Updated 3/17/2022

Topic 2: Complexity of Query Evaluation
Unit 3: Provenance (continued)
Lecture 15

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp22)

https://northeastern-datalab.github.io/cs7240/sp22/

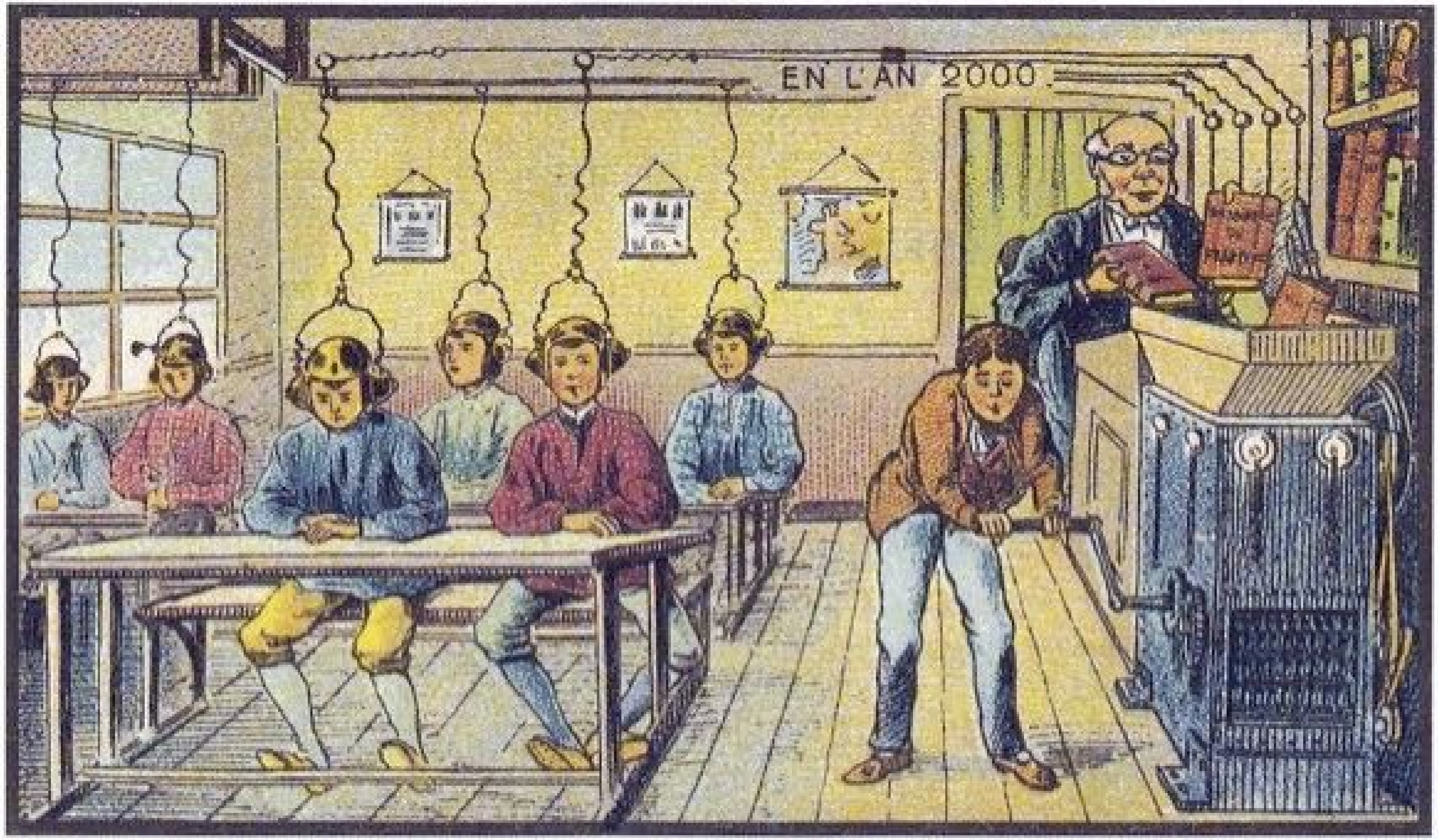
3/8/2022

Pre-class conversations

- Recapitulation of provenance semirings, including new exercise
- Projects & scribes: we are past halftime of the class
- Possible exercise: Provenance for relational division

- Today:
  - The algebra of provenance
  - a quick glimpse at reverse data management

The year 2000 imagined in 1900



At School

Source: https://publicdomainreview.org/collection/a-19th-century-vision-of-the-year-2000

The "Surfer Analogy" for time management

Source: http://stwww.surfermag.com/files/2013/10/Yak_Charlie-970x646.jpg

Topic 2: Complexity of Query Evaluation & Reverse Data Management

- **CONTINUED Lecture 11 (Tue 2/22):** T2-U1 Conjunctive Queries
- **Lecture 12 (Fri 2/25):** T2-U1 Conjunctive Queries
- **Lecture 13 (Tue 3/1):** T2-U1 Conjunctive Queries / T2-U2 Beyond Conjunctive Queries
- **Lecture 14 (Fri 3/4):** T2-U3 Provenance
- **Lecture 15 (Tue 3/8):** Provenance, Reverse Data Management

Pointers to relevant concepts & supplementary material:

- **Unit 1. Conjunctive Queries**: Query evaluation of conjunctive queries (CQs), data vs. query complexity, homomorphisms, constraint satisfaction, query containment, query minimization, absorption: [Kolaitis, Vardi'00], [Vardi'00], [Kolaitis'16], [Koutris'19] L1 & L2
- **Unit 2. Beyond Conjunctive Queries**: unions of conjunctive queries, bag semantics, nested queries, tree pattern queries: [Kolaitis'16], [Tan+'14], [G.'11], [Martens'17]
- **Unit 3. Provenance**: [Buneman+02], [Green+07], [Cheney+09], [Green,Tannen'17], [Kepner+16], [Buneman, Tan'18]
- **Unit 4. Reverse Data Management**: update propagation, resilience: [Buneman+02], [Kimelfeld+12], [Freire+15]

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-A,B,C}}(R \bowtie S)$

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-A,B,C}}(\underbrace{R \bowtie S})$

$R \bowtie S$

?

$\pi_{\text{-A,B,C}}(\ldots)$

?

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

Query plan 1: $\pi_{\text{-A,B,C}}(\underbrace{R \bowtie S})$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-A,B,C}}(\ldots)$

?

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-}A,B,C}(\underbrace{R \bowtie S})$

Query plan 2: $\pi_{\text{-}B}(\pi_{\text{-}A}(R) \bowtie \pi_{\text{-}C}(S))$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-}A,B,C}(\ldots)$

$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

Query plan 1: $\pi_{\text{-A,B,C}}(\underbrace{R \bowtie S})$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-A,B,C}}(\ldots)$

$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$

Query plan 2: $\pi_{\text{-B}}(\underbrace{\pi_{\text{-A}}(R)} \bowtie \underbrace{\pi_{\text{-C}}(S)})$

$\pi_{\text{-A}}(R)$

?

$\pi_{\text{-C}}(S)$

?

$\pi_{\text{-B}}(R' \bowtie S')$

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-}A,B,C}(R \bowtie S)$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-}A,B,C}(\ldots)$

$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$

Query plan 2: $\pi_{\text{-}B}(\pi_{\text{-}A}(R) \bowtie \pi_{\text{-}C}(S))$

$\pi_{\text{-}A}(R)$

B	
1	r_1
2	$r_2 + r_3$

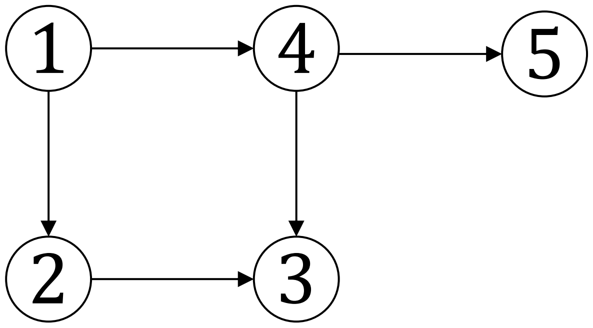
$\pi_{\text{-}C}(S)$

B	
1	$s_1 + s_2$
2	s_3

$\pi_{\text{-}B}(R' \bowtie S')$

$r_1 \cdot (s_1 + s_2) + (r_2 + r_3) \cdot s_3$

Back to our Example: now with Semiring notation



Now assume we use semiring notation.

Idea: keep the tuple identifiers abstract.

Use provenance polynomials ($\mathbb{N}[X], +, \cdot, 0, 1$)

E

1	2
2	3
1	4
4	3
4	5

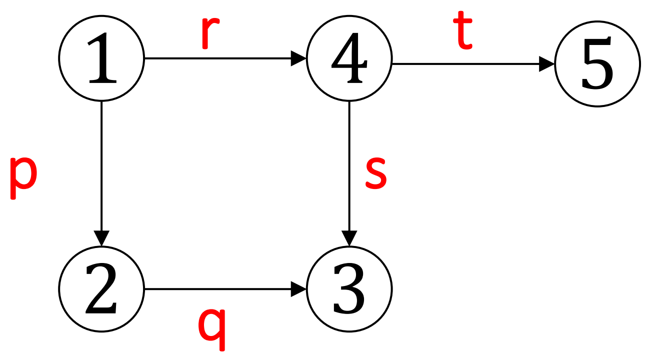
Q(z) :- E(1,y), E(y,z) → Q

Q

3
5

Q: Points reachable in 2 hops, starting at node "1"

Back to our Example: now with Semiring notation



Now assume we use semiring notation.
Idea: keep the tuple identifiers abstract.
Use provenance polynomials ($\mathbb{N}[X]$, +, ·, 0, 1)

$\oplus$ $\otimes$

E

1	2	p
2	3	q
1	4	r
4	3	s
4	5	t

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	r·s+p·q
5	r·t

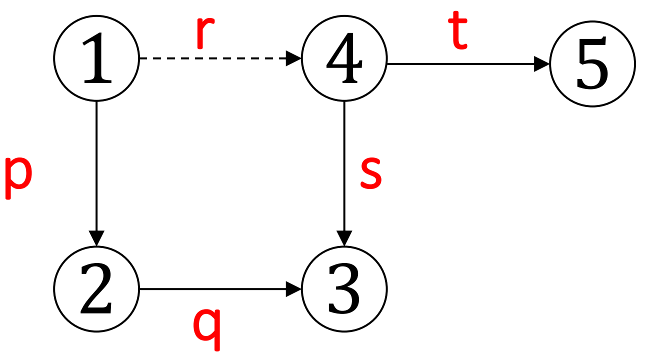
$\in \mathbb{N}(X)$

$X = \{p, q, \ldots, t\}$

$\mathbb{N}[X]$=($\mathbb{N}[X]$, +, ·, 0, 1): Provenance polynomials

Example variant 1

Provenance polynomials $(\mathbb{N}[X], +, \cdot, 0, 1)$



Now assume only certain edges are available (available yes/no or true/false). Which of the points remain reachable?

E

1	2	$p = 1$
2	3	$q = 1$
~~1~~	~~4~~	~~$r = 0$~~
4	3	$s = 1$
4	5	$t = 1$

$Q(z) \text{ :- } E(1,y), E(y,z)$ → Q

Q	
3	$r \cdot s + p \cdot q = 1$
5	$r \cdot t = 0$

$(0 \land 1) \lor (1 \land 1) = 1$

$(0 \land 1) = 0$

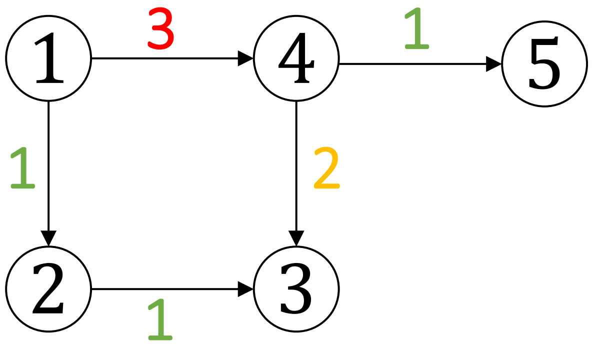
Q: Points reachable in 2 hops, starting at node "1"

{0,1}

$\mathbb{B} = (\mathbb{B}, \lor, \land, 0, 1)$: Boolean algebra

Example variant 2

Provenance polynomials $(\mathbb{N}[X], +, \cdot, 0, 1)$



Now assume passing along an edge needs a certain security clearance (1<2<3).
What clearance do you need for reaching each point?

infix ~> prefix

min[max[3,2], max[1,1]] = 1

E

1	2	$p = 1$
2	3	$q = 1$
1	4	$r = 3$
4	3	$s = 2$
4	5	$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

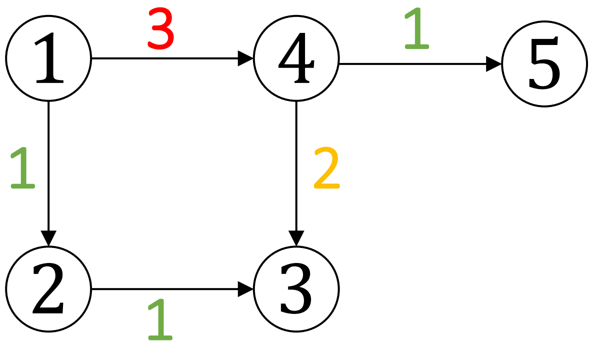
3	$r \cdot s + p \cdot q$	$= 1$
5	$r \cdot t$	$= 3$

max[3,1] = 3

$(\{1,2,3,\infty\}, \min, \max, \infty, 1)$

Example variant 3

Provenance polynomials $(\mathbb{N}[X], +, \cdot, 0, 1)$



Now assume each edge has a weight.
What is the shortest path to reach each point?

E

1	2	$p = 1$
2	3	$q = 1$
1	4	$r = 3$
4	3	$s = 2$
4	5	$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r \cdot s + p \cdot q$	$= 2$
5	$r \cdot t$	$= 4$

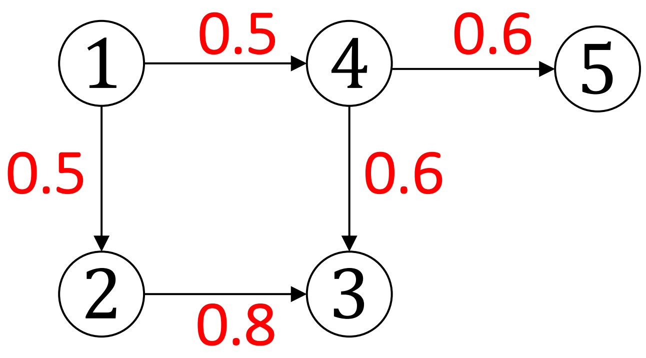
min[3+2,1+1] = 2

3+1 = 4

$\mathbb{T}=(\mathbb{R}_{+}^{\infty},\min,+,\infty,0)$: Tropical semiring

Example variant 4

Provenance polynomials ($\mathbb{N}[X]$, +, ·, 0, 1)



Now assume each edge has a confidence (probability of being available).
What is the probability of the most likely path?

E

1	2	$p = 0.5$
2	3	$q = 0.8$
1	4	$r = 0.5$
4	3	$s = 0.6$
4	5	$t = 0.6$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r{\cdot}s+p{\cdot}q$	$= 0.4$
5	$r{\cdot}t$	$= 0.3$

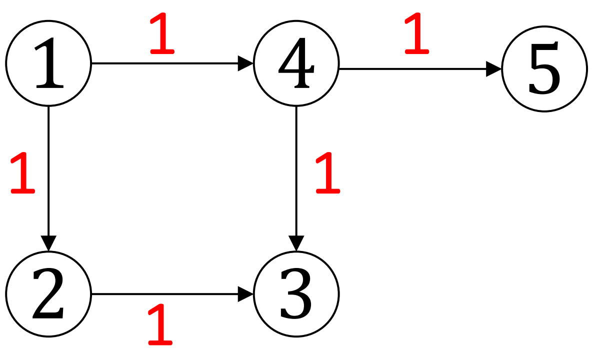
max[0.5·0.6,0.5·0.8] = 0.4

0.5·0.6 = 0.3

$\mathbb{V}$=([0,1],max, · ,0,1): Viterbi semiring (max likely sequence)

Example variant 5

Provenance polynomials $(\mathbb{N}[X], +, \cdot, 0, 1)$



Finally assume we want to calculate the number of paths to a node. We start by annotating the tuples in the database with their duplicity (which is 1 to start with)

E

1	2	$p = 1$
2	3	$q = 1$
1	4	$r = 1$
4	3	$s = 1$
4	5	$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r \cdot s + p \cdot q = 2$	$1 \cdot 1 + 1 \cdot 1 = 2$
5	$r \cdot t = 1$	$1 \cdot 1 = 1$

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{})$$

R

A	B	C
a	b	c
d	b	e
f	g	e

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C

?

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C

?

$Q_1 \cup Q_2$

A	B	C

?

Q

A	C

?

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R})$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
f	g	e

?

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}\ \cup\ \underbrace{\pi_{AC}R \bowtie \pi_{BC}R})$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}\ \cup\ \underbrace{\pi_{AC}R \bowtie \pi_{BC}R})$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{})$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

A more complex example with exponents

$$Q(R) = \pi_{AC}(\pi_{AB}R \bowtie \pi_{BC}R \ \cup \ \pi_{AC}R \bowtie \pi_{BC}R)$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C	
a	c	$2X^2$
a	e	XY
d	c	XY
d	e	$2Y^2+YZ$
f	e	$YZ+2Z^2$

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{})$$

R

A	B	C	
a	b	c	X =2
d	b	e	Y =5
f	g	e	Z =1

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C	
a	c	$2X^2$
a	e	XY
d	c	XY
d	e	$2Y^2+YZ$
f	e	$YZ+2Z^2$

Let's assume bag semantics and duplicities in the input. How many output tuples do we get?

?

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{})$$

R

A	B	C	
a	b	c	$X = 2$
d	b	e	$Y = 5$
f	g	e	$Z = 1$

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C		
a	c	$2X^2$	$=8$
a	e	XY	$=10$
d	c	XY	$=10$
d	e	$2Y^2+YZ$	$=55$
f	e	$YZ+2Z^2$	$=7$

Let's assume bag semantics and duplicities in the input. How many output tuples do we get?

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

A more complex example with exponents

617

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{\pi_{R.A,R.B,R2.C}(R \bowtie_{R.B=R2.B} \rho_{R \to R2} R)} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{\pi_{R.A,R2.B,R.C}(R \bowtie_{R.C=R2.C} \rho_{R \to R2} R)})$$

R

A	B	C	
a	b	c	X =2
d	b	e	Y =5
f	g	e	Z =1

```sql
SELECT A, C, COUNT(*)
FROM (
  SELECT R.A, R.B, R2.C
  FROM R, R R2
  WHERE R.B = R2.B
  UNION ALL
  SELECT R.A, R2.B, R.C
  FROM R, R R2
  WHERE R.C = R2.C) X
GROUP BY A, C
ORDER BY A, C
```

Q

A	C		
a	c	$2X^2$	=8
a	e	XY	=10
d	c	XY	=10
d	e	$2Y^2+YZ$	=55
f	e	$YZ+2Z^2$	=7

	a character varying	c character varying	count bigint
1	a	c	8
2	a	e	10
3	d	c	10
4	d	e	55
5	f	e	7

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

Outline: T2-3/4: Provenance & Reverse Data Management

- T2-3: Provenance
  - Data Provenance
  - The Semiring Framework for Provenance
  - Algebra: Monoids and Semirings
  - Query-rewrite-insensitive provenance
- T2-4: Reverse Data Management
  - View Deletion Problem
  - Resilience & Causality

Why algebra? Think abstraction and generalization

- Abstraction:

- Generalization:

Why algebra? Think abstraction and generalization

- Abstraction: to step back from concrete objects and consider a number of objects with identical properties (all at the same time)
  - main goal in "Abstract algebra"
- Generalization:

For instance, consider the following three objects:

1. The set of functions A, B, C defined on the set $\{1, 2, 3\}$ by
   $A(1) = 1,\ A(2) = 2,\ A(3) = 3,$
   $B(1) = 2,\ B(2) = 3,\ B(3) = 1,$
   $C(1) = 3,\ C(2) = 1,\ C(3) = 2,$
2. The set of complex numbers $a = 1, b = e^{i2\pi/3}, c = e^{i4\pi/3}$.
3. The set of matrices $\alpha = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \beta = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}, \gamma = \begin{pmatrix} -1 & -1 \\ -1 & 0 \end{pmatrix}$

Consider these notations: AB means A(B(x)), ab is ordinary multiplication of complex numbers, and $\alpha\beta$ means ordinary matrix multiplication. Verify the following "multiplication" tables:

	A	B	C
A	A	B	C
B	B	C	A
C	C	A	B

	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

	α	β	γ
α	α	β	γ
β	β	γ	α
γ	γ	α	β

Notice that these tables are identical. Then let us by *abstraction* define an abstract object which is the set of three elements $\{e, g, g^{-1}\}$ paired with a binary operator $\cdot$ such that set acts on itself in the following manner with respect to the operator:

$\cdot$	e	g	g^{-1}
e	e	g	g^{-1}
g	g	g^{-1}	e
g^{-1}	g^{-1}	e	g

In Group Theory an object with such structure is called the cyclic group of order three. Then the examples above are representations of this abstract object. It is an abstract object because while we have now given it a definition, notice that it is itself a stand-in for a variety of objects that have the properties that it demonstrates. You might even consider the abstract object to be more of a set

Example on the right from: https://matheducators.stackexchange.com/questions/10949/what-is-abstraction-and-generalization/10957

Why algebra? Think abstraction and generalization

- Abstraction: to step back from concrete objects and consider a number of objects with identical properties (all at the same time)
  - main goal in "Abstract algebra"
- Generalization: applying a solution for one problem to other problems
  - e.g. Algorithms: finding the shortest path not just in one graph but any graph

For instance, consider the following three objects:

1. The set of functions A, B, C defined on the set $\{1, 2, 3\}$ by
   $A(1) = 1,\ A(2) = 2,\ A(3) = 3,$
   $B(1) = 2,\ B(2) = 3,\ B(3) = 1,$
   $C(1) = 3,\ C(2) = 1,\ C(3) = 2,$
2. The set of complex numbers $a = 1, b = e^{i2\pi/3}, c = e^{i4\pi/3}$.
3. The set of matrices $\alpha = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \beta = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}, \gamma = \begin{pmatrix} -1 & -1 \\ -1 & 0 \end{pmatrix}$

Consider these notations: AB means A(B(x)), ab is ordinary multiplication of complex numbers, and $\alpha\beta$ means ordinary matrix multiplication. Verify the following "multiplication" tables:

	A	B	C
A	A	B	C
B	B	C	A
C	C	A	B

	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

	α	β	γ
α	α	β	γ
β	β	γ	α
γ	γ	α	β

Notice that these tables are identical. Then let us by *abstraction* define an abstract object which is the set of three elements $\{e, g, g^{-1}\}$ paired with a binary operator $\cdot$ such that set acts on itself in the following manner with respect to the operator:

$\cdot$	e	g	g^{-1}
e	e	g	g^{-1}
g	g	g^{-1}	e
g^{-1}	g^{-1}	e	g

In Group Theory an object with such structure is called the cyclic group of order three. Then the examples above are representations of this abstract object. It is an abstract object because while we have now given it a definition, notice that it is itself a stand-in for a variety of objects that have the properties that it demonstrates. You might even consider the abstract object to be more of a set

Example on the right from: https://matheducators.stackexchange.com/questions/10949/what-is-abstraction-and-generalization/10957

Let's start with groups! Why groups?

- Groups are one of the most important structures studied in abstract algebra
- What is so special about groups?

Groups have the minimum properties needed to solve equations

($\mathbb{Z}$,+, 0): Integers under addition

Equation	Properties
$x + 3 = 5$	✓ Integers under +
$[x + 3] + (-3) = 5 + (-3)$	✓ Inverses
$[x + 3] + (-3) = 2$	✓ Closed under +
$x + [3 + (-3)] = 2$	✓ Associativity
$x + 0 = 2$	✓ Identity
$x = 2$	

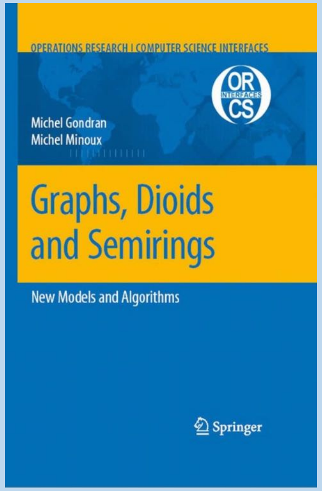
$(a + b) + c$

$a + (b + c)$

Screenshot from: Socratica: Abstract Algebra: Motivation for the definition of a group, https://www.youtube.com/watch?v=yHq_yzYZV6U

Why something weaker than groups?

- For some important computational problems like Dynamic Programming, we don't need to "solve equations".
  - Thus we don't need an inverse ("we don't need to go back")
- Let's look at weaker structures



Preface

During the last two or three centuries, most of the developments in science (in particular in Physics and Applied Mathematics) have been founded on the use of classical algebraic structures, namely groups, rings and fields. However many situations can be found for which those usual algebraic structures do not necessarily provide the most appropriate tools for modeling and problem solving. ...

Screenshot from: Gondran, Minoux. "Graphs, Dioids and Semirings: New Models and Algorithms", 2008. https://www.springer.com/gp/book/9780387754499

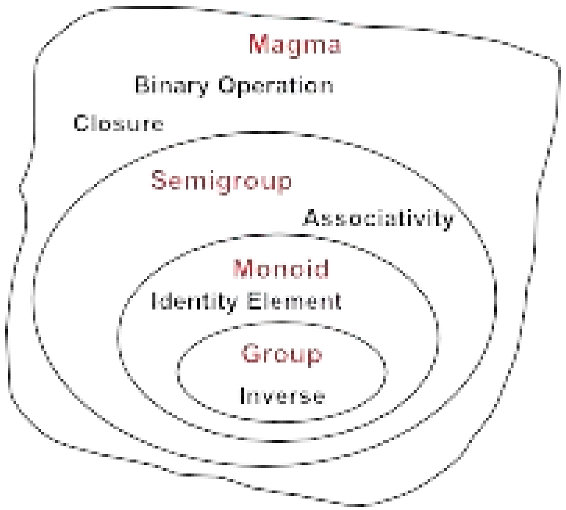
Group-like structures

Set S

+ 1. Closed binary operation ⊕:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

Group



Group-like structures

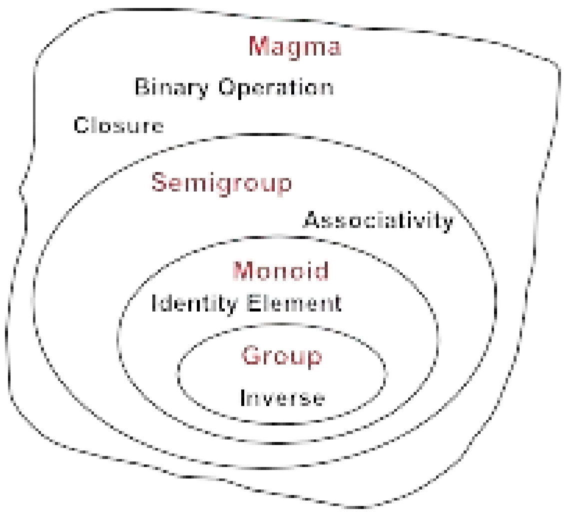
Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$



Group-like structures

Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$

+ 3. Identity element:
$\exists e \in S.\ \forall x \in S.\ [e \oplus x = x \oplus e = x]$

Monoid $(S,\oplus,e)$

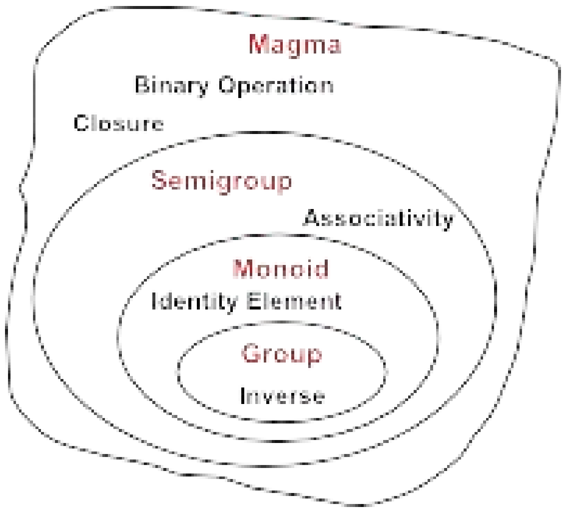


Figure credits: https://www.euclideanspace.com/maths/discrete/groups/monoid/index.htm

Group-like structures

Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$

+ 3. Identity element:
$\exists e \in S.\ \forall x \in S.\ [e \oplus x = x \oplus e = x]$

Monoid $(S,\oplus,e)$

+ 4. Inverse:
$\forall x \in S.\ \exists x^{-1} \in S.\ [x^{-1} \oplus x = x \oplus x^{-1} = e]$

Group $(S,\oplus,e)$

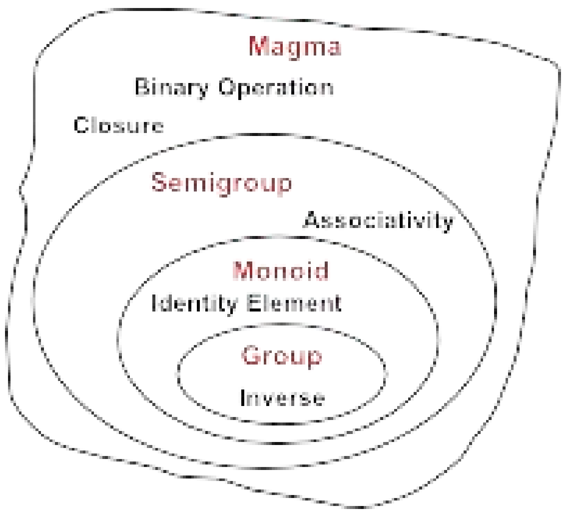


Figure credits: https://www.euclideanspace.com/maths/discrete/groups/monoid/index.htm

Group-like structures

Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$

+ 3. Identity element:
$\exists e \in S.\ \forall x \in S.\ [e \oplus x = x \oplus e = x]$

Monoid $(S,\oplus,e)$ → + 5. Commutativity: $x \oplus y = y \oplus x$ → Commutative Monoid $(S,\oplus,e)$

+ 4. Inverse:
$\forall x \in S.\ \exists x^{-1} \in S.\ [x^{-1} \oplus x = x \oplus x^{-1} = e]$

Commutative Monoid $(S,\oplus,e)$ + 4. → Abelian Group $(S,\oplus,e)$

Group $(S,\oplus,e)$ → + 5. → Abelian Group $(S,\oplus,e)$

What are intuitive examples for:

- a group ?
- monoids (that are not groups) ?
- semi-groups (that are not monoids)? ?

Group-like structures

Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$

+ 3. Identity element:
$\exists e \in S.\ \forall x \in S.\ [e \oplus x = x \oplus e = x]$

Monoid $(S,\oplus,e)$ → Commutative Monoid $(S,\oplus,e)$

+ 5. Commutativity: $x \oplus y = y \oplus x$

+ 4. Inverse:
$\forall x \in S.\ \exists x^{-1} \in S.\ [x^{-1} \oplus x = x \oplus x^{-1} = e]$

+ 4.

Group $(S,\oplus,e)$ → Abelian Group $(S,\oplus,e)$

+ 5.

What are intuitive examples for:

- a group
  - $(\mathbb{Z},+,0)$: Integers under addition
- monoids (that are not groups)

?

- semi-groups (that are not monoids)?

?

Group-like structures

Set S

+ 1. Closed binary operation ⊕:
If x,y∈S then (x⊕y)∈S

Magma (S,⊕)

+ 2. Associativity:
x⊕(y⊕z) = (x⊕y)⊕z

Semi-group (S,⊕)

+ 3. Identity element:
∃e∈S. ∀x∈S. [e⊕x = x⊕e = x]

Monoid (S,⊕,e) → Commutative Monoid (S,⊕,e)

+ 5. Commutativity: x⊕y = y⊕x

+ 4. Inverse:
∀x∈S. ∃x^{-1}∈S. [x^{-1}⊕x = x⊕x^{-1} = e]

+ 4.

Group (S,⊕,e) → Abelian Group (S,⊕,e)

+ 5.

What are intuitive examples for:

- a group
  - (ℤ,+, 0): Integers under addition
- monoids (that are not groups)
  - (ℕ,+, 0): Natural numbers under add. {0, 1, ...}
  - (ℝ,min,∞): minimum has no inverse
  - String concatenation with null string ε
  - Square matrices under matrix multiplication
  - (P(S),∪): Power set under union
- semi-groups (that are not monoids)?

?

Group-like structures

Set S

+ 1. Closed binary operation $\oplus$:
If $x,y \in S$ then $(x \oplus y) \in S$

Magma $(S,\oplus)$

+ 2. Associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

Semi-group $(S,\oplus)$

+ 3. Identity element:
$\exists e \in S.\ \forall x \in S.\ [e \oplus x = x \oplus e = x]$

Monoid $(S,\oplus,e)$ → + 5. Commutativity: $x \oplus y = y \oplus x$ → Commutative Monoid $(S,\oplus,e)$

+ 4. Inverse:
$\forall x \in S.\ \exists x^{-1} \in S.\ [x^{-1} \oplus x = x \oplus x^{-1} = e]$

Commutative Monoid $(S,\oplus,e)$ → + 4. → Abelian Group $(S,\oplus,e)$

Group $(S,\oplus,e)$ → + 5. → Abelian Group $(S,\oplus,e)$

What are intuitive examples for:

- a group
  - $(\mathbb{Z},+,0)$: Integers under addition
- monoids (that are not groups)
  - $(\mathbb{N},+,0)$: Natural numbers under add. {0, 1, ...}
  - $(\mathbb{R},\min,\infty)$: minimum has no inverse
  - String concatenation with null string ε
  - Square matrices under matrix multiplication
  - $(P(S),\cup)$: Power set under union
- semi-groups (that are not monoids)?
  - Even numbers under multiplication
  - $(\mathbb{N}_1,+)$: Positive integers under add. {1, 2, ...}
  - String concatenation without null string