

Highly-Parallel Systems, New Math, and New Tools

Richard Shoup

Boundary Institute
San Jose, California, USA

rshoup@boundary.org

R. Shoup – Oct '11

Copyright © 2011 - All Rights Reserved

Boundary Institute

Overview

- Computer engineering hasn't changed much in 50 years...
 - What has and hasn't changed, and why
- What's the Problem? Two particular challenges.
- New math foundations and tools
 - Boundaries, Complex Logic, Relations, Structure
- Relevance, short- and long-term
 - Building complex parallel systems, reliable & secure
 - Searching for knowledge, not just concepts

R. Shoup – Oct '11

Boundary Institute

-- Computing--



Tower of Babel - Bruegel

"Computing's core challenge is how not to make a mess of it. ... machines are so fast and storage capacities are so huge that we face orders of magnitude more room for confusion, the propagation and diffusion of which are easily inadvertently mechanized."
-- Edsger Dijkstra, 1997

R. Shoup – Oct '11

Boundary Institute

Computing - 50 years

- Digital technology = $F(\lambda, \dots)$
- Not much change except size/speed/cost
 - Exceptions: user interface, networking, apps
- Architecture: von Neumann (& Eckert)
- Theory: logic, set theory, Turing model
- Programming model: algorithms
 - "A step-by-step problem-solving procedure"
 - Sequential processors, instructions
 - Unfolding/mapping computation in Time



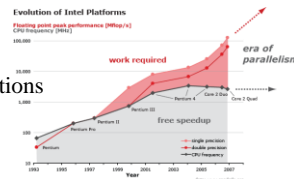
Muhammad al-Khwarizmi
– 780-850 AD
al-Khwarizmi ⇒ "algorithm"
Hisab al-jabr ⇒ "algebra"

R. Shoup – Oct '11

Boundary Institute

50 Years of Computing - What HAS Changed

- Faster, cheaper, smaller !!
 - But nearing practical limitations, $\text{Power} \propto \text{Freq}^3$
 - What are we going to do with N billion transistors?
- Better user interface (2D)
- Communications, networks
- Much broader range of applications
- ...



R. Shoup – Oct '11

Boundary Institute

50 Years of Computing - What Has NOT Changed

- Von Neumann model: stored program, serial execution
- Centralized memory & processing, buses
- Synchronous, global clocking
- Awful design tools, bad bugs
- Linear text-based programming
- Compilers, debuggers, crashes
- Chip & protocol chasing
- Little or no formal verification, security



*Kids running down a hill,
out of control...*

R. Shoup – Oct '11

Boundary Institute

50 Years vs 5B Years

Computing Systems

serial, sequential
digital, discrete
unidirectional, functional
singular, brittle
centralized, concentrated
linear, controlled
homogeneous
mechanical, intermittent
difficult, arcane
designed, crafted

Natural Systems

highly parallel, concurrent
analog, continuous
omnidirectional, relational
redundant, resilient
distributed, pervasive
non-linear, chaotic
heterogeneous
alive, continuous
accessible, familiar
self-organizing, evolving

Which is a better match to human beings and their needs?

R. Shoup – Oct '11

Boundary Institute

What's the Problem?

- Huge Complexity – datasets, processors, networks, etc.
- Parallelism – networks/clouds, multicore, FPGAs, etc.
 - Moore's Law limitations
 - Serial mindset, algorithms
- Power consumption & Efficiency
 - Global synchronous clocking, architecture
- Error tolerance & recovery
 - Perfection vs reality of failures
- Verification, proof of correctness
- Guarantee of security, privacy

*BUT... Free Ride from
semiconductor technology!*

R. Shoup – Oct '11

Boundary Institute

Heterogeneous Data Hierarchy

- Cluster, domain, etc
- Computer
- Disk, partition
- Folder
- File
- Byte, word
- Bit

} different types & storage

... but object-oriented languages & systems mitigate

R. Shoup – Oct '11

Boundary Institute

Heterogeneous Design Hierarchy

user interface, application	⇒	<i>menus, dialogs, commands</i>
very-high-level language	⇒	<i>threads, app objects, users</i>
high-level language	⇒	<i>processes, threads</i>
system libraries, OS calls	⇒	<i>procedures, synchronization</i>
low-level language (C)	⇒	<i>data structures, subroutines</i>
assembly/machine code	⇒	<i>storage allocation</i>
microcode, FSMs	⇒	<i>register allocation</i>
state flip-flops, registers	⇒	<i>timing, power</i>
transistors, gates	⇒	<i>layout, timing</i>
silicon structures	⇒	<i>electrical properties, yield</i>
semiconductor physics	⇒	<i>processes, chemistry, optics</i>

R. Shoup – Oct '11

Boundary Institute

Things I would not have believed in 1970

- Reconfigurable hardware (FPGAs) won't be commercialized for 15 years, won't become popular for 25, and isn't fully exploited 40 yrs later.
- In 40 years, most design will still be done informally, without math grounding, flying by seat of our pants, in both hardware & software.
- Computing will still be serial by default, parallel a special case.
- CPUs will be built on nmeter elements, run at many ghz, yet most will use essentially the same architecture and programming model.
- Everything will still run as "applications".
- Turn-on and boot will still take a long time.
- Industry will deploy, and consumers will accept, operating systems and applications with significant known bugs.
- Industry and academia will deploy networking on a global scale with little or no security ... and leave it that way for many years.
- The word "crash" will still be commonly used.

R. Shoup – Oct '11

Boundary Institute

Progress - CS?

"Our basic ideas about what a computer is, what it does, and how it does it, have hardly changed for decades."

-- Paul Dourish, Where the Action Is, 2001

"Of all undergraduate CS subjects, theoretical computer science has changed the least over the decades."

-- SIGACT News, March 2004

"Wegner's Conjecture: Interaction is more powerful than algorithms."

-- Peter Wegner, CACM'97

R. Shoup – Oct '11

Boundary Institute

Progress - Languages & Design?

"Modern programming languages are highly complex and are standardised by committees, ... the resulting languages will inevitably contain compromise, inconsistency, incompleteness and some bits that simply do not work as expected. ... failures occur with monotonous regularity and with a similar frequency to 20 years ago."

-- Les Hatton, IT Week, April 2006

"The introduction of the microprocessor in 1971 marked the beginning of a 30-year stall in design methods for electronic systems."

-- Nick Tredennick

"If a system has an extremely powerful debugger, it may be because the system needs an extremely powerful debugger." -- anonymous

R. Shoup – Oct '11

Boundary Institute

Progress - Software?

"Most software today is very much like an Egyptian pyramid with millions of bricks piled on top of each other, with no structural integrity, but just done by brute force and thousands of slaves." -- Alan Kay

"The most amazing achievement of the computer software industry is its continuing cancellation of the steady and staggering gains made by the computer hardware industry." -- Henry Petroski

"It has been said that the great scientific disciplines are examples of giants standing on the shoulders of other giants. It has also been said that the software industry is an example of midgets standing on the toes of other midgets." -- Alan Cooper

R. Shoup – Oct '11

Boundary Institute

Progress – AI?

"The question [Can machines think?] ... I believe to be too meaningless to deserve discussion. Nevertheless, I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."

-- Alan Turing, 1950

R. Shoup – Oct '11

Boundary Institute

Progress - Security & Privacy?

"Microsoft fixed 23 vulnerabilities across eight security bulletins as part of its October Patch Tuesday release. October's Patch Tuesday release resolved issues in Internet Explorer versions 6 through 9, all versions of Microsoft Windows from XP through 7, .NET and Silverlight, Microsoft Forefront Unified Access Gateway and Host Integration Server, Microsoft said Oct. 11. Two of the patches are rated "critical," and six are rated "important," Microsoft said."

-- eWeek, Oct 11, 2011

R. Shoup – Oct '11

Boundary Institute

Reconfigurable Computing - Challenging

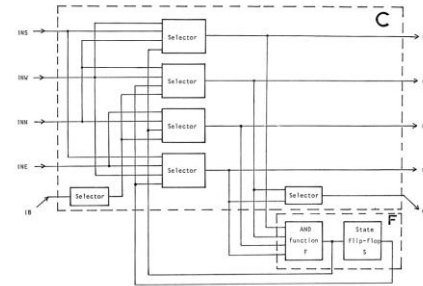
- Need for more computation (10x - 1000x - 1Mx)
 - Simulation, vision, AI, data mining, etc, etc
(or just more efficient use of existing hardware)
- ⇒ Parallelism - custom pipelines/structures
 - Unfold computation in *space* (data parallel), or *time* (pipeline parallel)
- ⇒ Hardware is more specific (lower generality)
- ⇒ *Reconfigurable* parallel hardware

... *But our programming paradigm and nearly all our tools are still conceptualized for fixed, sequential hardware.*

R. Shoup – Oct '11

Boundary Institute

Early FPGA - 1969

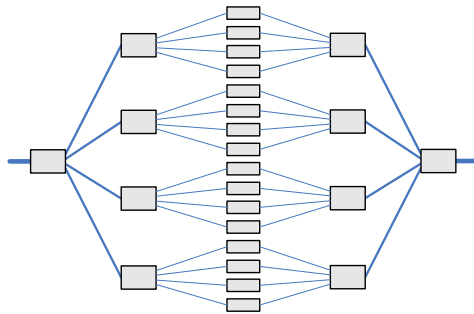


Richard Shoup, CMU – “Programmable Cellular Logic Arrays”, 1970

R. Shoup – Oct '11

Boundary Institute

Parallelizing - Easy

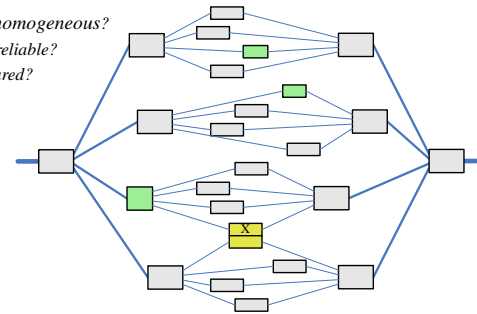


R. Shoup – Oct '11

Boundary Institute

Parallelizing - Challenging

*inhomogeneous?
unreliable?
shared?*



R. Shoup – Oct '11

Boundary Institute

Proposal: New Tools, New Math

- Single formal representation/language
 - Abstract definition of computation, not hardware or software
 - Declarative, relational, self-similar hierarchical, self-referring
- Deep primitives, mathematical foundations
 - Beneath/prior to Logic, Set Theory, algorithms
 - Pure structure, simple primitives, unambiguous meaning
- Formal from the beginning, and throughout
 - Initial specification, design, optimization, verification, debugging, field configuration, modification, upgrading, etc.
 - Operation provably correct by construction, verifiable
 - Security, privacy provable and verifiable *in situ*

R. Shoup – Oct '11

Boundary Institute

New Tools Needed

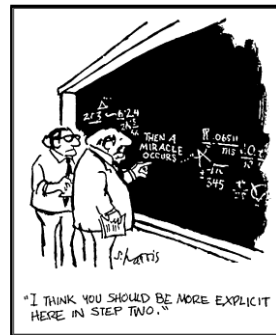
- Formally capture specifications from the beginning
- Formally add detail, transform, and extend design
- Allow exploration of implementations, design spaces, trade-offs
 - Parallel/serial simulation, emulation, FPGA, ASIC, full custom, SoC
- Provide interactive visual/graphic and textual representations
- Make complex designs visible, understandable, manipulable
- Provide formal verification, proofs, queries, analysis, security
- Provide a path to other tools and methods

R. Shoup – Oct '11

Boundary Institute

-- New Math --

- Boundary Math
 - Logic with one symbol
 - Hierarchical structure
- Complex Logic
 - Imaginary logic values
- Link Theory
 - Composite relations
- Constraint Graphs
 - Sequential inference



R. Shoup – Oct '11

Boundary Institute

“Basic” ZFC Set Theory

- $\forall x \forall y [(x = y) \leftrightarrow \forall z (z \in x \leftrightarrow z \in y)]$ (*extensionality*)
- $\forall x \forall y \exists z (z = \{x, y\})$ (*pairing*)
- $\forall z \exists y (z \in y \leftrightarrow z \in x \wedge \phi(z))$ (*comprehension schema*)
- $\exists x (\{ \} \in x \wedge \forall y \in x (S(y) \in x))$ (*infinity*)
- $\forall x \exists y (z \in y \leftrightarrow z \subset x)$ (*power set*)
- $\exists x (x \in a \wedge x \cap a = \{ \})$ (*foundation*)
- $\forall w \exists s (s = \{y: \exists x \in w \wedge \phi(x, y)\})$ (*replacement schema*)
- $\forall X \exists f (\forall x \in X - \{ \} (f(x) \in x))$ (*choice*)

“Later generations will regard set theory as a disease from which one has recovered.” -- Henri Poincaré

R. Shoup – Oct '11

Boundary Institute

Boundary Math: Logic Bedrock

~~F~~ ~~T~~ ~~∧~~ ~~∨~~ ~~∅~~ ()

⇓

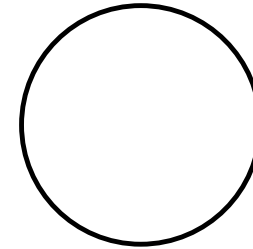
()

All of propositional logic with a single symbol/concept

(*Laws of Form* – George Spencer-Brown, 1969)

R. Shoup – Oct '11

Boundary Institute



R. Shoup – Oct '11

Boundary Institute

The Void - Nothing

- No-Thing, prior to any object or distinction
- Not a material or mathematical object
 - Not the vacuum, the empty set, zero, etc
- No properties or attributes
 - Not emptiness, no measure or dimensionality
- No location or place
 - Not here or there, yet everywhere
- Unique, shared by all, in common
 - The same for all viewpoints

R. Shoup – Oct '11

Boundary Institute

The Void - Nothing (con't)

- No consequences or effects
 - Yet is always dominated, available, free
- Cannot be combined with anything
 - Yet is always being combined, everywhere
- Cannot be named, referenced, pointed to
 - What we indicate/assert by not speaking
- “The Tao that is named is not the Eternal Tao.”

“God made everything out of nothing. But the nothingness shows through.”
-- Paul Valéry

R. Shoup – Oct '11

Boundary Institute

Distinction – Simplest Idea

- First Thing -- *Simplicity Itself*
- Pure difference, discrimination
- Pure form, no interpretation
- Smallest possible, unavoidable step from the Void
 - Not-Void, not-nothing, but something/everything
- First *object* (thing) and *action* (indication/cross)
 - Unique, prior to all else, bedrock, no "axioms"
- "It from Bit" - Wheeler



"Behind it all is surely an idea so simple, so beautiful, that when we grasp it - in a decade, a century, or a millennium - we will all say to each other, how could it have been otherwise? How could we have been so stupid for so long?"

-- John Archibald Wheeler

R. Shoup – Oct '11

Boundary Institute

About Distinctions

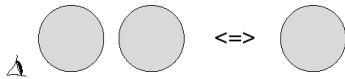
- Every D has two sides
 - Nothing exists without its complement
- Every D is made in a space (larger D)
 - Problem with naïve Set Theory
- *Difference is prior to Object*
- Always a point of view, a position
 - "Everything is relative", seriously
- cf Leibniz, Parker-Rhodes "Indistinguishables"
- First D = Big Bang?



R. Shoup – Oct '11

Boundary Institute

Two Distinctions



Outside the 1st D - symmetrical, cardinality, bosons



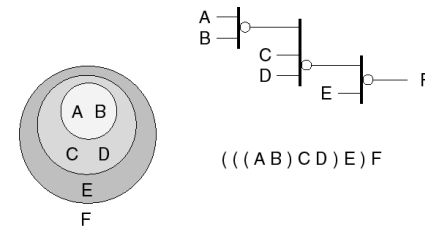
Inside the 1st D - asymmetrical, ordinality, fermions

Laws of Form – G. Spencer-Brown

R. Shoup – Oct '11

Boundary Institute

Boundary Visualizations



$(((A B) C D) E) F$

$\neg(\neg(\neg(A \vee B) \vee C \vee D) \vee E) = F$

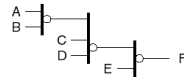
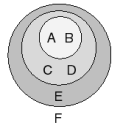
R. Shoup – Oct '11

Boundary Institute

Unifying Boundaries with Logic

- \vee operator vs “in same space”
 - $\neg(\neg(\neg(A \vee B) \vee C \vee D) \vee E) = F$
- Commutative? \otimes
 - no sequence, no linear order
- Associative? \otimes
 - no sequence, no linear order
- Arity? \otimes
 - arbitrary or none
- Relations vs Functions
- New basis for Set Theory

$((AB)CD)EF$



(William Bricken, Bricken Technologies)

R. Shoup – Oct '11

Boundary Institute

BMath: Notation

Boolean algebra Boundary Math

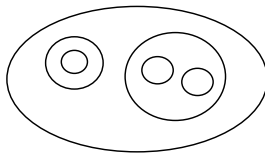
F	
T	()
a	a
$\sim a$	(a)
$a \vee b$	a b
$a \vee b \vee c$	a b c
$a \wedge b$	((a) (b))
$a \Rightarrow b$	(a) b

R. Shoup – Oct '11

Boundary Institute

Boundary Math Example

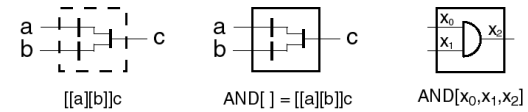
((()) (() ()))	original form
(((() ())))	cross
((()))	call
()	cross



R. Shoup – Oct '11

Boundary Institute

Unifying Boundaries with Objects

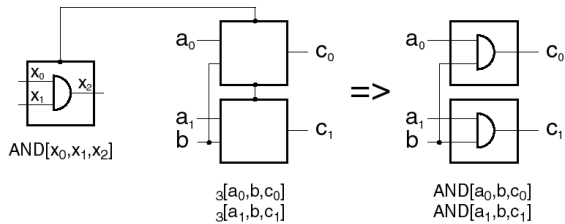


Encapsulation & Abstraction

R. Shoup – Oct '11

Boundary Institute

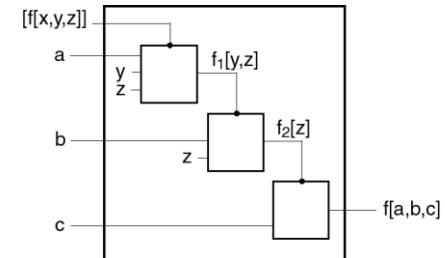
Instantiating & Replication



R. Shoup – Oct '11

Boundary Institute

Lambda Apply (Currying)



Computation is the propagation of change through structure

R. Shoup – Oct '11

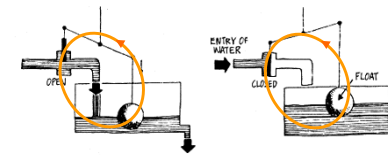
Boundary Institute

-- Imaginary Logic Values --

R. Shoup – Oct '11

Boundary Institute

Self-Reference, Feedback, and Paradox



Negative feedback

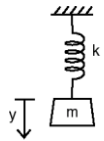
"All Cretans are liars." -- Epimenides of Crete
 "This sentence is False." -- anonymous
 "The set of all sets not containing themselves." -- Bertrand Russell
 "This statement is unprovable." -- Kurt Gödel
 "This program does not halt." -- Alan Turing

R. Shoup – Oct '11

Boundary Institute

Classical Self-Reference (Feedback)

- Mechanical systems
- Control Theory
- Systems Theory
- Cybernetics
- etc ...



$$y = -\frac{m}{k} \frac{d^2 y}{dt^2}$$

$$= C_1 e^{+i\omega t} + C_2 e^{-i\omega t}$$

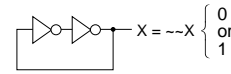
$$= C \sin(\omega t + \phi)$$

Negative feedback is commonly used in engineering,
... *but paradox is forbidden in Logic and Set Theory*

R. Shoup – Oct '11

Boundary Institute

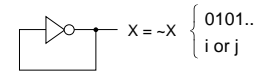
Self-Reference - Where Time Begins



X in _____

X out _____

Autonomy (Memory)



X in

X out

Antinomy/Paradox (Clock)

R. Shoup – Oct '11

Boundary Institute

Imag Numbers vs Imag Logic Values

Numbers

$$x^2 = -1 \quad (x = -1/x)$$

$$x = \pm\sqrt{-1} = i \text{ or } -i$$

1, -1, i, -i •

ok

Logic

$$x = \sim x$$

$$x = i \text{ or } j$$

F, T, i, j ⊗

not ok!

R. Shoup – Oct '11

Boundary Institute

Cantor Diagonalization

$$p_1 : \bullet b_{11} \quad b_{12} \quad b_{13} \quad b_{14} \quad b_{15} \dots$$

$$p_2 : \bullet b_{21} \quad b_{22} \quad b_{23} \quad b_{24} \quad b_{25} \dots$$

$$p_3 : \bullet b_{31} \quad b_{32} \quad b_{33} \quad b_{34} \quad b_{35} \dots$$

$$p_4 : \bullet b_{41} \quad b_{42} \quad b_{43} \quad b_{44} \quad b_{45} \dots$$

$$p_5 : \bullet b_{51} \quad b_{52} \quad b_{53} \quad b_{54} \quad b_{55} \dots$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \dots$$

$$p_k : \bullet \sim b_{11} \quad \sim b_{22} \quad \sim b_{33} \quad \sim b_{44} \quad \sim b_{55} \dots$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \dots$$

i $b_{kk} = \sim b_{kk} !$

(N. Hellerstein)

R. Shoup – Oct '11

Boundary Institute

Infinities

- series
- continued fractions
- etc, etc
- ... *all self-referring.*

$$\pi = \frac{4}{1 + \frac{1^2}{3 + \frac{2^2}{5 + \frac{3^2}{7 + \ddots}}}}$$

$$\sum_{n=1}^{\infty} \frac{1}{2^n} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$$

$$\begin{cases} S \leftarrow S + m & (S_0 = 0) \\ m \leftarrow m/2 & (m_0 = 1/2) \end{cases}$$

Closely related:

- limits, continuity
- computability
- verification, FM

R. Shoup – Oct '11

Boundary Institute

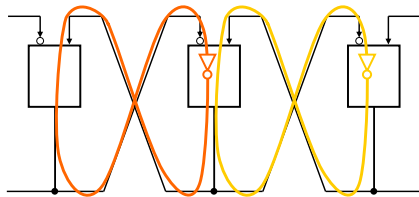
Complex Logics - Summary

- 4-valued logic holds simple Paradox (F, T, i, j).
 - Unify combinational and sequential domains
 - Time/Space trade-off: 4 timeless values \Leftrightarrow 2 values + time
 - Higher-order paradoxes, 2^{2^n} values \Leftrightarrow arbitrary waveforms
- Paradoxes are common, unavoidable (cf. Gödel), useful.
 - Bridge from finite to infinite forms, modeling Time
 - Looping synchronous or async circuits in hardware or software
 - Inconsistent databases, expert systems, logical controls, etc.
 - Proof techniques, re-examine *reductio ad absurdum*, etc.
- *Mathematics is not static & lifeless* – if self-ref is permitted.

R. Shoup – Oct '11

Boundary Institute

Imaginarities in Async Pipelines



Request/Acknowledge signaling \Rightarrow async transactions

R. Shoup – Oct '11

Boundary Institute

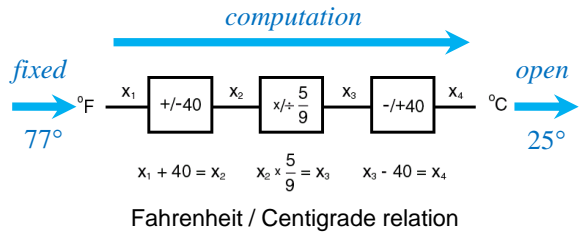
-- Combinatorial Relations --

*Tom Etter & Richard Shoup
Boundary Institute*

R. Shoup – Oct '11

Boundary Institute

Relations vs Functions

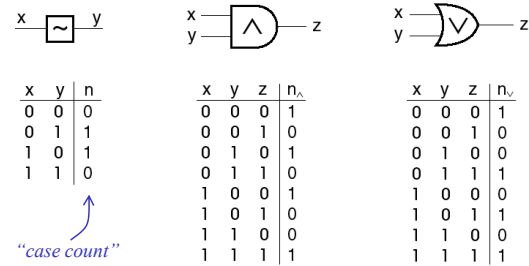


Today, digital computers are entirely functional, but quantum computers are naturally relational.

R. Shoup – Oct '11

Boundary Institute

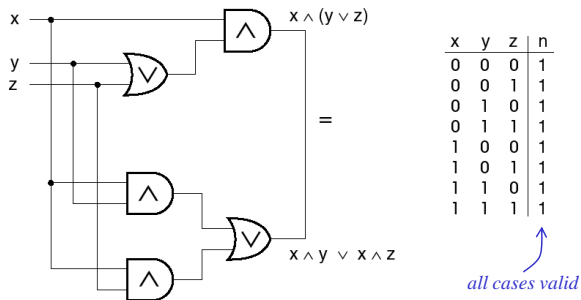
NOT, AND, OR as Relations



R. Shoup – Oct '11

Boundary Institute

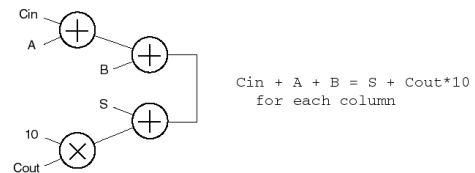
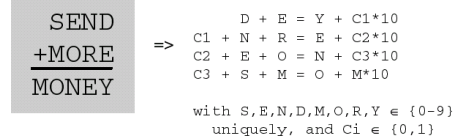
Example: Distributive Law



R. Shoup – Oct '11

Boundary Institute

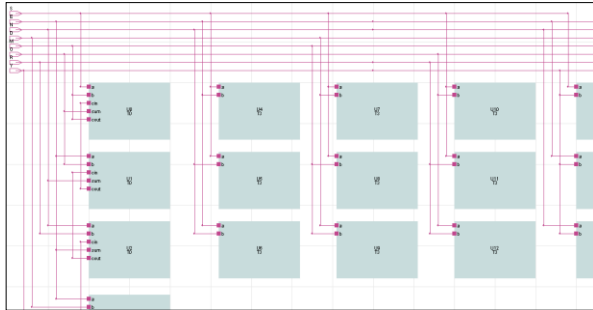
Example: Cryptarithm



R. Shoup – Oct '11

Boundary Institute

Cryptarithm Solution



R. Shoup – Oct '11

Boundary Institute

Cryptarithm Solution

SEND
+MORE

MONEY

$$\Rightarrow \begin{aligned} D + E &= Y + C_1 \cdot 10 \\ C_1 + N + R &= E + C_2 \cdot 10 \\ C_2 + E + O &= N + C_3 \cdot 10 \\ C_3 + S + M &= O + M \cdot 10 \end{aligned}$$

with $S, E, N, D, M, O, R, Y \in \{0-9\}$
uniquely, and $C_i \in \{0,1\}$

(4 tables, each 200 cases,
+ uniqueness tables)

S E N D M O R Y
9 5 6 7 1 0 8 2

=>

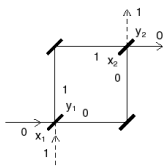
9567
+1085

10652

R. Shoup – Oct '11

Boundary Institute

Example: Quantum Interference



$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

x_1	y_1	n_1	x_2	y_2	n_2
0	0	1	0	0	1
0	1	1	0	1	-1
1	0	-1	1	0	1
1	1	1	1	1	1

\oplus

x_1	y_1	x_2	y_2	n
0	0	0	1	1
0	0	1	-1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	-1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

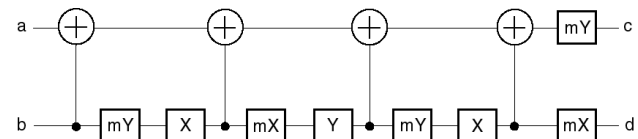
\Rightarrow

x_1	y_2	n
0	0	2
0	1	0
1	0	0
1	1	2

R. Shoup – Oct '11

Boundary Institute

Example: Quantum Computation



x	y	n_Y	n_{mY}	n_X	n_{mX}
0	0	1	1	1	1
0	1	1	-1	i	$-i$
1	0	-1	1	i	$-i$
1	1	1	1	1	1

x	y	z	n_{CNOT}
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

R. Shoup – Oct '11

Boundary Institute

Link Theory: Case Counts & Domains

all 1s : independent, uncorrelated
 one 1 : selector, state, classical
 more than one 1 : logic, relation, superposition
 one or more >1 : statistics, multisets, don't cares
 both + and - : quantum realm, interference

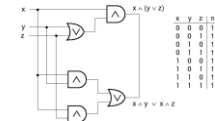
x	y	n ₀	n ₁	n ₂	n ₃	n ₄
0	0	1	0	0	1	1
0	1	1	0	1	2	1
1	0	1	0	0	2	-1
1	1	1	1	1	1	1

R. Shoup – Oct '11

Boundary Institute

Link Theory - Summary

- Theory of *composite relations* (multi-directional constraints)
- *Case counts*, a set of *possibilities*
- *Symmetrical*, directionless, timeless
- Represent a wide range of problems
 - Logic, combinatorics, arithmetic, statistical, ... , Qcomputing
 - New basis for discrete physics?
- General theory of *structure*
 - Knowledge representation
 - Quantum, molecular computing, etc
 - Evolutionary, self-organizing systems



R. Shoup – Oct '11

Boundary Institute

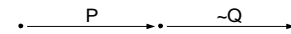
-- Sequential Constraints --

Fred Furtak
 Applied Combinatorics & Boundary Institute

R. Shoup – Oct '11

Boundary Institute

Sequential Constraints



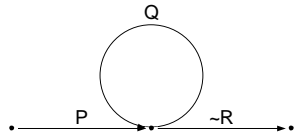
disallowed: [P] [~Q]

“Prohibited: P followed by NOT Q in the next state”
 “If P, then Q in the next state”
 “If NOT Q, then NOT P in the preceding state”

R. Shoup – Oct '11

Boundary Institute

Sequential Constraints III



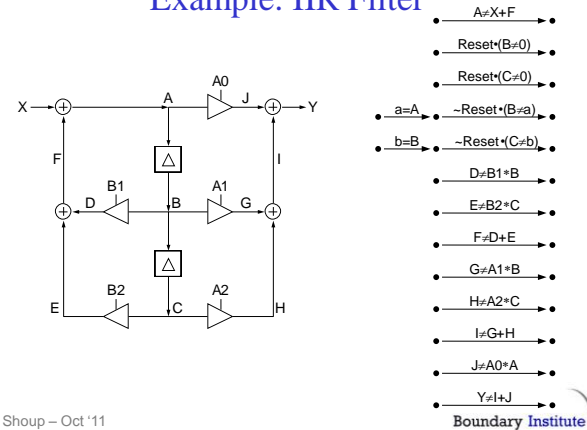
disallowed: $[P] [Q]^* [-R]$

“If P , then while Q , R in the next state”

R. Shoup – Oct '11

Boundary Institute

Example: IIR Filter

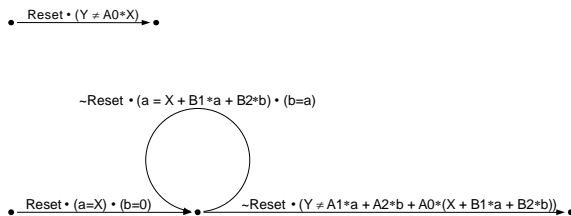


R. Shoup – Oct '11

Boundary Institute

IIR Filter: Visible Behavior

Resolve & delete the hidden signals (A,B,C,D,E,F,G,H,I,J) to obtain the visible (input-output) behavior:



R. Shoup – Oct '11

Boundary Institute

Sequential Constraint Graphs

- Set of behavioral, sequential *constraints*
- Constraint specifies *joint state* of variables in time
 - Logical, arithmetic, and timing *dependencies*
 - *Disallowed* states, what cannot occur
 - Equivalent to / based upon *regular expressions*
- Inference engine combines constraints
 - Proofs, queries, simulation, analyses of conditions, etc
- Formal Verification
 - Abstract, RTL or gate-level, hardware or software, etc
 - No global state, avoids most combinatorial explosions

R. Shoup – Oct '11

Boundary Institute

Present and Future Relevance

- Building highly *complex*, highly *parallel* systems
 - Scalability, fault tolerance, migration
 - Programming productivity, reuse
 - Power/efficiency
- Searching for *knowledge*, not just concepts
- Understanding represented via *relations* and *structure*
- Proving and maintaining *correctness*, *security*

All of these require computation as mathematics

R. Shoup – Oct '11

Boundary Institute

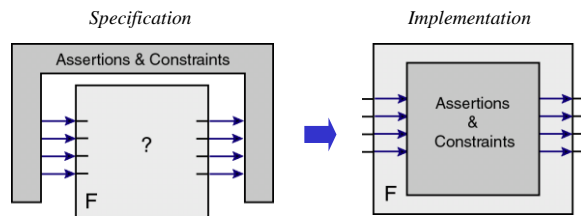
Future Computing Directions

- Single representation/language, unified hardware & software
- Deep primitives, foundations (e.g. Boundary Math)
 - Beneath/prior to Logic, Set Theory, Turing machines
- Theory of *structure* (e.g. Boundary Math + Link Theory)
 - Relational, hierarchical, self-referring, etc
- Formal techniques from the beginning
 - Initial specification, design, optimization, verification, debugging, field configuration, upgrading, etc.
 - Operation provably correct, by construction and anytime
 - Security, privacy provable and verifiable

R. Shoup – Oct '11

Boundary Institute

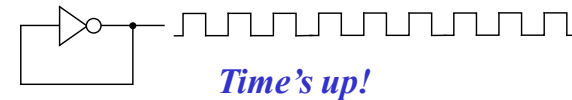
Example - Design by Eversion



A proper formal specification is an implementation.

R. Shoup – Oct '11

Boundary Institute



www.boundary.org
rshoup@boundary.org

Boundary Institute