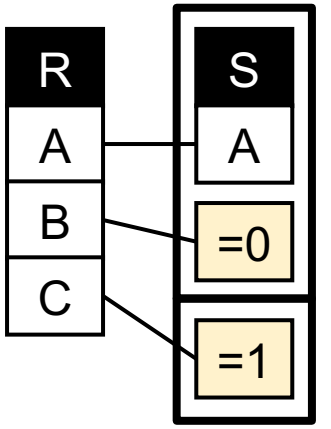
DATALAB
@Northeastern



A Principled Solution to the Disjunction Problem of Diagrammatic Query Representations

Wolfgang Gatterbauer

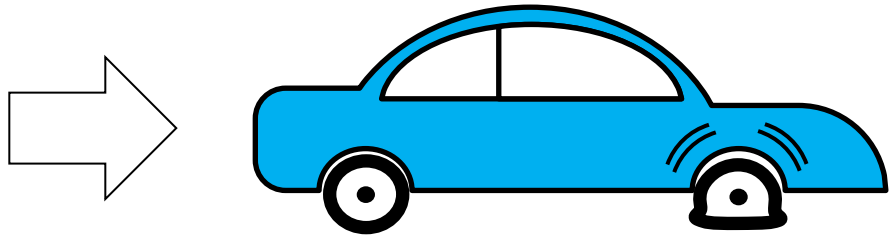
June 3, 2026 (SIGMOD'26)

Please ask questions or leave feedback ☺

An anonymous feedback form is linked from the project web page: https://RelationalDiagrams.com

Why is visualizing disjunctive information harder?

I see a car that is blue
and that has a flat tire

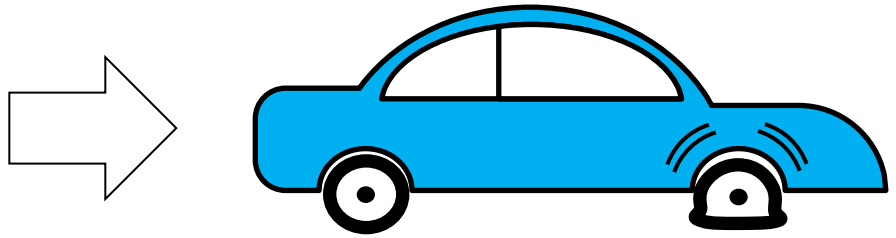


I see a car that is blue
or that has a flat tire

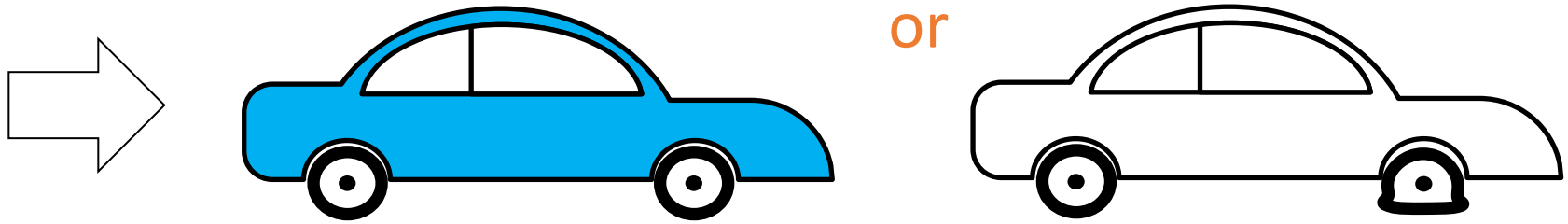
?

Why is visualizing disjunctive information harder?

I see a car that is blue
and that has a flat tire

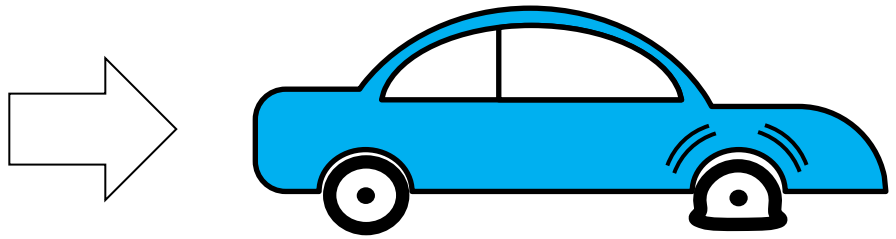


I see a car that is blue
or that has a flat tire

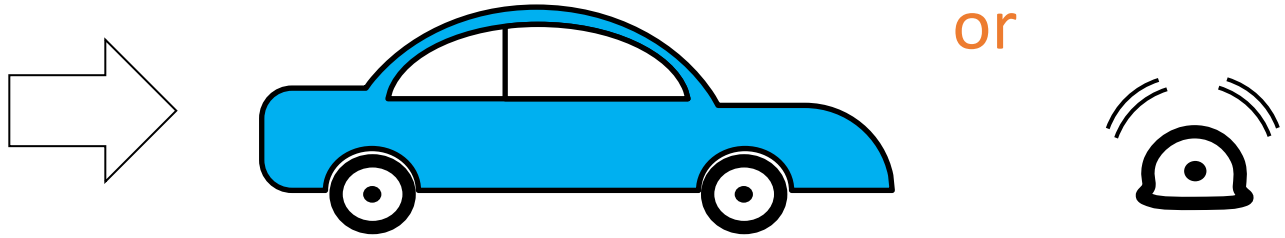


Why is visualizing disjunctive information harder?

I see a car that is blue
and that has a flat tire

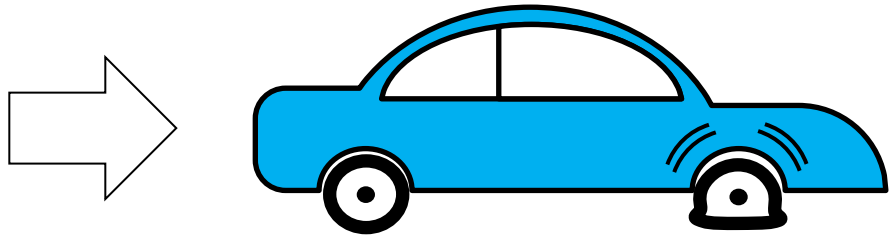


I see a car that is blue
or that has a flat tire



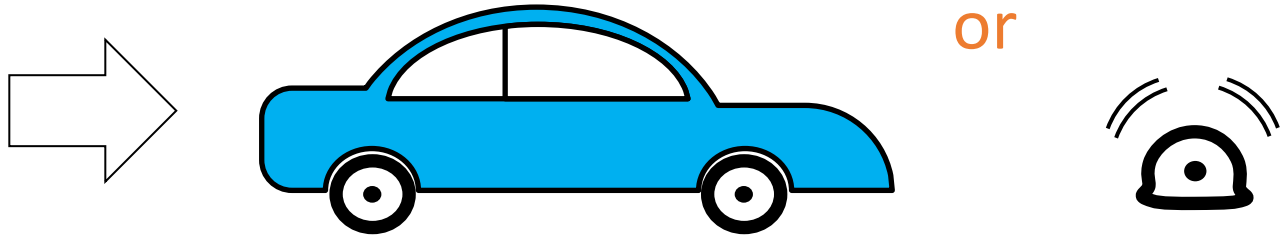
Why is visualizing disjunctive information harder?

I see a car that is blue
and that has a flat tire



$\sigma_{\text{color="blue"} \land \text{tire="flat}}(\ldots)$

I see a car that is blue
or that has a flat tire

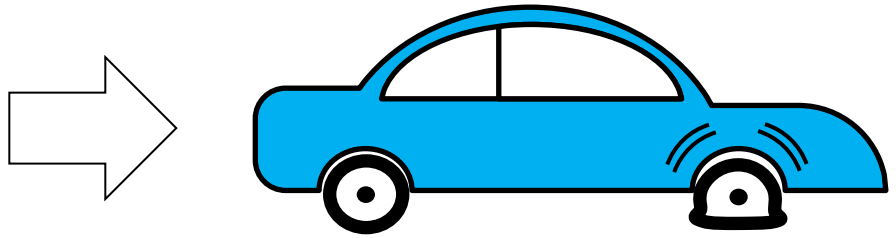


$\sigma_{\text{color="blue"} \lor \text{tire="flat}}(\ldots)$

Why is visualizing disjunctive information harder?

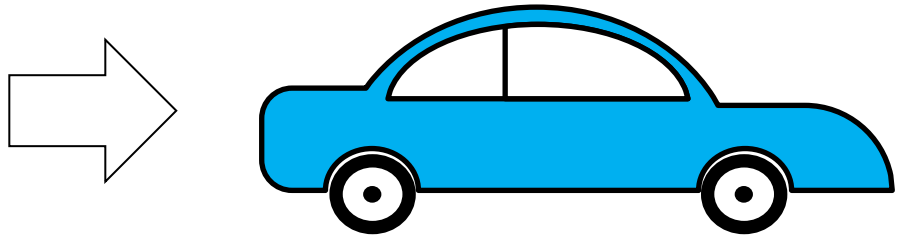
"A situation only displays conjunctive information."†

I see a car that is blue and that has a flat tire



$\sigma_{\text{color="blue"}}(\sigma_{\text{tire="flat"}}(\ldots))$

I see a car that is blue or that has a flat tire



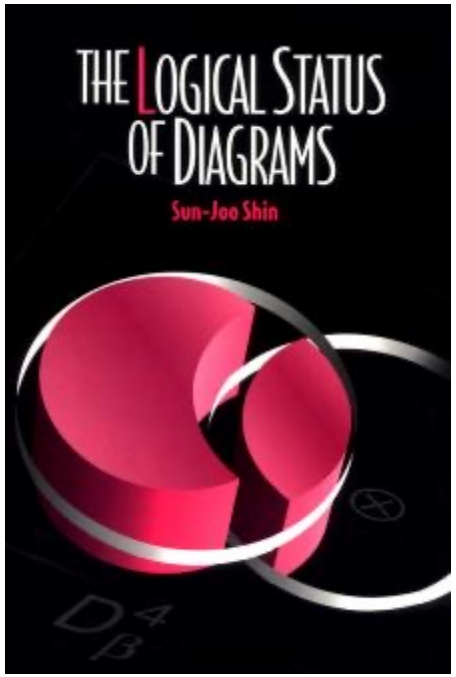
or

$\sigma_{\text{color="blue"} \vee \text{tire="flat"}}(\ldots)$

† Sun-Joo Shin, "The Logical Status of Diagrams", Cambridge University Press 1995. https://doi.org/10.1017/CBO9780511574696

Wolfgang Gatterbauer. *A Principled Solution to the Disjunction Problem of Diagrammatic Query Representations*, SIGMOD 2026. https://RelationalDiagrams.com

Disjunctions have long been treated as hard for diagrams

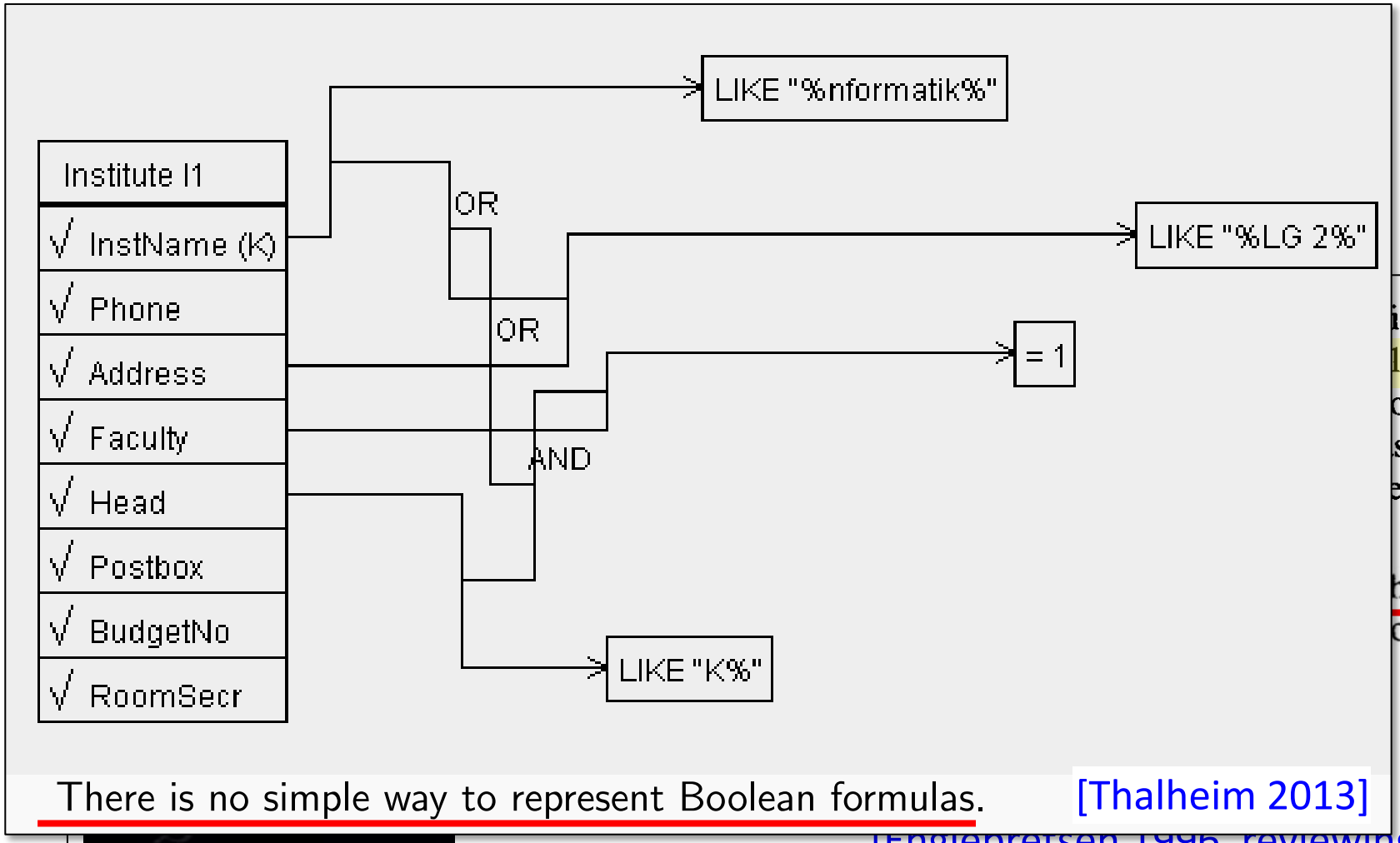


Still, not all relations can be viewed as membership or inclusion. Shin has been careful throughout her book to restrict herself to monadic systems. Relations per se (polyadic predicates) are not considered. And while it may be true that the formation of a system (such as Venn-II) that is provably both sound and complete would help mitigate the prejudice

perception. In her discussion of perception she shows that disjunctive information is not representable in *any* system. In doing so she relies on

[Englebretsen 1996, reviewing Shin 1995]

Disjunctions have long been treated as hard for diagrams



There is no simple way to represent Boolean formulas. [Thalheim 2013]

inclusion. Shin
lf to monadic
onsidered. And
s Venn-II) that
e the prejudice

hat disjunctive
o she relies on

[Englebretsen 1996, reviewing Shin 1995]

Disjunctions have long been treated as hard for diagrams

A few remarks should be made about diagraming compound statements with parentheses, such as: $(A \vee B) \supset (B \vee C)$. This asserts that if the relation "$A \vee B$" is a true relation then the relation "$B \vee C$" must also be true.

How can this be shown on the Venn circles? We can of course expand the statement by algebraic methods into a longer statement without parentheses, then make our diagram; but if we do this, we might as well proceed to use algebraic methods throughout. On the other hand, there seems to be no simple way in which the statement, as it stands, can be diagramed. The best procedure is probably to follow the suggestion Peirce made for handling parenthetical statements in class logic—to make separate Venn diagrams for the two relations inside of parentheses, then connect them with another Venn diagram shaded to represent implication.

[Gardner 1958]

There is no simple way to represent Boolean formulas. [Thalheim 2013]

[Englebretsen 1996, reviewing Shin 1995]

Disjunctions have long been treated as hard for diagrams

It is only disjunctions of conjunctions that cause some inconvenience; such as "Either some A is B while everything is either A or B, or else All A is B while some B is not A."

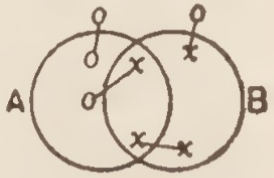


Fig. 59

[Peirce ~1903]

made about diagraming compound
uch as: $(A \vee B) \supset (B \vee C)$. This as-
B" is a true relation then the relation

the Venn circles? We can of course
raic methods into a longer statement
e our diagram; but if we do this, we
lgebraic methods throughout. On the
e no simple way in which the state-
gramed. The best procedure is prob-
Peirce made for handling parentheti-
to make separate Venn diagrams for
entheses, then connect them with an-
represent implication.

[Gardner 1958]

LG 2%"

inclusion. Shin
lf to monadic
onsidered. And
s Venn-II) that
e the prejudice

hat disjunctive
o she relies on

There is no simple way to represent Boolean formulas. [Thalheim 2013]

[Englebretsen 1996, reviewing Shin 1995]

Disjunctions have long been treated as hard for diagrams

It is only disjunctions of conjunctions that cause some inconvenience; such as "Either some A is B while everything is either A or B, or else All A is B while some B is not A." Even here there is no serious difficulty. Fig. 59 expresses this proposition. It is merely that there is a greater complexity in the expression than is essential to the meaning. There is, however, a very easy and very useful way of avoiding this. It is to draw an Euler's Diagram of Euler's Diagrams each surrounded by a circle to represent its Universe of Hypothesis. There will be no need of connecting lines in the enclosing diagram, it being understood that its compartments contain the several possible cases. Thus, Fig. 60 expresses the same proposition as Fig. 59.

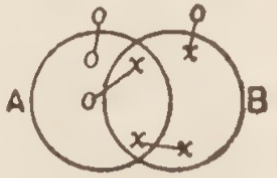


Fig. 59

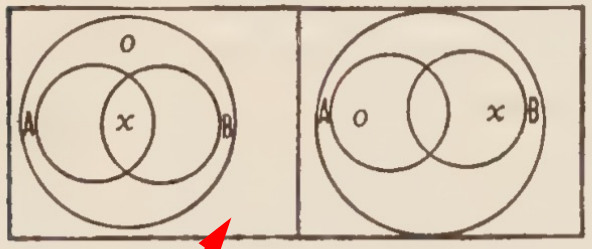


Fig. 60

[Peirce ~1903]

made about diagraming compound
uch as: $(A \vee B) \supset (B \vee C)$. This as-
B" is a true relation then the relation

the Venn circles? We can of course
raic methods into a longer statement
ke our diagram; but if we do this, we
lgebraic methods throughout. On the
e no simple way in which the state-
gramed. The best procedure is prob-
Peirce made for handling parentheti-
to make separate Venn diagrams for
entheses, then connect them with an-
represent implication.

[Gardner 1958]

LG 2%"

inclusion. Shin
lf to monadic
onsidered. And
s Venn-II) that
e the prejudice

hat disjunctive
o she relies on

There is no simple way to represent Boolean formulas. [Thalheim 2013]

[Englebretsen 1996, reviewing Shin 1995]

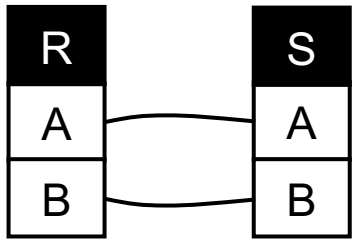
case-splitting may cause exponential blow-up

A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL



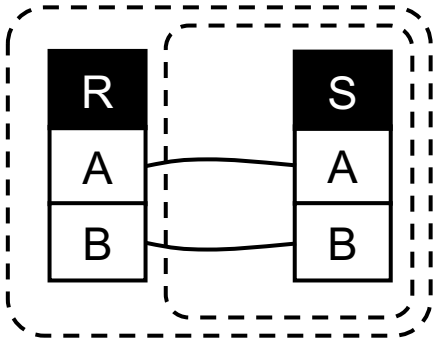
A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL

1. Boxes determine the negation hierarchy.



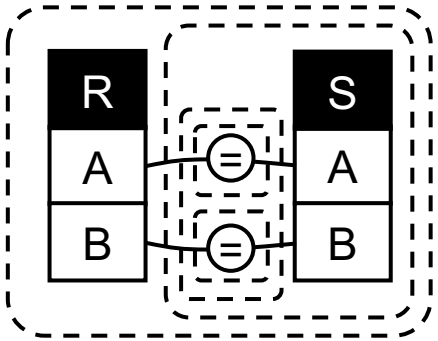
A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL

1. Boxes determine the negation hierarchy.
2. Anchors say where predicates live.



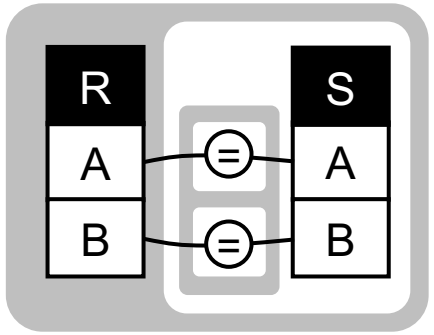
A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL

1. Boxes determine the negation hierarchy.
2. Anchors say where predicates live.
3. Peirce-shading simplifies the reading.



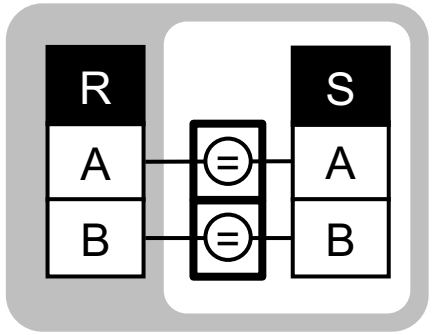
A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL

1. Boxes determine the negation hierarchy.
2. Anchors say where predicates live.
3. Peirce-shading simplifies the reading.
4. De Morgan-fuse boxes make or explicit.



$\forall r \in R\ [\ \exists s \in S\ [(r.A = s.A \vee r.B = s.B)]]$

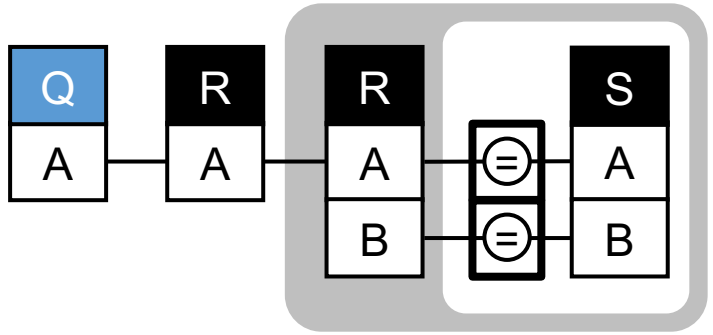
A principled solution to the disjunction problem

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

THE SOLUTION IN A NUTSHELL

1. Boxes determine the negation hierarchy.
2. Anchors say where predicates live.
3. Peirce-shading simplifies the reading.
4. De Morgan-fuse boxes make or explicit.



$\{q(A) \mid \exists r_2 \in R\,[q.A=r_2.A \wedge \forall r \in R\,[r_2.A=r.A \wedge \exists s \in S\,[(r.A=s.A \vee r.B=s.B)]]]\}$

Details at the poster

Next: intuition!

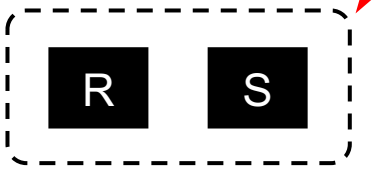
Conjunction

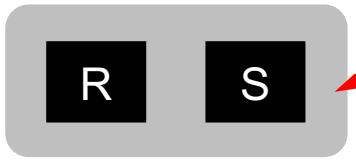
∧: juxtaposition of assertions

R ∧ S

Negation

¬: "cut" (denial)†
partition into inside and outside "zones"

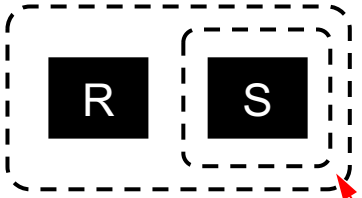


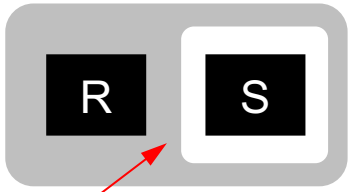


shading indicates polarity of a zone:
positive/monotone or negative/antitone

¬(R ∧ S)

Material Implication





¬(R ∧ ¬(S))

R ⇒ S

implication: nesting of cuts

† Variation on Peirce's concept of a "cut" (denial) in the "sheet of assertion": Charles Sanders Peirce. 1933. Collected Papers. Vol. 4. Harvard University Press. https://archive.org/details/collectedpaperso0000unse_r5j9/.
Wolfgang Gatterbauer. *A Principled Solution to the Disjunction Problem of Diagrammatic Query Representations*, SIGMOD 2026. https://RelationalDiagrams.com

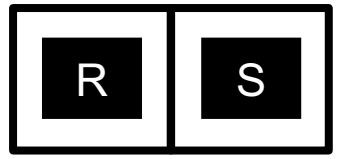
Conjunction

∧: juxtaposition of assertions

R ∧ S

Disjunction

∨: alternatives
via "De Morgan-fuse boxes"

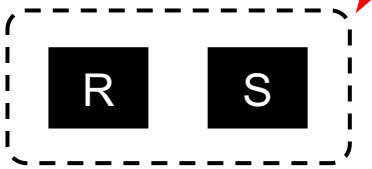


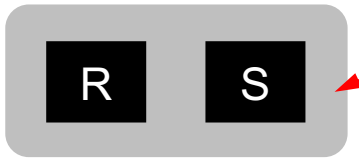
3 zones

R ∨ S

Negation

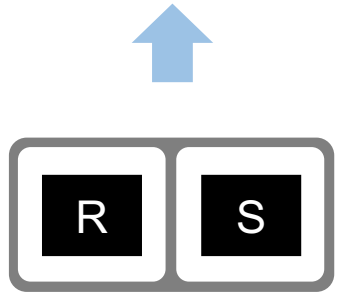
¬: "cut" (denial)†
partition into inside and outside "zones"



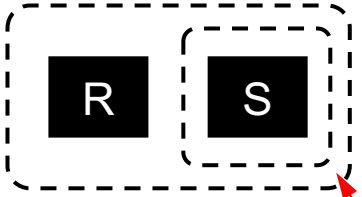


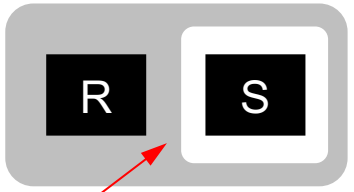
shading indicates polarity of a zone:
positive/monotone or negative/antitone

¬(R ∧ S)



Material Implication

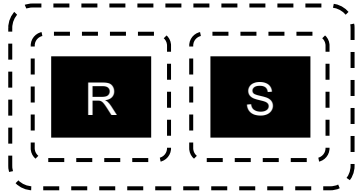


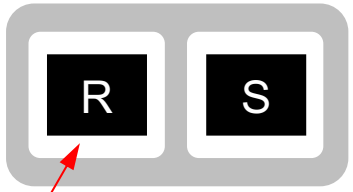


¬(R ∧ ¬(S))

R ⇒ S

implication: nesting of cuts





4 zones

¬(¬(R) ∧ ¬(S))

R ∨ S

disjunction does not
change the polarity

† Variation on Peirce's concept of a "cut" (denial) in the "sheet of assertion": Charles Sanders Peirce. 1933. Collected Papers. Vol. 4. Harvard University Press. https://archive.org/details/collectedpaperso0000unse_r5j9/.
Wolfgang Gatterbauer. *A Principled Solution to the Disjunction Problem of Diagrammatic Query Representations*, SIGMOD 2026. https://RelationalDiagrams.com

Propositional Logic → First-Order Logic

¬(R ∧ ¬(S))

¬(∃r∈R ∧ ¬(∃s∈S))

¬(∃r∈R [¬(∃s∈S)])

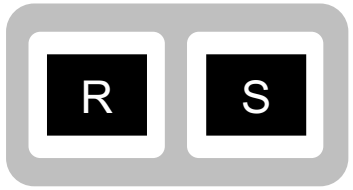
R ⇒ S

∀r∈R [∃s∈S]

two nested zones:
∀ outer [∃ inner]

R := ∃r∈R

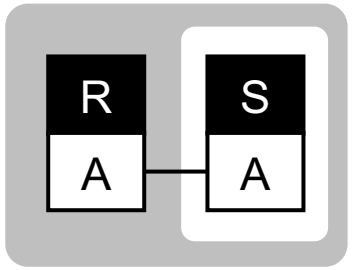
S := ∃s∈S



R ∨ S

Propositional Logic → First-Order Logic

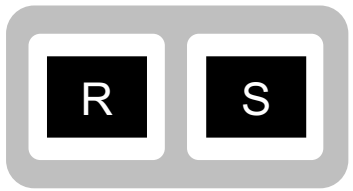
for all tuples in R
there is a tuple in S
that joins on A



$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$

$\forall r \in R\ [\ \exists s \in S\ ]$

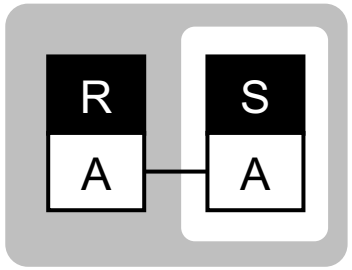
two nested zones:
∀ outer [∃ inner]



$\exists r \in R \vee \exists s \in S$

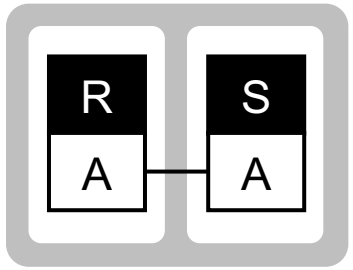
Propositional Logic → First-Order Logic

for all tuples in R
there is a tuple in S
that joins on A



$\forall r \in R\ [\ \exists s \in S\ [r.A=s.A]\ ]$

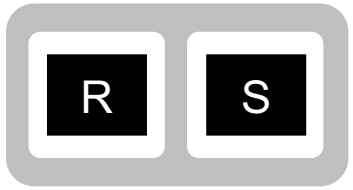
?



$(\exists r \in R \vee \exists s \in S)\ [r.A=s.A]$

$\forall r \in R\ [\ \exists s \in S\ ]$

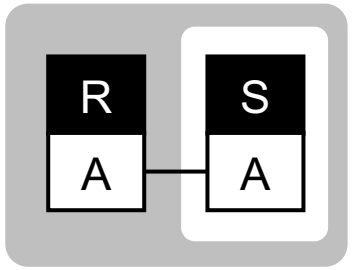
two nested zones:
∀ outer [∃ inner]



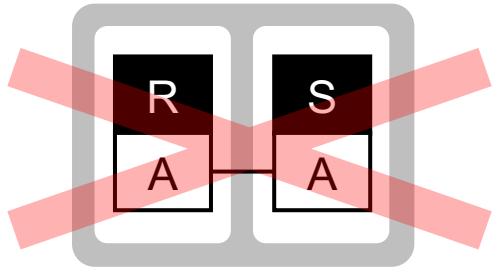
$\exists r \in R \vee \exists s \in S$

Propositional Logic → First-Order Logic

for all tuples in R there is a tuple in S that joins on A



$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$



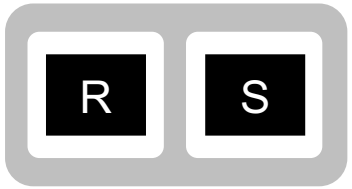
$(\exists r \in R \vee \exists s \in S)\ [r.A = s.A]$

not well-formed: r and s are not bound in a shared scope!

a predicate must be inside the scopes of both table occurrences

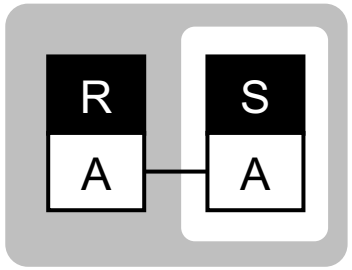
two nested zones:
∀ outer [∃ inner]

$\forall r \in R\ [\ \exists s \in S\ ]$

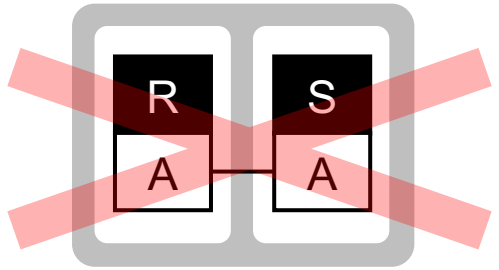


$\exists r \in R \vee \exists s \in S$

Predicate edges need anchors



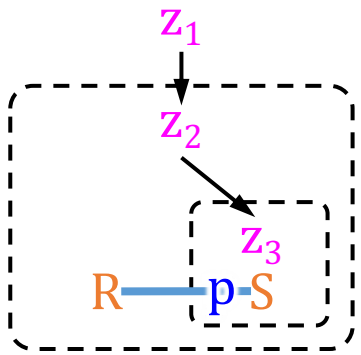
$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$



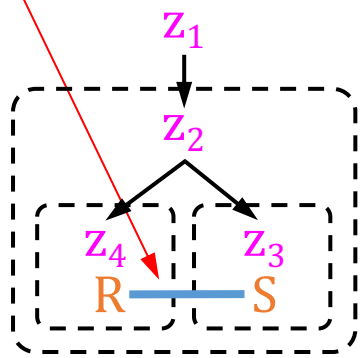
$(\exists r \in R \vee \exists s \in S)\ [r.A = s.A]$

not well-formed: r and s are not bound in a shared scope!

a predicate must be inside the scopes of both table occurrences

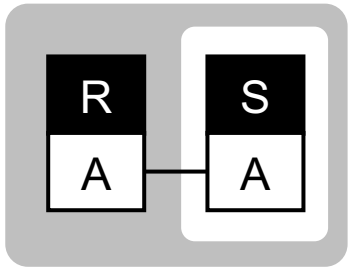


✓

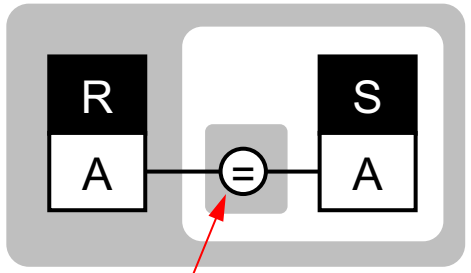


✗

Predicate edges need anchors

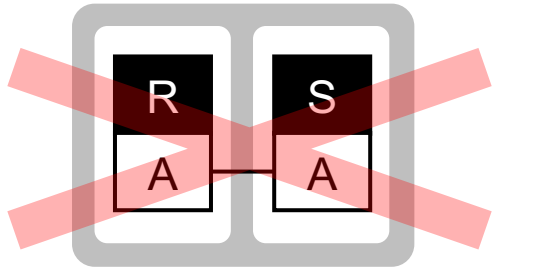


$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$



$\forall r \in R\ [\ \exists s \in S [\neg (r.A = s.A)]]$

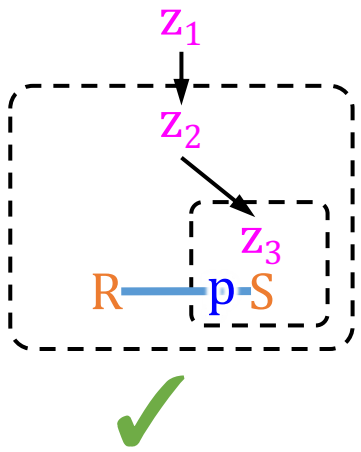
each predicate edge needs an independent anchor that determines its logical zone



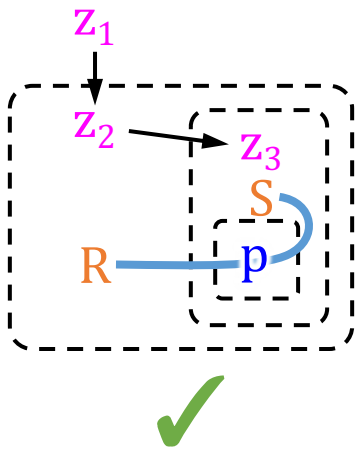
$(\exists r \in R \vee \exists s \in S)\ [r.A = s.A]$

not well-formed: r and s are not bound in a shared scope!

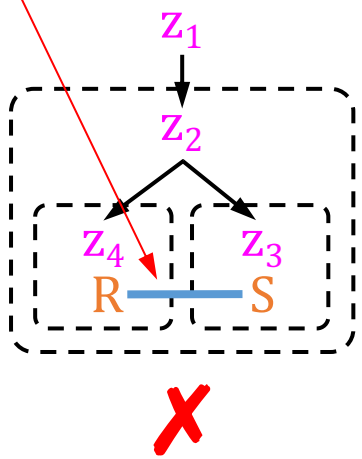
a predicate must be inside the scopes of both table occurrences



✓

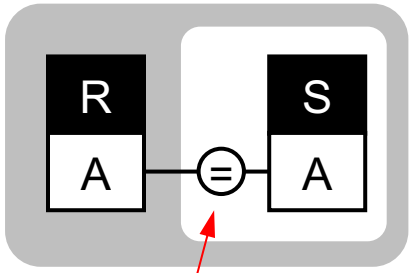


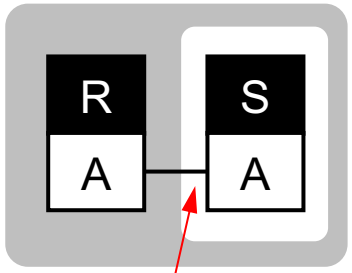
✓



✗

Predicate edges need anchors



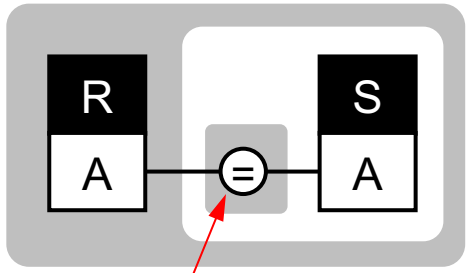


$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$

actual meaning

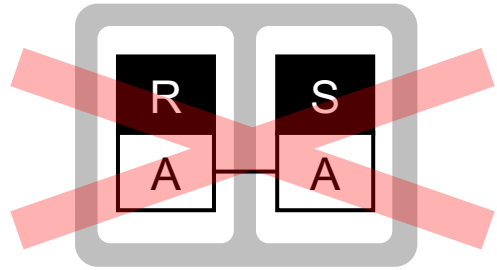
visual shortcut

No anchor is needed in the innermost shared table scope



$\forall r \in R\ [\ \exists s \in S [\neg (r.A = s.A)]]$

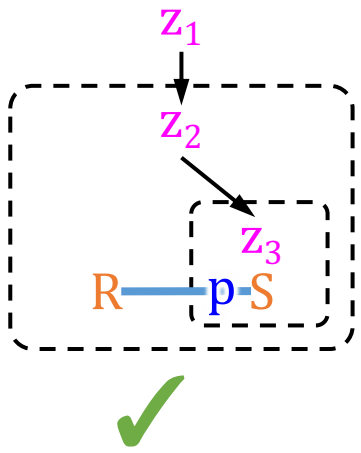
each predicate edge needs an independent anchor that determines its logical zone



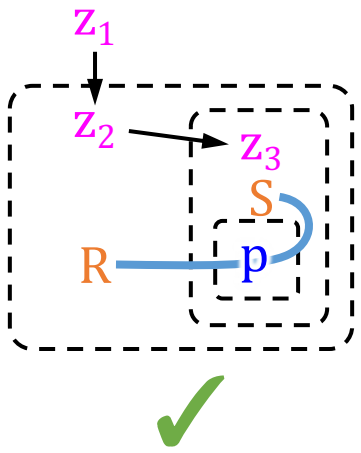
$(\exists r \in R \vee \exists s \in S)\ [r.A = s.A]$

not well-formed: r and s are not bound in a shared scope!

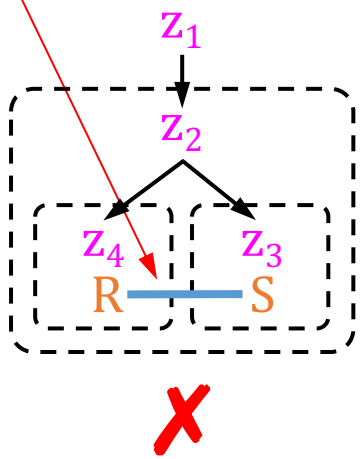
a predicate must be inside the scopes of both table occurrences



✓



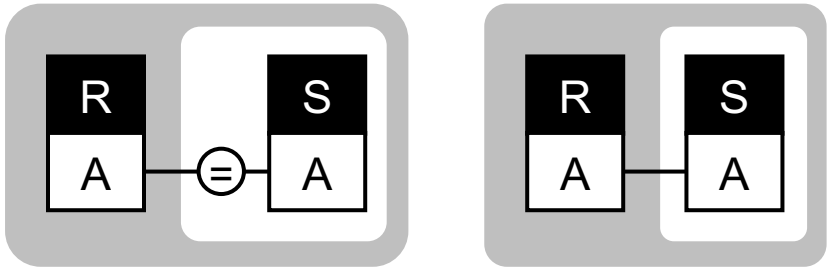
✓



✗

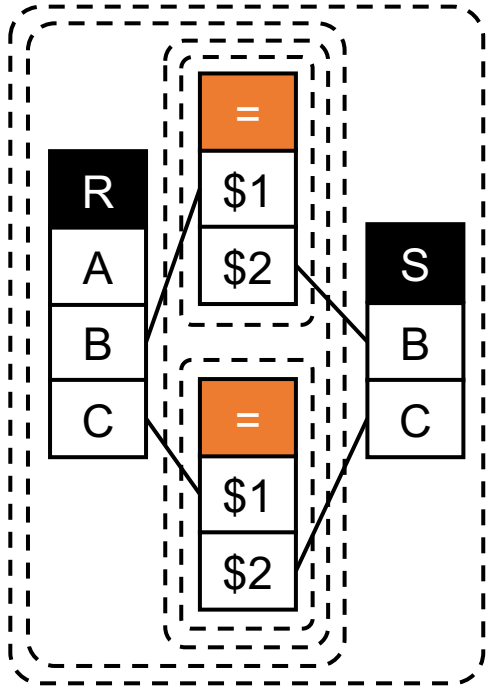
Nested Disjunctions

formal semantics (anchor relations + visual shortcuts) given in the paper



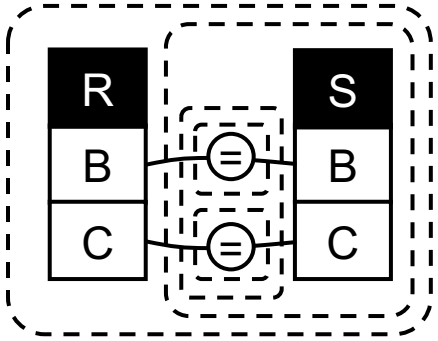
$\forall r \in R\ [\ \exists s \in S\ [r.A = s.A]\ ]$

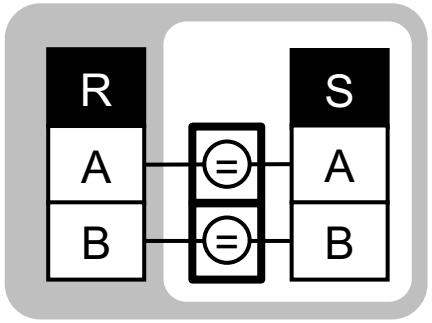
1 anchor relations



2 predicate anchors

3 Visual shortcuts: Peirce shading & De Morgan-fuse boxes



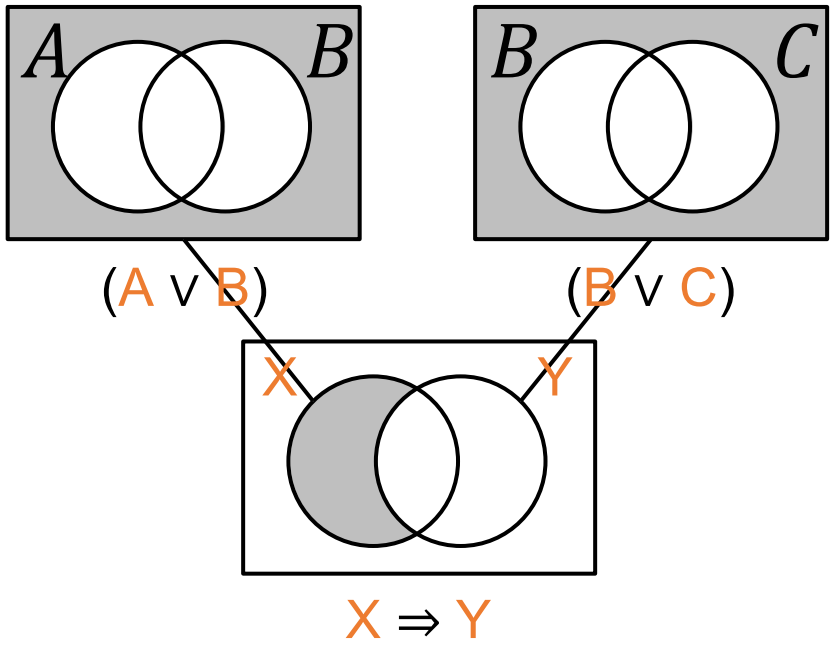


$\forall r \in R\ [\ \exists s \in S\ [\quad (r.A = s.A \quad \vee \quad r.B = s.B)\ ]\ ]$

$\forall r \in R\ [\ \exists s \in S\ [\neg(\neg(r.A = s.A) \wedge \neg(r.B = s.B))\ ]\ ]$

$(\exists j_1 \in \text{"="}\ [r.A = j_1.\$1 \wedge s.A = j_1.\$2])$ similarly

(A ∨ B) ⇒ (B ∨ C)

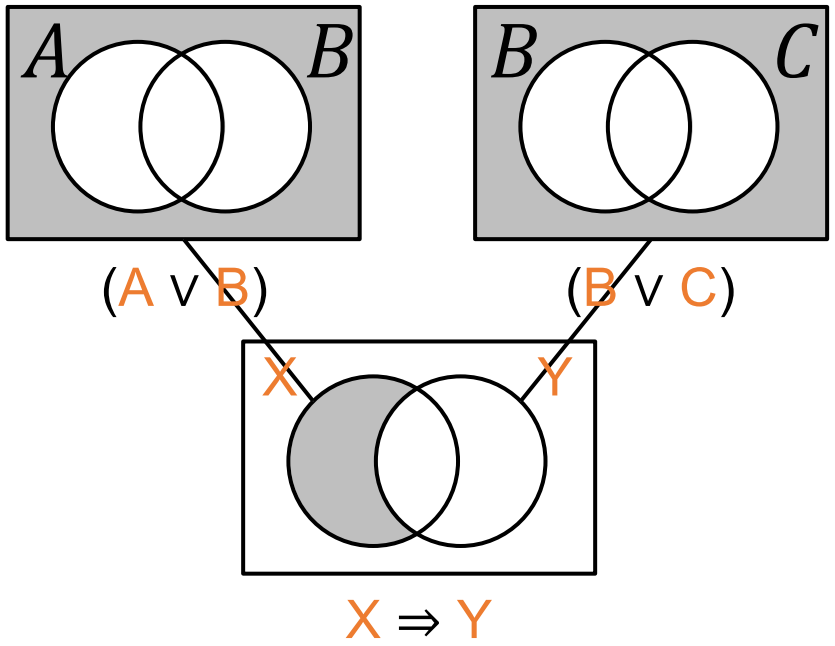


?

there seems to be no simple way in which the statement, as it stands, can be diagramed. The best procedure is probably to follow the suggestion Peirce made for handling parenthetical statements in class logic—to make separate Venn diagrams for the two relations inside of parentheses, then connect them with another Venn diagram shaded to represent implication.

[Gardner 1958]

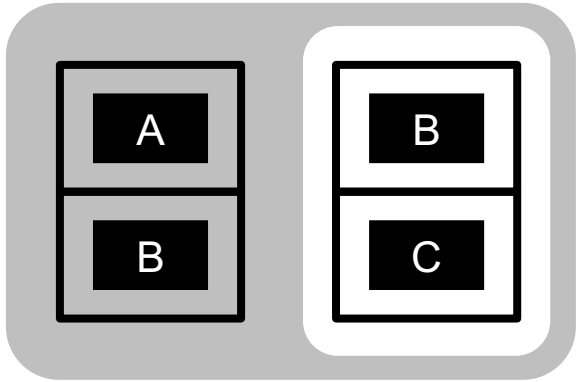
(A ∨ B) ⇒ (B ∨ C)



there seems to be no simple way in which the statement, as it stands, can be diagramed. The best procedure is probably to follow the suggestion Peirce made for handling parenthetical statements in class logic—to make separate Venn diagrams for the two relations inside of parentheses, then connect them with another Venn diagram shaded to represent implication.

[Gardner 1958]

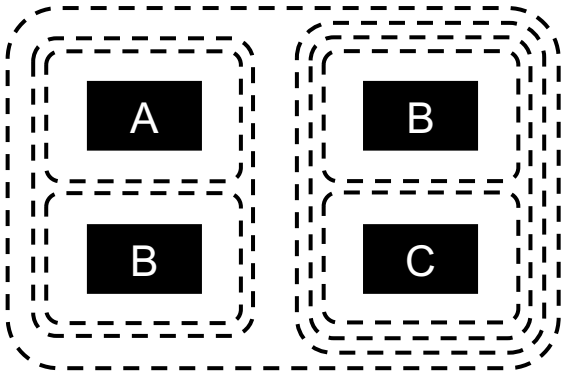
Representation B: one diagram, no case split



(A ∨ B) ⇒ (B ∨ C)

¬(A ∨ B) ∨ (B ∨ C)

¬(¬(¬(A) ∧ ¬(B)) ∧ ¬(¬(¬(B) ∧ ¬(C))))

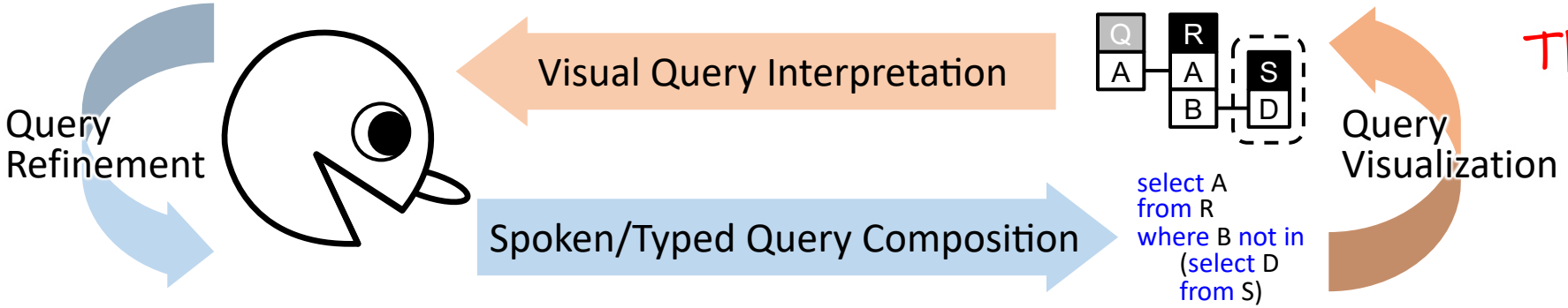


Takeaways & next steps

well-formed TRC query ⇄ Representation B diagram

same semantics,
same relational pattern & linear size,
syntactic safety is preserved

- Extensions (e.g. grouping, recursion, bags): Towards a catalogue of query patterns (CIDR'26 "*Database Research needs an Abstract Relational Query Language*")
- Beyond expressiveness: How languages help humans and machines read, write, understand, and modify queries? (SIGMOD'26 tutorial "*Relational Language Design*")
- A future DB interaction: LLMs compose, humans read, refine interactively



Thanks ☺

An anonymous feedback form is linked from the project web page: https://RelationalDiagrams.com