Das Heiratsproblem

Eva Hennefeld & Anaïs Siebers

Graphentheorie WS 19/20

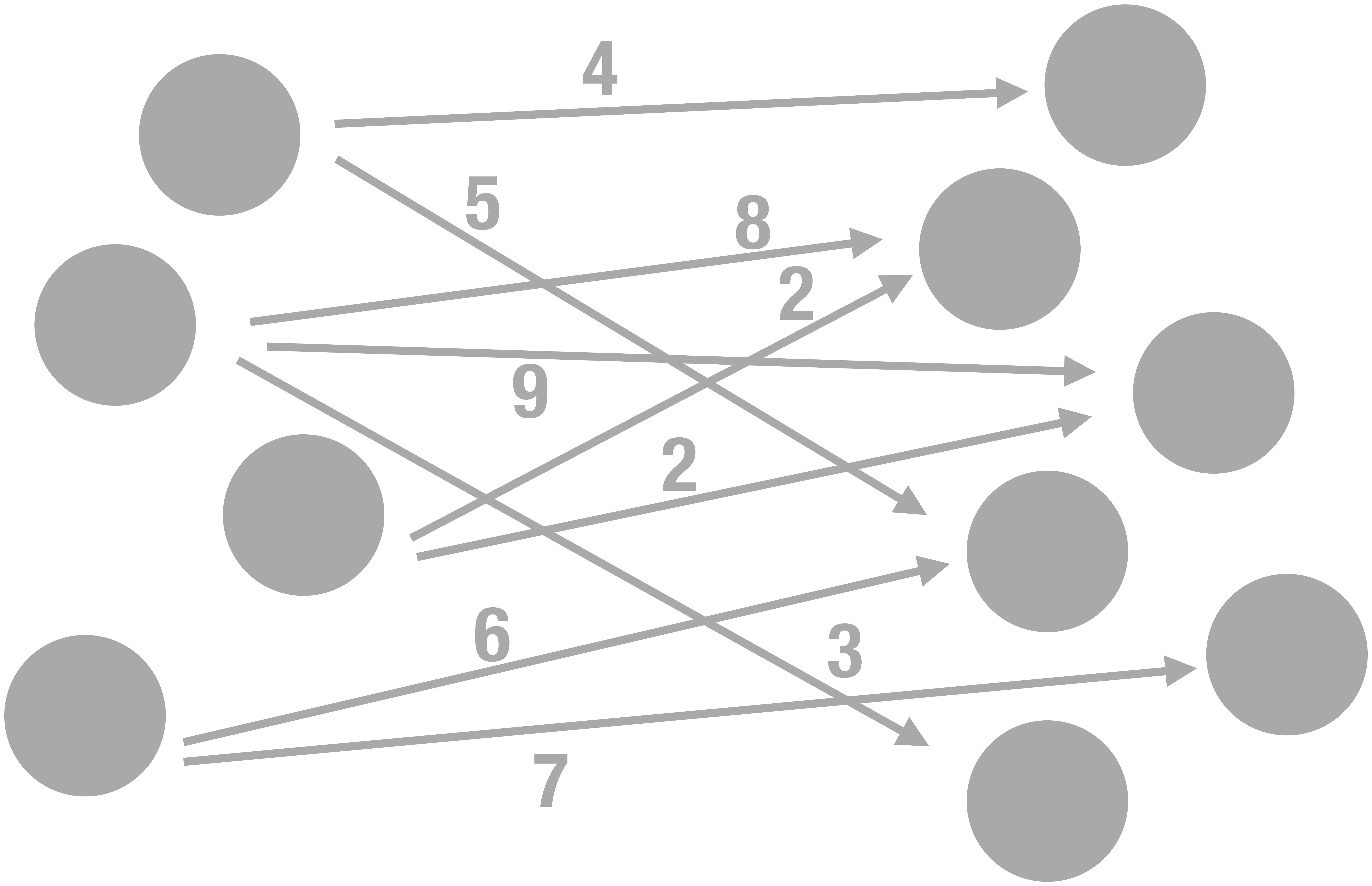
Hochschule Rhein-Main

Professor Reith

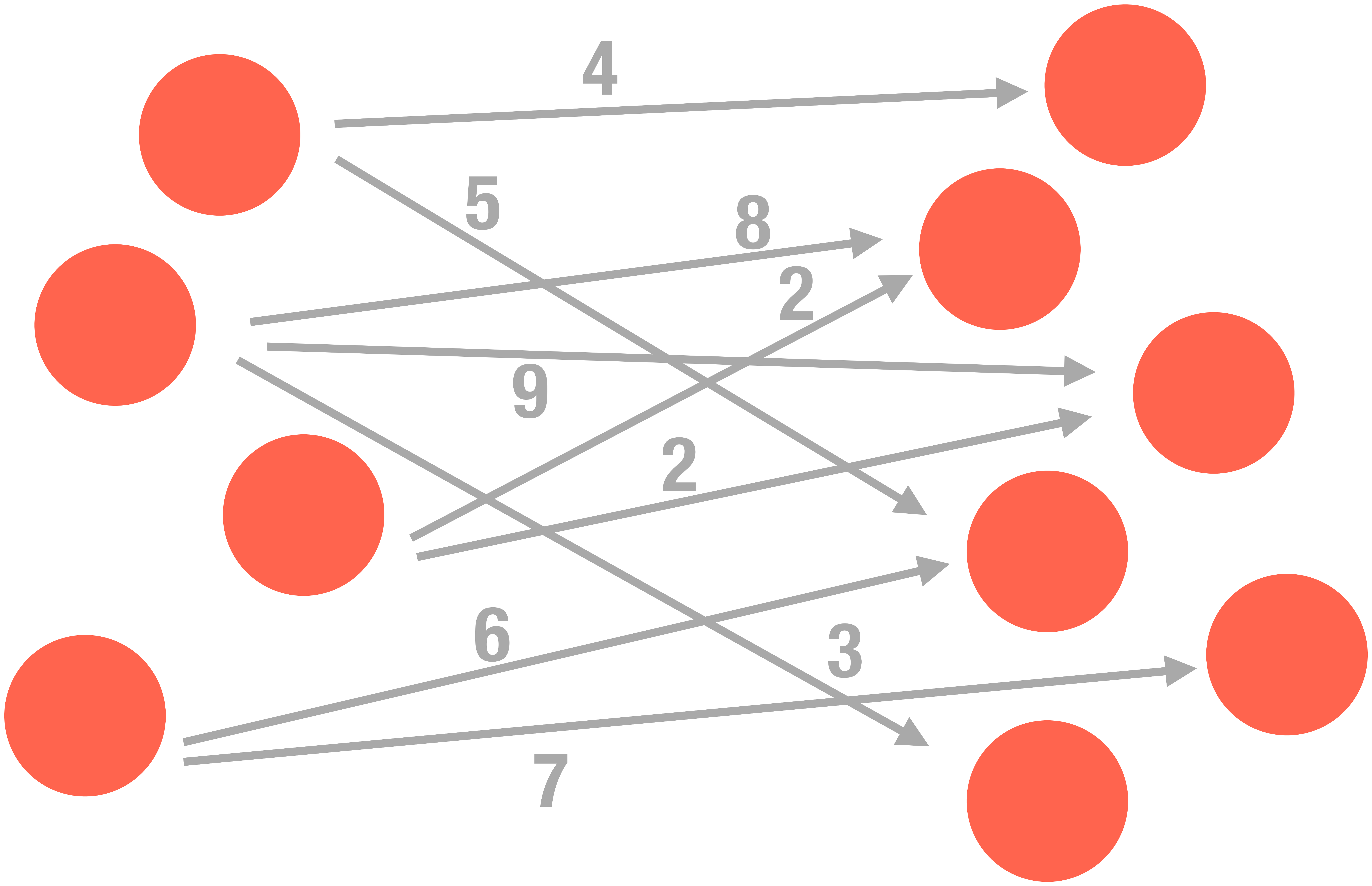
Wie lassen sich zwei gleichgroße
Gruppen an Männern und Frauen
glücklich verheiraten ?

Grundlagen

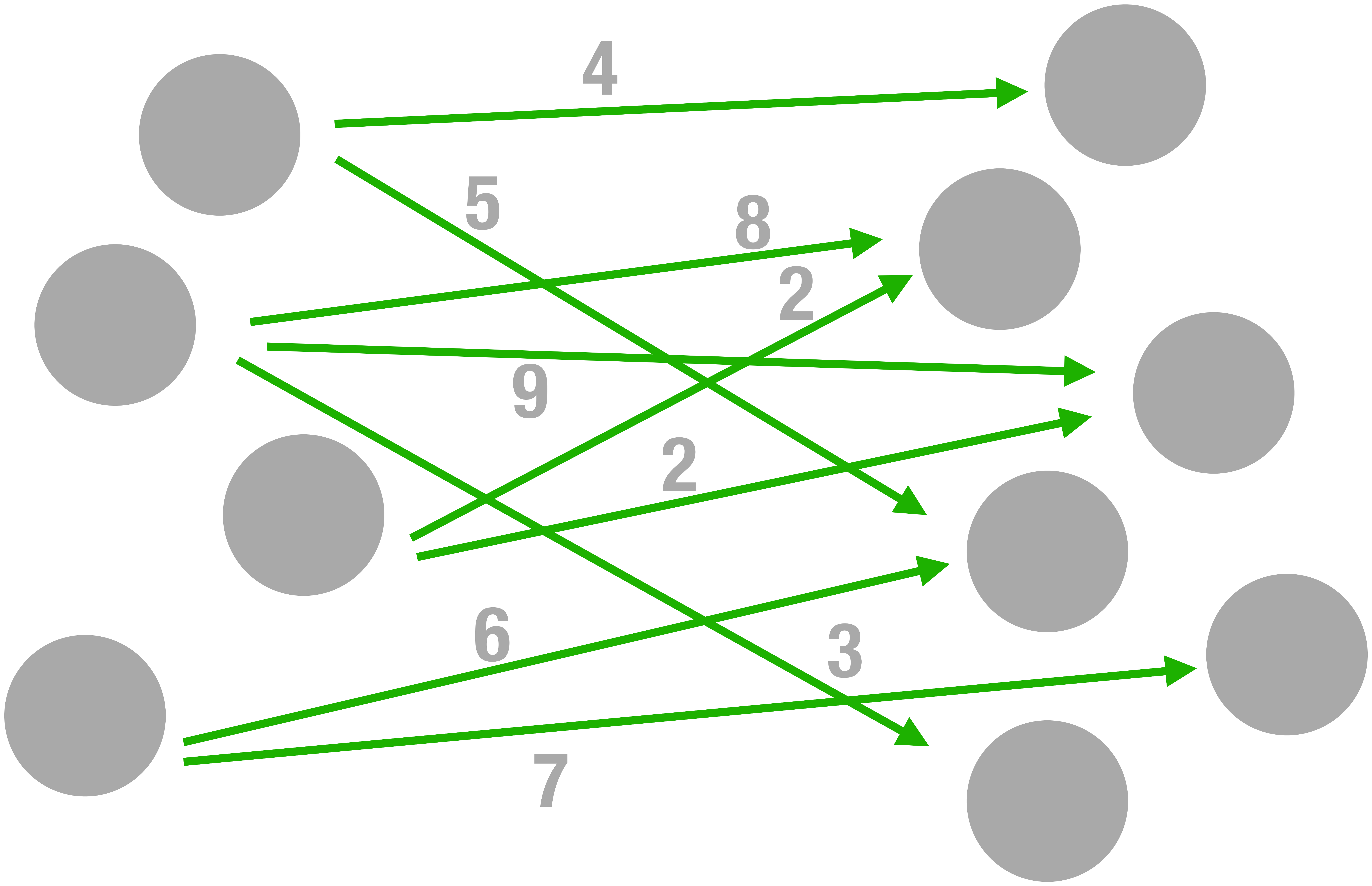
Definition



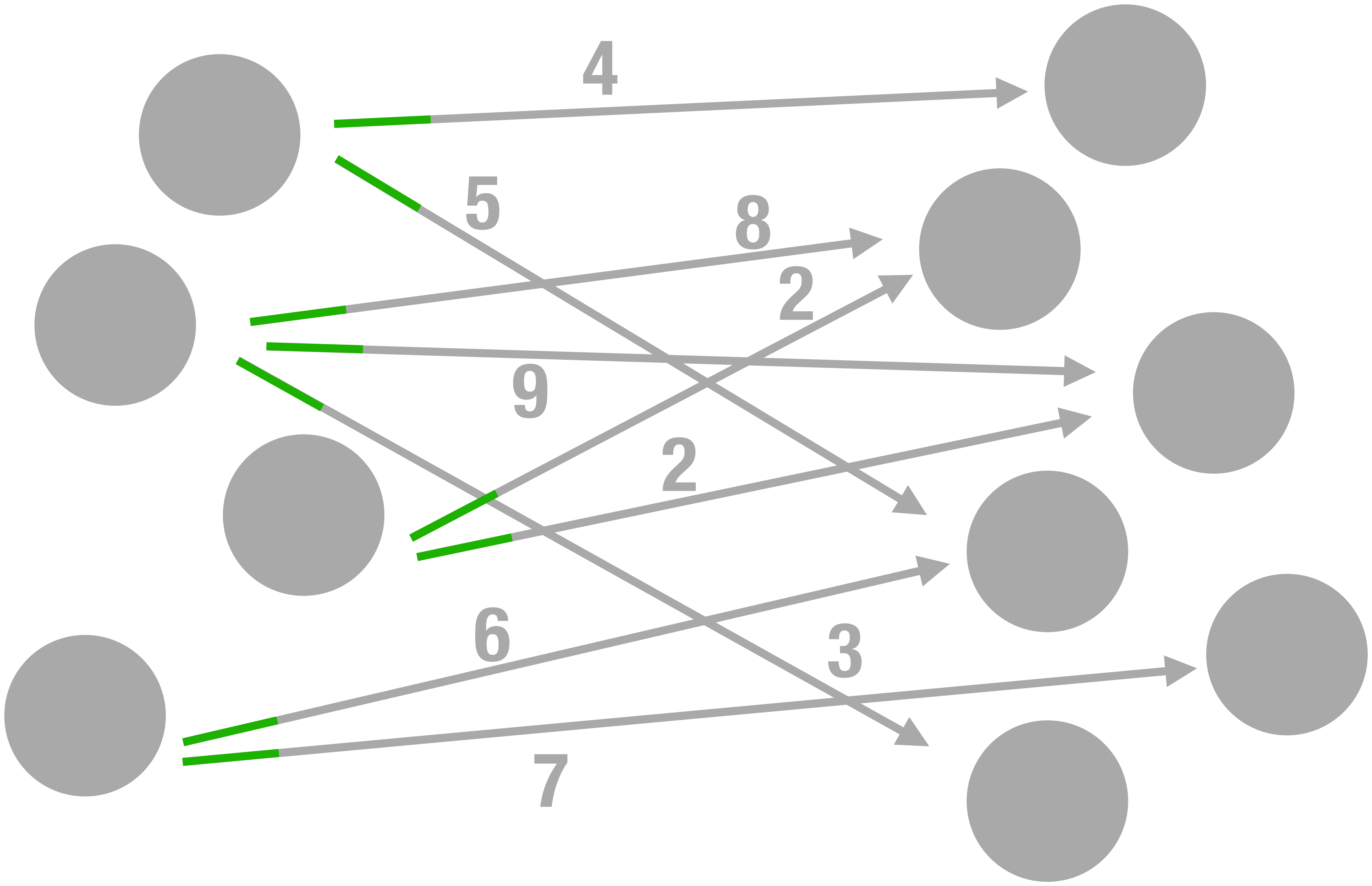
$$G = (V, E)$$



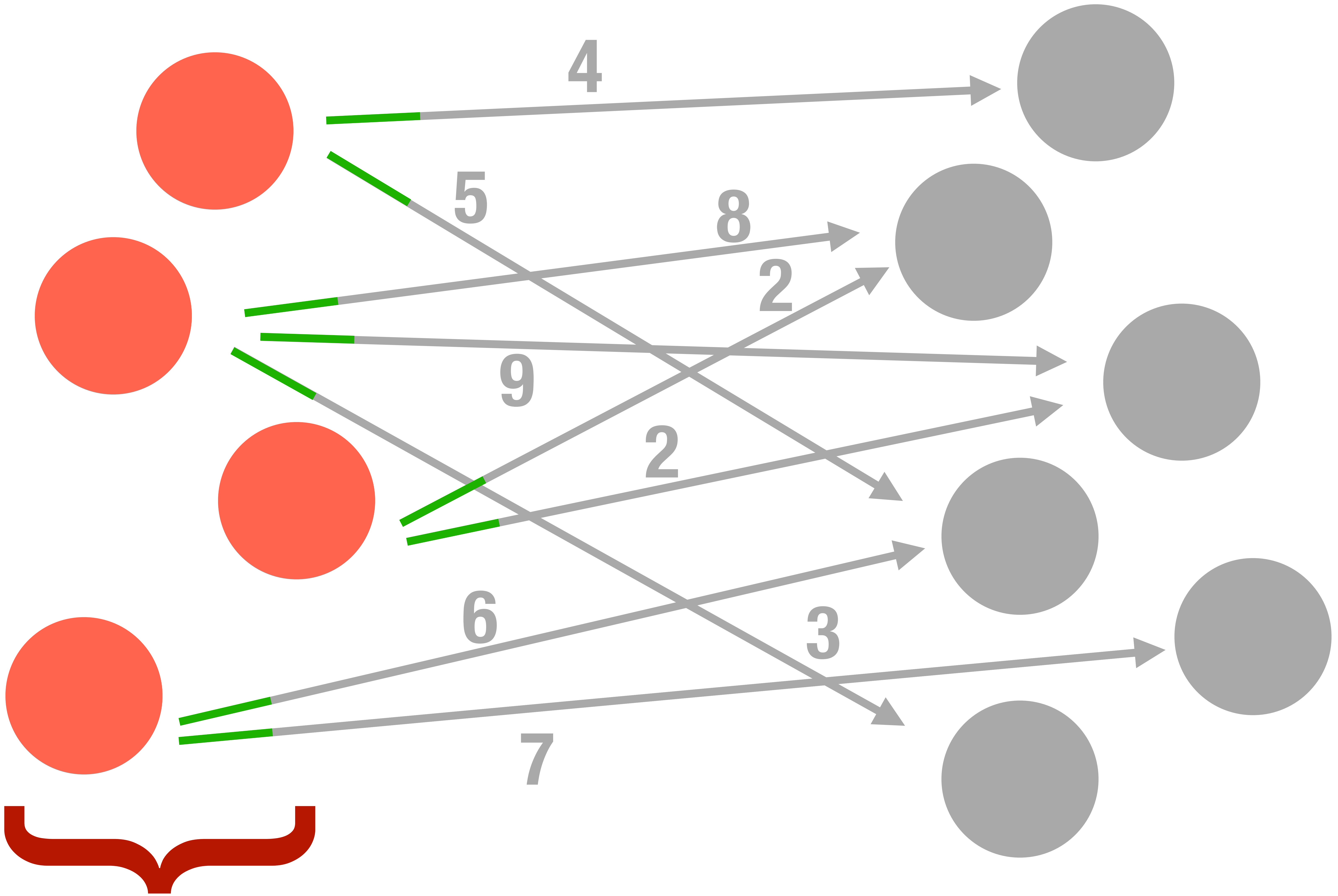
$$V = \{v_1, \ldots, v_n\}$$



$$E = \{e_1, \ldots, e_n\}$$

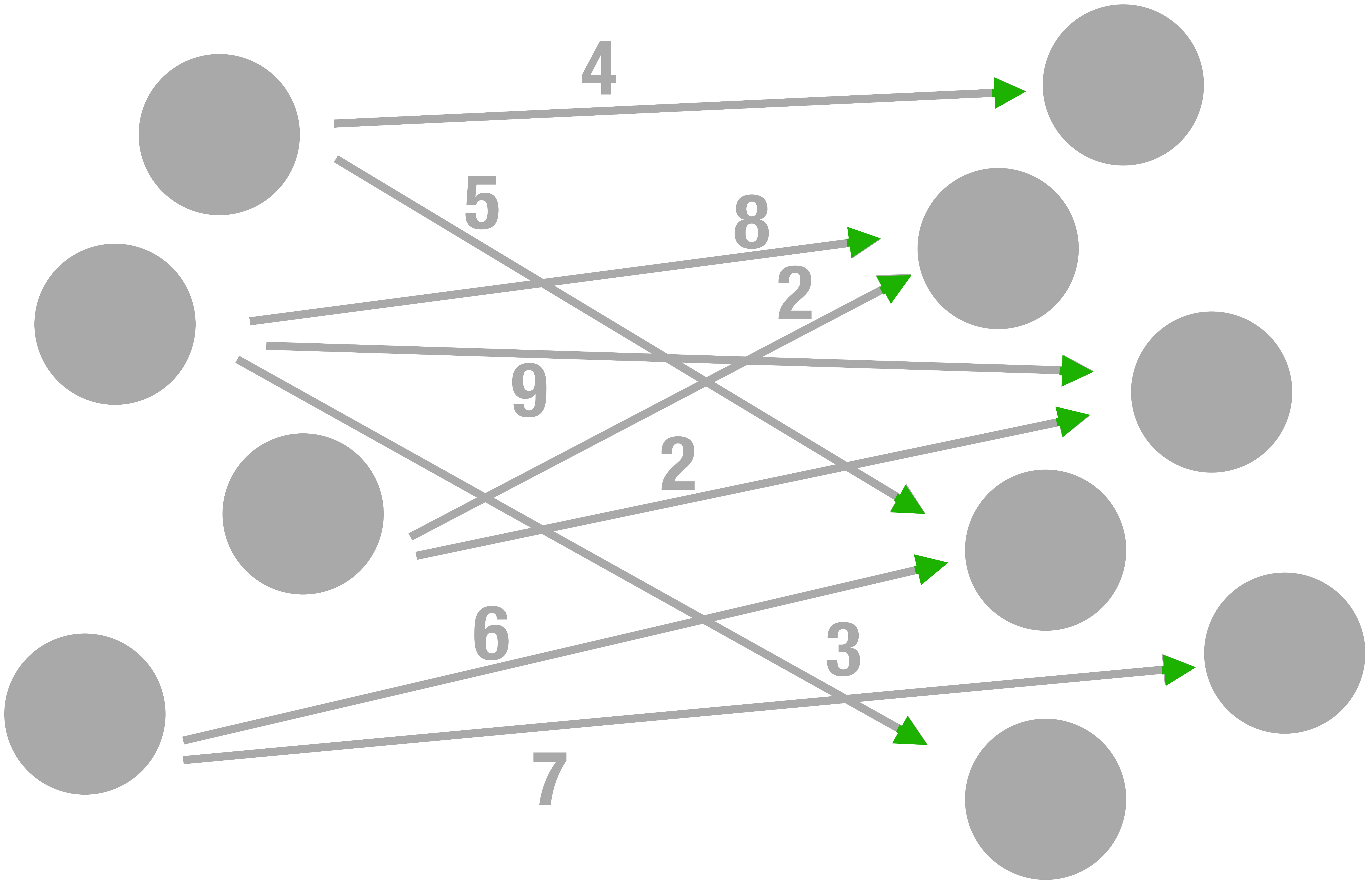


$\delta^+(v)$

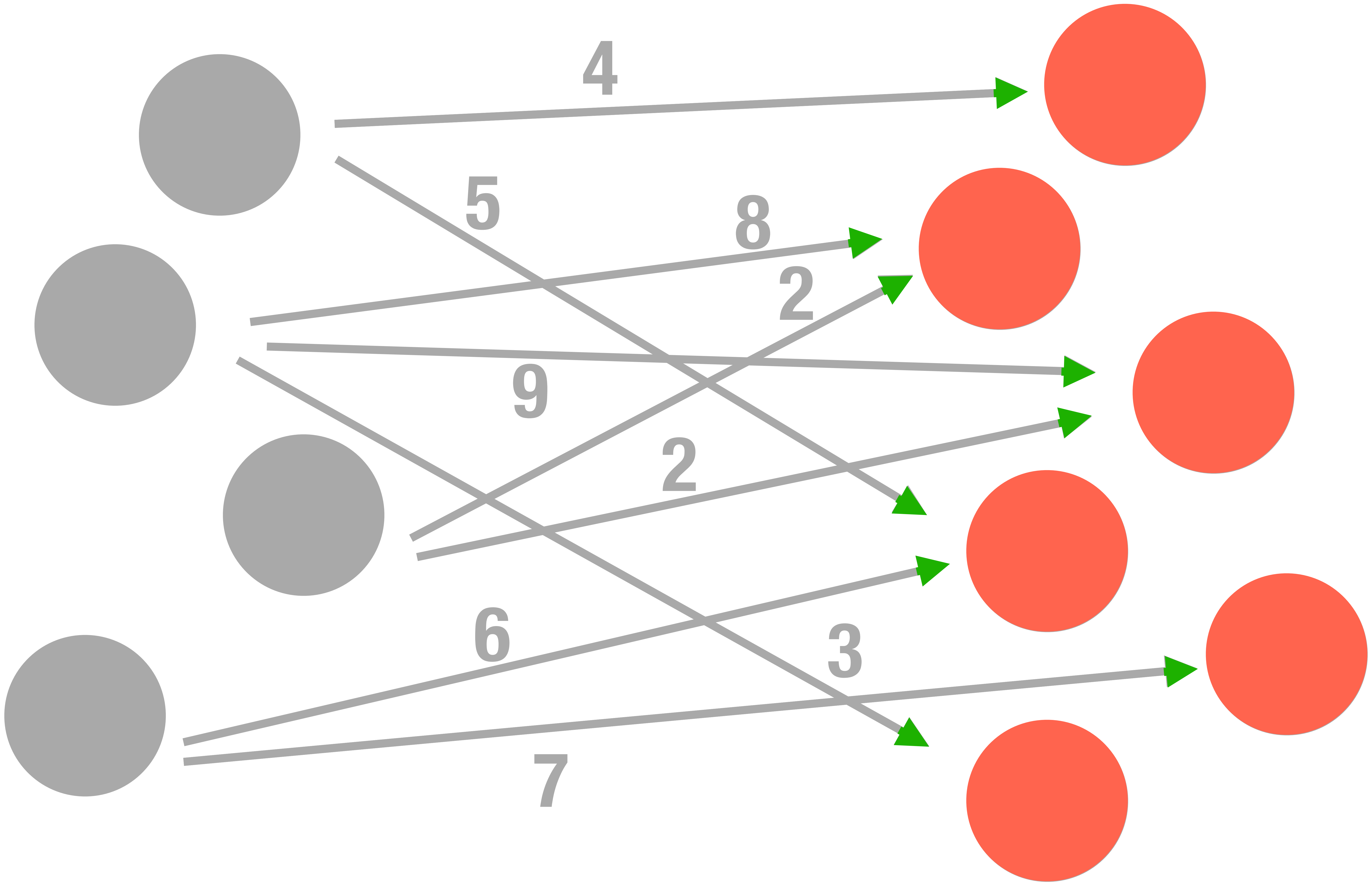


Quelle

$\delta^+(v)$



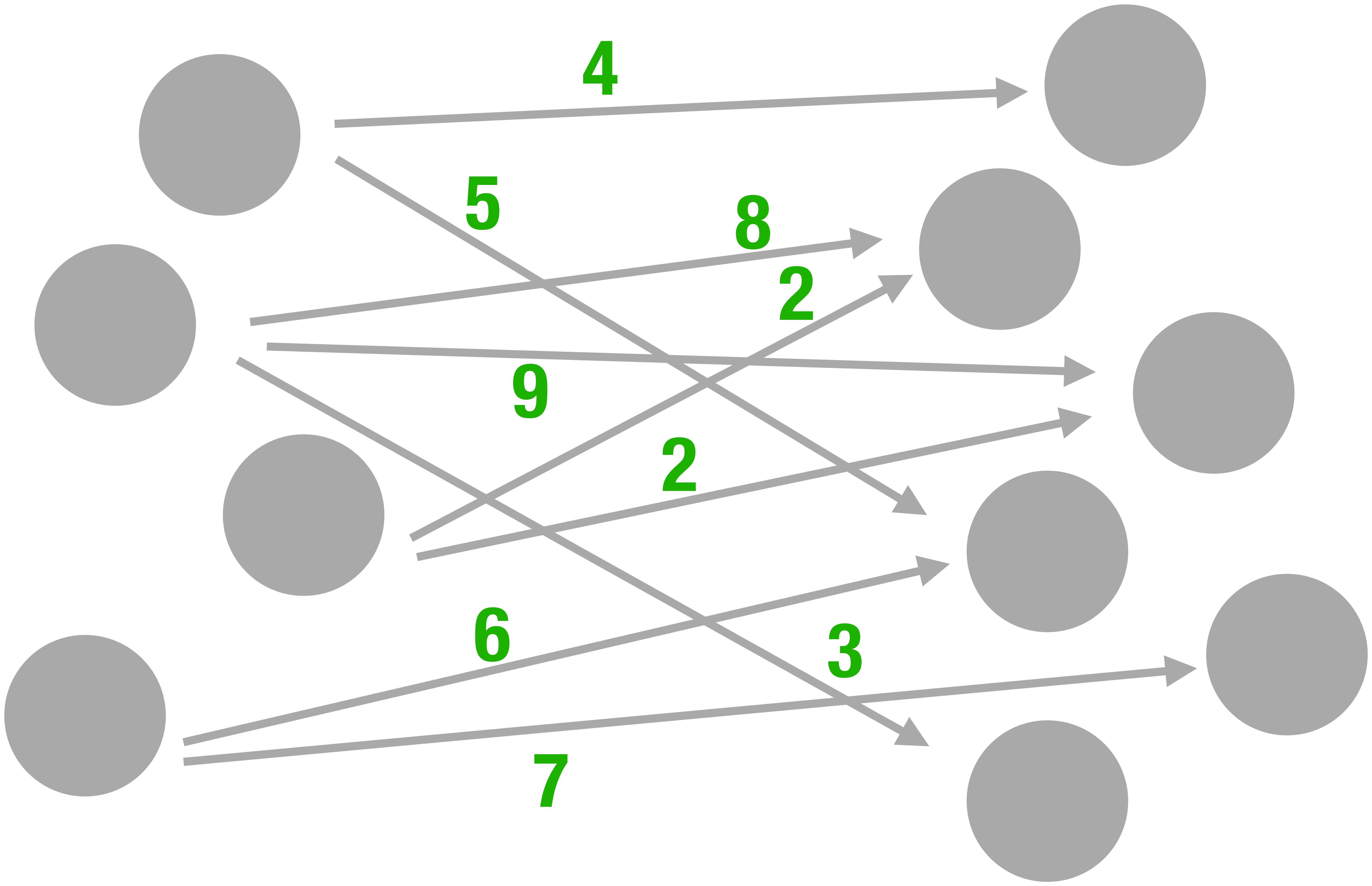
$\delta^{-}(v)$



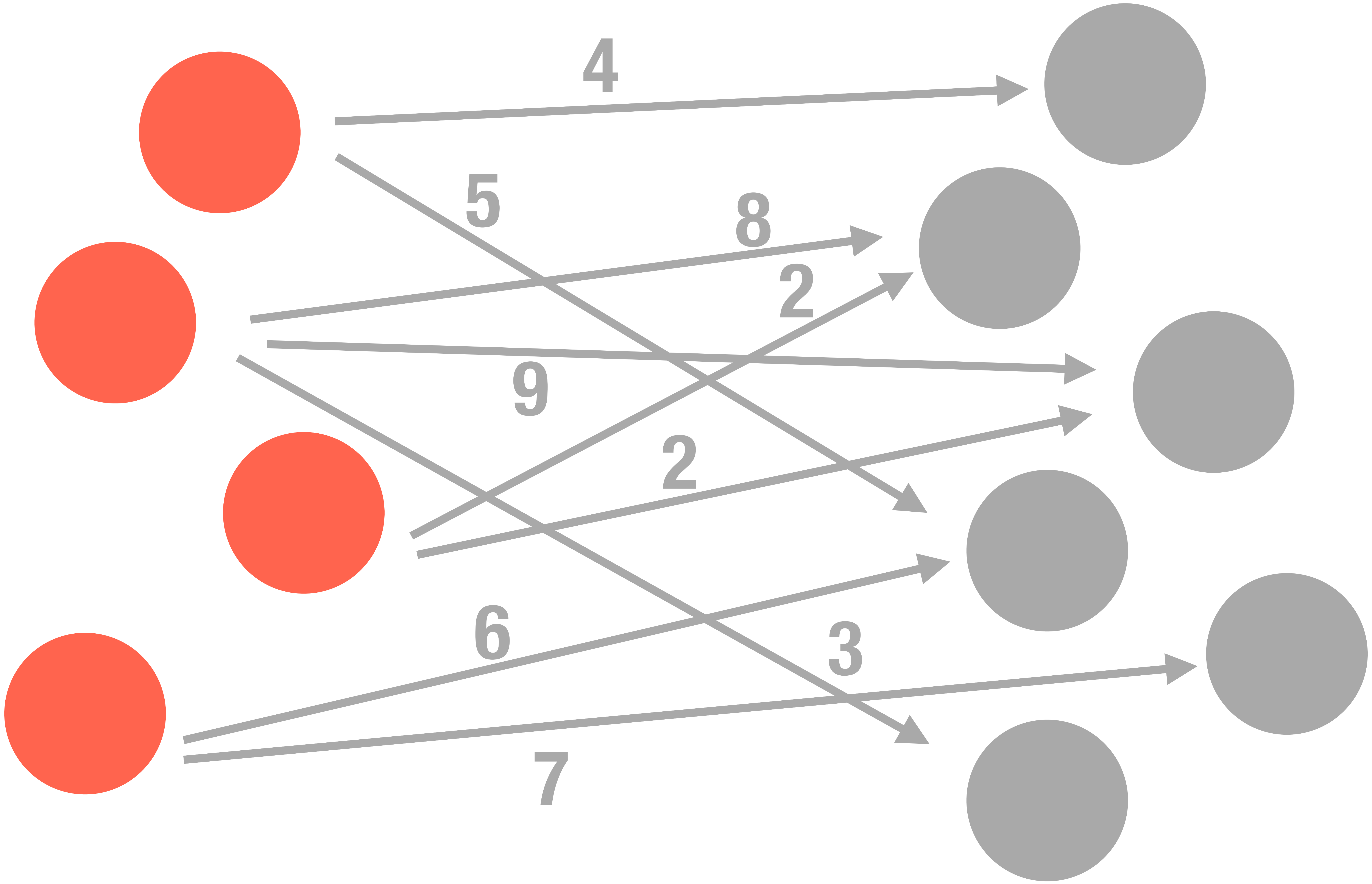
$\delta^-(v)$

$\underbrace{\qquad\qquad}$

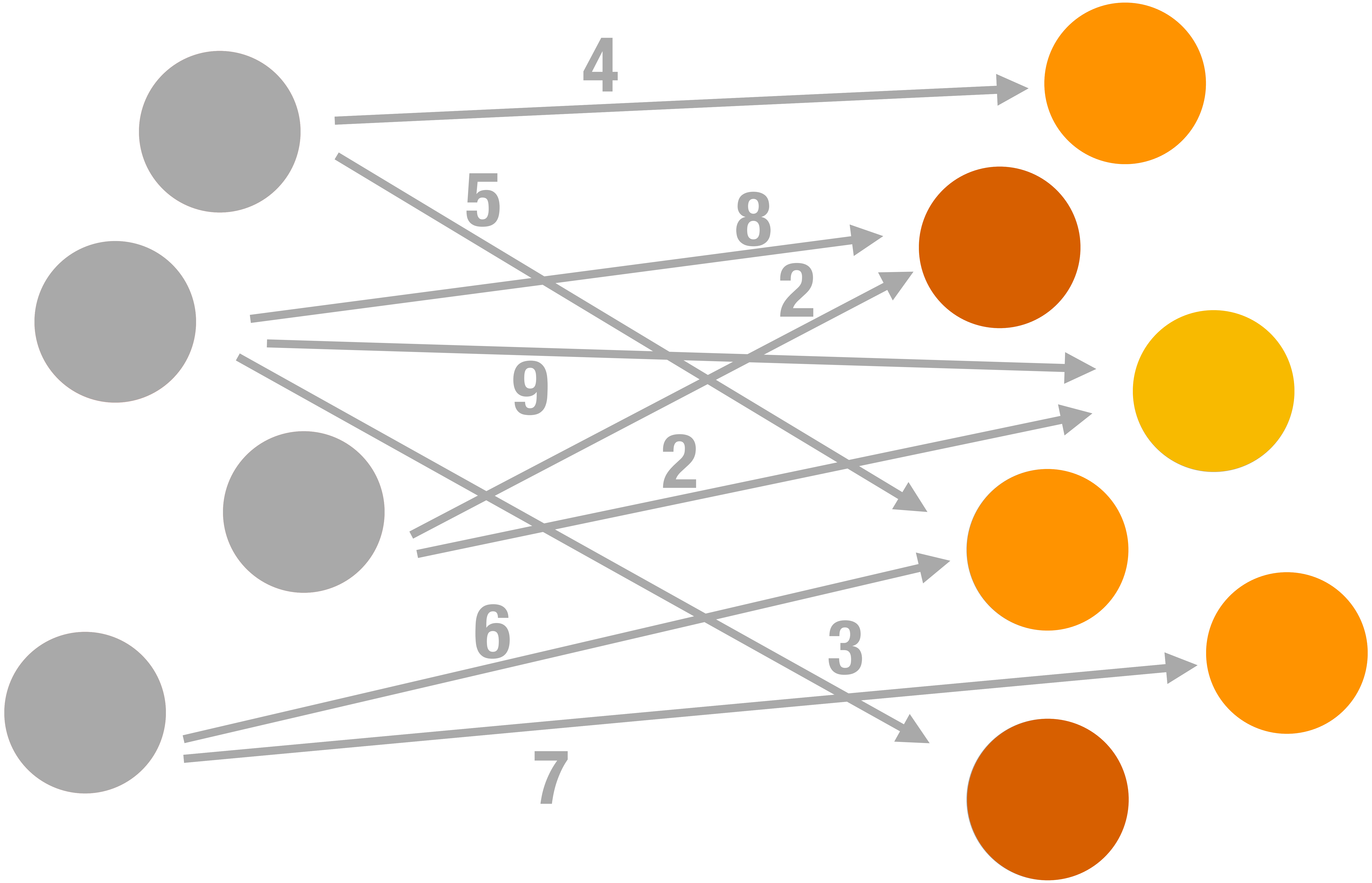
Senke



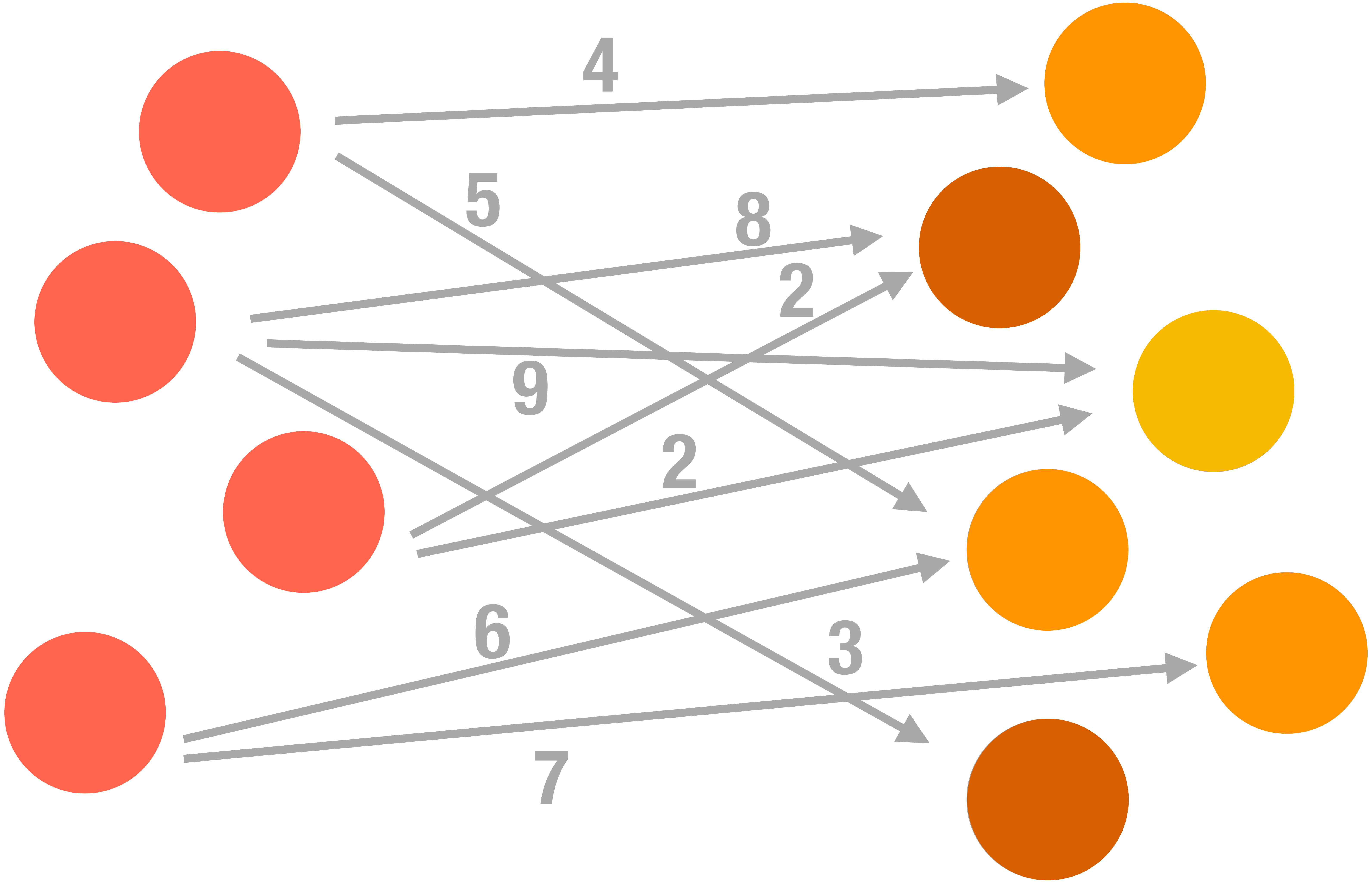
$$\forall e \in E \quad \exists w \in W$$



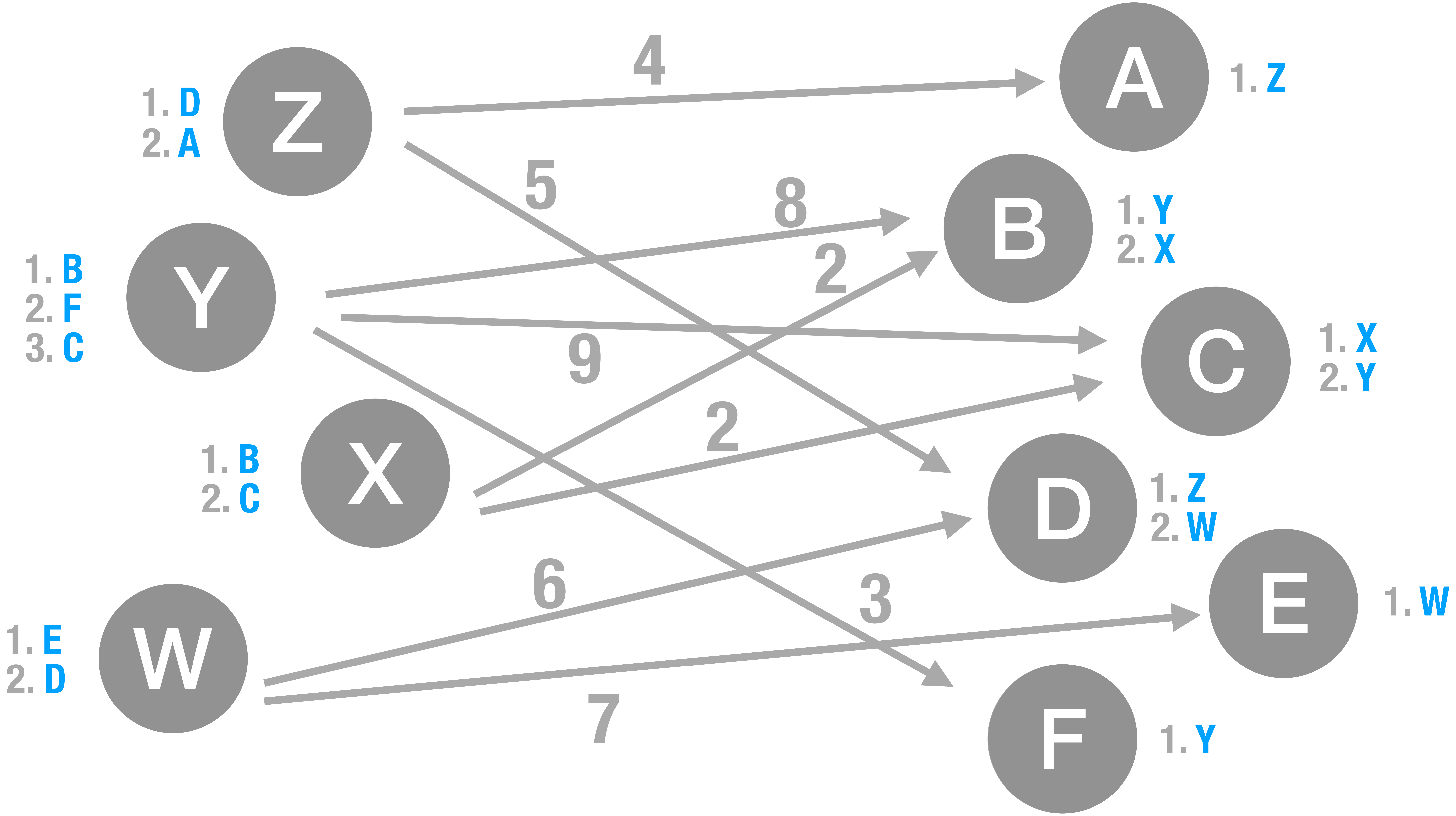
V_A



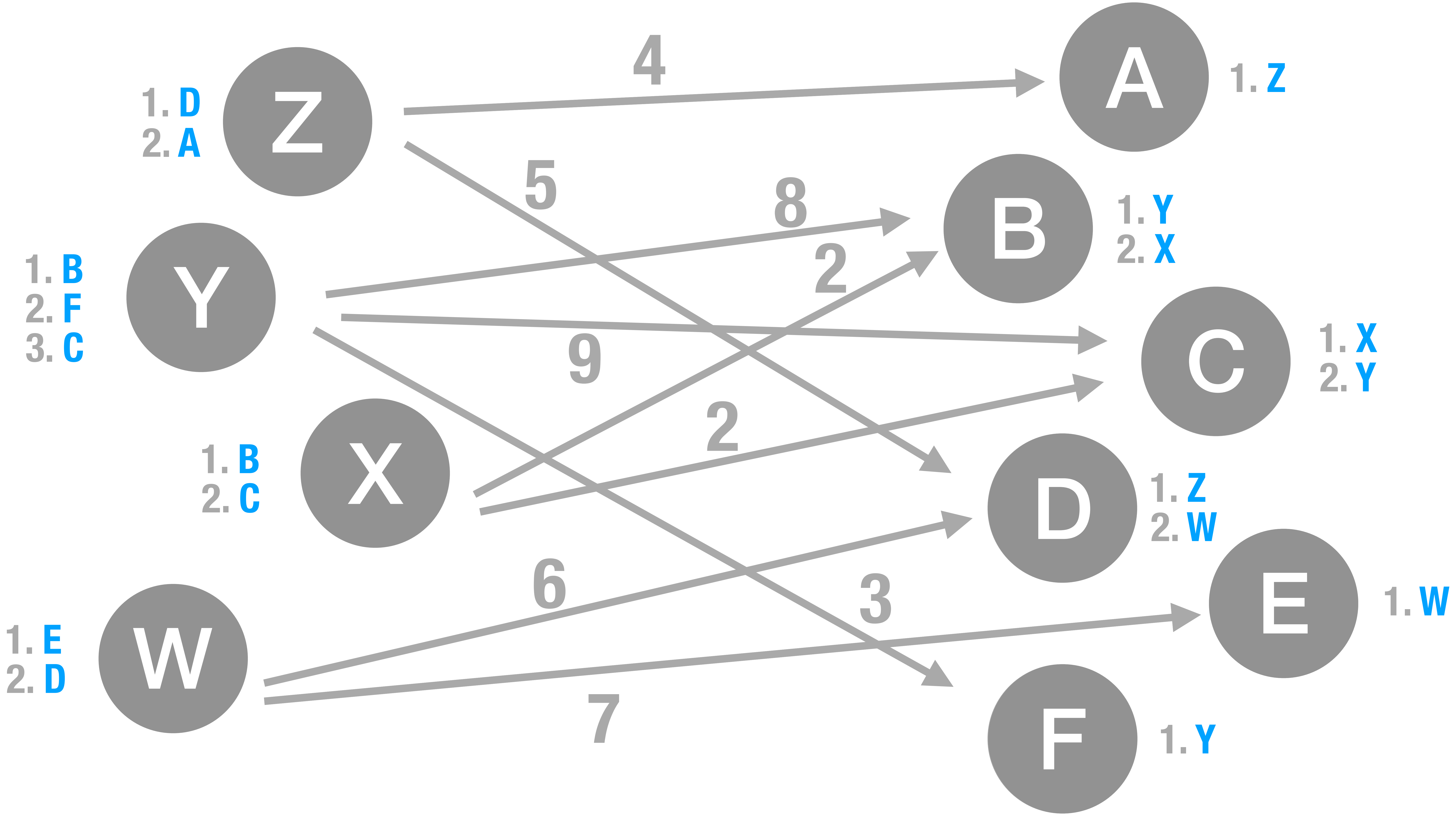
V_B



$$V_A \cap V_B = \emptyset$$



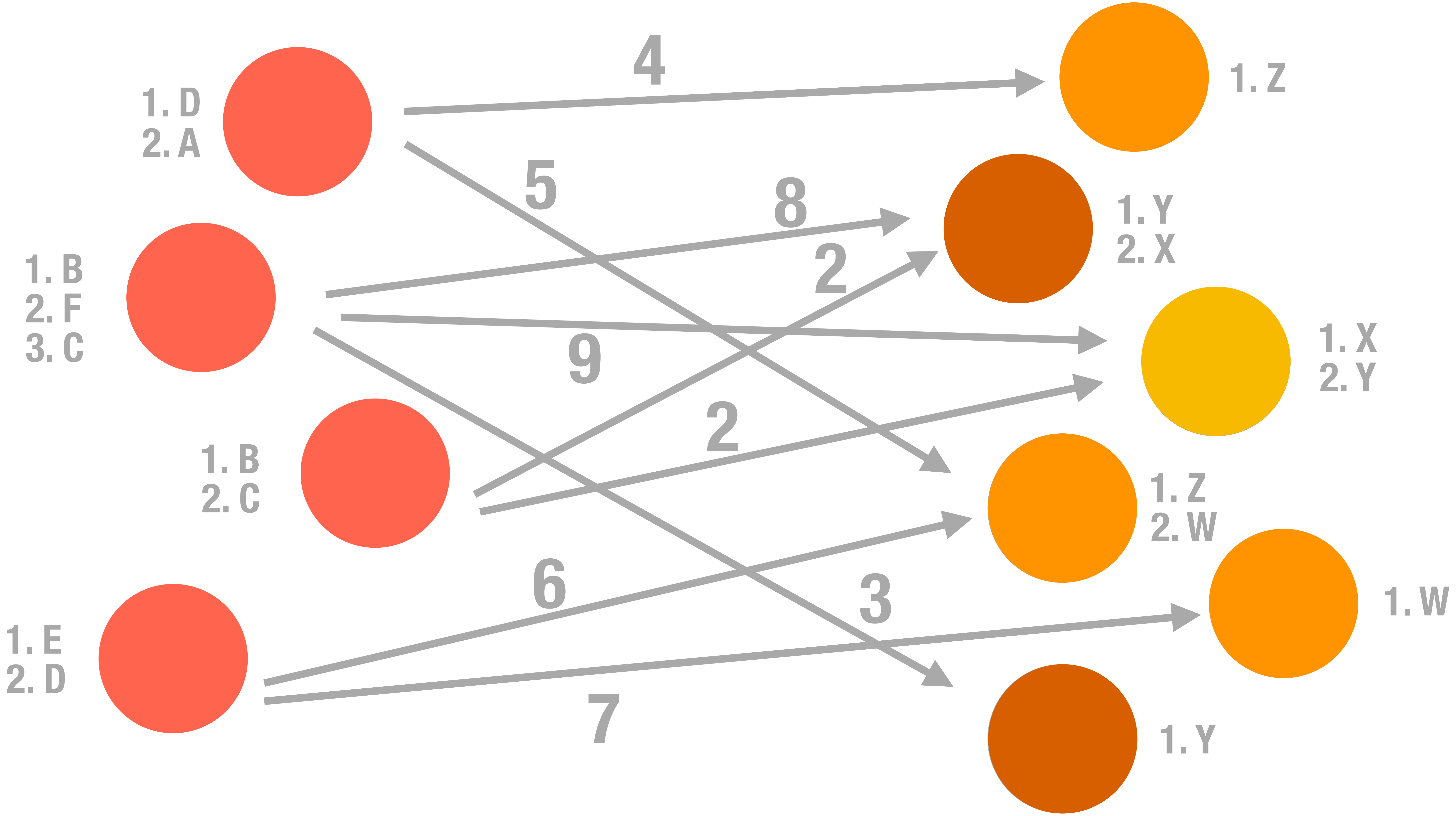
$$\forall v \in V \quad \exists p \in P$$



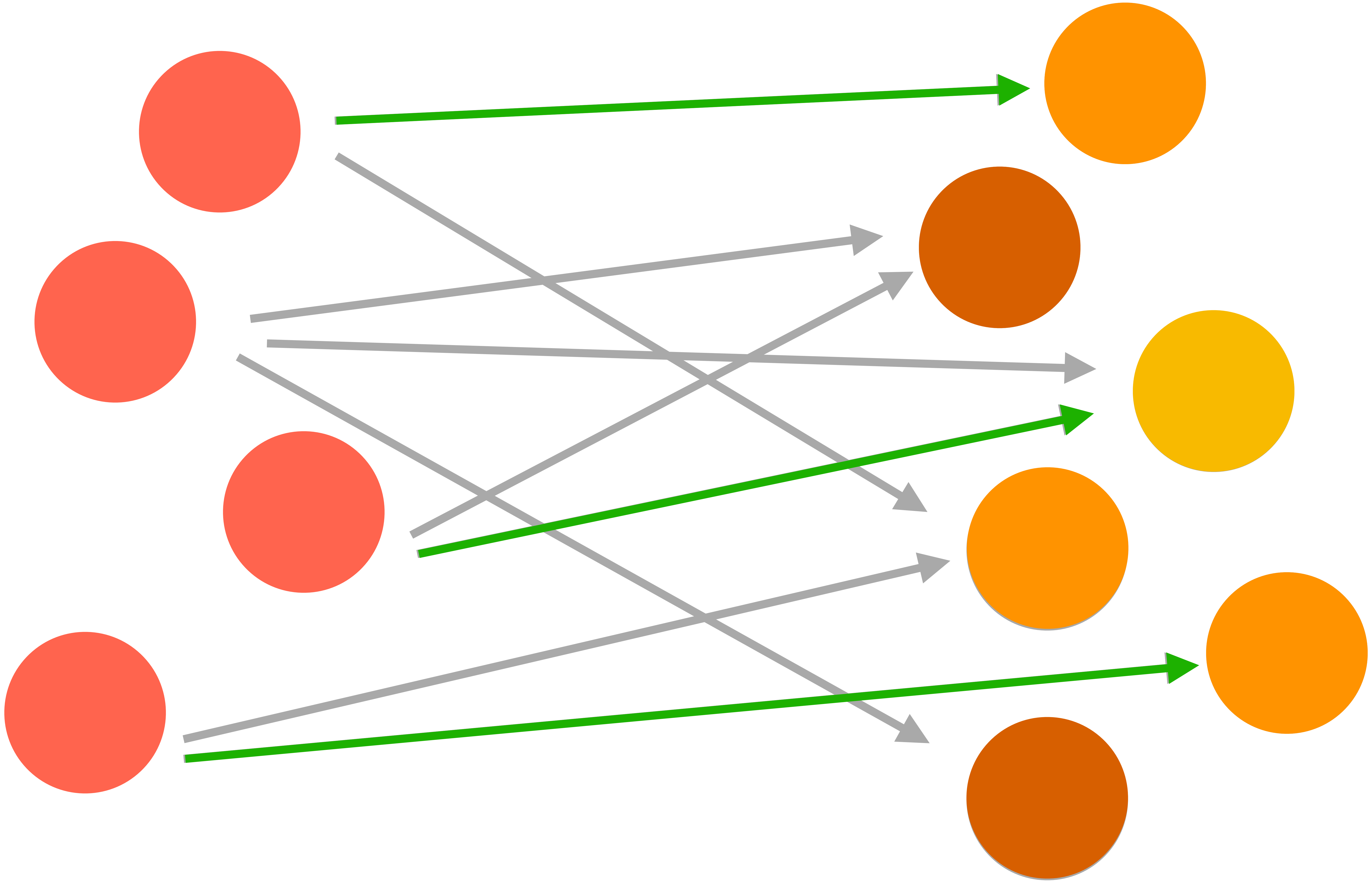
$$|p_i| \mathrel{\widehat{=}} |\delta^+(p_i)|$$

Matching

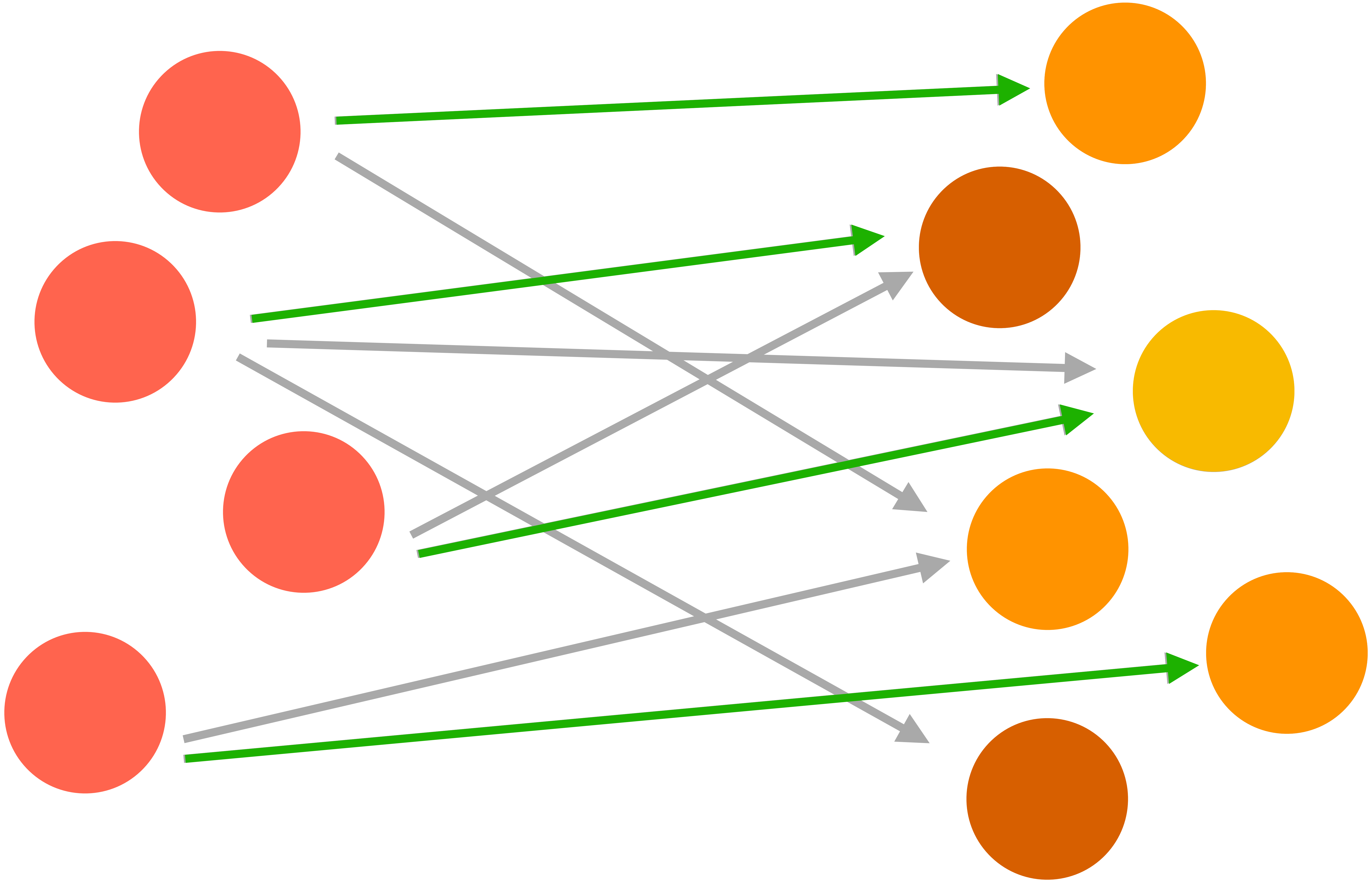
Definition



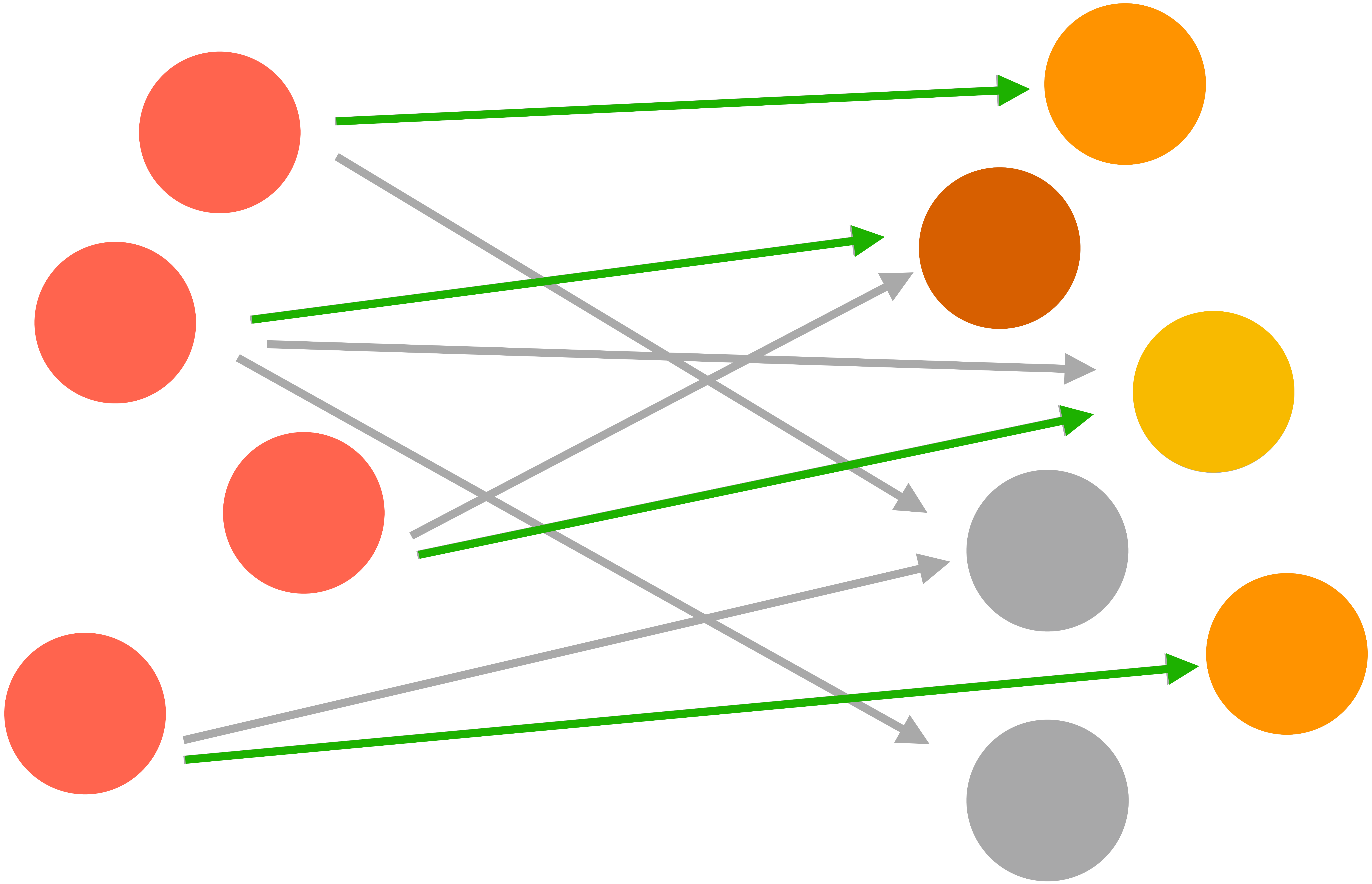
bipartiter Graph



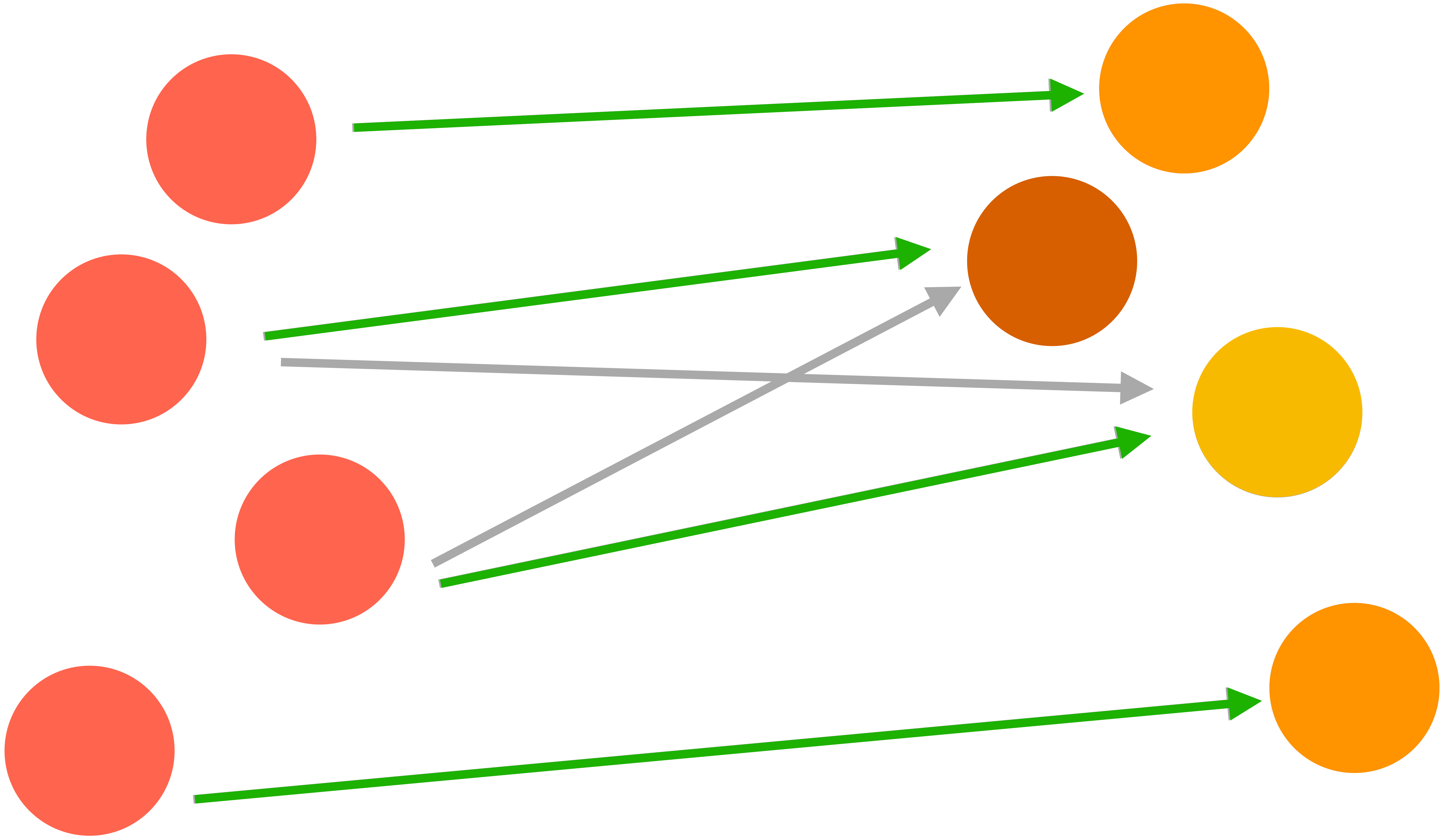
$$M = \{mat_1, \ldots, mat_n\}$$



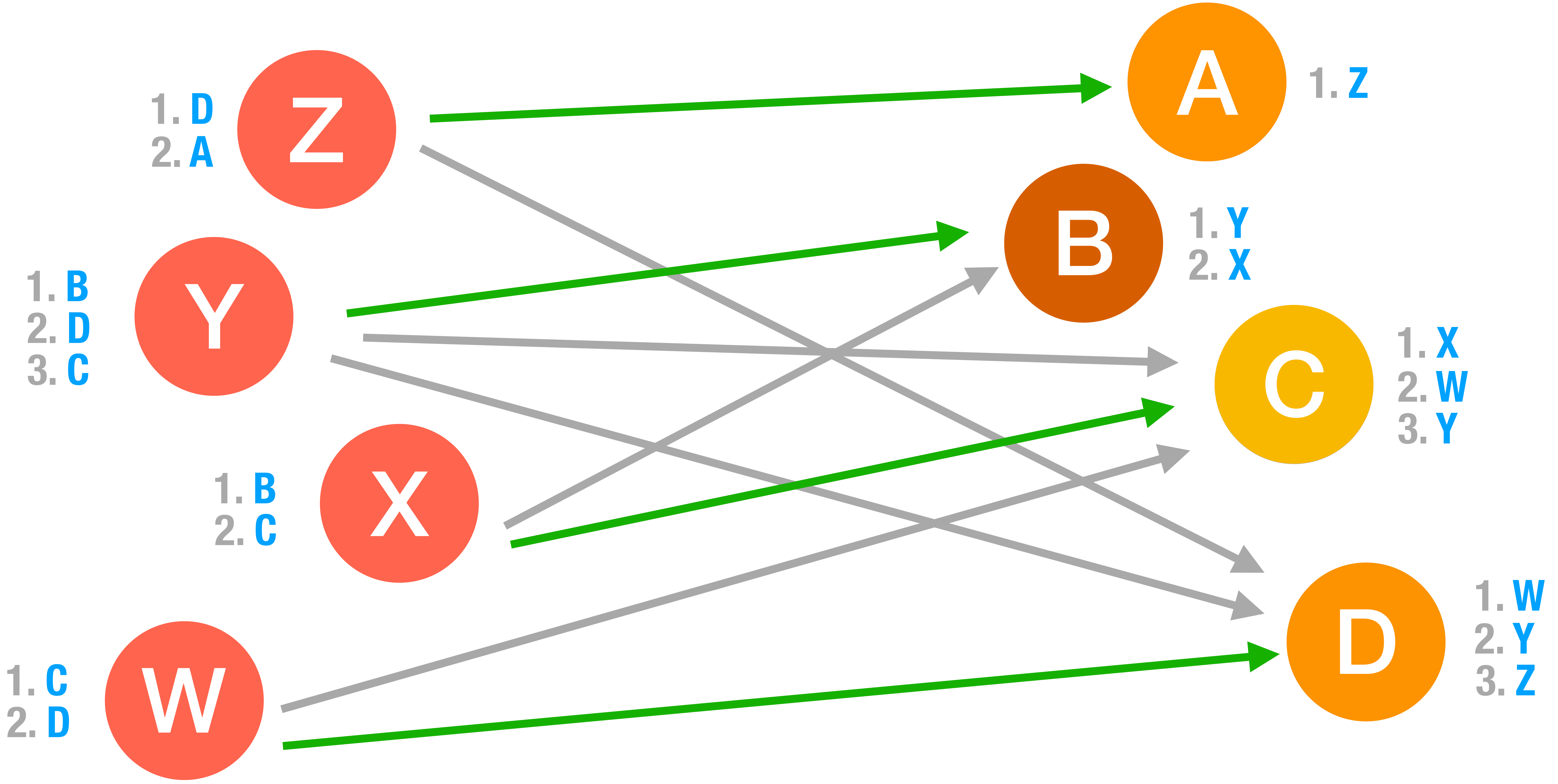
M_{max}



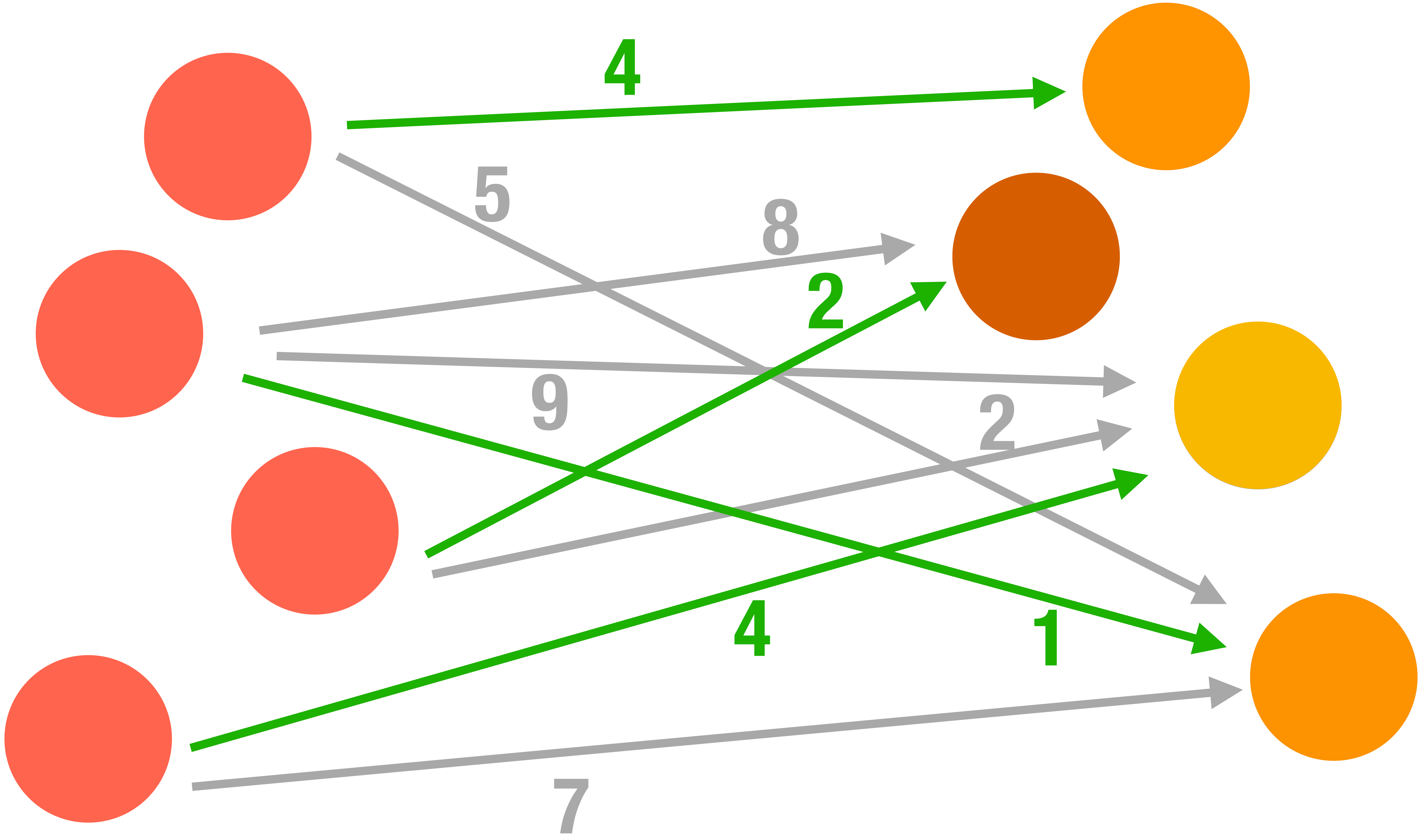
$$min(V_A, V_B)$$



$$|M_{per}| \;\widehat{=}\; |V_A| \;\widehat{=}\; |V_B|$$



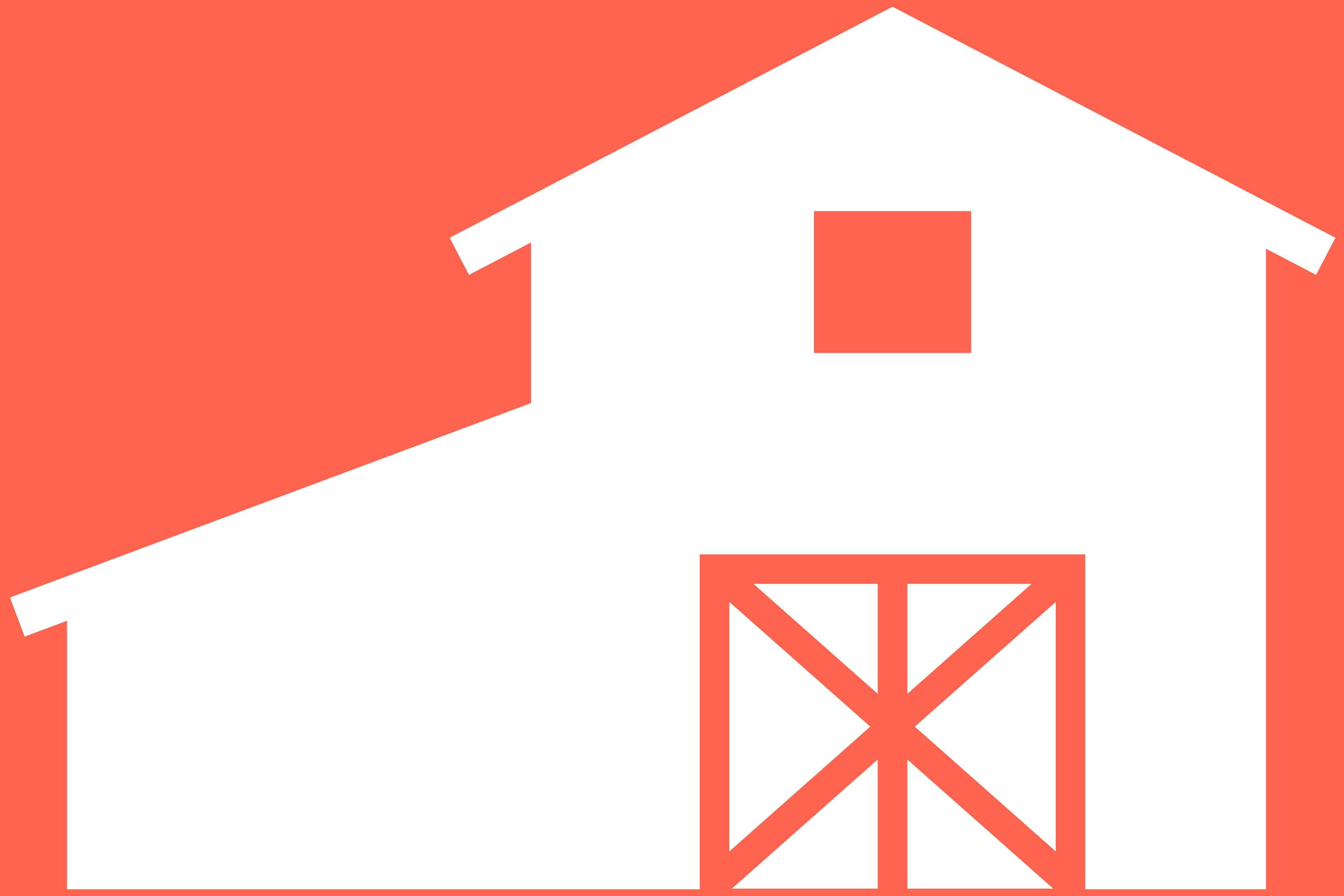
M_{sta}

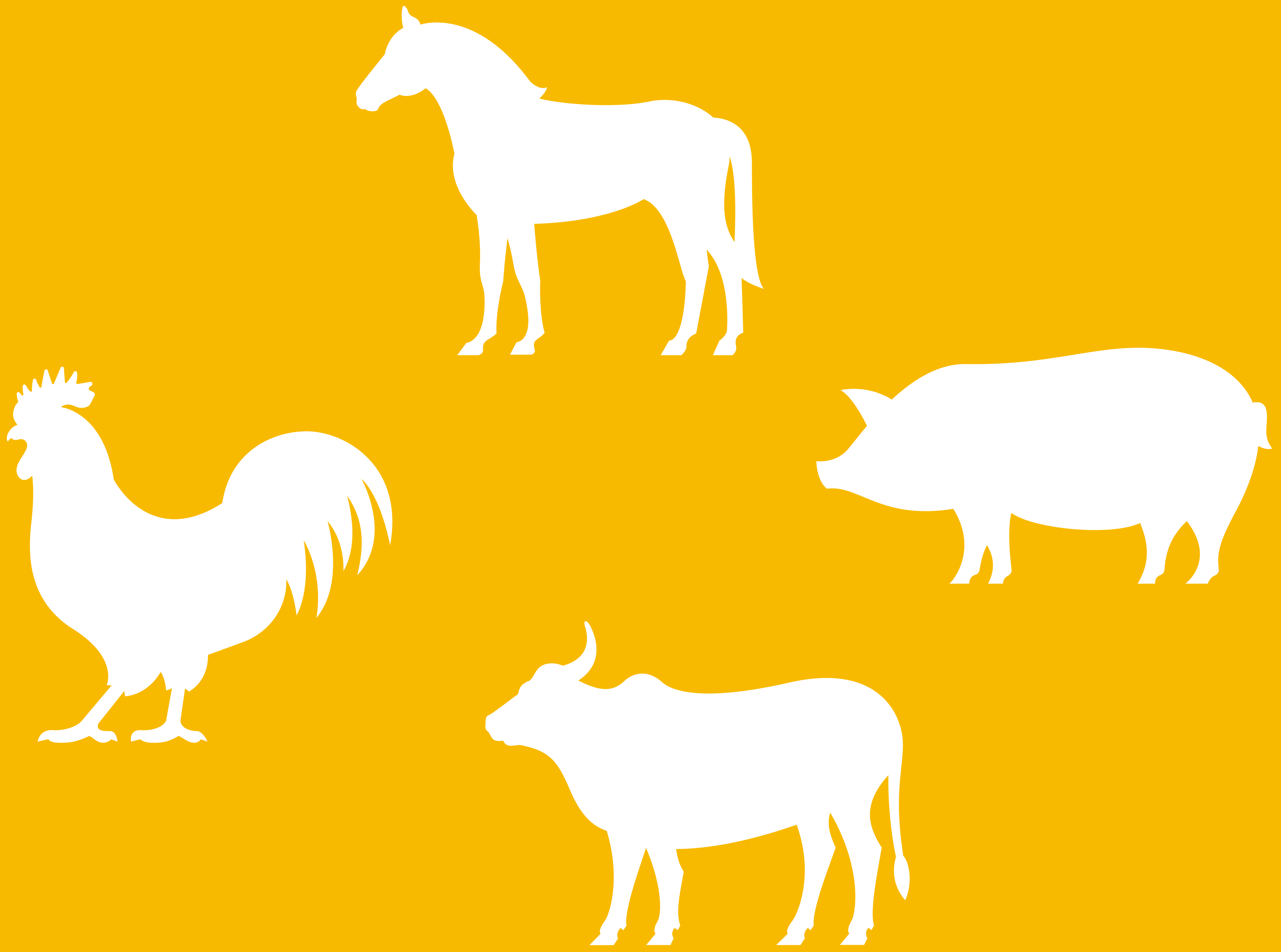


M_{ide}

Beschreibung

Problem





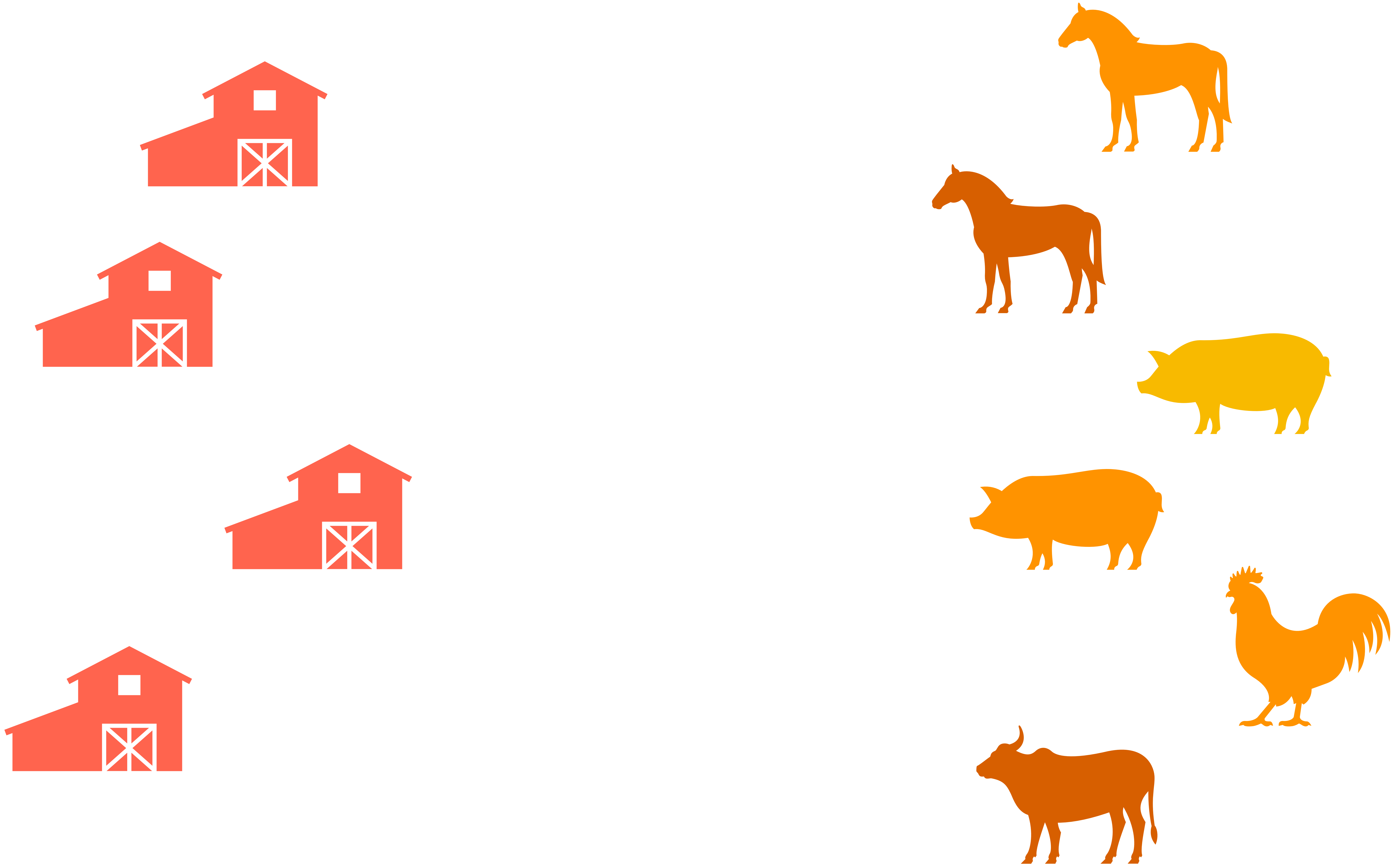
Stress

Priorisierung

alles gematcht

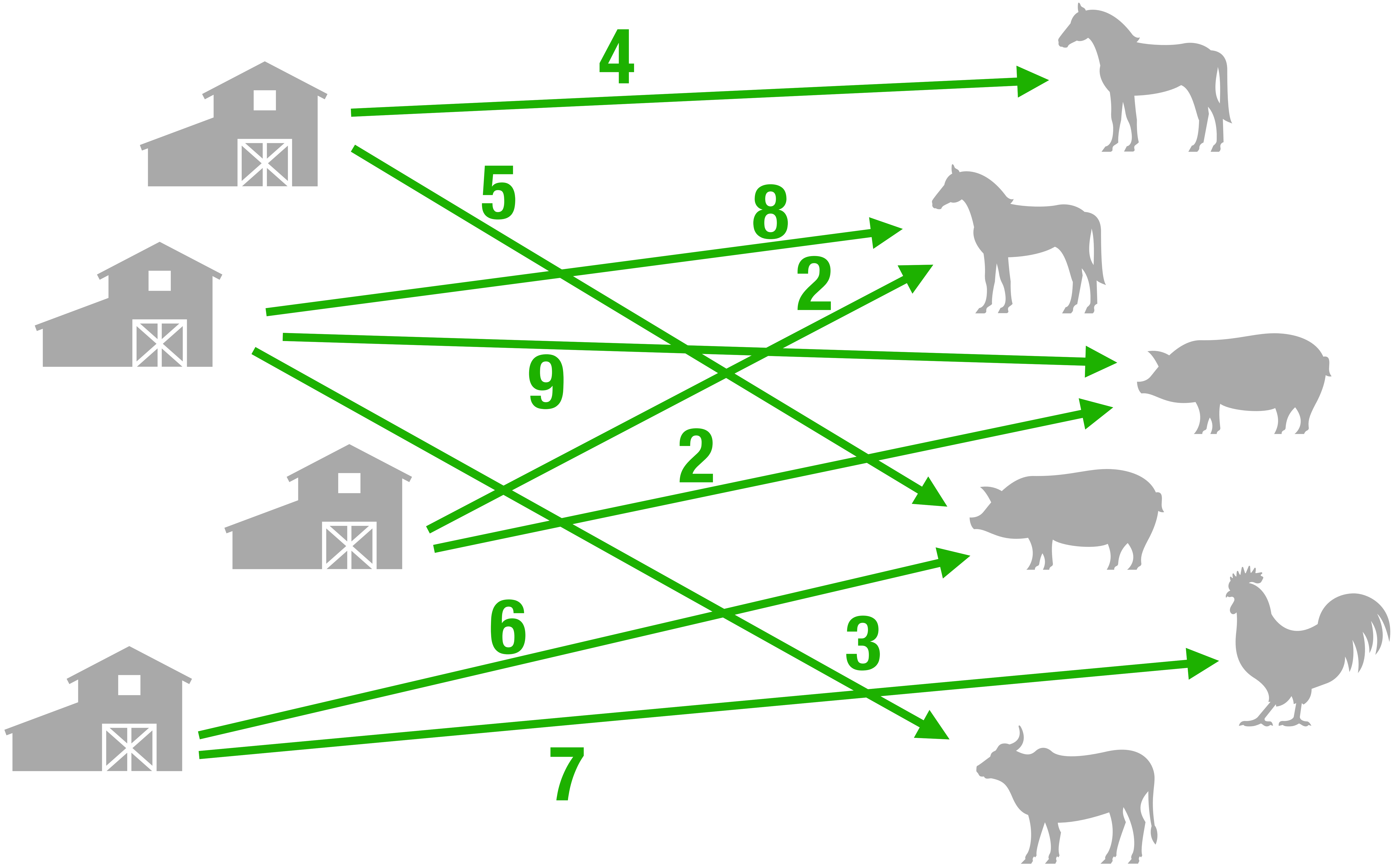
Definition

Problem

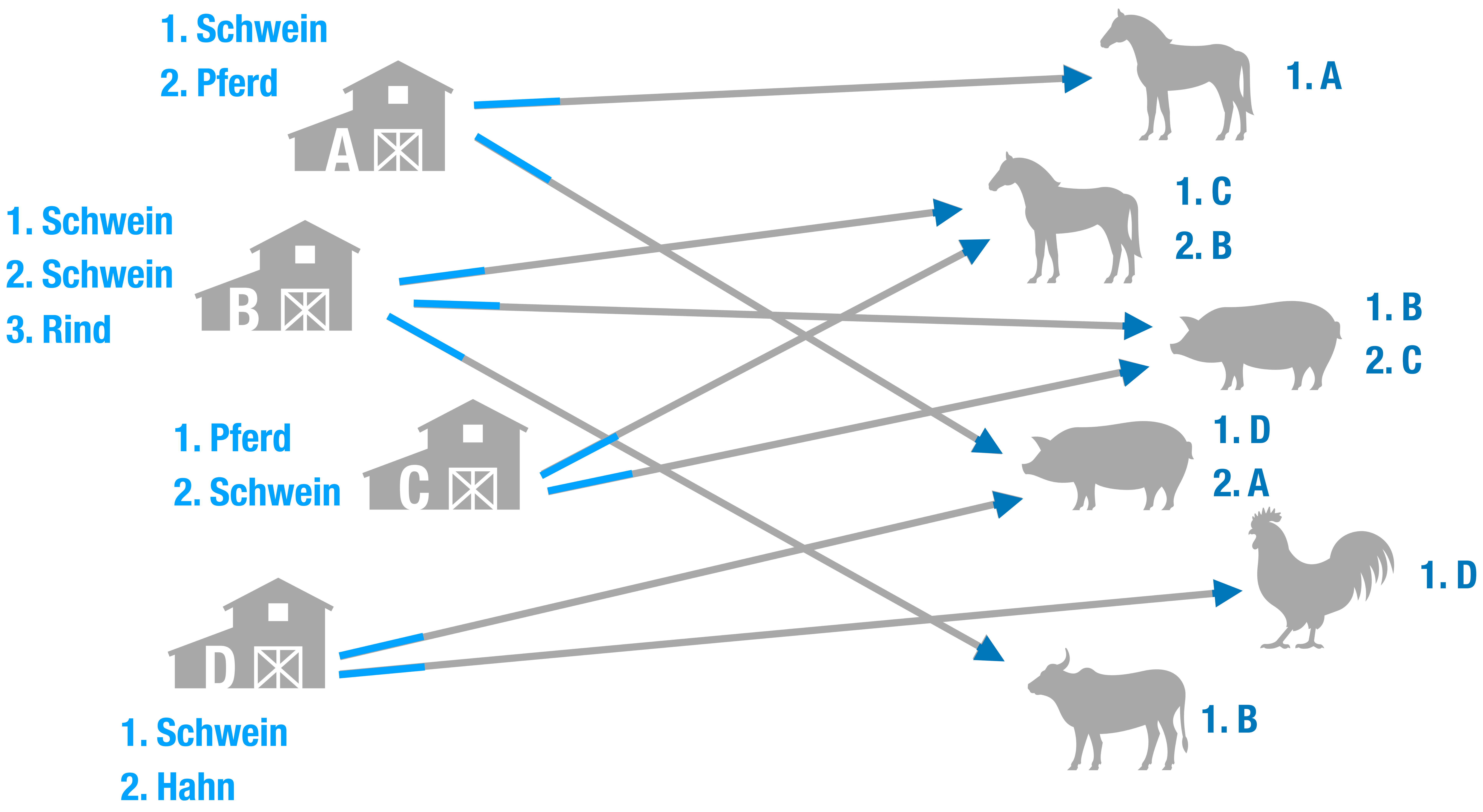


V_B

V_T



$$\forall e \in E \quad \exists w \in W$$



1. Schwein
2. Pferd

1. Schwein
2. Schwein
3. Rind

1. Pferd
2. Schwein

1. Schwein
2. Hahn

1. A

1. C
2. B

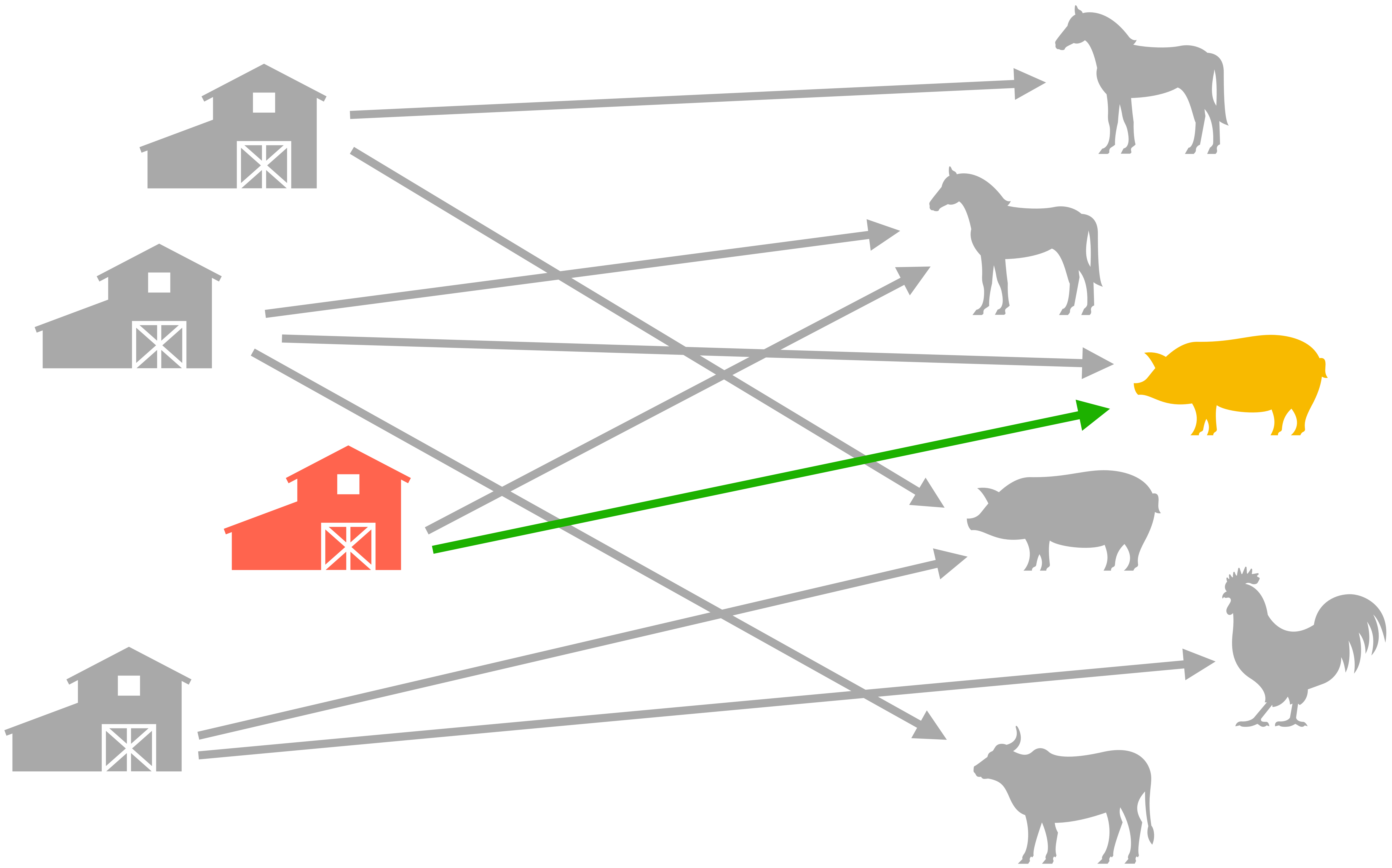
1. B
2. C

1. D
2. A

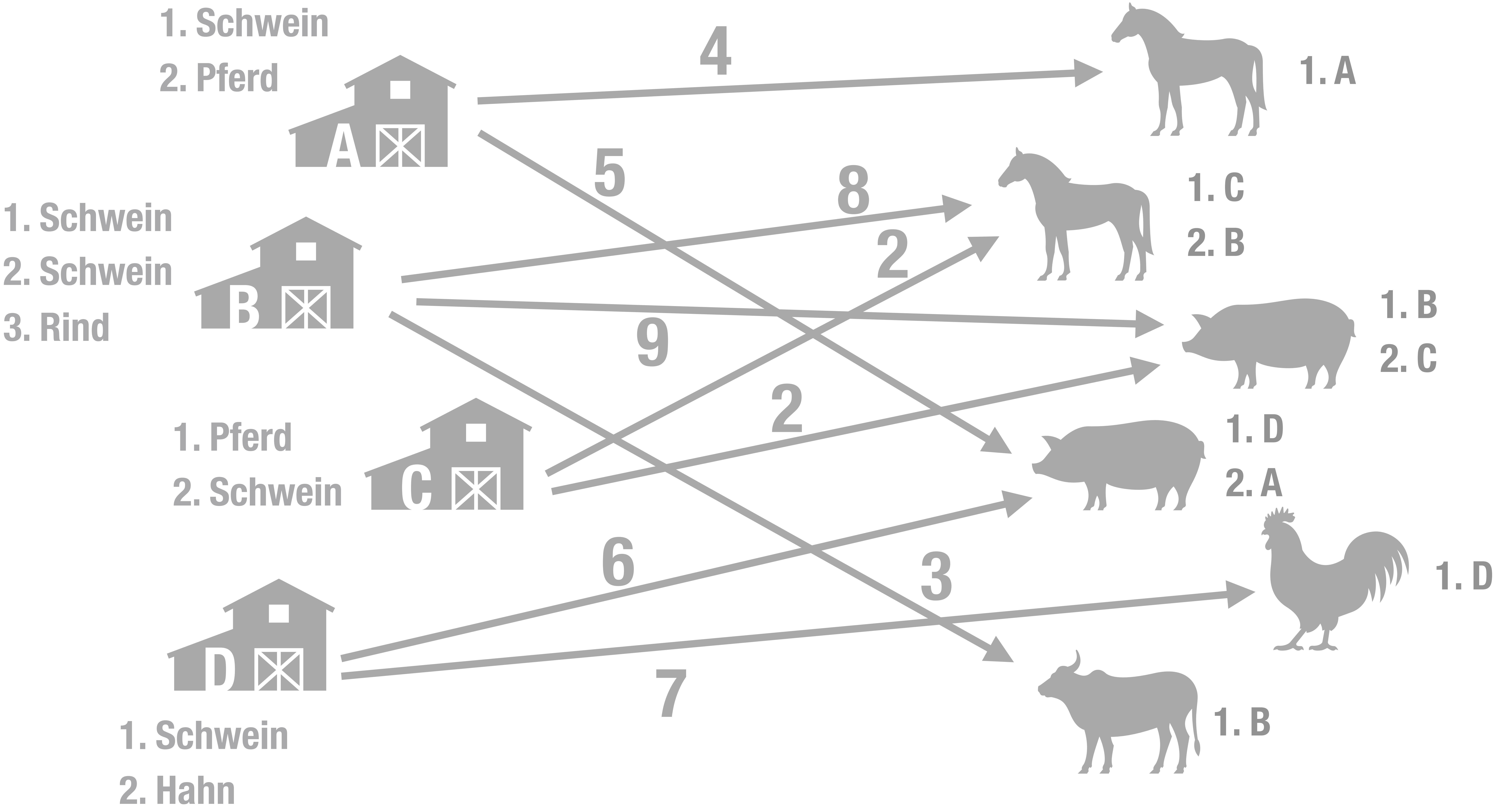
1. D

1. B

$$|p_i| \mathrel{\widehat{=}} |\delta^+(b_i)|$$



$$mat_x := \overline{b_i t_j}$$



1. Schwein
2. Pferd

1. Schwein
2. Schwein
3. Rind

1. Pferd
2. Schwein

1. Schwein
2. Hahn

1. A

1. C
2. B

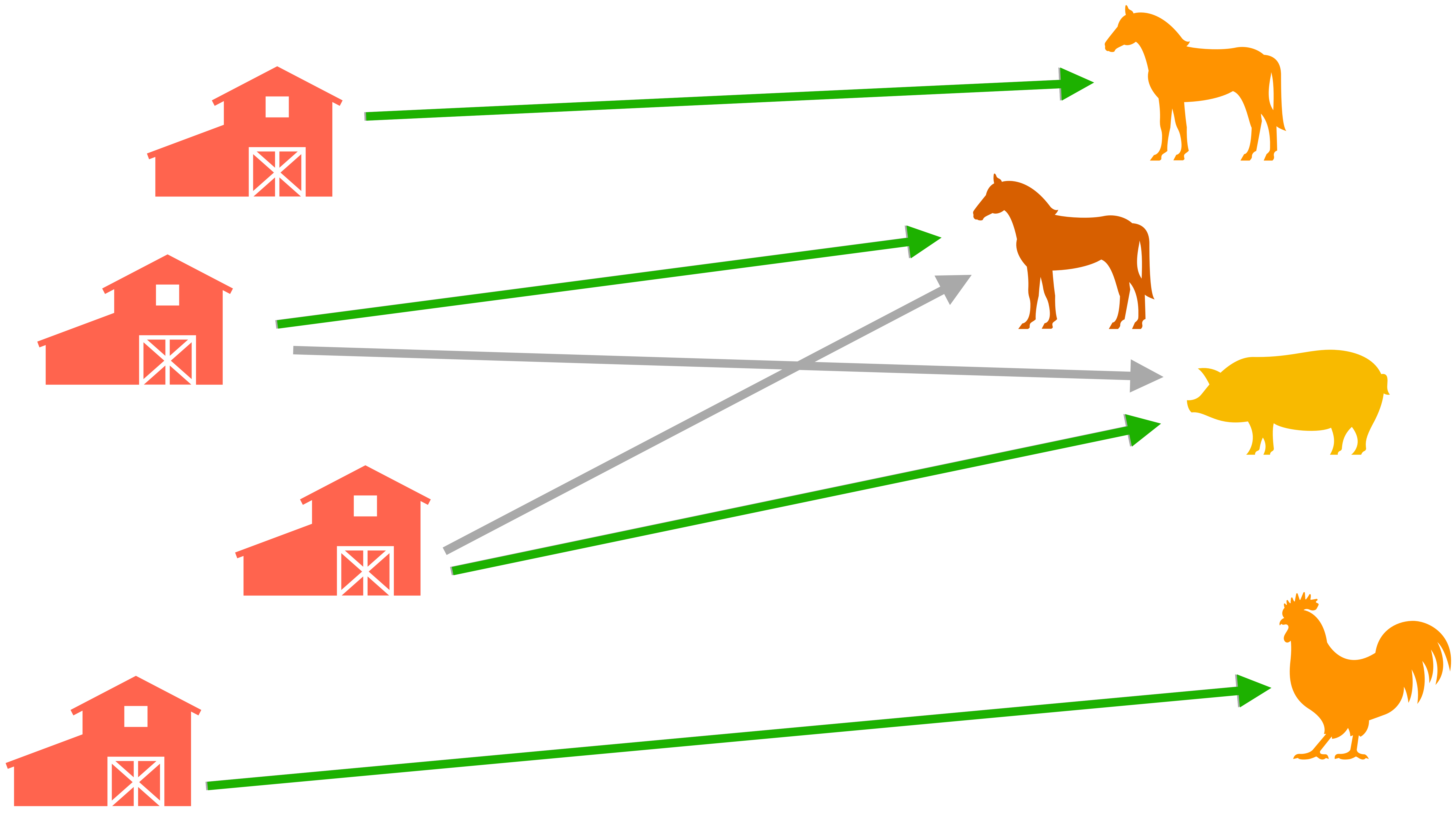
1. B
2. C

1. D
2. A

1. D

1. B

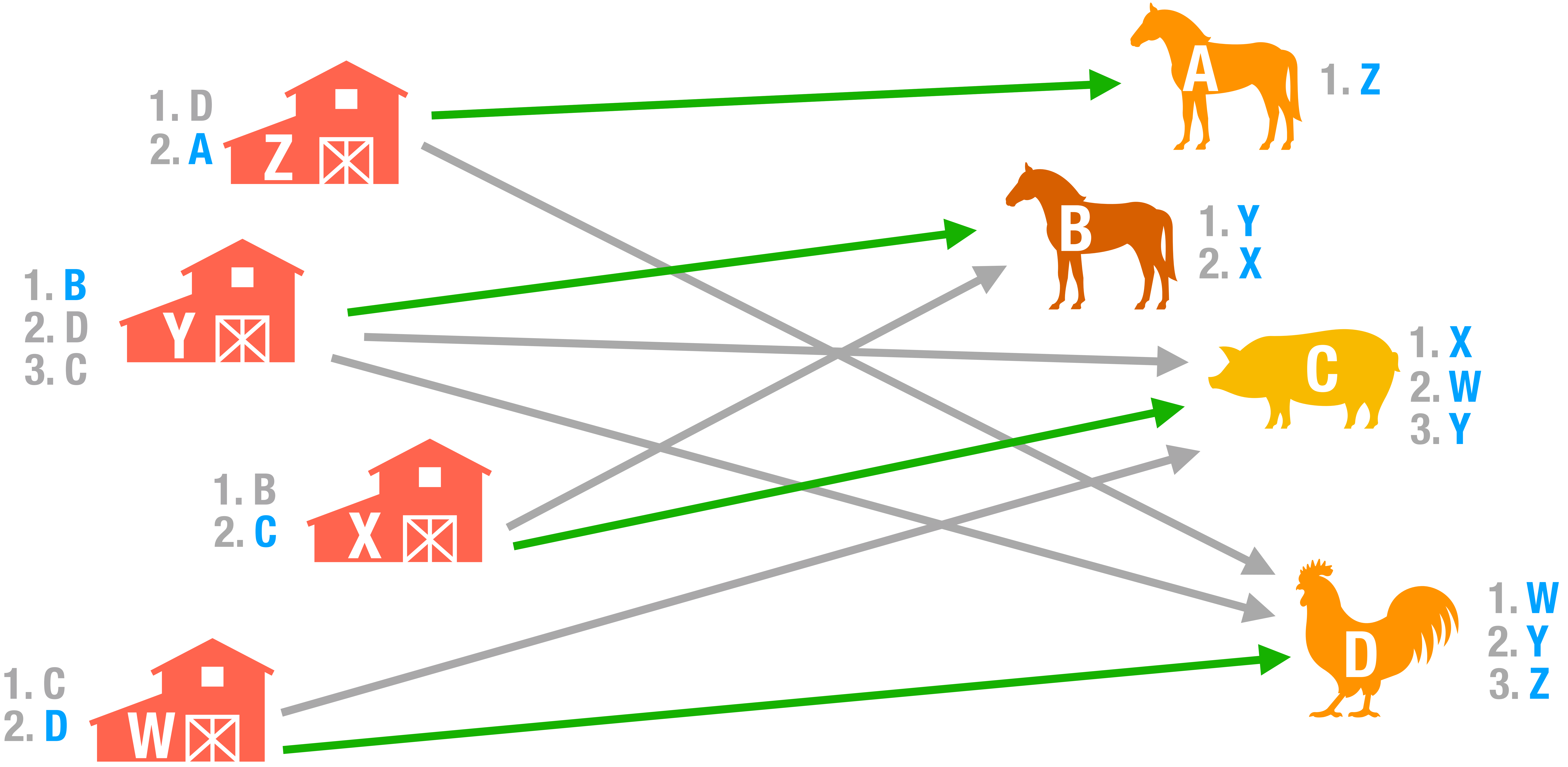
Perfektes Matching



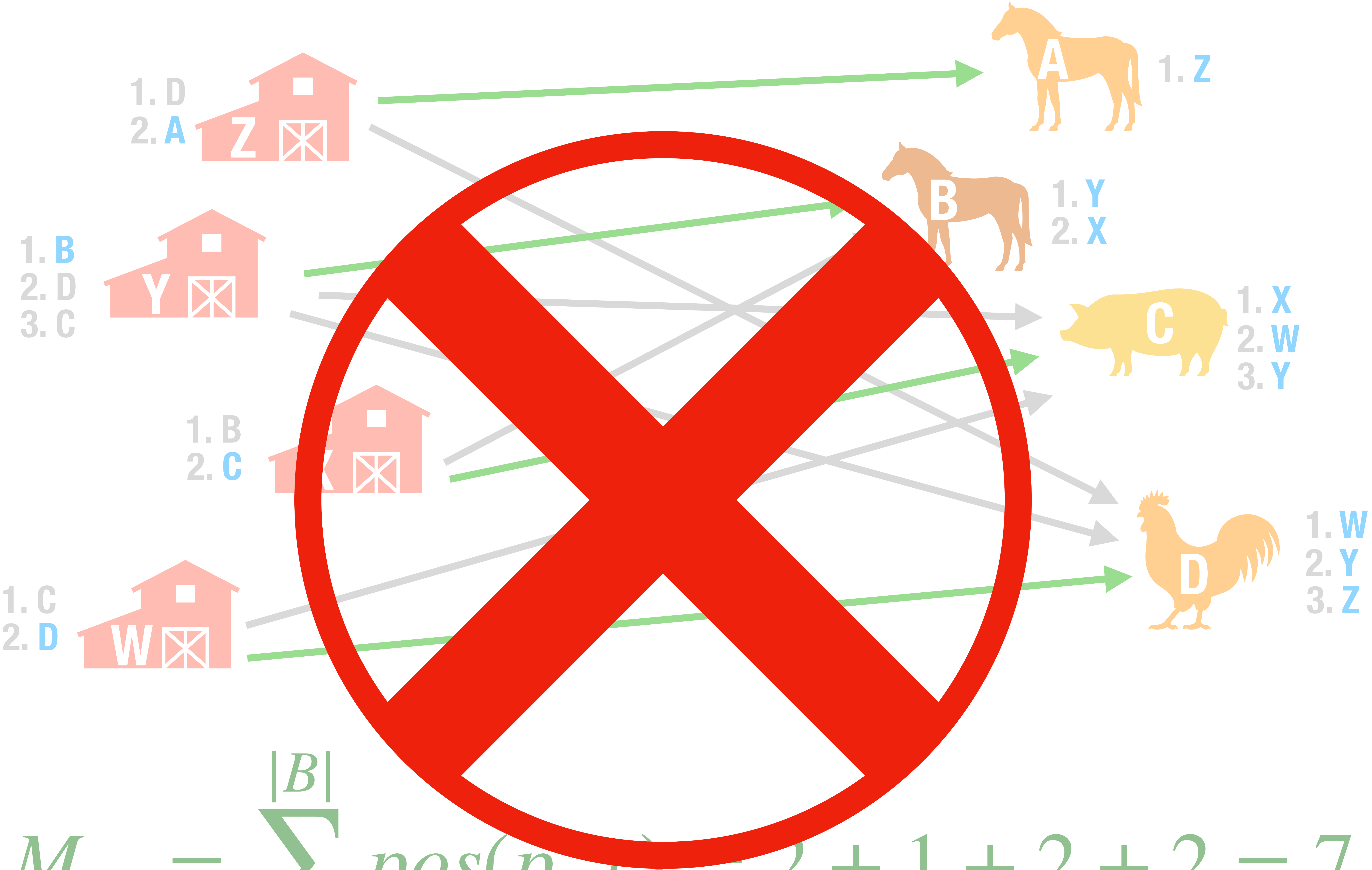
$$|M_{per}| \mathrel{\widehat{=}} |V_A| \mathrel{\widehat{=}} |V_B|$$

$$\sum_{i=1}^{|B|} pos(p_i, t_j) \quad \textbf{mit} \quad mat_{b_i t_j}$$

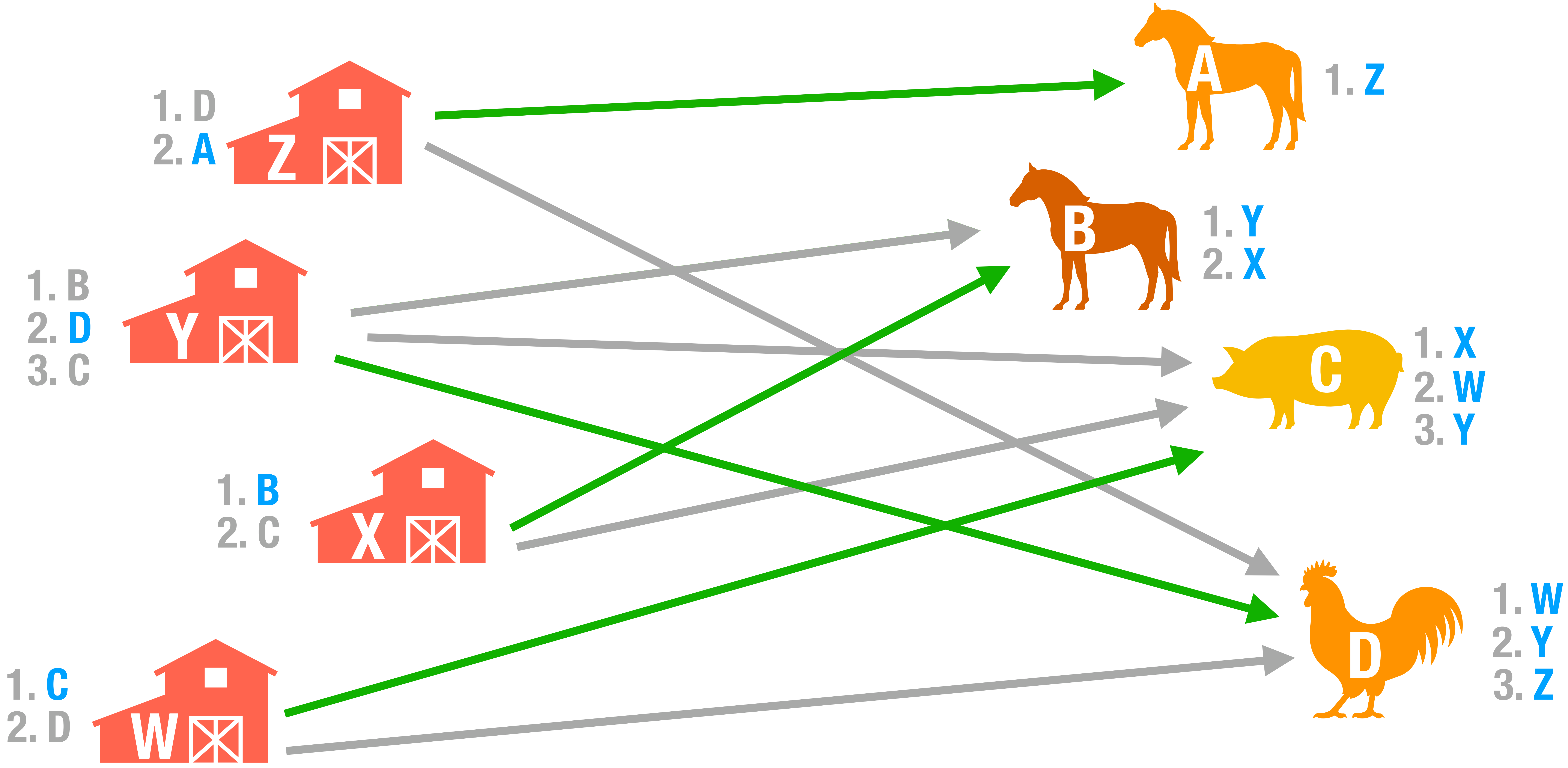
Stabilitätsmaß



$$M_{sta} = \sum_{i=1}^{|B|} pos(p_i, t_j) = 2 + 1 + 2 + 2 = 7$$



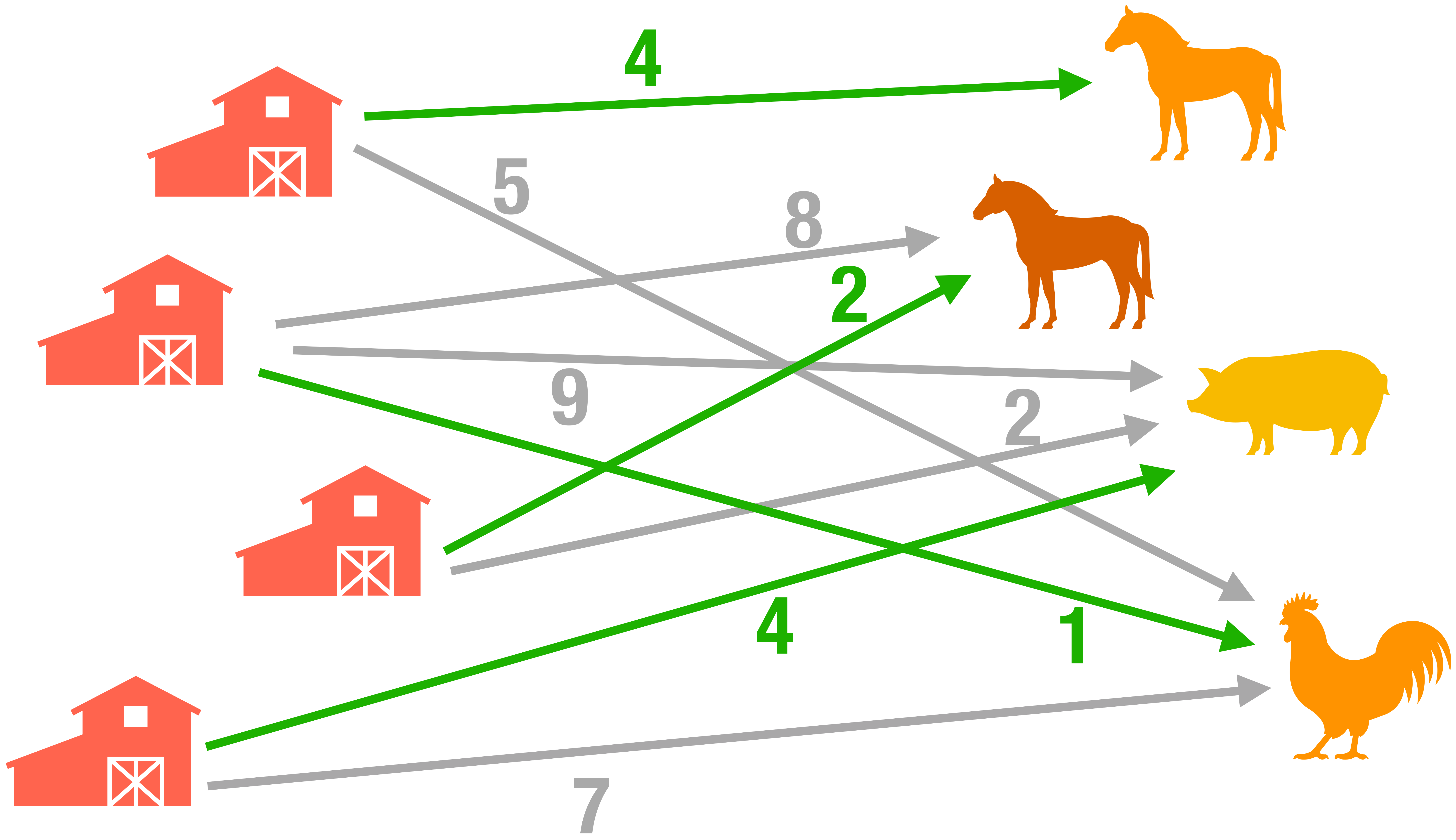
$$M_{sta} = \sum_{i=1}^{|B|} pos(p_i, t_j) = 2 + 1 + 2 + 2 = 7$$



$$M_{sta} = \sum_{i=1}^{|B|} pos(p_i, t_j) = 2 + 2 + 1 + 1 = 6$$

$$\sum_{i=1}^{|B|} w_{mat_i}$$ **minimal**

Idealitätsmaß



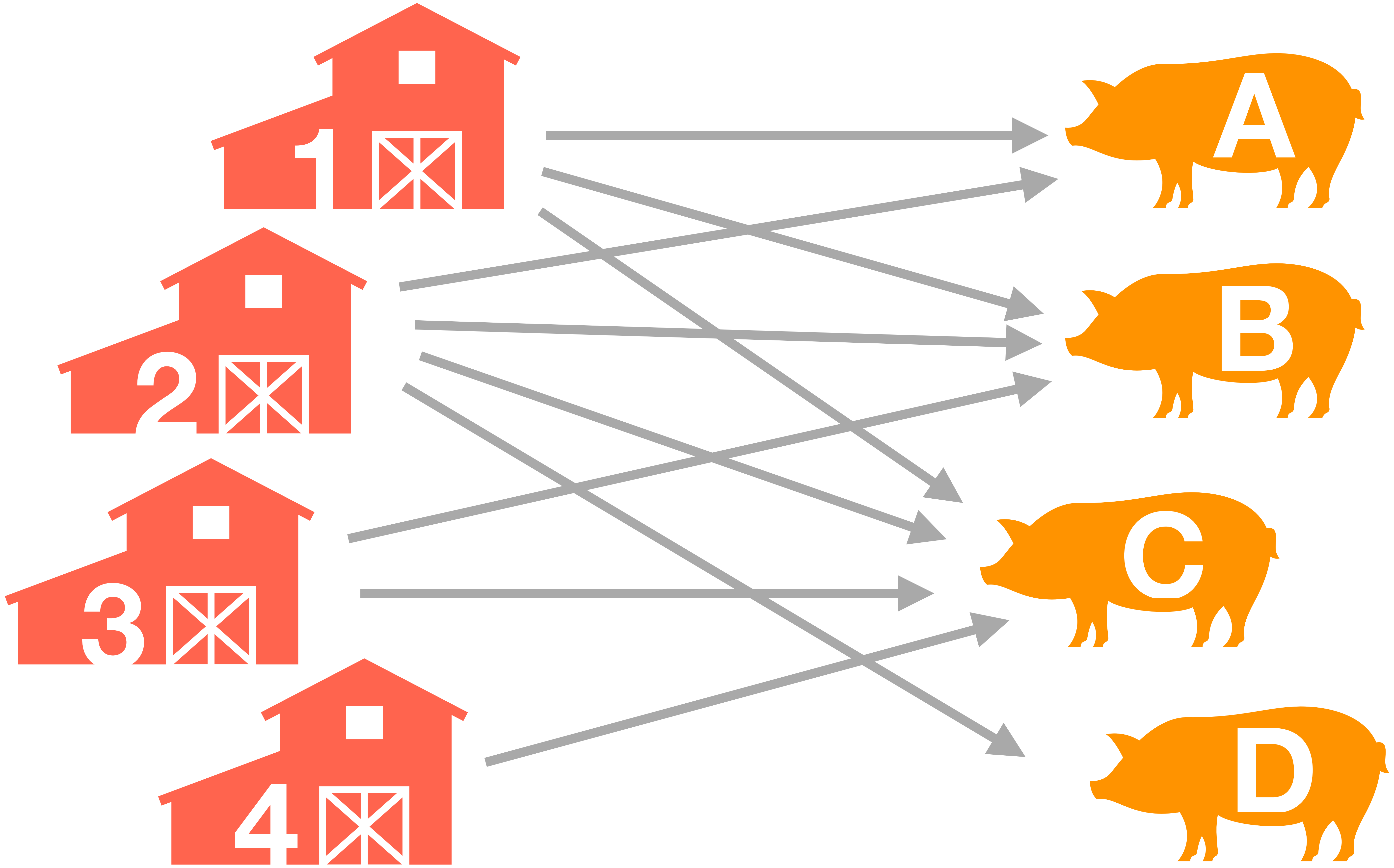
$$M_{ide} = \sum_{i=1}^{|B|} w_{mat_i} = 4 + 2 + 4 + 1 = 11$$

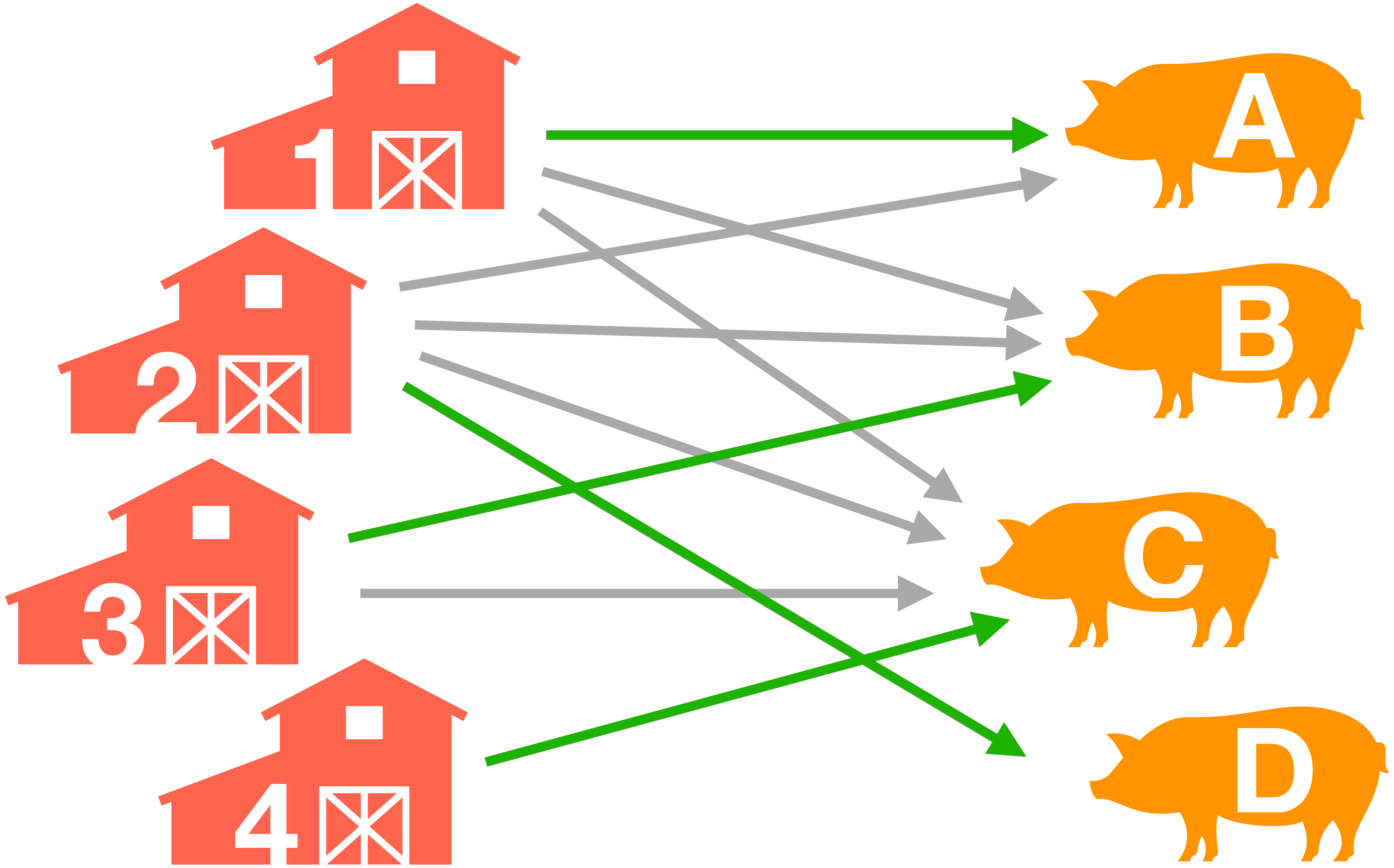
Herausforderung

Problem

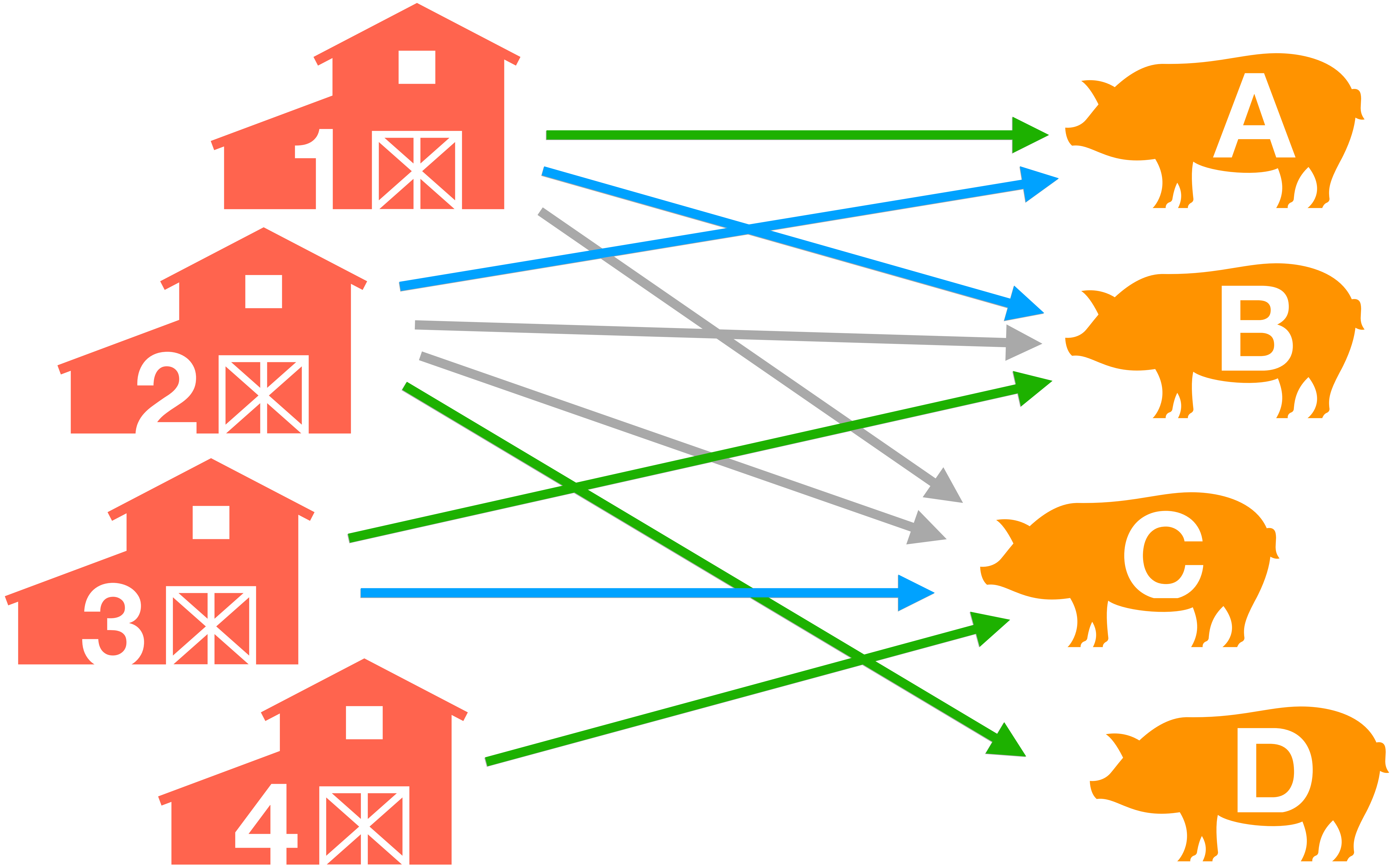
perfekt <> stabil

Widerspruch


1
2
3
4
A
B
C
D



	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			



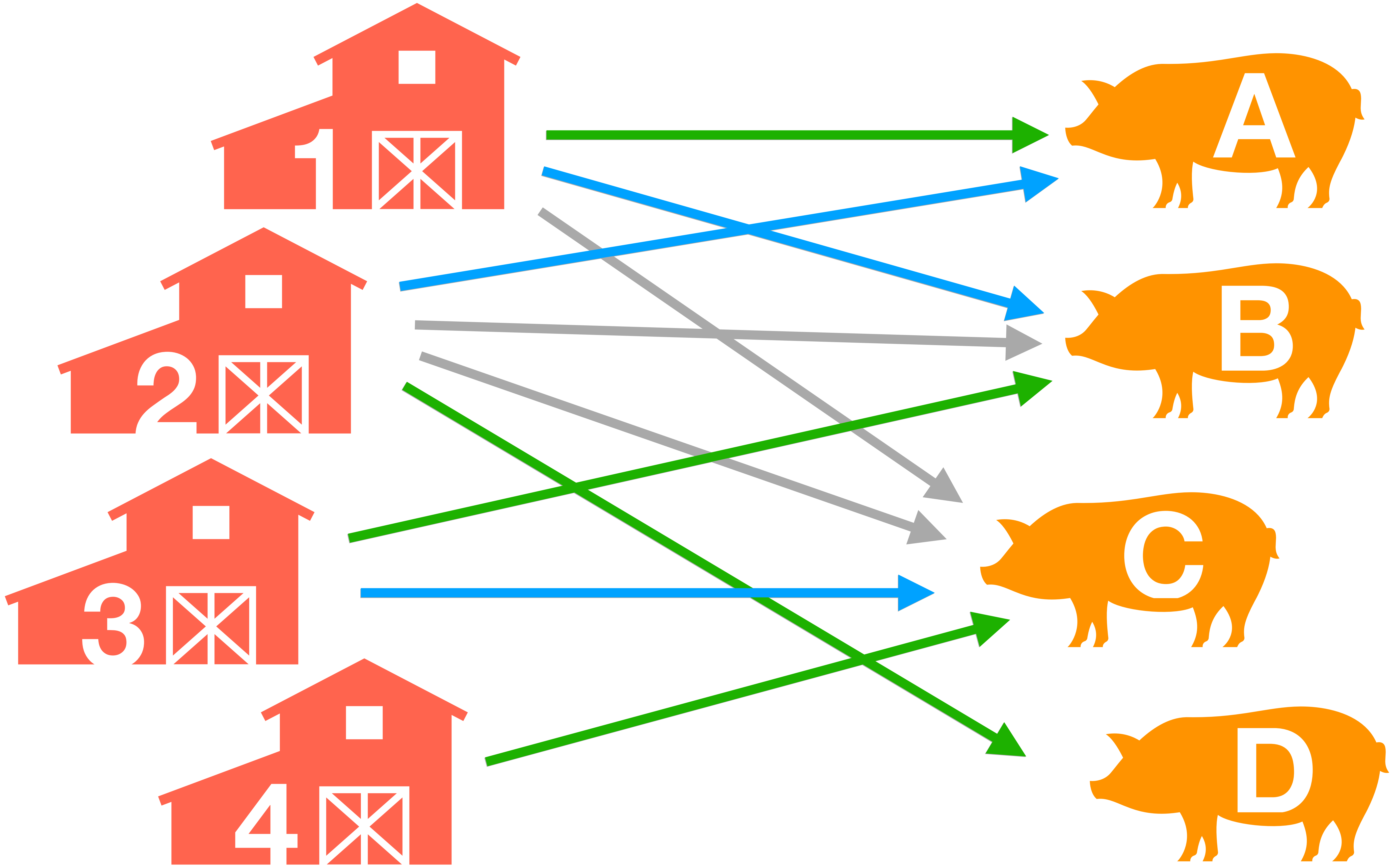
	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			

	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			

$$\sum_{i=1}^{|B|} pos(p_i, t_i)$$

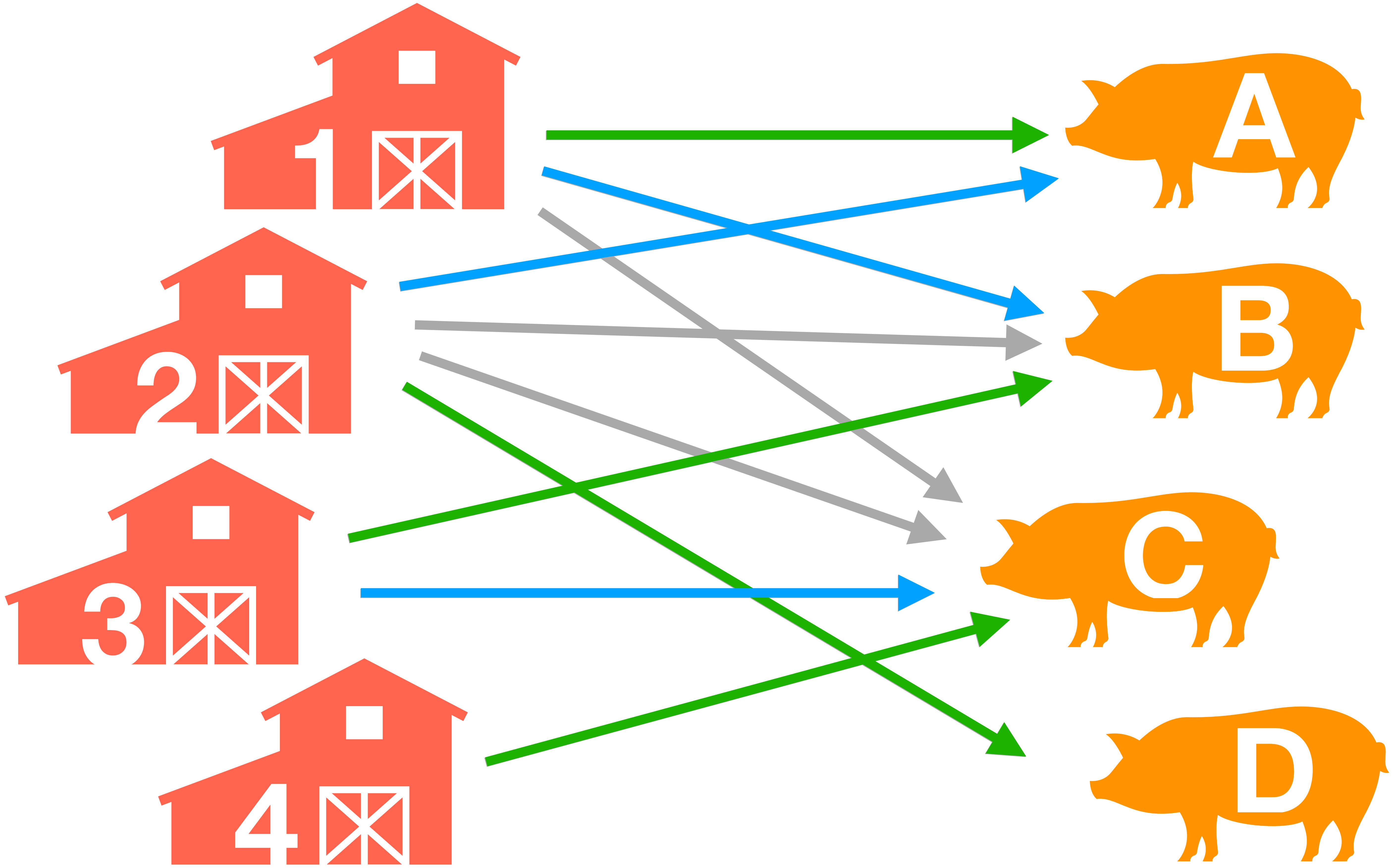
$$= 3 + 4 + 2 + 1$$

$$= 10$$



	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			

	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			



$$\sum_{i=1}^{|B|} pos(p_i, t_i)$$
$$= 3 + 4 + 2 + 1$$
$$= 10$$

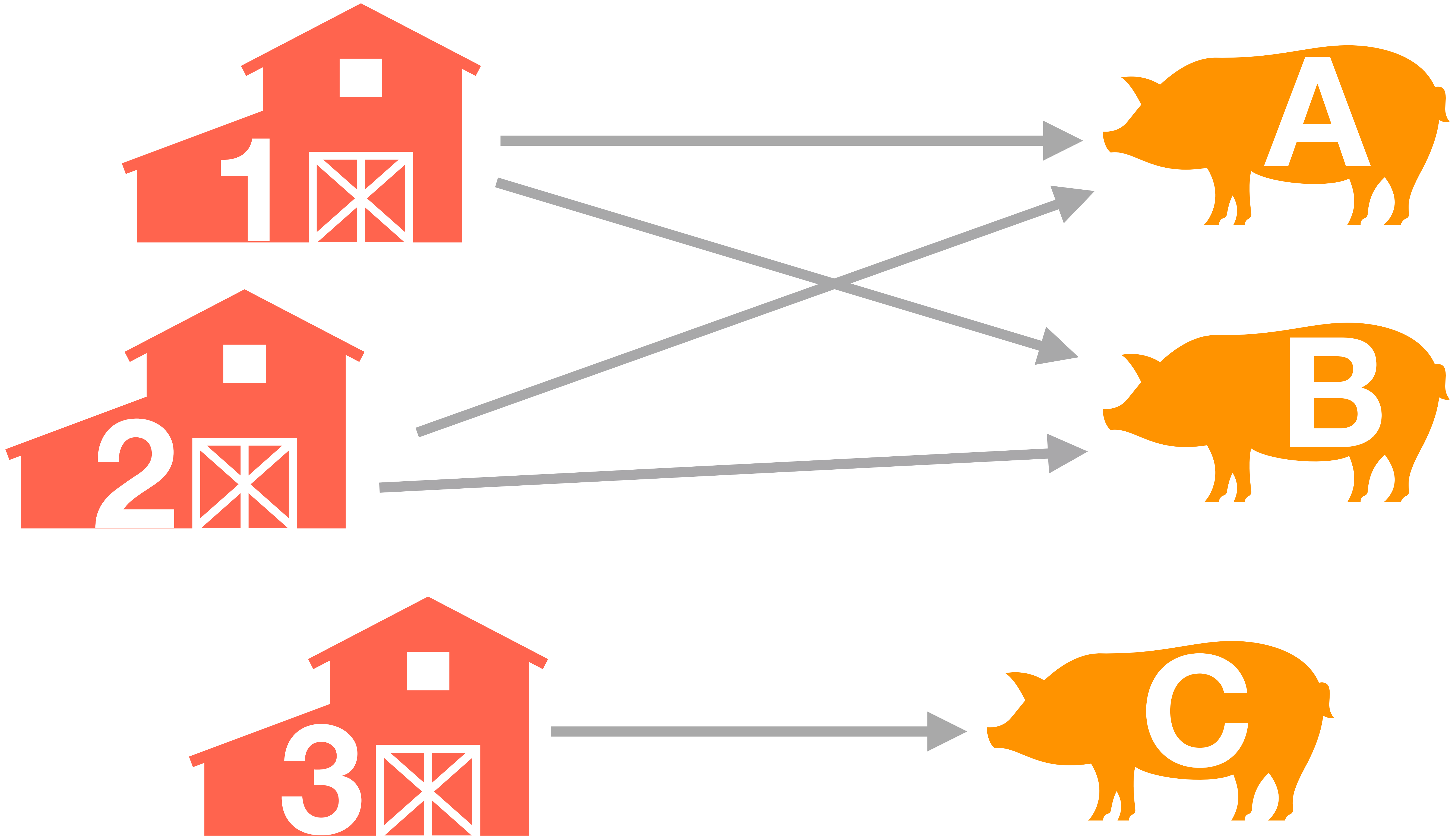
	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			

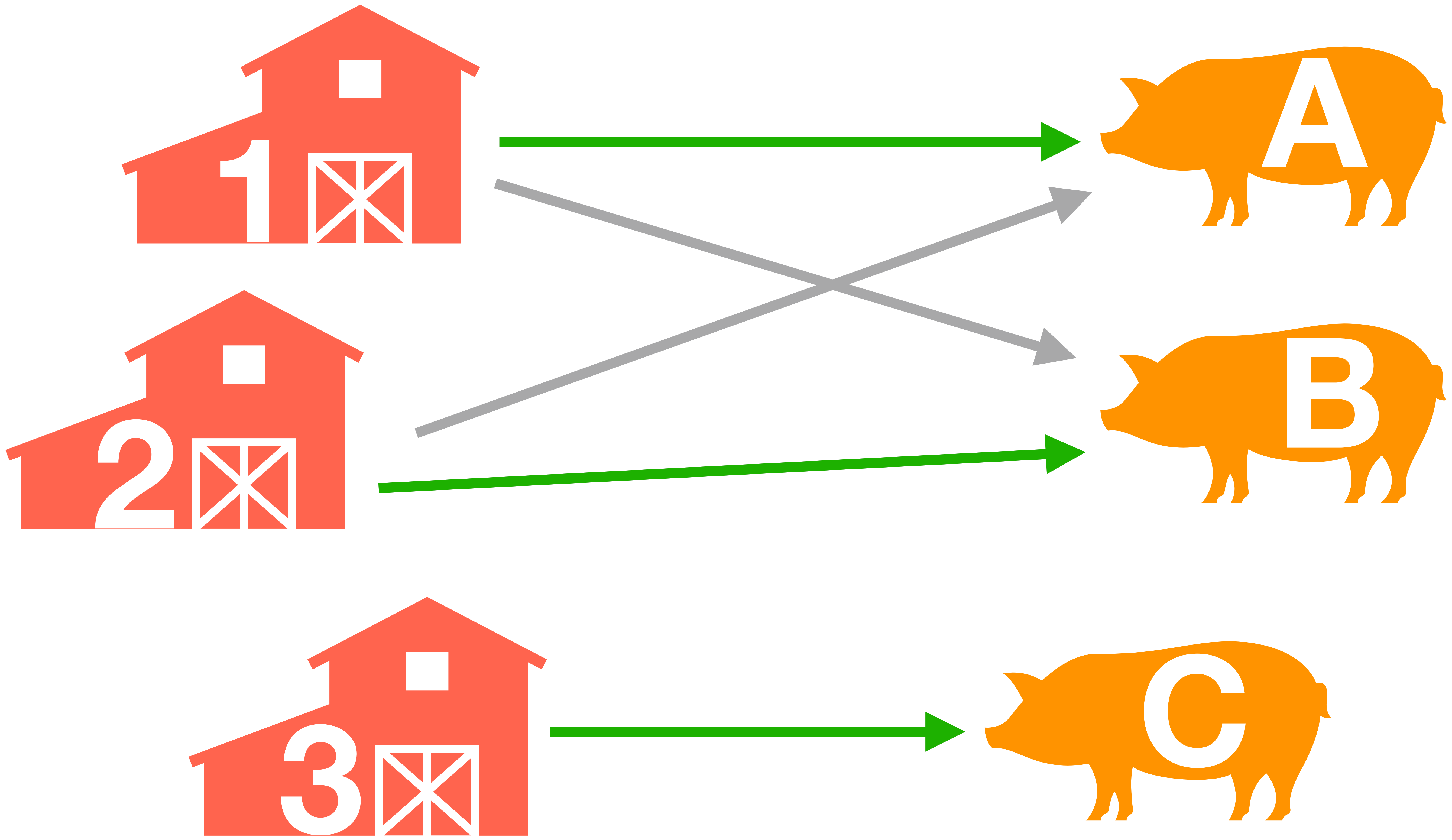
$$\sum_{i=1}^{|B|} pos(p_i, t_i)$$
$$= 1 + 1 + 1(+5)$$
$$= 3$$
$$(= 8)$$

	1. Prio	2. Prio	3. Prio	4. Prio
1	B	C	A	
2	A	B	C	D
3	C	B		
4	C			

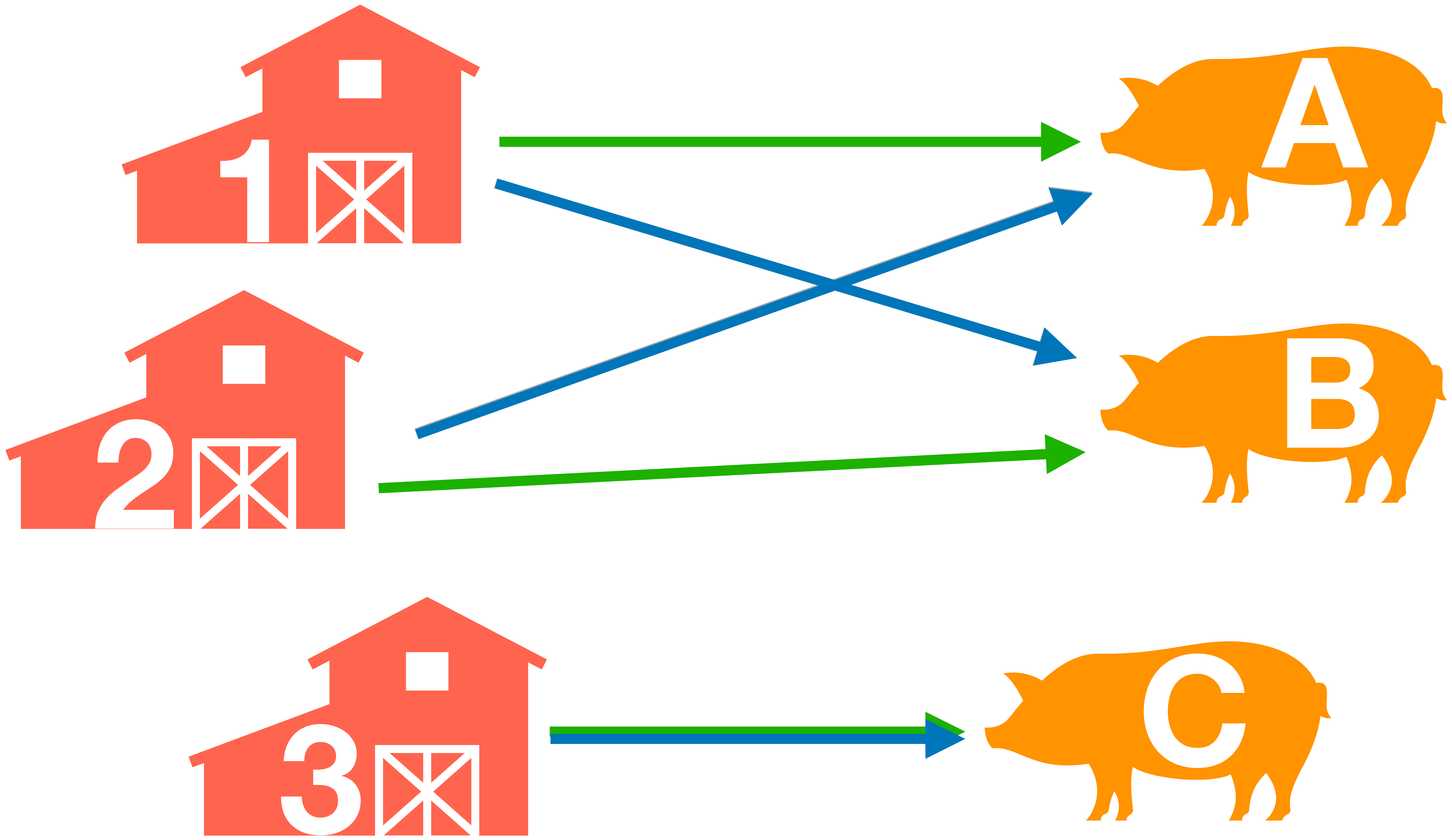
ideal <> stabil

Widerspruch


1
2
3
A
B
C

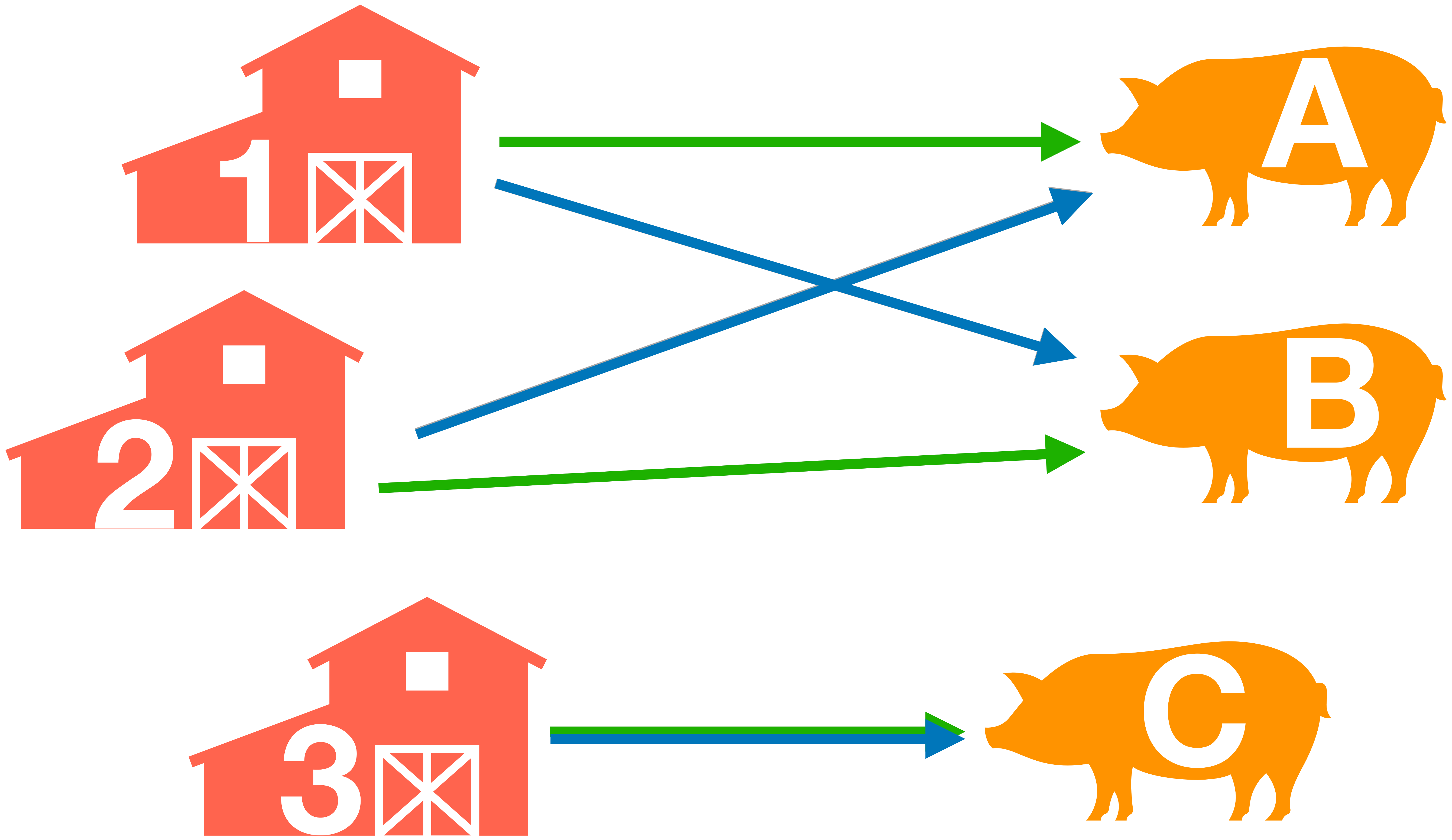


	1. Prio	2. Prio
1	A	B
2	B	A
3	C	–



	1. Prio	2. Prio
1	A	B
2	B	A
3	C	—

	A	B	C
1	0.7	0.4	—
2	0.2	0.9	—
3	—	—	0.8

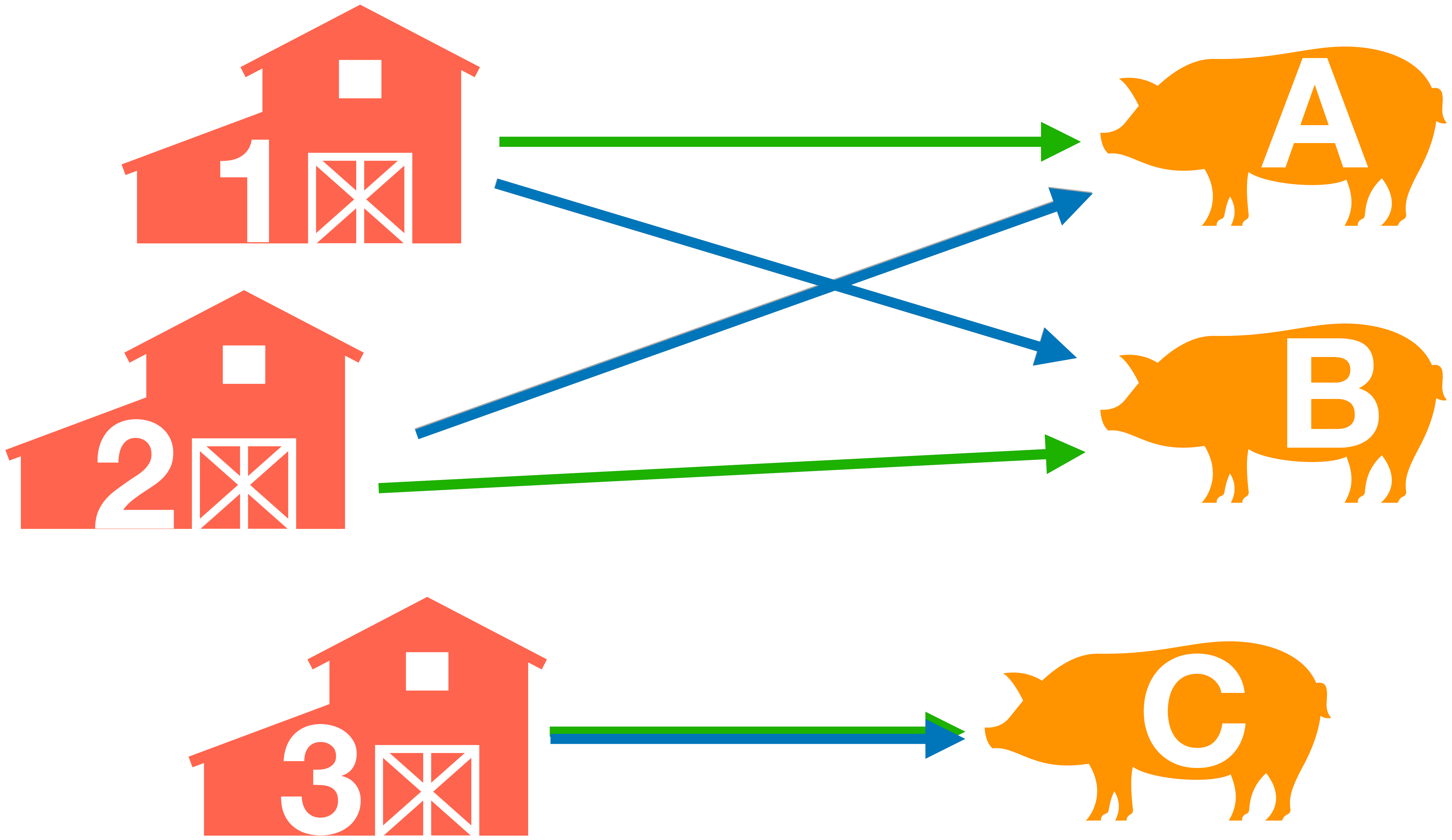


$$\sum_{i=1}^{|B|} w_i$$

$$= 0.4 + 0.2 + 0.8$$
$$= 1.4$$

	1. Prio	2. Prio
1	A	B
2	B	A
3	C	—

	A	B	C
1	0.7	0.4	—
2	0.2	0.9	—
3	—	—	0.8



$$\sum_{i=1}^{|B|} w_i = 0.7 + 0.9 + 0.8 = 2.4$$

$$\sum_{i=1}^{|B|} w_i = 0.4 + 0.2 + 0.8 = 1.4$$

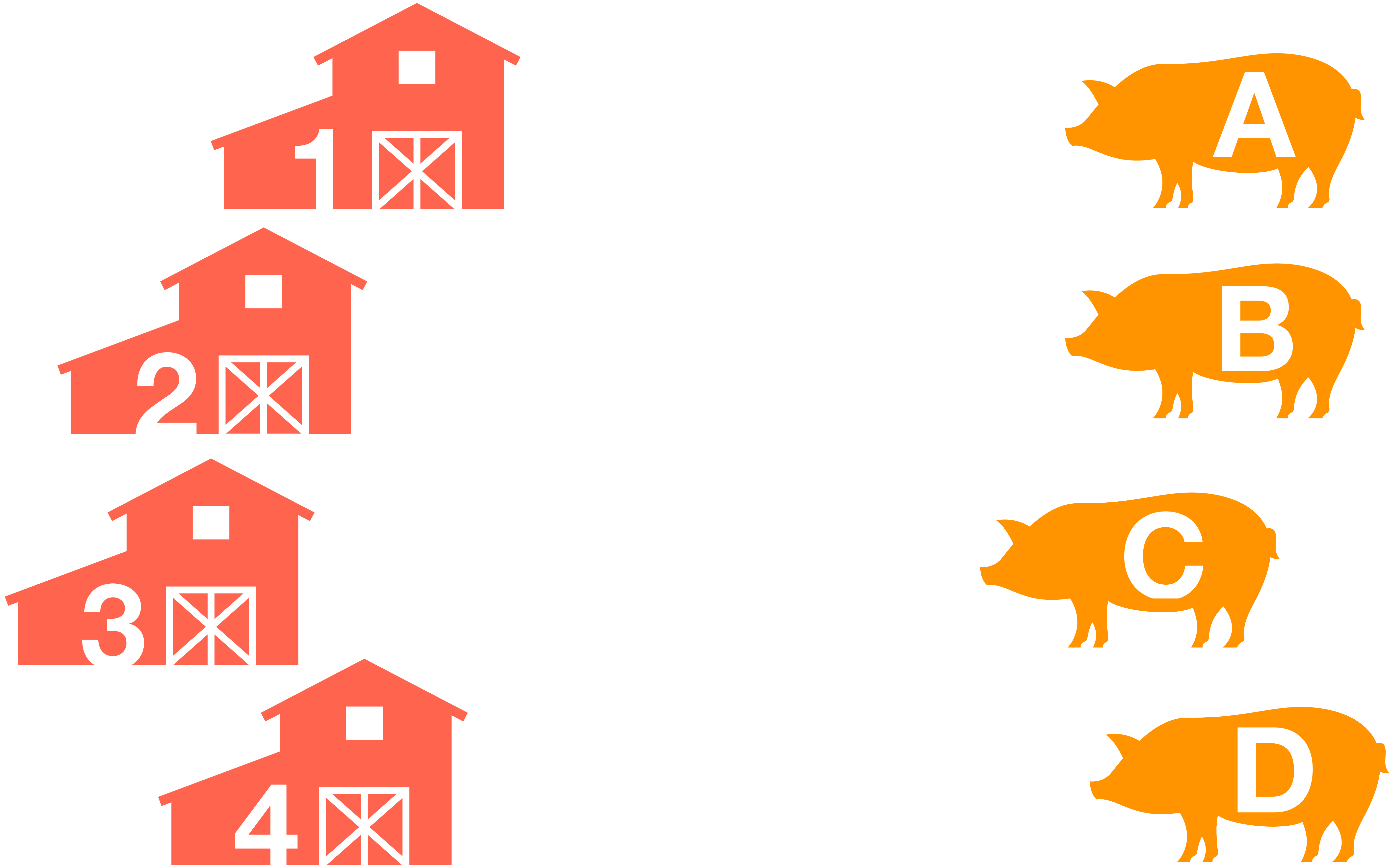
	1. Prio	2. Prio
1	A	B
2	B	A
3	C	—

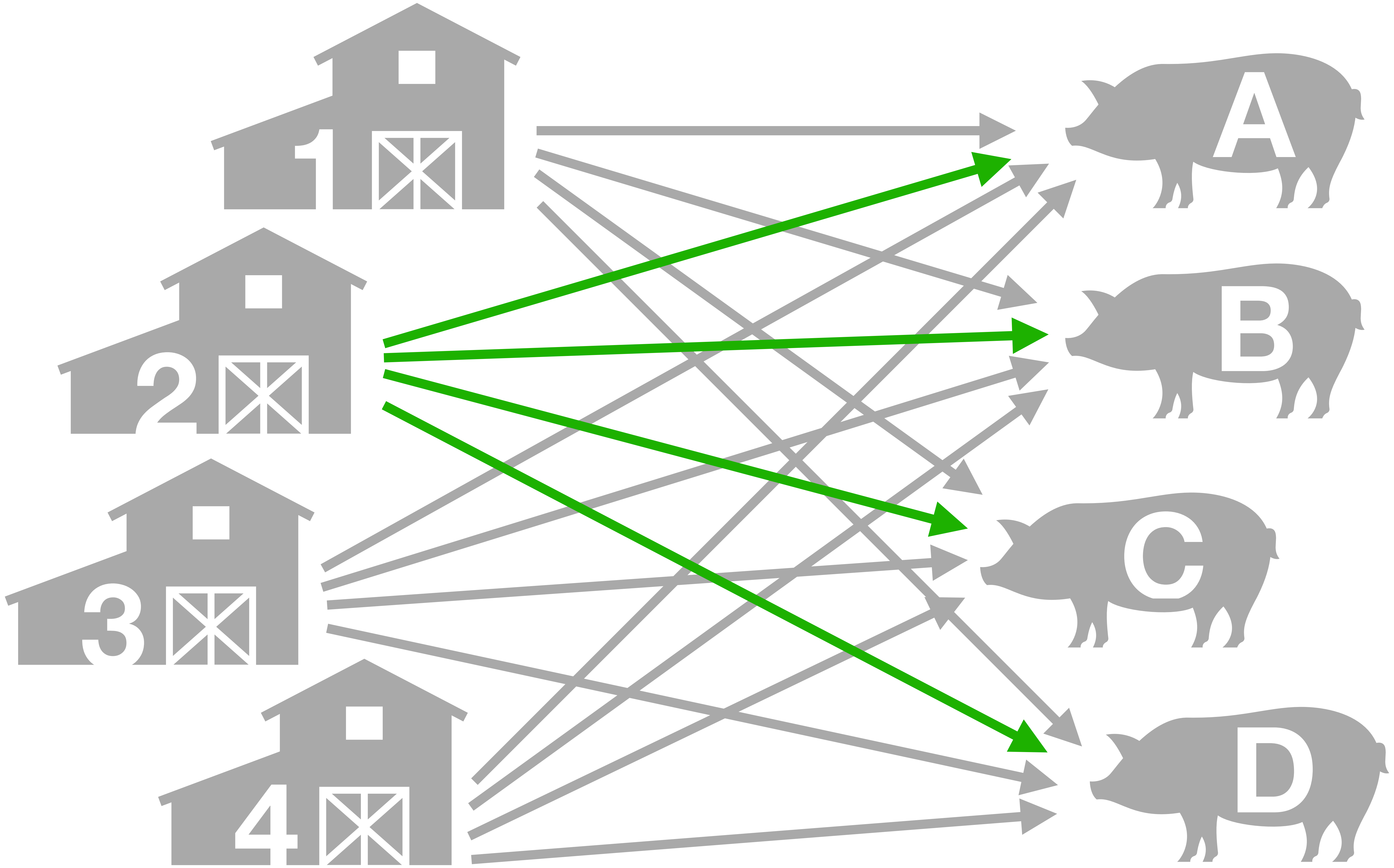
	A	B	C
1	0.7	0.4	—
2	0.2	0.9	—
3	—	—	0.8

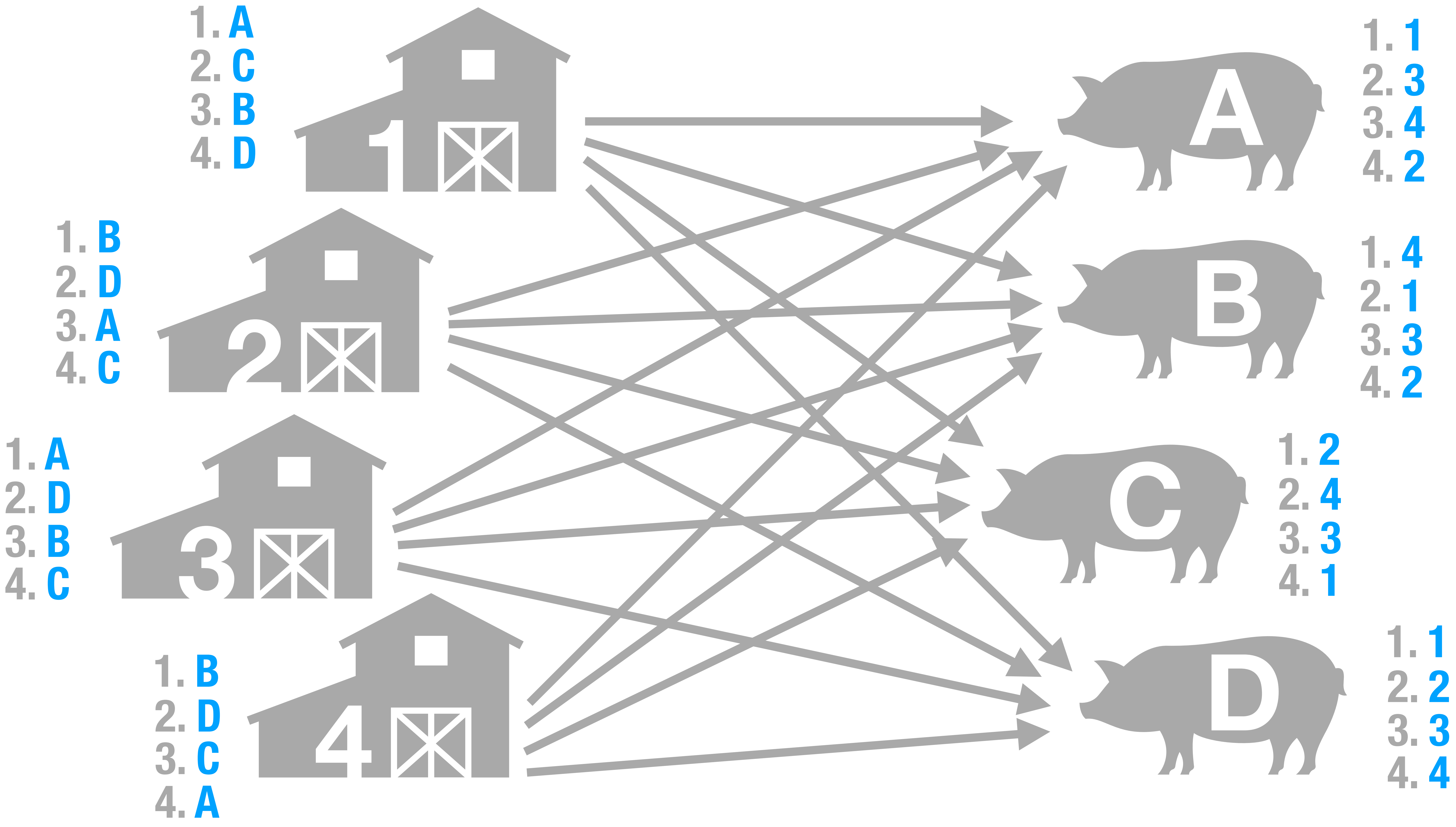
$$M_{gut}$$

Gale-Shapley

Algorithmus


1
2
3
4
A
B
C
D


1
2
3
4
A
B
C
D



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

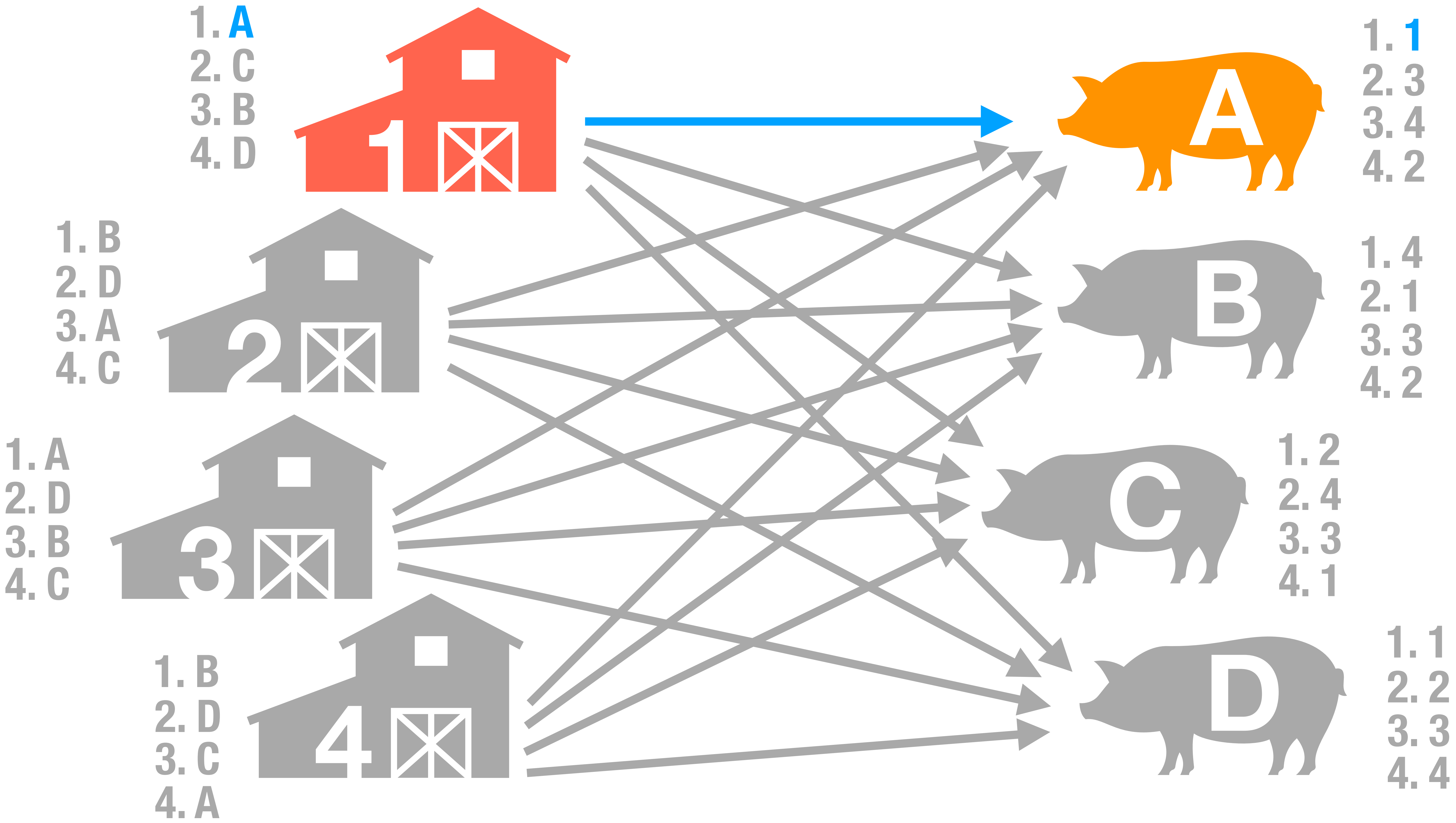
1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

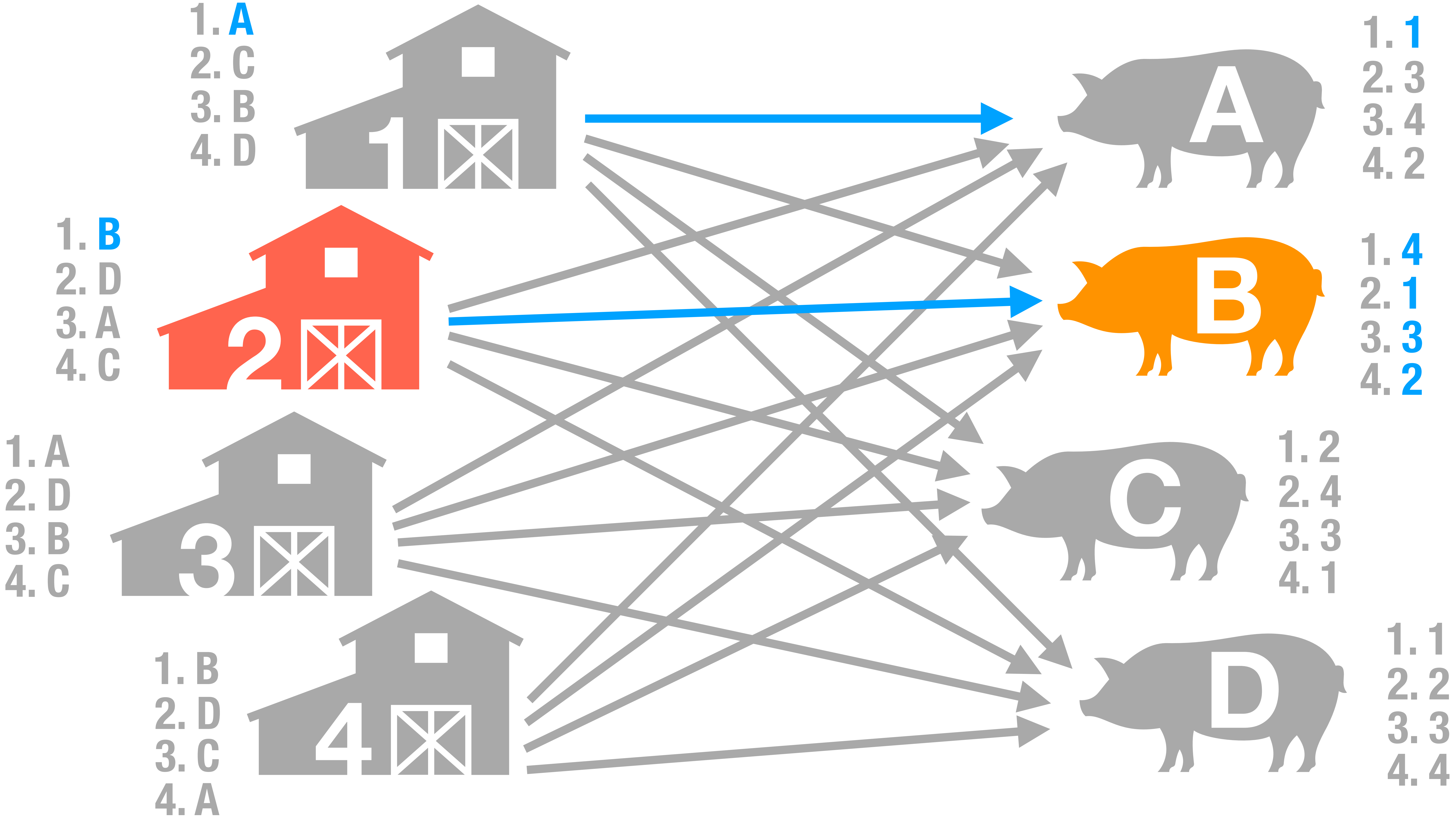
1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

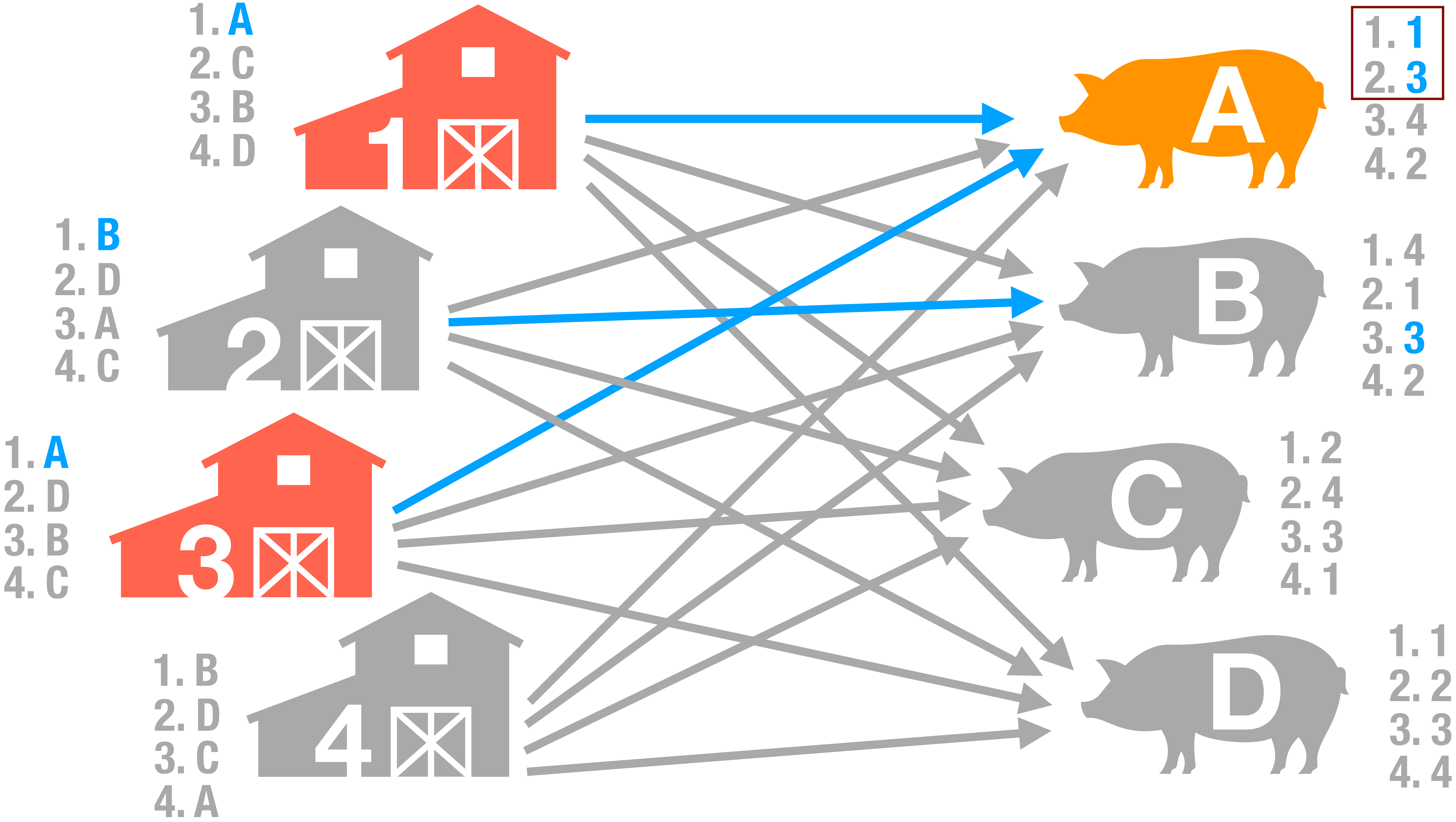
1. B
2. D
3. C
4. A

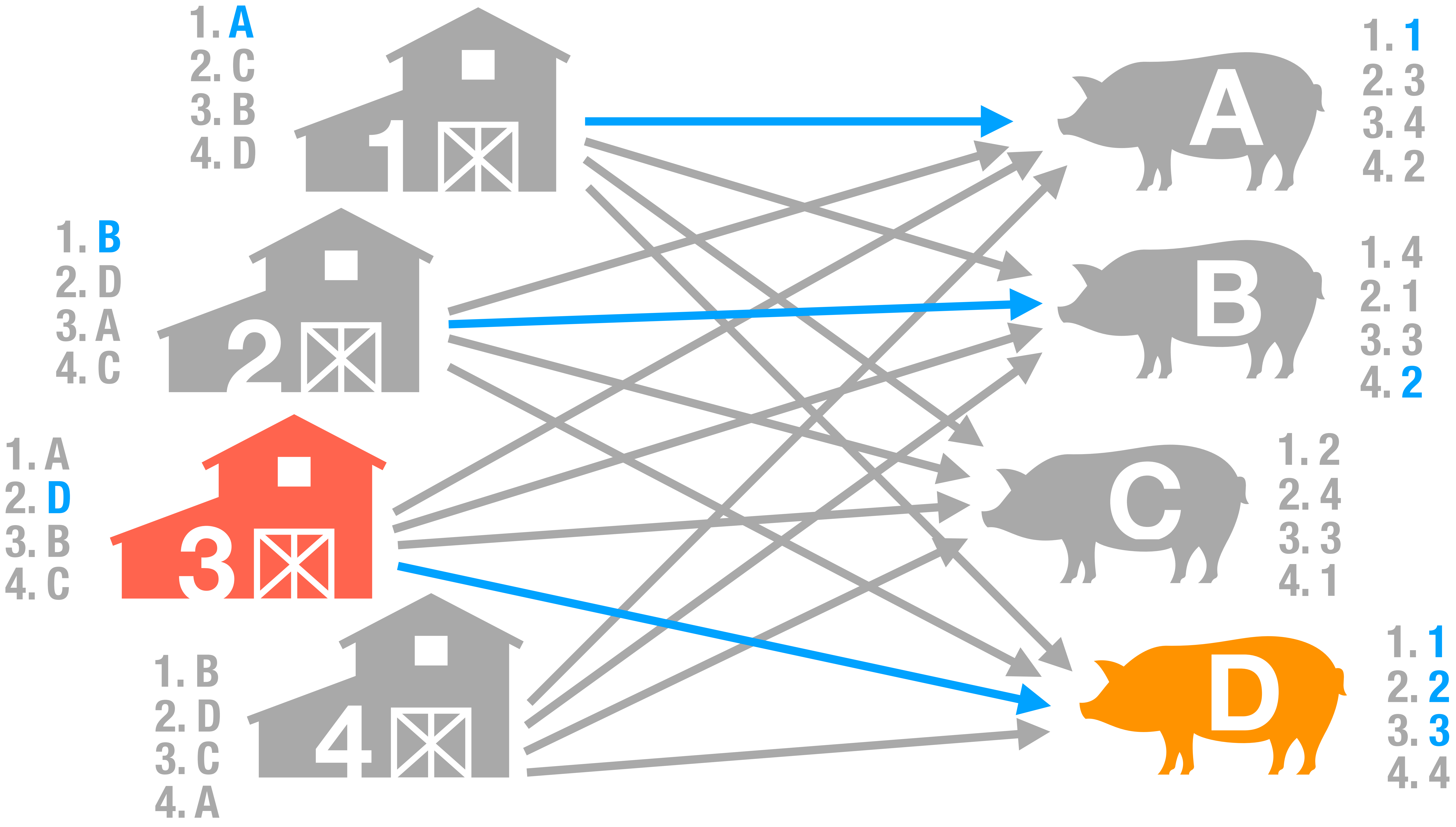
1. 1
2. 3
3. 4
4. 2

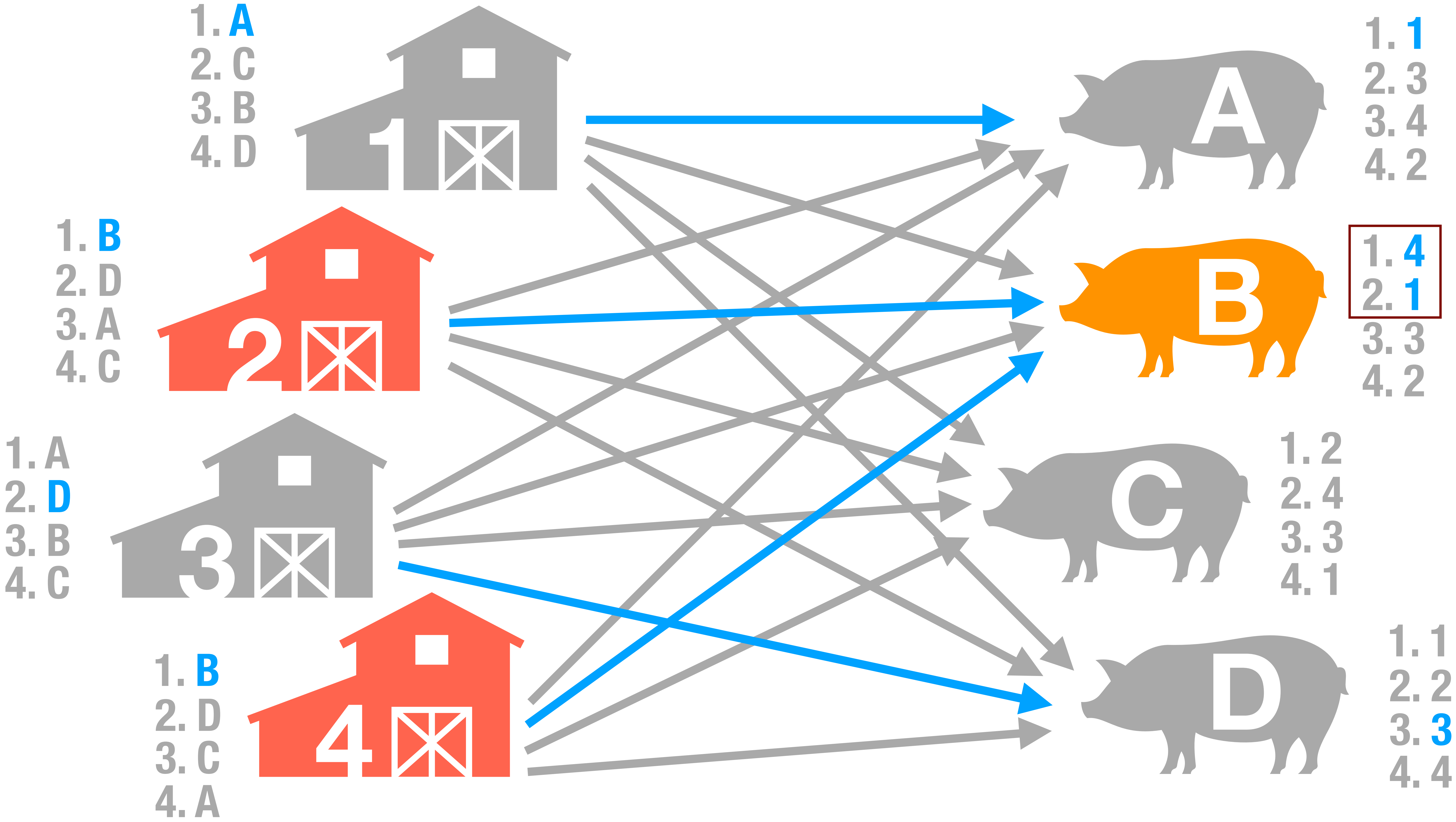
1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4

1. A
2. C
3. B
4. D
1
1. B
2. D
3. A
4. C
2
1. A
2. D
3. B
4. C
3
1. B
2. D
3. C
4. A
4
A
1. 1
2. 3
3. 4
4. 2
B
1. 4
2. 1
3. 3
4. 2
C
1. 2
2. 4
3. 3
4. 1
D
1. 1
2. 2
3. 3
4. 4


1. A
2. C
3. B
4. D
1
1. B
2. D
3. A
4. C
2
1. A
2. D
3. B
4. C
3
1. B
2. D
3. C
4. A
4
A
1. 1
2. 3
3. 4
4. 2
B
1. 4
2. 1
3. 3
4. 2
C
1. 2
2. 4
3. 3
4. 1
D
1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

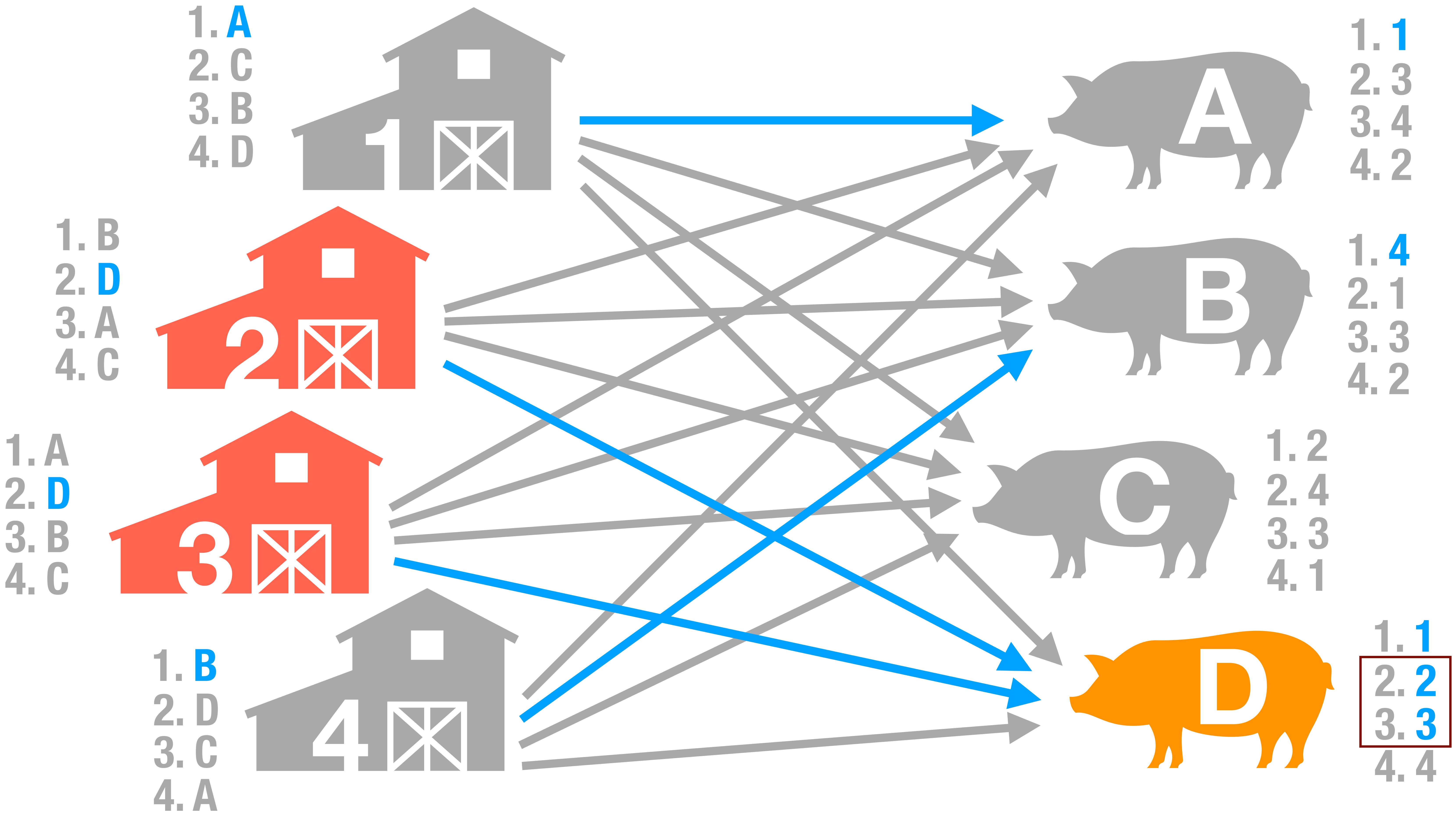
1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

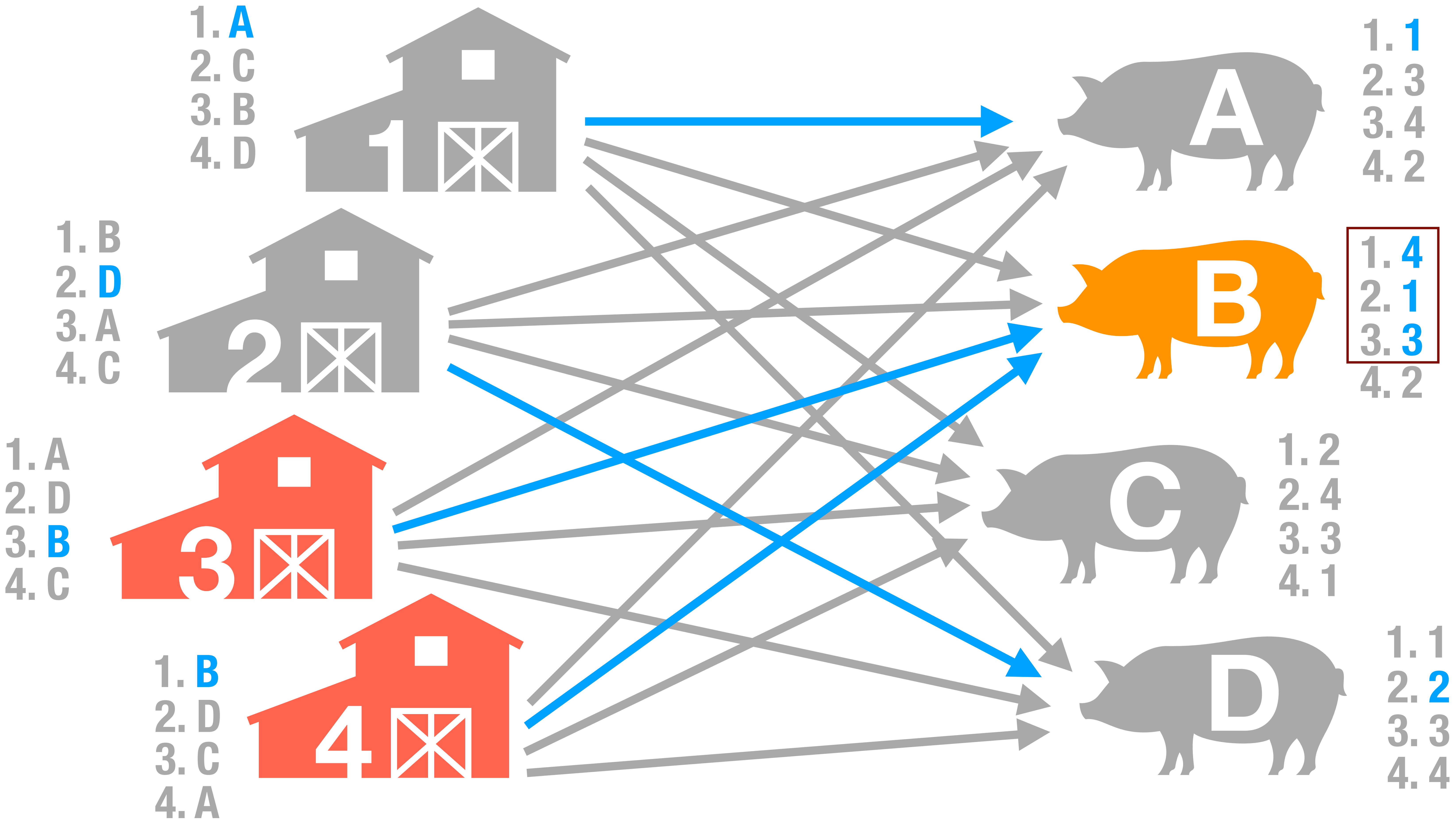
1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1

1. B
2. D
3. A
4. C

2

1. A
2. D
3. B
4. C

3

1. B
2. D
3. C
4. A

4

A

1. 1
2. 3
3. 4
4. 2

B

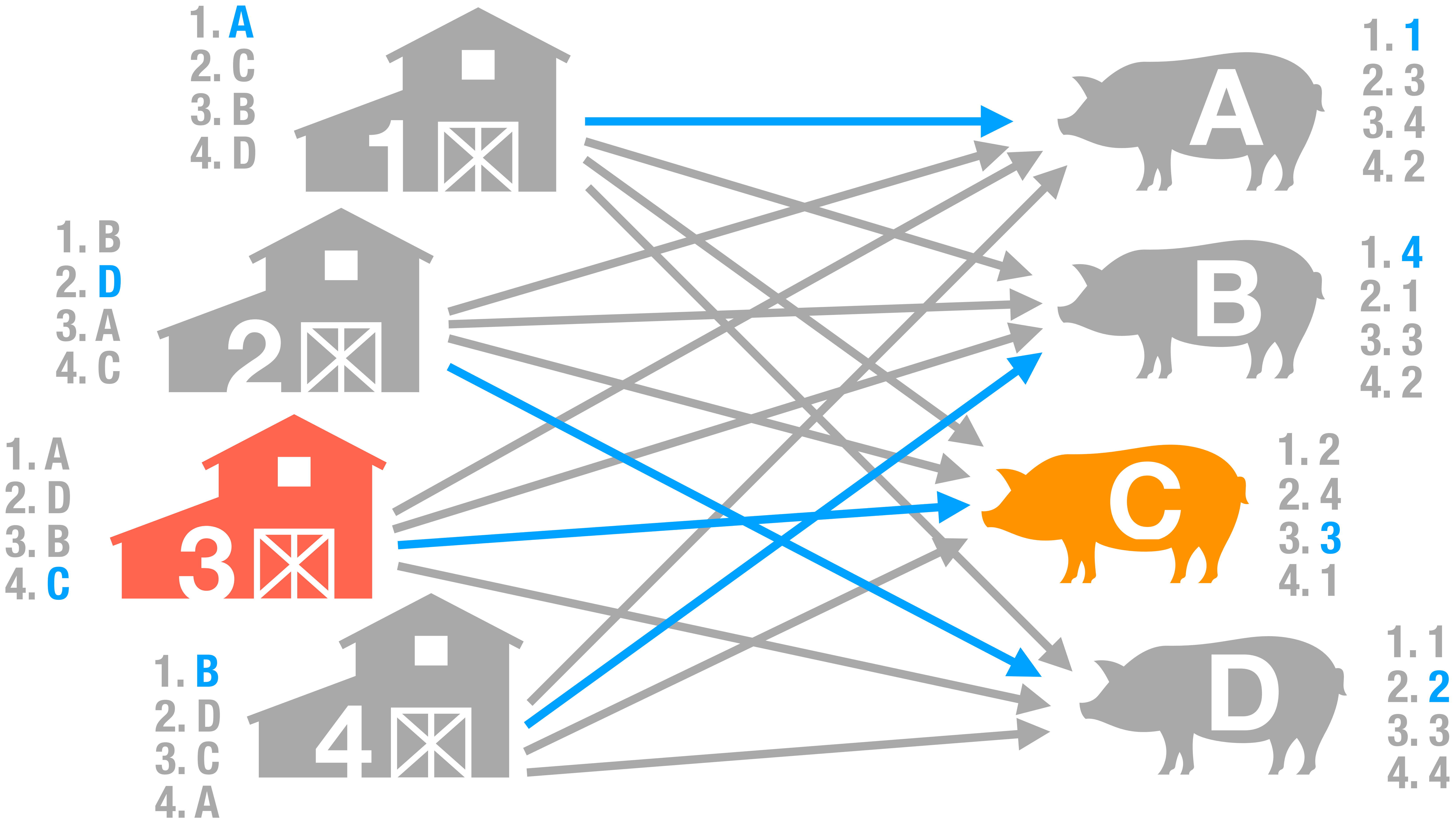
1. 4
2. 1
3. 3
4. 2

C

1. 2
2. 4
3. 3
4. 1

D

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

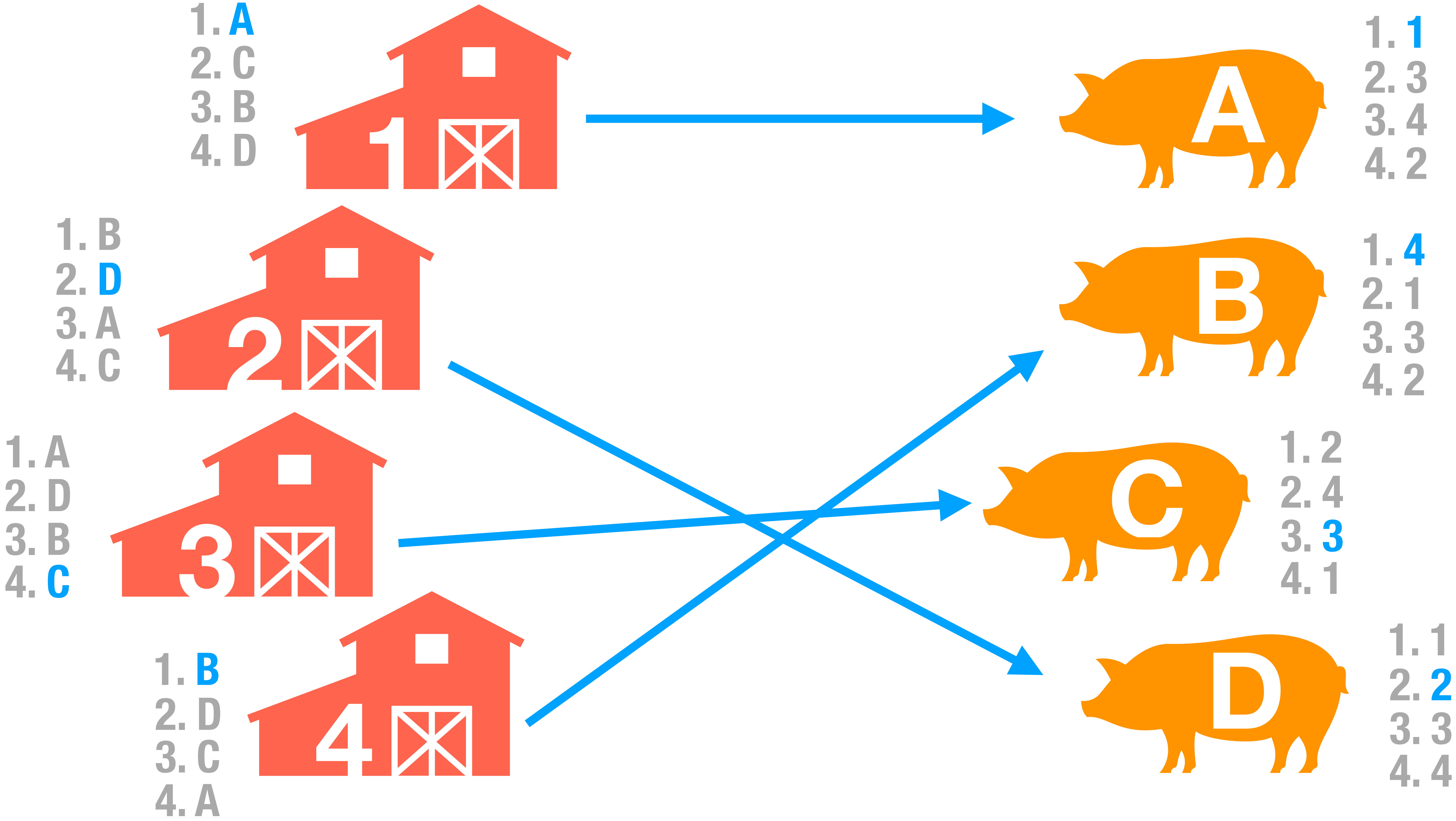
1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

1. 2
2. 4
3. 3
4. 1

1. 1
2. 2
3. 3
4. 4



1. A
2. C
3. B
4. D

1. B
2. D
3. A
4. C

1. A
2. D
3. B
4. C

1. B
2. D
3. C
4. A

1. 1
2. 3
3. 4
4. 2

1. 4
2. 1
3. 3
4. 2

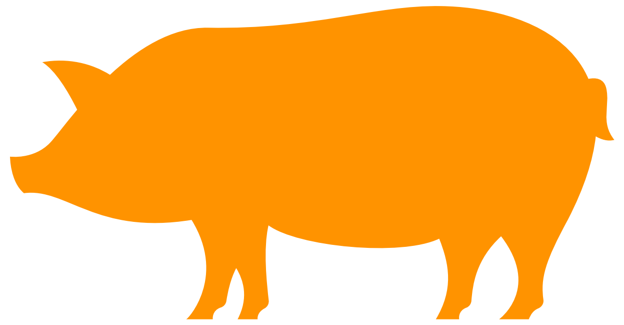
1. 2
2. 4
3. 3
4. 1

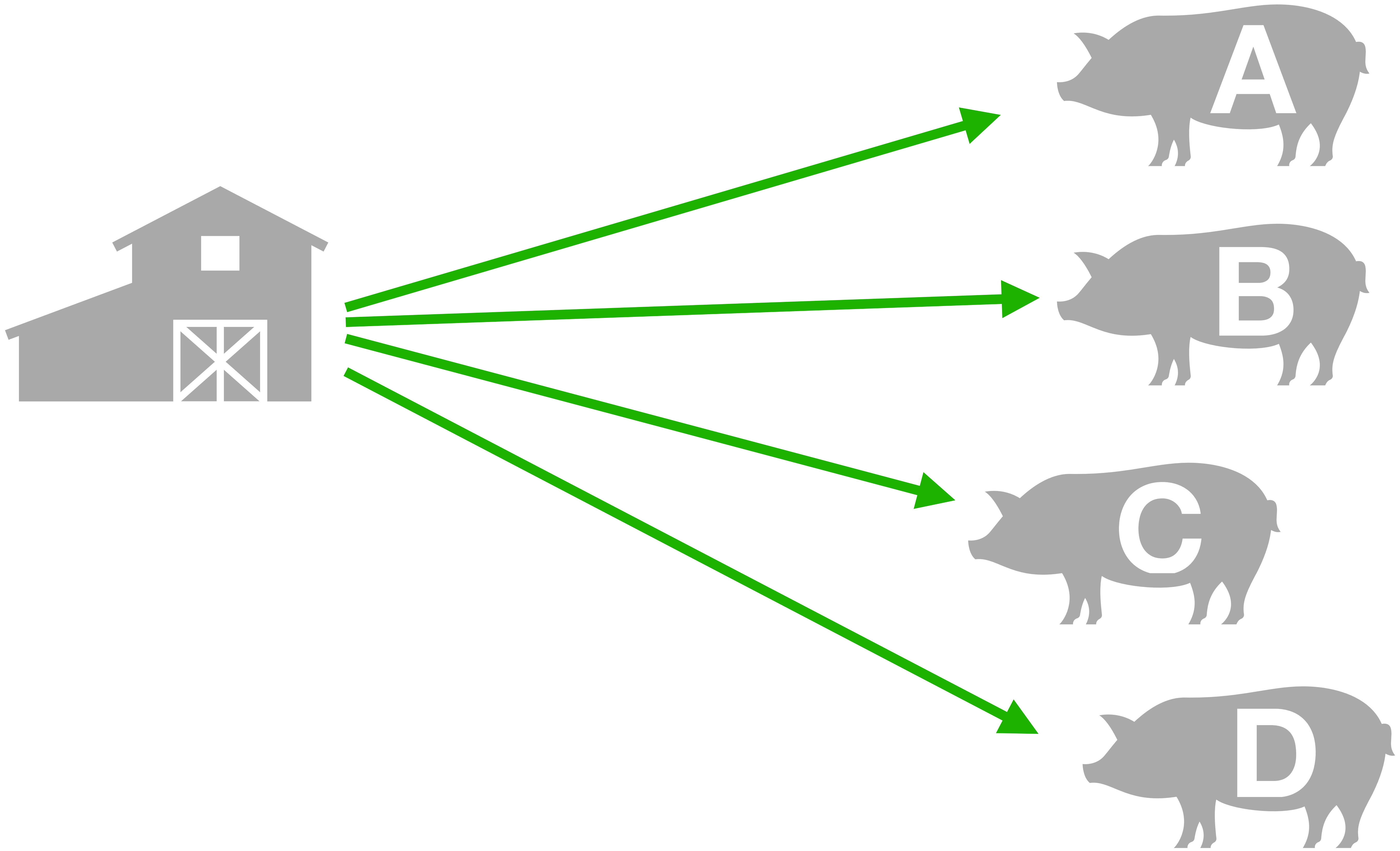
1. 1
2. 2
3. 3
4. 4

```
input farms, animals:

  while there are farms
    while farm is not matched
      take first farm from farms
      take most prioritised animal of farm

      if animal is not matched
        match farm and animal
        add pair to matching
        delete farm from farms

      else
        take pair which contains the animal
        if farm of pair is prioritised more highly by animal
          delete animal from prioritisation of current farm

        else
          delete pair from matching
          match current farm and animal
          add new pair
          delete farm from farms
          add farm from the pair to farms

  return matching
```

Gale-Shapley

Fazit



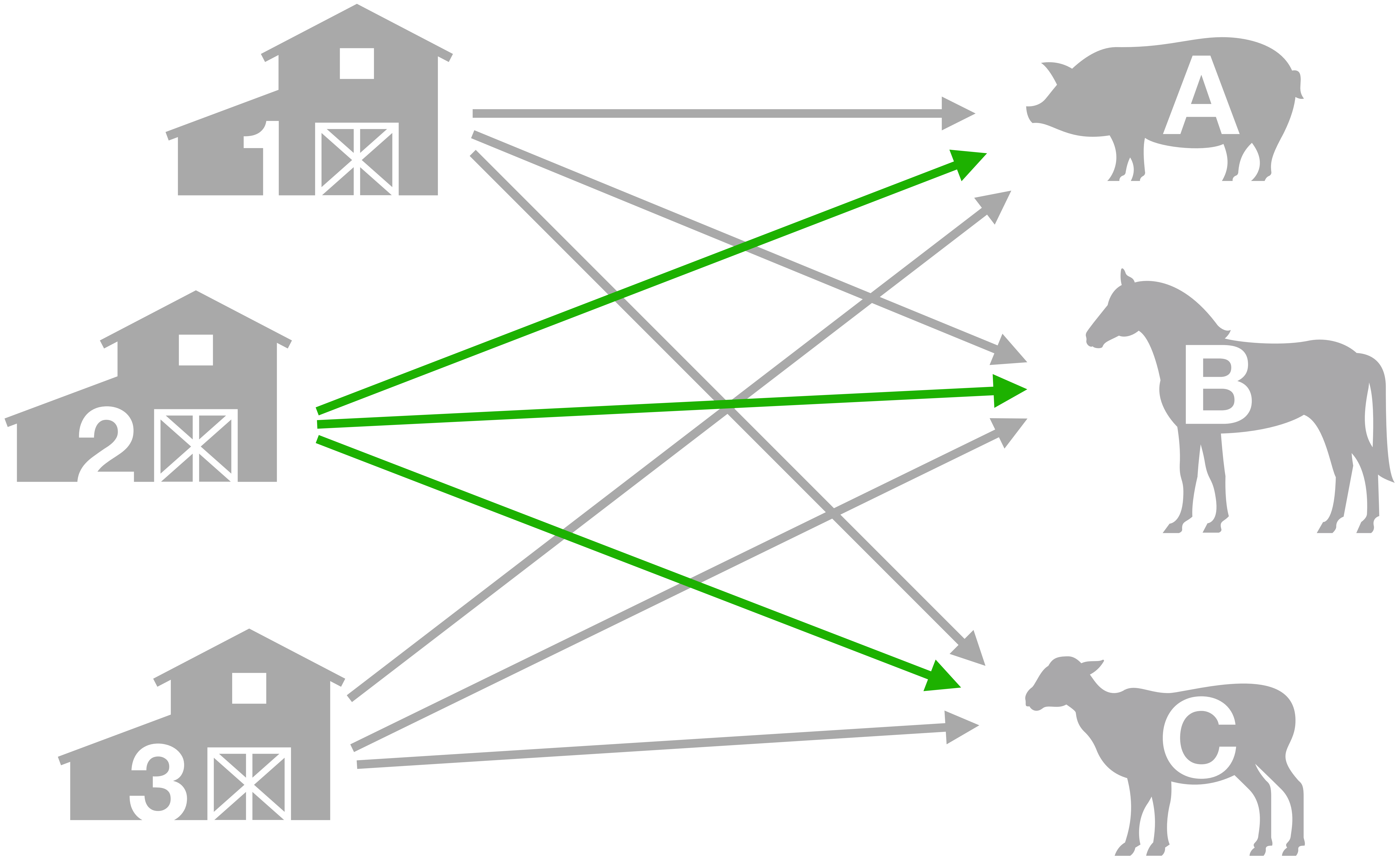


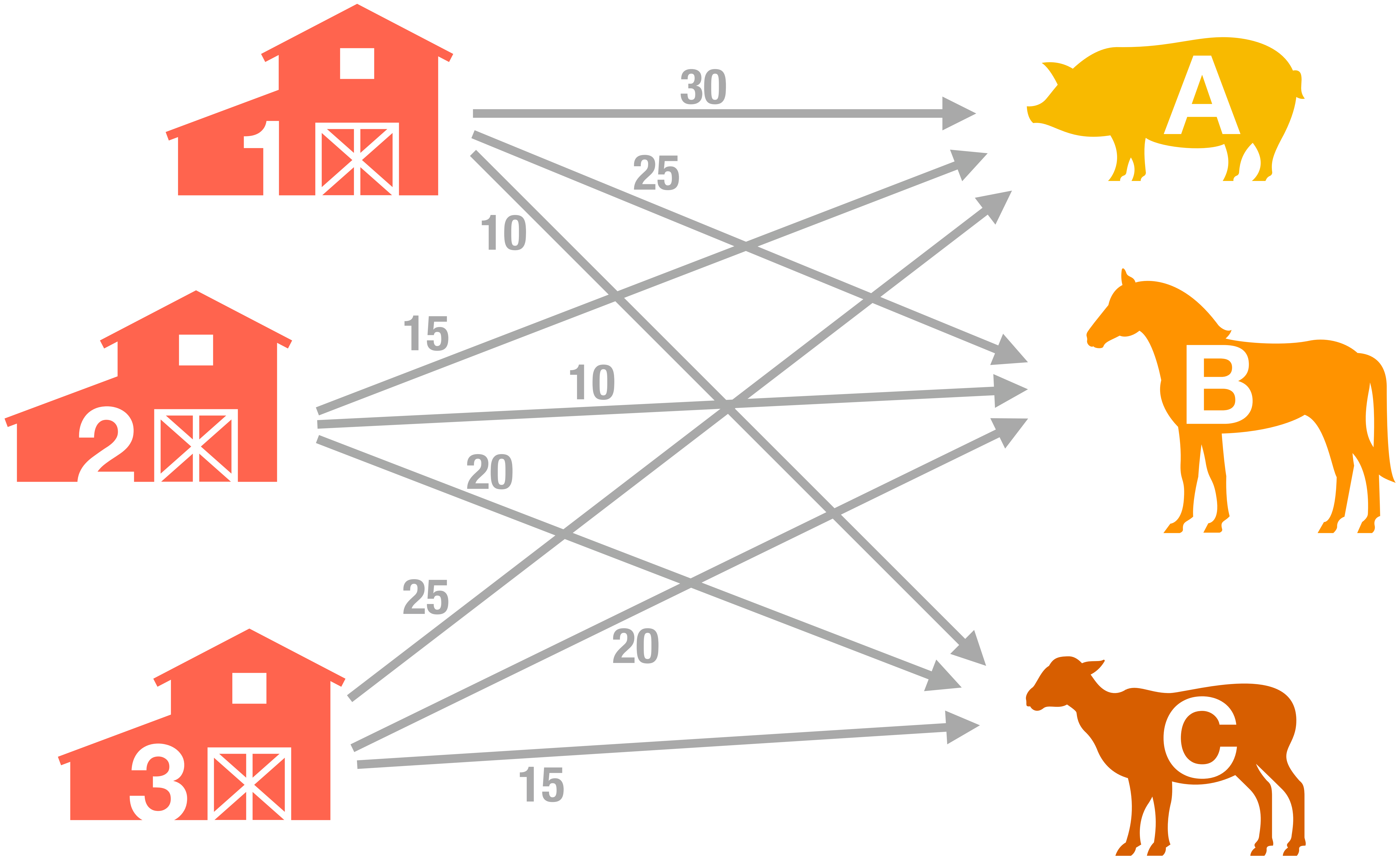
A
B
C
D

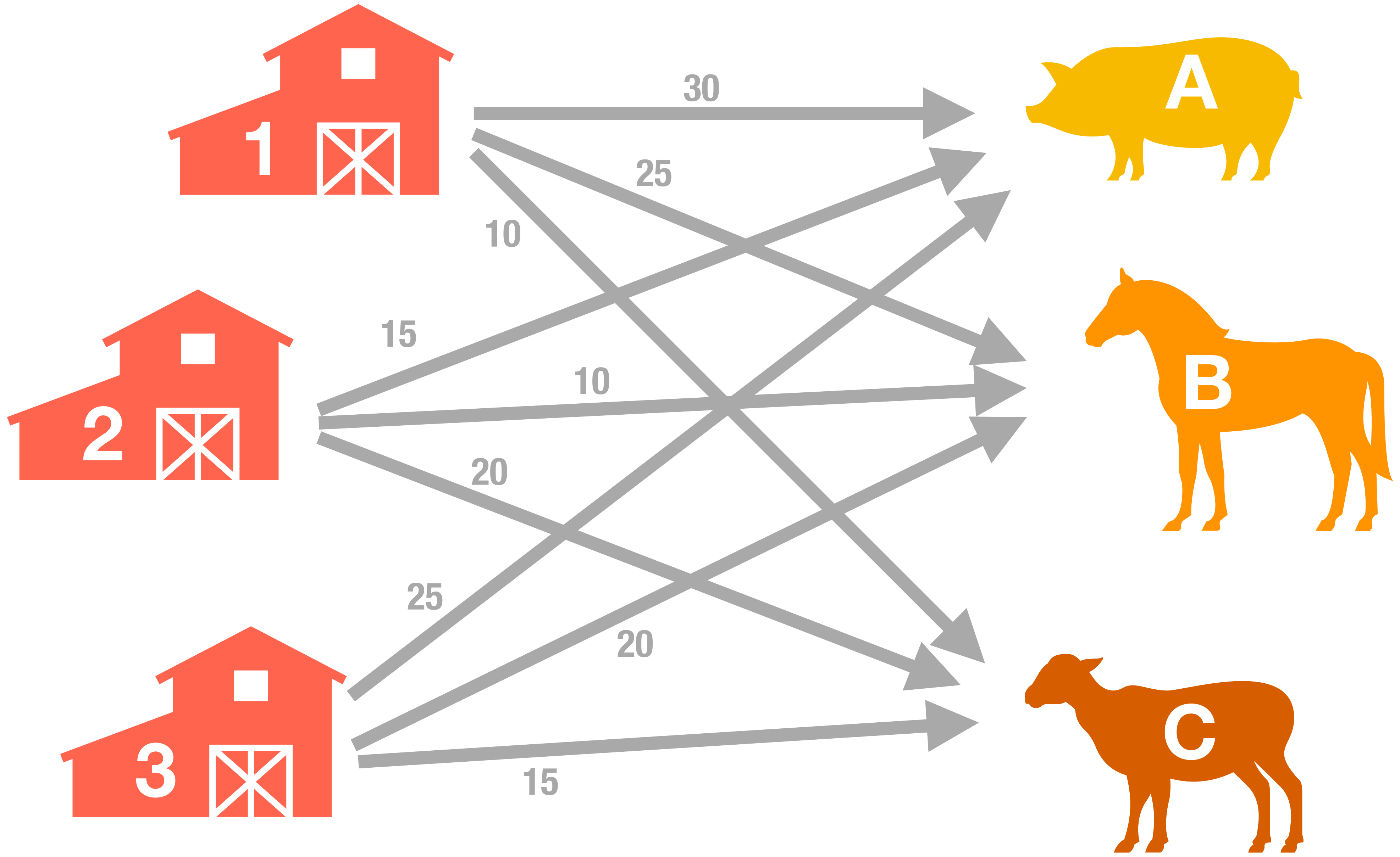
- □ min Stress
- ☑ max Priorisierung
- ☑ perfektes Matching

The Hungarian

Algorithmus


1
2
3
A
B
C


1
2
3
A
B
C
30
25
10
15
10
20
25
20
15



	A	B	C
1	30	25	10
2	15	10	20
3	25	20	15

	A 🐖	B 🐎	C 🐑
1 🏠	30	25	10
2 🏠	15	10	20
3 🏠	25	20	15

	A	B	C
1	30	25	10
2	15	10	20
3	25	20	15

	[illegible]	B	[illegible]
1	30	[illegible]	10
2	15	[illegible]	20
3	[illegible]	20	[illegible]

	A 🐖	B 🐎	C 🐑
1 🏠	30	25	10
2 🏠	15	10	20
3 🏠	25	20	15

	A	B	C
1	30	[illegible]	10
2	15	[illegible]	20
3	[illegible]	20	15

	A	B	C
1	**30**	25	10
2	15	**10**	20
3	25	20	**15**

55

	A	B	C
1	30	**25**	10
2	**15**	10	20
3	25	20	**15**

55

	A	B	C
1	30	25	**10**
2	15	**10**	20
3	**25**	20	15

45

	A	B	C
1	30	25	**10**
2	**15**	10	20
3	25	**20**	15

45

	A	B	C
1	30	**25**	10
2	15	10	**20**
3	**25**	20	15

70

	A	B	C
1	**30**	25	10
2	15	10	**20**
3	25	**20**	15

70

	A	B	C
1	**30**	25	10
2	15	**10**	20
3	25	20	**15**

55

	A	B	C
1	30	**25**	10
2	**15**	10	20
3	25	20	**15**

55

	A	B	C
1	30	25	**10**
2	15	**10**	20
3	**25**	20	15

45

	A	B	C
1	30	25	**10**
2	**15**	10	20
3	25	**20**	15

45

	A	B	C
1	30	**25**	10
2	15	10	**20**
3	**25**	20	15

70

	A	B	C
1	**30**	25	10
2	15	10	**20**
3	25	**20**	15

70

n

n

	A	B	C
1	30	25	10
2	15	10	20
3	25	20	15

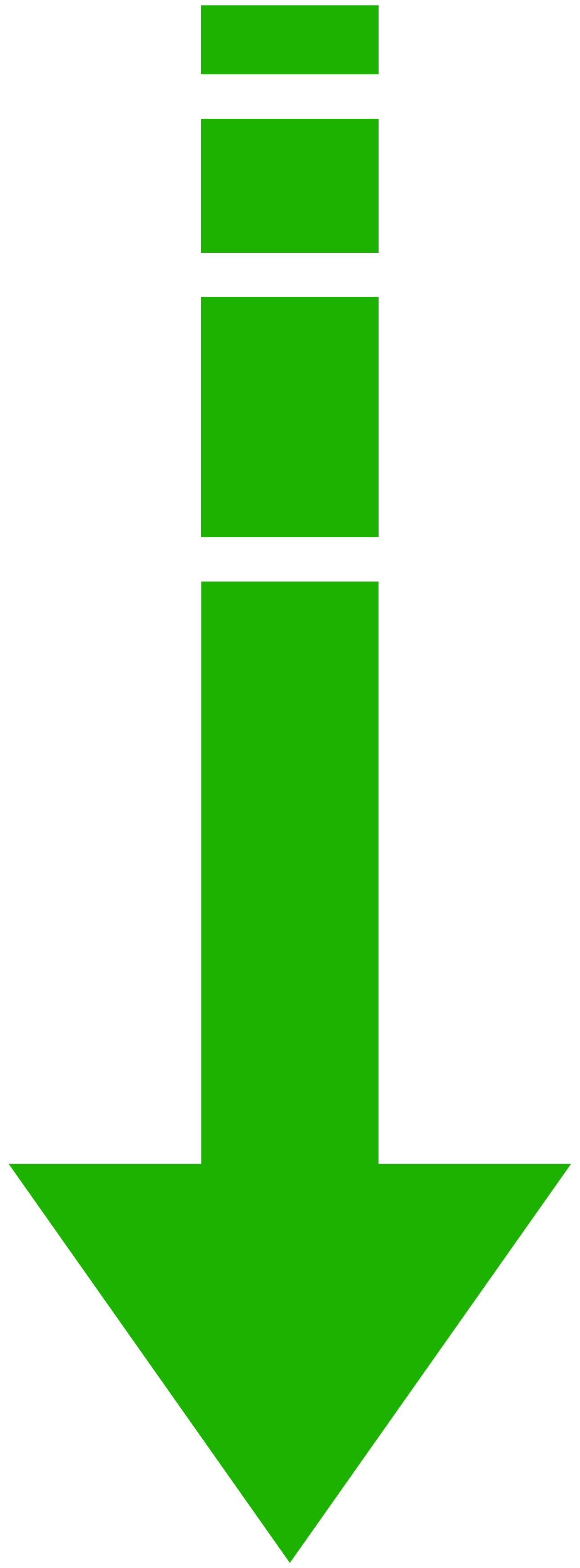
	A	B	C
1	30 - 10	25 - 10	10 - 10
2	15 - 10	10 - 10	20 - 10
3	25 - 15	20 - 15	15 - 15

	A	B	C
1	20	15	0
2	5	0	10
3	10	5	0

	A	B	C
1	20 - 5	15 - 0	0 - 0
2	5 - 5	0 - 0	10 - 0
3	10 - 5	5 - 0	0 - 0

	A	B	C
1	15	15	0
2	0	0	10
3	5	5	0

	A	B	C
1	15	15	0
2	0	0	10
3	5	5	0



$n \leq$

	A	B	C
1	15 - 5	15 - 5	0
2	0	0	10 + 5
3	5 - 5	5 - 5	0

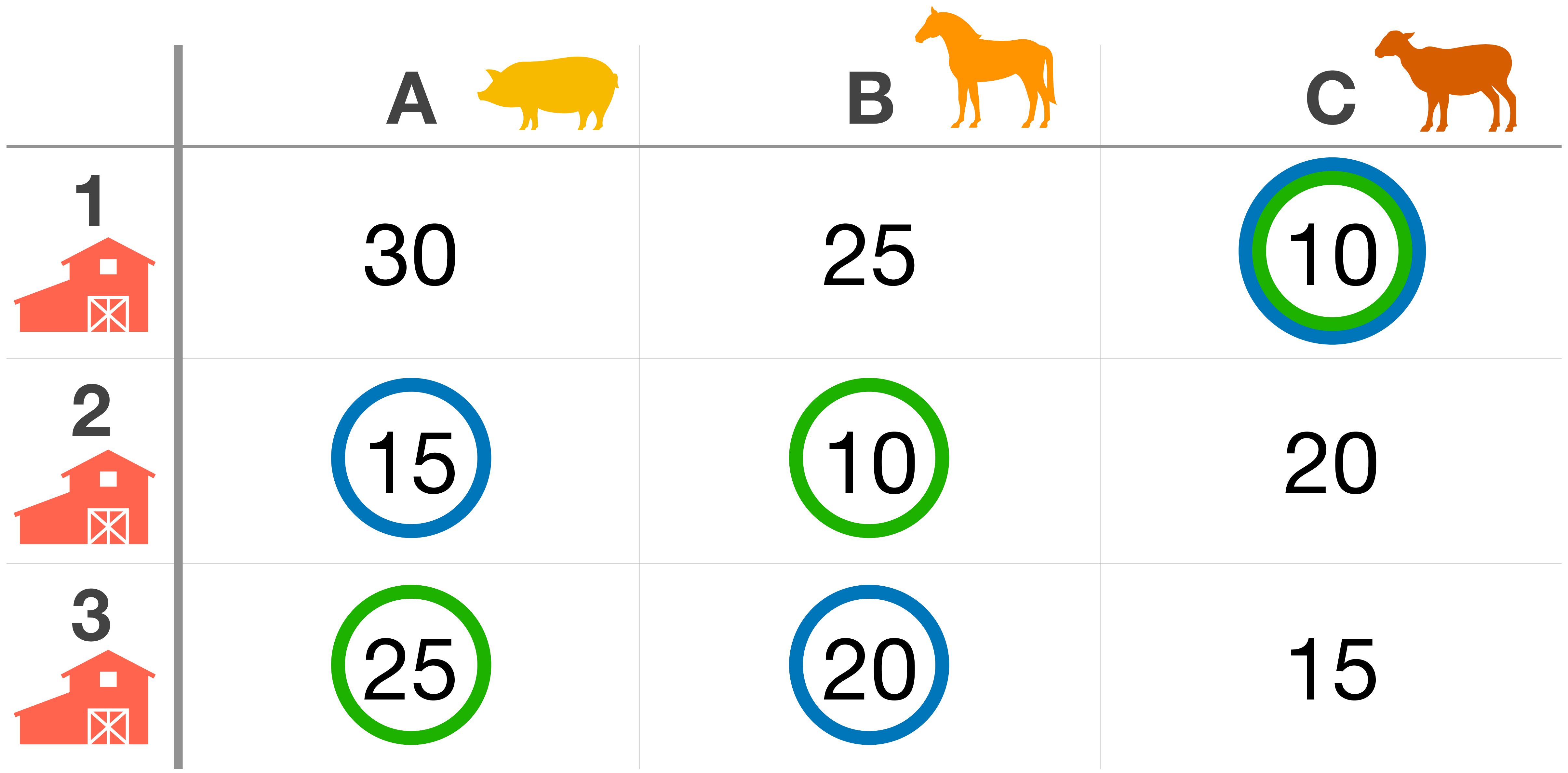
	A	B	C
1	10	10	0
2	0	0	15
3	0	0	0

	A	B	C
1	10	10	0
2	0	0	15
3	0	0	0

$$3 = n$$

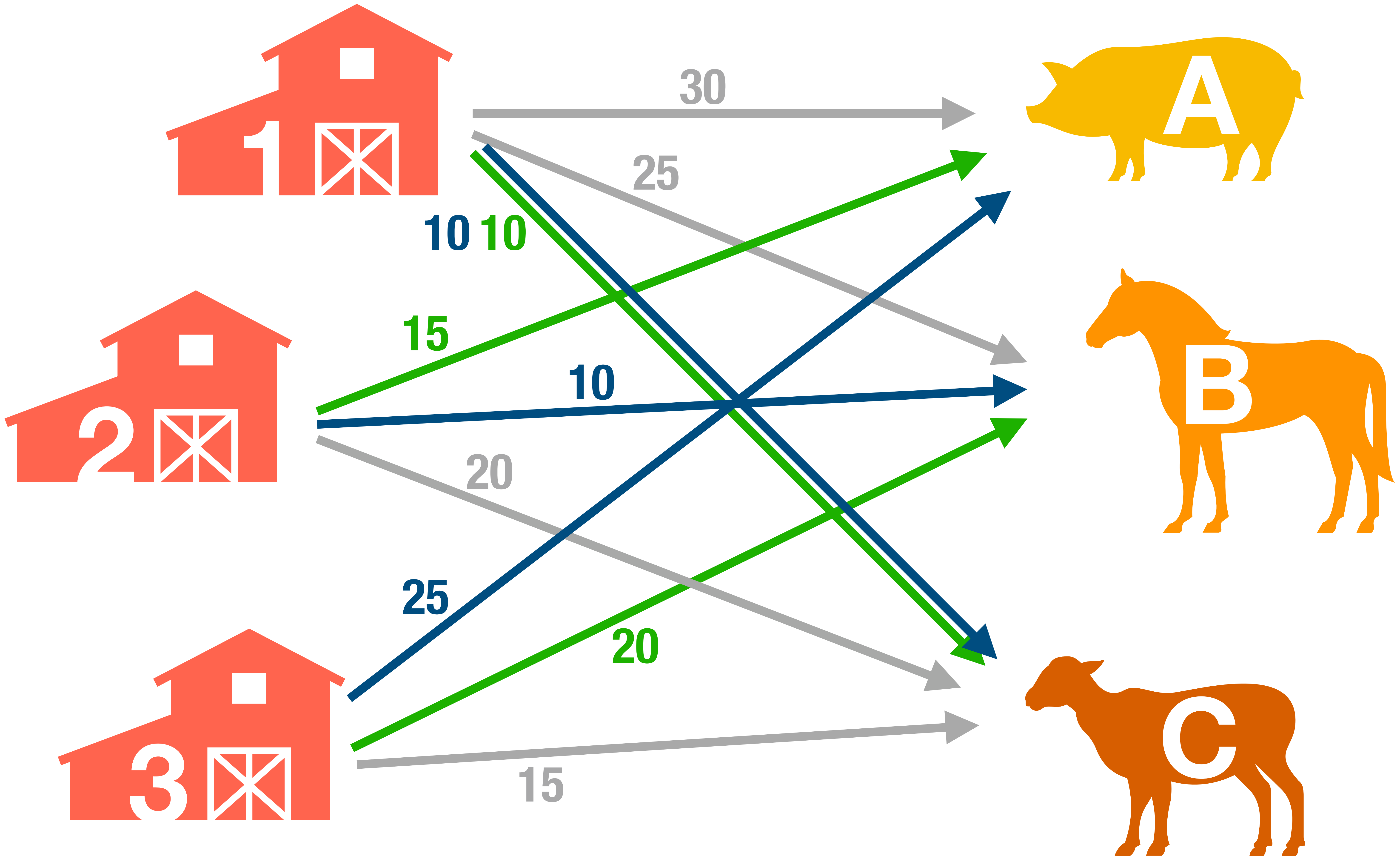
	A	B	C
1	10	10	0
2	0	0	15
3	0	0	0

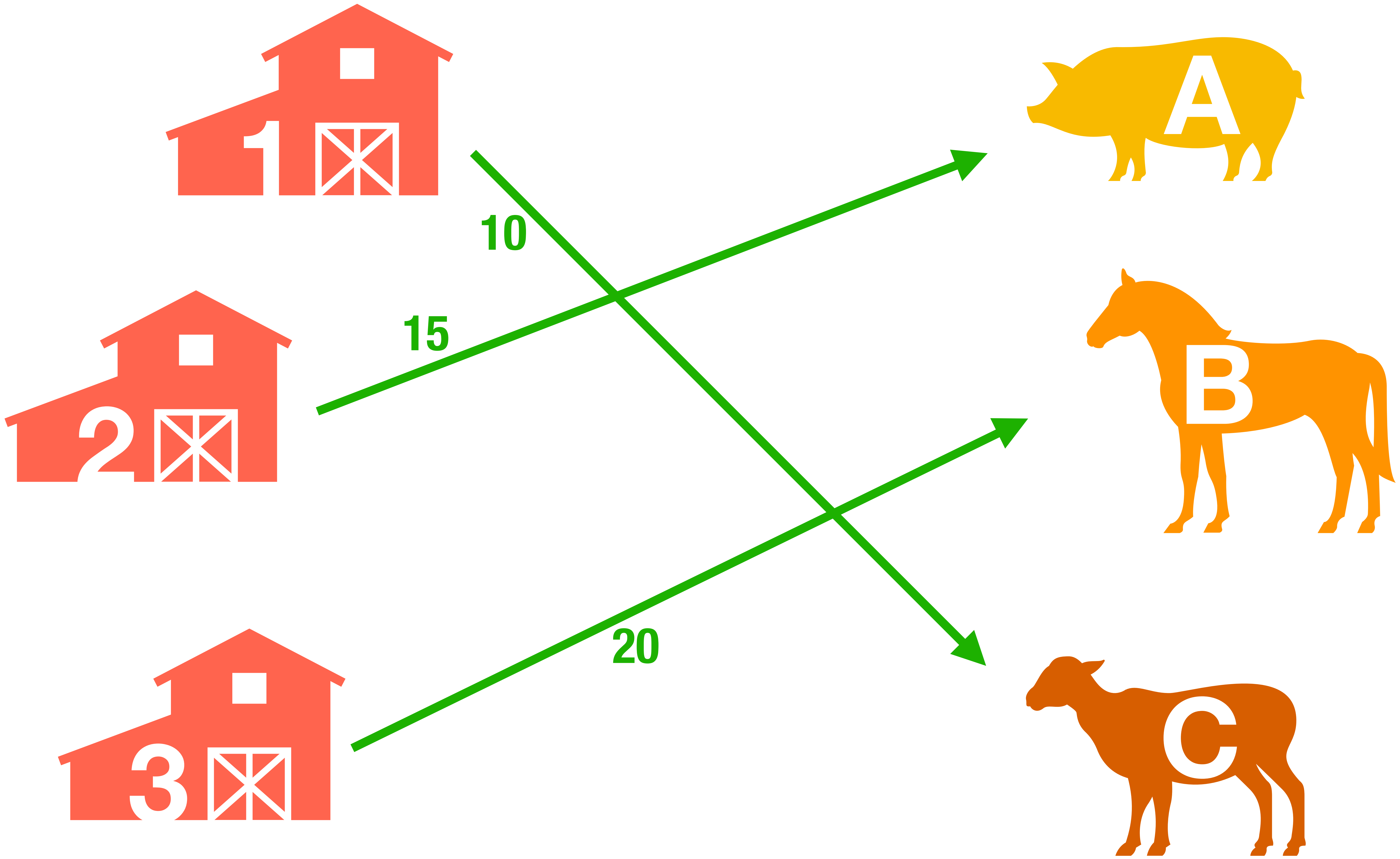
$$3 = n$$



	A	B	C
1	30	25	10
2	15	10	20
3	25	20	15

$$= 45$$


1
2
3
A
B
C
30
25
10
10
15
10
20
25
20
15


1
2
3
10
15
20
A
B
C

☑ min Stress

□ max Priorisierung

☑ perfektes Matching

Transfer

Problem

- ☐ min Stress
- ☑ max Priorisierung
- ☑ perfektes Matching

- ☑ min Stress
- □ max Priorisierung
- ☑ perfektes Matching

☑ min Stress

☑ max Priorisierung

☑ perfektes Matching

Aber wie lassen sich die Verfahren jetzt kombinieren ?

...to be continued

Graph Theory II (Tom Leighton and Ronitt Rubinfeld - Mathematics for Computer Science 2006)

Matching Theory (L. Lovász and M. D. Plummer - North Holland 1986) [Basic Terminology & Matchings in Bipartite Graphs]

The Hungarian method for the assignment problem (H. W. Kuhn - Bryn Nair College)

College Admissions and the Stability of Marriage (D. Gale and L. S. Shapley - The American Mathematical Monthly 1962) [Vol. 69, No. 1, pp. 9-15]

Gale-Shapley algorithm simply explained (Alexander Osipenko 2019) [https://towardsdatascience.com/gale-shapley-algorithm-simply-explained-caa344e643c2 - letzter Zugriff am 19.12.2019 - 07:57 Uhr]

Literatur