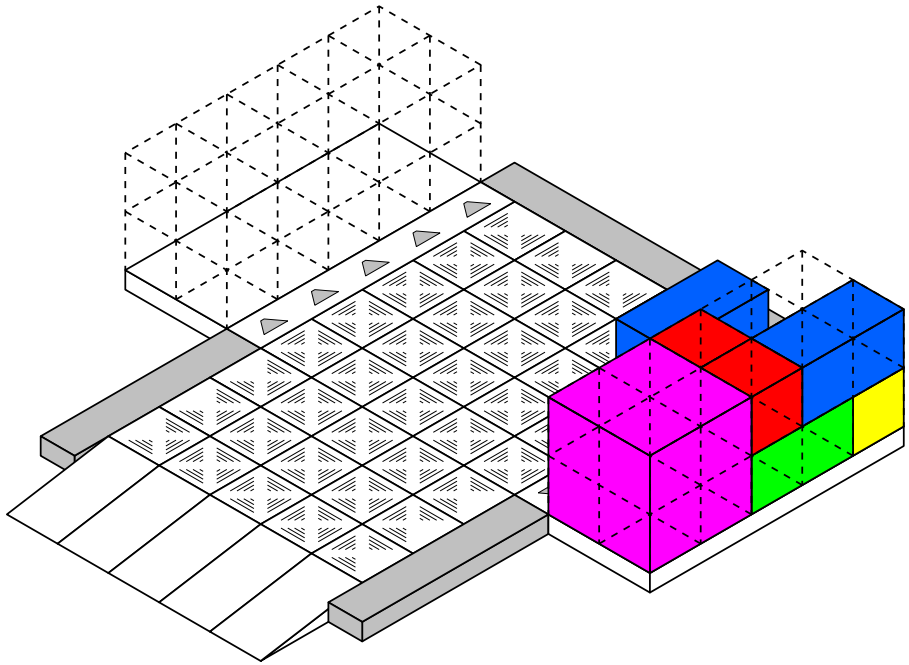
π-sorter/π-composer



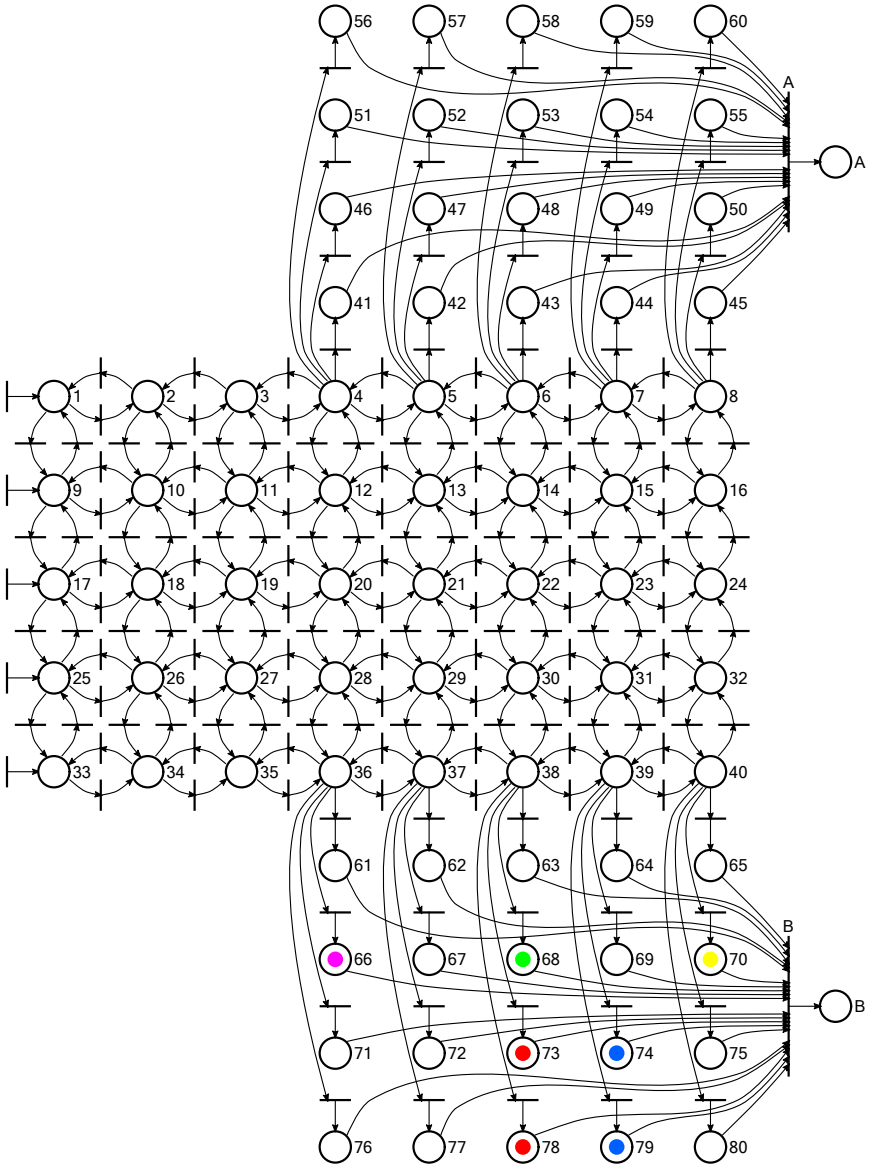


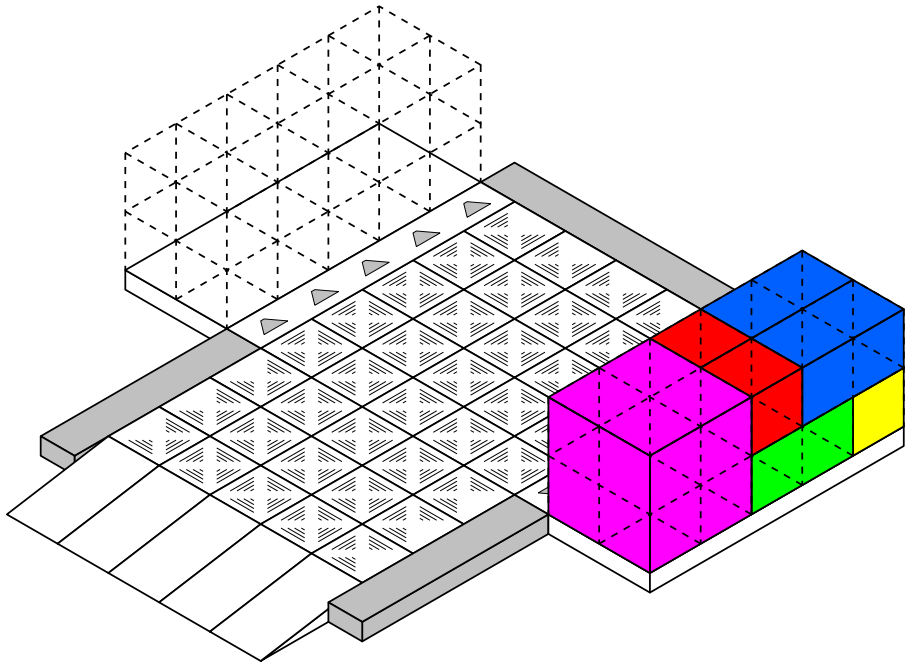
π-sorter/π-composer



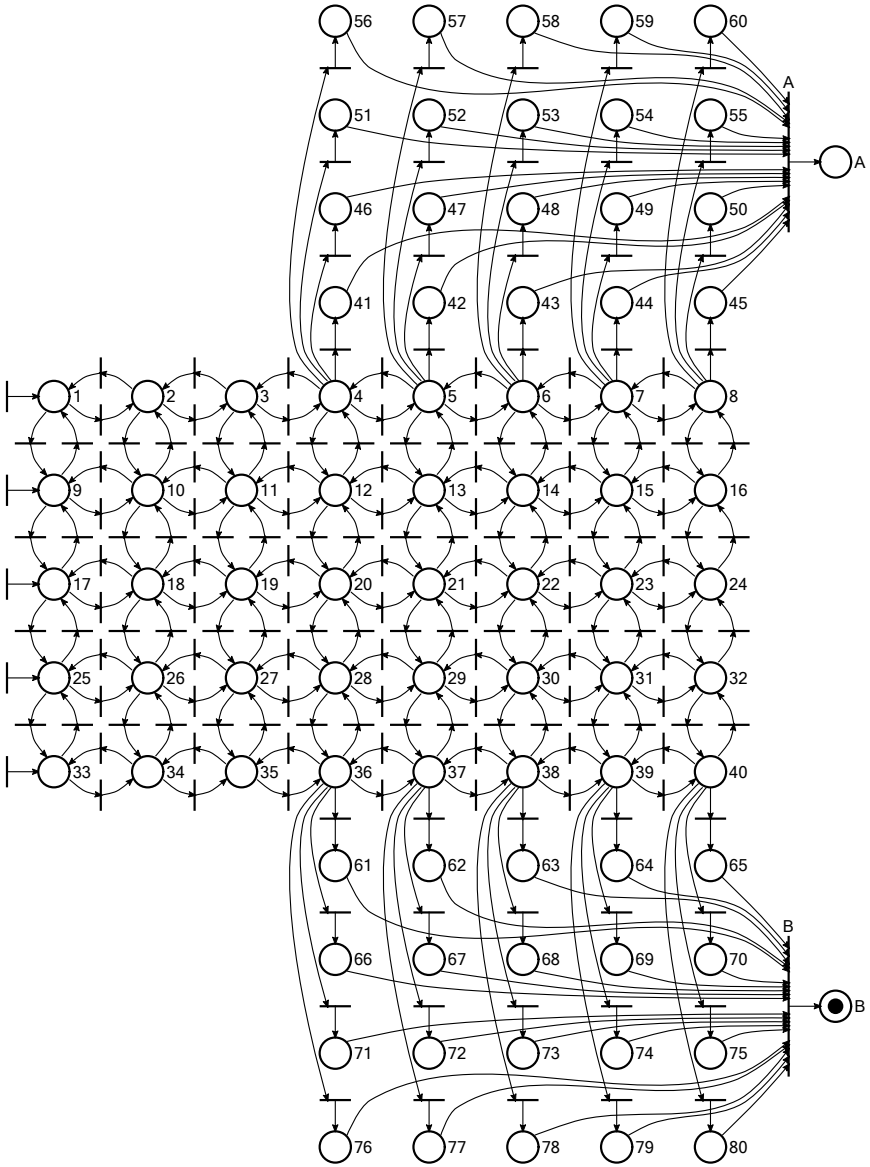


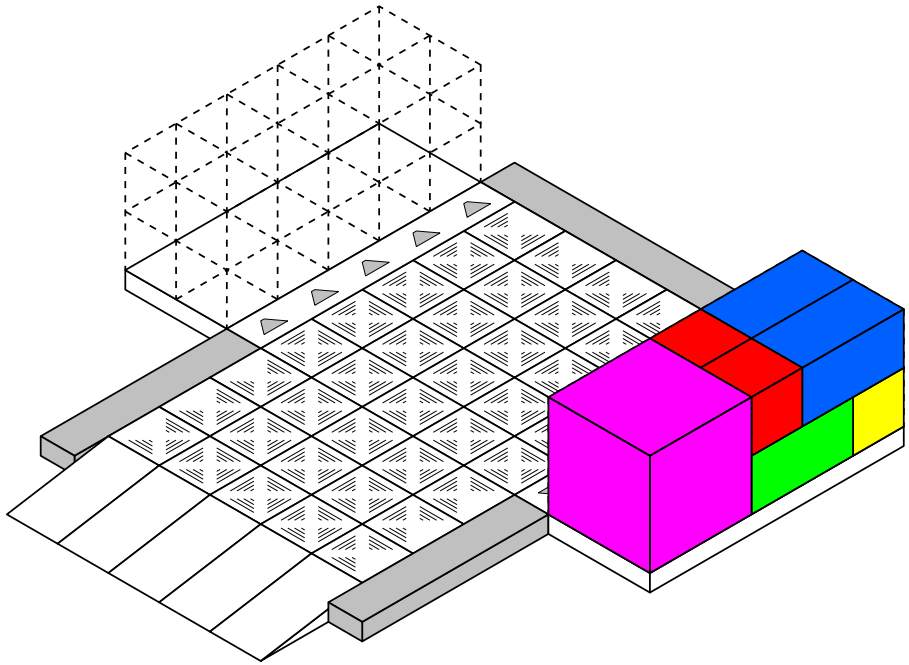
π-sorter/π-composer





π-sorter/π-composer





1. **Introduction**
   - Basics of Petri Nets and some applications
2. **The model of the multimodal hub**
   - π-core
   - π-containers (composed)
3. **The coloured Petri net (CPN)**
   - π-conveyor
   - π-sorter/π-composer
4. **Conclusions**
   - Further research directions

Conclusions

- Petri nets seem to be very suitable for modelling the activities that are carried out by the various π-resources when handling π-containers
- The CPN models here proposed (possibly modified on the basis of specific requirements or characteristics of PI resources and nodes) can be used to analyse system's structure and performances, and to test some control strategies aimed at optimizing system's behaviour
- Next steps:
  - Test and validation of the CPN model for PI's hubs
  - Design and development of a simulation tool (based on the CPN representations)

Further research directions

- The applicability of the Petri net formalism is not limited to the representation of hubs and nodes: they can be effectively adopted also to represent logistics networks with the aim of modelling and controlling the flows of π-containers through the various nodes of a network
  - Better exploitation of the transport capacity, as some mode might be filled with disaggregated π-containers but not with standard container
- To do that with CPNs, further colour sets can be adopted in order to associate more detailed information to the tokens representing the aggregations of π-containers travelling on the logistic network
  - destination of each π-container, status of them, etc.

NOLI, A Proposal for an Open Logistics Interconnection Reference Model for a Physical Internet

J.-Y. Colin, H. Mathieu and M. Nakechbandi

LITIS, Le Havre University, Le Havre, France
ISEL, Le Havre University, Le Havre, France

Work supported by the "Fond Européen de Développement Régional" (FEDER) of the European Union, and by the "Corridors Logistiques : Applications à la vallée de la Seine et Son Environnement" Project (CLASSE2), of the Région Normandie, France.

Outline

- Previous Works
- The Internet
- The TCP/IP Reference Model
- The OSI Reference Model
- The OLI Reference Model
- The NOLI Reference Model
- Discussion
- Future Work

Previous Works

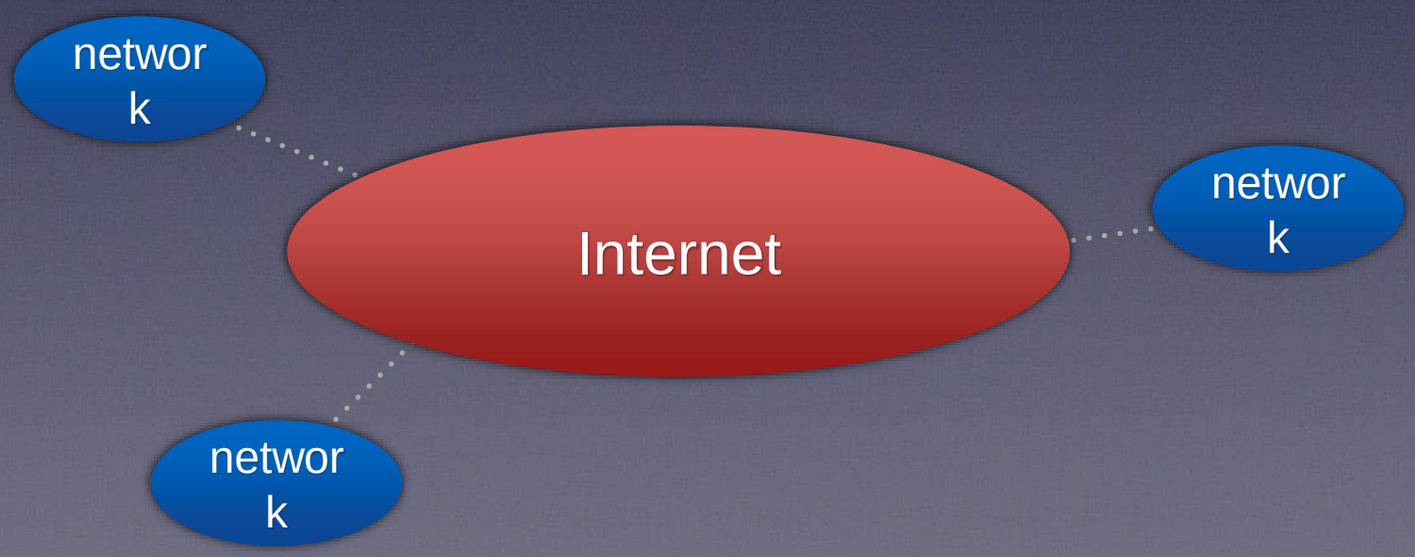
- B. Montreuil, "Physical Internet manifesto: globally transforming the way physical objects are handled, moved, stored, realized, supplied and used", Technical Report, 2009.
- B. Montreuil, E. Ballot, and F. Fontane, "An open logistics interconnection model for the Physical Internet". In Proceedings of INCOM 2012 Symposium, Bucharest, Romania. 2012.

Previous Works

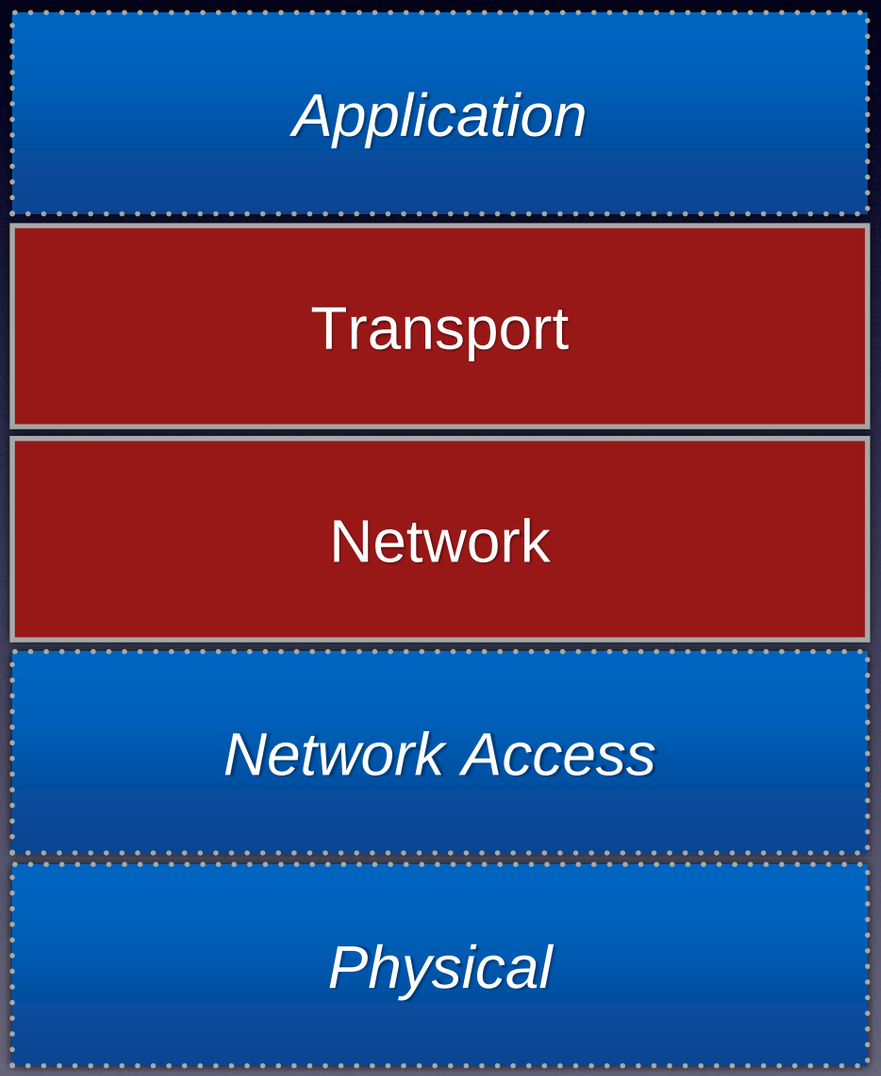
- International Physical Internet Conferences, IPIC 2014 - 2015 - 2016.
- http://physicalinternetinitiative.org/
- related : Modulushca « smart » containers project (http://www.modulushca.eu/)
- not directly related : Internet-of-Things (IoT)

The Internet

- Internet is a « network » of networks, not a network of computers
- The integration of the heterogeneous networks is ensured by a suite of precise protocols
  - inside, each network does what it wants
  - but data exchanged between networks must conform (using gateways) to the Internet protocols defined in the **TCP/IP Reference Model**

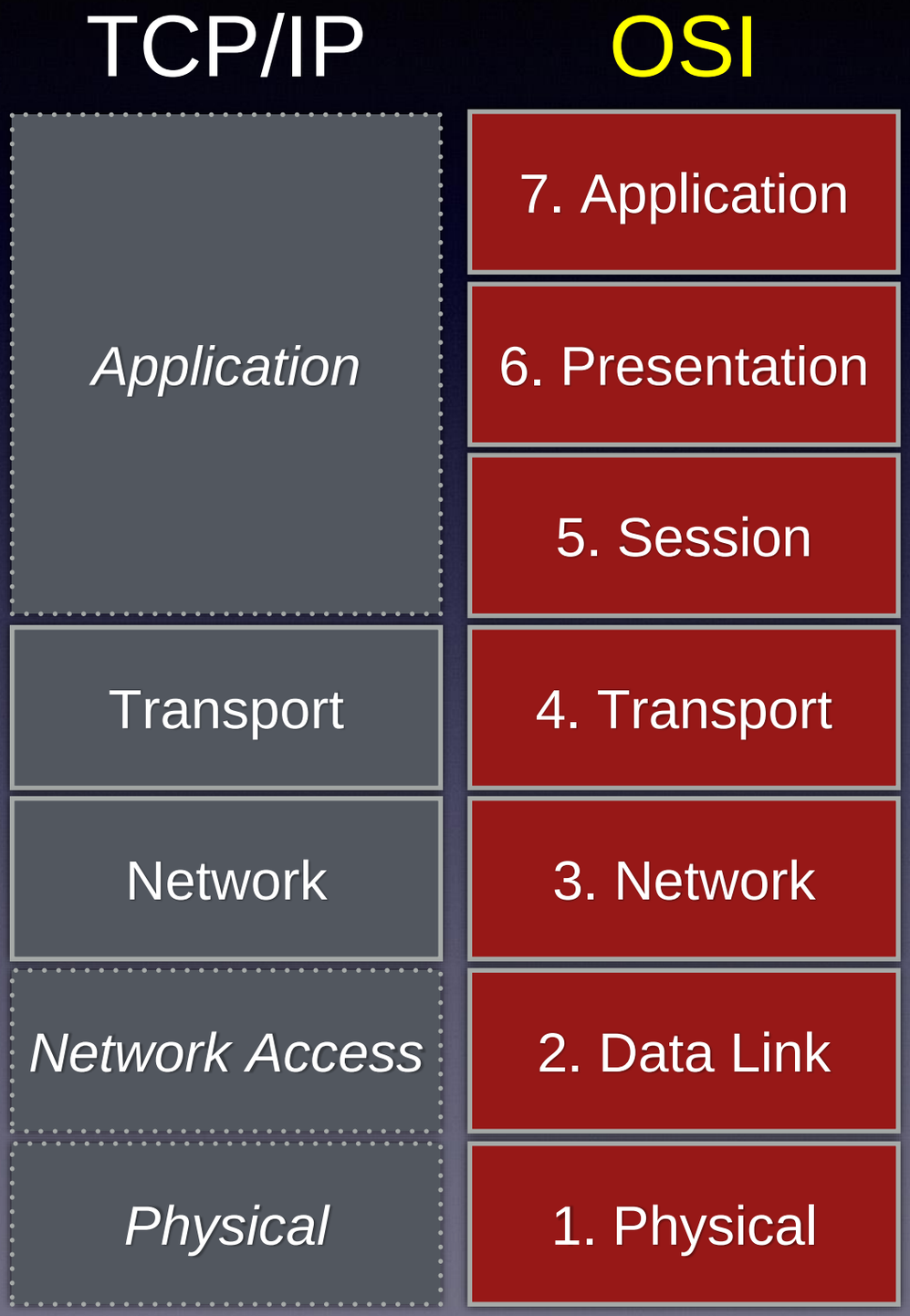


The TCP/IP Reference Model



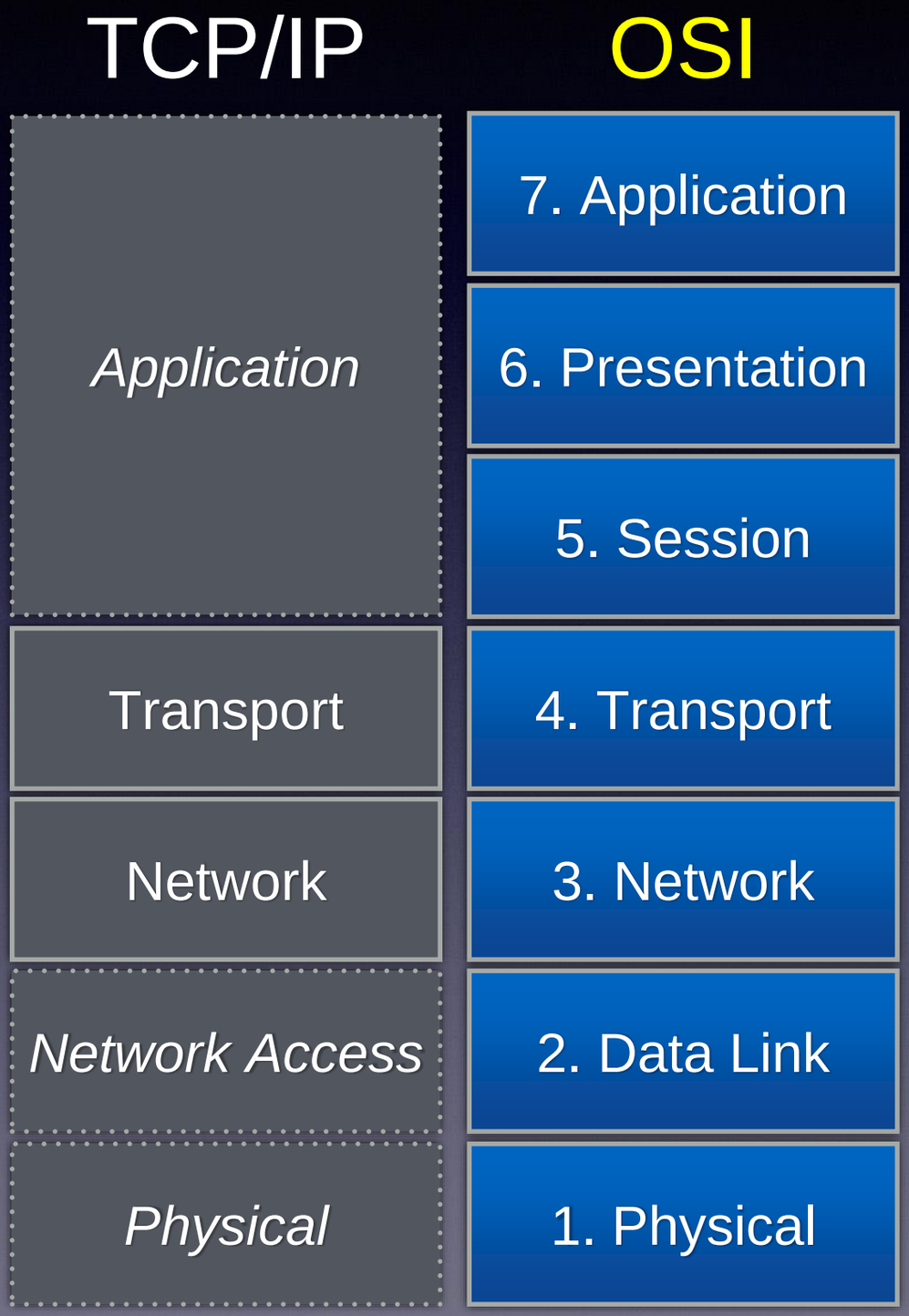
- Layered structure (« stack ») of protocols
- 2 precisely very defined layers in the middle: the Transport Layer and the Network Layer

The OSI Reference Model



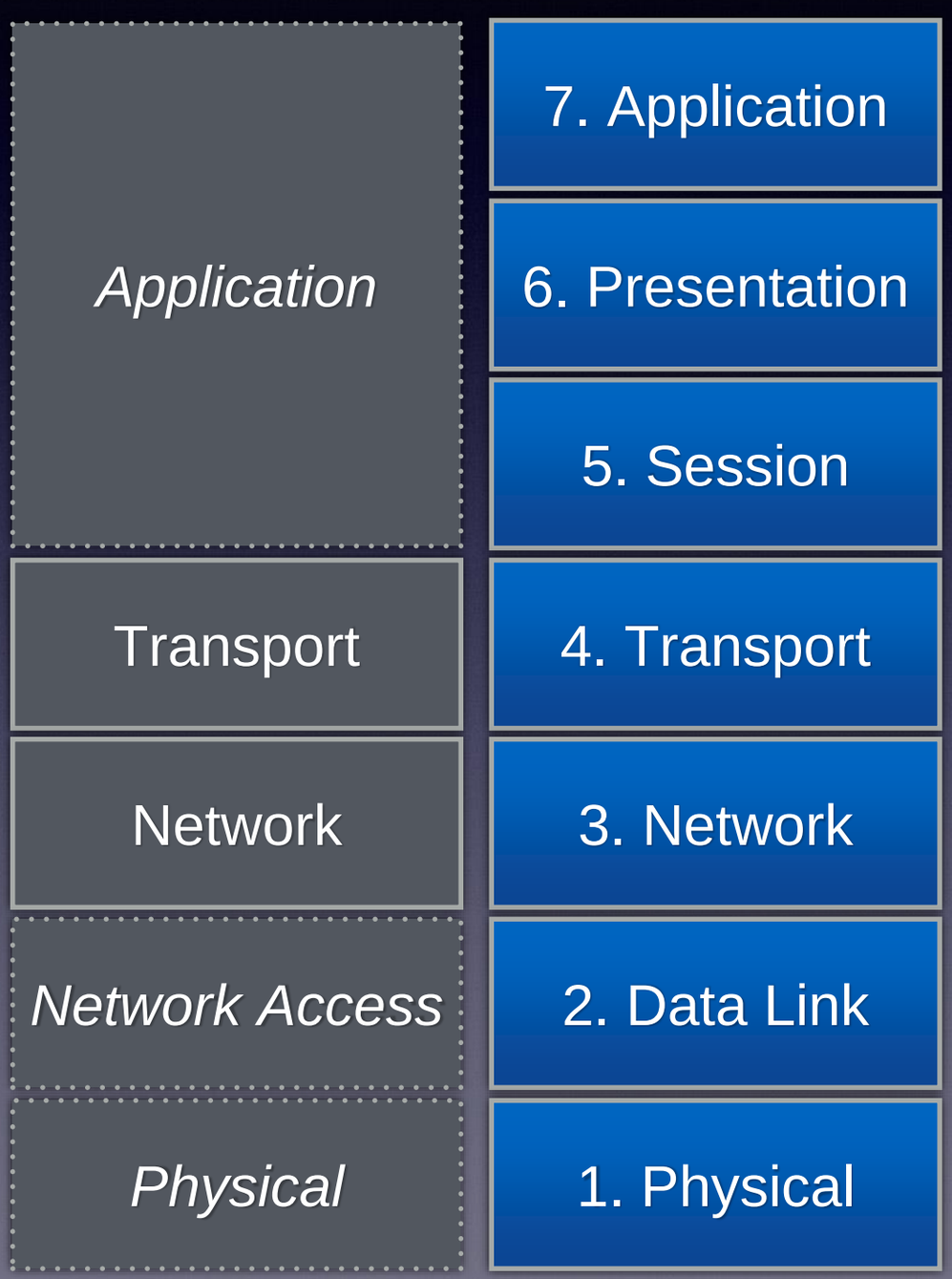
- mostly a pedagogical tool, but it influenced the evolution of TCP/IP, and is widely referred to when new technologies are introduced
- most OSI layers have 2 or 3 sub-layers
- a few differences between TCP/IP and OSI, on the boundaries of some layers
- all physical definitions are in the lowest layer

The OSI Reference Model



- **Example**: sending (from top to bottom)
  - 7. Application: email, etc.
  - 6. Presentation: data encoding, compressing, etc.
  - 5. Session: start transaction
  - 4. Transport: establishes contact, cut message into segments
  - 3. find route for each packet
  - 2. manage individual direct « hop »
  - 1. define, connect to and control physical means

The OSI Reference Model

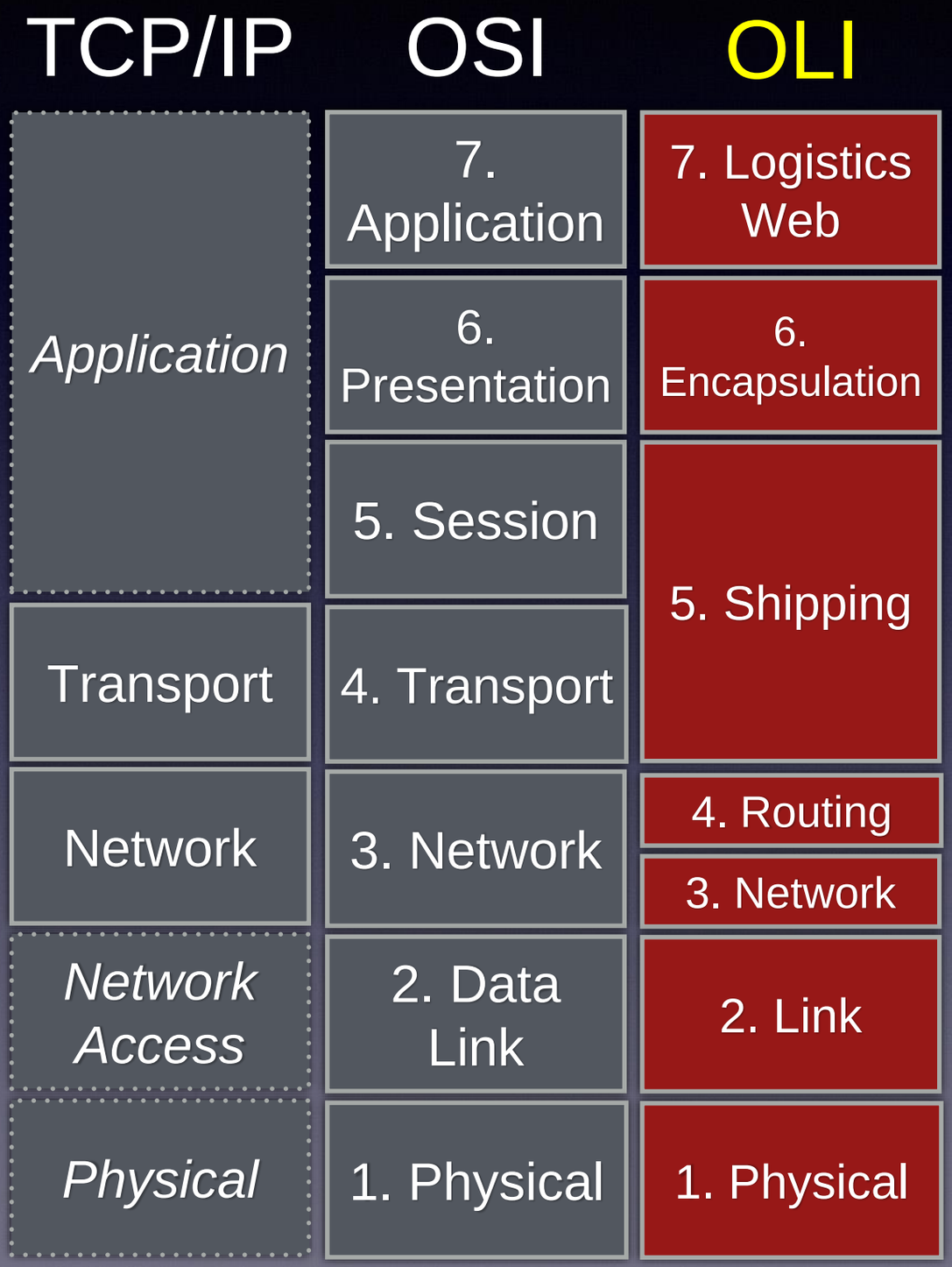


- **Example**: receiving (from bottom to top)
  - 7. Application: email, etc.
  - 6. Presentation: data decoding, decompressing, etc.
  - 5. Session: close transaction
  - 4. Transport: receives contact, merge segments into message
  - 3. find route for each packet
  - 2. manage individual direct « hop »
  - 1. define, connect to and control physical means

A Reference Model for Logistics:

The OLI Reference Model (Montreuil, Ballot et Fontane, 2012)

The OLI Reference Model



- 7 layers
- includes data and physical parts (containers, movers,…)
- some differences in what each layer manages (layers 3, 4 and 5) compared to the TCP/IP and OSI models
- all physical parts are defined in the lowest layer

We propose an other Reference Model for Logistics:

- The NOLI Reference Model

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

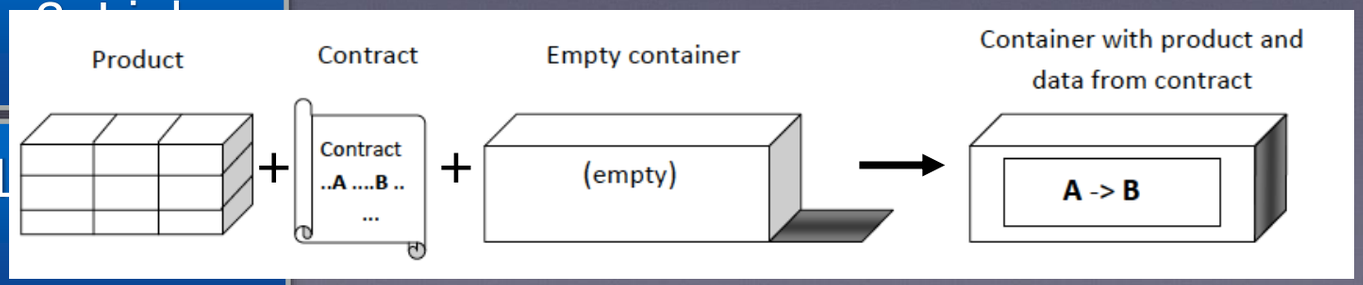
- 7 layers
- includes data and physical parts (containers, movers,…)
- structure closer to the TCP/IP and OSI models

Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	
Physical	1. Physical	1. Physical	1

- Product Layer (« ***Client Layer*** ») :
  - defines the possible physical cargoes and their specificities.
  - includes the exact identification of the type of goods, and its characteristics such as the fact that it is perishable or that it is fragile.
  - establishes the contract for each filled container
  - fills empty container with the products

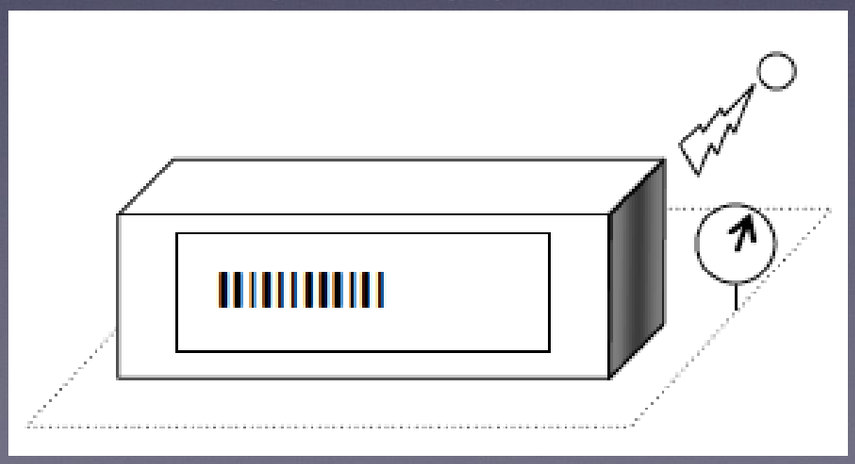


Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

- **Container Layer**:
  - defines the physical characteristics of the containers (its size, or the fact that the container is a refrigerated one, for example)
  - check containers
  - anonymize data
  - manages empty containers

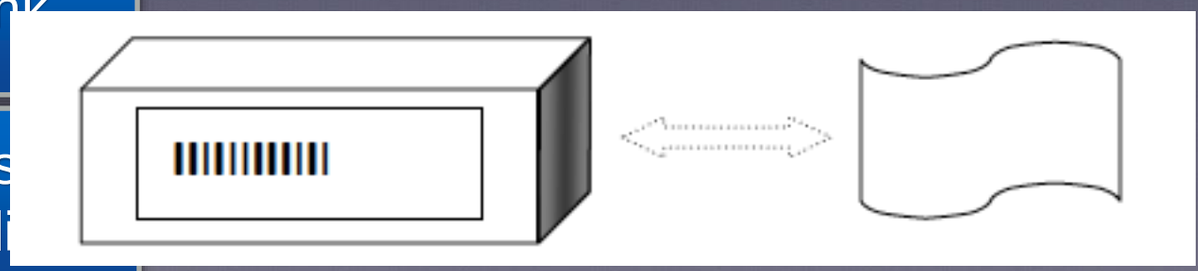


Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Phys Handli

- **Order Layer**:
  - establishes the "dispatch note" associated to each container of each set.
  - records priorities and deadlines of containers.
  - divides and/or combines the sets into "orders"
  - checks the possible problems (for example, does the final ending location accepts dangerous material? etc.)
  - manages transactions

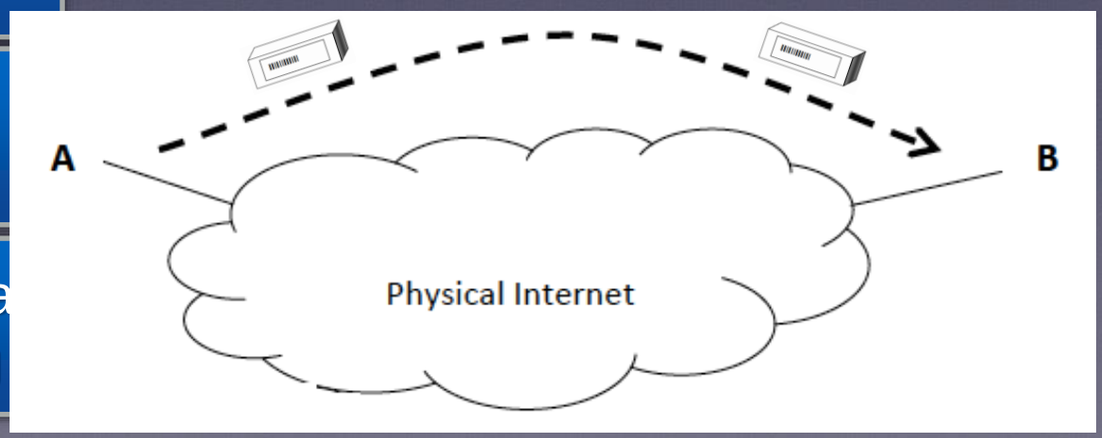


Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

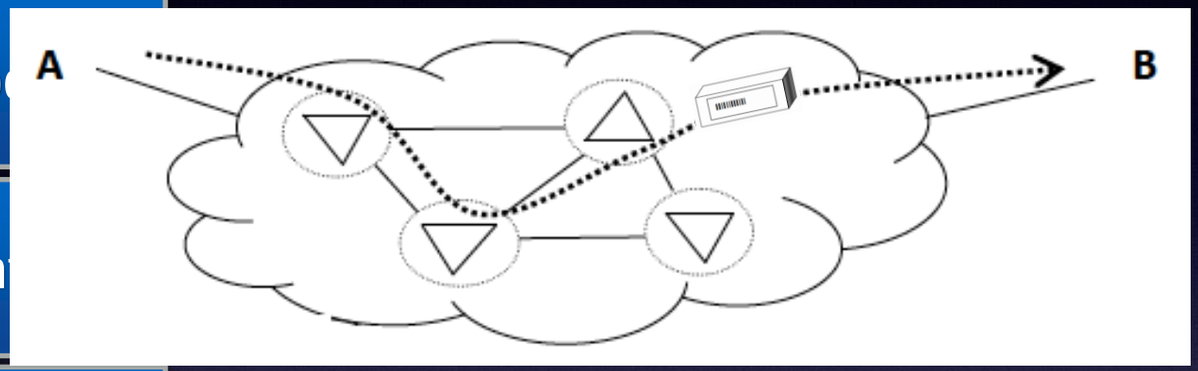
- **Transport Layer**:
  - manages the end-to-end trip of each load from its initial starting location to its final ending location.
  - checks that the final ending location can handle a load shipped there.
  - signals to the Order Layer the initial departure, the current location and the final arrival of each container.
  - ensures that deadlines are respected



Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Pro
	6. Presentation	6. Encapsulation	6. Con
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling



- **Network Layer**:
  - computes and manages the routing of each block from its initial starting location to its final ending location.
  - manages and maintains the data structures necessary to compute the best paths for the blocks

Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	
	6. Presentation	6. Encapsulation	
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

- **Link Layer**:
  - manages the individual steps of movements of containers on means. A "step" is one individual point-to-point movement.
  - also manages the handling of a block by a company/operator to another company/operator.

Colin, Mathieu and Nakechbandi, 2017

The NOLI Reference Model

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

- **Physical Handling Layer**:
  - defines physical characteristics of the means, such as ships, trucks, cranes, belt conveyors, etc. (long-distance conveyances and local handling gears are considered to be at the same level in this model).
  - manages the scheduling and mapping of containers on means.
  - gives the orders to the means.
  - signals problems (breakdowns, delays, etc.)

Colin, Mathieu and Nakechbandi, 2017

Discussion

Colin, Mathieu and Nakechbandi, 2017

Discussion

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

1.Physical components are defined in several NOLI layers

- Goods in the 7. Product Layer
- Containers in the 6. Container Layer
- Movers in the 1. Physical Handling Layer

Discussion

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

2. The containerization and de-containerization are done in the 7. Product Layer in the NOLI model, instead of in the 6. Encapsulation Layer of the OLI model.

Discussion

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

3.end-to-end transportation (as in the OSI model) is managed in two separate layers.

- one for administrative preparation.
- one for the physical management.

Discussion

TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

4. routing and network management is done in one layer

Discussion

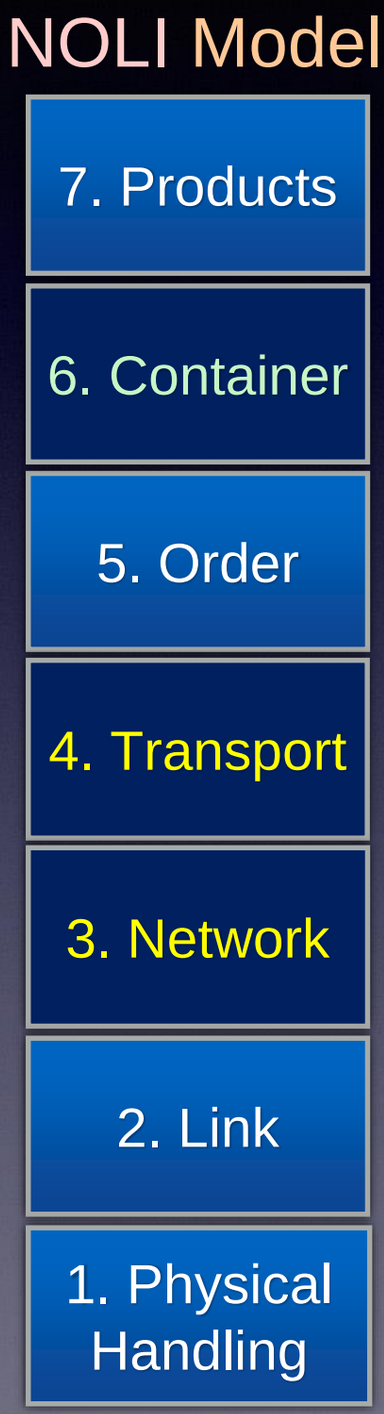
TCP/IP	OSI	OLI	NOLI
Application	7. Application	7. Logistics Web	7. Products
	6. Presentation	6. Encapsulation	6. Container
	5. Session	5. Shipping	5. Order
Transport	4. Transport		4. Transport
Network	3. Network	4. Routing	3. Network
		3. Network	
Network Access	2. Data Link	2. Data Link	2. Link
Physical	1. Physical	1. Physical	1. Physical Handling

5. the π-means are defined in the 1. Physical Handling Layer

Future Work

- In the Container Layer, efficiently manage rare mobile resources (such as « reefers ») that tend to accumulate in some places and are missing in others in a Physical Internet.

- In the Transport Layer and the Networks Layer, create algorithms that offer good alternative and reactive transportation solutions in a Physical Internet in the presence of random conditions (using dynamical graphs as models).



Thank you.

Questions?