Computer System Support for Scientific and Engineering Computation

Lecture 24 - July 21, 1988 (notes revised June 14, 1990)

1 Exceptions

The IEEE standard specifies 5 classes of exceptions: Inexact, Underflow, Overflow, Divide by Zero and Invalid. The inexact exception is the one exception on this list that is peculiar to the IEEE standard, the others exist (in possibly modified form) on at least some commercially available hardware. The inexact exception can be understood by considering old mechanical hand calculators. Those machines never discarded any digits: the user had to explicitly reenter a rounded quantity. The inexact exception corresponds to a human operator noticing when he discards digits.

The underflow exception occurs when quantities fall below the smallest positive normalized representable magnitude. When floating point numbers are represented with base β, precision p and exponents between $e_{\max}$ and $e_{\min}$, then the smallest normalized numbers are separated by a spacing of $\beta^{e_{\min}-p}$, but the spacing between 0 and the smallest positive normalized representable number is $\beta^{e_{\min}}$. So there is a gap around 0 which is β^p times larger than the spacing between normalized numbers. On machines which do not have denormalized numbers, when a quantity falls into this gap it is flushed to zero. On a machine with IEEE arithmetic, the quantity is rounded to the nearest denormal. An underflow exception occurs when a number falls into the gap and can not be represented exactly by a denormalized number. Cray's and Cyber's do not have an underflow exception, and although VAX's and IBM/370's do have an underflow exception, it is usually masked off.

Overflow is an attempt to create a number bigger than the biggest finite representable number $\beta^{e_{\max}}$, and on some machines defaults to the biggest number with the right sign. In IEEE arithmetic, this only occurs when the rounding mode is set to "round to zero", otherwise overflow defaults to $\pm\infty$ and signals an inexact exception as well as overflow. Invalid represents an invalid operation. Machines differ on what value is produced as the result of an invalid operation. APL sets $0/0 = 1$, IBM sets $0/0 = 0$, and the IEEE standard sets $0/0 = \text{NaN}$.

2 Singularities

Floating point calculations approximate mathematical functions, which are almost always piecewise analytic. Imagine carving the plane into regions. Then a piecewise analytic

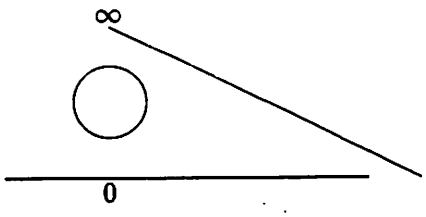

Figure 1: Stereographic projection.

function is analytic (representable by a power series) inside each region, but may have singularities on the boundary of the region. The function $f(x) = 1/x$ is analytic on $\{x : x > 0\}$ and $\{x : x < 0\}$ but has a singularity at the boundary $x = 0$. Some singularities are removable, that is, are only an artifact of the formula used to define the function. For example, $g(x) = \sin(x)/x$ appears to have a 0/0-singularity at 0, since $\sin(0) = 0$, but in fact setting $g(0) = 1$ defines a prefectly reasonable analytic function. Another example is $g(x) = (x^3 - y^3)/(x - y)$, whose singularity when $x = y$ can be removed by writing $g(x) = x^2 + xy + y^2$. The method of presubstitution can be used when computing functions with a removable singularity. When $f(x) = \sin(x)/x$, we presubstitute 1 for 0/0. Then if the computation happens to evaluate $f(0)$, it will be correctly computed as 1. On machines with IEEE arithmetic, the default is to compute 0/0 as a NaN. The hardware will have the constant NaN stored somewhere, and when computing `X = 0/0`, copy the NaN into `X`. In presubstitution, the programmer specifies what constant to use in place of NaN. This has the advantage that in pipelined floating point hardware, precise interrupts are not necessary. That is, if highly pipelined machines would offer presubstition to the progammer, they could handle removable singularities with no performance penalty and not have to build extra hardware to support precise interrupts.

The function $f(x) = 1/x$ does not have a removable singularity when considering only ordinary finite real numbers. The IEEE standard introduces the symbols $\pm\infty$ to deal with this situation. To understand these symbols better, it helps to introduce the closed line. There are two ways to close the line. The simplest is the 1-point closure. To explain this closure, consider the mapping between the line and the circle, illustrated in Figure 1. To map a point y on the line to a point Y on the circle, draw a line through y and the top of the circle. This line will intersect the circle at exactly one point Y. The map takes y to Y. This maps every point of the line to a point of the circle, and vice-versa with one exception. The top of the circle isn't mapped to any finite point of the line. This is the point that corresponds to $\pm\infty$.

Another way to think of the mapping between the line and the circle is via the formula $\theta = 2\arctan(x)$. If $-\pi \le \theta \le \pi$ represents the circle with $-\pi$ and $+\pi$ representing the same point, then this formula maps each point θ of the circle to a point x of the line. The points $\pm\infty$ of the line map to $\theta = \pm\pi$. Why introduce the 1-point closure? It makes all rational functions continuous. Consider $f(x) = 1/x$. Normally, we think of it as having a singularity at ∞. But if we think of it as a function on the circle, it is perfectly continuous. On the real line, it maps 0 to ∞. On the circle, it maps $\theta = 0$ to $\theta = \pi$. In a formula, $f(x) = 1/x$ mapping the line to the line gets transformed to $g(\theta) = 2\arctan(1/\tan(\theta/2))$, mapping the circle to the circle. And this formula simplifies to $g(\theta) = \pi - \theta$, which is a continuous function of θ mod 2π. To summarize, the 1-point closure introduces the new symbol ∞, and it makes all rational functions continuous.

For transcendental functions, we need to use the 2-point closure, which introduces two new symbols, $+\infty$ and $-\infty$. While the 1-point closure can be thought of as the circle, the 2-point closure can be thought of as the interval [-1, 1]. A formula that maps the line to the 2-point closure is $(2/\pi)\arctan(x)$). Only in this case, -1 and 1 are completely different points. The transcendental function $f(x) = e^x$ is not continuous in the 1-point closure, since $f(+\infty) = +\infty$ but $f(-\infty) = 0$, but it is continuous in the 2-point closure.

3 Signed 0

We would like to have the relation

$$1/(1/x) = x \tag{1}$$

hold true for all numbers, at least approximately. To have this relation hold true when $x = -\infty$, we must be able to distinguish $+0$ from -0. We also would like to know that

$$y + x = x \;\Rightarrow\; y = 0 \tag{2}$$

Since the left hand side is true for both $y = +0$ and $y = -0$, we must have $+0 = -0$. Finally we would like to know that

$$x = y \;\Rightarrow\; \frac{1}{x} = \frac{1}{y}, \tag{3}$$

at least approximately. Unfortunately, this doesn't hold true for $x = +0$, $y = -0$ in the 2-point closure of the reals. The problem is that we can't have (1), (2) and (3) all hold simultaneously. In the IEEE standard, it is equation (3) that is violated.

Although signed zero may appear to be mostly a nuisance, there is one situation where it is very helpful. That situation is complex arithmetic. To take a simple example, consider the equation $\sqrt{1/z} = 1/\sqrt{z}$. This is certainly true when $z \geq 0$. What about complex values of z? If $z = -1$, then we might naively compute $\sqrt{1/-1} = \sqrt{-1} = i$ and $1/\sqrt{-1} = 1/i = -i$. Thus $\sqrt{1/z} \neq 1/\sqrt{z}$! This is what happens on machines that either do not have signed 0, or if they have it, do not do arithmetic consistently with it. However, on IEEE machines, $\sqrt{1/z} = 1/\sqrt{z}$ even when $z = -1$. The reason is signed 0. To see why requires a short digression.

If numbers are represented in polar coordinates $z = re^{i\theta}$, where θ is a number modulo 2π, then $w = \sqrt{z} = \sqrt{r}e^{i\theta/2}$, or $w = se^{i\phi}$ with $s = \sqrt{r}$ and $\phi = \theta/2$. As θ goes from 0 to π, then ϕ goes from 0 to $\pi/2$ continuously. Similarly, as θ goes from 0 to $-\pi$, then ϕ ranges from 0 to $-\pi/2$. Thus in the range $-\pi < \theta < \pi$, $\sqrt{z}$ is continuous. But at $\theta = \pi \equiv -\pi \pmod{2\pi}$, there is a discontinuity. As $\theta < \pi$ approaches π, w approaches $se^{i\pi/2} = si$ while as $\theta > -\pi$ approaches $-\pi$, w approaches $se^{-i\pi/2} = -si$. This discontinuity occurs because $\sqrt{\ }$ is multi-valued, that is, each number has two square roots. The negative real axis $z < 0$ is a branch cut. If we consider the complex plane minus the branch cut, we can define $\sqrt{\ }$ as a continuous single-valued function. On the branch cut, $\sqrt{\ }$ has two values. In IEEE arithmetic, a number on the negative real axis is of the form $-x + i0$, where $x > 0$ and 0 is a signed 0. Thus it is natural to define $\sqrt{-x + i(+0)} = \lim_{y\to 0+} \sqrt{-x + iy} = i\sqrt{x}$ and $\sqrt{-x + i(-0)} = \lim_{y\to 0-} \sqrt{-x + iy} = -i\sqrt{x}$, and in fact, that natural formulas for compute $\sqrt{\ }$ will compute in just this way.

Back to $\sqrt{1/z} = 1/\sqrt{z}$. If $z = -1 = -1 + i0$, then $1/z = -1 + i(-0)$ and $\sqrt{1/z} = \sqrt{-1 + i(-0)} = -i$, while $1/\sqrt{z} = 1/i = -i$. So IEEE arithmetic preserves this identity for

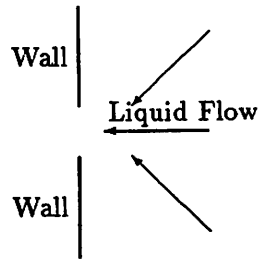

Figure 2: Conformal maps of slitted domains.

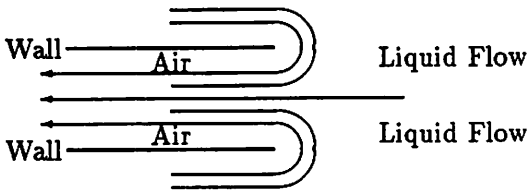

Figure 3: Liquid flow through a slot.

all z. To give another more realistic example, consider the conformal map (see Figure 2) $\zeta = f(z) = 1 + z^2 + z\sqrt{1+z^2} + \ln(z^2 + z\sqrt{1+z^2})$. The image of $\{z : \Re(z) > 0\}$ is the region wetted by a liquid that is being forced by high pressure to jet into a slot (see Figure 3). The line $\{z = iy : y > 1\}$ gets mapped to the line $\{\zeta = \xi + i\pi : \xi < 0\}$ and the line segment $\{z = iy : 0 < y < 1\}$ gets mapped to the curve connecting 0 to $-\infty + i\pi/2$. Since $f(z) = f(\bar{z})$, the graph is symmetrical about the real line. On an IEEE machine, everything goes well. But on a non-IEEE machine, the line $\{z = iy : y < -1\}$ does not get mapped to $\{\zeta = \xi - i\pi : \xi < 0\}$, but rather to $\{\zeta = \xi + i\pi : \xi < 0\}$; hence, part of the slot's boundary goes astray on a machine that lacks a proper signed zero.

4 Is 0^0 Exceptional?

It is not always obvious whether an operation should cause an exception. In this section, we argue that 0^0 is not exceptional, but rather should be equal to 1. What is important is not so much whether you believe this argument, but rather that it indicates the need for retrospective diagnostics, which will be discussed in the next lecture.

When establishing the value of mathematical functions like x^y, we would like to employ the principle of *parsimony*, that is, derive the value of the function from the fewest possible rules. The traditional rule for x^y goes back to Descartes: $c^n = c \cdot c \cdots c$ where there are n c's on the right hand side. In order to extend this to other values of n, we can use the traditional rules

$$c^1 = c \tag{4}$$

$$c^{m+n} = c^m c^n \tag{5}$$

The first rule is necessary because without it we could define $c^n \equiv c^{\alpha n}$, where the right hand side is ordinary exponentiation. Given these rules for $n > 0$. it is natural to apply

them for all n and thus extend the definition of c^n. Letting $m = 0$ in (5) $c^n = c^0 c^n$, so if $c \neq 0$ then we must have $c^0 = 1$.

These rules do not give us any information about the special values 0^{-x}, 0^0, ∞^{-x} and ∞^0. So we might try another set of rules.

$$c^0 = 1 \tag{6}$$
$$c^{i+1} = c^i c \tag{7}$$

These rules have appeared in textbooks, for example L. E. Sigler's *Algebra* published by Springer-Verlag. They appear to be as parsimonious as the first set, and they also are upward compatible with them. However they additionally define the cases that were ambiguous previously: $0^0 = 1$ (from (6)), $0^{-n} = 1/0^n = \infty$ (apply (7) n times), $\infty^0 = 1$ (since (6) is true for all c), $\infty^{-x} = 1/\infty^x = 0$. Any rule that gave a different value for 0^0 would have to be more complex than these simple rules. Another reason why we would like to have $0^0 = 1$ comes from the identity

$$a_0 + a_1 x^1 + \cdots + a_n x^n = \sum_0^n a_j x^j \ .$$

When $x = 0$ this says $a_0 = a_0 0^0$which forces $0^0 = 1$.

Since $c^0 = 1$ is independent of c, even $\mathrm{NaN}^0 = 1$. Although it may bother you that an expression involving a NaN produces a finite number, this is less strange than a function that is independent of an argument suddenly taking on a different value when that argument is a NaN. There are other examples where NaNs can disappear. The code fragment `if (y=0 or x/y < 3) then z=15` should set $z = 15$ when $x = y = 0$. Thus the 0/0 Nan disappears. If NaN's could never disappear, then there would be no point in generating them.