Transmathematics

CNIC
Chinese Academy of Sciences
2009

Dr James A.D.W. Anderson
安德生

Translation

- 何辛 (Xin HE) School of Systems Engineering, University of Reading, England
- 文鹏程 (Pengcheng WEN) School of Precision Instrument and Optoelectronics Engineering, Tianjin University, China
- 陈璐璐 (Lulu CHEN) School of Systems Engineering, University of Reading, England

译

Agenda

- Transreal arithmetic
  超广义实数算术
- Advances in ordinary computing
  传统计算的优点
- Transcomplex arithmetic
  超广义复数算术
- Advances in supercomputing
  超级计算的优点
- Conclusion
  总结

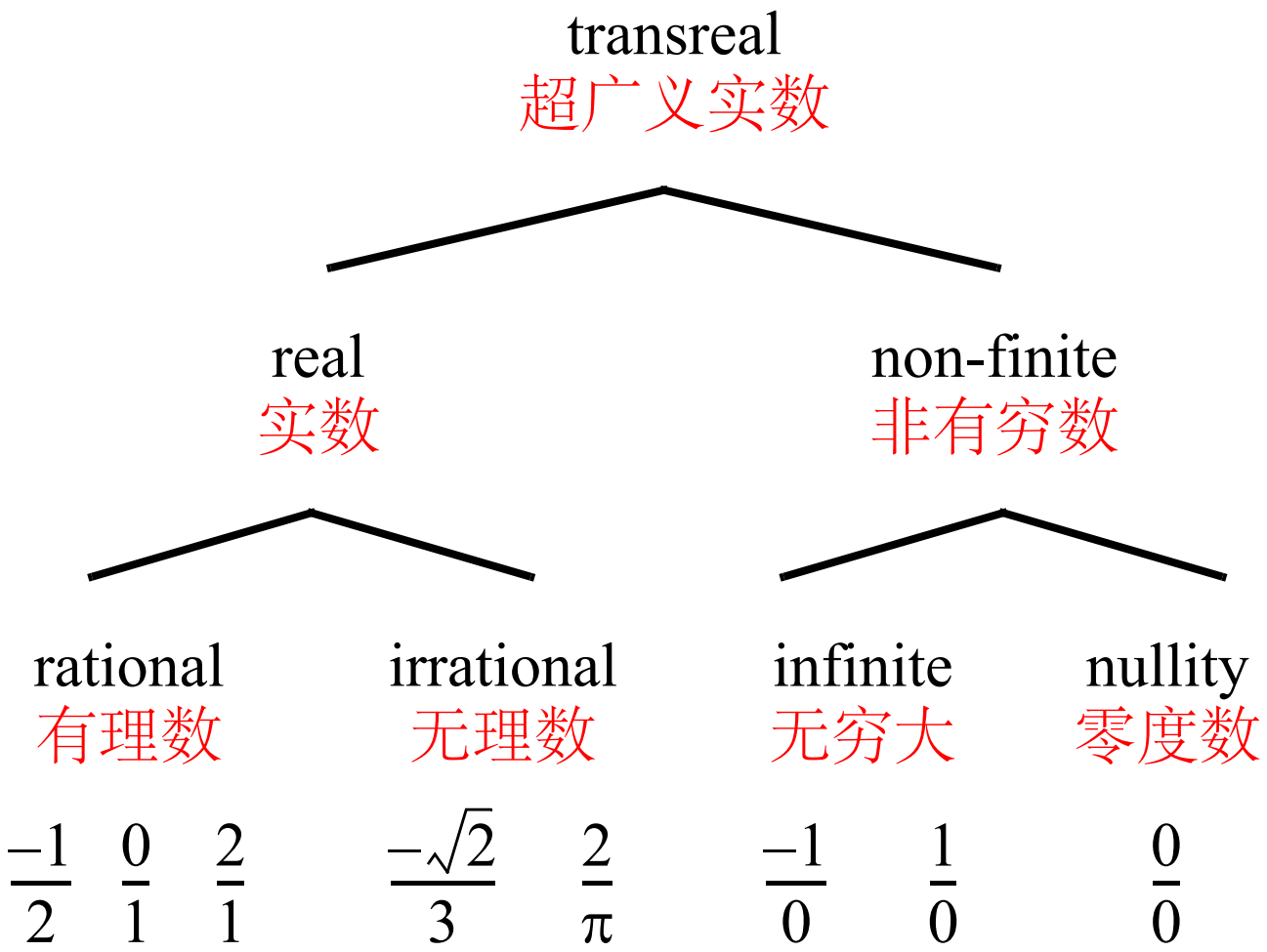
Introduction

- Ordinary arithmetic fails on division by zero
  传统数学无法解决一个数除以零的问题

- In 1997 the following ship was dead in the water for 2 hours and 45 minutes because a division by zero error cascaded through the ship's network crashing every computer on the network
  1997 年，下面这幅图中的轮船在海中意外滞留了 2 小时 45 分钟，分数分母为零的错误不断的发生在这艘船的网络中，导致所有联网的计算机都崩溃了。

USS Yorktown

Transreal Numbers



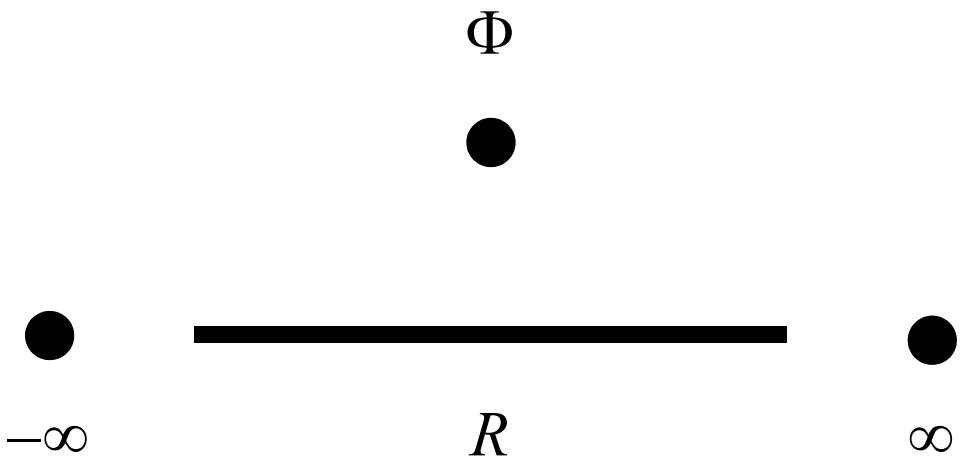
Transreal Numbers

The transreal numbers, R^T, are:

超广义实数 R^T 的定义是

- $R^T = R \cup \{-\infty, \infty, \Phi\}$

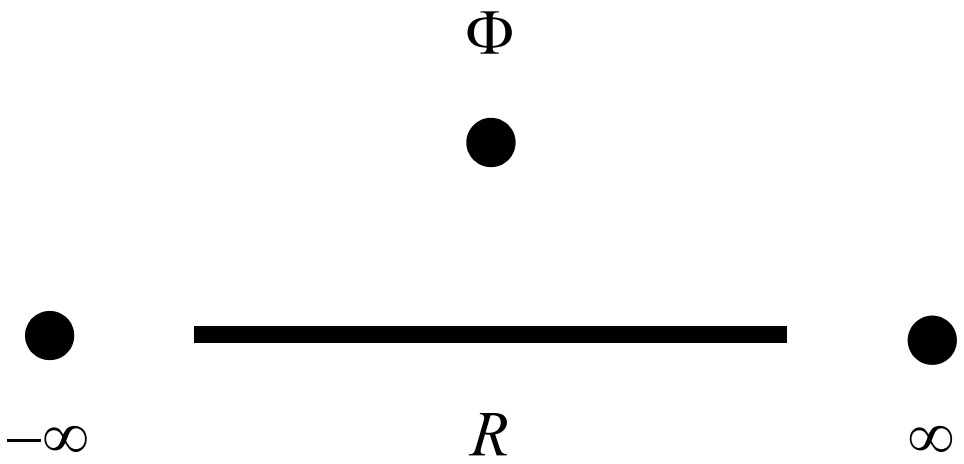
Transreal Numbers



- Positive infinity, ∞, is the biggest transreal number
  正无穷大，∞，是最大的超广义实数

- Negative infinity, −∞, is the smallest transreal number
  负无穷大，- ∞，是最小的超广义实数

Transreal Numbers



- Nullity, Φ, is the only transreal number that is not negative, not zero, and not positive
  零度数，Φ，是唯一的非负，非零，又非正的数

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Transreal Numbers

Let $k > 0$ then:
使 $k > 0$, 则有:

- $\infty = \frac{1}{0} \equiv \frac{k}{0}$
- $\Phi = \frac{0}{0}$
- $-\infty = \frac{-1}{0} \equiv \frac{-k}{0}$
- $0 = \frac{0}{1} \equiv \frac{0}{k} \equiv \frac{0}{-k}$

Transreal Fractions

A *transreal number* is a *transreal fraction* of the form $\frac{n}{d}$,

where:
每一个超广义实数都是一个形式为 n/d 的分数，其中：

- n is the *numerator* of the fraction
  n 是分子

- d is the *denominator* of the fraction
  d 是分母

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Transreal Fractions

- n, d are *real numbers*
  n 和 d 都是实数
- $d \geq 0$
- Examples:
  例如:

$$\frac{-1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{-\pi}{2}, \frac{-1}{2\pi}, \frac{1}{2\pi}, \frac{-1}{0}, \frac{1}{0}, \frac{0}{0}$$

Transreal Fractions

- An *improper transreal fraction*, $\frac{n}{-d}$, may have a negative denominator, $-d < 0$
  一个假超广义实数分数 n/-d 可能有一个负数的分子，即 $-d < 0$

- An improper transreal fraction is converted to a *proper transreal fraction* by multiplying both the numerator and the denominator by -1
  假超广义实数分数可以通过对分子和分母分别乘以 -1 的方法变成真超广义实数。

Transreal Fractions

- Example: $\frac{2}{-3} = \frac{-1 \times 2}{-1 \times (-3)} = \frac{-2}{3}$

- Example: $\frac{0}{-1} = \frac{-1 \times 0}{-1 \times (-1)} = \frac{0}{1}$

Transreal Multiplication

Two *proper transreal fractions* are multiplied like this:
两个真超广义实数分数相乘:

- $\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d}$

- Example: $3 \times \infty = \frac{3}{1} \times \frac{1}{0} = \frac{3 \times 1}{1 \times 0} = \frac{3}{0} = \infty$

- Example: $0 \times \infty = \frac{0}{1} \times \frac{1}{0} = \frac{0 \times 1}{1 \times 0} = \frac{0}{0} = \Phi$

Transreal Multiplication

- Example: $\frac{1}{2} \times \frac{3}{5} = \frac{1 \times 3}{2 \times 5} = \frac{3}{10}$

Transreal Division

Two *proper transreal fractions* are divided like this:
两个真超广义实数分数相除:

- $\frac{a}{b} \div \frac{c}{d} = \frac{a}{b} \times \frac{d}{c}$

- Example: $\infty \div 3 = \frac{1}{0} \div \frac{3}{1} = \frac{1}{0} \times \frac{1}{3} = \frac{1 \times 1}{0 \times 3} = \frac{1}{0} = \infty$

Transreal Division

- Example:

$$\infty \div (-3) = \frac{1}{0} \div \frac{-3}{1} = \frac{1}{0} \times \frac{1}{-3} = \frac{1}{0} \times \frac{-1 \times 1}{-1 \times (-3)}$$

$$= \frac{1}{0} \times \frac{-1}{3} = \frac{1 \times (-1)}{0 \times 3} = \frac{-1}{0} = -\infty$$

- Example: $\frac{1}{2} \div \frac{5}{3} = \frac{1}{2} \times \frac{3}{5} = \frac{1 \times 3}{2 \times 5} = \frac{3}{10}$

Transreal Addition

Two *proper transreal fractions* are added like this:
两个真超广义实数分数相加:

- $\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$, except that: 除了:

- $(\pm\infty) + (\pm\infty) = \frac{\pm 1}{0} + \frac{\pm 1}{0} = \frac{(\pm 1) + (\pm 1)}{0}$

Transreal Addition

- $(\pm\infty) + (\pm\infty) = \frac{\pm 1}{0} + \frac{\pm 1}{0} = \frac{(\pm 1) + (\pm 1)}{0}$

Examples:

- $\infty + \infty = \frac{1}{0} + \frac{1}{0} = \frac{1 + 1}{0} = \frac{2}{0} = \infty$

- $(-\infty) + (-\infty) = \frac{-1}{0} + \frac{-1}{0} = \frac{(-1) + (-1)}{0} = \frac{-2}{0} = -\infty$

- $\infty + (-\infty) = \frac{1}{0} + \frac{-1}{0} = \frac{1 + (-1)}{0} = \frac{0}{0} = \Phi$

Transreal Addition

- $\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$

Examples:

- $\frac{2}{3} + \infty = \frac{2}{3} + \frac{1}{0} = \frac{2 \times 0 + 3 \times 1}{3 \times 0} = \frac{3}{0} = \infty$
- $\frac{2}{3} + \Phi = \frac{2}{3} + \frac{0}{0} = \frac{2 \times 0 + 3 \times 0}{3 \times 0} = \frac{0}{0} = \Phi$
- $\frac{2}{3} + \frac{4}{5} = \frac{2 \times 5 + 3 \times 4}{3 \times 5} = \frac{22}{15}$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Transreal Subtraction

Two *proper transreal fractions* are subtracted like this:
两个真超广义实数分数相减:

- $\frac{a}{b} - \frac{c}{d} = \frac{a}{b} + \frac{-c}{d}$

Examples:

- $\infty - \infty = \frac{1}{0} - \frac{1}{0} = \frac{1}{0} + \frac{-1}{0} = \frac{1 + (-1)}{0} = \frac{1 - 1}{0} = \frac{0}{0} = \Phi$
- $\frac{1}{2} - \frac{3}{5} = \frac{1}{2} + \frac{-3}{5} = \frac{(1 \times 5) + (2 \times (-3))}{2 \times 5} = \frac{5 + (-6)}{10} = \frac{-1}{10}$

Moral

- The arithmetic you have just seen has been taught to 12 year old children in England
  你们看到的这些算术，英国很多 12 岁的孩子都学过
- These children understand infinity and nullity
  他们完全可以理解无穷大和零度数
- These children use an arithmetic that never fails
  而且在用这些算术方法计算时，也没遇到过任何问题
- What do you want for your children?
  你们也期望自己的孩子能这样吗

Axioms

- There is a computer proof that the following axioms of transreal arithmetic are consistent
  计算机验证表明，以下超广义实数算术的公理是前后一致，自成体系的

- The *sign* function is used as a short hand in axiom [31]
  以下符号函数在公理 [31] 中被简写为 sgn()

$$\begin{aligned}\mathrm{sgn}(a) = {} & \text{if } 0 < a \text{ then } 1 \\ & \text{elseif } 0 = a \text{ then } 0 \\ & \text{elseif } a < 0 \text{ then } -1 \\ & \text{else } \Phi\end{aligned}$$

Axioms (Addition)

Additive Associativity:

加法结合律:

$$a + (b + c) = (a + b) + c \qquad [A1]$$

Additive Commutativity:

加法交换律:

$$a + b = b + a \qquad [A2]$$

Axioms (Addition)

Additive Identity:

加法单位元:

$$0 + a = a \qquad [A3]$$

Additive Nullity:

加法零度数:

$$\Phi + a = \Phi \qquad [A4]$$

Additive Infinity:

加法无穷大:

$$a + \infty = \infty : a \neq -\infty, \Phi \qquad [A5]$$

Axioms (Subtraction)

Subtraction as Sum with Opposite:

减法是被减数与减数的相反数相加:

$$a - b = a + (-b) \qquad [A6]$$

Bijectivity of Opposite:

相反数的双射性:

$$-(-a) = a \qquad [A7]$$

Additive Inverse:

加法逆元:

$$a - a = 0 : a \neq \pm\infty, \Phi \qquad [A8]$$

Axioms (Subtraction)

Opposite of Nullity:

零度数的相反数:

$$-\Phi = \Phi \qquad [A9]$$

Non-null Subtraction of Infinity:

无穷大的非零度减法:

$$a - \infty = -\infty : a \neq \infty, \Phi \qquad [A10]$$

Subtraction of Infinity from Infinity:

无穷大减无穷大

$$\infty - \infty = \Phi \qquad [A11]$$

Axioms (Multiplication)

Multiplicative Associativity:

乘法结合律:

$$a \times (b \times c) = (a \times b) \times c \qquad [A12]$$

Multiplicative Commutativity:

乘法交换律:

$$a \times b = b \times a \qquad [A13]$$

Multiplicative Identity:

乘法单位元:

$$1 \times a = a \qquad [A14]$$

Axioms (Multiplication)

Multiplicative Nullity:

乘法零度数:

$$\Phi \times a = \Phi \qquad [A15]$$

Infinity Times Zero:

无穷大乘以零:

$$\infty \times 0 = \Phi \qquad [A16]$$

Axioms (Division)

Division:

除法:

$$a \div b = a \times (b^{-1}) \qquad [A17]$$

Multiplicative Inverse:

乘法逆元:

$$a \div a = 1 : a \neq 0, \pm\infty, \Phi \qquad [A18]$$

Bijectivity of Reciprocal:

倒数的双射性:

$$(a^{-1})^{-1} = a : a \neq -\infty \qquad [A19]$$

Axioms (Division)

Reciprocal of Zero:

零的倒数:

$$0^{-1} = \infty \qquad [A20]$$

Reciprocal of the Opposite of Infinity:

无穷大相反数的倒数:

$$(-\infty)^{-1} = 0 \qquad [A21]$$

Reciprocal of Nullity:

零度数的倒数:

$$\Phi^{-1} = \Phi \qquad [A22]$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Axioms (Ordering)

Positive:

正:

$$\infty \times a = \infty \Leftrightarrow a > 0 \qquad [A23]$$

Negative:

负:

$$\infty \times a = -\infty \Leftrightarrow 0 > a \qquad [A24]$$

Positive Infinity:

正无穷大:

$$\infty > 0 \qquad [A25]$$

Axioms (Ordering)

Ordering:

排序:

$$a - b > 0 \Leftrightarrow a > b \qquad [A26]$$

Less Than:

小于:

$$a > b \Leftrightarrow b < a \qquad [A27]$$

Greater Than or Equal:

大于等于:

$$a \geq b \Leftrightarrow (a > b) \vee (a = b) \qquad [A28]$$

Axioms (Ordering)

Less Than or Equal:

小于等于:

$a \leq b \Leftrightarrow b \geq a$ [A29]

Axioms (Quadrachotomy)

Quadrachotomy:
四分法:

Exactly one of
必为以下四种情况之一
$(a < 0)$, $(a = 0)$, $(a > 0)$, $(a = \Phi)$ [A30]

Axioms (Distributivity)

Distributivity:

分配律:

$$a \times (b + c) = (a \times b) + (a \times c) :$$

$$\neg((a = \pm\infty) \wedge (\operatorname{sgn}(b) \neq \operatorname{sgn}(c)) \wedge (b + c \neq 0, \Phi))$$

[A31]

Axioms (Lattice Completeness)

Lattice Completeness:
格完备性:

The set, X, of all transreal numbers, excluding Φ, is lattice complete because
对于除了 Φ 之外的所有超广义实数，集合 X 是格完备的，因为

$$\forall Y : Y \subseteq X \Rightarrow$$
$$(\exists u \in X : (\forall y \in Y : y \leq u) \wedge$$
$$(\forall v \in X : (\forall y \in Y : y \leq v)$$
$$\Rightarrow u \leq v))$$

[A32]

Transreal Arithmetic

- Transreal arithmetic is a superset of real arithmetic
  超广义实数算术是实数算术的扩展

- Transreal arithmetic is *total* – every operation of transreal arithmetic can be applied to any transreal numbers and the result is a transreal number
  超广义实数算术是完整的 —— 超广义实数算术的每一个运算都可以应用于任一超广义实数，其结果也是一个超广义实数

Transreal Arithmetic

- Real arithmetic is *partial* – it fails on division by zero and on each of the infinitely many mathematical consequences of division by zero
  实数算术是不完整的 —— 实数算术不能进行除以零的运算，也不能从无穷多个除以零的运算中得到任何数学结果

Advances in Standard Computing

- All mathematical software can be extended to use transreal numbers
  所有的数学软件都可以扩展应用于超广义实数

- Mathematical software using transreal numbers never fails on an arithmetical exception
  应用于超广义实数的数学软件绝不会在处理任何数学特例时崩溃。

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Exponential Function

For all transreal x:
对于所有的超广义实数 x:

$$\exp(x) = \begin{cases} (\exp(-x))^{-1} : x < 0 \\ \lim_{k \to \infty} 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^k}{k!} : \text{otherwise} \end{cases}$$

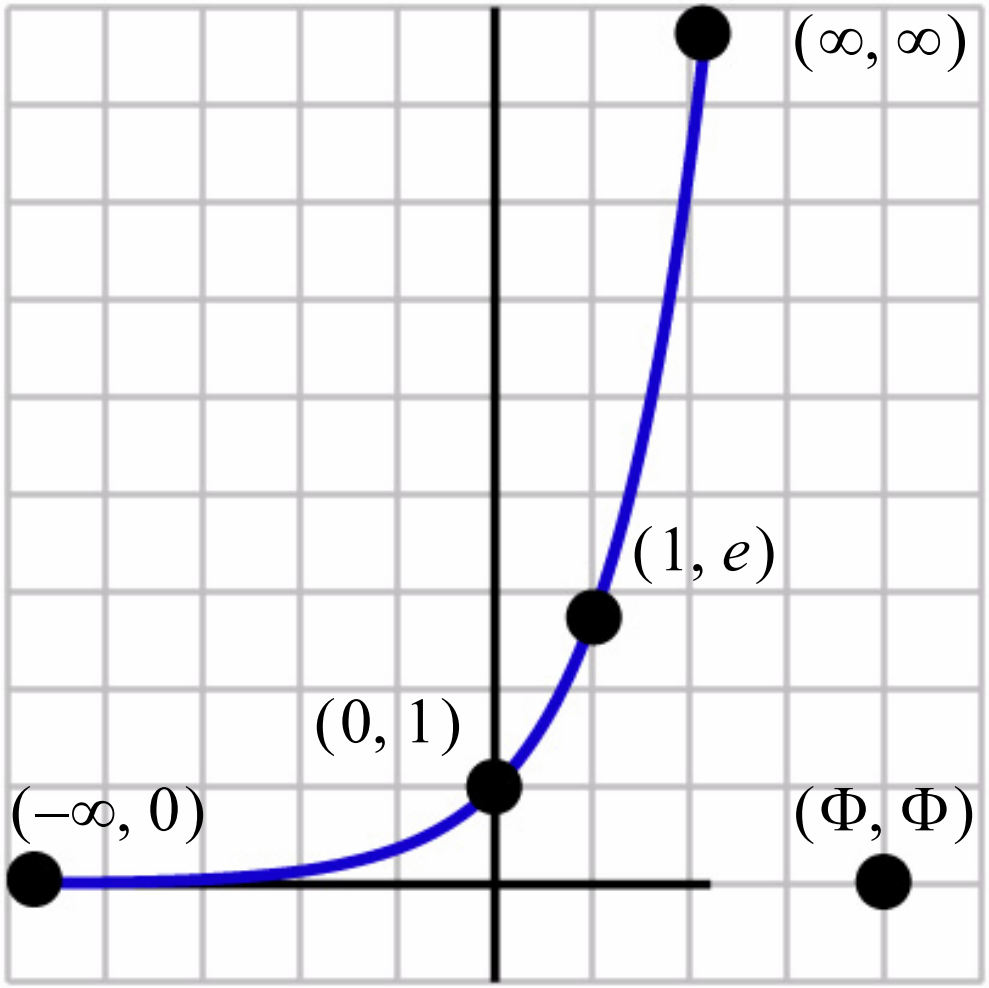
其他

Whence:
由此:

$$\exp(-\infty) = 0, \exp(\infty) = \infty, \exp(\Phi) = \Phi$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Exponential Function



Natural Logarithm

The transreal logarithm is well defined for all non-negative transreal numbers. In particular:
对所有的非负超广义实数，超广义实数对数都被很好地定义了，特别是：

- $\ln \Phi = \ln e^{\Phi} = \Phi$
- $\ln \infty = \ln e^{\infty} = \infty$
- $\ln 0 = \ln e^{-\infty} = -\infty$

The polar-transcomplex logarithm is used later
极坐标表示的超广义复数对数会在之后被介绍

Hyperbolic Trigonometry

The hyperbolic trigonometric equations are defined for all transreal numbers:
对所有的超广义实数，双曲线三角方程被定义为:

- $\sinh x = \frac{e^{x} - e^{-x}}{2}$

- $\cosh x = \frac{e^{x} + e^{-x}}{2}$

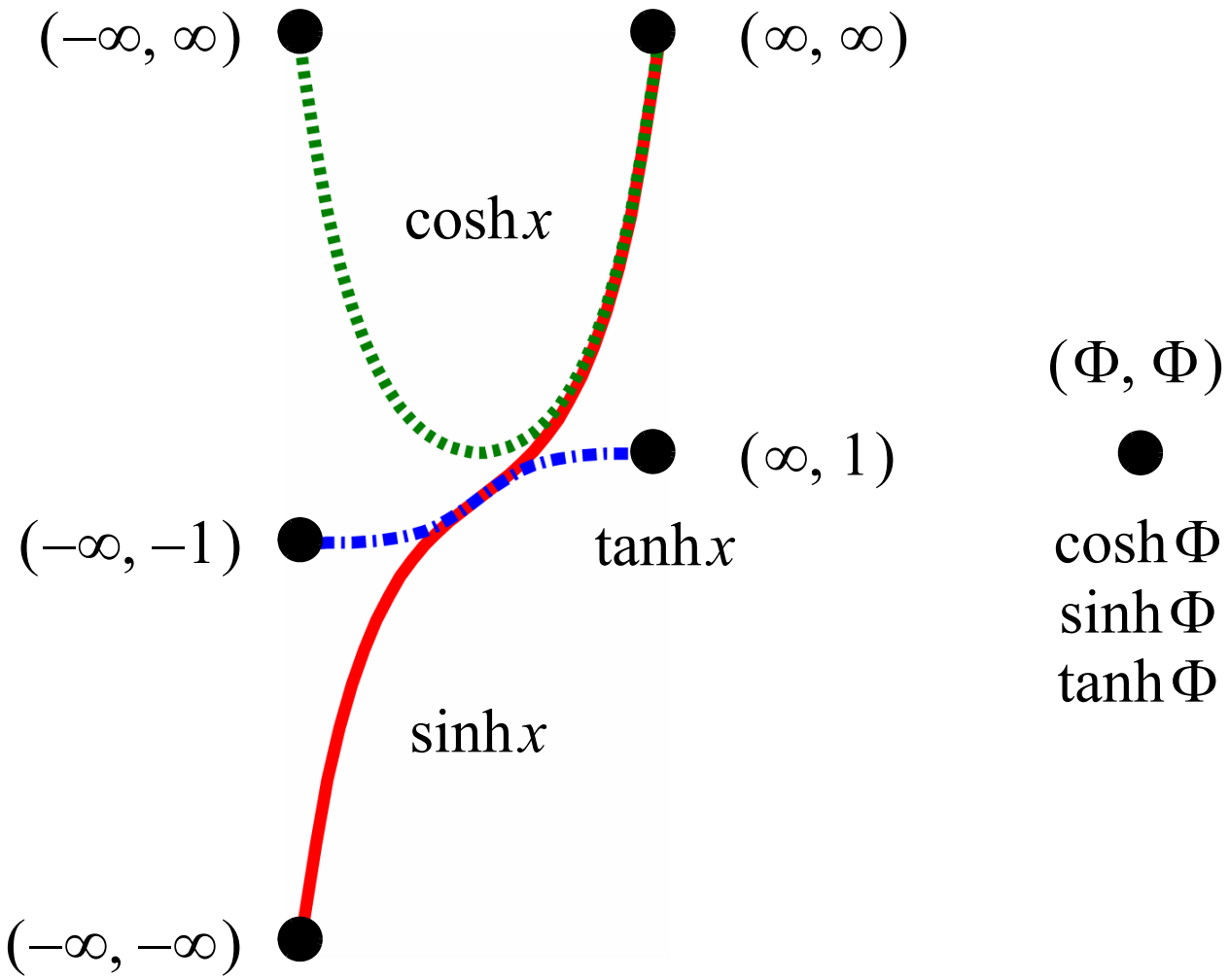
Hyperbolic Trigonometry

- The boundary conditions at $x = \pm\infty$ force *tanh* to be continuous on $-\infty \le x \le \infty$
  设定边界条件，x = ± ∞，则正切 tanh 在区间 - ∞ ⩽ x ⩽ ∞ 上连续：

$$\tanh x = \begin{cases} -1 : x = -\infty \\ 1 : x = \infty \\ \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}} : \text{otherwise} \end{cases}$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Hyperbolic Trigonometry



Hyperbolic Trigonometry

Without the boundary conditions:
在无边界条件下，

$$\tanh x = \begin{cases} \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \end{cases}$$

Whence:
由此，

$$\tanh(-\infty) = \tanh(\infty) = \tanh(\Phi) = \Phi$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

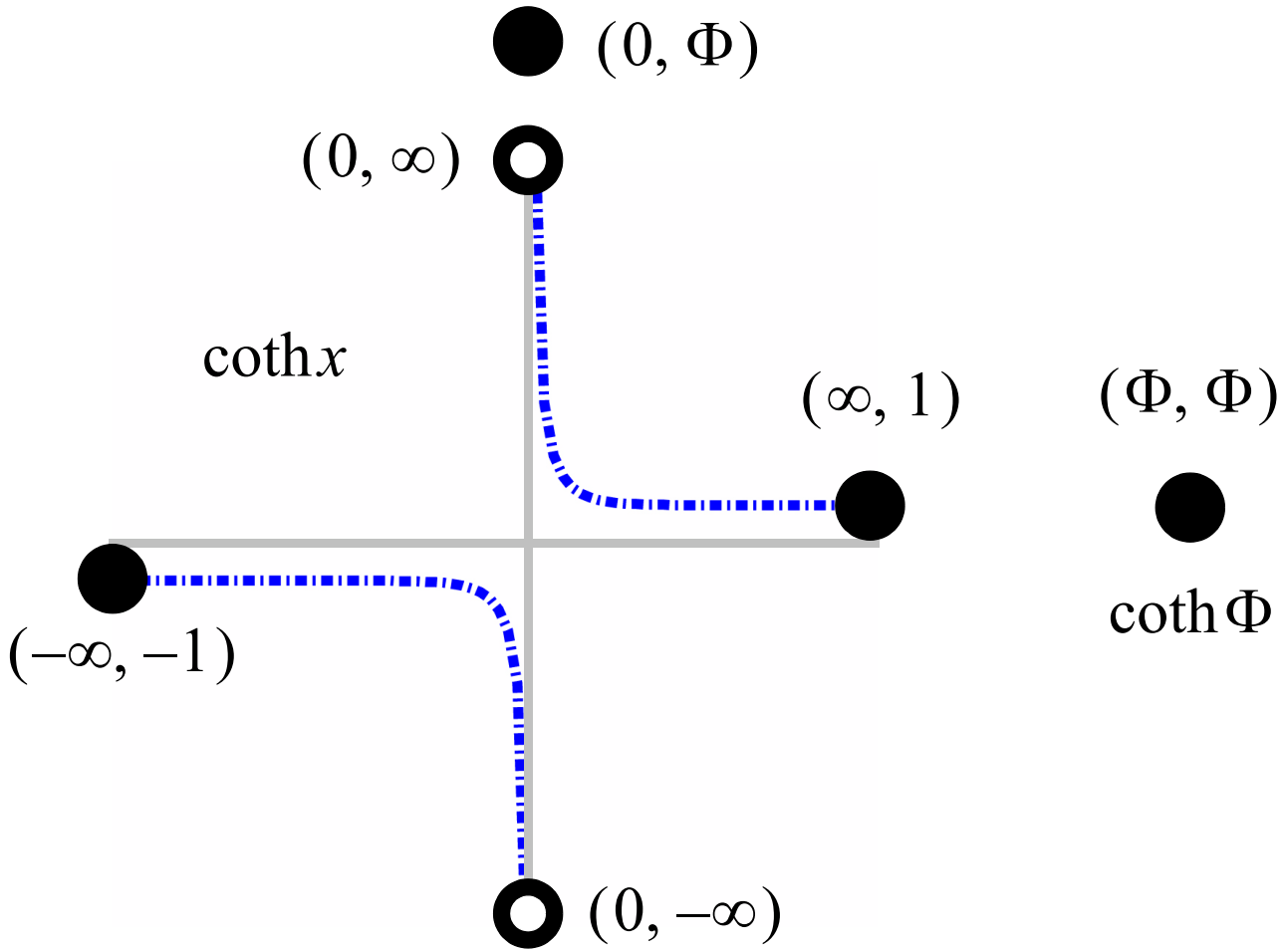
Hyperbolic Trigonometry

- The following boundary conditions puncture *coth* at $x = 0$, thereby breaking it into two, symmetrical, continuous pieces on $-\infty \leq x < 0$ and $0 < x \leq \infty$
在下面边界条件的约束下，在点 x = 0 处，余切曲线 coth 被分解为两段对称的连续曲线，分别在区 - ∞ ≤ χ < 0 和区间 0 < χ ≤ ∞ 上

$$\coth x = \begin{cases} -1 : x = -\infty \\ \Phi : x = 0 \\ 1 : x = \infty \\ \dfrac{e^{x} + e^{-x}}{e^{x} - e^{-x}} : \text{otherwise} \end{cases}$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Hyperbolic Trigonometry



Continuity

- In ordinary mathematics $\coth 0$ is *undefined:*
  在传统数学中，0 的余切 coth0 是没有定义的

$$\coth 0 = \frac{e^{0} + e^{-0}}{e^{0} - e^{-0}} = \frac{1+1}{1-1} = \frac{2}{0}$$

Continuity

- In ordinary mathematics there is no real continuity constraint that can be applied to make $\coth x$ continuous at $x = 0$ so we cannot take $\coth x$ as *indefinite*
在传统数学中没有实际存在的连续性约束条件使 cothx 在点 x ＝ 0 处连续，因而 cothx （当 x= 0 时）不能被定义为无穷大

Continuity

- In ordinary mathematics we could use *topology* to force continuity of coth$x$ at $x = 0$ by *identifying* the infinities as $-\infty = \infty$
  在传统数学中，我们可以通过拓扑结构，认定无穷大为 $-\infty = \infty$，使 cothx 在点 $x = 0$ 处连续

Continuity

- In transreal arithmetic we can compute an unsigned infinity $y = \infty$ from two signed infinities, $+\infty$ and $-\infty$, as $y = |\pm\infty|$. We do this using *arithmetic,* we do not need to invoke or program a topological solution
在超广义实数算术中，一个无符号无穷大 $y = \infty$ 可以通过用两个有符号的无穷大 $+\infty$ 和 $-\infty$ 计算得到，记 $y = |\pm\infty|$。这里只需要使用算术，而不需要调用或者编制一个拓扑结构的程序去解决这个问题。

Continuity

- We need to be careful to use signed infinities on the x-axis so that $\coth(-\infty) = -1$ and $\coth(\infty) = 1$, whilst using unsigned infinities on the y axis so that
$\coth(0) = -\infty = +\infty = \infty$
对于 x 坐标轴上的有符号无穷大应注意，$\coth(-\infty) = -1$, $\coth(\infty) = 1$. 而对于 y 坐标轴上无符号无穷大也应注意，$\coth(0) = -\infty = +\infty = \infty$

Continuity

- But there is a better solution using transreal arithmetic:
  用超广义实数算术可以更好地解决这个问题:

$$\frac{e^{0}+e^{-0}}{e^{0}-e^{-0}} = \frac{1+1}{1-1} = \frac{2}{0} = \infty$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

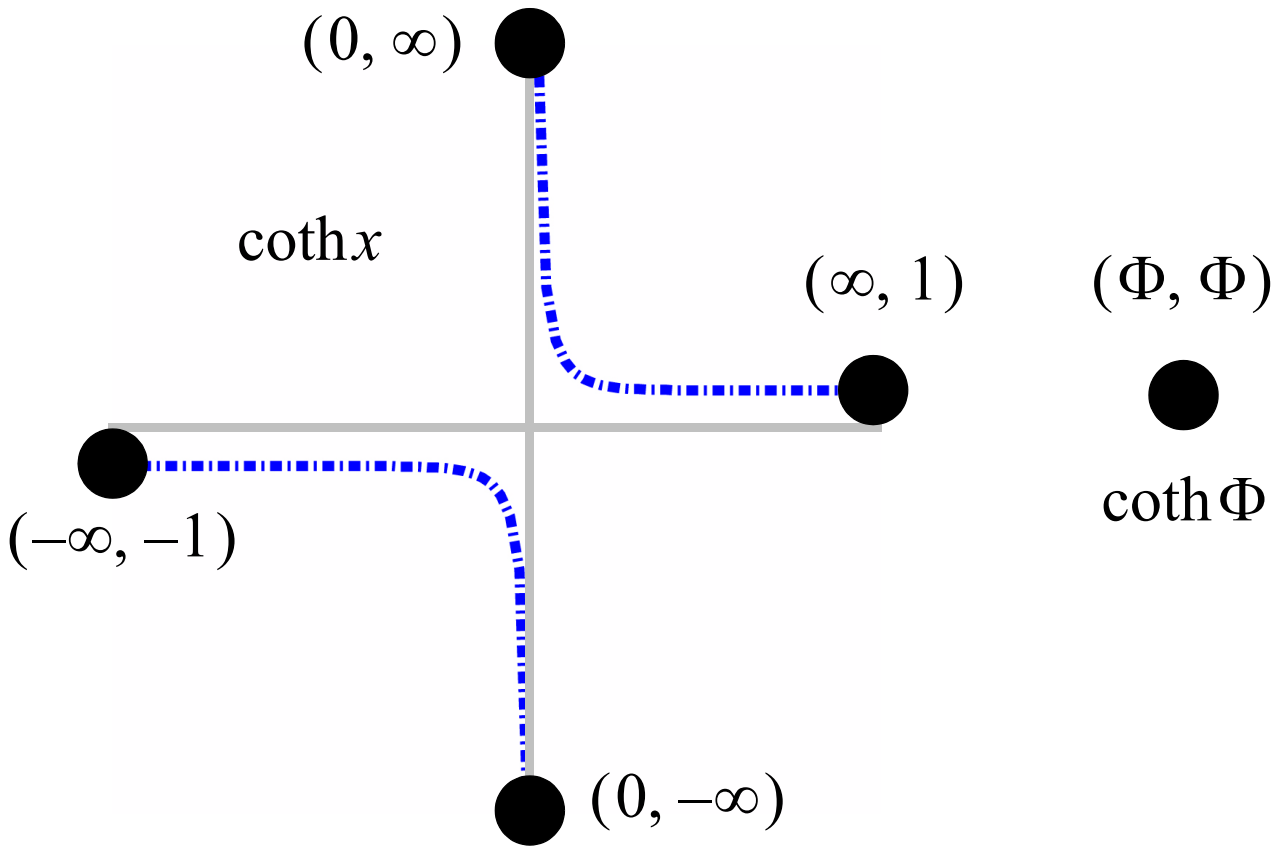
Continuity

- Now we define $\coth x$ as a piecewise continuous function in two pieces with one isolated nullity:
  现在，定义 cothx 为一个在两区间（其中一个区间里包括零度数）上的分段连续函数。

$$\coth x = \begin{cases} -\coth(-x) & : \ x < 0 \\ 1 & : \ x = \infty \\ \dfrac{e^{x} + e^{-x}}{e^{x} - e^{-x}} & : \ \text{otherwise} \end{cases}$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Continuity



Continuity

- The topology of the transreal numbers breaks continuity at the non-finite numbers so as to avoid contradictions in transreal arithmetic and in all of its mathematical consequences, such as in *trigonometry* and *calculus,* but we can choose to restore continuity in special functions, as we have just done

超广义实数的拓扑结构打破了非有穷数的连续性，从而避免了在超广义实数算术及其一切相关的数学结果中出现不一致，如三角算术和微积分。但是，我们可以选择在特殊函数中恢复其连续性，如以上刚刚介绍的法则

Continuity

- Ordinary mathematics does not settle the question of what to do with undefined or indefinite values
  传统数学无法解决有关无定义或无确定值的数的问题

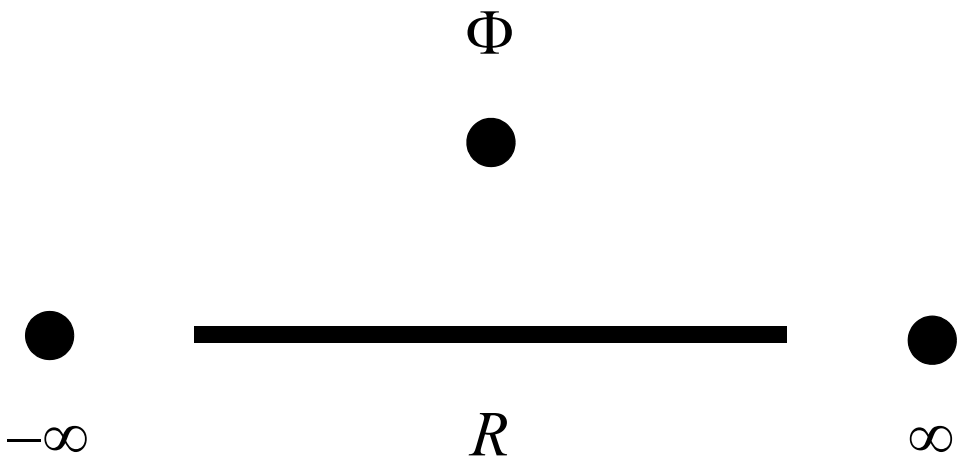
- In transreal arithmetic, and in all of its mathematical consequences, no undefined or indefinite values exist
  在超广义实数算术及其一切相关的数学结果中，无定义或无确定值的数都是不存在的。

Continuity

- Transreal arithmetic is total so it forces us to consider every possible case when solving a problem. Consequently the solution is a constructive proof that no undefined or indefinite value exists

超广义实数算术是完整的，在解决问题时涵盖了各种可能性，它的所有数学结果都可以有力地证明，（在传统数学中可能碰到的）无定义或者无确定值的数都不会出现。

Continuity



Metric Spaces

Metric spaces are defined over a metric, m, which obeys four axioms:
度量空间是在度量 m 上定义的，并遵循四条公理：

$$m(a, b) = m(b, a) \quad \text{[M1]}$$

$$m(a, b) \geq 0 \quad \text{[M2]}$$

$$m(a, b) = 0 \Leftrightarrow a = b \quad \text{[M3]}$$

$$m(a, b) + m(b, c) \geq m(a, c) \quad \text{[M4]}$$

Transmetric Spaces

Replacing *greater-than-or-equals* with *not-less-than* generalises metric spaces to transmetric spaces with a transmetric t:

替换 大于等于 为 不小于，并引入一个超广义度量 t，使得度量空间扩展为超广义度量空间：

$$t(a, b) = t(b, a) \qquad [T1]$$

$$t(a, b) \not< 0 \qquad [T2]$$

$$t(a, b) = 0 \Leftrightarrow a = b \qquad [T3]$$

$$t(a, b) + t(b, c) \not< t(a, c) \qquad [T4]$$

Transmetric Spaces

- Transmetric spaces are a superset of metric spaces
  超广义度量空间是（传统数学中的）度量空间的一个超集
- Transmetric spaces allow a distance of nullity
  超广义度量空间可以定义长度为零度数的距离
- Transmetric spaces generalise limiting processes so that, for example, power series and calculus are generalised so that they have no undefined or indefinite values
  超广义度量空间扩展了有限运算，例如，幂级数和微积分通过对定义的扩展，去除了无定义或无确定值的数

Transmetric Spaces

- In particular, every power series has a sum to infinity
  特别是，每一个冥级数都可以是无穷多个项（即包括 n = ∞）的加和，而非趋近于无穷多个项（即包括 $n \to \infty$）的加和（即无限项之和）。

- The sum to infinity of 1 – 1 + 1 – 1 + … is given later
  1 － 1 ＋ 1 － 1 ＋⋯ 这样无穷次的加和会在之后被引出

Trigonometry

The trigonometric equations are defined for all transreal numbers by taking their power series without boundary conditions. In particular:

通过在无边界条件下对冥级数的展开，三角方程在所有超广义实数上都有定义。特别是

- $\cos(\pm\infty) = \cos(\Phi) = \Phi$
- $\sin(\pm\infty) = \sin(\Phi) = \Phi$
- $\tan(\pm\infty) = \tan(\Phi) = \Phi$

Powers of Unity

- $1^{\Phi} = e^{\ln 1^{\Phi}} = e^{\Phi \ln 1} = e^{\Phi \times 0} = e^{\Phi} = \Phi$
- $1^{\infty} = e^{\ln 1^{\infty}} = e^{\infty \ln 1} = e^{\infty \times 0} = e^{\Phi} = \Phi$
- $1^{-\infty} = e^{\ln 1^{-\infty}} = e^{-\infty \ln 1} = e^{-\infty \times 0} = e^{\Phi} = \Phi$

Transcomplex powers of -1 are given later
-1 的超广义复数幂会在之后被介绍

Trigonometric Identities

Trigonometric identities hold for all transreal x. In particular:
三角恒等式对所有超广义实数 x 都有意义。特别的：

- $\cos^2 x + \sin^2 x = 1^x$
- $\cosh^2 x - \sinh^2 x = 1^x$

Moral

- Ordinary mathematics is guaranteed to fail on division by zero and on each of the infinitely many consequences of division by zero
  传统数学中，被零除的运算及其得到的所有结果都没有意义。

- Transreal arithmetic never fails on division by zero nor on any of its infinitely many consequences – division by zero always succeeds
  相反地，在超广义实数运算中，被零除的运算及其得到的所有结果都有意义。

Moral

- Ordinary mathematics has infinitely many undefined and indefinite values
  传统数学中有无穷多的无定义和无确定值的数
- Transmathematics has no undefined or indefinite values
  超广义数学中不存在任何无定义或无确定值的数

Moral

- There is a very great deal of work to do to extend ordinary mathematics to transmathematics. Some of this work is easy, some of it is hard
  将传统数学扩展为超广义数学存在着大量的工作。其中一些很容易实现，一些则相对很难

- Do you have the strength to remove guaranteed failure from the mathematics that supports your society?
  在已经在现代社会里根深蒂固的传统数学中，你能避免那些被认为必然会存在的运算失败吗？

Advances in Standard Computing

Every syntactically correct transarithmetical expression is semantically correct so:
对于语法上正确的所有超广义算术表达式，其在语义上也是正确的，因此：

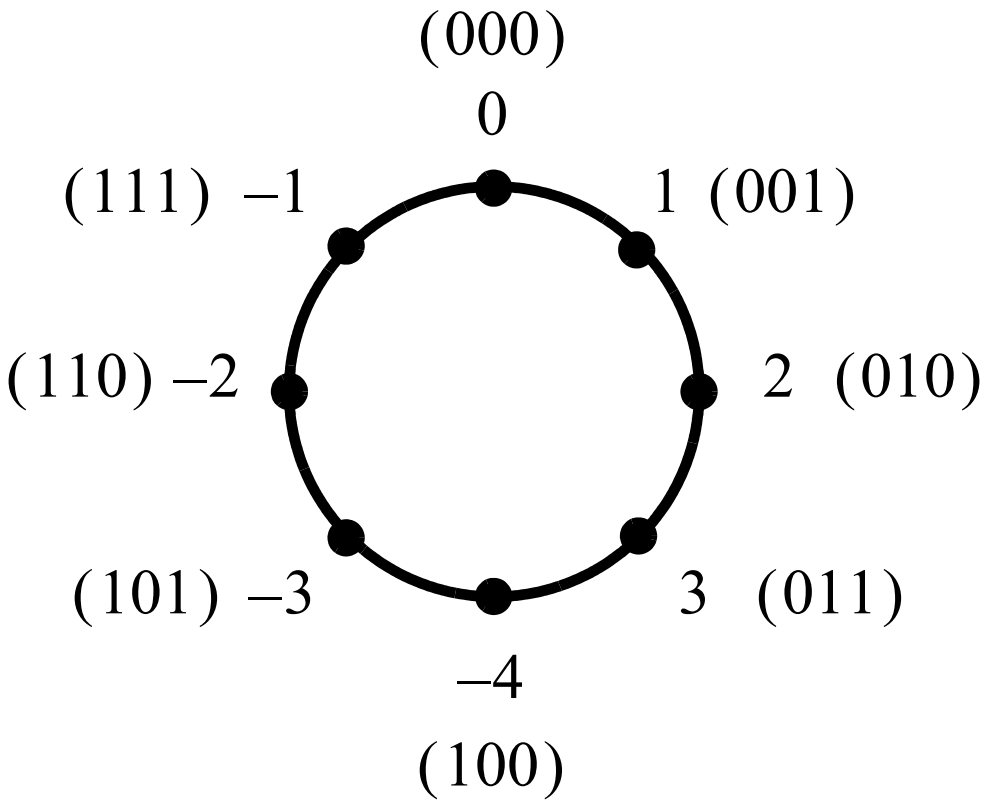
- Compilers can perform full type checking
  编译器可以实行完整的型别检查
- There are no arithmetical run-time errors
  算术上也没有任何运行时错误

Advances in Standard Computing

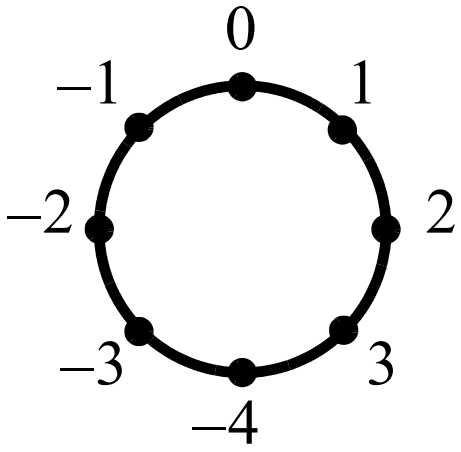
- Transreal arithmetic removes an intrinsic bug from integer (two's complement) arithmetic, making both hardware and software safer
  超广义实数运算修复了由整数（补码）运算引起的内部漏洞，从而使在硬件和软件上的操作都更加安全

- Floating-point hardware can have all wasted states re-allocated to transreal numbers, thereby improving arithmetical range or accuracy
  浮点硬件设备能够使所有被浪费的位模式都重新分配到超广义实数中去，从而扩大了运算范围，同时提高了准确度

Two's Complement



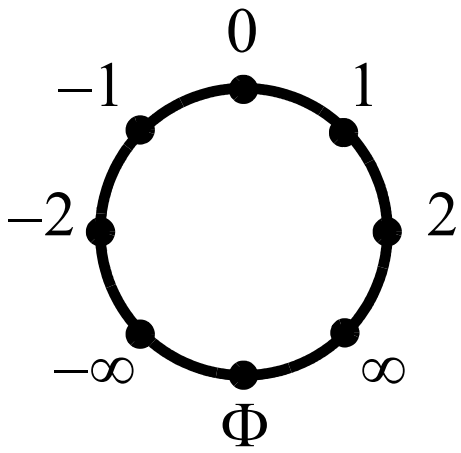
Two's Complement



- The complement of the most negative number is not its negation –(–4) = –4
  最小负数的补数并不是其本身的相反数，如，-(-4) = -4

- Almost every computer suffers this weird-number fault
  几乎每一台计算机都可能受累于由以上异常数（weird-number）引起的错误

Trans Two's Complement

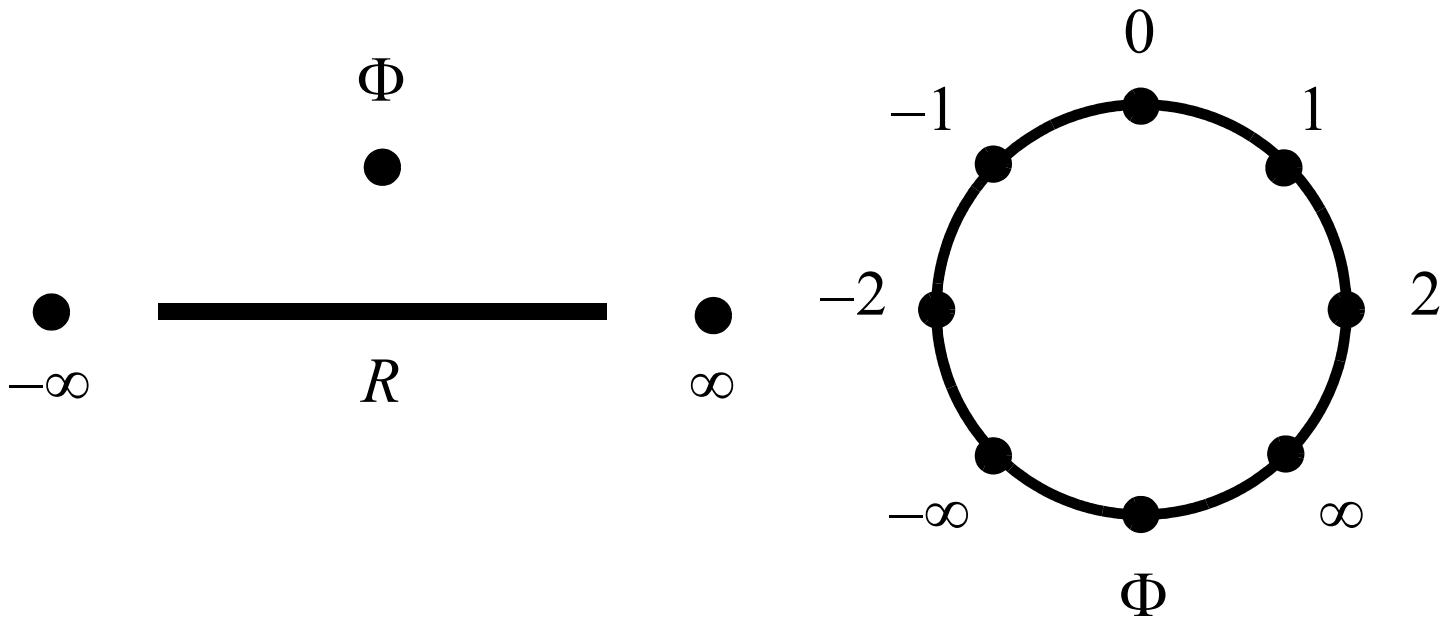


- The complement of the most negative number is now its negation $-(-\infty) = \infty$
  在引入了零度数之后，负数的极值永远都是其本身的相反数，如图中 $-(-\infty) = \infty$

- And the complement of nullity is its negation $-\Phi = \Phi$
  而且，零度数的补数也是其自身的相反数，即 $-\Phi = \Phi$

Trans Two's Complement

- Trans two's complement removes the weird-number fault and preserves the topology of the transreal numbers
  超广义补码排除了异常数错误（weird-number error），从而保留了超广义实数的拓扑结构



Trans Two's Complement

- Trans two's complement gives transinteger and transfixed-point programming superior exception handling to floating-point arithmetic
  超广义补码的引入使超广义实数和超广义定点程序有更强的浮点运算异常处理能力

- The topology of transreal numbers extends to floating-point arithmetic so that it can match the exception handling of transinteger and transfixed arithmetic
  当超广义实数的拓扑结构增加了浮点运算时，它就可以同时对超广义实数和超广义定点运算进行异常处理

Floating-Point

- IEEE 754 defines floating point numbers in terms of three of bit fields that encode the Sign (S), Exponent (E) and Mantissa (M)

  IEEE754 浮点运算标准是从浮点数组成的三个位元域来定义的，它们分别对应符号位元 （S），指数位元 （E），和尾数位元 （M）

Floating-Point

- In general, a floating point number $n = -1^{S}2^{E}M$, but bit patterns are reserved for –0, –∞, ∞, NaN_i where $i = 2^{m+1} - 2$ with m being the number of bits explicitly represented in the mantissa. The "+1" arises from the sign bit and the "–2" from –∞ and +∞

通常情况下，浮点数可以表示为 $n = -1^{S}2^{E}M$，但一些位模式会被预留下，来表示 -0, -∞, ∞, NaN_i（非数值），其中，$i = 2^{m+1} - 2$, (m 是尾数域位元的个数）；“+1”是因为多了一个有符号位，而“-2”是因为正负无穷（±∞）不应被算在内。

Floating-Point

- IEEE arithmetic encodes –0, but –0 does not occur in transreal arithmetic so transfloating-point arithmetic reallocates the binary code for –0 to Φ

  IEEE 运算对－０有编码，但是，由于超广义实数运算中不产生－０，超广义浮点运算会沿用 IEEE 标准中对－０的编码，但视之为 Φ

Floating-Point

- Nullity now lies in the middle of the lexical collation range of floating-point numbers so sorting routines must handle the unique nullity as a special case (IEEE sorting routines must handle all NaNs as special cases)
零度数处于浮点数语句整理的区域上，因此排序例程在运行中只需要对零度数一个值作特殊值处理（IEEE 排序例程必须对所有的非数值 （NaN）作特殊值处理）

Floating-Point

- Notice that IEEE arithmetic collates numbers in reverse order because S is the most significant bit and $S = 1$ encodes negative numbers
  注意，IEEE 标准中对数值的整理是逆向的，这里 S 是最高有效位 （MSB），当 S ＝ 1 时，代表此位模式为负

Floating-Point

- IEEE arithmetic uses reverse collation so that the binary codes for integer zero and floating-point positive-zero are identical. As many computers have a test zero instruction this makes positive-zero comparisons quick

  IEEE 运算标准使用逆向整理方式，从而使整数零和浮点型正零相同。很多计算机都存在一个测试零的指令，这使得对正零的比较处理更迅速

Floating-Point

- IEEE arithmetic uses the most positive binary code for ∞ and the most negative binary code for −∞, transfloating-point arithmetic does the same
  IEEE 运算标准定义正二进制码的极值为∞，负二进制码的极值为－∞，此定义在超广义浮点运算中同样适用

Floating-Point

- IEEE arithmetic has $2^{m+1} - 2$ NaNs, but transreal arithmetic has no NaNs so all of these states can be reallocated to transreal numbers

  IEEE 运算标准中有 2^{m+1}-2 个非数值，而超广义实数运算中不存在任何的非数值，因此，每一个这样的位模式都可以被定义为一个新的超广义实数

Floating-Point

- Taking the reallocated codes as signed numbers means that there are $2^m - 1$ new mantissas. This is almost one binade: $2^m - 1$ bit patterns in a near binade as against 2^m bit patterns in an entire binade

把以上的这些模式位看成有符号数，无形中增加了 2^{m-1} 个新的尾数。这几乎就是一个 binade，因为一个整个的 binade 包含 2^m 个位模式，而这里有 2^{m-1} 个，因此我们称之为近似 binade （near binade）。

Floating-Point

- Leaving the exponent bias unchanged takes this near binade with a positive exponent so as to almost double the arithmetical range of floating-point numbers
  在保持指数偏差不变的情况下，取指数为正的近似binade （near binade）可以使浮点数的范围扩大几乎一倍。

Floating-Point

- Incrementing the bias takes this near binade with a positive exponent, but frees up an entire binade with a negative exponent, thereby delaying underflow to denormal numbers
  试图增大偏差（使偏差加一），会使这个近似 binade 的指数为正，但多出了一个指数为负的整个的 binade，在下溢之前产生了许多极小的大于零的数，从而延迟了到反常值的下溢。

Floating-Point

- Only one of these options can be taken in one thread: either increase the range or else increase the accuracy
  在一个线程中只有以下一种可以实现：要么扩大范围，要么提高精确度

Floating-Point

- 64-bit Intel chips have an 80-bit accumulator with $m = 64$. This wastes $2^{65} - 2$ states

  使用 带有尾数域位元个数为 64 的 80 位累加器 的英特尔 64 位芯片，相当于浪费了 2^{65} - 2 个位模式

$$2^{65} - 2 = 36\,893\,488\,147\,419\,103\,230$$

Floating-Point

- The IEEE standard defines four relational operations: *less-than* (<), *equal* (=), *greater-than* (>), *unordered* (?)
  IEEE 标准定义了四种关系运算：小于 （<），等于 （＝），大于 （>），无序 （？）
- Transreal arithmetic defines three relational operations: *less-than* (<), *equal* (=), *greater-than* (>)
  而超广义实数运算中定义了三种关系运算：小于 （<），等于 （＝），和大于 （>）

Floating-Point

- The IEEE standard defines 14 composite relations:
  IEEE 标准中定义了 14 种组合关系:

  =, ?<>, >, >=, <, <=, ?, <>, <=>, ?>, ?>=, ?<, ?<=, ?=

- The IEEE standard defines negations of 12 out of 14 of the composite relations:
  并且定义了其中 12 种关系的否定形式:

  NOT(>), NOT(>=), NOT(<), NOT(<=), NOT(?),
  NOT(<>), NOT(<=>), NOT(?>), NOT(?>=),
  NOT(?<), NOT(?<=), NOT(?=)

Floating-Point

- I have never seen a computer language that supports all of these 26 composite relations with 26 relational operators
  我从没见过任何一种计算机语言支持所有这 26 种组合关系 （用到所有这 26 种关系运算符）

Floating-Point

- Transreal arithmetic supports 7 composite relations:
  超广义实数运算支持 7 种组合关系:

  =, >, >=, <, <=, <>, <=>

- Transreal arithmetic supports negations of all of the composite relations:
  并且支持所有这 7 种组合关系的否定形式

  !=, !>, !>=, !<, !<=, !<>, !<=>

Floating-Point

- Transreal arithmetic preserves symmetry of negation that the IEEE standard breaks, this makes it easier to program with transreal numbers
  不同于 IEEE 标准，超广义实数运算保有对所有组合关系肯定和否定形式的一一对应关系，这使得应用超广义实数做程序设计更容易

Floating-Point

- 12 of the IEEE relations can raise an exception:
  IEEE 标准中的以下 12 种关系可能会产生异常：

  >, >=, <, <=, <>, <=>, NOT(>), NOT(>=), NOT(<), NOT(<=), NOT(<>), NOT(<=>)

- Specifically, all of the relations that do not contain the predicate *unordered* can raise an exception on NaN
  特别是那些不包含无序的关系，可能会在非数值上产生异常

Floating-Point

- None of the transreal relations can raise an exception so it is easier and safer to program with transreal numbers
  没有任何超广义实数运算关系可能产生异常，因此，用超广义实数做程序设计更容易、更安全。

Floating-Point

- The IEEE standard defines that $\mathrm{NaN}_i \neq \mathrm{NaN}_j$ so that it is false that $x = x$ for some floating-point objects, x
  IEEE 标准定义 NaNi ≠ NaNj, 因此有，对于一些浮点对象 x，x ＝ x 是错误的

- The above breaks a cultural stereotype that everything is equal to itself and destroys equality in mathematical physics so that mathematical physics does not work with NaN
  上面的结果打破了“任何物事都等于其自身”的传统意识理念，同时也破坏了数学物理学中的等价概念，因此非数值在数学物理学中不适用

Floating-Point

- Transreal arithmetic has $x = x$ for all transreal numbers, x
  超广义实数运算中，x = x 对所有超广义实数 x 都适用

- The above preserves a cultural stereotype that everything is equal to itself and maintains equality in mathematical physics
  以上结论符合“每一个事物都等于其自身”的传统意识理念，也符合数学物理学中的等价性概念

Moral

- Do you want Chinese computer chips to outperform Western ones?
  你希望中国的计算机芯片结构优于西方吗？
- Do you want Chinese computer languages to outperform Western ones?
  你希望中国的计算机语言优于西方吗？

Moral

- Do you want Chinese mathematics libraries to outperform Western ones?
  你希望中国的数学库优于西方吗？
- Do you want Chinese computer applications to outperform Western ones?
  你希望中国的计算机应用程序优于西方吗？

Supercomputing

China's intellectual property laws prevent us from telling you how our supercomputer works, but we will tell you about one of our old designs that is in the public domain

由于中国的知识产权法，我们不能具体地介绍我们的超级计算机是如何工作的，但是今天我们会介绍我们过去的一个用在公共领域上的设计

Supercomputing

Our old machine operates on numbers and aims to achieve truly massive, fine-grain parallelism, with extremely fast memory by:
我们过去的机器在数字运算上，以（用超快速存储器）实现真正的大规模、精细的并行计算 为目标，并通过以下技术实现：

- Aggressively simplifying the cores so that they are very much smaller than conventional cores
  大规模简化处理核的结构，使其比传统内核要小很多

Supercomputing

- Passing tokens on an escalator bus so that all cores can simultaneously receive and transmit tokens in every direction with zero Input/Output (I/O) latency
  在 escalator 传送总线上作令牌传递，使所有的处理核在每个方向上都能同时接收和传送令牌，并且时延为零

- Holding all working memory in on-chip cache, thereby accessing memory at processor speeds
  通过把所有运行中的内存负荷在芯片缓存上，可以使内存访问速度达到中央处理器的速度

Supercomputing

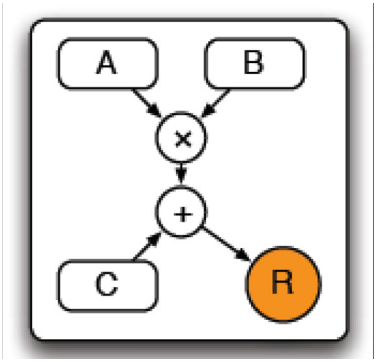
- Running I/O from every edge of the chip with zero latency so that there is no memory wall
  在芯片的边缘部分进行零时延的读写操作，则避免了内存墙问题。

Aggressive Simplification

- Hardware layouts say we can achieve 10 k cores on a chip
  精心设计的硬件结构使我们可以在一块芯片上装一万个处理核

Aggressive Simplification

- In the fixed-point machine, the only arithmetical operation is $A \times B + C \rightarrow R$:
  在定点计算机中，唯一的算术运算为 A×B+C → R:



Aggressive Simplification

- All other arithmetical operations are synthesised from this one, because fabricating them would waste space in most of the processors on a chip
  所有其它的算术运算都是通过对这个运算的不同组合得到的，因为如果专门定义那些运算会耗费芯片上绝大多数的处理器空间。

- The floating-point chip also has one non-arithmetical operation
  浮点处理芯片也能进行非算术运算

Aggressive Simplification

Fetchless architecture:
非提取结构:

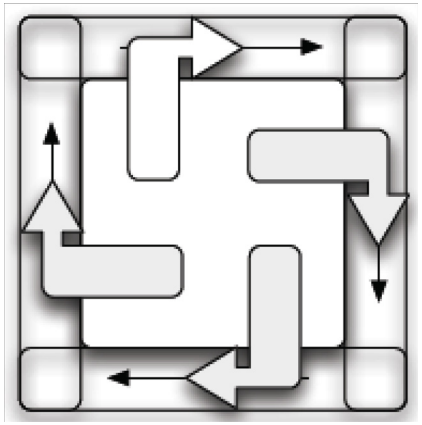
- Minimises circuitry
  最小化电路

- Reduces on-chip fetch-latency to zero
  将片上提取延迟时间降低到零

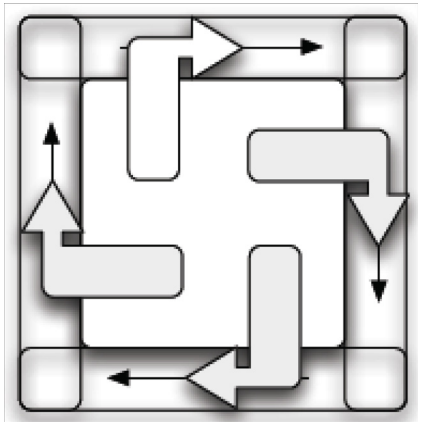
Achieved by token passing
由令牌传递方式实现

Aggressive Simplification

- Tokens transmitted, conditionally on the sign of $A \times B + C \rightarrow R$, through every edge of the square processor and internally (Grey arrow blocked, white transmitted)
  根据 A×B+C → R 的符号，令牌在方形处理器的各边缘及内部传递 （灰色箭头代表通道堵塞，白色箭头代表传递）

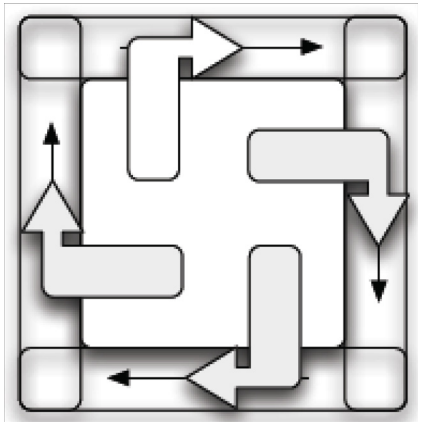


Aggressive Simplification



- There are four signs – negative, zero, positive, nullity – encoded in two sign bits so all selectors are maximally efficient
  四种符号（负、零、正、零度数）用两个符号位进行编码，这使得所有的选择器都能实现效率的最大化

Aggressive Simplification



- If desired, each of the four separate signs can transmit a token in a different direction
  如果需要，四种独立符号中的每一种都能在不同的方向上传递令牌

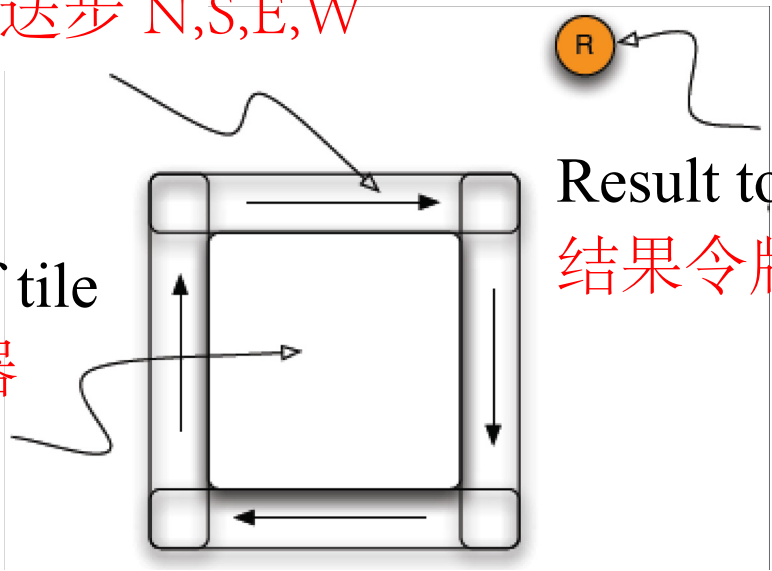
Aggressive Simplification

There are no arithmetical error states so:
由于没有算术异常状态:

- There is no arithmetical error handling circuitry on a processor
  在我们的处理器中，没有应对算术异常的处理电路
- Program execution is more secure
  程序执行更加安全

Core Tile

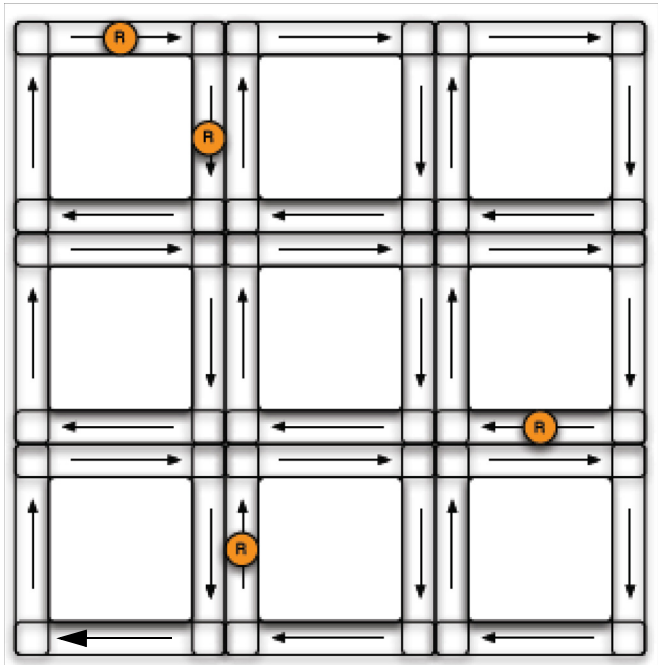
Escalator step
N, S, E, W
Escalator 传送步 N,S,E,W



Result token
结果令牌

Processor in centre of tile
瓷方元中心的处理器

Escalator Bus

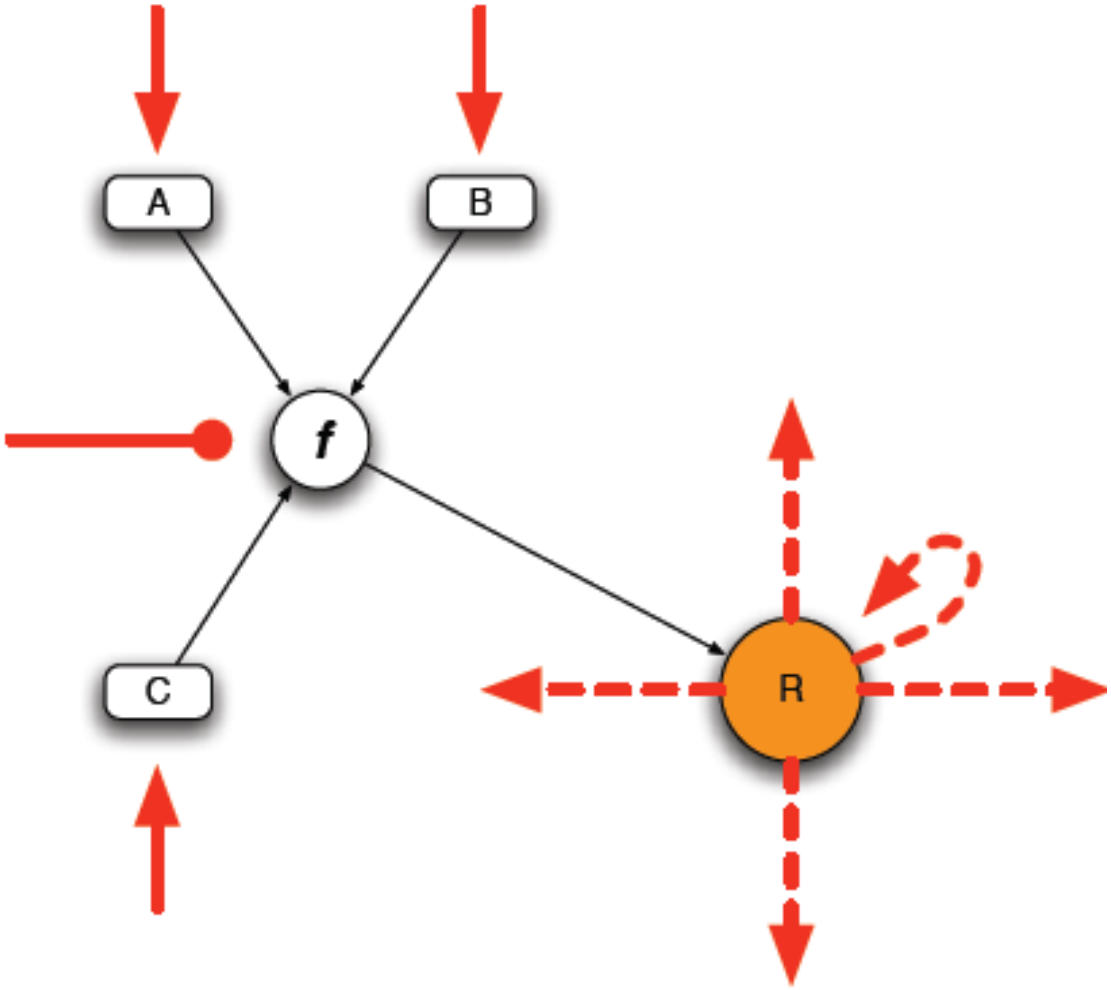


- Escalator bus emerges from the processor tiles
  Escalator 传送总线由处理器中的瓷方元构成

Escalator Bus

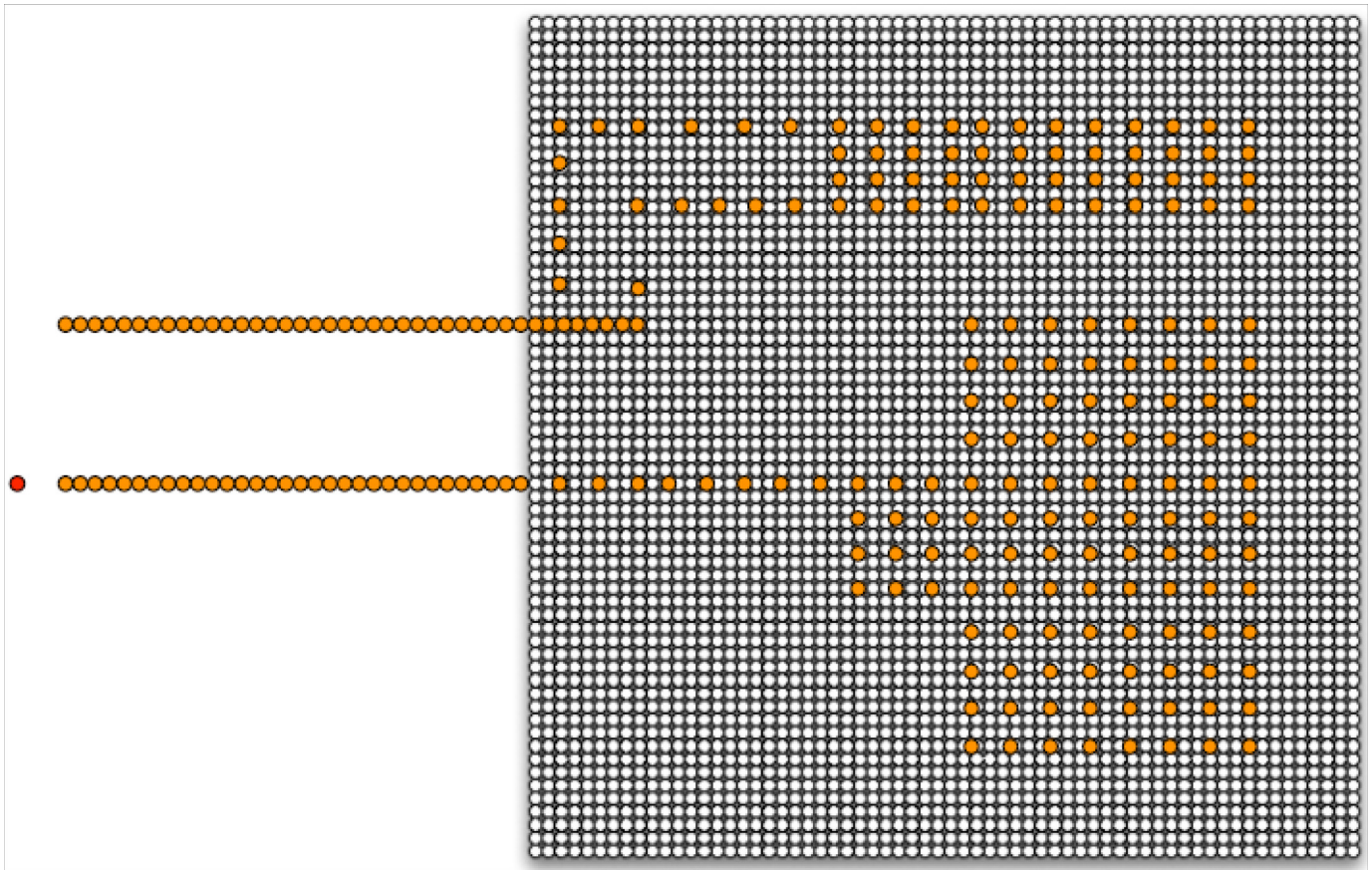
- Every core can simultaneously read from and write to the escalator bus on every processor clock cycle with zero latency
  在处理器的任何一个时钟周期内，每一个处理核都能同时对 escalator 传送总线进行零时延的读写操作

Programming: an Army of Ants



James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Programming: Initial Orders



Programming: Is it Possible?

We have programmed these machines:
我们已经 编程实现了:

- Emulated a Turing complete machine
  仿真了一台图灵完全机
- Eliminated race conditions by using a single thread
  应用单线程模式消除竞态条件
- Eliminated race conditions by travel-time inequalities
  应用传播时间不等性消除竞态条件

Programming: Is it Possible?

- Implemented fully pipelined, mathematical functions with a throughput of one result per clock cycle, and a latency down to half that of the Itanium 2 on: reciprocal, reciprocal square root, square root, exponential

  实施全流水线式结构，数学函数的计算达到每时钟周期一个结果的速度，而且在计算倒数、平方根倒数、平方根、指数时，其延迟时间降低到安腾 2 处理器的一半

Programming: Is it Possible?

- Pipelines do not break on branches, they just tee-off in some city-block direction within the 2D surface of a chip
  流水线在分支中不中断，它们会沿着（从二维平面的角度看）芯片上的某一城市街区方向继续

- Non-recursive subroutines have call and return implemented by branching so when they are in-lined they do not break pipelines
  非递归的子程序通过分支实现调用与返回，因此当它们内联时，不会使流水线中断

Programming: Is it Possible?

- Multiple calling points for a single subroutine introduce bubbles in each pipeline, and may break the pipeline
  对于单一子程序的多调用点，在每一条流水线中引入冒泡排序机制，这有可能使流水线中断

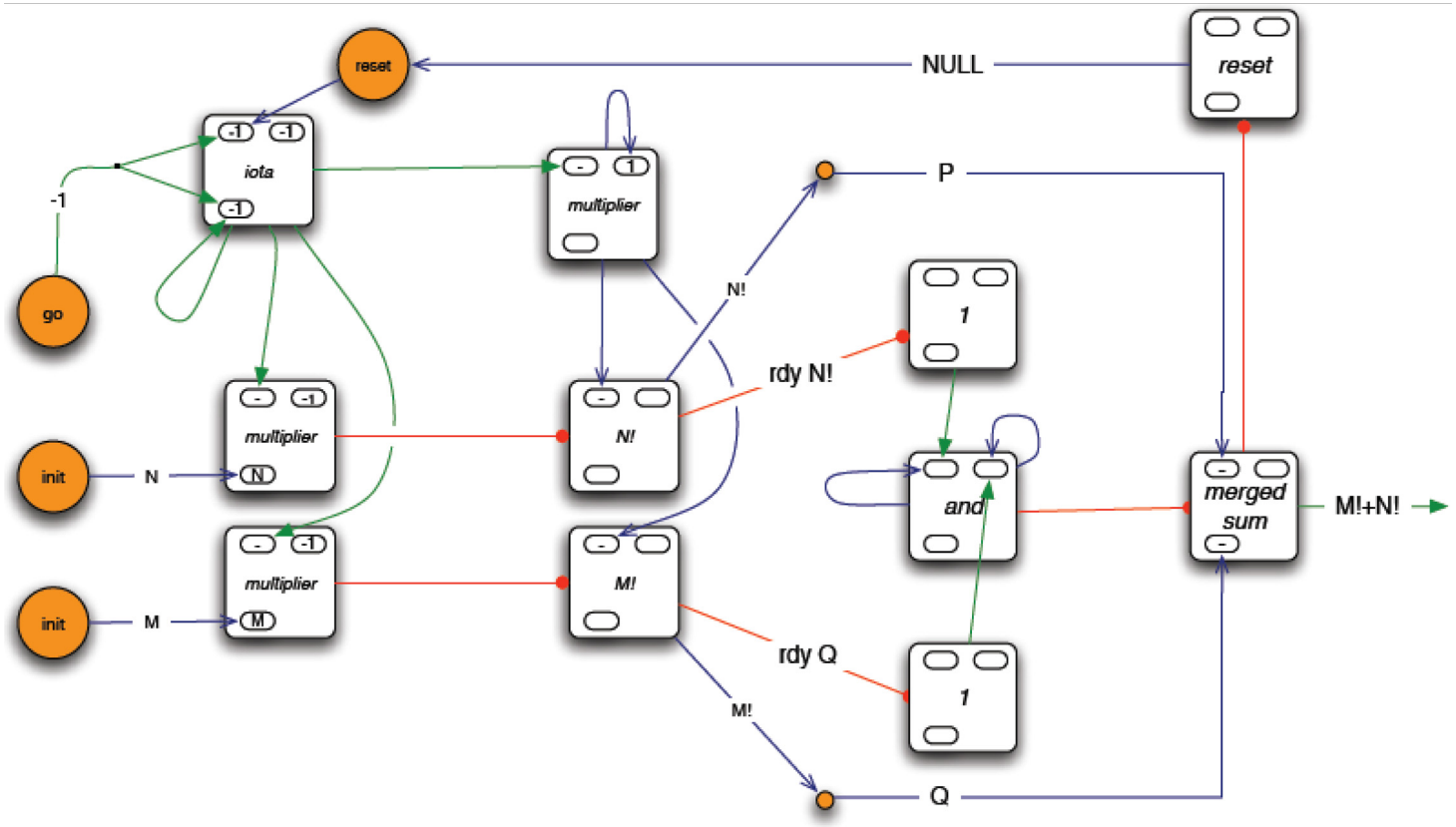
- Loops force pipeline data into blocks of a size that will fit inside the loop, this breaks the pipeline
  循环使流水线数据进入大小相同的块区，这种方式在循环内部是适宜的，但是却中断了流水线

Sum of Two Factorials

- Can you implement the sum of two factorials, $s = a! + b!$, in 11 machine instructions?
  你能用 11 条机器指令计算两个阶乘之和 s = a! +b! 吗？

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum of Two Factorials



Compilation

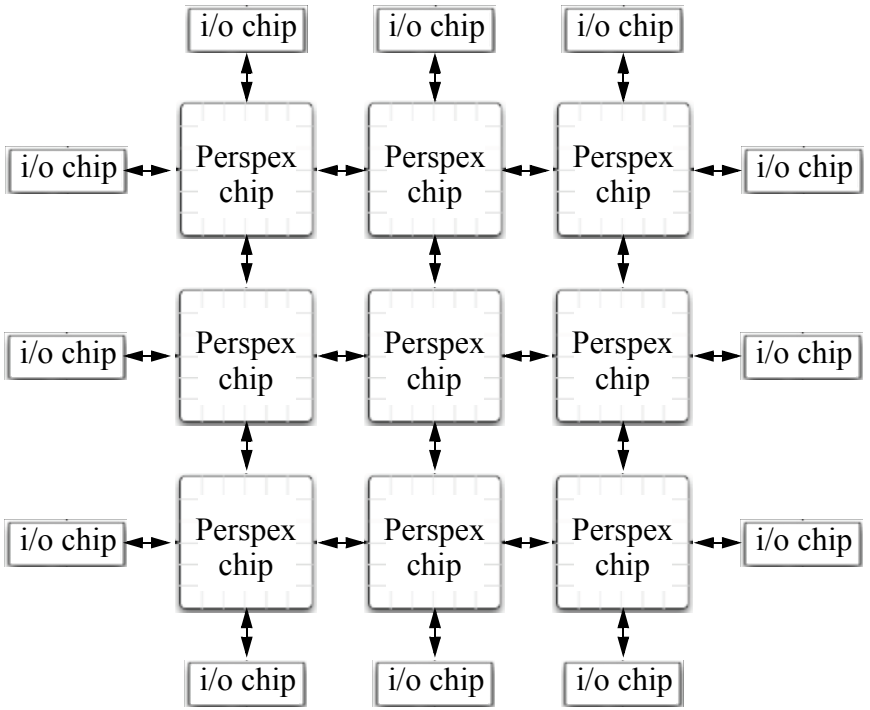
- General compilation is straightforward, but uses our supercomputer as a co-processor
  使用我们的超级计算机作为协同处理器进行常规编译是简单直接的
- Compilation via single assignment (functional programming) is attractive because of pipelining
  由于采用流水线技术，通过单赋值 （函数式编程）进行编译 很有吸引力
- Systolic programming is highly applicable
  脉动阵列式编程有很强的适用性
- Pipeline programming is highly applicable
  流水线式编程**也**有很强的适用性

I/O

- Perspex chips can be tiled together with one escalator input and one escalator output per edge of the chip, each running at 250 M Tokens Per Second
  有机玻璃芯片可以平铺排列在一起，芯片上每条边缘有一个 escalator 传送输入端和一个 escalator 传送输出端，以每秒传递 250 兆个令牌的速度运行

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

I/O



I/O

- Chips have an address horizon, not an address space, so arbitrarily many chips can be tiled together
  芯片上有一条地址水平线，而非地址空间，因此任意多个芯片可以平铺排列在一起

Supercomputer

The current specification of the chip has:
现行的芯片规格为:

- 10 k cores
  1 万个处理核
- Cores clocked at 250 MHz
  处理核时钟为 250 兆赫兹
- 4 input channels, and 4 output channels, each running at 250 M Tokens Per Second
  4 个输入通道，4 个输出通道，每一个通道的运行速度为每秒传递 250 兆个令牌

Supercomputer

- Power consumption of 10 kW per PFLOP
  每秒进行 1 千兆次的浮点运算时，功耗为 10 千瓦

Supercomputer

- We have implemented conventional programs
  我们运行了现有的程序
- We have implemented pipelined programs
  流水线式程序
- We have implemented systolic programs
  以及脉动阵列式程序

Supercomputer

- We get better pipeline latencies than other architectures
  相比其它结构，流水线式结构有更短的延迟时间
- We get better pipeline throughput than other architectures
  更快的运行速度
- We can sometimes match, but can never beat, systolic architectures
  在我们的实验中，流水线式结构有时在性能上可以和脉动阵列式结构相当，但从未占优

Supercomputer

- It is not practical to implement *programmable* systolic arrays in ASIC, but our chip is a viable alternative
  在 ASIC 电路中搭建可编程的脉动阵列并不实际，我们的芯片是一种可行的替代品
- Compiling by travel-time inequalities builds in some robustness to asynchronous operation of the array of processors
  应用传播时间的不等性进行编译增强了处理器阵列非同步运算的鲁棒性

Supercomputer

- Asynchronicity can be built into the array to smooth power usage
  在阵列中实施非同步性使电能使用更加平稳
- Our power consumption is about 1% of competing architectures
  我们的功耗大约是其它结构的百分之一

Transcomplex Numbers

The transcomplex numbers are still being developed so the following slides are subject to revision

由于超广义复数的理论仍然在发展中，因此下面的幻灯片也在修订中

Transcomplex Numbers

- The Cartesian complex numbers, $c = (x + iy)$, do not generalise well to transcomplex numbers, but the polar-complex numbers $c = (r, \theta)$ do generalise well
  笛卡尔坐标表示的复数形式 c = (x + iy) 不能很好地推广应用于超广义复数，但是极坐标表示的复数形式 c = (r, θ) 却可以

- Henceforth we say that C^T is the set of polar-transcomplex numbers
  这里，我们规定 C^T 为极坐标表示的超广义复数集

Transcomplex Numbers

- If desired, we may identify the Cartesian complex zero, $0 = (0 + i0)$, with all those polar-transcomplex zeros, $0 = (0, \theta)$, that have a real angle $\theta \in R$
  如果需要，我们可以认为笛卡尔坐标表示的复数零 $0 = (0 + i0)$ 等同于所有那些具有实数方位角 $\theta \in R$ 的极坐标表示的超广义复数零 $0 = (0, \theta)$

Transcomplex Numbers

- Regardless of whether or not we make this identification, polar-transcomplex arithmetic is a superset of polar-complex arithmetic, transreal arithmetic, and real arithmetic
不论我们是否这样认同，极坐标表示的超广义复数算术都是极坐标表示的复数算术、超广义实数算术和实数算术的超集

Transcomplex Numbers

- We have to make the identification at some point if polar-transcomplex arithmetic is to be a superset of Cartesian complex arithmetic
  如果要让极坐标表示的超广义复数算术成为笛卡尔坐标表示的复数算术的超集，我们必须在某些地方作这样的认同

Polar-Transcomplex Numbers

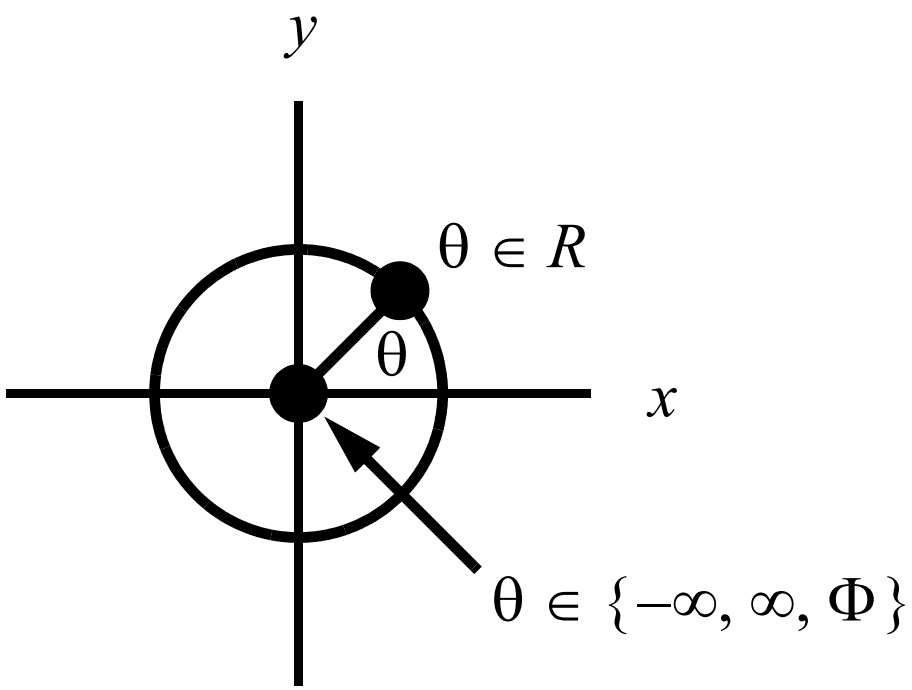
- The polar-transcomplex numbers are (d, θ) where d is a transreal displacement and θ is a transreal angle or the transreal argument of an angle
  极坐标表示的超广义复数为 (d, θ)，其中 d 是超广义实数表示的位移，θ 是超广义实数表示的方位角或者是一个方位角的超广义实数表示的辐角

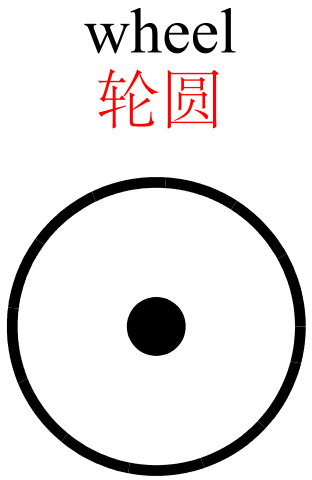
Polar-Transcomplex Numbers

- θ is bijective with the points on a wheel of unit radius when the non-finite angles are identified as $\theta_1 = \theta_2$ when $\theta_1, \theta_2 \in \{-\infty, \infty, \Phi\}$
  当角度为非有限值的角 $\theta_1 = \theta_2$ 时，其中 $\theta_1, \theta_2 \in \{-\infty, \infty, \Phi\}$，有 θ 与单位半径轮圆上的点是双射的

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Polar-Transcomplex Numbers

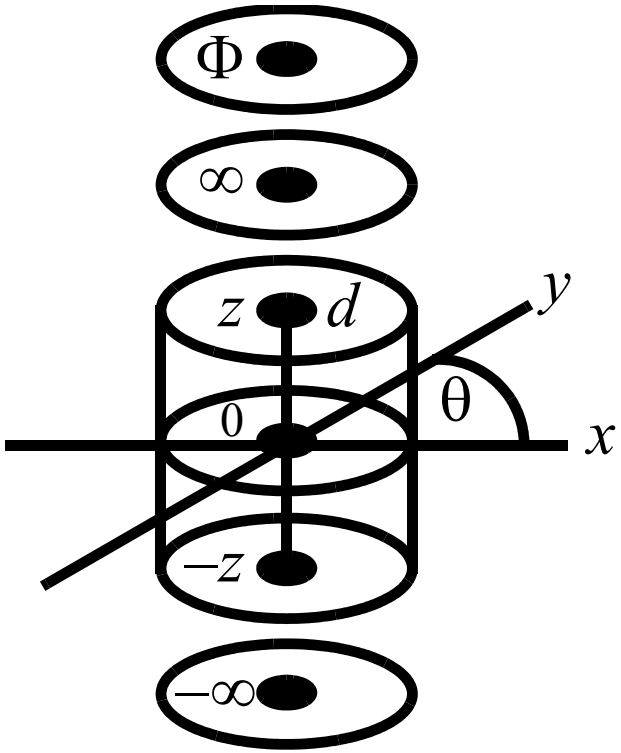




James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Polar-Transcomplex Numbers

- The wheel is swept along a transreal axis which is bijective with displacement, *d*
  轮圆沿着超广义实数的坐标轴进行扫描，这条轴与 d 是双射的



Polar-Transcomplex Numbers

English has many words for describing the parts of a wheel:
英语中有很多词用来形容车轮形状物体的各部分:

- We say that the wheel in the range $-\infty \leq d \leq \infty$ is a *wide wheel*
  当 $-\infty \leqslant d \leqslant \infty$ 时，称为宽轮

- We say that the wheels at $d = -\infty$ and $d = \infty$ are *side walls*
  当 $d = -\infty$ 和 $d = \infty$ 时，称为轮侧壁

- We say that the wheel at $d = \Phi$ is a *hub cap*
  当 $d = \Phi$ 时，称为轮毂帽

Polar-Transcomplex Numbers

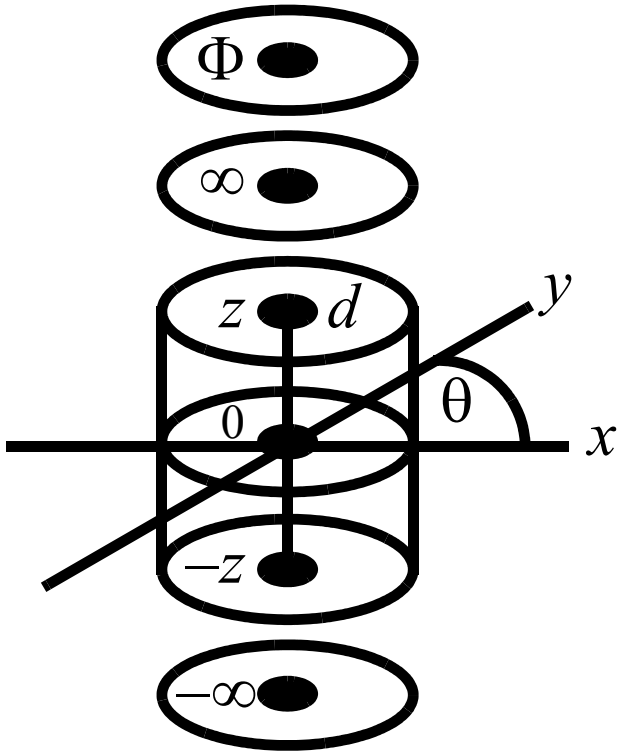
- Topologically, Φ lies outside the range $-\infty \leq d \leq \infty$ and may be drawn at an arbitrary position outside this range. Here the wheel at $d = \Phi$ is drawn above $d = \infty$ only because this placement makes it easier to describe the addition of polar-transcomplex numbers. It could be drawn elsewhere

从拓扑学观点出发，Φ 在范围 $-\infty \leqslant d \leqslant \infty$ 之外，也就是可以落在该范围之外的任意一个位置。这里将 $d = \Phi$ 时的轮子画在 $d = \infty$ 时的轮子之上，因为这样放置能够更容易地描述极坐标表示的超广义复数的加和。它完全可以画在其它位置上

Polar-Transcomplex Numbers

- We say that the axis of a wheel is an *axel*
  我们称轮圆的轴为轮轴
- We say that the centre of a wheel is a *hub*
  轮圆的中心为轮毂
- We say that the circumference of a wheel is a *rim*
  轮圆的圆周为轮辋
- We say that the whole wheel shown in the figure is a *swept wheel*
  图示的整套轮圆为扫描轮

Polar-Transcomplex Numbers



Polar-Transcomplex Numbers

- The swept wheel is a double cover of the polar-transcomplex numbers identified as
  $(d, \theta) = (-d, \theta + \pi)$
  扫描轮是极坐标表示的超广义复数的二重覆叠（double cover），可认为：(d, θ)= (-d, θ + π)

- If θ is the transreal argument of an angle then θ lies on a cylinder or its axis at nullity
  如果 θ 是一个方位角的超广义实数辐角，那么 θ 要么落在圆柱体的表面，要么落在零度数代表的中心轴上

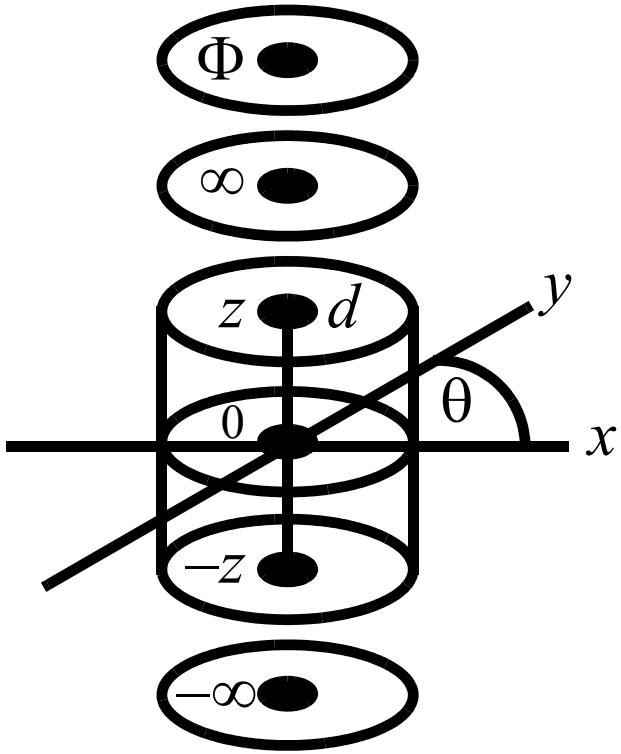
- We say that the cylinder is a *single-ply wheel*
  我们称这个圆柱体为单股轮

Polar-Transcomplex Numbers

- If θ is a transreal angle then θ lies on a Riemann surface or its axis at nullity
  如果 θ 是一个超广义实数方位角，那么 θ 要么落在黎曼曲面上，要么落在零度数代表的轴上

- We say that the Riemann surface is a *multi-ply wheel*
  我们称这个黎曼曲面为多股轮

Polar-Transcomplex Numbers



Polar-Transcomplex Numbers

- Because of the double cover, the argument of an angle is defined over half a rotation not a full rotation:
  因为二重覆叠，一个角的辐角可以只在半个圆上定义，而不是整个圆。

$$\arg\theta = \begin{cases} \pi/2 \ : \ \arcsin(\sin\theta) = -\pi/2 \\ \arcsin(\sin\theta) \ : \ \text{otherwise} \end{cases}$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Polar-Transcomplex Numbers

- The function $\text{can}(d, \theta) = (d', \theta')$ maps arbitrary, polar-transcomplex co-ordinates, (d, θ), onto (d', θ'), in canonical form, so as to fix the sense of the displacement

  按照数学中的标范形式，对于任意 d 和 θ，函数 can(d, θ) = (d’, θ’) 将极坐标表示的超广义复数坐标 (d, θ) 映射到 (d’, θ’)，从而使位移为正。

$$\text{can}(d, \theta) = (d', \theta')$$

$$d' = \begin{cases} -d : \cos\theta < 0 \text{ or } \arcsin(\sin\theta) = -\pi/2 \\ d : \text{otherwise} \end{cases}$$

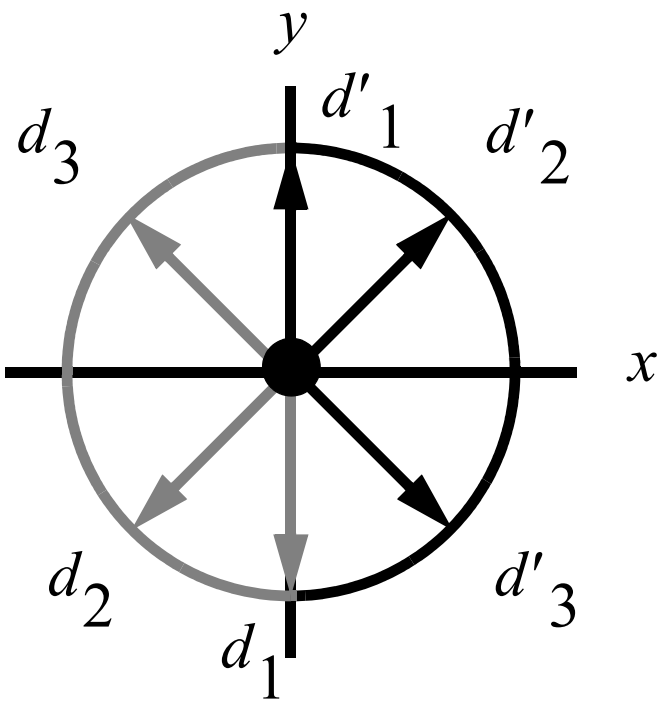
$$\theta' = \arg(\theta)$$

Polar-Transcomplex Numbers

- The canonical form is an involution of the grey parts of the rim of the wheel onto the solid black parts of the rim, with the hub of the wheel being mapped onto itself by identity
  按照标范形式，下图中灰色部分轮辋上的点会被对合到黑实线部分，而轮毂会被映射到它本身，即保持不变。

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Polar-Transcomplex Numbers



Polar-Transcomplex Numbers

Multiplication and division are defined as follows:
用极坐标表示的超广义复数辅角的乘法和除法

- $(d_1, \theta_1) \times (d_2, \theta_2) = (d_1 \times d_2, \theta_1 + \theta_2) : d_i, \theta_i \in R^T$

- $(d_1, \theta_1) \div (d_2, \theta_2) = (d_1 \div d_2, \theta_1 - \theta_2) : d_i, \theta_i \in R^T$

Polar-Transcomplex Numbers

- Note that non-finite, orthogonal transformations (rotations and reflections), O, do not have a multiplicative inverse, O^{-1}, but they do have a transpose, O^T, so the *contra rotation,* is O^T
注意，非有限的（超广义实数的）正交变换（旋转和反射）O 没有倒数 O^{-1}，但是有转置矩阵 O^T，因此反向旋转只能用 O^T 表示。

Polar-Transcomplex Numbers

- Addition and subtraction are obtained by ordinary vector addition in a plane which is subject to transformations in the swept wheel
  辅角的相加或相减由平面上的由有限值表示的向量（有限向量）相加得到。该平面取决于在扫描轮上的变换的情况。

- The parallelogram rule is preserved for non-finite vectors, which makes the sum of non-finite vectors generally non-associative
  平行四边形法则是针对有限向量定义的。由非有限值表示的向量 （非有限向量）组成的平行四边形不遵守结合律。

Polar-Transcomplex Numbers

$(d_1, \theta_1) + (d_2, \theta_2) = (d_3, \theta_3)$ may be computed as follows:
方程式 (d1, θ 1) + (d2, θ 2) = (d3, θ 3) 可以计算如下：

- Map the arguments (d_1, θ_1) and (d_2, θ_2) into canonical form via an involution
  辅角 (d1, θ 1) 和 (d2, θ 2) 会被对合为 标准形式

Polar-Transcomplex Numbers

- Map the transformed arguments by a dilatation into a common, general plane, for example, whichever argument plane is higher in the swept wheel model
  将变换完的辅角膨胀（dilatation）到一个更普通的平面上，例如，在扫描轮模型中哪个辅角平面高，就放到哪个平面上。

- Perform ordinary vector addition in the common plane with displacement, d_c, to obtain the resultant vector
  $v_r = (d_r, \theta_r)$
  在位移为 d_c 的一个普通的平面上进行有限向量的加合，从而得到合向量 $v_r = (d_r, \theta_r)$。

Polar-Transcomplex Numbers

- Map the resultant vector onto the swept wheel model by a dilatation $d_d = d_c \times |d_r|$
  将合向量映射到位移为 $d_d = d_c \times |d_r|$ 的扫描轮模型上。

- Map the dilatated vector at displacement d_d into canonical form as (d_3, θ_3)
  将膨胀的向量映射到位移为 d_d 的 标准形式，即 (d_3, θ_3)。

- This completes the addition
  从而完成了向量的加和

Polar-Transcomplex Numbers

- Subtraction is defined as follows:
  辅角的减法

$$(d_1, \theta_1) - (d_2, \theta_2) = (d_1, \theta_1) + (-d_2, \theta_2)$$

Polar-Transcomplex Numbers

- All polar-transcomplex functions $C^T \to C^T$ are mappings on the swept wheel

  所有用极坐标表示的超广义复数的函数 $\mathrm{C}^{\mathrm{T}} \to \mathrm{C}^{\mathrm{T}}$ 都会映射到扫描轮上

- Specifically, they are transformations or projections of the wheel

  特别是，它们是轮圆的变换或投影

Polar-Transcomplex Numbers

- The complex logarithm generalises easily
  复数对数可以很容易地被扩展
- The complex exponential is the inverse mapping of the complex logarithm
  复数指数是复数对数的逆映射
- In particular, all complex trigonometric functions are loci in the swept wheel
  特别是，所有的复数三角函数都是扫描轮上的轨迹

Polar-Transcomplex Numbers

- There are no undefined or indefinite values in polar-transcomplex analysis because polar-transcomplex arithmetic is total
  在极坐标表示的超广义复数的微积分中，没有无定义或无确定值的数，因为极坐标表示的超广义复数算术是无需限制条件，对一切都适用的。

Polar-Transcomplex Numbers

- Therefore, every real or complex equation that is used in mathematical physics, and is extended to a transreal or transcomplex equation, is mathematically well defined everywhere, including at singularities (but it might not be physically well defined)
因此，用在数学物理学中的每一个实数方程或复数方程都能被扩展为超广义实数方程或超广义复数方程。这些超广义实数或复数方程 (包括奇点在内的各个部分) 都在数学上被很好的定义了，虽然不一定在真实的世界里有意义。

Powers of Minus Unity

$$-1^{\Phi} = e^{\ln -1^{\Phi}} = e^{\Phi \ln -1} = e^{(\Phi, 0)(0, \pi)}$$

- $$= e^{(\Phi \times 0, 0 + \pi)} = e^{(\Phi, \pi)} = (\Phi, \pi) = -\Phi$$

$$= \Phi$$

$$-1^{\infty} = e^{\ln -1^{\infty}} = e^{\infty \ln -1} = e^{(\infty, 0)(0, \pi)}$$

- $$= e^{(\infty \times 0, 0 + \pi)} = e^{(\Phi, \pi)} = (\Phi, \pi) = -\Phi$$

$$= \Phi$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Powers of Minus Unity

$$-1^{-\infty} = e^{\ln -1^{-\infty}} = e^{-\infty \ln -1} = e^{(-\infty, 0)(0, \pi)}$$

- $$= e^{(-\infty \times 0, 0 + \pi)} = e^{(\Phi, \pi)} = (\Phi, \pi) = -\Phi$$

$$= \Phi$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum to Infinity

We define that every transreal or transcomplex power series has a sum to infinity which is the limit, if the limit exists, or else is the formal sum to infinity including the infinity term

我们定义，每一个超广义实数或复数的幂级数，例如 x^n，如果有上限存在，用无限项之和表示，即 $\lim_{n \to \infty} \sum_{n=1} x^n$，否则，用无限项之和加上 n 为无穷大的项来表示，即 $\sum_{n=1}^{n \to \infty} x^n + x^\infty$。

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum to Infinity

We wish to find the sum to infinity, S_∞, of a power series that generates the terms 1 – 1 + 1 – 1 + …

我们希望得到能产生 1-1+1-1+ … 的幂级数的无穷多个项 （即包括 n = ∞）的加和 S_∞

In general, 即

$$S_\infty = a_0 + \left(\sum_{n=1}^{n \to \infty} a_n \varepsilon_n x^n \right) + a_\infty \varepsilon_\infty x^\infty$$

Note that the middle term ranges over all integers $n > 0$

注意，中间的表达式对任何 n > 0 的整数都适用。

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum to Infinity

But $0 \times \infty = \Phi$ so we cannot always use a coefficient $a_n = 0$ to elide a term from a power series, instead we define an explicit elision operator, ε_n, such that:

但是，由于 $0 \times \infty = \Phi$, 我们不能简单地通过使系数 $a_n = 0$ 来削去上式中间的项。对此，我们特别定义了一个删除符 ε_n，使得：

$$a_n \varepsilon_n x^n = \begin{cases} 0 : \varepsilon_n = 0 \\ a_n x^n : \varepsilon_n = 1 \end{cases}$$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum to Infinity

Let, 设,

$$S_{\infty} = 1 + \left(\sum_{n=1}^{n \to \infty} 1(-1)^{n} \right) + 1\varepsilon_{\infty}(-1)^{\infty}$$

With $\varepsilon_{\infty} = 0$ this generates the terms

$1 - 1 + 1 - 1 + \ldots$, but then S_{∞} does not converge to a limit so we take $\varepsilon_{\infty} = 1$

当 $\varepsilon_{\infty} = 0$ 时，S_{∞} 为 1-1+1-1+…，但是，因此时 S_{∞} 不能收敛成有限项，我们于是使 $\varepsilon_{\infty} = 1$

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Sum to Infinity

$$S_{\infty} = 1 + \left(\sum_{n=1}^{n \to \infty} -1^{n} \right) + (-1)^{\infty}$$

$$= -1^{\infty} + 1 + \sum_{n=1}^{n \to \infty} -1^{n}$$

$$= \Phi + 1 + \sum_{n=1}^{n \to \infty} -1^{n}$$

$$= \Phi$$

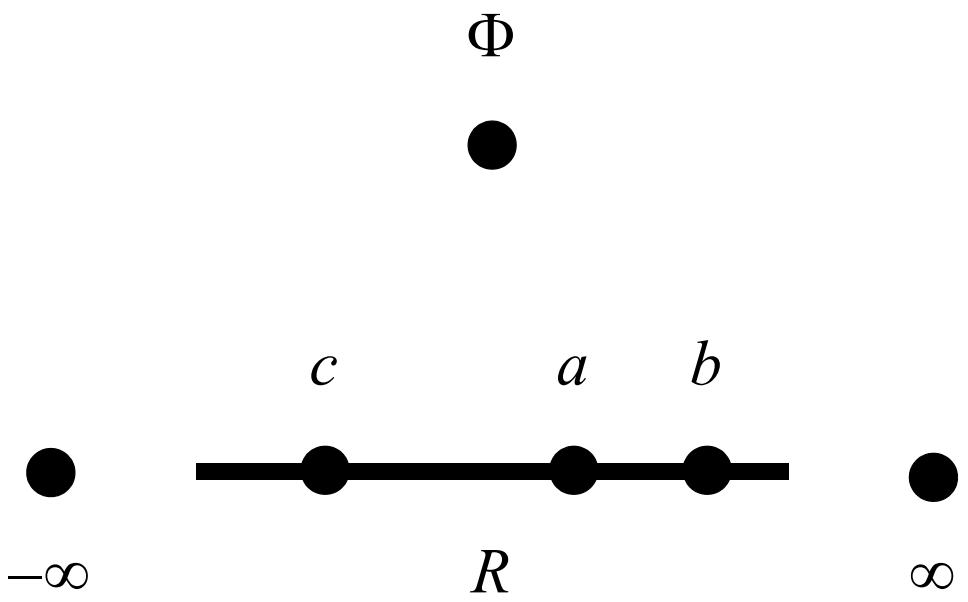
Sum to Infinity

- Every power series has a sum to infinity
  每一个幂级数都可以是无穷多个项 （即包括 n = ∞）的加和

- Therefore, every power series used in mathematical physics is mathematically well defined (but might not be physically well defined)
  因此，用在数学物理学中的每一个幂级数都在数学上被很好的定义了，虽然不一定在真实的世界里有意义。

Physics

We need to link transreal mathematics to physics
我们需要把超广义实数数学和物理学联系起来。



Physics

- Any real-numbered, physical quantity, a, can move continuously to a physical quantity, b, that lies between a and ∞
  任何用超广义实数表示的物理量 a 都可以连续不断地移动到物理量 b，其中 b 在 a 和 ∞ 之间

- Similarly, any real-numbered, physical quantity, a, can move continuously to a physical quantity, c, that lies between a and $-\infty$
  相似地，任何用超广义实数表示的物理量 a 也都可以连续不断地移动到物理量 c，其中 c 在 a 和 - ∞ 之间

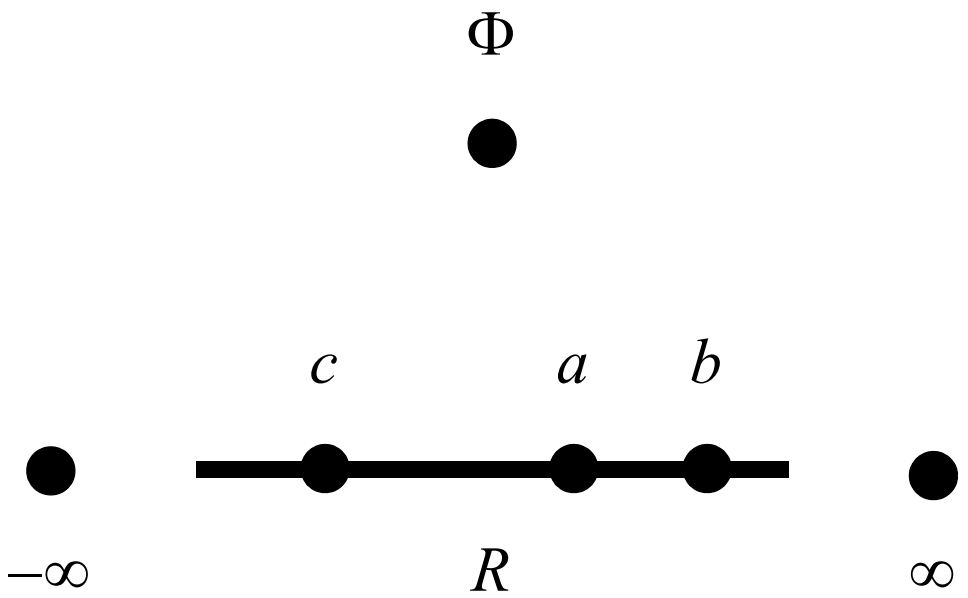
Physics

- In particular, *a* can feel infinite positive and negative forces that cause it to move classically
  特别是，a 受到来自正负两个方向的力作用，会以经典物理学的方式移动。该力的大小可以为有限（即用普通实数表示），或无限 （即用无穷大表示）。

- I hypothesise that quantum physics works similarly
  我觉得量子物理学也类似

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

Physics



Physics

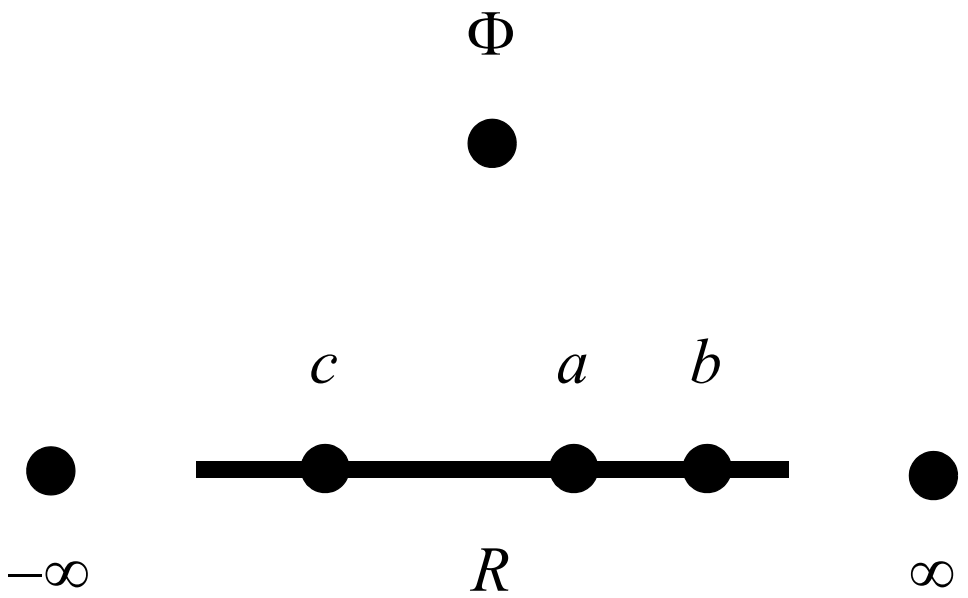
- There is no quantity between Φ and any real-numbered or transreal-numbered quantity, a, so there is no continuous (classical) motion between a and Φ
  由于没有任何物理量位于 Φ 和用实数或超广义实数表示的物理量 a 之间，在 a 和 Φ 之间没有任何经典物理学方式的连续移动可以发生

- In particular, a cannot feel a nullity force that causes it to move classically
  特别是，a 不能受到来自零度数方向的力，因此无法以经典物理学的方式移动

- I hypothesise that quantum physics works similarly
  我觉得量子物理学也类似

James A.D.W. Anderson
CNIC - Chinese Academy of Sciences
2009

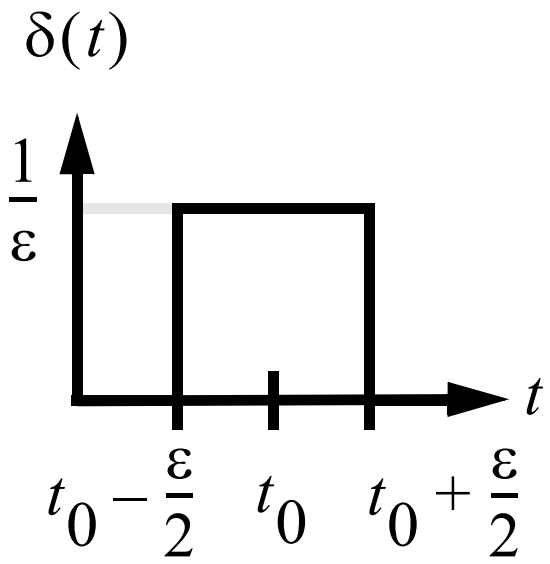
Physics



Physics

The Dirac Delta is used to transfer an action asymptotically quickly, but not instantaneously

狄拉克函数（单位脉冲函数）以极其接近于零延时（但非零延时）的方式传递动能



Physics

- Feynman wanted to exempt a moving electron from acting on itself via the electric field, but couldn't
（二十世纪美国物理学家）理查德 · 费曼曾试图通过电场的作用来除去一个运动中的电子，但他没有做到

- If we assume that the interaction of the electric field with a distant electron is asymptotically fast, because the field perturbation has to cross the electron, then the Dirac Delta transfers the action between the electric field and the distant electron in the usual way
如果我们假设电场以极其接近于零延时的方式作用于上述电子，由于场微扰会通过该电子，狄拉克函数会以它典型的方式在电场和电子之间传递作用力

Physics

- If we assume that the interaction of the electric field with an electron itself is instantaneous, because the field perturbation does not have to cross the electron, then the Dirac Delta collapses to a Box function with an area of nullity so, by our linking hypothesis, the electron does not feel a force from the electric field, nor does the electric field feel a force from the electron
如果我们假设在电场和电子之间的相互作用是在瞬间发生的（零延时），由于场微扰不一定通过该电子，狄拉克函数成了框面积为零度数的逻辑框函数，由我们之前的假设，我们可以推理得出，电子不会受到电场的作用力，电场也不会受到电子的作用力。

Physics

- When epsilon is exactly zero, $\varepsilon = 0$, we have width exactly zero, $w = \varepsilon = 0$, and height exactly infinity, $h = 1/\varepsilon = 1/0 = \infty$. Whence the area, a, is exactly nullity, $a = w \times h = 0 \times \infty = (0/1) \times (1/0) = (0 \times 1)/(1 \times 0) = 0/0 = \Phi = 0/0 = \varepsilon/\varepsilon \neq 1$
当 ε 确定等于零时，即 ε =0，逻辑框的宽度等于零，即 w= ε =0，框高度等于无穷大，即 h=1/ε =1/0=∞ . 因此，框面积 a 等于 零度数，即 a=…

Physics

- This gives Feynman the behaviour he wanted, but it does it by assuming that the universe operates according to transreal arithmetic not real arithmetic
这使得费曼的设想变得完全可行，当然这需要以超广义实数算术作为背景，而不是传统的实数算术。

- If the universe operates according to ordinary arithmetic then there must be a physical censor that outlaws division by zero and all of its infinitely many mathematical consequences
如果以传统算术为背景，我们则需要专门找一个人来阻止所有除以零的运算，以避免所有相关的数学结果。

Physics

- If the universe operates according to transarithmetic then it can divide by zero
  如果以超广义算术作为背景，则可以进行除以零的运算

- As transarithmetic supports an infinitely simpler explanation of the universe than ordinary arithmetic, Occam's razor says that we should abandon ordinary arithmetic as a model of physics and use transarithmetic instead
  与传统算数相比，超广义算术可以更简单，清楚地解释世界。因此根据奥卡姆剃刀理论，我们应该选择超广义算术，而非传统算术，作为解读真实世界的物理模型。

Offer

- We plan to sell supercomputers at 5 M US Dollars (USD) per PFLOP, three years after funding a supercomputing company
  在持续三年对一个超级计算机工司投资之后，我们计划以每 PFLOP 500 万美元的价格出售我们的超级计算机 （PFLOP，即每秒进行 1 千兆次的浮点运算，为一个运算单元。我们以 PFLOP 作为销售单位，而非每台超级计算机器，因为一台计算机可以有多个 PFLOP，例如 2×PFLOP，则价格为 1000 万美元。当然通常每台计算机都是 1 PFLOP。）

Offer

- We offer a discount of two, 1 PFLOP machines for 5 M USD for anyone willing to make staged payments of one third on contract, one third on tape out, and one third on delivery
  对卖出的每一 PFLOP，我们会加送一 PFLOP。另外，我们也接受分期付款，即在合同阶段付 1/3，送交制造期间付 1/3，交货的时候付 1/3。

- We plan to accept discounted orders for 5 to 10 PFLOPs
  如果能买 5-10 PFLOP, 我们还会考虑给额外的折扣

Conclusion

- Transreal arithmetic is a superset of real arithmetic
  超广义实数算术是传统实数算数的超集

- Polar-transcomplex arithmetic is a superset of polar-complex arithmetic, transreal arithmetic and real arithmetic
  极坐标表示的超广义复数算术是极坐标表示的传统复数算术的超集

Conclusion

- If polar zeros are identified with the Cartesian zero then polar-complex arithmetic is a superset of Cartesian complex arithmetic
  设极零等同于笛卡尔零，则有极坐标表示的超广义复数算术是笛卡尔复数算术的超集

- Transmetric spaces are a superset of metric spaces
  超广义度量空间是传统度量空间的超集

Conclusion

- There are no undefined or indefinite values in transreal arithmetic nor in any of its mathematical consequences
在超广义实数算术及其一切数学结果中，不存在无定义或无确定值的数

- In particular, every equation in mathematical physics has a solution, even at singularities, though the mathematical solution might not be physically valid
特别是，数学物理学中的每一个方程（包括在奇点上）都有解，虽然数学中的解在真实世界里不一定有意义

- In particular, every power series has a sum to infinity
而且，每一个幂级数都可以是不穷多个项 (**即包括** $n = \infty$) 的加和

Conclusion

- Transreal arithmetic makes conventional computers more efficient
  超广义实数算术使传统计算机更高效
- Transreal arithmetic makes supercomputers vastly more efficient
  超广义实数算术使超级计算机效率显著增加
- Transreal arithmetic can be taught to 12 year olds
  超广义实数算术理论 12 岁的孩子都能学会
- Do you want to buy a PFLOP computer for your child?
  你想给自己的孩子买一台 PFLOP 计算机吗

Moral

- Do you want China to be a follower or a leader?
  你希望中国是世界科技的领导者，还是跟在别人后面