Algebra, Logic, Geometry: at the Foundation of CS

Tony Hoare
Honorary Professor at Griffith University

ICFEM

13 November 2018

Theses

- Foundations of the Theory of Programming can be taught as an aid to practical programming throughout a degree course in Computing Science.
- A Program Development Environment for teaching should provide features similar to those of modern industrial tool chains.
- The level of Math required in the first practical course is that of High School courses in Algebra, Logic, and Geometry.

Summary

1. Review of Boolean Algebra
2. Deductive Logic
3. Spatio-temporal Logic
4. Sequential Composition
5. Concurrent Composition
6. Unifying Theories of Programming

1.Review of Boolean Algebra

relevant also for Mathematics, and Philosophy

and in CS for Hardware Design, and Program Development.

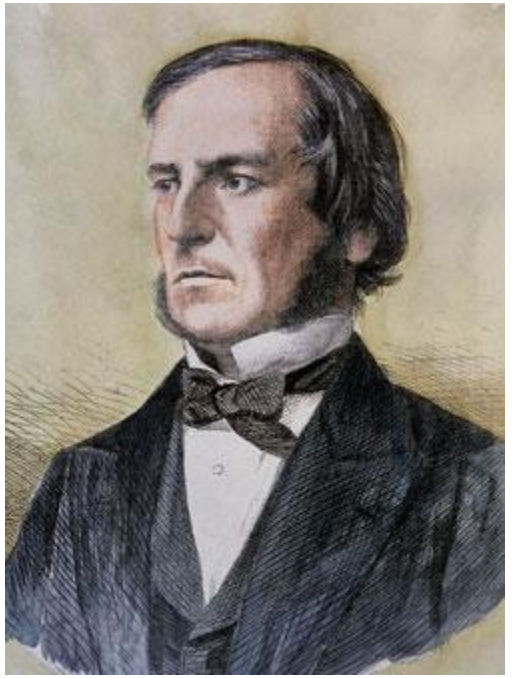
George Boole (1815-1864)

- Professor of Mathematics at Queen's College Cork, Ireland
- Book: 1854 An Investigation of the Laws of Thought

proposed the binary algebraic operators not, and, or,

and a binary comparison for predicates: ≤ (implies).

These are the foundation for a deductive logic of propositions $(p, q, r, \ldots)$



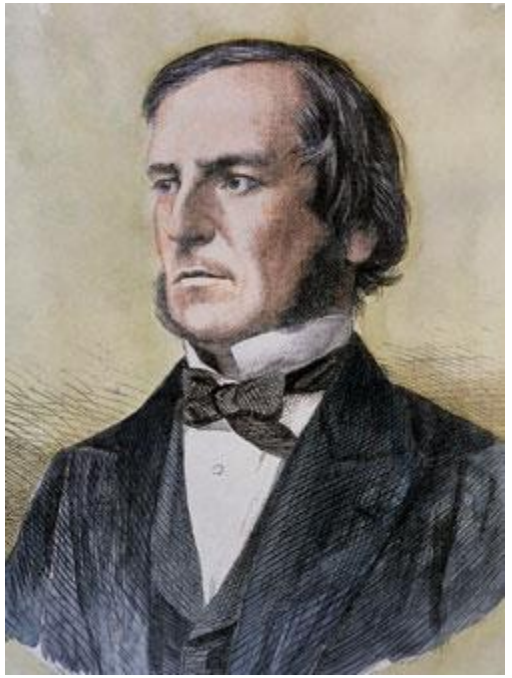
George Boole (1815-1864)

- Professor of Mathematics at Queen's College Cork, Ireland
- Book: 1854 An Investigation of the Laws of Thought

proposed the binary algebraic operators not, and, or,

and a binary comparison predicate: ≤ (implication)

as the foundation of a deductive logic of propositions $(p, q, r, \ldots)$



Disjunction: ∨ (‘or’)

- Axioms: ∨ is associative, commutative and idempotent
- Theorem: ∨ distributes leftward through ∨

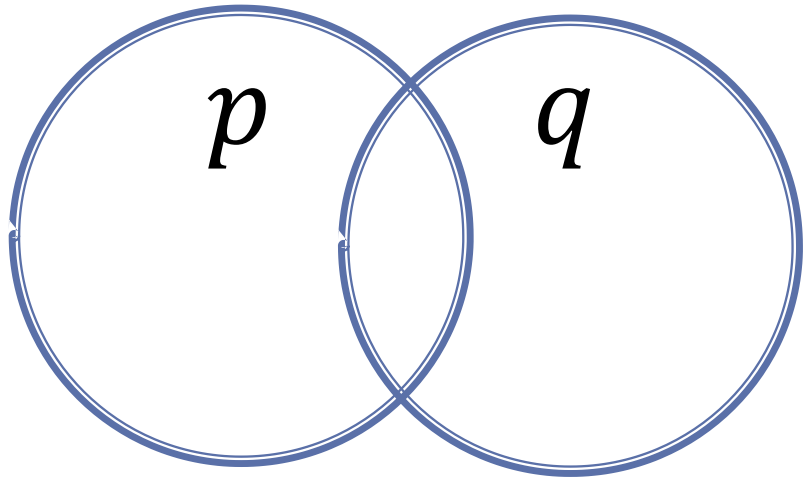
$$(p \vee q) \vee r = (p \vee r) \vee (q \vee r)$$

- Proof:

$$\begin{aligned} rhs &= p \vee \big(r \vee (q \vee r)\big) && \text{assoc} \\ &= p \vee \big((q \vee r) \vee r\big) && \text{comm} \\ &= p \vee \big(q \vee (r \vee r)\big) && \text{assoc} \\ &= p \vee (q \vee r) && \text{idem} \\ &= lhs && \text{assoc} \end{aligned}$$

Corollary: Rightward distribution (follows by comm)

Geometry: $(p \vee q)$

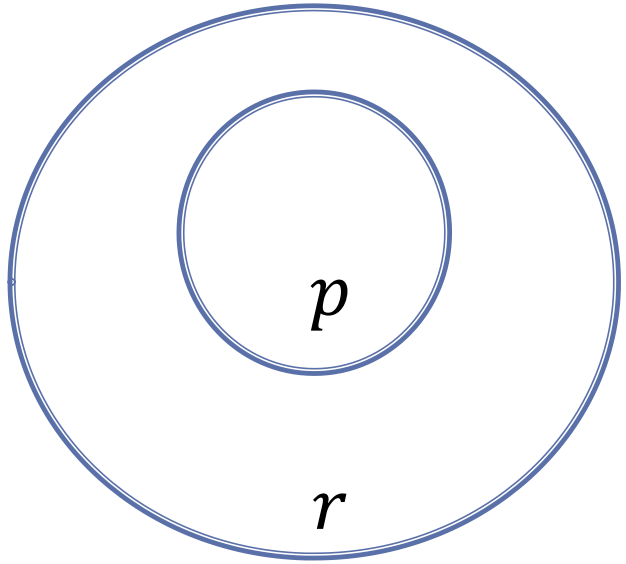


Venn diagram

Comparison: $p \leq r$

- Define $p \leq r$ as $r = p \vee r$
- p implies r , p is stronger than r, r is weaker than p

- Geometry:



$\vee$ is a weakening operator

- Theorem: $p \leq p \vee r$
- Proof: $p \vee r = (p \vee p) \vee r$ by idempotence
  $= p \vee (p \vee r)$ by association

The theorem follows by definition of $\leq$

- Corollary: $p \leq r \vee p$ by commutation

Henceforth we omit brackets around associative operators, and proofs of theorems that follow by commutation.

2.Deductive Logic

Relevant for all branches of mathematics, science and engineering

The Aristotelian Syllogism

$$\frac{All\ men\ are\ animals \qquad All\ animals\ are\ mortal}{All\ men\ are\ mortal}$$

- If the two antecedents above the line have been proved
  the consequent below the line is also provable
- To use a proof rule: first prove the antecedents
  and thereafter assume the consequent whenever required
- To validate a proof rule: first assume the antecedents
  and then use algebra prove the consequent

Aristotle 384-322 BC.

Founded the Lyceum in Athens, and lectured on

sciences:	physics, **biology**, zoology;
aesthetics:	poetry, theatre, music;
ethics:	politics, government, rhetoric;
philosophy:	metaphysics, **logic**, linguistics.

Recognised as the originator of **logic** and of classificatory **biology**, in which syllogisms are suited for deducing consequences from its classifications.

Rule of Proof by Cases

$$\frac{p \leq r \qquad\qquad q \leq r}{(p \vee q) \leq r}$$

Validation: Assume the antecedents: $r = p \vee r$ and $r = q \vee r$

$r = r \vee r$	the idempotence axiom
$= (p \vee r) \vee (q \vee r)$	by substitution for each r
$= (p \vee q) \vee r$	by distribution of r through $\vee$

The conclusion follows by the definition of $\leq$

Ordering: ≤

- Theorem: ≤ is a partial order

reflexive: $\dfrac{}{p \leq p}$ by idempotence

transitive: $\dfrac{p \leq q \qquad q \leq r}{p \leq r}$ by association

antisymmetric: $\dfrac{p \leq q \qquad q \leq p}{p = q}$ by commutation

Covariance (monotonicity) of ∨

Theorem: $\dfrac{p \leq q}{p \vee r \leq q \vee r}$ also $\dfrac{p \leq q}{r \vee p \leq r \vee q}$

Proof : $p \leq q \leq q \vee r$ and $r \leq q \vee r$

The result follows by the proof rule for cases

A covariant operator preserves the ordering of each of its operands

‘strengthening a part can only strengthen the whole product’

(and weakening similarly)

3.Spatio-temporal Logic

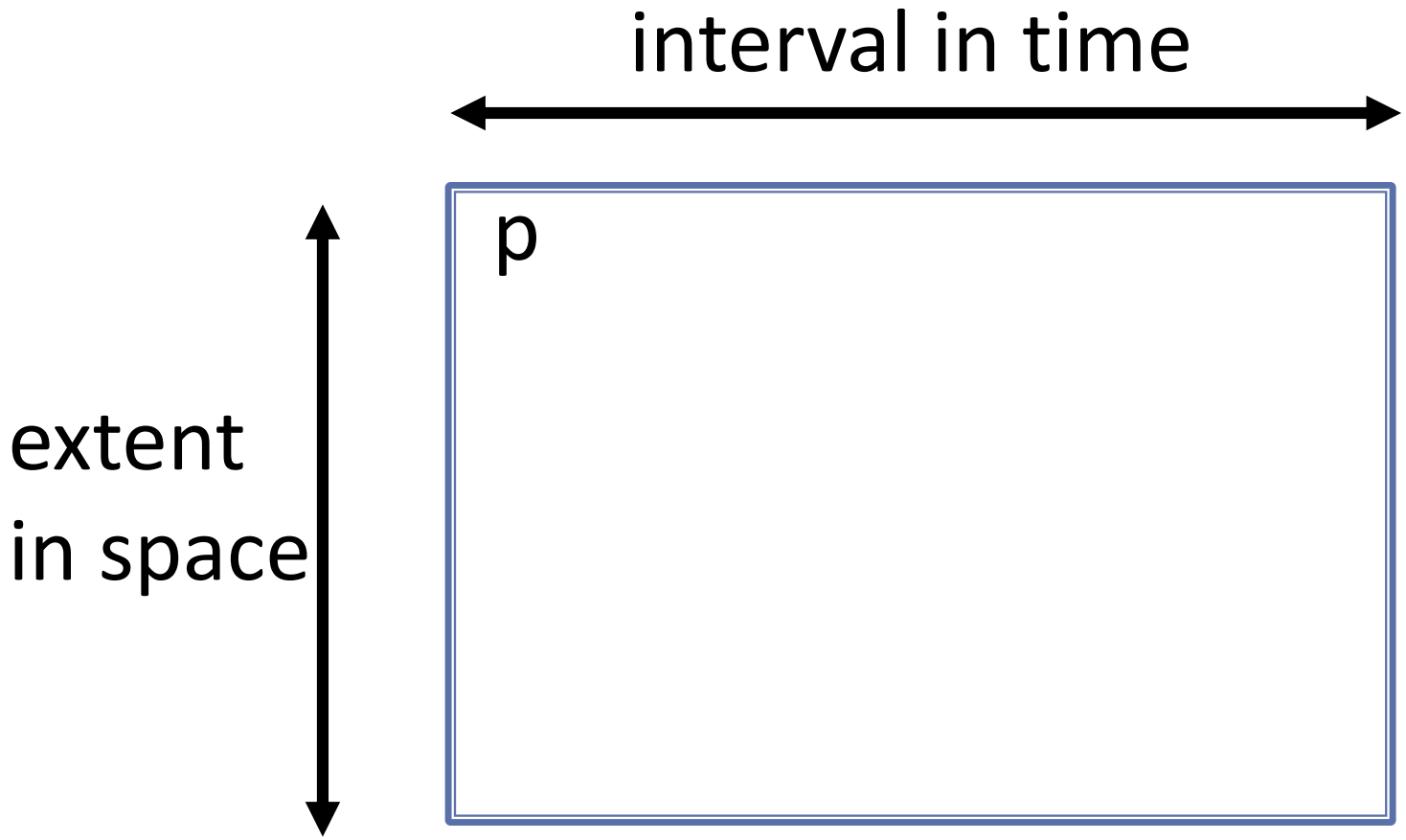
for non-metric reasoning about what happens in space and in time

William of Occam (1287-1347)

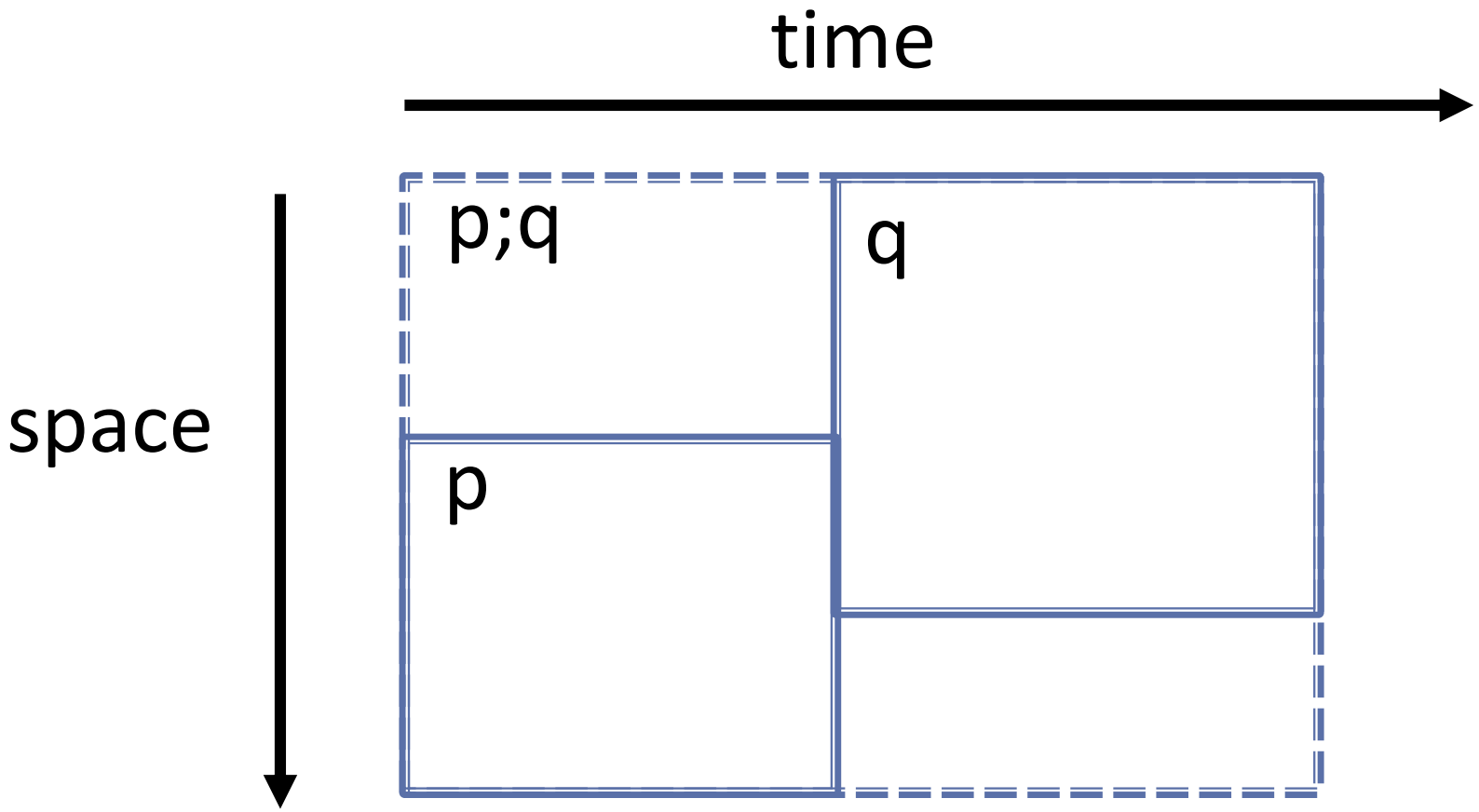
- Franciscan friar, Scholar at Merton College Oxford
- excommunicated (1328) rehabilitated (1359)
- Occam's razor: entities should not be postulated beyond necessity
- Book: Summa Logicae (1323)

with operators for implication, disjunction, conjunction, causation; and temporal operators while (|), and then (;)

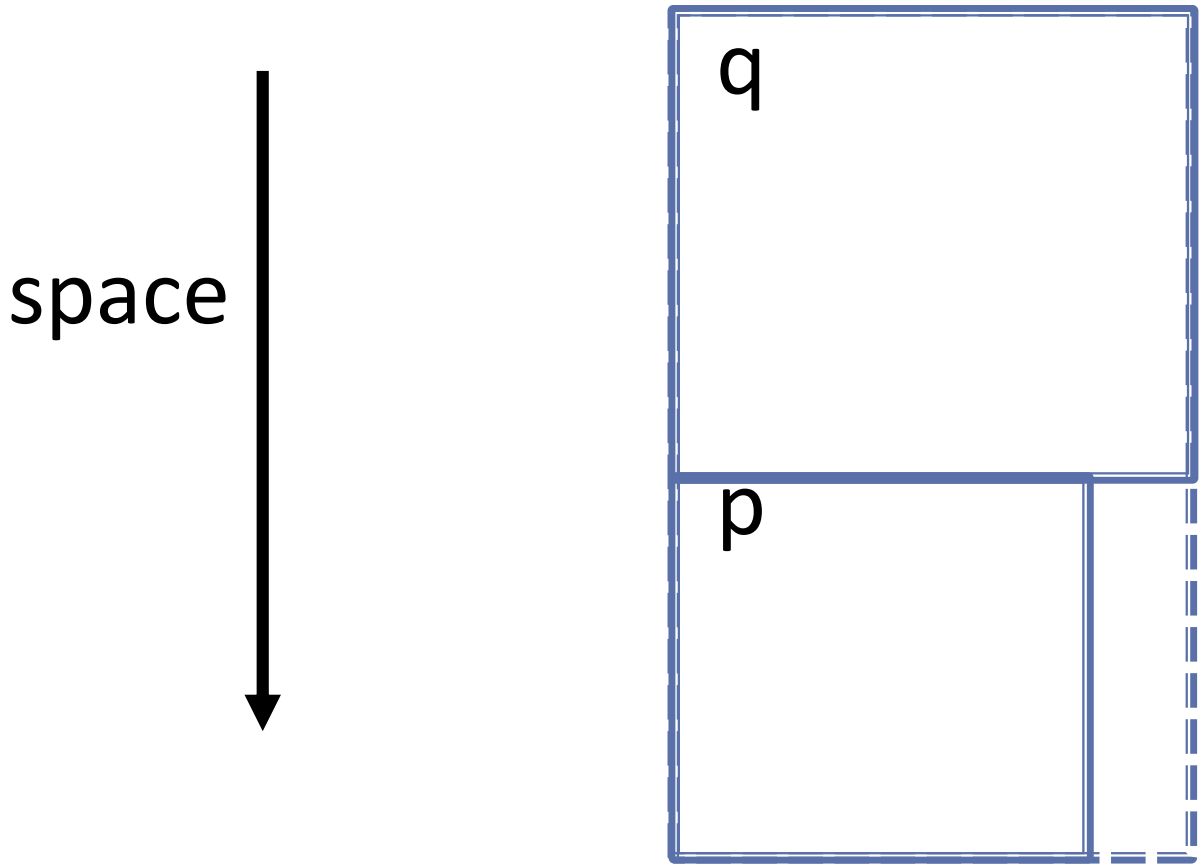
Geometry



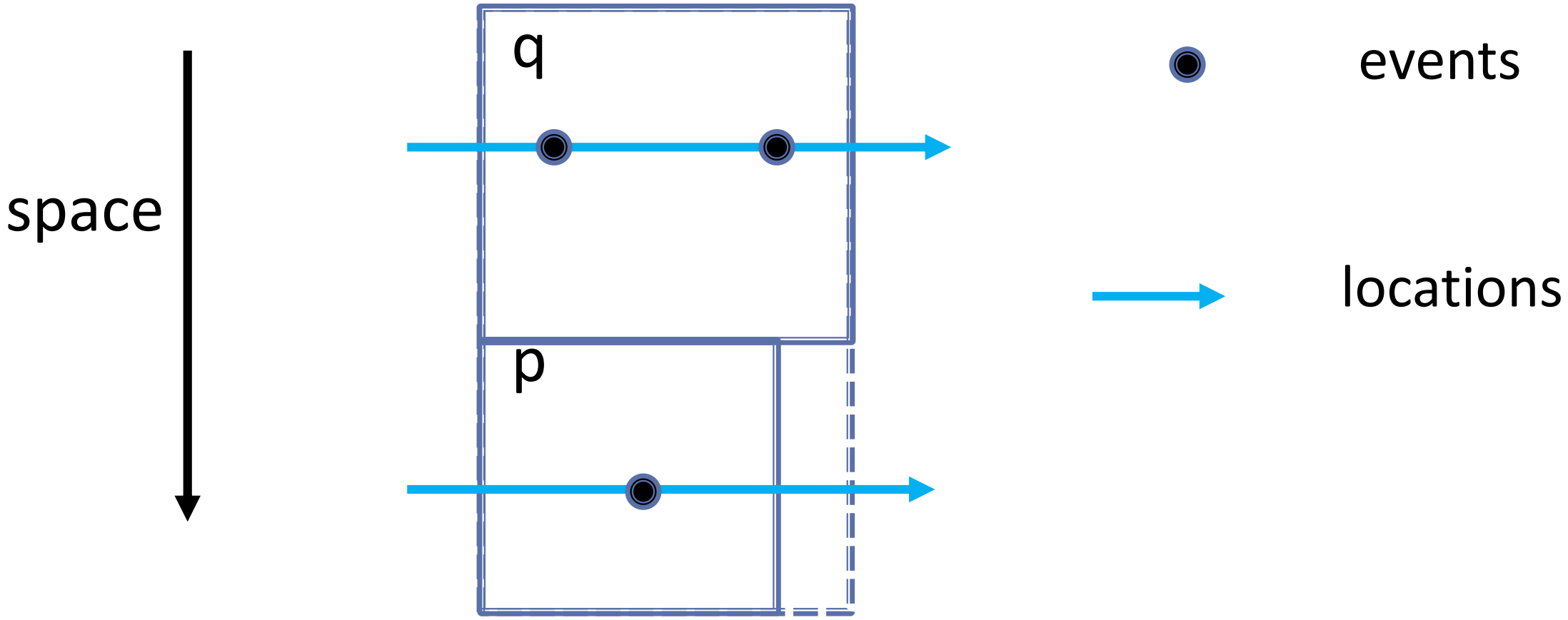
Geometry of p;q



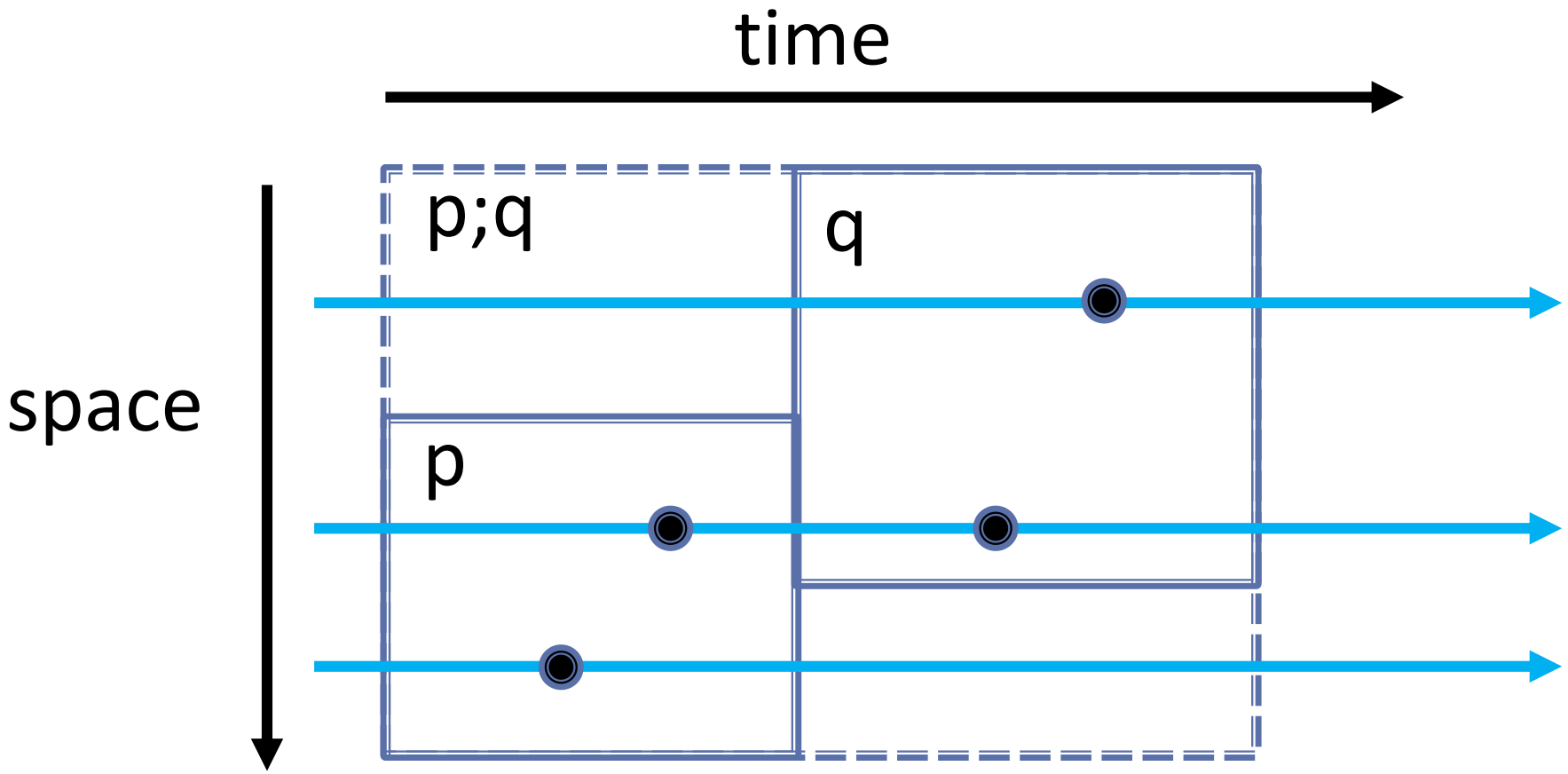
Geometry of (p|q)



Geometry of (p|q)



Geometry of p;q



The small print

- A term is defined only if all its operands are defined
- $(p \vee q)$ and $p \leq q$ are defined only if p and q have the same region
- $(p; q)$ and $(p|q)$ are defined only if their regions are disjoint.
- events$(p; q)$ = events$(p|q)$ = events(p) u**nion** events(q)
- and the same for intervals and extents.

4. Sequential Composition

The algebraic axioms for sequential composition validate the relevant proof rules

Algebraic Axioms for ;

- [ ] describes a region in which nothing happens (aka null, skip)
  - padding can have any extent or duration, consistent with its context
- ; is associative and has unit [ ]
- ; distributes through ∨ (both leftward and rightward):

  e.g., $p;(q \vee q') = (p;q \vee p;q')$

Distribution justifies giving ; a precedence stronger than ∨

Proof by cases

- Theorem:

$$\frac{p;q \leq r \qquad\qquad p';q \leq r}{(p \vee p');q \leq r}$$

- Assume (1) $r = p;q \vee r$ and (2) $r = p';q \vee r$

Therefore $r = p;q \vee p';q \vee r$ substitute (2) in (1)

$= (p \vee p');q \vee r$; distributes thru $\vee$

= consequent of the rule (by definition of $\leq$)

Proof rule for sequential composition

- Theorem:

$$\frac{p;q \leq m \qquad\qquad m;r \leq t}{p;q;r \leq t}$$

- Assume (1) $m = p;q \vee m$ and (2) $t = m;r \vee t$

Therefore $t = (p;q \vee m);r \vee t$ substitute (1) in (2)

$= p;q;r \vee (m;r \vee t)$ distribute r thru $\vee$

$= p;q;r \vee t$ substitute back by (2)

Rules of consequence

The theorem

$$\frac{p;q \leq m \qquad m;r \leq t}{p;q;r \leq t}$$

has corollaries

$$\frac{p \leq m \qquad m;r \leq t}{p;\ r \leq t}$$

$$\frac{p;q \leq m \qquad m \leq t}{p;q \leq t}$$

Proof: substitution of [ ] for q in the first, and for r in the other

Hoare triple

Consider the proposition $p;q \leq r$.

p describes the interval from the start of r to the start of q ,
and q describes the interval from the end of p to the end of r,

then r correctly describes the whole of $(p;q)$

Define $\{p\}\ q\ \{r\}$ as $p;q \leq r$

Verification Rules for ;

- By substitution of the definition of the triple into the Proof Rule for ;

$$\frac{\{p\}\ q\ \{m\} \qquad \{m\}\ r\ \{t\}}{\{p\}\ q;r\ \{t\}}$$

which is the Hoare rule for ;

- The two corollaries give:

$$\frac{\{p\}\ q\ \{m\} \qquad m \leq t}{\{p\}\ q\ \{t\}} \qquad \frac{p \leq m \qquad \{m\}\ r\ \{t\}}{\{p\}\ r\ \{t\}}$$

which are the Hoare rules of Consequence

In Praise of Algebra

- Simple, elegant, reusable, tractable by people and by machines,
- Algebraic transformation is essential in the top-down design of application system architecture by successive refinement
- They are also used in compilation, optimisation, refactorization, obfuscation, and automatic generation of program code
- Algebra unifies theories which underlie a range of programming tools, It is clearly essential for their correct interworking
- and for the introduction of Theory into Computer Science education

Milner transition $r \xrightarrow{p} q$

- One way of executing r is to execute p first, saving q as a continuation for subsequent execution

- Define $r \xrightarrow{p} q$ as $p;q \leq r$

Operational rules for ;

$$\frac{r \xrightarrow{p} m \qquad m \xrightarrow{q} t}{r \xrightarrow{p;q} t}$$

the Milner rule for ;

- The two corollaries are

$$\frac{m \leq r \qquad m \xrightarrow{q} t}{r \xrightarrow{q} t} \qquad \frac{r \xrightarrow{p} m \qquad t \leq m}{r \xrightarrow{p} t}$$

i.e., Milner's 'rules of structural equivalence' , with ≡ replaced by ≤

5. Concurrent Composition

| has the same laws as ; . An additional Interchange axiom permits a concurrent program to be executed sequentially by interleaving.

Algebraic Axioms for |

- | is associative with unit [ ]
- | distributes through ∨
- $(p \mid q);(p' \mid q') \leq (p;p') \mid (q;q')$ (the interchange axiom)
  - The rhs and the lhs differ by interchange of operators ; with | ,
  - and of operands p' with q

Theorems: $p;q' \leq p \mid q'$ by interchange, with $p' = q = []$

$q;p' \leq p' \mid q$ similarly, with $q' = p = []$

Hence $p;q \vee q;p \leq p \mid q$ by the rule for cases

$$(p \mid q);(p' \mid q') \leq (p;p') \mid (q;q')$$

Theorems

$$(p \mid q);q' \leq p \mid (q;q') \qquad p' = []$$
$$p;(p' \mid q') \leq (p;p') \mid q' \qquad q = []$$
$$q;(p' \mid q') \leq p' \mid (q;q') \qquad p = []$$
$$(p \mid q);p' \leq (p;p') \mid q \qquad q' = []$$

All four are proved by substitution of [ ]

They are known as small interchange laws (or frame laws in separation logic)

Hence $p; q; q' \leq (p \mid q);q' \leq p \mid (q;q')$

Interleaving longer strings

- Let x,y,z,w,a,b,c,d be characters representing single events
- Let us omit ; in strings except for emphasis. Thus

  xyzw = x;y;z;w

Example of Interleaving

	$abcd \mid xyzw$	is the rhs of interchange
$\geq$	$(a;bcd) \mid (xy;zw)$	associativity (twice)
$\geq$	$(a \mid xy);(bcd \mid zw)$	interchange
$\geq$	$(a \mid x;y);(b;cd \mid zw)$	associativity (twice)
$\geq$	$(a \mid x);y;(b \mid zw);cd$	small interchange (twice)
	.....	.....
$\geq$	$xayzbwcd$	similarly

Each step of the proof reduces length of same-coloured strings.
Termination is assured when this is no longer possible.

Basic Principle of Concurrent Programming

- Every interleaving which preserves the order of the operands of all sequential and of all concurrent compositions is reachable by strengthening applications of the interchange axiom.
- first proved for Turing machines by simulation (interpretation)
- a direct algebraic proof (omitted) uses structural induction.

6.Unifying Theories of Concurrency

We repeat for concurrent programs the same unification achieved before for sequential programming.

Interchange Rule (O'Hearn)

$$\frac{p;q \ \leq \ r \qquad\qquad p';q' \ \leq \ r'}{(p \mid p');(q \mid q') \leq (r \mid r')}$$

The rule tells how to prove a complicated concurrent theorem by splitting it into two proofs of two much simpler sequential theorems.

Theorem: This rule is equivalent to the Interchange axiom

Proof: next two slides

The rule implies axiom

$$\frac{p;q \leq r \qquad\qquad p';q' \leq r'}{(p \mid p');(q \mid q') \leq (r \mid r')}$$ (concurrency rule)

Proof: Replace r by $p;q$ and r' by $p';q'$ throughout

The antecedents are true by the reflexivity of $\leq$

and the conclusion is:

$$(p \mid p');(q \mid q') \leq (p;q) \mid (p';q')$$

which is the interchange law

The axiom implies the rule

Assume the antecedents of the rule: $p;q \leq r'$ and $p';q' \leq r'$

$$(p;q) \mid (p';q') \leq (r \mid r') \qquad \text{(covariance of } \mid \text{ twice)}$$

$$(p \mid p');(q \mid q') \leq (p;q) \mid (p';q') \qquad \text{(interchange axiom)}$$

So $(p \mid p');(q \mid q') \leq (r \mid r')$ (by transitivity of $\leq$)

Therefore $$\frac{p;q \leq r \qquad p';q' \leq r'}{(p \mid p');(q \mid q') \leq (r \mid r')}$$ (the interchange rule)

Proof rule for Concurrent Composition

- $\dfrac{p;q \leq r \qquad p';q' \leq r'}{(p \mid p');(q \mid q') \leq (r \mid r')}$ copied from previous slide
- $\dfrac{\{p\}\ q\ \{r\} \qquad \{p'\}\ q'\ \{r'\}}{\{p \mid\mid p'\}\ q \mid\mid q'\ \{r \mid\mid r'\}}$ translated to Hoare triples
- $\dfrac{r \xrightarrow{q} p \qquad r' \xrightarrow{q'} p'}{(r \mid\mid r') \xrightarrow{(q \mid\mid q')} (p \mid\mid p')}$ translated to Milner transitions

are all equivalent to the exchange law

Applications to Programming

- Most interpreters and compilers for programming languages follow an operational semantics expressed as Milner Transitions.
- Most program analysers and proof tools for sequential languages follow a verification semantics expressed as Hoare Triples.
- Many papers in the Theory of Programming prove the consistency between these two ‘rival’ theories for particular languages
- Algebra unifies the theories, by proofs which could be understood or even discovered under guidance by CS students in their practical programming courses.

Acknowledgements to my Colleagues

- at Oxford University: Jean-Raymond Abrial, Richard Bird, Joe Stoy, Dana Scott, Bill Roscoe, Steve Brookes, Carroll Morgan
- at Microsoft Research: Jade Alglave, Ernie Cohen, Byron Cook, Rustan Leino, Matt Parkinson, Jon Pincus
- at Cambridge and Grifffith University: Nada Amin, Zhe Hou
- my recent coauthors: Bernhard Moeller, Peter O'Hearn, Stephan van Staden, Georg Struth, Ian Wehrman, John Wickerson
- my heroes: Bertrand Russell, Edsger W. Dijkstra, Robin Milner
- Jill Hoare and Jonathan Lawrence

Further reading

- Consult my website www.cl.cam.ac.uk/~carh4/

- Lecture1. Geometric theory of program testing
- Lecture 2. Algebra for program transformation
- Lecture 3 is an early version of today's lecture

Limitations of Algebra

- It has insufficient expressive power: no quantification.
- Logic can also specify and verify interfaces between components of a program.
- It cannot specify basic commands
- Logic specifies basic commands (assignment, input, output, ...)
- It has no negation: it cannot prove that a formula is not a theorem
- Geometry is a model of both algebra and logic. It provides test cases for incorrect programs and counterexamples for false conjectures

- see https://www.cl.cam.ac.uk/~carh4/
  www.Heidelberg-laureate-forum.org/?s=2016

Isaac Newton (1642-1726)

Communication with Richard Gregory (1694)

"Our [my] specious [falsely convincing] algebra [the infinitesimal calculus] is fit enough to find out [has some heuristic value], but entirely unfit to consign to writing and commit to posterity [it cannot and must not be published]."

(with translation to Modern English)

Newton's proofs were geometric, establishing properties of the Keplerian ellipses that describe the orbits of the planets

Bertrand Russell (1872 – 1970)

The method of “postulating” what we want has many advantages; they are the same as the advantages of theft over honest toil. Let us leave them to others and proceed with our honest toil.

Introduction to Mathematical Philosophy.

Russell then refused to postulate the existence of real numbers (such as the sqrt of 2), and proceeded to model them by the Dedekind cut.

Gottfried Leibniz (1646-1716)

- calculemus Let us calculate (symbolically)