نشر ریاضی، سال ۱۰، شمارهٔ ۱، صص ۶-۱۱

حساب دقیق حقیقی چیست؟

عباس عدالت*

امروزه تقریباً در همهٔ زبانهای برنامه‌نویسی از نمایش اعداد با ممیز شناور برای محاسبات عددی استفاده می‌کنند. در این نمایش، برای نمونه در پایهٔ ۲، اعداد به‌صورت $\pm m \times 2^e$ نوشته می‌شوند که m مانتیس عدد است و با یک رقم صحیح و تعداد معینی رقم پس از ممیز مشخص می‌شود؛ e توان عدد است که متشکل از تعداد معینی ارقام صحیح است. برای نمونه در دقت ساده[1] مجموعاً ۳۲ بیت برای مشخص کردن علامت ($\pm$) و ارقام موجود در m و e به‌کار گرفته می‌شوند. در نتیجه، مجموعهٔ اعدادی که در نمایش ممیز شناور از آنها استفاده می‌شود یک مجموعهٔ متناهی است و هر عدد حقیقی را باید با یکی از این اعداد تقریب زد. در اثر هر یک از این تقریبها، خطای محاسباتی جدیدی پدیدار می‌شود که امکان دارد این خطاها روی هم انباشته شده و دست آخر موجب خطای بسیار بزرگی در کل محاسبه گردند.

در اینجا برای نمونه، دو مثال از خطاهای محاسباتی ناشی از نمایش با ممیز شناور می‌آوریم.

(۱) دنبالهٔ زیر را در نظر بگیرید

$$a_0 = 2$$
$$a_1 = -4$$
$$a_{n+1} = 111 - \frac{1130}{a_n} + \frac{3000}{a_n a_{n-1}}$$

به‌سادگی می‌توان نشان داد که $\lim_{n\to\infty} a_n = 6$. لیکن اگر a_n را در هر نوع سیستم مبتنی بر ممیز شناور (دقت ساده، دقت مضاعف[2] یا هر دقت بالاتر) به ازای مقادیر مختلف n محاسبه کنیم، به‌نظر خواهد رسید که $\lim_{n\to\infty} a_n = 100$. در نتیجه، خطای محاسباتی نه تنها ناچیز نیست بلکه با بالا بردن دقت ممیز شناور هم کاهش نمی‌یابد.

(۲) اکنون دنبالهٔ زیر را در نظر بگیرید

$$b_0 = 1.5100050721318$$
$$b_{n+1} = \frac{3b_n^4 - 20b_n^3 + 35b_n^2 - 24}{4b_n^3 - 30b_n^2 + 70b_n - 50}$$

1. single precision 2. double precision

بسته به اینکه محاسبهٔ b_n با کدام کامپیوتر و زبان برنامه‌نویسی انجام شود به‌نظر خواهد رسید که $\lim_{n\to\infty} b_n = 1, 2, 3, 4$.

در دههٔ اخیر چند پروژهٔ تحقیقاتی دربارهٔ حساب دقیق حقیقی انجام یافته است. اگر بخواهیم محاسبات عددی را بدون خطا اجرا کنیم ناگزیریم نمایشی نامتناهی از اعداد حقیقی را به‌کار گیریم، زیرا برای نمونه بدیهی است که $\sqrt{2}$ را نمی‌توان به صورت متناهی نمایش داد. مثلاً می‌توانیم یک عدد حقیقی را به صورت حد یک دنبالهٔ کوشی از اعداد گویا یا به صورت اشتراک دنباله‌ای از بازه‌های فشردهٔ تودرتو با نقاط انتهایی گویا نمایش بدهیم. فرض کنید که در حالت کلی نمایش یک عدد حقیقی به صورت دنبالهٔ $a_0a_1a_2a_3\ldots$ باشد که در آن a_n متعلق به یک مجموعهٔ شمارا (مانند اعداد صحیح یا اعداد گویا) است. در این صورت در این نمایش، تابع حقیقی $f:\mathbb{R}\to\mathbb{R}$ **محاسبه‌پذیر** است هرگاه با ورودی $x = a_0a_1a_2\ldots$ خروجی $f(x) = b_0b_1b_2\ldots$ را بتوانیم طوری تعیین کنیم که به ازای هر عدد طبیعی n، عدد طبیعی m وجود داشته باشد که مقادیر $b_0, b_1, \ldots, b_n$ تنها با استفاده از مقادیر $a_0, a_1, \ldots, a_m$ قابل محاسبه باشند. از اینجا نتیجه می‌شود که سیستم اعشاری نمایش اعداد برای محاسبهٔ دقیق مناسب نیست، زیرا برای مثال ضرب اعداد در عدد ۳ در این نمایش غیر قابل محاسبه است. درواقع با ورودی $x = 0.333\ldots$ نمی‌توان حتی رقم اول خروجی $f(x) = 3x$ را تعیین کرد. نمی‌توانیم اولین رقم را صفر قرار دهیم چون ممکن است پس از یک سلسله رقم ۳ بالاخره رقمی بزرگتر از ۳ در ورودی ظاهر شود و نمی‌توانیم اولین رقم را ۱ قرار دهیم چون امکان دارد سرانجام در ورودی رقمی کوچکتر از ۳ پیدا شود. درواقع، برای محاسبهٔ دقیق باید از یک نمایش اضافی[1] اعداد حقیقی استفاده کرد. سیستم اعشاری اساساً نمایشی غیراضافی است زیرا مثلاً هر عدد گنگ، تنها دارای یک نمایش اعشاری است.

یک مطلب اساسی دیگر، محاسبهٔ دقیق روی اعداد حقیقی را از محاسبه روی اعداد گویا (و در نتیجه محاسبه روی اعداد با نمایش ممیز شناور) کاملاً متمایز می‌نماید. مقایسهٔ اعداد گویا **تصمیم‌پذیر** است یعنی همواره می‌توان در مدت زمان متناهی تعیین کرد که دو عدد داده شدهٔ گویا با هم برابرند یا اینکه یکی بزرگتر از دیگری است. به عکس مقایسهٔ دو عدد حقیقی تصمیم‌پذیر

1. redundant

نیست. درواقع حتی مقایسهٔ یک عدد حقیقی با یک عدد گویا، مثلاً با صفر، نیز تصمیم‌پذیر نیست. این بدان معنی است که صرف‌نظر از اینکه کدام نمایش اعداد حقیقی را انتخاب نماییم نمی‌توانیم در مدت زمان متناهی تعیین کنیم که یک عدد حقیقی برابر با صفر است یا خیر. پس نتیجه می‌گیریم که در تقسیم اعداد نمی‌توان از تقسیم یک عدد حقیقی بر صفر جلوگیری کرد. لاجرم در حساب دقیق حقیقی باید بینهایت (∞) را به‌عنوان نتیجهٔ یک محاسبه بپذیریم. به همین ترتیب چون نمی‌توانیم مانع ایجاد $\frac{0}{0}$ شویم، باید ابهام ($\perp$) را نیز به‌عنوان نتیجهٔ برخی محاسبات قبول کنیم.

نکات بالا دلایل منسجمی برای استفاده از دستگاه توسعه‌یافتهٔ اعداد حقیقی، $\mathbb{R}^*$، به جای دستگاه اعداد حقیقی، $\mathbb{R}$، هستند. درواقع، $\mathbb{R}^*$ حاصل فشرده‌سازی تک‌نقطه‌ای $\mathbb{R}$ است که همسان‌ریخت با دایره است. معنی نماد ابهام ($\perp$) نیز کل دستگاه توسعه‌یافتهٔ حقیقی است، یعنی نتیجهٔ محاسبه‌ای که به ابهام منجر شود می‌تواند هر عنصری از $\mathbb{R}^*$ باشد. اکنون به این سؤال می‌پردازیم که چه نمایشی، برای عناصر $\mathbb{R}^*$ انتخاب کنیم. از آنجا که معنی نماد ابهام درواقع همان $\mathbb{R}^*$ است که خود یک بازهٔ فشردهٔ R^* به‌شمار می‌آید، طبیعی است که هر عنصر $x \in \mathbb{R}^*$ را با دنباله‌ای از بازه‌های فشردهٔ تودرتو و کوچک‌شونده با نقاط انتهایی گویا، $([\frac{a_n}{b_n}, \frac{c_n}{d_n}])$، نمایش دهیم که اشتراکشان $\{x\}$ باشد. روشن است که این یک نمایش اضافی از اعداد حقیقی است. هر یک از این بازه‌ها تقریب بهتری از بازهٔ قبلی برای مقدار x می‌دهد. شایان ذکر است که $[p,q] = [p,\infty] \cup [\infty,q]$ وقتی $p > q$.

در اینجا نمایش جدیدی از اعداد و توابع حقیقی ارائه می‌دهیم که هم دارای شالودهٔ محکم و خواص مناسب ریاضی است و هم به الگوریتمهای کارایی برای محاسبهٔ دقیق منجر می‌شود. در این مورد از دو زمینهٔ تحقیقاتی کاملاً متفاوت بهره‌گیری شده است. زمینهٔ اول، نمایش اعداد حقیقی به‌صورت کسرهای مسلسل است که در چند دههٔ اخیر مورد بررسی قرار گرفته است [Gos72,Vui90,NK95]. زمینهٔ دوم، کاربرد قلمروهای پیوسته[1] در ریاضیات و فیزیک از جمله در محاسبات عددی [Eda97] و همچنین کاربرد آنها در تشکیل نوع داده[2] برای اعداد حقیقی است [DiG93,Esc96]. بقیهٔ این مقاله به تشریح این نمایش جدید از اعداد حقیقی می‌پردازد که به طور بسیار مفصلتر در منابع [EP97,Pot98,EPS98,ES97a,ER98] بررسی شده است.

یک روش بسیار مناسب برای کدسازی دنبالهٔ $[\frac{a_n}{b_n}, \frac{c_n}{d_n}]$، استفاده از تبدیلهای کسری خطی $\mathbb{R}^* \to \mathbb{R}^* : x \mapsto \frac{ax+c}{bx+d}$ است. بازهٔ $[0,\infty]$ را بازهٔ پایه در نظر می‌گیریم و هر بازهٔ فشردهٔ گویای $[\frac{a}{b}, \frac{c}{d}]$ را به صورت برد یک تبدیل کسری خطی به روی این بازهٔ پایه می‌نویسیم. در واقع، هرگاه $ad-bc<0$، داریم $[\frac{a}{b}, \frac{c}{d}] = \phi[0,\infty]$ که در اینجا $\phi : x \mapsto \frac{ax+c}{bx+d}$ و هرگاه $ad-bc<0$، تابع $\phi : x \mapsto \frac{cx+a}{dx+b}$ در همان رابطه صدق می‌کند. از طرف دیگر به راحتی می‌توان نشان داد که هرگاه دو بازهٔ فشردهٔ گویای غیر تک‌نقطه‌ای را با تبدیلهای کسری خطی ϕ و ψ به شکل بالا نشان دهیم، داریم $\psi[0,\infty] \subseteq \phi[0,\infty]$ اگر و تنها اگر تبدیل کسری خطی $\phi^{-1}\circ\psi$ دارای مؤلفه‌های صحیح نامنفی و یا غیر مثبت باشد. یادآوری می‌کنیم که

1. continuous domains 2. data type

تبدیل کسری خطی $x \mapsto \frac{ax+c}{bx+d}$ را می‌توان با ماتریس $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ نمایش داد. در واقع اگر $\mathrm{GL}(2,\mathbb{R})$ گروه عمومی خطی ماتریسهای 2×2ی حقیقی ناتکین باشد و M گروه تبدیلهای کسری خطی حقیقی ناتکین، آنگاه همریختی $\Theta : \mathrm{GL}(2,\mathbb{R}) \to \mathbb{M}$ که ماتریس $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ را به تبدیل کسری خطی $x \mapsto \frac{ax+c}{bx+d}$ می‌برد دارای هستهٔ $\{\lambda I | \lambda \in \mathbb{R}\setminus\{0\}\}$ است که I ماتریس همانی است.

از آنچه گفته شد نتیجه می‌گیریم که هر عنصر $x \in \mathbb{R}^*$ را می‌توان به صورت ترکیب نامتناهی تبدیلهای کسری خطی یا ضرب نامتناهی ماتریسهای 2×2 نشان داد:

$$x = M_0 M_1 M_2 \ldots$$

که در آن $\{x\} = \cap_{n\geq 0} M_0 M_1 \ldots M_n[0,\infty]$ و M_0 یک ماتریس 2×2 با مؤلفه‌هایی از اعداد صحیح است و M_n ($n \geq 1$) ماتریسهای 2×2 با مؤلفه‌هایی از اعداد طبیعی هستند.

تا اینجا فرض ما بر آن بود که M_n به ازای هر عدد طبیعی n یک ماتریس وارون‌پذیر و یا تبدیل کسری خطی ناتکین است. با استفاده از ماتریسهای تکین می‌توان نمایشی متناهی از اعداد گویا به‌دست داد. توجه کنید که اگر $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ تکین باشد ولیکن دارای ستون $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ نباشد، آنگاه یک تبدیل کسری خطی نمایش می‌دهد که مقدارش ثابت و برابر $\frac{a}{b}$ است، مگر در نقطهٔ $x = -\frac{d}{b} = -\frac{c}{a}$ که در آنجا مقدارش مبهم می‌شود. پس به شرط اینکه نقطهٔ تکین $x = -\frac{d}{b}$ خارج از $[0,\infty]$ قرار داشته باشد یعنی به شرط اینکه همهٔ ضرایب نامنفی و یا نامثبت باشند و $ad-bc=0$، آنگاه می‌توانیم به‌جای $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ بردار $\begin{pmatrix} a \\ b \end{pmatrix}$ و یا عدد گویای $\frac{a}{b}$ را قرار دهیم. در نتیجه اگر n کوچکترین عدد طبیعی باشد به طوری که M_n یک ماتریس تکین از نوع بالا باشد آنگاه ضرب نامتناهی ماتریسها در $x = M_0 M_1 \ldots M_n M_{n+1} \ldots$ به ضرب متناهی $x = M_0 \ldots M_{n-1} M_n = M_0 \ldots M_{n-1} V$ محدود می‌شود که در اینجا بردار V از رابطهٔ $M_n = (V, V)$ به‌دست می‌آید. روشن است که در اینجا x یک عدد گویاست.

بنا بر آنچه گفته شد ضرب نامتناهی ماتریسها نمایشگر همهٔ اعداد دستگاه توسعه‌یافتهٔ حقیقی است و ضرب متناهی ماتریسها که به یک بردار ختم شود نمایشگر اعداد گویاست که شامل $\frac{1}{0} = \infty$ نیز هست. این گونه نمایش اعداد را **ضرب نرمال** می‌نامیم.

اولین ماتریس ضرب نرمال راکه می‌تواند دارای اعداد صحیح با علامتهای متفاوت باشد *ماتریس علامت* می‌خوانیم و مابقی ماتریسها را که هر یک دارای اعداد صحیحی است که همگی نامنفی و یا همگی نامثبت هستند *ماتریس رقم* می‌نامیم. این نامگذاری با الهام از دستگاه اعشاری صورت پذیرفته است که در آن هر عدد حقیقی ابتدا با یکی از دو علامت + یا − مشخص می‌شود و سپس دنباله‌ای از ارقام ۰، ۱، ۲، ...، ۹، که بدون علامت‌اند مقدار عدد را در سمت چپ و یا راست محور حقیقی مشخص می‌کند. ماتریس علامت نیز ابتدا بازهٔ فشرده‌ای بر روی دایره تعیین می‌نماید که موقعیت کلی عدد مورد نظر را روشن می‌کند و سپس ماتریسهای رقم، قدم به قدم این بازه را تظریف می‌کنند.

می‌گوییم عدد حقیقی x **محاسبه‌پذیر** است هرگاه برنامه‌ای متناهی در یک زبان عمومی برنامه‌نویسی، مانند ماشین تورینگ[1] و یا $C++$، وجود داشته باشد که خروجی آن عناصر دنباله‌ای از بازه‌های فشردهٔ گویای $([\frac{a_n}{b_n}, \frac{c_n}{d_n}])$ باشد به طوری که $\{x\} = \cap_n [\frac{a_n}{b_n}, \frac{c_n}{d_n}]$. در نمایش اعداد به‌صورت ضرب ماتریسها، عدد حقیقی x محاسبه‌پذیر است اگر برنامه‌ای وجود داشته باشد که خروجی آن دنباله‌ای از ماتریسهای ۲×۲ با مؤلفه‌هایی از اعداد صحیح تشکیل دهد (M_n) به طوری که (الف) مؤلفه‌های M_n برای $n > ۰$ همگی نامنفی باشد و (ب) $x = M_۰ M_۱ M_۲$ به مفهومی که پیشتر ذکر نمودیم، یعنی $\{x\} = \cap_{n \geq ۰} M_۰ \ldots M_n[۰, \infty]$.

برای مثال، $\sqrt{۲}$ را به صورت ضرب نامتناهی ماتریسها نشان می‌دهیم. می‌دانیم که کسر مسلسل منظم برای $\sqrt{۲}$ برابر است با

$$\sqrt{۲} = ۱ + \cfrac{۱}{۲ + \cfrac{۱}{۲ + \cfrac{۱}{۲ + \ldots}}}$$

دقت کنید که این کسر مسلسل خود ترکیبی از تبدیلهای کسری خطی است. اگر قرار دهیم

$$x = ۲ + \cfrac{۱}{۲ + \cfrac{۱}{۲ + \ldots}}$$

آنگاه داریم $\sqrt{۲} = ۱ + \frac{۱}{x}$ یا به عبارت دیگر $x\begin{pmatrix} ۱ & ۱ \\ ۱ & ۰ \end{pmatrix} = \sqrt{۲}$. به همین ترتیب می‌توانیم سایر ماتریسهای یک ضرب نرمال برای $\sqrt{۲}$ را به‌دست آوریم

$$x = \begin{pmatrix} ۱ & ۱ \\ ۱ & ۰ \end{pmatrix}\begin{pmatrix} ۲ & ۱ \\ ۱ & ۰ \end{pmatrix}\begin{pmatrix} ۲ & ۱ \\ ۱ & ۰ \end{pmatrix}\ldots$$

از طرف دیگر ضربهای نرمال دیگری برای $\sqrt{۲}$ وجود دارد زیرا همان طور که تأکید کردیم نمایش ما اضافی است. برای نمونه $\sqrt{۲}$، نقطهٔ ثابت تابع $\begin{pmatrix} ۱ & ۲ \\ ۱ & ۱ \end{pmatrix}$ است. در نتیجه یک ضرب نرمال دیگر برای $\sqrt{۲}$ بدین ترتیب خواهد بود

$$\sqrt{۲} = \begin{pmatrix} ۱ & ۲ \\ ۱ & ۱ \end{pmatrix}\begin{pmatrix} ۱ & ۱ \\ ۱ & ۱ \end{pmatrix}\begin{pmatrix} ۱ & ۲ \\ ۱ & ۱ \end{pmatrix}\ldots$$

در هر ضرب نرمال برای یک عدد حقیقی $x = M_۰ M_۱ M_۲ \ldots$، هر بخش متناهی اولیهٔ $M_۰ \ldots M_n$ بازهٔ گویای $M_۰ \ldots M_n[۰, \infty]$ را به‌عنوان تقریبی برای x به‌دست می‌دهد، که از تقریب قبلی $M_۰ \ldots M_{n-۱}[۰, \infty]$ بهتر است زیرا این بازهٔ اخیر بازهٔ اولی را شامل می‌شود. در نتیجه اطلاعات[2] یک ماتریس را برد آن ماتریس به روی $[۰, \infty]$ تعریف می‌کنیم

$$\text{Info}(M) = M[۰, \infty]$$

و به همین ترتیب اطلاعات یک بردار $\binom{a}{b}$ را عدد گویای $\frac{a}{b}$ تعریف می‌کنیم هرگاه a و b هر دو صفر نباشند و در غیر این صورت اطلاعات آن را $\mathbb{R}^*$ تعریف می‌کنیم

$$\text{Info}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{cases} \frac{a}{b} & a \neq ۰ \text{ یا } b \neq ۰ \\ \mathbb{R}^* & a = ۰ \text{ و } b = ۰ \end{cases}$$

1. Turing Machine
2. information

اکنون به نمایش توابع حقیقی در این چارچوب می‌پردازیم. در اینجا از ایدهٔ گاسپر در [Gos72] برای انجام عملیات اصلی روی کسرهای مسلسل از طریق تبدیلهای کسری خطی دو متغیره استفاده می‌کنیم. این تبدیلها به صورت زیر هستند

$$(x, y) \mapsto \frac{axy + cx + ey + g}{bxy + dx + fy + h} : \mathbb{R}^* \times \mathbb{R}^* \to \mathbb{R}^*$$

که آنها را با تانسور $\begin{pmatrix} a & c & e & g \\ g & d & f & h \end{pmatrix}$ نشان می‌دهیم. بدیهی است که چهار عمل اصلی را می‌توان با انتخاب مناسب ضرایب تانسور انجام داد. برای نمونه $\begin{pmatrix} ۰ & ۱ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۱ \end{pmatrix}$ تانسور جمع و $\begin{pmatrix} ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ \end{pmatrix}$ تانسور ضرب دو عدد در $\mathbb{R}^*$ است.

همچون مورد ماتریسها، اطلاعات یک تانسور را برد آن به روی $([۰, \infty], [۰, \infty])$ تعریف می‌کنیم:

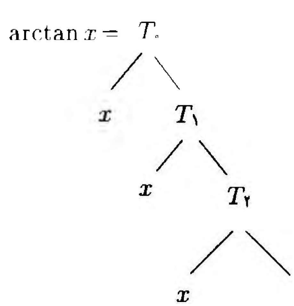
$$\text{Info}T = T([۰, \infty], [۰, \infty])$$

اکنون مفهوم ضربهای نرمال را به درختهای عبارتی[1] که در آنها تانسورها نیز جای دارند تعمیم می‌دهیم. درخت عبارتی یک درخت دودویی است که هر رأس آن یا یک تانسور است با دو فرزند، یا یک ماتریس است با یک فرزند و یا یک بردار است بدون فرزند که برگ درخت محسوب می‌شود. مانند ضرب نرمال، تنها ریشهٔ یک درخت عبارتی می‌تواند تانسور، ماتریس و یا برداری باشد که مؤلفه‌هایش دارای علامتهای متفاوت هستند. مابقی رئوس یک درخت عبارتی تانسورها، ماتریسها یا بردارهایی هستند که مؤلفه‌هایشان همگی نامنفی و یا همگی غیر مثبت هستند.

همان‌گونه که از کسر مسلسل $\sqrt{۲}$ برای نمایش این عدد به صورت ضرب نرمال استفاده کردیم، می‌توانیم از کسرهای مسلسل توابع ابتدایی نظیر $\arctan x, \tan x, \log x, e^x$ و غیره برای نمایش این توابع به صورت درخت عبارتی استفاده کنیم. به‌عنوان مثال، کسر مسلسل زیر برای $\arctan x$ را در نظر بگیرید.

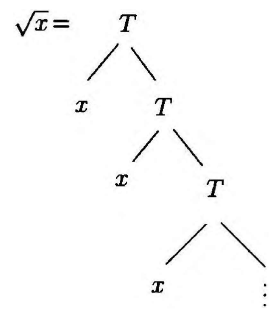
$$\arctan x = \cfrac{x}{۱ + \cfrac{x^۲}{۳ + \cfrac{۴x^۲}{\ddots\, ۲n - ۱ + \cfrac{n^۲x^۲}{\vdots}}}}$$

این کسر مسلسل را می‌توان با استفاده از روشهای مختلف به درخت عبارتی



1. expression trees

تبدیل کرد که در آن $T_n = \begin{pmatrix} 0 & 1 & 0 & 0 \\ n^2 & 0 & 0 & 2n-1 \end{pmatrix}$، و فرض شده است که x یک عدد حقیقی مثبت است که به وسیلهٔ یک عبارت درختی یا ضرب نرمال بدون علامت نمایش داده شده است. مثلاً x می‌تواند $\sqrt{2}$ باشد که به صورت یک ضرب نرمال نمایش داده شده است. به عنوان نمونهٔ دیگر، می‌توان مستقیماً نشان داد که



که در آن $T = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix}$.

همان‌گونه که ضرب نرمالْ دنباله‌ای از بازه‌های فشردهٔ تودرتو وکوچک‌شونده را نمایش می‌دهد، درخت عبارتیْ یک مجموعهٔ پالایش‌شده[1] از بازه‌های فشرده را نمایش می‌دهد. در واقع، هرگاه یک درخت عبارتی را بریده و تبدیل به یک درخت متناهی کنیم و به جای هر شاخهٔ بریده شده بازهٔ $[0, \infty]$ را قرار دهیم آنگاه ترکیبی متناهی از تبدیلهای کسری خطی یک و دو متغیره به دست می‌آوریم که بر بازهٔ $[0, \infty]$ اثر می‌کند و در نتیجه، یک بازهٔ فشرده حاصل می‌شود. مجموعهٔ بازه‌های فشرده‌ای که از طریق برشهای مختلف درخت عبارتی به صورت فوق به‌دست می‌آید پالایش شده است و اشتراک آنها برابر مقدار درخت عبارتی است. برای مثال درخت عبارتی $\arctan\sqrt{2}$ را که از طریق جایگزینی x در درخت عبارتی $\arctan x$ با یک ضرب نرمال برای $\sqrt{2}$ به دست می‌آید در نظر بگیرید. آنگاه اشتراک کلیهٔ بازه‌های فشرده ناشی از درختهای متناهی فوق برابر $\{\arctan\sqrt{2}\}$ خواهد بود.

برای محاسبهٔ یک درخت عبارتی نظیر $\arctan\sqrt{2}$، آن را به یک ضرب نرمال تبدیل می‌کنیم تا بتوانیم با هر دقت دلخواه مقدار تقریبی آن را به‌دست آوریم. توجه کنید که جهت سیر اطلاعات در هر درخت به سمت بالاست. برای نمونه، اگر ریشهٔ درخت یک تانسور T باشد، آنگاه تابع دو متغیره‌ای داریم که ورودی آن دو زیردرختِ تانسور هستند. برای محاسبهٔ درخت باید ببینیم که آیا در تانسور ریشه اصولاً اطلاعاتی موجود است یا نه. به عبارت دیگر آیا $T([0,\infty],[0,\infty]) \neq \mathbb{R}^*$؟ اگر چنین بود، می‌توانیم ماتریس M را طوری از T استخراج کنیم که $\text{Info}M \supseteq \text{Info}T$. فرض کنید $T = (T_1, T_2)$ که T_1 و T_2 دو ماتریس 2×2ی تشکیل‌دهندهٔ T هستند. پس از استخراج $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ ریشهٔ درخت یعنی T تبدیل به $M^\dagger T = (M^\dagger T_1, M^\dagger T_2)$ می‌شود که در اینجا $M^\dagger$ تبدیل کسری خطی وارون M با ضرایب صحیح است، یعنی $MM^\dagger = M^\dagger M = ad - bc$. اکنون M اولین ماتریس یعنی ماتریس علامتِ ضرب نرمال برای درخت عبارتی است.

اگر در T اطلاعاتی نبود یعنی اگر $\text{info}T = \mathbb{R}^*$، آنگاه باید از زیردرختهای موجود در T اطلاعات جذب کنیم تا بتوانیم پس از آن از T ماتریسی استخراج کنیم. قواعد مربوط به جذب اطلاعات همان قواعد مربوط به ترکیب تبدیلهای کسری خطی یک و دو متغیره است. برای مثال به آسانی می‌توان نشان داد که برای تانسور $T = (T_1, T_2)$ و ماتریس M همواره رابطهٔ زیر صادق است: $T(x, My) = T'(x, y)$ به‌طوری که $T' = (T_1 M, T_2 M)$. به همین شکل قاعده‌ای برای جذب ماتریس از متغیر اول به T موجود است و همچنین برای جذب یک بردار به‌عنوان متغیر اول و یا دوم به T.

برای همهٔ توابع اصلی (چهار عمل اصلی، توان، e^x، $\log x$، $\tan x$، $\arctan x$ و غیره) درختهایی عبارتی وجود دارد که می‌توان مقدار آنها را از طریق فوق با دقت دلخواه محاسبه کرد. همانگونه که گفتیم در این چارچوب همهٔ ماتریسهایِ با مؤلفه‌های صحیح، به‌عنوان ماتریس علامت و همهٔ ماتریسهایِ با مؤلفه‌های صحیح نامنفی یا نامثبت، به‌عنوان ماتریس رقم پذیرفته می‌شوند. در اینجا دو اشکال اساسی پیش می‌آید. یکی اینکه بزرگی اعداد صحیح در یک ماتریس ممکن است هیچگونه تناسبی با اطلاعات آن ماتریس نداشته باشد. در نتیجه، یک ماتریس با اعداد بسیار بزرگ صحیح ممکن است تنها تقریب ضعیفی از عدد حقیقی مورد نظر ما به‌دست دهد، که از نظر استفادهٔ متناسب از حافظه و منابع کامپیوتری غیر قابل قبول است. از طرف دیگر، در نمایش اعداد حقیقی به‌صورت ضرب نرمال، آهنگ انتشار اطلاعات یکنواخت نیست زیرا ماتریسهای مختلف می‌توانند به درجات کاملاً متفاوتی بازه‌های تقریبی را تظریف کنند. در نتیجه، چارچوب مناسبی برای تحلیل پیچیدگی مکانی و زمانی الگوریتمهای عددی در این شرایط به‌دست نمی‌آید.

راه‌حل این مشکلات این است که، همانند دستگاه اعشاری، ماتریسهای پذیرفته شده در ضرب نرمال را به یک مجموعهٔ متناهی محدود نماییم. ابتدا از ماتریسهای علامت شروع می‌کنیم. ماتریس $S_0 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ ماتریس دوران دایره به‌اندازهٔ $\frac{\pi}{2}$ درجهت عقربه‌های ساعت است که مولد یک گروه دوری از مرتبه ۴ است:

$$S_- = S_0^1 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, S_\infty = S_0^2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix},$$

$$S_+ = S_0^4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

می‌توان ثابت کرد که این تنها گروه متناهی از دَورانهای دایره است، که دارای یک نمایش ماتریسی با اعداد صحیح است به‌طوری که اطلاعات ماتریسهای گروه با یکدیگر همپوشانی دارند. اکنون به ماتریسهای رقم می‌پردازیم. نخست متذکر می‌شویم که دستگاه دودویی علامتدار برای نمایش اعداد حقیقی که از ارقام مجموعهٔ $\{-1, 0, 1\}$ به جای $\{0, 1\}$ استفاده می‌کند به‌وسیلهٔ سه تابع زیر مشخص می‌شود

$$f_i : x \longmapsto \frac{x+i}{2} : [-1, 1] \to [-1, 1] \qquad (i \in \{-1, 0, 1\})$$

1. filtered

درواقع، هرگاه در این دستگاه $x = i_1 i_2 i_3 \cdots = \sum_{n=1}^{\infty} \frac{i_n}{2^n}$،

$$\{x\} = \bigcap_{n \geq 1} f_{i_1} \cdots f_{i_n}[-1, 1] \text{ آنگاه } (i_n \in \{-1, 0, 1\})$$

از نظر اهمیتی که این نمایش اضافی در حساب کامپیوتری دارد، ماتریسهای رقم را طوری انتخاب می‌کنیم که از طریق تابع $S_0 : [0, \infty] \to [-1, 1]$ با سه تابع $f_i : [-1, 1] \to [-1, 1]$، به‌ازای $i = -1, 0, 1$، مزدوج شوند. یعنی سه ماتریس رقم را بدین‌ترتیب تعیین می‌کنیم: $D_i = S_0^{\dagger} f_i S_0$، به‌ازای $i = -1, 0, 1$، یا

$$D_{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}, D_0 = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}, D_1 = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

نتیجهٔ جالبی که به‌دست می‌آید، این است که عدد حقیقی x که در دستگاه دودویی علامتدار دارای نمایش $i_1 i_2 i_3 \cdots$ است همچنین دارای نمایش $S_0 D_{i_1} D_{i_2} \cdots$ در ضرب نرمال است. توجه کنید که $\text{info} D_{-1} = [0, 1]$، $\text{info} D_0 = [\frac{1}{3}, 3]$ و $\text{info} D_1 = [1, \infty]$.

چهار ماتریس علامت S_0، S_{-1}، S_{∞}، S_+ و سه ماتریس رقم D_{-1}، D_0، D_1 نمایشی اضافی از اعداد حقیقی را به‌صورت ضرب نرمال تشکیل می‌دهند که آن را **نمایش با ممیز شناور دقیق** در پایهٔ ۲ می‌نامیم. گفتنی است که سه تابع $D_i : [0, \infty] \to [0, \infty]$، به‌ازای $i = -1, 0, 1$، نسبت به متر $d(x, y) = |S_0(x) - S_0(y)|$ در بازهٔ $[0, \infty]$ توابع منقبض‌کننده با ضریب انقباض $\frac{1}{2}$ هستند. در نتیجه در نمایش با ممیز شناور دقیق، آهنگ انتشار اطلاعات به‌صورت یکنواخت نسبت به متر d صورت می‌گیرد. در ضمن اندازهٔ ارقام در ماتریسهای رقم $(0, 1, 2, 3)$ متناسب با ضریب انقباض‌کنندهٔ این ماتریسها، یعنی $\frac{1}{2}$، است.

در مجموع، نمایش اعداد حقیقی با ممیز شناور دقیق چارچوب مناسبی برای محاسبات دقیق عددی است. ولی باید ببینیم که در نمایش دقیق اعداد چگونه می‌توان تابعی را از طریق پیوند دو تابع، که قلمرو هر یک بخشی از $\mathbb{R}^*$ است، تعریف نمود. در اینجا بار دیگر به مسألهٔ تصمیم‌ناپذیری مقایسهٔ اعداد حقیقی بر می‌خوریم. در محاسبه روی اعداد گویا می‌توان برای نمونه تابعی را چنین تعریف کرد

$$f : \mathbb{Q} \longrightarrow \mathbb{Q}$$
$$x \longmapsto \begin{cases} g(x) & x < 0 \\ h(x) & x \geq 0 \end{cases}$$

که در آن $g : \mathbb{Q}^- \to \mathbb{Q}$ و $h : \mathbb{Q}^+ \to \mathbb{Q}$ توابع داده شده‌ای هستند. اما به‌دلیل تصمیم‌ناپذیری مقایسهٔ اعداد حقیقی چنین تعریفی برای توابع حقیقی امکان پذیر نیست. برای حل این مشکل، متغیر حقیقی را با دو عدد گویا به‌طور موازی مقایسه می‌کنیم. چون عدد حقیقی مزبور حداکثر می‌تواند برابر یکی از دو عدد گویا باشد، یکی از دو مقایسهٔ فوق در زمان متناهی به نتیجه می‌رسد، که از آن می‌توان برای تعریف تابع حقیقی استفاده کرد. به‌طور مشخص فرض کنید a و b دو عدد مختلف گویا باشند، مثلاً $0 < a < b < \infty$. همچنین فرض کنید توابع $g : [0, b] \to \mathbb{R}^*$ و $h : [a, \infty] \to \mathbb{R}^*$ داده شده‌اند به‌طوری که $h(x) = g(x)$ به‌ازای $x \in [a, b]$. در این صورت می‌توانیم تابع

$$\begin{cases} f : [0, \infty] \longrightarrow \mathbb{R}^* \\ \quad x \longmapsto \begin{cases} g(x) & x \leq b \\ h(x) & x \geq a \end{cases} \end{cases}$$

را تعریف کنیم. این نوع ساختار را، که در آن تابع را به‌طور موردی تعریف می‌کنیم، «اگر اضافی[1]» می‌نامیم [EPS98]. «اضافی» از این نظر که تابع در بازهٔ $[a, b]$ به دو صورت مختلف ولی برابر تعریف شده است. توجه کنید که این روشِ تعریف تابع مانند ادامهٔ تحلیلی[2] توابع مختلط است. می‌دانیم که هرگاه دو تابع تحلیلی مانند g و h به روی یک بازهٔ باز مانند (a, b) برابر باشند، آنگاه هر یک ادامهٔ تحلیلی دیگری به‌شمار می‌رود.

موضوع دیگر، مسألهٔ پیچیدگی است. برخی نتایج اولیه در زمینهٔ پیچیدگی مکانی و زمانی الگوریتمها در چارچوب ممیز شناور دقیق به‌دست آمده است [Hec97,Hec97a,Hec97b]. به‌طور خلاصه، پس از جذب یا استخراج n ماتریس رقم از یک تانسور (یا ماتریس) دلخواه، حداقل یکی از مؤلفه‌های ماتریس دارای اندازهٔ بیت n خواهد بود، مگر در موارد استثنائی که اندازهٔ آن کراندار باقی می‌ماند. پیچیدگی زمانی این محاسبات نیز برابر پیچیدگی ضرب دو عدد صحیح با اندازهٔ بیت n است. بر این اساس پیچیدگی زمانی و مکانی الگوریتمهای مربوط به توابع اولیه باید بررسی شود.

برخی تحقیقات نیز در زمینهٔ نوع دادهٔ حقیقی[3] انجام پذیرفته است. هدف این است که در یک زبان برنامه‌نویسی که دارای نوع دادهٔ بولی[4] و نوع دادهٔ صحیح[5] است، نوع داده‌ای برای اعداد حقیقی اضافه شود که معرف نمایش ضرب نرمال برای اعداد حقیقی و درختهای عبارتی برای توابع حقیقی گردد. مناسبترین زبانهای برنامه‌نویسی برای این منظور زبانهای برنامه‌نویسی تابعی[6] هستند. برای مثال در [EPS98]، زبان برنامه‌نویسی PCF[7] که دارای نوع دادهٔ بولی و نوع دادهٔ صحیح است به زبان کاملتری با نوع دادهٔ حقیقی توسعه یافته است. در این زبان جدید قواعدی برای تحویل قطعی[8] عبارتهای محاسباتی تعریف شده‌اند که قدم به قدم مقدار عددی یک عبارت را محاسبه می‌کنند. ثابت می‌شود که قواعد فوق معنای عبارتها یا به زبان دیگر مقدار آنها را تغییر نمی‌دهند و در نتیجه زبان برنامه‌نویسی ما دارای کفایت محاسباتی[9] است. زبان برنامه‌نویسی دیگری با نوع دادهٔ حقیقی در [Pot98] تشریح شده است.

چارچوب کلی نمایش اعداد بر مبنای ضرب نرمال و همچنین چارچوب ممیز شناور دقیق در برخی از زبانهای برنامه نویسی از جمله C، $C++$، کمل[10]، میراندا[11] و هسکل[12] پیاده‌سازی شده است. اکنون پروژه‌ای برای پیاده کردن این چارچوب در جاوا[13] در دست اجراست.

1. redundant if 2. analytic continuation 3. real number data type 4. Boolean data type 5. integer data type 6. functional programming languages 7. Programming Language for Computable Functions 8. deterministic reduction rules 9. computational adequacy 10. Caml 11. Miranda 12. Haskel 13. Java

•

یکی از شاخه‌های دیگر تحقیق در حساب دقیق حقیقی کاربردهای مختلف آن در زمینه‌های گوناگون است. تاکنون الگوریتمهای کارایی برای محاسبهٔ دقیق ریشه و نقطهٔ ثابت یک تابع حقیقی به‌دست آمده و پیاده‌سازی شده‌اند [ER98]. همچنین الگوریتمهایی برای محاسبهٔ دقیق انتگرال یک تابع حقیقی استنتاج و پیاده‌سازی شده‌اند [Krz98].

در خاتمه متذکر می‌شویم که مهمترین مسأله در استفاده از حساب دقیق حقیقی کارایی کمتر آن از لحاظ پیچیدگی زمانی و مکانی نسبت به محاسبات ممیز شناور است. در نتیجه یکی از مهمترین وظایف تحقیقی بعدی افزایش کارایی الگوریتمهای ممیز شناور دقیق از جمله از طریق پیاده کردن آنها در سخت افزار است.

با ذکر دو مسألهٔ حل نشده به مقاله حاضر پایان می‌دهیم.

الف) حدس زده می‌شود که هر تابع حقیقی که بتوان آن‌را از طریق یک درخت عبارتی نمایش داد یک تابع برخه‌ریخت است. به‌عکس هر تابع برخه‌ریخت[1] حقیقی را می‌توان به‌صورت یک درخت عبارتی نمایش داد.

ب) آیا می‌توان نوع دادهٔ مختلطی از طریق تبدیلهای کسری خطی با ضرایب متشکل از اعداد گاوسی (یعنی اعدادی به‌صورت $a + ib$ که a و b اعداد صحیح هستند) تشکیل داد، که کارایی نوع دادهٔ حقیقی را داشته باشد؟ یادآوری می‌کنیم که هر تبدیل کسری خطی مختلط، هر قرص صفحهٔ مختلط را به یک قرص می‌نگارد. لیکن مشخص ساختن تبدیلهای کسری خطی که یک قرص پایه (برای مثال نیم‌صفحهٔ راست) را به‌درون خود بنگارد، از لحاظ محاسباتی بسیار پیچیده‌تر از مورد حقیقی آن است. در ضمن، یافتن مرکز و شعاع قرصی که برد یک تبدیل کسری خطی به روی قرص پایه باشد از لحاظ محاسباتی گران است. با وجود این مشکلات، درختهای عبارتی توابعی چون $\tan x$ و $\arctan x$ که در [Pot98] آمده است، از طریق ماتریسهای مزدوج مختلط به یکدیگر تبدیل می‌شوند، اگر چه اساس ریاضی این تبدیلها هنوز روشن نیست. حدس زده می‌شود که بررسی کامل نمایش اعداد از طریق تبدیلهای کسری خطی در صفحهٔ مختلط نه‌تنها به سؤالات بالا جواب دهد بلکه به شناخت بهتر نوع دادهٔ حقیقی منجر گردد.

سپاسگزاری

این مقاله در زمانی که در تهران مهمان مرکز تحقیقات فیزیک نظری و ریاضیات بودم تهیه شده است. از الهام کاشفی به‌خاطر کمک ایشان به ویرایش مقاله سپاسگزارم.

مراجع

[DiG93] P. Di Gianantonio, *A Functional Approach to Real Number Computation*, Ph.D. thesis, University of Pisa (1993).

[Eda97] A. Edalat, "Domains for computation in mathemetics, physics and exact real arithmetic", *Bulletin of Symbolic Logic*, (4) **3** (1997) 401-452.

[EP97] A. Edalat and P. J. Potts, "A new representation for exact real numbers", in *Proceedings of Mathematical Foundations of Programming Semantics 13* (1977), edited by S. Brookes and M. Mislove, URL: `http://www.elsevier.nl/locate/entcs/volume6.html`.

[EPS98] A. Edalat and P.J. Potts and P. Sünderhauf, "Lazy computation with exact real numbers", *Proceedings of ICFP98*, URL: `http://theory.doc.ic.ac.uk/~ae` (1998).

[ER98] A. Edalat and F. Rico, "Root finding in exact real arithmetic using linear fractional transformations", *Proceedings of Real Numbers and Computers III* (1998).

[ES97a] A. Edalat and P. Sünderhauf, "A domain theoretic approach to computability on the real line", to appear in *Theoretical Computer Science*, URL: `http://theory.doc.ic.ac.uk/people/Edalat/reals.ps.gz`.

[Esc96] Escardó, M. H., "PCF extended with real numbers", *TCS*, (1) **162** (1996) 79-115.

[Gos72] W. Gosper, *Continued Fraction Arithmetic*, MIT (1972).

[Hec97] R. Heckmann, "The appearance of big integers in exact real arithmetic based on linear fractional transformations", in *Proc. Foundations of Software Science and Computation (FoSSaCS '98)*, Springer-Verlag (1998).

[Hec97a] R. Heckmann, "Contractivity of linear fractional transformations", in *Third Real Numbers and Computers Conference (RNC3)*, edited by J.-M. Chesneaux and F. Jézéquel and J.-L. Lamotte and J. Vignes, (1998).

[Hec97b] R. Heckmann, "Big integers and complexity issues in exact real arithmetic", Presented at the third Comprox workshop (Sept. 1997 in Birmingham), accepted for publication in *ENTCS* (*Electronic Notes in Theoretical Computer Science*), **13** (1998).

[Krz98] M. Krznaric, *Numerical Integration in Exact Real Arithmetic*, M.Sc. thesis, Imperial College (1998).

[NK95] A. Nielsen and P. Kornerup, "MSB-first digit serial arithmetic", *J. of Univ. Comp. Scien.*, (7) **1** (1995) 523-543.

[Pot98] P. Potts, *Exact Real Arithmetic Using Mobius Transformations*, Ph.D. thesis, Imperial College (1998).

[Vui90] J. E. Vuillemin, "Exact real computer arithmetic with continued fractions", *IEEE Transactions on Computers*, (8) **39** (1990) 1087-1105.

1. meromorphic

******

* عباس عدالت، کالج امپریال، لندن، انگلستان

ae@doc.ic.ac.uk