



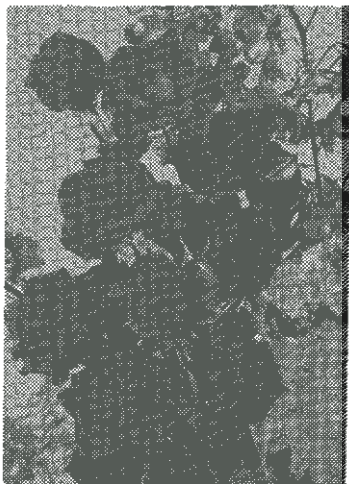
گزارش کامیوتر

ماهی ۱۳۸۷
شماره ۱۱۱

مجموعه‌های کتاب

- گزارش
- مجموعه‌های کتاب
- صاحب‌با حورف





تصویر روی جلد: گل سرخ

گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران

صاحب امتیاز: انجمن انفورماتیک ایران

مدیر مسئول: ابراهیم نقیبزاده مشایخ

سردبیر: علی یارسا

گزارش کامپیوتر فعلا هر دو ماه یکبار منتشر می شود.

مسئولان انجمن

رئیس: ابراهیم نقیبزاده مشایخ (۲۴۳۸۶۰)

نایب رئیس: داریوش جوان تیریزی (۸۶۳۳۶۷)

خزانه دار: میرعلی زیدی منفرد (۶۸۳۶۴۷)

دبیر: علی یارسا (۶۱۴۴۲۵)

سرپرستان کمیته ها

انتشارات: سعید وحید (۸۹۸۴۵۹)

روابط عمومی: داریوش جوان تیریزی (۸۶۳۳۶۷)

فعالتهای علمی و فنی: محمد حسن محوری

آموزش: ابراهیم نقیبزاده مشایخ (۲۴۳۸۶۰)

عضویت: میرعلی زیدی منفرد (۶۸۳۶۴۷)

دیگراعضای هیئت اجرایی

مسئول هماهنگی کمیته آموزش: دکتر مهدی بهشتیان (۲۲۳۸۳۶)

مسئول هماهنگی کمیته فعالتهای علمی و فنی: حسین کمالی (۲۸۰۹۵۸)

بازرس انجمن: علی اکبر نهرودی (۶۷۰۷۹۳)

مقاله ها و گزارشهای تالیفی یا ترجمه خود را برای گزارش کامپیوتر بفرستید. مطالب ارسالی را با خط خوانا (یاماشین شده) بر یک روی کاغذ بنویسید. فاصله کافی برای افزودن اصلاحات و ویرایشی در بین خطها و حاشیه در نظر بگیرید. در صورت ارسال مطلب ترجمه شده، یک نسخه از اصل مطلب را نیز ارسال دارید. شکلها باید با روان نویس مشکی ترسیم و به صورت آماده چاپ ارسال شوند.

انجمن انفورماتیک ایران انجمنی است علمی و مستقل که به منظور تبادل اطلاعات علمی و فنی و ارتقا سطح دانش انفورماتیک در ایران تاسیس شده است. این انجمن توسط هیئت اجرایی و زیر نظر مجمع عمومی اداره می شود. اعضای هیئت اجرایی با رای اعضای انجمن برای مدت دو سال انتخاب می شوند.

مطالب این شماره از گزارش کامپیوتر با استفاده از برنامه TEX^۱ چاپی حروف چینی و صفحه بندی شده است. در این شماره از همکاری شرکت داده کاوی ایران برخوردار بوده ایم که بدینوسیله از ایشان تشکر می کنیم. آقایان حسین یادبانیچی و احمد یزدی پور در انجام مراحل مختلف صفحه بندی و حروفچینی ما را یاری داده اند که از ایشان نیز قدردانی می شود.

چاپ: چاپخانه سکه

نشانی دفتر انجمن: خیابان دکتر فاطمی، خیابان پروین اعتصامی، کوچه دوم، پلاک ۲۱، طبقه سوم.

ساعات کار دفتر: هر روز بجز پنجشنبه ها از ۹/۵ تا ۱۲.

شماره تلفن دفتر: ۶۵۶۶۸۷

شماره حساب بانکی انجمن: ۱۱۱۷ بانک صادرات شعبه ۱۶۵۲

نشانی پستی انجمن: انجمن انفورماتیک ایران، صندوق پستی ۱۱۹۶-۱۴۱۵۵ تهران.

* * *

GOZARESH-E-COMPUTER

(Computer Report)

Vol.13, No.111, February-March 1992

P.O. Box 1196, Tehran 14155

A publication of the Informatics Society of Iran (ISI). It features original and translated articles, reports, news and reviews on all aspects of computers in Iran and abroad.

ISI is an independent technical and scientific society committed by its constitution to the advancement of computer knowledge in Iran. It is governed by an executive body, members of which are elected every two years by members of the society.

در این شماره

مقاله

مجموعه های مشکک ۸

زبانهای برنامه سازی ۱۴

بخش ها

اخبار انجمن ۳

اخبار جهان ۳

نامه ها ۵

سخن رئیس ۷

مصاحبه ۲۴

فهرست سالانه ۳۳

گزارش

سمینار اسپارک ۲۰

شبکه ها علیه استقلال ۲۱

دومین مرحله المپیاد ملی ۲۲

واژه نامه

حرف M ۲۶

ترجمه یک کتاب

نفرماه افسانه ای (۱۱) ۲۸

اخبار انجمن

صدور پروانه انتشار گزارش کامپیوتر

پروانه انتشار گزارش کامپیوتر به نام انجمن انفورماتیک ایران و با مدیر مسئولی آقای ابراهیم نقیب‌زاده مشایخ از سوی اداره کل مطبوعات وزارت فرهنگ و ارشاد اسلامی صادر شد. این امر باعث حذف مرحله اخذ مجوز برای هر شماره نشریه از آن اداره کل و در نتیجه تسریع در امر آماده سازی و ارسال گزارش کامپیوتر خواهد گردید.

سمینارهای سال آینده انجمن

در آخرین نشست هیأت اجرایی در بهمن ماه تصمیم گرفته شد که «مهندسی نرم‌افزار» و «شبکه‌های کامپیوتری» عنوان دو سمیناری باشند که در سال آینده از سوی انجمن برگزار خواهند شد. در این ارتباط اعضای کمیته‌های برگزاری این دو سمینار مشخص شدند و در اسفند ماه جلسات مقدماتی این کمیته‌ها تشکیل گردید. محورهای موضوعی سمینار مهندسی نرم‌افزار در این شماره آمده است. اطلاعات تکمیلی در شماره‌های آینده گزارش کامپیوتر به اطلاع اعضای گرامی انجمن رسانده خواهد شد.

برگزاری سخنرانیهای علمی بهمن ماه

اولین سخنرانی علمی بهمن ماه انجمن تحت عنوان «سیستم مدیریت پایگاه داده‌های شیء‌گرا و نگهداری نرم‌افزار» در تاریخ ۱۶ بهمن ۱۳۷۰ در محل آمفی‌تئاتر دانشکده علوم دانشگاه تهران و در حضور عده زیادی از علاقه‌مندان برگزار گردید. سخنران این جلسه آقای کامیار صادقی دانشجوی دکترای علوم کامپیوتر در دانشگاه سنتاکلارا بودند که در باره تز دکترای خود به بحث پرداختند.

آقای صادقی در سخنرانی خود با مروری بر ویژگیهای سیستمهای مدیریت پایگاه داده‌های شیء‌گرا و مقایسه تطبیقی آنها با سیستمهای

رابطه‌ای موجود، به تشریح مشخصات سیستم تحلیل و نگهداری نرم‌افزار SAMS که در دانشگاه سنتاکلارا ایجاد شده است پرداختند. به گفته ایشان سیستم SAMS دو هدف عمده زیر را تعقیب می‌کند:

- امکان و مطلوبیت استفاده از یک برای سیستم مدیریت پایگاه داده‌های شیء‌گرا ذخیره‌سازی و کار با حجم عظیمی از اطلاعات لازم برای نگهداری برنامه‌های بسیار طولانی در زبان C.

- بررسی چگونگی فراهم ساختن محیط مناسب برای افزودن ابزارهای جدید به یک سیستم نرم‌افزاری موجود توسط سیستم مدیریت پایگاه داده‌های شیء‌گرا.

انجمن انفورماتیک ایران بدین وسیله از آقای کامیار صادقی که در زمان تعطیلات خود دعوت انجمن را برای برگزاری این سخنرانی قبول کردند و نیز از مدیریت محترم دانشکده علوم دانشگاه تهران برای در اختیار گذاشتن محل برگزاری سخنرانی صمیمانه سپاسگزاری می‌نماید.

خبر برگزاری این سخنرانی توسط روزنامه‌های سلام و کیهان به اطلاع علاقه‌مندان رسانده شده بود.

دومین سخنرانی علمی بهمن ماه انجمن تحت عنوان «برنامه‌سازی به منطق» در تاریخ ۲۳/۱۱/۷۰ در محل گروه ریاضی دانشکده علوم دانشگاه تهران و در حضور عده زیادی از علاقه‌مندان برگزار گردید. سخنران این جلسه آقای حسین کمالی بودند که مشروح سخنان خود را نیز به صورت کتبی جهت درج در گزارش کامپیوتر در اختیار کمیته انتشارات انجمن قرار دادند.

انجمن انفورماتیک ایران ضمن تشکر از آقای کمالی به خاطر قبول دعوت انجمن، از مسئولان محترم گروه ریاضی دانشکده علوم دانشگاه تهران نیز به خاطر در اختیار قراردادن محل سخنرانی صمیمانه قدردانی می‌نماید.

برگزاری سخنرانی علمی اسفند ماه

سخنرانی علمی اسفندماه انجمن، تحت عنوان «منطق غیرصوری و نظریه پیچیدگی» در تاریخ ۷۰/۱۲/۱۴ در محل گروه ریاضی دانشکده علوم دانشگاه تهران برگزار گردید. سخنران این جلسه آقای دکتر ضیا موحد عضو هیأت علمی انجمن حکمت و فلسفه ایران بودند.

انجمن انفورماتیک ایران ضمن تشکر از آقای دکتر موحد به خاطر قبول دعوت انجمن، از مسئولان محترم گروه ریاضی دانشکده علوم دانشگاه تهران نیز که با وجود تداخل سخنرانی انجمن با برگزاری امتحانات کارشناسی ارشد، محل سخنرانی را در اختیار انجمن قرار دادند، صمیمانه سپاسگزاری می‌نماید.

اخبار جهان

ثروتمندتر از ملکه انگلیس

اگرچه ممکن است این خبر به مذاق مخالفان مایکروسافت خوش نیاید ولی مجله امریکایی فوربز می‌گوید که بیل گیتز از نظر میزان دارائیهای در امریکا مقام دوم را دارد. دارائیهای شخصی او حدود ۲/۸ بیلیون پوند تخمین زده می‌شود که شاید بیشتر از ثروت ملکه انگلیس باشد. سال قبل در رده‌بندی مجله فوربز، گیتز مقام شانزدهم را داشت. هیچ‌یک از اشخاص دست‌اندرکار کامپیوتر حتی در ده رده بالا هم نیستند.

همان حقوق سابق

شرکت کامپیوتری سان امریکا پنجاه نفر از طراحان ابرکامپیوترهای شوروی سابق را استخدام کرده است. این عده تحت نظر سرپرست فعلی خود یعنی باریس بابایان طراح برجسته کامپیوتر در آزمایشگاهی در مسکو کار خواهند کرد و دستمزدشان کمی بیشتر از دستمزد سابقشان خواهد بود. بنابه اظهار مقامات شرکت سان این گروه به طراحی ابرکامپیوترهایی که با انواع موجود امریکایی رقابت کند خواهند پرداخت.

ظاهراً این اولین مورد از استخدام کارشناسان و دانشمندان شوروی سابق از سوی یک شرکت امریکایی است.

حمله میکال آنز

جمعه ششم مارس (۱۶ اسفند) یک ویروس به بعضی از ریز کامپیوترها حمله کرد. این ویروس که میکال آنز نام دارد طوری تنظیم شده که در روز تولد میکال آنز به کار افتد و بساط کامپیوترها را در هم ریزد. اگرچه بسیاری از کامپیوترها با اقدامات دفاعی از قبیل تغییر تاریخ روز و گرفتن نسخه‌های پشتیبان از دیسک از این خطر جستند ولی به گزارش خبرگزاریها در سراسر جهان (و به دلایل نامعلوم) بویژه در افریقای جنوبی عده زیادی گرفتار عواقب این ویروس شده‌اند.

ادغام اشتون-تیت در بورلند

انتظار می‌رود خبر ادغام اشتون-تیت در بورلند باعث پایین آمدن قیمت نرم‌افزارهای اشتون-تیت شود. به این ترتیب بورلند سهم بسیار بزرگتری از بازار نرم‌افزارهای بانکهای اطلاعاتی را به خود اختصاص خواهد داد و مایکروسافت و لوتوس باید بکوشند محصولات خود را با بورلند همساز کنند.

بازار داغ شکایات

بعد از اختلاف بین مایکروسافت و اپل و اقامه دعوا علیه مایکروسافت توسط اپل، ظاهراً بازار شکایاتی از این دست هنوز داغ است: شرکت لوتوس سازنده نرم‌افزار کاربرگ 1-2-3 علیه بورلند اقامه دعوا کرده و ادعا می‌کند که بورلند کاربرگ Quattro را با تقلید از 1-2-3 ساخته است. لوتوس قبول دارد که یک «منوی ساده» را نمی‌توان تحت قانون کپی‌رایت قرار داد ولی می‌گوید مواردی نیز وجود دارند که منو «ساده» نیست. حال این دادگاه است که باید تصمیم بگیرد آیا منوی لوتوس یکی از آنهاست یا نه؟

از اپل نیز یک شرکت هم‌نام که ابزار آلات موسیقی تولید می‌کند شکایت کرده، چرا که اپل در سال ۸۱ با این شرکت توافق کرده بود که آرم خود را روی دستگاههای تولید موسیقی به کار نخواهد برد. کامپیوترهای مکینتاش این روزها تواناییهای بالای تولید و ترکیب موسیقی الکترونیکی را دارند. سال گذشته دو طرف ۷ میلیون پوند صرف مخارج دادگاه کردند و گفته می‌شود اپل محکوم به پرداخت ۱۷ میلیون پوند شده است.

نسل جدید برنامه‌های ویروس‌کش

شرکت «فیفت جنریشن سیستمز» نرم‌افزار ویروس‌کشی به نام Untouchable

ساخته است که نیازی به تمدید تاریخ انقضای آن نیست. این برنامه از سه روش برای شناسایی و از بین بردن ویروسها استفاده می‌کند. برای پی‌بردن به اینکه آیا در پرونده‌ها تغییراتی ایجاد شده یا نه، روشی به کار گرفته شده است که اثبات ریاضی دارد. به این ترتیب پرونده‌ها در مقابل ویروسهای شناخته شده و ناشناس محافظت می‌شوند. یک برنامه مقیم در حافظه (TSR) نیز از حافظه سیستم در مقابل ویروسهای قطاع راه‌اندازی (boot sector) و جدول افزاز (partition table) محافظت می‌کند. علاوه بر اینها از یک برنامه ویروس‌کشی عادی نیز برای یافتن و از بین بردن صدها ویروس شناخته شده، استفاده می‌شود.

ابر کامپیوترهای دهه ۹۰

شرکت «کری» جدیدترین محصول خود را با نام CRAY Y-MP C90 روانه بازار کرده است. این سیستم ۱۶ پردازنده مرکزی دارد و چهار برابر سریعتر از سریعترین محصول قبلی این شرکت عمل می‌کند. پردازنده این سیستم می‌تواند یک بیلیون دستور ممیز شناور را در هر ثانیه (Gflop/s) انجام دهد. با کنار هم آمدن ۱۶ عدد از این پردازنده‌ها و داشتن ۲ گیگابایت حافظه اصلی توان نهایی سیستم به ۱۶ گیگافلاپ در ثانیه می‌رسد.

توضیح درباره دو خبر

• در شماره ۱۱۰ گزارش کامپیوتر (آذر و دی ۷۰) در بخش «اخبار ایران» خبری تحت عنوان «ایستگاه کارسان در ایران» درج گشته بود که در آن به نقل از آقای سیمون یکی از مدیران اروپایی شرکت سان اعلام شده بود که «دریافت اجازه صدور این کامپیوترها برای دانشگاههای امیرکبیر و تهران قطعی و حمل تجهیزات قریب الوقوع است.» از آنجا که عده‌ای از خوانندگان تصور کرده بودند که چاپ این مطلب نشان‌دهنده تأیید این سخنان است، بار دیگر تأکید می‌کنیم که نقل مطالب در نشریه به معنی تأیید یا تکذیب آنها نیست. در عین حال برای آگاهی بیشتر علاقه‌مندان به موضوع لازم است ذکر شود که بر اساس تماسهایی که با مسئولین دو دانشگاه مزبور گرفته شد هنوز اجازه صدور این کامپیوترها گرفته نشده است. همچنین از مرکز تحقیقات فیزیک نظری و ریاضیات نیز اطلاع یافتیم که دو دستگاه کامپیوتر سان سفارش این مرکز نیز اجازه صدور دریافت نکرده و به همین دلیل اعتبار اسنادی مربوطه پس از دو بار تجدید بالاخره در آخر مارس ۱۹۹۲ باطل گشته است.

• بعد از درج خبر تأسیس نمایندگیهای آی‌بی‌ام در ایران در شماره قبل، شرکت دیاباسیستم ضمن ارسال صفحه اول قرارداد خود با آی‌بی‌ام که در آن این شرکت Remarketer آی‌بی‌ام معرفی شده، درخواست تصحیح آن خبر را کرده است. با توجه به اینکه در زمان تنظیم اخبار شماره قبل اطلاعات در این زمینه در اختیار گزارش کامپیوتر نبود، بدین وسیله خبر شماره قبل را اصلاح می‌کنیم. در عین حال از شرکتهای ذینفع دعوت می‌کنیم که برای آگاهی خوانندگان نشریه و سایر علاقه‌مندان تفاوت Remarketer و Reseller را از نظر شرکت آی‌بی‌ام، طی نامه یا مقالاتی توضیح دهند.

نامه‌ها

نامه‌های الکترونی

کمی دیر

اگرچه کمی دیر است، اما اشتباهات زیر در مقاله «الگوریتمهای ضرب و تقسیم اعداد ثابت» (سال ۱۲، شماره ۶) وجود داشتند:

• دکتری من از دانشگاه کالیفرنیا در لوس آنجلس بوده و نه از دانشگاه کالیفرنیا جنوبی.

• صفحه ۸ ستون ۲ سطر ۱۶ «زیربرنامه» اصلاح شود به «ریزبرنامه».

• صفحه ۹ ستون ۱ سطر ۲۷، «ظرفیت» اصلاح شود به «ظرفیت».

• صفحه ۹ ستون ۲ سطر ۲۰، $(1 - 2^{-n})(m/2^n)$ اصلاح شود به $(m/2^n)/(1 - 2^{-n})$.

• صفحه ۱۰ سطر ۲۱، «بزرگی» اصلاح شود به «بر اثر بزرگی».

بهروز پرهامی - سنتا باربارا

با پوزش/امیدواریم در آینده گزارش کامپیوتر را دقیقتر و صحیحتر از قبل عرضه کنیم. سردبیر

از طرف انجمن نوروز

ارسال نشریات IEE با بودجه انجمن نوروز در برایشون انگلستان بوده است و اینجانب تنها زحمت تهیه و ارسال آنها را تقبل نمودم. البته قصد داشتم برای سال جاری پیشنهاد کنم که در مقابل پرداخت حق عضویت و دریافت کتب منتشر شده توسط انجمن، شخصاً هزینه تهیه و ارسال مجلات فوق را برای سال ۱۹۹۲ به عهده بگیریم. سعید فراروی - برایتون

دو پیشنهاد

از اینکه مجله به موقع به دستمان می‌رسد تشکر می‌کنیم. اکثر سمینارها در دانشکده علوم دانشگاه تهران تشکیل می‌شود. خوب است از آمفی‌تئاترهای دانشکده فنی یا دانشگاه ملی، صنعتی امیرکبیر، شرکت زیمنس و ... نیز به دلیل آشنا شدن بیشتر با محیطهای جدید و راحتی رفت و آمد دیگر دانشجویان استفاده کنید. چرا از اساتیدی چون خانم نیاکی، آقای ووحانی یا آقای صالحی برای سخنرانی دعوت نمی‌کنید؟ مریم نراقی

برای برگزاری سخنرانیها قبلاً از دانشگاهها و مراکز دیگر استفاده کرده‌ایم و در آینده نیز خواهیم کرد. از همکاری استادانی که نام برده‌اید و سایرین بسیار خوشحال خواهیم شد و امیدواریم همین نامه را به عنوان دعوت بپذیرند. سردبیر

پس از دایر شدن پست الکترونی (E-mail) در مرکز تحقیقات فیزیک نظری و ریاضیات، عده‌ای از ایرانیان مقیم خارج به تصور اینکه این گره شبکه متعلق به انجمن است از این طریق با رئیس انجمن تماس گرفتند. به تماس گیرندگان اطلاع داده شده است که از این گره فعلاً برای تماس با انجمن استفاده نکنند. در زیر ترجمه برخی از نامه‌های رسیده را می‌خوانید.

دوره نوین

به نظر من این [دایر شدن پست الکترونی] نشانه آغاز دوره نوینی برای ایران و ما ایرانیان مقیم خارج است. من خبر فرخنده تماس خودمان را به کاربران شبکه اینترنت خواهم داد و هزاران نفر از دانشجویان و متخصصان ایرانی از شنیدن این خبر خوشحال خواهند شد. لطفاً سلام مرا به دوستان و اعضای انجمن برسانید. انوش حسینی
شرکت میپس
سانی ویل، کالیفرنیا

ستایش

از انوش شنیدم که انجمن هم روی شبکه رفته است. می‌خواهم احترام و ستایش خود را به شما و دیگرانی که گزارش کامپیوتر را تهیه می‌کنند ابراز کنم. بهمن خامنه‌یان
شرکت موتورولا
باتل، واشینگتن

همکاری بیشتر

آقای انوش حسینی به ما خبر داده که ارتباط ایران با شبکه بیت‌نت برقرار شده است. بسیار خوشوقتم که از طریق شبکه کامپیوتری بین‌المللی به شما نامه می‌نویسم. امیدوارم در آینده نزدیک شاهد همکاری بیشتر بین دانشمندان و مهندسان ایرانی در سراسر جهان باشیم. از مطالعه نشریات انجمن انفورماتیک ایران نیز خوشحال خواهم شد. دکتر کیومرث زمانیان
مرکز تحقیقات طراحی مهندسی
دانشگاه کارنگی-ملون
پیتسبورگ، پنسیلوانیا

دانشگاه جدید

به خاطر تأسیس انجمن انفورماتیک ایران از شما تشکر می‌کنیم. دو سؤال داریم:

کامپیوشعر

آقا یا خانمی تحت عنوان «یک فارغ التحصیل رشته ادبیات فارسی» و با علامت اختصاری الگدس به قول خودشان این «شش شعرکامپیوتری» (به انگلیسی Six Compoems) را به انجمن فرستاده اند و آنها را به «تمامی برویجه های کامپیوتر» تقدیم کرده اند:

۱
مونیتور دلم را
روشن می کنم.
نقش گرافیکی عشقت
دیس پلی می شود!

۲
می شناسم
پردازنده هایت را.
با اسمبلی دلت
سخن می گویم!

۳
دیتاهای اشتیاقم را
به طریق اسکی
روانه PC دلت کردم.
سیگنال دریافت را ارسال کن!

۴
از چشمان توست
مشکل سخت افزاری من
ترکیبهای منطقی دلم را
در هم ریخته ای!

۵
ویروس عشق
استک ذهنم را درهم ریخته است.
دوچندان شده است
اکسس تایم من!

۶
بایت های غربت
حافظه ام را پر کرده است.
برای یافتن تو
جامپ می کنم!



۱- آیا اشخاص دیگری هم به این پست الکترونی دسترسی دارند؟

۲- آیا در ایران دانشگاه جدیدی در دست ساختمان است؟ ما شنیده ایم که داشتند یک دانشگاه بین تهران و قزوین می ساختند.

با آرزوی موفقیت در کوششهایتان.

علیرضا آسیائی و رویا صالحی
دانشگاه واشینگتن
میلتون، واشینگتن

چه کسی آنجاست؟

سلام علیکم. من یک دانشجوی ایرانی در امریکا هستم و می خواهم بدانم آیا این پیام سر از ایران درمی آورد یا نه. اگر آدرس این پست الکترونی در ایران است لطفاً هر که هستید به من جواب دهید، بسیار ممنون می شوم. اگر در ایران هستید آیا می توانید اطلاعات بیشتری در مورد اینکه چگونه می توانیم با دوستانمان در دانشگاههای ایران تماس بگیریم به ما بدهید؟
علی زندیه
دانشگاه مریلند

تبریک

مایلم به مناسبت این موفقیت بزرگ به همه تبریک بگویم. این نامه را برای آزمایش و نیز تبریک گفتن به همه می فرستم.
امیر تقوی
دانشگاه ایالتی اهایو

تشکر

• آقای دکتر ناصر مدیری از ردوود سیتی (Redwood City) در ایالت کالیفرنیا آمریکا شش مقاله نوشته خود را برای انجمن فرستاده اند. بدین وسیله از ایشان تشکر می کنیم و امیدواریم ترجمه برخی از مقالات ارسالی ایشان را بزودی در گزارش کامپیوتر چاپ کنیم.

• از دوستانی که نامشان در زیر می آید به خاطر اهدای کتب و نشریات به کتابخانه انجمن تشکر می کنیم:

• آقای امیرسعید تابنده (یکسال اشتراک مجله C Users' Journal)
• خانم لادن اصفهانی (سه نسخه از آخرین شماره های مجله Personal Computer World)
• آقای نادر انوری زاده (یک نسخه از شماره های اخیر مجله PC World)

سخن رئیس

سالی نوید بخش

انجمن در مورد معرفی اینجانب به عنوان مدیر مسئول نشریه گزارش کامپیوتر موافقت کرد و پروانه انتشار این نشریه به نام انجمن انفورماتیک ایران از سوی وزارت فرهنگ و ارشاد اسلامی صادر و به شماره ۱۰۲/۸۵۷۹ مورخ ۷/۱۲/۷۰ ثبت گردید. این مهم که با حسن نظر مسئولان فعلی اداره کل مطبوعات داخلی وزارت فرهنگ و ارشاد اسلامی برای انجمن فراهم آمده، زمینه را برای فعالیت کمیته انتشارات انجمن هموارتر می نماید. و بسیاری از مشکلات اجرایی انتشارات گزارش کامپیوتر را تخفیف می دهد. امیدواریم بدین ترتیب، دیگر وقفه ای در کار انتشار بیگیر و مستمر نشریه پیش نیاید.

برنامه ریزی برای برگزاری دو سمینار با عناوین «مهندسی نرم افزار» و «شبکه های کامپیوتری» در سال جدید، سرعت بخشیدن به کار انتشار نشریه، ادامه برگزاری سخنرانیهای علمی ماهانه و برنامه ریزی برای برگزاری دوره های آموزشی تخصصی از سایر فعالیتهای هیأت اجرایی در سال ۱۳۷۰ است که نوید بخش سالی پر بار و پرتحرک می باشد. البته لازم به یادآوری چندباره است که هیأت اجرایی فقط متکی به همکاری اعضای انجمن است و در صورت عدم همکاری و همفکری و پشتیبانی لازم، هیأت اجرایی به خودی خود قادر به انجام این امور نیست.

در خاتمه، از فرصت استفاده کرده مجدداً فرارسیدن سال نو را به تمامی اعضای گرامی انجمن تبریک می گویم و از همه آنان تقاضا می کنم در سال جدید همکاری بیشتر و پرثمرتری با هیأت اجرایی منتخب خود داشته باشند تا انجمن بتواند نقش فعالتری در جامعه انفورماتیک کشور ایفا نماید. از کلیه انجمنهای علمی، شرکتهای کامپیوتری و اعضای گرامی انجمن نیز که با ارسال کارت تبریک، فرارسیدن سال نو را به هیأت اجرایی انجمن تبریک گفته اند صمیمانه قدردانی و تشکر می کنم و متقابلاً سالی سرشار از موفقیت برای آنان آرزو می نمایم.

رئیس انجمن

فرارسیدن سال نو را از طرف هیأت اجرایی انجمن به تمامی اعضای گرامی تبریک می گویم برای آنان سالی سرشار از موفقیت و شادکامی آرزو می کنم. گزارش فعالیتهای یکساله انجمن همزمان با برگزاری مجمع عمومی سالانه در مهرماه هر سال ارائه می گردد و من قصد ندارم در اینجا به تکرار آنها بپردازم ولی می خواهم به بهانه پایان سال ۱۳۷۰ و فرارسیدن سال جدید به چند جنبه بارز از فعالیتهای سال گذشته انجمن اشاره کنم. نخست، برگزاری انتخابات هیأت اجرایی و برگزیده شدن اعضای هفتمین هیأت اجرایی انجمن است. برای من که در تمامی شش دوره گذشته نیز افتخار خدمتگزاری و همکاری داشته ام، ترکیب هیأت اجرایی فعلی بسیار امیدبخش است. تقریباً همه جلسات هیأت اجرایی با شرکت کلیه اعضای اصلی و علی البدل تشکیل می گردد و همگی با اشتیاق قابل توجهی در صدد پیشبرد اهداف انجمن و زمینه سازیهای لازم هستیم. کوشش اصلی هیأت اجرایی، در جهت رفع مشکل مالی که همچنان به عنوان بلندترین مانع بر سر راه گسترش فعالیتهای انجمن خودنمایی می کند، معطوف گشته است و برای این منظور تمامی راههای داخلی و خارجی تأمین سرمایه اولیه لازم در نظر گرفته شده اند. امید فراوان داریم که در سال ۱۳۷۱ بتوانیم دفتر و مرکز آموزش انجمن را تأسیس کنیم و بدین ترتیب، بودجه مستمری را برای ادامه و گسترش سایر فعالیتهای علمی انجمن فراهم آوریم.

انتخاب یکی از اعضای علاقمند انجمن به عنوان عضو رابط خارج از کشور، از دیگر فعالیتهای بارزی است که برای اولین بار از سوی هیأت اجرایی صورت گرفت. فعالیت و همکاری چشمگیر آقای مهندس حسینی در شناساندن انجمن و جلب اعضای جدید از بین متخصصین ایرانی خارج از کشور تاکنون بسیار ثمر بخش بوده است و امیدواریم در سال جدید نیز ادامه و گسترش یابد.

در آخرین روزهای سال ۱۳۷۰، بالاخره پس از گذشت چند سال، هیأت نظارت مطبوعات در جلسه مورخ ۷۰/۱۱/۲۸ خود با تقاضای



مقاله

مجموعه‌های مشکک

محسن رئیس قاسم*

مقدمه

ولی در مورد اعداد ۶، ۷ و ۸ چه می‌توان گفت. بدیهی است که نمی‌توان راجع به عضویت یا عدم عضویت این اعداد در مجموعه مورد نظر هیچ اظهار نظر قاطعی کرد. از سوی دیگر وجود عدد ۶ در این مجموعه از وجود اعداد ۷ و ۸ پذیرفتنی‌تر است یا به عبارت دیگر امکان اینکه ۶ نفر در پیکان سوار شوند از سوار شدن ۷ و ۸ نفر بیشتر است.

این مفهوم باورکردنی بودن عضویت در مجموعه در نهایت به مفهوم درجه عضویت^۲ و نظریه امکان^۳ منجر می‌شود. توجه کنید درجه عضویت احتمال نیست بلکه میزان همسازی و همگونی^۴ شیئی را با مفهوم مجموعه بیان می‌کند. برای مثال اگر درجه عضویت عدد ۷ برابر ۰/۶ باشد آنگاه این عدد بیان می‌کند که عدد ۷ تا چه میزان با عددی که نمایانگر تعداد سرنشینان ماشین پیکان است، انطباق دارد.

تعمیمی از مجموعه‌های مرسوم روی مفهوم «عضویت» به گونه‌ای که برد تابع مشخصه به جای مجموعه $\{0, 1\}$ بازه $[0, 1]$ باشد به مفهوم مجموعه مشکک^۵ منجر می‌شود. تابع مشخصه در این مجموعه‌ها تابع عضویت^۶ خوانده می‌شود. در زیر تعریف رسمی این مجموعه‌ها و برخی عملیات معمول آنها آورده شده‌اند.

تعریف ۱: اگر $X = \{x\}$ گروهی از اشیا باشند که به طور کلی با X نشان داده می‌شوند، آنگاه مجموعه مشکک A در X مجموعه‌ای است از زوجهای مرتب $\{(x, \mu_A(x)) \mid x \in X\}$ که در آن $\mu_A(x)$ تابع عضویت یا درجه عضویت است. برد تابع عضویت زیرمجموعه‌ای است از اعداد حقیقی غیر منفی که کران بالای آن محدود است. اگر این کران بالا برابر ۱ باشد آنگاه مجموعه A ، مجموعه مشکک نرمال خوانده می‌شود.

تعریف ۲: تابع عضویت اشتراک دو مجموعه مشکک A و B به صورت زیر تعریف می‌شود (شکل ۱):

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)); x \in X$$

^۲ degree of membership

^۳ possibility theory

^۴ compatibility

^۵ fuzzy set

^۶ membership function

تاریخ علم گواهی بر تمایل بشر به برخورد کمی با مفاهیم، پدیده‌ها و روشهاست. با ظهور کامپیوتر به عنوان ابزار قدرتمندی برای پردازش کمیتها، این تمایل شتاب بیشتری گرفته است. نظریه مجموعه‌ها یکی از مؤثرترین ابزارهای ریاضی برای فراهم آوردن دیدگاهی کمی از پدیده‌هاست. دکتر لطفی‌زاده در سال ۱۹۶۵ پیشنهاداتی برای بسط این نظریه و تعریف مجموعه‌های جدید ارائه کرد که به اعتقاد وی در مواردی که بیان دقیق مسئله مشکلاتی دربر دارد، از مجموعه‌های مرسوم کارآترند. این نظریه به دلیل قابلیت‌های زیاد در مدل‌سازی مسائل نادقیق و دستاوردهای کاربردی آن توجه زیادی را به خود جلب کرده است. این نوشته سعی در شناسایی مقدماتی این نظریه و اشاره‌ای به برخی موارد کاربرد آن دارد.

مجموعه و مجموعه‌های مشکک

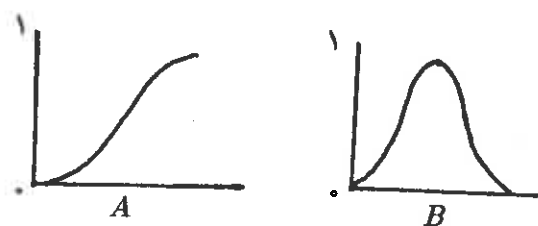
طبق تعریف، مجموعه تجمعی است از اشیاى دنیای U . به ازای هر مجموعه دلخواهی مثل A می‌توان تابعی تعریف کرد که به ازای هر عنصری از U ، عضویت یا عدم عضویت آن را در مجموعه A تعیین کند. چنین تابعی را تابع مشخصه^۱ می‌نامند و به صورت زیر تعریف می‌شود:

$$Char_A(x) = \begin{cases} 0 & x \in A \\ 1 & x \notin A \end{cases}$$

این تابع در واقع نگاشتی است از U به مجموعه $\{0, 1\}$ و روشن است که در عضویت یا عدم عضویت عنصری در مجموعه مورد نظر هیچ جای ابهامی باقی نمی‌گذارد.

فرض کنید بر اساس تعریف فوق بخواهیم مجموعه‌ای از اعداد طبیعی را که می‌توانند به عنوان تعداد سرنشینان یک ماشین پیکان در نظر گرفته شوند، تعریف کنیم. در عضویت اعداد ۰، ۱، ۲، ۳، ۴ و ۵ و عدم عضویت اعداد ۹، ۱۰ و اعداد بزرگتر هیچ ابهامی وجود ندارد.

* آقای رئیس قاسم لیسانس نرم‌افزار از دانشگاه شهیدبهشتی و فوق لیسانس نرم‌افزار از دانشگاه صنعتی شریف دارند و فعلاً در دفتر طراحی و ایجاد سیستمهای سازمان برنامه و بودجه مشغول به کار هستند.

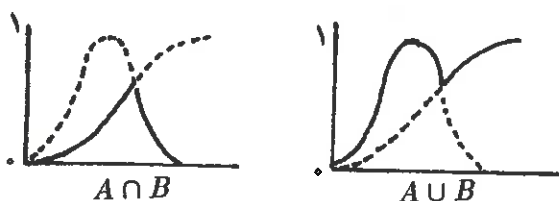


تعریف ۳: تابع عضویت اجتماع دو مجموعه مشکک A و B به صورت زیر تعریف می شود (شکل ۱):

$$\mu_D(x) = \max(\mu_A(x), \mu_B(x)); x \in X$$

تعریف ۴: تابع عضویت متمم یا نفی مجموعه مشکک A به صورت زیر تعریف می شود (شکل ۲):

$$\mu_{A'}(x) = 1 - \mu_A(x)$$



شکل ۱- اجتماع و اشتراک مشکک

در نظریه مجموعه های مشکک کارهای زیادی برای تعیین روشهایی برای تعیین درجه عضویت صورت گرفته است ولی به هر حال معمولاً تعیین درجه عضویت کاری است وابسته به زمینه مورد نظر و مشکل. در بسیاری از کاربردها می توان درجه عضویت مجموعه مشککی را با استفاده از اعداد مشککی مثل «نزدیک به ۰/۸»، «تقریباً ۰/۶»، «کاملاً زیاد» یا «خیلی کم» بیان کرد. چنین مجموعه مشککی را که تابع عضویت آن از مقادیر مشکک استفاده می کند، مجموعه فوق مشکک^۷ می نامند.

الگوریتم مشکک

بر خلاف الگوریتم قطعی^۸ و الگوریتم غیرقطعی^۹، الگوریتم مشکک می تواند حاوی احکام و جملات مشکک باشد. برای مثال به دستورالعمل زیر توجه کنید:

اگر x تقریباً برابر ۵ است آنگاه y را تقریباً مساوی ۱۰ قرار دهید.

منبع شک در این دستورالعمل، زیرمجموعه های مشککی است که با حروف خمیده نوشته شده اند. ما در زندگی روزمره خود با دستورالعملهای زیادی از قبیل دستورات آشپزی، تغییر کانال تلویزیون برای دریافت تصاویر بهتر، نحوه پارک کردن ماشین و غیره سروکار داریم که معمولاً شان آن را نمی یابند که به عنوان الگوریتم در نظر گرفته شوند ولی زاده می گوید که آنها می توانند اشکال بسیار ابتدایی از الگوریتمهای مشکک باشند.

با استفاده از مفهوم تابع عضویت مجموعه های مشکک می توان به یک دستورالعمل مشکک معنی دقیقی را نسبت داد. ولی به هر حال ابهام

نکته جالب توجه در این تعریف آن است که ارزش درستی S and not S با فرض اینکه ارزش درستی S، s باشد آنگاه چنین خواهد شد $\min(s, 1-s) \neq 0$ که این برخلاف دانسته های ما از منطق دو ارزشی است ولی زاده استدلال می کند که ما گاهی در توصیف رنگ یک گل سرخ زیبا می گوئیم هم سرخ هست و هم سرخ نیست. معمولاً دو مشخصه اصلی برای مجموعه های مشکک ذکر می کنند:

۱- تابع عضویت و عملیات مشکک نقشی قاطع در این نظریه ایفا می کنند.

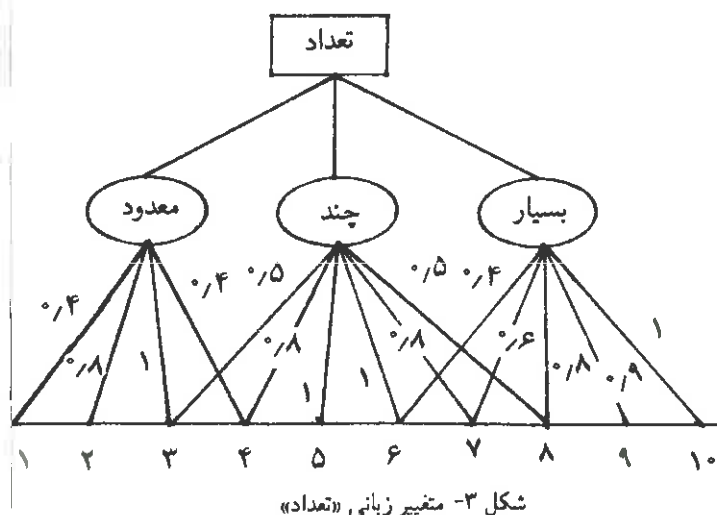
۲- نظریه مجموعه های مشکک در اساس نظریه ای است بسیار کلی، انعطاف پذیر و صوری و اگر بخواهد در کاربردهای واقعی مورد استفاده قرار گیرد، بایستی به دقت با کاربرد مربوطه منطبق شود. این تطبیق شامل تعریف توابع عضویت مجموعه ها و عملیات روی آنها است.

درجه عضویت

درجه عضویت در اصل بیش از آنکه مقوله ای قابل اندازه گیری باشد مفهومی است که به تعریف، طبیعت و ماهیت مفهوم مورد نظر بستگی دارد. زاده معتقد است که انسانها به طریقی که هنوز مشخص نشده است توانایی قابل توجهی در تعیین درجه عضویت اشیا در مجموعه های مورد نظر خود دارند. برای مثال تقریباً همه ما تصوره های مشابهی از «مرد بلند قد» و «اتاق بزرگ» داریم.

تابع عضویت یک مجموعه مشکک یا در واقع توزیع امکان آن معمولاً یا بر پایه محدودیتهای فیزیکی اعمال شدنی یا از ماهیت و منشأ مفهوم مورد نظر تعیین می شود. برای مثال در مجموعه تعداد افرادی که می توانند در یک ماشین پیکان سوار شوند، محدودیت فیزیکی را می توان «سهولت سوار شدن سرنشینان» دانست. روشن است که استناد به چنین محدودیتهایی برای مفاهیمی مثل سردی، گرمی، بلندی قد و رنگ سبز چندان روشن نیست و در چنین مواردی است که از تعریف و ماهیت مفهوم مورد نظر استفاده می شود.

ultra fuzzy set	۷
deterministic	۸
non-deterministic	۹



شکل ۳- متغیر زبانی «تعداد»

که درک اینکه در بسیاری مسائل واقعی، جستجو برای الگوریتمهای غیر مشکک عملی نیست، می‌تواند به الگوریتمهای مشکک احترام شایسته‌ای را که اکنون فاقد آن است اعطا کند.

بالاخره آخرین سوالی که مطرح می‌شود ارتباط بین الگوریتم مشکک و برنامه مکاشفه‌ای^{۱۲} است. زاده بیان می‌کند که در عمل برنامه مکاشفه‌ای تقریبی است غیر مشکک از الگوریتم مشکک که در یک زبان برنامه‌سازی بیان شود.

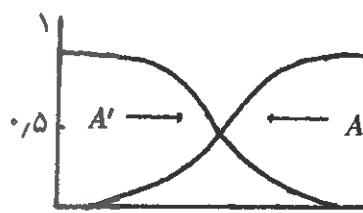
متغیرهای زبانی

مفهوم پایه‌ای در منطق مشکک که نقشی کلیدی در بسیاری از کاربردهای آن مثل کنترل مشکک و سیستمهای خبره مشکک ایفا می‌کند، متغیر زبانی^{۱۳} است. یک متغیر زبانی متغیری است که مقادیر آن کلمات یا جملاتی از یک زبان طبیعی یا ساختگی^{۱۴} است. برای مثال «سن» یک متغیر زبانی است که مقادیر «جوان»، «نوجوان»، «میان‌سال»، «خیلی جوان»، «پیر» و غیره را می‌پذیرد.

یک متغیر زبانی را از سوی دیگر می‌توان برچسب^{۱۵} یا عنوان یک مجموعه مشکک دانست. شکل ۳ متغیر زبانی «تعداد»، برخی مقادیر آن و درجه‌های عضویت متناظر با اعداد ۱ تا ۱۰ را نشان می‌دهد. برای مثال عدد ۳ با درجه عضویت ۱ در زیر مجموعه مشکک «معدود» و با درجه عضویت ۰/۵ در زیر مجموعه مشکک «چند» مشخص شده است.

به‌طور کلی یک متغیر زبانی می‌تواند از یک لفظ اولیه^{۱۶} مثل «جوان»، متضاد آن مثل «پیر»، تعدادی پیراینده^{۱۷} مثل «خیلی»، «نه خیلی»،

heuristic	۱۲
linguistic variable	۱۳
synthetic	۱۴
label	۱۵
prime term	۱۶
modifier	۱۷



شکل ۲- متمم (نفی) مشکک

در اجزای چنین دستورالعملهایی همچنان باقی است. برای ملموستر شدن مطلب جمله ساده «چند گام به جلو بروید» را در نظر بگیرید و فرض کنید تابع عضویت مجموعه مشکک «چند گام» (مجموعه A) به صورت زیر تعریف شده باشد:

$$\mu_A(0) = \mu_A(1) = \mu_A(2) = \mu_A(3) = 0$$

$$\mu_A(4) = 0/8$$

$$\mu_A(5) = \mu_A(6) = \mu_A(7) = 1$$

$$\mu_A(8) = 0/7$$

$$\mu_A(x) = 0; x \geq 9$$

توجه کنید که مجری الگوریتم حتی با در دست داشتن مقادیر $\mu_A(x)$ نمی‌داند فرمان فوق را چگونه اجرا کند و سرانجام چه مقدار جلو برود. به‌طور کلی دو شیوه اجرایی وجود دارد: اجرای احتمالاتی^{۱۸} و اجرای غیرقطعی به همراه یک مقدار آستانه^{۱۹}. در شیوه اول فرض می‌شود که x به احتمال $\mu_A(x)$ انتخاب می‌شود. بنابراین در مثال فوق مقادیر ۰، ۱، ۲ و ۳ با احتمال صفر، مقدار ۴ با احتمال ۰/۸/۴/۵، مقدار ۵، ۶ و ۷ با احتمال ۱/۴/۵، مقدار ۸ با احتمال ۰/۷/۴/۵ و بقیه مقادیر با احتمال صفر انتخاب می‌شوند. در روش دوم مقدار α به عنوان آستانه انتخاب می‌شود و مجموعه A_α از مقادیر x به گونه‌ای انتخاب می‌شود که $\mu_A(x) \geq \alpha$. در این حالت هر x متعلق به A_α (مشابه الگوریتمهای غیرقطعی) می‌تواند یک انتخاب مجاز باشد.

بدیهی است که وجود دقت و یقین ایده‌آل، شک و نایقینی محدود است ولی از نظر زاده باید پذیرفت که تلاش برای دستیابی به دقت هرچه بیشتر در علوم محض و علوم کاربردی، منجر به نادیده گرفتن این حقیقت شده است که گروه مسائل غیرجزئی‌ای که انسان می‌تواند برای آنها راه‌حلهای الگوریتمی پیدا کند، بسیار محدود است. متأسفانه اکثر مسائل واقعی تمایل به پیچیدگی دارند و مسائل بسیار پیچیده یا به شکل الگوریتمی حل ناشدنی هستند یا اگر در اصول قابل حل باشند، از نظر محاسباتی دور از دسترس هستند. زاده معتقد است

probabilistic	۱۸
threshold	۱۹

پایگاه داده‌ای فرضی را که حاوی اطلاعاتی در زمینه مورد نظر هستند، انتخاب کنیم. فرض می‌کنیم این رابطه‌ها چنین باشند:

اقامت (اسم فرد، اسم شهر)، برای تعیین محل اقامت افراد.

جمعیت (اسم شهر، جمعیت)، برای تعیین جمعیت هر شهر.

کوچک (جمعیت، درجه عضویت)، برای تعیین درجه عضویت هر شهر با جمعیت مشخص در مجموعه مشکک «کوچک».

نزدیک (اسم شهر، اسم شهر، درجه عضویت)، برای تعیین درجه عضویت دو شهر مشخص در مجموعه مشکک «نزدیک».

محدودیت‌های اعمال شده توسط این جمله، کوچک بودن شهر محل اقامت حسن و نزدیکی آن با گرگان است.

آزمونهای لازم برای هر کدام از این محدودیتها عبارتند از:

۱- آزمون محدودیت اعمال شده از طرف «کوچک» از طریق پیدا کردن اسم محل اقامت حسن (رابطه «اقامت») و پیدا کردن جمعیت آن (رابطه «جمعیت») و رجوع به رابطه «کوچک» و تعیین درجه عضویت این جمعیت در مجموعه «کوچک» (میزان انطباق شهر محل اقامت حسن با مفهوم شهر کوچک).

۲- آزمون محدودیت اعمال شده از طرف «نزدیک» از طریق رابطه «نزدیک» برای مقادیر شهرهای گرگان و محل اقامت حسن (میزان انطباق این دوشهر با مفهوم شهرهای نزدیک).

و بالاخره همگون کردن این دو مقدار می‌تواند به صورت پیدا کردن بزرگترین مقدار آنها تعریف شود.

بنابراین در صورت وجود چنین پایگاه‌های داده‌ای میزان درستی یا صحت جمله مورد نظر تعیین می‌شود. ولی مهمتر از آن این است که ذکر این موارد و تعیین روابط اطلاعاتی لازم و آزمونهایی برای تعیین میزان برآورده شدن محدودیتها می‌تواند بازگوکننده مفهوم و معنی جمله باشد.

زاده معتقد است که یک عامل انسانی می‌تواند هر گزاره زبان طبیعی را به این شکل بازنمایی کند ولی نوشتن برنامه‌ای که بتواند چنین کاری را انجام دهد بسیار پیچیده و مشکل است. به علاوه به نظر می‌رسد تعریف روابط پایگاه‌های اطلاعاتی‌ای که بتوانند در برگیرنده اطلاعات

«کم و بیش» و «کاملاً» و حرف ربط «و» و «یا» تشکیل شود. از آنجا که برای بسیاری از پیراینده‌ها توابع عمل‌کننده معادل روی مجموعه‌های مشکک تعریف می‌شود، آنگاه با در دست داشتن تعریف یک مجموعه مشکک متناظر با یک متغیر زبانی، می‌توان ترکیبات آن با این پیراینده‌ها را به دست آورد. شکل ۴ منحنی‌های توابع عضویت را برای «سن» و ترکیبات آن را برحسب میزان سالهای عمر نشان می‌دهد.

برنامه مکاشفه‌ای تقریبی است غیر مشکک از الگوریتم مشکک که در یک زبان برنامه‌سازی بیان شود.

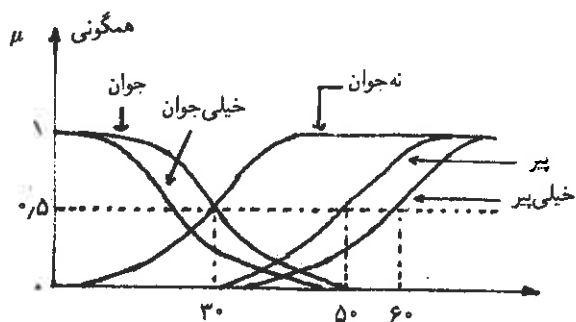
بازنمایی معنی

زاده برای بازنمایی گزاره‌های معنایی نهفته در زبان طبیعی روشی را پیشنهاد می‌کند که بر اساس مفهوم معنی‌شناسی ثبت-آزمون^{۱۸} قرار دارد. ایده اصلی در این نظریه آن است که هر نمود زبانی مثل گزاره، سؤال یا فرمان به طور کلی از نظر تأثیر مشابه است با تحمیل محدودیت‌های انعطاف‌پذیر روی مجموعه‌ای از اشیا و روابط دنیای مورد بحث. همیشه سعی می‌شود که هر گزاره‌ای مثل P در زبان طبیعی به شکل متعارف^{۱۹} $P \equiv N \text{ is } F$ (است) تبدیل شود که در آن N توزیع امکان مجموعه‌ای از متغیرهایی است که به طور صریح یا ضمنی در P قرار دارند و F مجموعه‌ای مشکک است که محدودیتی را اعمال می‌کند. برای مثال جمله «حسن دارای یک دفتر کار وسیع است» ابتدا به شکل متعارف خود «دفتر کار حسن وسیع است» تبدیل می‌شود. در این جمله متغیرهای مبنایی ضمنی عبارتند از طول و عرض دفتر کار حسن؛ و محدودیت اعمال شده «وسیع» بودن است که روی مقادیر ممکن طول و عرض این دفتر کار بیان می‌شود.

در روش پیشنهادی زاده معنی یک نمود زبانی در طی موارد زیر تعریف می‌شود:

- تعیین محدودیت‌هایی که نمود زبانی اعمال می‌کند.
- تعیین آزمونهایی که میزان برآورده شدن محدودیت‌های مذکور را نشان می‌دهند.
- مشخص کردن شیوه‌ای برای همگون کردن ثبت-آزمون‌های جزیی به دست آمده از آزمونها و تعیین ثبت-آزمون سراسری.

این موارد برای جمله «حسن در شهر کوچکی نزدیک گرگان اقامت دارد» نشان داده می‌شوند. در اولین گام بایستی رابطه‌هایی از یک



شکل ۴- مقادیری از متغیر زبانی «سن»

^{۱۸} test-score semantics
^{۱۹} canonical

جمله‌های نسبتاً نامحدودی از زبان باشند چندان عملی نباشد.

در سیستم‌های خبره هستند. پوسته‌های^{۲۱} متعددی برای سیستم‌های خبره بر مبنای منطق مشکک به بازار عرضه شده‌اند.

دورنما و کاربردها

مجموعه مشکک در طی سالیان گذشته در زمینه‌های متعددی از مسائل مالی گرفته تا مهندسی زمین لرزه کاربرد پیدا کرده‌است. برخی از این زمینه‌های کاربردی عبارتند از:

از آنجا که اکثر قواعد سیستم‌های خبره
مقدم و تالی مشکک دارند، بنابراین
سیستم‌های خبره زمینه کاربردی مناسبی
برای مجموعه‌های مشکک هستند.

کنترل فرایند

اولین کاربرد مجموعه‌های مشکک در این زمینه صورت گرفت. مددانی و اصیلیان در سال ۱۹۷۴ اولین پیاده‌سازی یک سیستم کنترل فرایند^{۲۰} مبتنی بر مجموعه‌های مشکک را ارائه کردند. یک نمونه موفق تجارتی دیگر کنترل کننده کوره سیمان است که در سال ۱۹۸۰ به بازار عرضه شد و در بسیاری کوره‌ها مورد استفاده قرار گرفت. دلیل این موفقیت آن است که منطق مشکک این امکان را فراهم می‌آورد که قواعد کنترلی مرسوم در محیط‌های عملیاتی به همان زبان نادقیق برای سیستم تشریح شوند. بنابراین متصدیان دستگاه‌ها به سادگی و بدون نیاز به هیچ اطلاعات کامپیوتری می‌توانند قواعد کنترلی جدیدی وضع کنند یا قواعد موجود را اصلاح کنند. روشن است که تقلید و مشابه‌سازی کامپیوتری قواعد کنترلی مورد استفاده انسان، به کمک منطق مشکک تا حد زیادی تسهیل می‌شود.

سیستم‌های کنترلی متعددی در چین براساس منطق مشکک طرح شده‌اند که بیشتر آنها دارای قابلیت اعتماد و کارایی بیشتری نسبت به سیستم‌های مرسوم بوده‌اند.

سیستم‌های خبره

از آنجا که اکثر قواعد سیستم‌های خبره مقدم و تالی مشکک دارند، بنابراین سیستم‌های خبره زمینه کاربردی مناسبی برای مجموعه‌های مشکک هستند. در حقیقت قواعد کوتاه و خلاصه، معمولاً مشکک هستند. اگر نخواهیم از منطق مشکک استفاده کنیم آنگاه هر کدام از چنین قواعدی به تعدادی قواعد دقیق‌تر تبدیل می‌شود. بنابراین مجموعه‌های مشکک دارای تاثیر مثبت در کاهش تعداد قواعد مورد نیاز

بینایی

زمینه کاربردی مهم دیگر بازشناسی الگوها^{۲۲}، بینایی^{۲۳} کامپیوتری و روبوتیک است. در بازشناسی الگوها و بینایی، سیستم از مشخصه‌های بسیاری مثل اندازه، ابعاد و شکل برای تعیین شکل یا الگو استفاده می‌کند. استفاده از مشخصه‌های مشکک در تعریف وجوه تمایز می‌تواند کارایی را افزایش دهد. محققین مؤسسه صنعتی سیرنیتیک و روبوتیک در صوفیه بلغارستان روبات جوشکاری ساخته‌اند که می‌تواند با استفاده از بینایی کامپیوتری مبتنی بر منطق مشکک یک شکاف را دنبال کند.

در اکثر کاربردهای جاری از نرم‌افزار به عنوان واسطی برای پیاده‌سازی الگوریتم‌ها و قواعد کنترلی مشکک استفاده می‌شود. بدیهی است که استفاده از تراشه‌های مبتنی بر منطق مشکک و در نهایت کامپیوترهای مشکک بسیار ارزانتر و کاراتر خواهد بود. اولین تراشه مشکک در سال ۱۹۸۵ توسط واتانابه^{۲۴} و توگای^{۲۵} در آزمایشگاه‌های بل^{۲۶} ساخته شد. این تراشه برای انجام استنتاجات منطقی مشکک ساخته شده و دارای چهار بخش است: حافظه حاوی قواعد، پردازشگر استنتاجی، کنترل کننده و مدارهای ورودی-خروجی. در یک پیاده‌سازی، حافظه مجموعه قواعد به کمک تعدادی حافظه با دستیابی دلخواه^{۲۷} ساخته شد و پردازشگر دارای ۱۶ مسیر داده‌ای بود. هر ۱۶ قاعده موجود در تراشه به صورت موازی اجرا می‌شدند و بعد از ۶۴ چرخه زمان سنج^{۲۸}، خروجی به دست می‌آمد. این بدان معنی است که سرعت اجرایی آن در حدود ۲۵۰۰۰۰ استنتاج مشکک در ثانیه^{۲۹} در زمان سنجی با فرکانس ۱۶ مگاهرتز است. در این پیاده‌سازی متغیرهای کنترلی در برد مجموعه محدودی با حداکثر ۳۱ عنصر تغییر می‌کنند و تابع عضویت در ۱۶ سطح (۰ تا ۱۵) تغییر می‌کند. طرح یک کامپیوتر مشکک از طرف تاماکاوا^{۳۰} در دانشگاه کوماموتو^{۳۱} ژاپن اعلام شده است. این کامپیوتر که از معماری موازی استفاده می‌کند قادر به انجام استنتاج مشکک با سرعتی بسیار بالای ۱۰ میلیون فلیپس است. زاده معتقد است که کامپیوتر تاماکاوا می‌تواند گامی باشد به سوی

shell	۲۱
pattern recognition	۲۲
vision	۲۳
Watanabe	۲۴
Togai	۲۵
Bell	۲۶
Random Access Memory-RAM	۲۷
clock	۲۸
Fuzzy Logical Inference Per Second-FLIPS	۲۹
Tamakawa	۳۰
Kumamoto	۳۱

تصحیح

با وجود تمام سعی و تلاشی که در ارائه بی‌عیب و نقص گزارش کامپیوتر داریم، باز هم متأسفانه شاهد بعضی اشکالات چاپی و غیرچاپی هستیم. ضمن معذرت‌خواهی لطفاً موارد زیر را در شماره‌های قبل تصحیح کنید:

• در قسمت اول مقاله «زبانهای برنامه‌سازی» که در شماره قبل به چاپ رسید، در صفحه ۱۳، ستون اول، بعد از سطر ۱۹ خط زیر افزوده شود:

$$Sum(n) = Sum(n - 1) + n \quad n > 0 \text{ برای مقادیر}$$

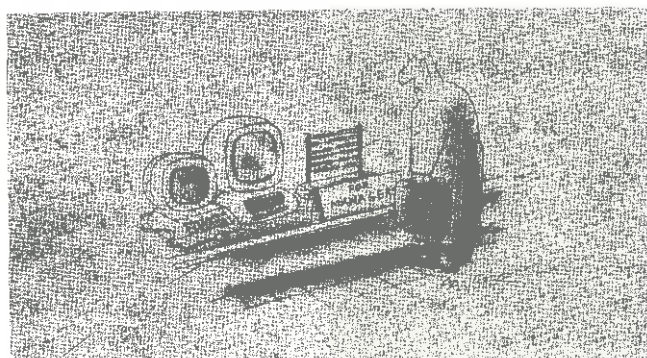
• در قسمت دهم مقاله نفرماه افسانه‌ای که در شماره ۱۰۹ به چاپ رسید، یک جمله از متن از قلم افتاده بود. صفحه ۳۴، ستون اول، سطر سوم از آخر را به این شکل تصحیح کنید: «... از آنجا که وظیفه اصلی مدیر هدایت افراد در یک جهت واحد است، کار اصلی روزمره او نیز باید ارتباط‌گیری باشد و نه تصمیم‌گیری».

• در شماره ۱۰۹ در ترازنامه سال مالی ۷۰-۱۳۶۹ انجمن، درآمد انباشته اول دوره اشتباهاً مبلغ ۴۸۹۳۱۰ ریال ذکر شده که به رقم ۴۸۹۳۷۰ ریال تصحیح می‌شود. ضمناً بر اثر این تغییر مبلغ مازاد کل نیز از ۵۶۵۲۰۱ ریال به ۵۶۵۲۶۱ ریال تبدیل خواهد شد.

به نقل از نشر دانش (آذر و دی ۱۳۷۰):

• گزارش کامپیوتر، ماهنامه «انجمن انفورماتیک ایران» است که تاکنون نزدیک به صد و ده شماره آن انتشار یافته است و مجموعه سال دوازدهم آن (شماره‌های ۱۰۰ تا ۱۰۵) چندی پیش به دفتر نشر دانش رسیده است.

هر شماره گزارش کامپیوتر حاوی تعدادی خبر و گزارش تخصصی و مقاله تحقیقی داخلی و خارجی است. انجمن انفورماتیک ایران در سیزده سال پیش تأسیس شده و در پایان سال ۱۳۶۹ بیش از ۶۵۰ عضو داشته است. جای تأسف است که این نشریه علمی سودمند در تعدادی اندک منتشر می‌شود و به دست همه علاقه‌مندان نمی‌رسد.



کامپیوترهای نسل ششم که قادر به پردازش معارف عقل سلیم^{۳۲} خواهند بود. این قابلیت پیش‌نیازی است برای حل بسیاری از مسایل موجود در هوش مصنوعی مثل بازشناسی گفتار، بازشناسی متون دست‌نوشته، ترجمه ماشینی، تلخیص^{۳۳} و درک تصاویر که در محدوده تکنولوژی مرسوم، به راه‌حل‌های ارزان و کارایی منجر نمی‌شوند.

مراجع و منابع:

- [1] K.J. Schmucker, "Fuzzy Sets, Natural Language Computations, and Risk Analysis", Computer Science Press, 1982.
- [2] L.A. Zadeh, "Fuzzy Algorithms", Information and Control, 1968.
- [3] L.A. Zadeh, "Test-Score Semantics for Natural Language and Meaning Representation via PRUF", 1981.
- [4] L.A. Zadeh, "Making Computer Think Like People", *IEEE Spectrum*, August 1987.
- [5] L.A. Zadeh, "Fuzzy Logic", *IEEE Computer*, April 1988.
- [6] H.J. Zimmermann, "Operations Research: Fuzzy Set Theory", *Systems and Control Encyclopedia*, vol. 5 M-O, 1987.



مقاله

زبانهای برنامه‌سازی به‌مثابه مدل‌های تفکر (۲)

نوشته پیتز رخنبرگ*
ترجمه علی شکوفنده و عباس نوذری

مدل تفکر منطقی

?- pythagoras(3,4,C)

C = 5

?- pythagoras(A,4,5)

A = 3

این بدین معنی است که اگر یکی از پارامترها متغیر باشد، سیستم برنامه‌سازی سعی در یافتن یک مقدار برای آن پارامتر می‌کند (انتقید متغیر به مقدار) به قسمی که رابطه دارای ارزش درست شود. یک رابطه می‌تواند دارای دو پارامتر متغیر نیز باشد. برای مثال:

?- pythagoras(A,4,C)

A = 3; C = 5

در این حالت، سیستم سعی در انتقید هر دو متغیر به مقادیری می‌کند که رابطه دارای ارزش درست شود. جالبترین وضعیت زمانی است که هر سه پارامتر متغیر باشند:

?- pythagoras(A,B,C)

A = 3; B = 4; C = 5;

A = 6; B = 8; C = 10;

A = 5; B = 12; C = 13;

A = 9; B = 12; C = 15;

A = 8; B = 15; C = 17;

A = 12; B = 16; C = 20;

در صورتی که سیستم درست برنامه‌ریزی شده باشد، اقدام به تعیین تمامی سه تاییه می‌کند (البته تا آنجایی که محدوده مقدار C از لحاظ نمایش داخلی اجازه می‌دهد) به‌طوری‌که رابطه فیثاغورث دارای ارزش درست باشد.

برنامه محاسبه رابطه فیثاغورث در زبان پرولوگ می‌تواند به شکل زیر باشد:

pythagoras (X,Y,Z) :-

natle(Z,20), natle(Y,Z),

natle(X,Y), equal(X*Y+Y*Y,Z*Z).

natle(1,N) :- N > 1.

برنامه‌سازی عملیاتی تنها راه دستیابی به برنامه‌سازی در یک سطح تجریدی بالاتر نیست. برنامه‌سازی منطقی یا رابطه‌ای نیز چنین است. از آنجایی که ممکن است خواننده با برنامه‌سازی منطقی چندان آشنایی نداشته باشد با ذکر یک مثال ساده تا حدودی رئوس این شیوه برنامه‌سازی را مورد بررسی قرار می‌دهیم.

مثال

در مدل تفکر منطقی، محاسبه مقدار c به‌عنوان تابعی از مقادیر a و b، به‌صورت رابطه‌ای بین a، b و c نگریسته می‌شود که می‌تواند درست یا نادرست باشد. برای نمونه قضیه فیثاغورث را در نظر بگیرید:

$$a^2 + b^2 = c^2$$

این قضیه بیانگر رابطه‌ای بین a، b و c است که می‌تواند توسط محمول منطقی زیر بیان گردد:

Pythagoras(a,b,c)

اگر فرض کنیم که یک سیستم برنامه‌سازی منطقی نظیر پرولوگ، قضیه فیثاغورث را «بداند»، استفاده کننده می‌تواند بر روی پایانه عبارتی به شکل زیر را وارد کند:

?- pythagoras(3,4,5)

و سیستم پاسخ زیر را خواهد داد:

true

و صدور یک دستور به صورت:

?- pythagoras(3,4,6)

مقدار false را باز می‌گرداند. به عبارت دیگر، رابطه نظیر فراخوانی یک تابع بولی عمل می‌کند. حالت جالبتر این است که یکی از پارامترها متغیر باشد:

Peter Rechenberg, "Programming Languages as * Thought Models", *Structured Programming*, no. 11, 1990, Springer Verlag.

predicate

Z	Y	X	equal()	
1	1	1	false	
1	1	-		X does not exist
1	-			Y does not exist
2	1	1	false	
2	1	-		X does not exist
2	2	1	false	
2	2	2	false	
...	...	...		
5	4	3	true	first result
...	...	...		
10	8	6	true	second result
...	...	...		

شکل ۱- دنباله مقیدسازیها در خلال محاسبه $\text{pythagoras}(X,Y,Z)$.

استنتاج مورد استفاده در این زبان، دلالت منطقی است:

$$A_1 \wedge A_2 \wedge \dots \wedge A_n \rightarrow B$$

با این تفاوت که قسمت سمت راست عبارت در ابتدا نوشته می‌شود:

$$B : -A_1, A_2, \dots, A_n.$$

و به صورت «B اگر A_1 و $A_2, \dots$ و A_n خوانده می‌شود.

همانگونه که می‌بینید این روش، کاملاً توصیفی است و وظیفه سیستم برنامه‌سازی اجرایی است که مقیدسازی مناسب را (از بین امکانات تقریباً نامحدود) به نحوی بیابد که هدف را درست کند.

لب مطلب در مدل تفکر منطقی چنین است: جهان مرکب است از محمولاتی که برحسب عناصر تشکیل دهنده‌شان می‌توانند ارزش درست یا نادرست داشته باشند. هر محمول یا به صورت یک واقعیت است (درست بلاشرط) و یا درستی مقدارش بستگی به درستی مقادیر محمولات دیگر، با توجه به یک قاعده استنتاج دارد. با در دست داشتن یک مجموعه از واقعیتها، یک مجموعه از قواعد استنتاج و یک محمول هدف متشکل از تعدادی متغیر، برنامه منطقی سعی در یافتن تمام مقید سازیهای ممکن متغیرهای محمول هدف با مقادیر ثابت می‌کند به نحوی که محمول هدف مقدار درست یابد.

این مدل تفکر تشابه زیادی با جهان تفکر یک ریاضیدان دارد که می‌خواهد قضیه‌ای را اثبات کند. واقعیتها به منزله اصول موضوعه هستند. وی باید با اعمال متوالی قواعد استنتاج به روشی دست بیابد که نشان دهد حکم قضیه، پیامد اصول موضوعه است. به این جهت، برنامه‌سازی منطقی شباهت زیادی به اثبات قضیه پیدا می‌کند و بسیاری از نویسندگان تا آن حد پیش رفته‌اند که می‌گویند برنامه‌سازی

$\text{natle}(M,N) :- N > 1, \text{natle}(X, N-1), \text{succ}(X,M).$
 $\text{succ}(M,N) :- \text{equal}(M+1,N).$

برنامه از چهار قاعده تشکیل شده است و هر قاعده شامل محمولات و یک علامت به شکل «:-» است که بیانگر دلالت^۲ است. برای مثال قاعده ۱ به صورت زیر خوانده می‌شود: «محمول $\text{pythagoras}(X,Y,Z)$ درست است هر گاه مقادیری برای Y, X و Z وجود داشته باشند به نحوی که تمام محمولات $\text{natle}(X,Y)$, $\text{natle}(Y,Z)$, $\text{natle}(Z,20)$ و $\text{equal}(X*X+Y*Y,Z*Z)$ قواعد ۲ و ۳، محمول $\text{natle}(M,N)$ را هنگامی دارای ارزش درست تعریف می‌کنند که M یک عدد طبیعی و کوچکتر یا مساوی مقدار N باشد) و قاعده ۴ محمول $\text{succ}(M,N)$ را هنگامی دارای ارزش درست تعریف می‌کند که N یک واحد بزرگتر از M باشد. equal یک محمول از پیش تعریف شده و درونی برای آزمون تساوی در زبان پرولوگ است. حداکثر مقدار پارامتر C می‌تواند ۲۰ باشد.

برنامه pythagoras نشاندهنده روش برنامه‌سازی در زبان پرولوگ است که در آن، قواعد فقط بیانگر روابط مستقل از زمان هستند و برخلاف زبانهای متعارف مفهوم توالی اجرای دستورالعملها نقشی ندارد. در این زبان برنامه‌ساز تنها به توصیف مسئله می‌پردازد و تعیین راه به عهده سیستم برنامه‌سازی پرولوگ است که معمولاً به شکل مفسر عمل می‌کند.

سیستم برنامه‌سازی پرولوگ شامل یک سازوکاری جویی^۳ درونی است که سعی می‌کند تمام متغیرها را به شیوه‌ای نظام‌مند به نحوی مقید به مقادیر سازد که «هدف» محاسبه (در این حالت، محمول pythagoras) ارزش درست یابد. برای رابطه $\text{pythagoras}(X,Y,Z)$ این عمل به صورت یک دنباله زمانی از مقید سازیها انجام می‌شود که در شکل ۱ نشان داده شده است.

هرگاه در حین اجرا، کلیه محمولات سمت راست دارای ارزش درست شوند، بدین معنی است که یک نحوه انقیاد متغیرها یافته شده است که هدف را برآورده می‌سازد. این نحوه انقیاد یک راه حل برای مسئله است.

مدل در نظر گرفته شده در این شیوه برنامه‌سازی همانند منطق محمولات است. برخی از روابط وجود دارند که همیشه درست هستند (مانند قاعده ۲ در مثال ما) که آنها را «اصل»^۴ یا «واقعیت»^۵ می‌نامند. این قواعد ذخیره شده و تشکیل پایگاه معرفت^۶ را می‌دهند. کلیه روابط دیگر توسط یک یا چند قاعده استنتاج^۷ (که درستی آنها را از روی درستی روابط ساده‌تر استنتاج می‌کنند) تعریف می‌شوند. تنها شیوه

- ۲ implication
- ۳ backtrack mechanism
- ۴ axiom
- ۵ fact
- ۶ knowledge base
- ۷ rules of inference

پارامتر خاص، این جهت برای هر فراخوانی در طول اجرای برنامه، ثابت است. در زبان پرولوگ مکانیزم کلیتری به کار گرفته می شود به این صورت که به جای «تخصیص» پارامترهای واقعی و صوری، سعی در «یگانه سازی» آنها به عمل می آید. چنانچه یکی از پارامترها متغیری مقید نشده و پارامتر دیگر، مقدار ثابت یا متغیر مقید شده ای باشد، یگانه سازی و تخصیص مشابهت دارند. اما اگر هر دو متغیرهایی مقید نشده باشند مانند آنچه در قواعد زیر ممکن است اتفاق افتد

... :- goal(Y) ...

goal(X) :- ...

دو متغیر X و Y به «یک» متغیر مقید نشده یگانه سازی می شوند به نحوی که از آن پس یکسان در نظر گرفته می شوند. مثلاً پس از آن، انقیاد X به مقداری مانند ۱۰، به سبب انقیاد Y به همان مقدار ۱۰ نیز می شود. این تأثیری است که در سایر زبانها وجود ندارد، هر چند وجود چنین خصیصه ای شرایط تهیه برنامه های دقیقتر و شگردهای برنامه سازی قویتری را در اختیار می گذارد اما درک آن گاهی دشوار است.

اساسیترین ساختار داده ها در زبان پرولوگ، «درخت» است که شباهت زیادی به ساختار داده های لیست در زبان لیسپ دارد. یک درخت ممکن است شامل ترکیبی از متغیرها و مقادیر ثابت باشد. به عنوان مثال سبزی را حاوی مقدار معینی از یک نوع میوه در نظر بگیرید. این سبزی را می توان توسط درخت $\text{fruitbasket}(\text{Fruitkind}, \text{Amount})$ نشان داد. حال دو درخت $\text{fruitbasket}(\text{apple}, Z)$ و $\text{fruitbasket}(X, 20)$ را در نظر می گیریم که در آن X و Z متغیرهای مقید نشده و 20 و apple مقادیر ثابت خواهند بود. دو پارامتر موجود ممکن است با مقید شدن X به apple و Z به 20 یگانه سازی شده درخت $\text{fruitbasket}(\text{apple}, 20)$ را نتیجه دهند. بدون وارد شدن به جزئیات بیشتری در مورد یگانه سازی، در مثال اخیر دیده می شود که انتقال یک پارامتر از نوع درخت ممکن است باعث چندین انقیاد و فراتر از آن، در جهت های مختلف گردد. بدین ترتیب، تقسیم بندی پارامترها به دو دسته کلی ورودی و خروجی در زبانهای منطقی چندان معنی ندارد.

به عنوان نتیجه بحث، یک متغیر منطقی در هر لحظه می تواند در یکی از سه وضعیت زیر باشد:

- مقید نشده،
- نسبتاً مقید (به یک متغیر یا به درختی با چند متغیر) و یا
- کاملاً مقید.

و این سه وضعیت ممکن است تحت تأثیر مکانیزم پی جویی به یکدیگر تغییر یابند.

یک متغیر منطقی را می توان به عنوان مجموعه تمام مقادیر ممکن در یک فضای معین و مکانیزم یگانه سازی را می توان به عنوان عملی جهت تبدیل این مجموعه به زیر مجموعه ای کوچکتر در نظر گرفت. درخت

منطقی چیز جز اثبات قضیه نیست. البته این نظر نیز بسیار اغراق آمیز است زیرا برای مثال نمی توان تعیین یک مقدار برای متغیر A به وجهی که محمول $\text{pythagoras}(A, 4, 5)$ ارزش درست یابد را اثبات یک قضیه دانست.

با این حال، این نظر صحیح است که در برنامه سازی منطقی، هر مسئله باید به صورت مجموعه ای از محمولات، واقعیتهای و قواعد استنتاج تعریف شود. از لحاظ نظری، نوشتن یک برنامه منطقی به نحوی که یک ماشین تیورینگ دلخواه را شبیه سازی کند، همواره امکان پذیر است. ولی از نظر تجربی این امر به زمینه کاربردی مربوطه و مهارت برنامه ساز بستگی دارد. به کارگیری برنامه سازی منطقی در بعضی از زمینه های کاربردی کاملاً با ماهیت مسائل همخوانی دارد (به ویژه، مسائلی نظیر جستجوهای ترکیبی) ولی در بسیاری از موارد، تعریف کلیه خواسته ها توسط قواعد منطق رتبه اول بسیار غیر طبیعی و دشوار است.

در برنامه سازی منطقی، هر مسئله باید به صورت مجموعه ای از محمولات، واقعیتهای و قواعد استنتاج تعریف شود.

متغیرهای منطقی و یگانه سازی

علاوه بر مکانیزم پی جویی، ویژگی دیگری نیز وجود دارد که معرف مدل تفکر در برنامه سازی منطقی است و آن، رفتار متغیرهای منطقی است. در نگاه اول، تفاوت چندانی بین رفتار یک متغیر در زبانهای عملیاتی و زبانهای منطقی به چشم نمی خورد زیرا یک متغیر منطقی نیز در نتیجه ارزیابی و محاسبه محمول به یک مقدار مقید می شود و تا انتهای برنامه نیز به همان مقدار مقید می ماند. به علاوه، در برنامه سازی منطقی نیز عبارتهای تخصیصی وجود ندارند و متغیرها به عنوان جایگاهی برای ذخیره سازی مقادیر محسوب نمی شوند. اما این تشابه به دلایل متعددی فقط یک تشابه ظاهری است. اولاً به خاطر عدم وجود مفهوم ترتیب زمانی در اجرای یک برنامه عملیاتی، متغیرهای عملیاتی از ابتدای برنامه تا انتها، «همواره» به مقداری مقیدند اما در زبانهای منطقی متغیرها تا زمان تفسیر قواعد به مقادیر مقید نیستند. تفاوت دیگر به مفهوم پی جویی مربوط است. متغیرها در یک زبان منطقی فقط تا زمانی به یک مقدار مقید می مانند که پی جویی رخ ندهد. پی جویی باعث انقیاد متغیر به مقداری جدید و یا بی قید ماندن آن می شود.

سومین اختلاف، ناشی از مکانیزم انتقال پارامتر در پرولوگ است. این کار توسط عمل یگانه سازی^۴ صورت می گیرد که بین کلیه زبانهای برنامه سازی منحصر به فرد است. در سایر زبانها، یک پارامتر کلاً از فراخواننده به فراخوانده شده یا برعکس تخصیص می یابد و برای یک

جمع بندی

در نتیجه تحلیل مدل‌های مختلف تفکر، طبقه‌بندی جدیدی از زبانهای برنامه‌سازی به غیر از آنچه در مقدمه ارائه شد، به دست می‌آید (شکل ۲). در یک سو، زبانهایی قرار می‌گیرند که مدل تفکرشان مبتنی بر محاسبه مقادیر نامشخص از روی مقادیر مشخص است. این مدل تفکر را می‌توان «دیدگاه ریاضی»^۱ یا «دیدگاه مقدارگرا»^۲ و یا «دیدگاه نتیجه‌گرا»^۳ از برنامه‌سازی نامید. در اینجا، کامپیوتر نقش یک ماشین حساب را ایفا می‌کند که مقادیر ورودی را دریافت و مقادیر خروجی را محاسبه می‌کند. محاسبات اساساً مستقل و خودکار هستند. تمام ورودیها در آغاز محاسبات باید وجود داشته باشند و کامپیوتر بدون هیچ وقفه‌ای از جهان خارج، عملیات را تا انتها انجام می‌دهد و تنها در این هنگام است که حاصل محاسبات نمایش داده می‌شود. این دیدگاه به طور صریح بیانگر مدل تفکر دستوری و الگوریتمی است ولی به طور ضمنی بیانگر دیدگاه تفکر عملیاتی و منطقی نیز هست زیرا برنامه‌ساز زبانهای منطقی و عملیاتی باید تفسیری از برنامه مورد نظرش را در ذهن داشته باشد و به همین خاطر مجبور می‌شود که در بستر یک مدل دستوری فکر کند. این دیدگاه به سه مدل ریاضی زیر مربوط است:

ماشین تورینگ — برنامه‌سازی الگوریتمی
توابع بازگشتی — برنامه‌سازی عملیاتی
منطق محمولات — برنامه‌سازی منطقی

به نظر می‌رسد که خصیصه بارز تمام این خانواده از زبانها، رفتار مستقل آنها باشد. هیچ ارتباطی با جهان خارج در این مدل نمی‌گنجد. خواه این ارتباط، یک ورودی، یک خروجی یا یک وقفه در حین اجرای محاسبات باشد. از آنجاکه در عالم واقع، محاسبات ماهیت مستقل ندارند، مدل‌های تفکر ارائه شده باید به منظور یافتن ارزش عملی، تغییراتی را بپذیرند. آنها باید امکان انجام عملیات ورودی، خروجی و همچنین احتمالاً وقفه در حین محاسبات را در اختیار بگذارند.

شیوه سنتی برنامه‌سازی مبتنی بر
برنامه‌سازی مقدارگراست ولی طی ده
سال آتی، دیدگاه موضوع‌گرا اهمیت
فزاینده‌ای خواهد یافت.

در طرف دیگر، دیدگاهی وجود دارد که کامپیوتر را به عنوان شبیه‌ساز مؤلفه‌های یک سیستم بزرگتر در نظر می‌گیرد. کل سیستم می‌تواند شامل کامپیوتر و استفاده‌کننده یا شبکه‌ای متشکل از تعدادی کامپیوتر، دستگاههای فنی دیگر و استفاده‌کنندگان باشد. در اینجا، کامپیوتر تنها

mathematical view ۱
value-oriented ۲
result oriented ۳

fruitbasket(X,20) نشانگر تمام سبدهای حاوی ۲۰ عدد میوه و درخت fruitbasket(apple,Z) بیانگر کلیه سبدهای میوه حاوی سیب است و یگانه‌سازی این دو، باعث به وجود آمدن اشتراک آنها یعنی سبدهای با ۲۰ سیب می‌شود.

تفکر دستوری در برنامه‌سازی منطقی

طرفداران برنامه‌سازی منطقی ادعا می‌کنند که این روش، یک روش برنامه‌سازی کاملاً توصیفی است که در آن، برنامه فقط به شکل مجموعه‌ای از قواعد منطقی تعریف می‌شود. این امر نه تنها سطح تجرید در برنامه‌سازی را بالا می‌برد بلکه برنامه‌ساز را از فکر کردن راجع به راه حل نیز به کلی آزاد می‌کند. او فقط مسئله را در منطق مرتبه اول بیان می‌کند و سیستم برنامه‌سازی پرولوگ، خود راه حل را می‌یابد.

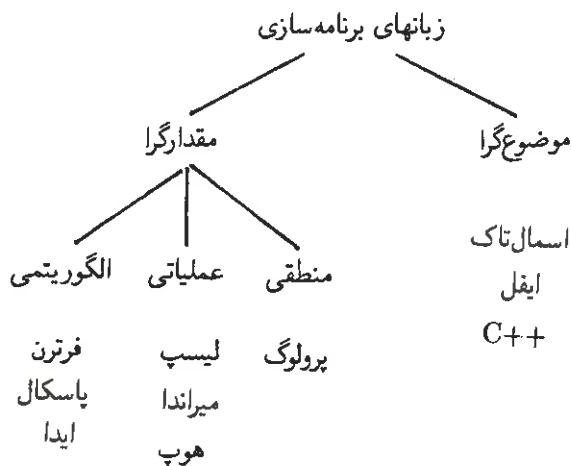
البته این ادعا در عالم واقع نمی‌تواند به طور کامل برآورده شود. برای نمونه بد نیست قواعد لازم برای برنامه pythagoras را مجدداً در نظر بگیریم. اگر آنها در واقع کاملاً رابطه‌ای بودند، کافی بود که قانون اول به شکل زیر نوشته شود:

pythagoras(X,Y,Z) :- nat(Z), nat(Y),
nat(X), equal(X*X+Y*Y,Z*Z).

که در آن nat(N) بیانگر این رابطه است که: «N یک عدد طبیعی است». به علاوه، سمت راست عبارت باید به هر ترتیب دیگری نیز قابل بازنویسی باشد. برای مثال:

pythagoras(X,Y,Z) :- nat(Y),
equal(X*X+Y*Y,Z*Z), nat(X), nat(Z).

ولی هیچیک امکانپذیر نیست زیر الگوریتم پی‌جویی در زبان پرولوگ، تنها مجموعه‌ای متناهی از عبارات را در نظر گرفته و از چپ به راست پیش می‌رود و قواعد را از بالا به پایین به ترتیبی که برنامه‌ساز نوشته است مورد ارزیابی قرار می‌دهد. بدین ترتیب برنامه‌ساز موظف است که محدودیت تعداد و ترتیب قواعد و عناصر سمت راست عبارت را در نظر داشته باشد به نحوی که مکانیزم پی‌جویی به راه حلی دست یابد و دچار انحراف نشود. علت به کارگیری محمول nat، حد بالایی مصنوعی برای Z و همچنین ترتیب به کار گرفته شده در سمت راست عبارت، همین محدودیت است. به بیان دیگر، برنامه‌ساز برای تولید یک برنامه به زبان پرولوگ باید رفتار مفسر زبان را در ذهن خود با دقت قابل توجهی شبیه‌سازی کند. به علاوه، او باید از ویژگیهای خاص و غیر منطقی زبان، به ویژه آنچه که cut نامیده می‌شود، برای کنترل پی‌جویی استفاده کند. در نتیجه، برنامه‌های پرولوگ هر چند بسیار کوتاه به نظر می‌رسند ولی درک آنها برای انسان مشکل است زیرا شبیه‌سازی ماشین پرولوگ در ذهن ممکن است بسیار پیچیده باشد.



شکل ۲- دسته بندی دیگری از زبانهای برنامه‌سازی

$g(E, Z, \text{Next}Z)$,
perpetual-process(NextZ)

به عقیده نویسندگان، چنین کاربردی بسیار مصنوعی است و مدل تفکر زبانهای مقدارگرا را مخدوش می‌کند. در عمل نیز به دلیل دشواری زیاد تقریباً غیرممکن است. برای مثال، حالتی را در نظر بگیرید که برنامه‌ای به زبان عملیاتی بخواهد محاوره بین انسان و ماشین را انجام دهد.

با این وجود پیشنهادات و تلاشهای بسیاری جهت تلفیق مزایای مدل‌های مختلف تفکر در «زبانهای با طیف وسیع»^{۱۳} به عمل آمده‌است. حاصل این تلاشها زبانهای جدیدی است که از دو یا حتی سه مدل تفکر متفاوت پشتیبانی می‌کنند (برخی از آنها امکان عملیات موازی را نیز در اختیار می‌گذارند). اما این گونه ترکیبات مدل‌های تفکر در چهارچوب این مقاله نمی‌گنجد.

تمایز بین دیدگاه مقدارگرا و دیدگاه موضوع‌گرا و در نظر گرفتن اینکه دیدگاه مقدارگرا حالت خاصی از دیدگاه عمومیت موضوع‌گراست می‌تواند به جمع بندی دیگری بیانجامد. اگر قبول کنیم که دیدگاه مقدارگرا دیدگاهی ریاضی است و دیدگاه موضوع‌گرا به علوم کامپیوتر تعلق دارد می‌توان نتیجه گرفت که ریاضیات محاسباتی بخش خاصی از علوم کامپیوتر است (و نه برعکس آن، آنگونه که بسیاری از ریاضیدانها اعتقاد دارند) و از آن مهمتر اینکه مفهوم الگوریتم به خاطر مقدارگرا بودنش نباید به عنوان مفهوم اصلی علوم کامپیوتر تلقی شود. یکی از خصوصیات اساسی الگوریتم پایان‌پذیری آن است (در غیر این صورت، آن را الگوریتم نمی‌خوانند بلکه به آن رویه می‌گویند). به علاوه الگوریتم فاقد هر گونه نشانه‌ای از ارتباط با محیط است و عملیات ورودی و خروجی در حین اجرا را در نظر نمی‌گیرد. به این جهت، این تعریف به عنوان پایه علوم کامپیوتر مناسب نیست.

یک عامل در بین سایر عوامل است و هدف اصلیش تغییر شکل مقادیر نیست بلکه وظیفه‌اش شبیه‌سازی رفتار یک یا چند موضوع یا شی است. می‌توانیم این را «دیدگاه شبیه‌سازی» یا «دیدگاه موضوع‌گرا» بنامیم. موضوعها بوجود می‌آیند و از بین می‌روند. آنها دارای دوره حیاتی هستند که معمولاً در آن رفتارشان را تغییر می‌دهند. وجه تمایز موضوعها نسبت به مقادیر این است که موضوعها دارای خصیصه‌ای به نام وضعیت هستند و این وضعیت در طول زمان متحول می‌شود. بنابراین «وضعیت»، خصیصه اصلی دیدگاه شبیه‌سازی است.

دیدگاه مقدارگرا حالت خاصی از دیدگاه عمومیت موضوع‌گراست.

شیوه سنتی برنامه‌سازی مبتنی بر برنامه‌سازی مقدارگراست ولی طی ده سال آتی، دیدگاه موضوع‌گرا اهمیت فزاینده‌ای خواهد یافت. باید روشن شده باشد که دیدگاه موضوع‌گرا از عمومیت بیشتری برخوردار است زیرا یک سیستم مستقل، حالت خاصی از یک شبکه سیستمهای ارتباطی است. هر چیز را که بتوان بر مبنای دیدگاه مقدارگرا برنامه‌سازی کرد، در دیدگاه موضوع‌گرا نیز قابل برنامه‌سازی است. عکس مسئله نیز از نظر تئوری ولی نه در عمل، امکان‌پذیر است. در مدل‌های تفکر عملیاتی و منطقی نیز در واقع می‌توان، به نوعی، مفهوم وضعیت را گنجانند. این زبانها برای آن که مورد استفاده قرار گیرند باید به هر حال دستورات ورودی و خروجی داشته باشند و سپس به کمک مفهوم بازگشت‌پذیری، یک پردازش بی‌پایان در آنها قابل شبیه‌سازی است. فرض کنید چنین پردازشی با «معادلات ماشین خودکار»^{۱۴} زیر تعریف شود:

$$\begin{aligned} a_i &:= f(e_i, z_i) && \text{output equation} \\ z_{i+1} &:= g(e_i, z_i) && \text{state equation} \end{aligned}$$

در مقاطع زمانی $i = 1, 2, \dots$ و دنباله زمانی $e_1, e_2, \dots$ برای ورودی و $a_1, a_2, \dots$ برای خروجی و $z_1, z_2, \dots$ برای وضعیتها. رفتار چنین ماشین خودکاری می‌تواند در یک زبان عملیاتی به شکل زیر:

```
perpetual-process(z) =
let e = input(); output(f(e,z));
perpetual-process(g(e,z))
```

و در زبان پرولوگ به صورت زیر نشان داده شود:

```
perpetual-process(Z) :-
input(E), f(E,Z,A), output(A),
```

- [3] Bird R, Walder P, "Introduction to Functional Programming", Prentice Hall, 1988.
- [4] Cleaveland JC, "An Introduction to Data Types", Addison-Wesley, 1986.
- [5] Field AJ, Harrison PG, "Functional Programming", Addison-Wesley, 1988.
- [6] Ghezzi C, Jazayeri M, "Programming Language Concepts", Wiley, 1987.
- [7] Pratt TW, "Programming Languages-Design and Implementation", 2. ed. Prentice Hall, 1984.
- الگوریتم باید با مفهوم دیگری مبتنی بر موضوعهای مرتبط در پردازشهای دائمی (که الگوریتم نیز حالت خاصی از آن است) جایگزین شود. البته یافتن چنین مفهومی دشوار به نظر می‌رسد. در منبع [1] آگامی در این جهت برداشته شده است.
- منابع:
- [1] Agha GA, Actors, "A Model of Concurrent Computation in Distributed Systems", The MIT Press, 1988.
- [2] Aho AV, Ullman JD, "The Theory of Parsing, Translation, and Compiling", Prentice Hall, 1972.

فراخوان برای مقاله سمینار مهندسی نرم افزار

انجمن انفورماتیک ایران در تابستان سال ۱۳۷۱، سمیناری تحت عنوان «مهندسی نرم افزار» را در تهران برگزار می‌کند. بحث‌های این سمینار عبارتند از:

چرخه عمر و متدولوژی نرم افزار
مدلهای چرخه عمر
متدولوژی و مستندسازی

جایگاه مهندسی نرم افزار
بحران نرم افزار
تاریخچه
برنامه سازی و مهندسی نرم افزار
ملاحظات اقتصادی در نرم افزار

مباحث دیگر
اعتبارسنجی و واریسی (V&V)
مدیریت پیکربندی
مدیریت پروژه‌های نرم افزار
مدلهای برآورد هزینه
سنجش نرم افزار (software metrics)
اطمینان پذیری
قابلیت حمل

تشریح مدل آبشاری
برنامه ریزی پروژه
تعیین مشخصات و نیازها
طراحی
پیاده سازی
آزمایش
نگهداری و تکامل

بدین وسیله از کلیه کسانی که مایلند مقاله‌ای در این سمینار ارائه کنند دعوت می‌شود تا خلاصه‌ای در حدود ۵۰۰ کلمه از مقاله خود را حداکثر تا پایان تیر ۱۳۷۱ به نشانی زیر ارسال کنند:

کمیته برگزاری سمینار مهندسی نرم افزار
انجمن انفورماتیک ایران
صندوق پستی ۱۴۱۵۵-۱۱۹۶
تهران

گزارش

سمینار اسپارک

راه این قسمت پارامترهای بین دو زیربرنامه انتقال می‌یابند. مطالعات آماری نشان می‌دهد که درصد سرریز شدن پنجره‌ها و انتقال آنان به حافظه اصلی به‌میزانی است که مزیت این ساختار را از نظر سرعت مخدوش نمی‌کند.

هر چند در ابتدای مطرح شدن مفهوم پردازنده کم‌دستور، سادگی دستورات و در نتیجه ساده شدن قسمت کنترل پردازنده، از مزایای این معماری محسوب می‌شد ولی نیاز به پردازنده‌های سریعتر و هم‌چنین افزایش چگالی ترانزیستورها روی تراشه‌ها موجب شد که قسمتهایی مانند «واحد محاسبات ممیز شناور»^۵ و «واحد مدیریت حافظه»^۶ نیز روی پردازنده‌های کم‌دستورالعمل قرار گیرند. به‌طور کلی با امکان‌پذیر شدن ساخت تراشه‌هایی با چگالی بالا که می‌توانند تا ۱۰ میلیون ترانزیستور را در خود جای دهند (پردازنده‌های ۸۰۴۸۶ و ۶۸۰۴۰ کمی بیش از یک میلیون ترانزیستور دارند)، مسئله محدودیت فضای تراشه که در زمان طراحی اولیه پردازنده‌های کم‌دستور مطرح بود اینک کم‌رنگ شده و حتی این سؤال مطرح است که از این تعداد زیاد ترانزیستور روی تراشه چگونه می‌توان به‌نحو احسن استفاده کرد.

موفقیت معماری کم‌دستور و رقابت بین پردازنده‌های موجود برای رسیدن به سرعت بالاتر موجب شده‌است که طراحان پردازنده‌های متداول^۷ نیز به اصول معماری کم‌دستور توجه داشته باشند و به‌عنوان مثال در طراحی پردازنده‌های اینتل ۸۰۴۸۶ مفاهیمی از پردازنده‌های کم‌دستور برای افزایش سرعت و بهبود کارایی پیاده شده‌است (از جمله تجدید نظر در طراحی واحد وقفه^۸ و گنجاندن ۸ کیلو بایت حافظه پنهان^۹ روی تراشه). به این ترتیب کارایی این پردازنده‌ها بالا رفته و به پردازنده‌های کم‌دستور موجود رسیده ولی به‌رحال پردازنده‌های اینتل همواره محدود به سازگاری با ۸۰۸۶ هستند که این خود نکته‌ای منفی و بازدارنده برای طراحی‌های جدید است.

در طراحی‌های جدید پردازنده‌های کم‌دستور به پردازش موازی نیز توجه شده‌است. البته این امر قسمت کنترل پردازنده را بسیار پیچیده‌تر می‌کند که هر چند مغایر با سادگی، از اصول اولیه این معماری است ولی نیاز به سرعت بیشتر که امروزه مطرح است جریان طراحی پردازنده‌ها را پیچیده‌تر می‌کند. اما با افزایش و بهبود روزافزون ابزارها و نرم‌افزارهای کمک طراحی روز به روز کار طراحی، شبیه‌سازی و آزمایش مدارها آسانتر می‌شود.

نکته آخر اینکه بر اساس مفاهیم تعریف شده برای معماری کم‌دستور، این پردازنده‌ها هنوز از نظر قابلیت‌های گرافیکی ضعیف هستند ولی در طراحی‌های جدید، گنجاندن امکانات گرافیکی نیز مطرح است.

گزارش از کامران قانع و جواد رضوی‌لر

در روز چهارشنبه ۱۱ دی ماه در سالن آمفی‌تئاتر دانشگاه صنعتی امیرکبیر (پلی تکنیک) سمیناری توسط دو تن از طراحان ایستگاه کار اسپارک، آقایان دکتر نامجو و مهندس مظفری کرمانی که برای مدت کوتاهی به ایران سفر کرده بودند، برگزار شد.

ایستگاه کار اسپارک محصول شرکت سان^۱ است^۲ و بر اساس یک پردازنده کم‌دستورالعمل^۳ ساخته شده‌است. آقای دکتر نامجو از ابتدا در تمام مراحل طراحی و تولید این پردازنده فعالیت داشته‌اند. آقای مهندس مظفری کرمانی نیز مدیر گروه طراحی و آزمایش مدارهای ایستگاه کار اسپارک هستند.

سمینار با سخنان آقای دکتر صفابخش آغاز شد و سپس دکتر نامجو به تفصیل در مورد خصوصیات پردازنده کم‌دستور اسپارک و ایده‌های به‌کار رفته در طراحی آن سخن گفتند. ایشان تاکید داشتند که هدف اصلی شرکت سان این است که پردازنده اسپارک به‌عنوان استاندارد پردازنده‌های کم‌دستور در بازار شناخته شود. سپس آقای مهندس مظفری کرمانی در مورد مراحل تولید و روشهای آزمایش مدارها و هم‌چنین نرم‌افزارهایی که برای طراحی مدارها به‌کار گرفته می‌شود، توضیحاتی ارائه دادند. روشهای مدیریت پروژه‌ها نیز از موضوعاتی بود که آقای کرمانی مختصراً در مورد آن صحبت کردند.

سپس جلسه پرسش و پاسخ برگزار شد که در طی آن سخنرانان سمینار به سؤالات شرکت‌کنندگان در مورد پردازنده اسپارک و امکانات آن در مقایسه با سایر پردازنده‌های کم‌دستور از جمله ایستگاه کار RS/6000 ساخت شرکت آی‌بی‌ام، فعالیتهای آینده شرکت سان در رقابت با سایر تولیدکنندگان پردازنده‌های کم‌دستور و هم‌چنین امکانات سخت‌افزاری و نرم‌افزاری این ایستگاه کار پاسخ دادند. بعضی از مطالب مطرح شده در سمینار از این قرارند:

یکی از نقاط قوت اسپارک که آن را از بقیه پردازنده‌های کم‌دستور موجود متمایز می‌سازد، استفاده از ساختمان حلقوی برای «پنجره ثباتها»^۴ است. این ساختار در ابتدا در پردازنده‌های کم‌دستور دانشگاه برکلی معرفی و پیاده‌سازی شد و در آن به‌ارای فراخوانی هر زیربرنامه، قسمتی از فضای ثباتهای پردازنده به‌صورت یک پنجره به زیر برنامه اختصاص داده می‌شود. قسمتی از هر پنجره با پنجره مجاور مشترک است و از

Sun^۱

شرکت سان در ابتدا «اسپارک» را مخفف Sun Processor Architecture قرار داده بود ولی بعد از عرضه آن به بازار و سعی در همه‌منظوره ساختنش، عبارت Scalable Processor ARChitecture را برای این منظور انتخاب کرد.

Reduced Instruction Set Computer-RISC^۲
register window^۳

Floating Point Unit-FPU^۵
Memory Management Unit-MMU^۶
Complex Instruction Set Computer-CISC^۷
Interrupt Unit-IU^۸
cache memory^۹

گزارش

شبکه‌ها علیه استقلال *

به دنبال شبکه‌ها فرصت مناسب دیگری برای قانون‌گذاران جدید فراهم شد — آلودگی با ویروس. به زودی زنگهای خطر به صدا درآمد: آلودگی ناشی از نرم‌افزار غیراستاندارد می‌تواند شرکت را به زانو درآورد. البته احتمال وقوع چنین اتفاقی کم بود. اکثر موارد مشهور حمله ویروسها یا کرمها مربوط به سیستمهایی می‌شد که به‌طور متمرکز اداره می‌شدند.

ضدانقلاب در شرف تکمیل شدن بود. شرکت آی‌بی‌ام سیستم عامل OS/2 را با این وعده که ریزکامپیوترها را به کامپیوترهای بزرگ ارتباط دهد، وارد بازار کرد. یونیکس که روی شبکه کامپیوترهای شخصی کار می‌کرد، کم‌کم شبیه مینی کامپیوترهای دیروزی شده بود. رویای مدیر مرکز کامپیوتر از دور نمایان شد: ریزکامپیوترهای بدون دیسک که از همه لحاظ مانند ترمینالهای کامپیوترهای بزرگ بودند.

مسئله ریزکامپیوترهای امروزی بیش از هر زمانی نیرومند هستند ولی انقلاب ریزکامپیوترها برای رسیدن به آزادی بود نه قدرت. ریزکامپیوترهای اولیه در مقایسه با مینی کامپیوترها و کامپیوترهای بزرگ عصر خود کوچک بودند، ولی به استفاده‌کنندگان آزادی شخصی در انتخاب عرضه می‌کردند که چیز تازه‌ای بود. می‌توانستیم به فروشندگان مختلف سرزنیم و سیستمهای کامپیوتری دلخواه‌مان را انتخاب کنیم. نرم‌افزار را خودمان انتخاب کرده یا گسترش دهیم و همه را به طریقی که مناسب با خلاقیت طبیعی خودمان باشد، به کار ببریم. خودمان تصمیم می‌گرفتیم که چگونه داده‌ها را وارد کنیم، سازمان دهیم، پردازش کنیم و نتایج را تهیه کنیم و احیاناً آن را از بین ببریم.

در پاسخ به اشتیاق ما، سخت‌افزار مطمئنتر و همسازتر شد و نصب آن آسان گردید. کاربرد نرم‌افزار آسانتر، و خود نرم‌افزار مستندتر و ارزانتر شد. اکنون نرم‌افزار بیشتری موجود است ولی ما فقط می‌توانیم از بین برنامه‌های موجود در شرکت یا چند نرم‌افزار لوکس موجود در فروشگاههای کامپیوتری، برنامه مورد نیاز خود را انتخاب کنیم.

نگران‌کننده‌تر از همه این است که شبکه‌ها مدیریت متمرکز کامپیوترها را از شرکتهای بزرگ به کوچکترین و نوآورترین شرکتها انتقال داده‌اند. شرکتهایی وجود دارند که در آنها چهار یا پنج ریزکامپیوتر به هم متصل شده‌اند و «مدیر سیستم» با راهنمایی مشاورین شبکه یا کتابهای راهنما مشغول نوشتن روشها و تعیین استانداردهاست و دستوراتی در مورد دسترسی به شبکه‌ها صادر می‌کند.

شکی نیست که مدیریت شبکه لازم است اما بهتر است مطمئن باشیم که کنترل دستگاهها به معنای محدود کردن خلاقیت استفاده‌کنندگان آنها نیست.

ترجمه هما فاضل‌نیا



آیا انقلاب کامپیوترهای شخصی به پایان رسیده‌است؟ به یاد بیاورید چند سال پیش وقتی که وارد دفتر سرپرست بخش‌مان شدیم و انبوهی از کاغذهای بانوارهای سبز رنگ را روی میز گذاشتیم و درخواست یک کامپیوتر شخصی کردیم. ما از نرم‌افزارهای پر از خطا و دمدمی مزاجی که در مینی کامپیوترها و کامپیوترهای بزرگ شرکت وجود داشت، خسته شده بودیم. همچنین از طرز برخورد غیردوستانه کارکنان مرکز کامپیوتر و از وعده‌های سرخرمن آنها در مورد اینکه تواناییهای مورد انتظار در مشخصات سیستم عملاً تحقق خواهد یافت، ناراضی بودیم.

آن وقت مقالاتی از مجله «بایت» را در باره زد ۸۰، اپل ۲ و آی‌بی‌ام بی‌سی نشان دادیم و گفتیم که این دستگاهها خودکفا هستند و به قدری ارزان که می‌توان سیستمی را بدون دخالت بخشهای کامپیوتری شرکت و مستقل از سیاست‌گذاریها و بودجه‌بندیهای آنها، راه‌اندازی کرد. با نرم‌افزارهایی چون مفسر بیسیک مایکروسافت، Word Star، VisiCalc و dBase می‌توانستیم برنامه‌های کاربردی داخل شرکت را با صرفه‌جویی در وقت و انرژی ایجاد کنیم. انقلاب کامپیوترهای شخصی آغاز شده بود.

چون فعالیت کامپیوترهای شخصی بخشهای مختلف با کامپیوتر بزرگ شرکت رقابت می‌کرد و حتی از آن جلو زده بود، مدیران مراکز کامپیوتر شرکتها متوجه شدند که حوزه اقتدار و منافعتشان در معرض خطر است. چون نمی‌توانستند با انقلاب کامپیوترهای شخصی رقابت کنند و مایل هم نبودند که به آن ملحق شوند، درصدد برآمدند که خود کنترل کامپیوترهای شخصی را به دست گیرند.

در ابتدا موفقیت کمی داشتند اما پس از مدتی دنیای کامپیوترهای شخصی حربه مناسب را در اختیارشان قرار داد و آن اتصال به شبکه‌های کامپیوتری بود. در سالهای میانی دهه ۱۹۸۰ بحث روز، مرتبط ساختن همه ریزکامپیوترها در تمام بخشها و فراهم آوردن امکان تسهیم داده‌ها و دستگاههای جانبی بود. پس واحد هماهنگ‌کننده‌ای مورد نیاز بود تا مسائل ارتباطی، تنظیم مسیر کابل و روشهای شبکه‌ای را حل کند. چه کسی برای این کار بهتر از کارمندان باتجربه مرکز کامپیوتر شرکت بود؟

بعد از استاندارد کردن ارتباطهای کامپیوتری نوبت به استاندارد کردن خود کامپیوترها و سپس نرم‌افزارها، چاپگرها، کارتهای گرافیک و غیره رسید. به زودی بخشنامه‌هایی در مورد چگونگی سازماندهی و نامگذاری دیسکها و حتی رنگ برچسب دیسکها صادر شد. ضدانقلاب شروع شده بود.

گزارش

دومین مرحله المپیاد ملی
انفورماتیک

دومین مرحله المپیاد ملی انفورماتیک روزهای ۱۸ و ۱۹ بهمن ماه امسال در اهواز برگزار شد. در این مرحله شش نفر دانش آموزان نامبرده زیر از میان ۱۰۷ نفر شرکت کننده مرحله دوم، برگزیده شدند.

۱- سعید ایرانفر، دانش آموز سال چهارم، دبیرستان ایثار، منطقه ۱۴ تهران.

۲- فرشاد رستم آبادی، دانش آموز سال چهارم، دبیرستان شهید منتظری، منطقه ۳ تهران.

۳- کیومرث کاوه مریان، دانش آموز سال چهارم، دبیرستان علامه حلی، منطقه ۱۱ تهران.

۴- مرتضی محمدنوری، دانش آموز سال چهارم، دبیرستان نیکان، منطقه ۱ تهران.

۵- محمد مهدیان، دانش آموز سال سوم، دبیرستان مفید، منطقه ۲ تهران.

۶- سعید میرزایی، دانش آموز سال چهارم، دبیرستان ایثار، منطقه ۱۴ تهران.

شش نفر دانش آموز فوق الذکر در دوره آموزشی ویژه‌ای که در دانشگاه صنعتی شریف برگزار می‌شود، شرکت می‌کنند و از بین آنان چهار نفر که در مرحله نهایی گزینش خواهند شد برای شرکت در المپیاد جهانی اعزام می‌گردند. لازم به ذکر است که این شش نفر از شرکت در آزمون ورودی دانشگاهها معاف بوده و می‌توانند در هر یک از رشته‌های گروه ریاضی-فنی ادامه تحصیل دهند.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

شکل الف

۱۸ اسفند، نوبت صبح

وقت: دو ساعت و نیم
در مسائل ۱ و ۲ الگوریتمهای خواسته شده را می‌توانید با هر یک از روشهای نمودارگردشی یا زبان برنامه‌نویسی (بیسک یا پاسکال) بیان کنید.

۱- چندجمله‌ایهای $P(x) = a_0 + a_1x + \dots + a_nx^n$ ($a_n \neq 0$) و $Q(x) = b_0 + b_1x + \dots + b_mx^m$ ($b_m \neq 0$) را در نظر بگیرید. الگوریتمی بنویسید که a_0 تا a_n و b_0 تا b_m را بگیرد و ضرایب چندجمله‌ایهای $P(x) + Q(x)$ و $P(x) \times Q(x)$ را به دست آورد و به ترتیب در آرایه‌های S و P ذخیره کند. همچنین درجات چندجمله‌ایهای حاصل را به ترتیب در متغیرهای DS و DP ذخیره نماید. (۱۰ نمره)

۲- در یک دوره مسابقه، n تیم با شماره‌های ۱ تا n شرکت دارند و به صورت دوره‌ای هر تیم با تمامی تیمها مسابقه می‌دهد. هر مسابقه یک برنده و یک بازنده دارد. نتایج مسابقات در یک ماتریس $n \times n$ بدین ترتیب ثبت شده‌است که در درایه (i, j) شماره تیم برنده (i یا j) قرار دارد. عناصر روی قطر اصلی ماتریس نتایج صفر در نظر گرفته می‌شود.

یک دنباله $a_1, a_2, \dots, a_n$ از شماره تیمها را «دنباله برنده» می‌گوئیم اگر به ازای $i = 1, \dots, n-1$ ، تیم a_i از تیم a_{i+1} برده باشد (دقت کنید که $a_i \in \{1, 2, \dots, n\}$ و اگر $j \neq i$ آنگاه $a_i \neq a_j$).

الگوریتمی بنویسید که تعداد تیمها ($n \leq 20$) و ماتریس نتایج را بگیرد و یک دنباله برنده پیدا کرده، در یک آرایه n تایی قرار دهد. (۱۵ نمره)

در مسئله ۳ برنامه‌های خواسته شده را لزوماً با زبان برنامه‌نویسی (بیسک یا پاسکال) بیان کنید.

۳- یک ماتریس 12×12 متشکل از اعداد صفر و یک را در نظر بگیرید. هر سطر این ماتریس را می‌توان به عنوان یک عدد ۱۲ رقمی در مبنای ۲ در نظر گرفت که با تبدیل آن به مبنای ۱۰ یک عدد

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	1	1	0	1	1	0
0	0	0	0	0	0	1	1	0	1	1	0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

شکل ب

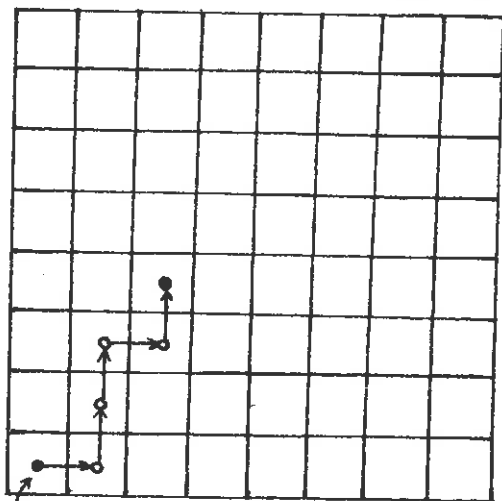
سپس ماشین از کاربر می‌خواهد که یک عدد صحیح نامنفی کوچکتر از 2^n در نظر بگیرد و بعد با n سؤال عدد مورد نظر کاربر را پیدا می‌کند. در هر سؤال 2^{n-1} عدد روی صفحه نمایش ظاهر می‌شود و کاربر باید در صورت وجود عدد مورد نظرش در بین آنها، پاسخ Y و در غیر این صورت پاسخ N را وارد کند. (۱۵ نمره)

مثال: اگر $n = 3$ و سه دسته عدد ظاهر شده روی صفحه نمایش و جوابهای کاربر به صورت زیر باشد، آنگاه عدد مورد نظر کاربر مساوی ۵ است.

۱	۳	۵	۷	Y
۲	۳	۶	۷	N
۴	۵	۶	۷	Y

۳- یک صفحه شطرنج (8×8) را در نظر بگیرید. یک مهره در خانه سمت چپ و پائین این صفحه قرار دارد. این خانه را «خانه شروع» می‌نامیم و با مختصات $(1, 1)$ نمایش می‌دهیم. مهره فوق را، در هر بار حرکت، فقط می‌توان یا یک خانه به سمت راست و یا یک خانه به سمت بالا حرکت داد. برنامه‌ای بنویسید که کلیه مسیرهای ممکن برای رسیدن این مهره از خانه شروع به خانه‌ای با مختصات (i, j) را تولید کند. هر مسیر با دنباله‌ای از R و U مشخص می‌شود که در آن هر حرکت به سمت راست با R و هر حرکت به سمت بالا با U نشان داده می‌شود. (۲۰ نمره)

مثال: در شکل زیر یک مسیر از خانه شروع به خانه $(3, 4)$ رسم شده است که به صورت دنباله RUURU نشان داده می‌شود.



خانه شروع

گزارش از یحیی تابش

گزارش اولین مرحله المپیاد ملی انفورماتیک نیز که در شماره قبل به چاپ رسید، توسط آقای تابش تهیه شده بود ولی متأسفانه نام ایشان از قلم افتاده بود؛ بدین وسیله ضمن پوزش خواهی، از آقای تابش به خاطر همکاری مداوم با گزارش کامپیوتر تشکر می‌کنیم. سردبیر

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0
0	0	0	0	0	0

شکل ج

صحیح نامنفی به دست می‌آید. با تبدیل این عدد به مبنای ۲ و در صورت لزوم افزایش ارقام از سمت چپ تا ۱۲ رقم دوباره می‌توان به سطر ماتریس دست یافت. در ماتریس فوق ارقام ۱ شکلی را به وجود می‌آورند (مانند شکل الف). ماتریس فوق را «۹۰ درجه دوران یافته» می‌نامیم در صورتی که شکل داخل آن نسبت به مرکز ماتریس، ۹۰ درجه در جهت مثبت مثلثاتی دوران داده شود (مانند شکل ب). ماتریس فوق را «فشرده شده» می‌نامیم در صورتی که هر ماتریس 2×2 داخل آن را با شرایط زیر به یک درایه تبدیل کنیم. در صورتی که تعداد درایه‌های مساوی ۱ یک ماتریس 2×2 داخلی بیشتر از یک باشد، آن ماتریس را به یک درایه ۱ و در غیر این صورت به یک درایه صفر تبدیل می‌کنیم (مانند شکل ج)، (ماتریسهای 2×2 را از گوشه سمت چپ و بالا در نظر می‌گیریم).

الف. برنامه‌ای بنویسید که ۱۲ عدد صحیح نامنفی کوچکتر از ۴۰۹۶ را از ورودی گرفته و با تبدیل آنها به مبنای ۲ به ترتیب سطریهای یک ماتریس 12×12 متشکل از صفر و یک را تولید کند.

ب. برنامه‌ای بنویسید که ماتریس حاصل از مرحله الف را ۹۰ درجه دوران دهد.

ج. برنامه‌ای بنویسید که ماتریس حاصل از مرحله الف را فشرده کند. (۲۵ نمره)

۱۹ اسفند، نوبت بعد از ظهر وقت: دو ساعت و نیم در مسائل ۱، ۲ و ۳ الگوریتمهای خواسته شده را می‌توانید با هر یک از روشهای نمودارگردشی یا زبان برنامه نویسی (پیسیک یا پاسکال) بیان کنید.

۱- یک الگوریتم غیر بازگشتی، فقط با استفاده از عمل جمع، بنویسید که تعداد ترکیبهای مختلف m شیئ متمایز از میان n شیئ متمایز را با استفاده از فرمول زیر محاسبه کند. (۱۵ نمره)

$$\binom{n}{m} = \begin{cases} 1 & n=m \text{ or } m=0 \\ \binom{n-1}{m} + \binom{n-1}{m-1} & n > m \end{cases}$$

۲- برنامه‌ای برای یک بازی بین کامپیوتر و کاربر (استفاده کننده از کامپیوتر) با شرایط زیر بنویسید:
ابتدا کاربر یک عدد طبیعی یک رقمی، n ، به ماشین می‌دهد.

مصاحبه

علم و اخلاق*

تکنولوژی «هوشمند» و مسئولیت اخلاقی دانشمند در قبال کاربردهایی که سرانجام نتایج کارش خواهد داشت؛ از موضوعاتی بوده که وایزن باوم در قالبی سازمان یافته، به بررسی آنها پرداخته است.

او همچنین نویسنده کتاب «توان کامپیوتر و عقل انسانی»^۴ است؛ کتابی که در ایالات متحده بسیار بحث‌انگیز بود و با استقبال فراوانی روبه‌رو شد. این مصاحبه در «جشنواره نرم‌افزار اروپا» که اخیراً در مونیخ برگزار شد، تهیه شده است.

سؤال: پروفیسور وایزن باوم، هوش مصنوعی در طی چند دهه اخیر دستخوش تغییر معنایی شده است. امروز هوش مصنوعی به چه معناست؟

وایزن باوم: هوش مصنوعی در دو سوی اقیانوس اطلس دو معنای متفاوت پیدا کرده است زیرا تعریف هوش در آمریکا با اروپا متفاوت است. به نظر من اروپایی‌ها گرایش محافظه‌کارانه‌تر و عاقلانه‌تری نسبت به هوش مصنوعی دارند. در آمریکا راجع به هوش مصنوعی و اینکه چگونه در عرض ده سال آینده انقلابی در جهان برپا خواهد کرد، زیاد صحبت می‌شود. در باره امکانات بالقوه آن، اظهار نظرهای آمیخته به احساسات فراوان است ولی مسئله اصلی یافتن کاربردهای ملموس برای آن است.

جامعه ما به آن درجه از بلوغ نرسیده که با تکنولوژی و تجهیزاتی که از قبل داریم کنار بیاید.

سؤال: ظاهراً شما با نظرات پروفیسور ماروین مینسکی، یکی از همکارانتان در انستیتوی تکنولوژی ماساچوست، مبنی بر اینکه هدف نهایی هوش مصنوعی غلبه بر مرگ است، موافق نیستید. آیا می‌توانید این نگرش نسبت به جاودانگی با کمک کامپیوتر را بپذیرید؟

وایزن باوم: اگر آن را بپذیرید باید قبول کنید که وسیله هدف را توجیه می‌کند. ولی من موافق نیستم. البته سؤال اینجاست که حتی اگر این کار از نظر فنی عملی باشد، آیا انجام آن خوشایند است. به نظر من این کار به معنای سقوط فرهنگ بشری است.

سؤال: امکانات بالقوه هوش مصنوعی برای انجام تغییرات اجتماعی چیست؟

وایزن باوم: این مسئله بیشتر به اوضاع اجتماعی بستگی دارد تا به صنعت کامپیوتر. در جامعه‌ای عقلانی مطمئناً کاربردهای معقولی یافت خواهد شد، ولی تصور چنین جامعه‌ای بسیار دشوار است.

سؤال: شما یکی از بنیان‌گذاران سازمانی موسوم به «متخصصان کامپیوتر مدافع مسئولیت اجتماعی» هستید. این سازمان چه می‌کند و اعضایش چه کسانی هستند؟

جوزف وایزن باوم^۱ استاد بازنشسته انستیتوی تکنولوژی ماساچوست است. او در سال ۱۹۲۳ در برلین به دنیا آمد و در سال ۱۹۳۶ همراه والدینش به آمریکا مهاجرت کرد. در سال ۱۹۴۵ از دانشگاه وین در دیترویت در رشته ریاضیات فارغ‌التحصیل شد. در اوایل دهه ۵۰ روی کامپیوتری قیاسی به نام «تحلیل‌گر تفاضلی بوش» کار کرد و در طراحی یک کامپیوتر رقمی همکاری داشت. وایزن باوم بعداً به گروهی در شرکت جنرال الکتریک پیوست که اولین کامپیوتر خاص عملیات بانکداری توسط آنها ساخته شد. انستیتوی تکنولوژی ماساچوست در سال ۱۹۶۳ به او عنوان «استاد علوم کامپیوتر» را اعطا کرد. او پیش‌از اندیشه‌های بشردوستانه در علوم کامپیوتر است و نسبت به تکنولوژی اطلاعاتی جدید و تأثیرات آن بر فرهنگ و جامعه بشری چندان خوشبین نیست.



وایزن باوم سیستم پردازش لیست اسلیپ^۲ و برنامه «درک» زبان طبیعی الایزا^۳ را در کارنامه تجربیات فنی خویش دارد. او حتی گونه‌ای از الایزا را ساخت که کارش تقلید روانکاوا بود. این برنامه گفته‌های بیماران را به جملات سؤالی تبدیل می‌کرد و همان جملات را به خود بیمار برمی‌گرداند و به این ترتیب با آنها «گفتگو» می‌کرد.

هدف وایزن باوم از طراحی الایزا ارائه کاربردی کاملاً تجربی از هوش مصنوعی بود. اما وقتی دریافت که نه تنها بیماران تمایل به ایجاد وابستگی عاطفی با برنامه دارند، بلکه حتی روانکاوان حرفه‌ای نیز از آن به عنوان قدمی مهم در خودکارسازی روانکاوی یاد می‌کنند؛ بسیار شگفت‌زده شد. این موضوع در کنار سرخوردگی او از نقش تکنولوژی جدید در جنگ ویتنام، باعث تغییر اساسی در طرز تفکر علمی وایزن باوم شد. از آن پس مسائلی بنیادین از قبیل تردید در باره

* "Ethics at the leading edge", DIALOG, no. 2, 1991

^۱ Joseph Weizenbaum

^۲ SLIP

^۳ ELIZA

سؤال: بر اساس استعاره‌ای آشنا، بشریت سوار بر قطاری است که با سرعت به سوی ورطه نابودی پیش می‌رود. آیا هنوز فرصت داریم که جهت قطار را عوض کنیم یا اینکه فقط می‌توانیم زمان وقوع فاجعه را به تأخیر بیندازیم؟

وایزن‌باوم: من با این نظر موافق نیستم که تنها این دو راه پیش روی ماست. مسلماً می‌توانیم ترمز کنیم، اما همانطور که هنگام عوض کردن مسیر ممکن است اتفاق بیفتد، می‌ترسم فاجعه مهلت ندهد. تا هنگامی که واقعاً در لبه پرتگاه نباشیم، سر عقل نمی‌آییم. نمی‌دانم آیا تا آن موقع وقت کافی داریم تا چیزی را تغییر دهیم.

دانشمند نمی‌تواند مسئولیتهای اخلاقیش را نادیده بگیرد و بگوید، «من فقط تحقیق می‌کنم، آنچه دیگران بعداً با یافته‌هایم می‌کنند، ربطی به من ندارد.»

سؤال: آیا به نظر شما افکاری وجود دارند که نباید راجع به آنها اندیشید؟
وایزن‌باوم: نه. نمی‌توانید مردم را از اندیشیدن بازدارید. بهتر است این‌طور بگوییم که آیا افکاری وجود دارند که نباید دنبالشان را گرفت؟ جهان آزمایشگاه خصوصی دانشمندان نیست. مهندسی ژنتیک مثالی روشن از تحقیقاتی است که از نظر اخلاقی مورد تردید است.
سؤال: این دیدگاه تفاوتی با نظریاتی که اعمال سانسور را در علوم مجاز می‌دانند، ندارد.

وایزن‌باوم: ترجیح می‌دهم آن را خودداری داوطلبانه بنامم. دانشمند نمی‌تواند مسئولیتهای اخلاقیش را نادیده بگیرد و بگوید، «من فقط تحقیق می‌کنم، آنچه دیگران بعداً با یافته‌هایم می‌کنند، ربطی به من ندارد.» به عنوان دانشمند باید از خود بیرسم نتایج تحقیقاتم در چه راههایی ممکن است به‌کار گرفته شود. آیا آثار آن برای بشریت مفید است یا مضر؟ اگر نتیجه تحقیقات، بالقوه مضر باشد اصلاً نباید آن را شروع کرد. مشخص کردن این مرز آسان نیست — هر کس خود باید آن را بیابد.
ترجمه سعید وحید



وایزن‌باوم: سازمان ما «انجمن جهانی فیزیکدانان مخالف جنگ هسته‌ای» را الگوی خود قرار داده و در پاسخ به طرح جنگ ستارگان شکل گرفته است. ما گفتیم که طرح جنگ ستارگان عملی نبوده و نیست. اکنون نزدیک به ۵۰۰۰ عضو داریم، شاید اندکی کمتر. رقم دقیقش را نمی‌دانم. آنها از تمام زمینه‌های کاری هستند — از برنامه‌نویس و مهندس نرم‌افزار گرفته تا دانشگاهیان.

سؤال: سازمان چه تأثیری داشته است؟

وایزن‌باوم: در طول سالهای اخیر هرگاه مسئله‌ای راجع به کامپیوتر در کنگره مطرح شده، سازمان ما طرف مشورت قرار گرفته است. اکنون دفتری در واشنگتن داریم، گرچه کوچک است اما کارمندی دائمی در آنجا مشغول به‌کار است. و از ما مرتباً برای ارائه نظرات کارشناسی در کمیته‌های کنگره دعوت می‌شود.

سؤال: شما دانشمندان رشته‌های پیشروی علمی را دعوت کرده‌اید که موقتاً تحقیقاتشان را متوقف کنند. این نظر عملی است یا ساده‌لوحانه؟ به عبارت دیگر آیا این فراخوان از نظر شما ضروری است و حتی ممکن است دیگران نیز ضرورت آن را درک کنند اما امکان دارد در دراز مدت با بی‌اعتنایی مواجه شود؟

وایزن‌باوم: اگر این سؤال را یکی دو سال قبل از من می‌پرسیدید، جوابم مثبت بود. ولی اوضاع نسبت به آن موقع تغییر کرده است. حتی رئیس «فرهنگستان علوم امریکا» که سازمانی بسیار مهم است نظرات مشابهی را ابراز کرده و اعلام کرده است که شاید لازم باشد اجرای نهایی برخی از پروژه‌ها به تأخیر بیفتد. بنابراین هنوز این مسئله جای بحث دارد.



مهمتر از آن، برنامه‌های تحقیقاتی در ایالات متحده اخیراً با مشکلات مالی مواجه بوده‌اند. این مشکلات در مقیاسی است که تصورش در اروپا امکان‌پذیر نیست. این خود دلیلی دیگر برای متوقف کردن برخی پروژه‌ها یا حتی شروع نکردن آنهاست.

سؤال: و این همان چیزی است که شما دیگران را نهایتاً به آن فراخوانده‌اید؟

وایزن‌باوم: بگذارید این‌طور بگویم: جامعه ما به آن درجه از بلوغ نرسیده که با تکنولوژی و تجهیزاتی که از قبل داریم کنار بیاید.

واژه‌نامه

تغییرات و اضافات واژه‌نامه مقدماتی

کمیته بازیینی واژه‌نامه در نشست‌های متعدد خود، بازیینی واژه‌هایی را که با حرف M شروع می‌شوند به پایان رسانده است. در زیر، نتیجه این بازیینی و تغییرات و اضافاتی که توسط کمیته صورت گرفته است، جهت اطلاع علاقه‌مندان درج می‌گردد.

اعضای گرامی انجمن و دیگر خوانندگان ماهنامه می‌توانند برای ابراز نظرات خود و یا دانستن علل تغییرات اعمال شده و واژه‌های پیشنهادی، کتباً با کمیته بازیینی واژه‌نامه به نشانی انجمن تماس حاصل نمایند. کار بازیینی واژه‌نامه مقدماتی ادامه دارد و نتایج حاصله، حرف به حرف به اطلاع اعضای گرامی انجمن و خوانندگان ماهنامه خواهد رسید. کمیته بازیینی واژه‌نامه امیدوار است که از نظرات سازنده صاحب‌نظران و علاقه‌مندان برخوردار شود. قبل از ارائه فهرست واژه‌ها، یادآوری نکات زیر را لازم می‌دانیم:

۱- در یافتن معادل فارسی برای واژه‌ها، غالباً جنبه «اسمی» واژه‌ها در نظر گرفته شده است.

۲- واژه‌های غیرکامپیوتری واژه‌نامه مقدماتی حذف گردیده‌اند.

۳- معادل بعضی از واژه‌ها کلاً تغییر یافته است و برای بعضی دیگر معادل یا معادلهای جدیدی به معادلهای قبلی اضافه شده و یا تقدم و تأخر معادلهای پیشنهادی تغییر یافته است.

۴- بدیهی است که به مرور زمان ممکن است تغییرات و اضافات دیگری نیز پیشنهاد شوند که نتایج مربوطه به تدریج اعلام خواهند شد.

M

MACHINE	ماشین	MAIN PROGRAM	برنامه اصلی	MARK	نشان
MACHINE ADDRESS	نشانی، ماشین	MAINFRAME	کامپیوتر بزرگ	MARK-SENSE READER	نشان خوان
MACHINE CHECK INTERRUPT	وقفه بررسی ماشین	MAINTAINABILITY	نگهداشت پذیری، قابلیت نگهداری	MARKER	نشانگر
MACHINE LANGUAGE	زبان ماشین	MAINTENANCE	نگهداری، نگهداشت	MASK	پوشش، نقاب
MACHINE LEARNING	یادگیری ماشین، یادگیری ماشینی	MAINTENANCE TESTING	آزمایش نگهداری	MASKABLE INTERRUPT	وقفه پوشش پذیر
MACHINE TRANSLATION	ترجمه ماشینی	MAINTENANCE TOOL	ابزار نگهداری	MASKING	پوشش، نقاب گذاری
MACHINE-READABLE	خواندنی توسط ماشین	MAJOR CYCLE	چرخه بزرگ، چرخه اصلی	MASS MEMORY	حافظه انبوه
MACRO	کلان، دستور کلان، کلان‌دستور	MALFUNCTION	سوء عمل، اشکال	MASTER	اصلی، راهبر
MACRO ASSEMBLER	همگذار کلان	MANAGEMENT	مدیریت	MASTER FILE	پرونده اصلی
MACRO CALL	فراخوانی دستور کلان	MANAGEMENT INFORMATION SYSTEM	سیستم اطلاعات مدیریت	MASTER INDEX	نمایه اصلی
MACRO GENERATOR	مولد دستور کلان			MASTER-CONTROL PROGRAM	برنامه کنترلی اصلی
MACRO-INSTRUCTION	دستور کلان، کلان‌دستور	MANIPULATION	دستکاری	MASTER-SLAVE	فرمانده و فرمانبر، راهبر و پیرو
MACROCODE	برنامه کلان	MANTISSA	ماتیس	MATCH	تطبیق
MACROPROGRAMMING	برنامه‌سازی کلان	MANUAL	کتاب، راهنما، دستی	MATCHING ERROR	خطای تطبیق
MAGIC SQUARE	مربع جادویی، مربع وقتی	MANUAL SCHEDULING	زمانبندی دستی	MATHEMATICAL	ریاضی
MAGNETIC	مغناطیسی	MANUFACTURING	ساخت	MATHEMATICS	ریاضیات
MAGNETIC DISK	دیسک مغناطیسی	MANY-VALUED VARIABLE	متغیر چند مقدار	MATRIX	ماتریس
MAGNETIC TAPE	نوار مغناطیسی			MATRIX PRINTER	چاپگر ماتریسی
MAGNETIC-TAPE READER	نوار مغناطیسی خوان، نوار خوان	MANY-VALUED LOGIC	منطق چند ارزشی	MAXIMAL	بیشین
		MAN/MACHINE INTERFACE	رابط انسان ماشین	MAXIMUM	بیشینه، حداکثر
MAILBOX	صندوق پست، جعبه پیام، پیامدان	MAPPING	نگاشت	MAZE	ماز، مسیر پیچ و خم
MAIL-MERGE	ادغام پیام	MARGIN	حاشیه، مرز	MEAN	میانگین
MAIN MEMORY	حافظه اصلی	MARGINAL CHECK	بررسی مرزی	MEANS-ENDS ANALYSIS	تحلیل وسیله هدف
				MEASUREMENT	اندازه‌گیری

MECHANICAL TRANSLATION	ترجمه ماشینی	MICRO	ریز	MONITOR PROGRAM	برنامه مبصر
MECHANISM	سازوکار، مکانیسم	MICROCODE	ریز برنامه	MONITORING	نظارت، دیده‌بانی
MEDIA	رسانه‌ها	MICROCOMPUTER	ریز کامپیوتر، ولانک	MONOLITHIC PROGRAMMING	برنامه سازی یکپارچه
MEDIAN	میان	MICROPROCESSOR	ریز پردازنده	MONOPROGRAMMING	تک برنامه سازی
MEDIUM FREQUENCY	بسامد متوسط	MICROPROGRAM	ریز برنامه	MONOSTABLE	تک پایا
MEDIUM-SCALE	در مقیاس متوسط	MICROPROGRAMMING	ریز برنامه سازی	MONOTONIC	یکنواخت
MEDIUM-SCALE INTEGRATION	مجموع سازی در مقیاس متوسط	MICROSCHEDULER	ریز زمانبند، زمانبند خرد	MONTE-CARLO METHOD	روش مونت کارلو
MEDIUM-SPEED NETWORKS	شبکه‌های با سرعت متوسط	MICROSWITCH	ریز گزینه	MOTHERBOARD	تخته مدار مادر، تخته مدار اصلی
MEDIUM-TERM SCHEDULING	زمانبندی میان مدت	MILESTONE	مقطع مهم	MOUNT	سوار کردن، نصب کردن
MEGA	میلیون، مگا	MINIATURIZATION	کوچک سازی	MOUNTED	سوار شده، نصب شده
MEGABYTE	میلیون بایت، مگا بایت	MINICOMPUTER	کامپیوتر کوچک	MOUSE	موش
MEGACYCLE	میلیون چرخه، مگا چرخه	MINIMAL	کمین	MOVABLE-HEAD	نوک متحرک
MEGAFLOP	یک میلیون عمل روی اعداد با ممیز شانور	MINIMAX PROCEDURE	رویه کمین-بیشین، رویه حداقل-حداکثر کمین	MOVE	حرکت، نقل مکان
MEGAHERTZ	میلیون هرتز، مگا هرتز	MINIMUM	کمینه، حداقل	MULTI-ADDRESS	چند نشانی
MEMORY	حافظه	MINIMUM WEIGHT SPANNING TREE	درخت پوشا با کمترین وزن	MULTI-PASS	چند گذری
MEMORY ACCESS TIME	زمان دستیابی به حافظه	MINIMUM SPANNING TREE	کوچکترین درخت پوشا	MULTI-PURPOSE	چند منظوره
MEMORY ALLOCATION	تخصیص حافظه	MISSING PAGE	صفحه غایب	MULTI-USER	چند کاربری، چند کاربره
MEMORY CYCLE	چرخه حافظه	MIXED	آمیخته، ترکیبی	MULTICHANNEL	چند مجرای
MEMORY LOCATION	مکان حافظه	MIXED MODE	حالت آمیخته	MULTIKEY	چند کلیدی
MEMORY MANAGEMENT	مدیریت حافظه	MNEMONIC	یادمان، حفظی	MULTILINKED STRUCTURE	ساختار چند پیوندی
MEMORY MAP	نقشه حافظه	MNEMONIC CODE	رمز یادمان	MULTIPLE INHERITANCE	وراثت چند گانه
MEMORY SNAPSHOT	تصویر لحظه‌ای حافظه	MNEMONIC OPCODE	رمز عمل یادمان	MULTIPLE PARTITION	چند بخشی
MEMORY-RESIDENT	مقیم حافظه، مستقر در حافظه	MOD	به پیمانه، در سنج	MULTIPLEX	تسهیم کردن
MENU	فهرست انتخاب، صفحه گزینش	MODE	حالت، باب	MULTIPLEXED	تسهیم شده
MENU-DRIVEN	فهرستی، گزینشی	MODAL	وجهی	MULTIPLEXOR	تسهیم کننده
MERGE	ادغام	MODAL LOGIC	منطق موجها	MULTIPLEXOR CHANNEL	مجرای تسهیم کننده
MERGE PATTERN	الگوی ادغام	MODEL	مدل، الگو	MULTIPOINT	چند نقطه‌ای
MERGE-SORT	مرتب کردن ادغامی	MODEM	مودم، تلفیق و تفکیک کننده	MULTIPROCESSING	چند پردازشی
MESH NETWORK	شبکه توری	MODIFICATION	اصلاح	MULTIPROGRAMMING	عملکرد چند برنامه‌ای، چند برنامه‌ای
MESSAGE	پیام	MODULAR	واحدی، پیمانه‌ای	MULTIPURPOSE	چند منظوره
MESSAGE BUFFER	میانگیر پیام	MODULAR PROGRAMMING	برنامه سازی واحدی	MULTISTABLE	چند پایا
MESSAGE SWITCHING	پیام گزینی	MODULARITY	واحد مندی، واحدی بودن	MULTITASKING	چند تکلیفی، چند وظیفه‌ای
METAClass	فوق رده	MODULATE	تلفیق کردن، سوار کردن	MULTITRACK TAPE	نوار چند شیاره
METACOMPILER	فوق مترجم، فوق همگردان	MODULATION	تلفیق، سوار سازی، مولاسیون	MULTIVIBRATOR	نوسان ساز
METAKNOWLEDGE	معرفت مرتبه بالاتر	MODULATION FREQUENCY	بسامد تلفیق	MULTIWAY TREE	درخت چند راهه
METALANGUAGE	فوق زبان	MODULE	واحد، پیمانه	MUTUAL BLOCKING	مسدود کردن متقابل
METANETWORK	فوق شبکه	MODULO-N	به پیمانه N	MUTUAL EXCLUSION	دو به دو ناسازگاری، مانع‌الجمعی
METARULE	فوق قاعده	MOLECULAR STRUCTURE	ساخت مولکولی	MUTUAL EXCLUSIVE	دو به دو ناسازگار، مانع‌الجمع
METHOD	روش	MONADIC	تکین		
		MONITOR	صفحه نمایش (در پایانه)، مبصر (در سیستم عامل)، دیده‌بان		

ترجمه یک کتاب

نفر-ماه افسانه‌ای (۱۱)

نوشته فردریک بروکز

ترجمه علی پارسا

۱۱- یکی را دور بیاندازید

می‌گیرد که فرض کرده اولین محصول ساخته شده را باید به مشتری تحویل داد.

در اغلب پروژه‌ها، به‌ندرت اولین سیستمی که ساخته می‌شود قابل استفاده است. ممکن است این سیستم کند باشد، زیادی بزرگ باشد، استفاده از آن مشکل باشد یا همه این عیبها را یکجا داشته باشد. چاره دیگری نیست جز اینکه دوباره شروع به کار کنیم و با زیرکی بیشتری نسخه دوباره طراحی شده‌ای از سیستم بسازیم که در آن این عیبها رفع شده است. کار دور انداختن طرح قبلی و طراحی مجدد را می‌شود یکجا انجام داد و یا خرده خرده به آن پرداخت. آنچه که تجربه تمام سیستمهای بزرگ نشان می‌دهد این است که کار دور انداختن به هر نحوی که باشد، به‌رحال انجام خواهد گرفت. اگر مفاهیم جدید یا تکنولوژی نو در سیستم به کار گرفته می‌شوند آنگاه باید حتماً سیستمی ساخت و آن را کنار گذاشت، چون حتی بهترین روشهای برنامه‌ریزی نیز آنقدر قدرتمند نیستند که چنین سیستمی را در بار اول درست

مهندسان شیمی مدتهاست دریافته‌اند که فرایندی را که در آزمایشگاه درست کار می‌کند نمی‌توان فقط در یک مرحله در یک کارخانه پیاده کرد. یک مرحله میانی بنام کارخانه آزمایشی لازم است تا تجربه لازم برای بالا بردن سطح کار و فعالیت در محیط غیرحفاظت شده را فراهم کند. به عنوان مثال، فرایندی آزمایشگاهی برای نمک زدایی آب را ابتدا در کارخانه‌ای آزمایشی با ظرفیت ۴۵۰۰۰ لیتر در روز آزمایش می‌کنند تا بعد تصفیه‌خانه‌ای با ظرفیت ۹ میلیون لیتر در روز را برای استفاده عمومی تأسیس کنند.

سازندگان سیستمهای برنامه‌سازی نیز با این درس برخورد داشته‌اند ولی به نظر می‌رسد که هنوز آن را فراموش کرده‌اند. در پروژه‌های پی‌درپی الگوریتمهایی طرح می‌شود و بعد هجوم می‌آورند تا نرم‌افزارهای قابل ارائه به مشتری را برای آنها بسازند و همه اینها با برنامه زمانی‌ای انجام



محدودتر کرد، چرا که در غیر این صورت هیچوقت محصولی حاصل نخواهد شد.

اما با این همه، برخی تغییرات در اهداف پروژه ناگزیر است و بهتر آن است که آماده قبول آنها باشیم نه اینکه فرض کنیم هیچگاه مطرح نخواهند شد. نه فقط تغییرات در اهداف پروژه ناگزیرند، بلکه از تغییر در نحوه تولید و روشهای فنی نیز گریزی نیست. مفهوم «یکی را دور بیندازید» فقط قبول این حقیقت است که وقتی انسان چیزی می آموزد طرحتش را هم تغییر می دهد.

سیستم را برای تغییر کردن برنامه ریزی کنید

روشهای طراحی سیستمها برای انواع تغییرات بسیار مشهورند و در نوشته ها و مقالات نیز بسیار در این باره بحث شده است - و شاید بیشتر بحث شده تا اینکه عمل شده باشد. از جمله این روشها می توان از قطعه بندی سنجیده، استفاده گسترده از زیرروالها، تعریف دقیق و کامل فاصلهای بین قطعات، و بالاخره مستندسازی تمام اینها نام برد. در درجه بعدی می توان لزوم فراخوانهای استاندارد شده و روشهای مبتنی بر جدولها در صورت امکان را نیز ذکر کرد.

از همه مهمتر استفاده از زبان سطح بالا و روشهای «خودمستندساز» است که خطاهای ناشی از تغییر را کاهش می دهد. استفاده از عملیات زمان کامپایل برای افزودن جملات اعلانی استاندارد نیز به نحو مؤثری در اعمال تغییرات یاری می رسانند.

تعیین یک میزان کمی از تغییرات نیز یک روش اساسی است. هر محصول باید دارای نسخه های شماره دار باشد و هر نسخه باید دارای برنامه زمانی و تاریخ «انجماد» مخصوص به خود باشد یعنی پس از آن تاریخ باید تغییرات را روی نسخه دیگری اعمال کرد.

تشکیلات را نیز برای تغییرات برنامه ریزی کنید

کاسگرو مبلغ این نظر است که باید تمام برنامه ریزها و فرهنگ شمارها و برنامه های زمانی را موقتاً تلقی کرد تا کار اعمال تغییرات ساده تر شود. اما این کار دیگر خیلی افراطی است چرا که رایجترین اشکال گروه های برنامه سازی امروزی از کمبود کنترل مدیریتی ناشی می شود و نه از کنترل بیش از اندازه مدیران.

اما با این همه او نکته مهمی را مطرح می کند. او معتقد است که عدم تمایلی که در مورد مستند کردن طراحی وجود دارد فقط ناشی از تنبلی یا کمبود وقت نیست، بلکه نتیجه عدم تمایل طراح به قرار گرفتن در موضع دفاع از طراحی است که می داند موقتاً است و تغییر خواهد کرد. او می گوید: «طراح با مستند کردن طرح، خود را در معرض انتقاد همگان قرار می دهد و باید قادر به دفاع از هر چه که می نویسد باشد. اگر ساختار تشکیلات به نحوی تهدیدآمیز به نظر رسد، آنگاه هیچ چیز تا زمانی که قابل دفاع کردن باشد، مستند نخواهد شد.» سازماندهی تشکیلات به طوری که آماده قبول تغییرات باشد بسیار

بسازند.

بنابراین مسئله ای که در مقابل مدیریت قرار دارد این نیست که آیا باید یک سیستم آزمایشی بسازیم و آن را دور بیندازیم یا نه. چون این کار را به هر حال خواهیم کرد. تنها مسئله ای که هست این است که آیا باید در آغاز کار برنامه ای برای ساختن یک سیستم دوراندختنی داشته باشیم، یا اینکه قول تحویل این سیستم دوراندختنی را به مشتری بدهیم. اگر مسئله را به این صورت بنگریم آنگاه جواب دادن به آن بسیار ساده تر خواهد بود. اگر سیستم دوراندختنی را به مشتری تحویل دهیم در وقت صرفه جویی کرده ایم، ولی این کار به قیمت ناخشنودی مشتری، دلسردی سازندگان (زمانی که به طراحی مجدد آن مشغولند) و شهرت نامطلوب برای محصول (که حتی بهترین طراحی مجدد نیز نمی تواند آن را جبران کند) تمام خواهد شد.

بنابراین برای دور انداختن اولین سیستم برنامه ریزی کنید، چرا که به هر حال آن را دور خواهید انداخت.

تنها امر ثابت تغییر است

زمانی که به این نتیجه می رسیم که باید یک سیستم آزمایشی بسازیم و آن را دور بیندازیم، و نیز معلوم می شود که طراحی مجدد بر پایه اندیشه های تغییر یافته لازم است، آنگاه مفید است که خود را برای رو در رویی با کل مفهوم تغییر آماده کنیم. اولین قدم این است که تغییر را به عنوان یک حقیقت زندگی بپذیریم نه اینکه آن را استثنایی مزاحم و ناخواسته ببینداریم. کاسگرو به فراست این حقیقت را متذکر شده که کار برنامه ساز ارائه محصولی قابل لمس نیست بلکه او باید رضایت مشتری را تأمین کند. و باید توجه داشت که هم نیاز واقعی مشتری و هم تلقی مشتری از این نیاز، در جریان ساختن و آزمایش و استفاده برنامه کامپیوتری تغییر می کند.

برای دور انداختن اولین سیستم برنامه ریزی کنید، چرا که به هر حال آن را دور خواهید انداخت.

البته این امر در مورد محصولات سخت افزاری اعم از اتومبیلهای جدید یا کامپیوترها نیز صادق است. اما در این موارد وجود داشتن یک محصول قابل لمس باعث محدود شدن و معین بودن تقاضای مشتری برای اعمال تغییرات می شود. در حالی که سهولت کار نوشتن نرم افزار و نیز غیر ملموس بودن آن، باعث می شود که سازندگان آن مدام در معرض درخواستهای تغییر باشند.

دور باد از من اگر بخواهم بگویم که تمام تغییرات مورد نظر مشتری را می توان، یا باید در طراحی در نظر گرفت. مسلم است که باید برای این امر حدی قائل شد، و این حد را باید با پیشرفت کار محدودتر و

افراد ارشد فنی باید آموزش مدیریتی فراهم کرد. اهداف پروژه، نحوه پیشرفت آن، و مشکلات مدیریتی را باید با تمام افراد ارشد در میان گذارد.

در صورتی که استعداد افراد فنی ارشد اجازه دهد باید آنان را از نظر روحی و فنی آماده نگاه داشت تا به مدیریت یک گروه یا برنامه‌سازی با دستهای خود بپردازند. اجرای این کارها مسلماً زحمت زیادی دارد ولی حتماً به زحمتش می‌ارزد!

غرض از سازماندهی یک گروه برنامه‌سازی از نوع گروه جراحی نیز حمله‌ای اساسی به همین مسئله است. تأثیر این روش این است که یک فرد ارشد وقتی به برنامه‌سازی می‌پردازد احساس کسر شأن نمی‌کند، و هدف این روش برطرف کردن موانع اجتماعی در راه لذت بردن او از این کار خلاق است.

علاوه بر این، ساختار گروه جراحی برای به حداقل رساندن ارتباطات درون‌گروهی طرح شده است. به همین دلیل، این روش کار تغییر دادن سیستم و گماردن تمام گروه جراحی را به فعالیت برنامه‌سازی دیگر، در صورتی که تغییرات سازمانی لازم باشد، بسیار ساده می‌کند. این روش حقیقتاً پاسخی درازمدت به مسئله سازمان انعطاف‌پذیر است.

دوگام به‌پیش و یک گام به‌پس

یک برنامه، اگر برای استفاده به مشتری تحویل شود، باز هم در معرض تغییر خواهد بود. تغییراتی را که بعد از تحویل انجام می‌شوند نگهداری برنامه می‌نامند ولی این فرایند اساساً با نگهداری سخت‌افزار تفاوت دارد.

نگهداری یک سخت‌افزار در مورد یک سیستم کامپیوتر شامل سه نوع فعالیت است — تعویض قطعات خراب شده، تمیزکاری و روغن‌کاری، و افزودن تغییرات مهندسی برای رفع اشکالات طراحی. (اغلب تغییرات مهندسی، و نه همه آنها، اشکالات متحقق‌سازی و اجرایی را برطرف می‌کنند و به اشکالات مربوط به معماری سیستم نمی‌پردازند، بنابراین از دید کاربر مخفی هستند).

نگهداری برنامه کاری به تمیزکاری و روغن‌کاری و تعمیر خرابیها ندارد بلکه عمدتاً به تغییراتی برای رفع اشکالات طراحی می‌پردازد. برخلاف سخت‌افزار، این تغییرات شامل افزودن به عملکرد سیستم می‌شود و معمولاً برای کاربر آشکار است.

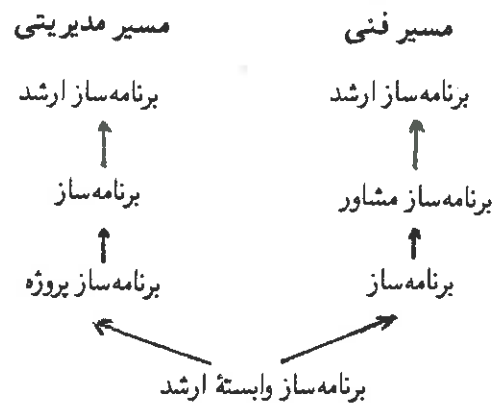
هزینه کل نگهداری یک برنامه پرمصرف معمولاً ۴۰ درصد یا بیشتر هزینه تولید آن است. عجیب آن است که این هزینه به شدت متأثر از تعداد کاربران سیستم است. هرچه کاربران بیشتر باشند، خطاهای بیشتری یافت می‌شود.

بتی کمپیل از آزمایشگاه علوم هسته‌ای ام‌آی‌تی به سیر جالبی در طول عمر یک نسخه از برنامه اشاره می‌کند که در شکل ۲-۱۱ نشان داده شده است. در آغاز خطاهای قدیمی که در نسخه‌های قبلی پیدا شده و برطرف گردیده بودند در نسخه جدید دوباره سر برمی‌آورند. معلوم می‌شود که امکانات جدید در نسخه تازه نیز اشکالاتی دارند.

مشکلتر از طرح یک سیستم که قابل تغییر باشد، است. هر یک از افراد را باید به کارهایی گمارد که دامنه معلومات او را وسیع کند و بدین ترتیب باعث افزایش انعطاف‌پذیری کامل نیروهای سازمان شود. در یک پروژه بزرگ بهتر است مدیر پروژه دو سه نفر از برنامه‌سازان قابل به صورت سواره نظامان فنی نگه دارد تا جاهایی که نبرد شدید می‌شود به کمک همکاران بشتابند.

ساختار مدیریت نیز همراه با تغییر در سیستم نیاز به تغییر دارد. معنی این مطلب این است که رئیس باید توجه زیادی به این امر داشته باشد که مدیران و افراد فنی او تا حدی که استعدادشان مجاز می‌دارد قابل جابه‌جایی باشند.

موانعی که در این راه وجود دارد جنبه‌های اجتماعی دارند و باید با آنها با هشیاری دائمی مبارزه کرد. در درجه اول خود مدیران و افراد فنی خود را «ارزشمندتر» از آن می‌دانند که به کار عملی برنامه‌سازی بپردازند. در درجه بعد مشکل این است که کارهای مدیریتی دارای اعتبار بیشتری شناخته می‌شوند. برای غلبه بر این مشکل برخی از مؤسسات، مثل آزمایشگاههای بل، تمام عناوین شغلی را حذف کرده‌اند. در این مؤسسات هر کارمندی فقط «عضوی از گروه فنی» است. مؤسسات دیگر مثل آی‌بی‌ام، از مسیر دوگانه ارتقای شغلی، مانند شکل ۱-۱۱، استفاده می‌کنند. پله‌های متناظر در هر دو مسیر از لحاظ نظری یکسان هستند.



شکل ۱-۱۱ مسیر دوگانه در آی‌بی‌ام

تعیین دستمزدهایی مناسب با این پله‌ها ساده است. آنچه که بسیار مشکلتر است ایجاد اعتبار مناسب است. برای این منظور باید دفترهای کار افراد دارای وسعت برابر و جنبه‌های احترام‌آمیز برابر باشند. خدمات منشیگری و پشتیبانی نیز باید به همین ترتیب یکسان باشند. اگر کسی از مسیر فنی به مسیر مدیریتی در همان مرتبه منتقل می‌شود، هرگز نباید این انتقال او همراه با افزایش حقوق باشد و چنین انتقالی همیشه باید به عنوان «مأموریت جدید»، و نه «ارتقا شغلی»، اعلام شود. در مقابل تجدید مأموریت در جهت عکس (یعنی از مسیر مدیریتی به مسیر فنی) همیشه باید همراه با افزایش حقوق باشد، چرا که در این مورد جبران فشارهای اجتماعی، لازم است.

مدیران را باید به دوره‌های بازآموزی فنی فرستاد، و به همین ترتیب برای

هزینه‌های نگهداری دارند. به همین ترتیب، روشهای اجرای طرح با افراد کمتر و ارتباطگیریهایی کمتر باعث بروز خطاهای کمتری می‌شوند.

یک گام به پیش و یک گام به پس

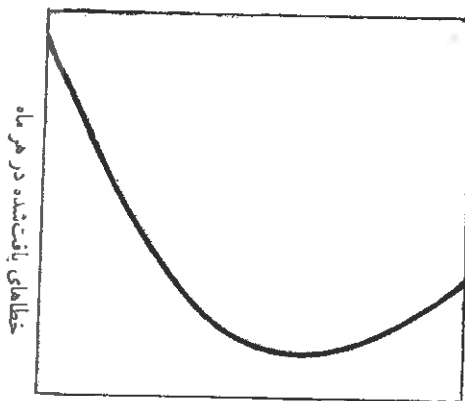
لمان و بیلیدی تاریخچه نسخه‌های متوالی یک سیستم عامل را بررسی کرده‌اند و دریافته‌اند که تعداد کل قطعات برنامه به صورت خطی با افزایش شماره نسخه افزایش می‌یابند، اما تعداد قطعاتی که تغییر یافته‌اند با شماره نسخه به نسبت نمایی تغییر می‌کنند. تمام کارهای تعمیراتی تمایل به تخریب ساختار سیستم دارند و آنتروپی و بی‌نظمی سیستم را افزایش می‌دهند. در جریان کار کوشش کمتر و کمتری صرف رفع اشکالات اولیه طراحی می‌شود در حالی که وقت بیشتر و بیشتری در رفع خطاهایی می‌شود که در اثر تعمیرات قبلی ایجاد شده است. با گذشت زمان سیستم به بی‌نظمی بیشتری دچار می‌شود. بعد از مدتی دیگر فایده‌ای از رفع اشکالات حاصل نمی‌شود یعنی هر گام به پیش با یک گام به پس خنثی می‌شود. با اینکه سیستم به طور اصولی باید تا ابد قابل استفاده باشد، ولی پس از مدتی دیگر به عنوان پایه‌ای برای پیشرفتهای بعدی مفید نیست. علاوه بر این، کامپیوترها نیز تغییر می‌کنند و نیازهای کاربران نیز تغییر می‌کند و به همین دلیل و در واقعیت سیستم تا ابد قابل استفاده نمی‌ماند. پس یک سیستم جدید، که از اول طراحی شده باشد لازم می‌آید.

بدین ترتیب بیلیدی و لمان از یک مدل آماری و مکانیکی به نتایجی عمومی در مورد سیستمهای برنامه‌سازی می‌رسند که با تجربه تمام تاریخ روی زمین تأیید می‌شود. یعنی همان که پاسکال گفته است: «هر چیزی در آغاز در بهترین حالت است». سی. اس. لوتیس این نکته را با احساس بیشتری مطرح کرده است:

این کلید فهم تاریخ است. نیروی زیاد صرف می‌شود تمدنها ساخته می‌شوند نهادهای عالی ابداع می‌گردند. ولی هر بار چیزی به خطا می‌رود. یک اشکال مهلک خودخواهان و ظالمان را حاکم می‌کند، و همه چیز به سوی بدبختی و خرابی باز می‌رود. در واقع ماشین جامعه ریپ می‌زند. اول به نظر می‌رسد که خوب راه می‌افتد و چند متر جلو می‌رود، ولی بعد دوباره خراب می‌شود و می‌ماند.

کار ساختن برنامه‌های سیستم فرایندی کاهش دهنده آنتروپی است و به همین دلیل ذاتاً باثبات است. کار نگهداری برنامه‌ها فرایندی افزایش دهنده آنتروپی است، و حتی اجرای بسیار ماهرانه آن نیز نمی‌تواند از تبدیل شدن نهایی سیستم به موجودی غیرقابل اصلاح و منسوخ جلوگیری کند.

تمام این اشکالات برطرف می‌شوند و تا چند ماه اوضاع روبه‌راه است. سپس میزان خطاها دوباره افزایش پیدا می‌کند. خانم کمپیل معتقد است که این امر ناشی از رسیدن کاربران به سطح حدیدی از پیشرفتگی و پیچیدگی است، یعنی آنها شروع به استفاده از تمام امکانات تازه نسخه جدید می‌کنند. این استفاده زیاد باعث آشکار شدن خطاهای ظریفی می‌شود که در امکانات جدید نهفته هستند.



تعداد ماهها پس از راه اندازی

شکل ۲-۱۱ وقوع خطا به صورت تابعی از عمر نرم افزار

مشکل اساسی نگه‌داری برنامه‌ها این است که رفع یک اشکال به احتمال بسیار (۲۰ تا ۵۰ درصد) باعث ایجاد اشکال دیگری می‌شود. بنابر این کل فعالیت به صورت دو گام به پیش و یک گام به پس انجام می‌گیرد.

چرا رفع اشکالات به نحو تمیزتری صورت نمی‌گیرد؟ اولاً به این دلیل که حتی یک اشکال ظریف نیز خود را به صورت یک نوع خرابی محلی آشکار می‌سازد. در واقع چنین اشکالی ایجابات سراسری در تمام سیستم دارد که اغلب بدیهی نیستند. هر کوشش برای رفع کردن این اشکال با حداقل تلاش، باعث رفع خرابی محلی و بدیهی می‌شود ولی نتایج وسیعتر این رفع خرابی از نظر دور می‌ماند، مگر اینکه ساختار سیستم واضح یا مستندات آن بسیار عالی باشد. ثانیاً به این دلیل که تعمیرکار معمولاً همان کسی نیست که برنامه را نوشته است بلکه اغلب برنامه‌نویس تازه‌کار یا کارآموز است.

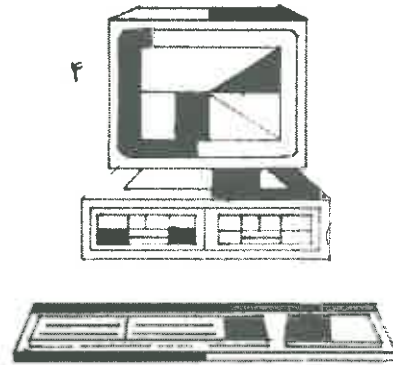
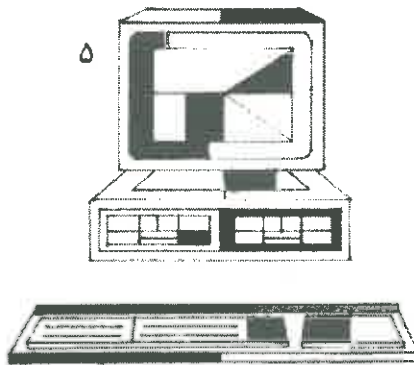
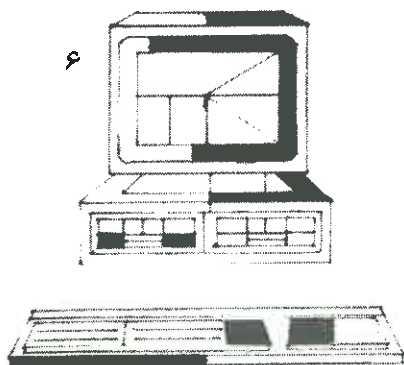
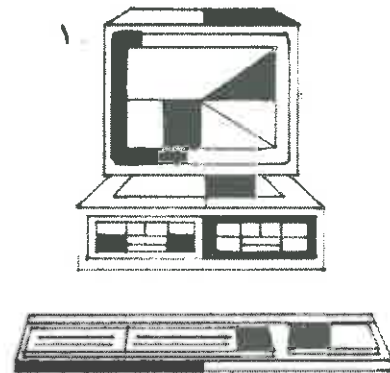
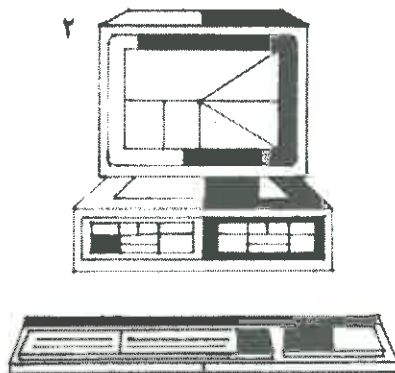
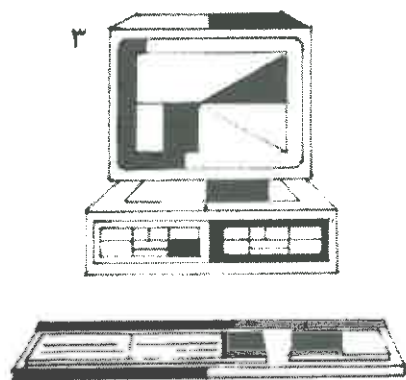
نگه‌داری برنامه، به دلیل اینکه به افزودن خطاهای جدید منجر می‌شود بسیار بیشتر از هر نوع برنامه‌نویسی دیگر نیازمند آزمایش سیستم به ازای هر خط برنامه است. از لحاظ نظری، پس از هر رفع اشکالی باید دوباره تمام موارد آزمایش سیستم را که قبلاً اجرا کرده‌ایم، اجرا کنیم تا مطمئن شویم که سیستم به نحو ناشناخته‌ای آسیب ندیده است. در عمل این آزمایش بازگشتی باید به این حد نظری نزدیک شود و البته بسیار هم پرهزینه است.

مسلم است که روشهای طراحی برنامه‌ها به نحوی که باعث حذف، یا حداقل معلوم شدن آثار جنبی شوند تأثیر بسیار زیادی در کاهش

سرگرمی

شکلهای مشابه

در بین کامپیوترهایی که در زیر می بینید، دو تای آنها شبیه هم هستند. آیا می توانید آنها را پیدا کنید؟



جدول کلمات کامپیوتری

کلمات زیر را از جدول پیدا کرده و خط بزنید. از حروف باقیمانده کلمه دیگری به دست می آید.

R	M	J	E	R	A	W	T	F	O	S
E	E	E	S	T	A	M	R	O	F	E
S	T	S	M	S	U	S	D	R	A	C
A	S	U	C	O	M	P	U	T	E	R
L	Y	O	K	D	R	E	N	T	S	D
E	S	M	S	F	A	Y	K	I	P	N
S	P	R	I	N	T	E	R	T	R	A
A	X	L	D	C	Y	E	U	N	E	M
B	E	S	A	B	R	L	X	R	A	M
A	O	G	O	L	E	O	L	T	D	O
T	S	A	Y	P	P	O	L	F	S	C
A	R	G	R	A	P	H	I	C	H	R
D	E	T	P	R	O	M	P	T	E	O
C	H	I	P	M	H	Z	A	G	E	M
N	E	E	R	C	S	P	P	O	T	I

SCREEN
SPREADSHEET
PRINTER
DISK
SHOPPER
DATABASE
GRAPHIC
KEYBOARD
SYSTEM
DOS

SOFTWARE
MEMORY
PROMPT
MENU
CHIP
COMPUTER
COMMAND
MICRO
MOUSE
RAM

LOGO
INPUT
FORMATS
LASER
FILE
MEGA
GEM
ROM
TEXT
CARDS

فهرست سالانه

فهرست مطالب سال سیزدهم گزارش کامپیوتر ۱۳۷۰

آنچه در زیر به نظر تان می‌رسد فهرست مطالب سال سیزدهم، ۱۳۷۰، شماره‌های ۱ تا ۶، شماره‌های پیاپی ۱۰۶ تا ۱۱۱ است. مطالب برحسب نوع (مقاله، گزارش و غیره) طبقه‌بندی شده و در هر طبقه به ترتیب درج در مجله ارائه شده‌اند.

فهرست ترتیبی مقالات	شماره صفحه	کامپیوترهای چندرسانه‌ای	شماره صفحه
تحول مفهوم پیمانه‌ای بودن نرم‌افزار	۷ ۱۰۶	به جای صفحه‌کلید	۲۸ ۱۱۰
نواده پاسکال	۱۲ ۱۰۶	المپیاد ملی انفورماتیک	۲۹ ۱۱۰
مباحثی در برنامه‌سازی موازی	۱۶ ۱۰۶	سمینار اسپارک	۳۱ ۱۱۰
نفرماه افسانه‌ای (۷)	۲۹ ۱۰۶	شبکه‌ها علیه استقلال	۲۰ ۱۱۱
سیستمهای ذخیره و بازیابی اطلاعات (۱)	۵ ۱۰۷	دومین مرحله المپیاد ملی انفورماتیک	۲۱ ۱۱۱
واحد «سمینار» برای رشته کامپیوتر	۱۴ ۱۰۷	فهرست مطالب گوناگون	۲۲ ۱۱۱
نفرماه افسانه‌ای (۸)	۳۱ ۱۰۷	مشارکت همه‌جانبه (سخن رئیس)	شماره صفحه
سیستمهای ذخیره و بازیابی اطلاعات (۲)	۶ ۱۰۸	شماره خانه (سرگرمی)	۶ ۱۰۶
انفورماتیک در شوروی	۱۵ ۱۰۸	استفاده درست از صفحه‌کلید (کامپیوتر و بهداشت)	۲۰ ۱۰۶
نفرماه افسانه‌ای (۹)	۲۷ ۱۰۸	معرفی کتاب	۲۱ ۱۰۶
آشنایی با یونیکس	۷ ۱۰۹	حرف D (واژه‌نامه)	۲۵ ۱۰۶
مقدمه‌ای بر سیستمهای خبره	۱۵ ۱۰۹	بدون شرح (سخن رئیس)	۲۶ ۱۰۶
نفرماه افسانه‌ای (۱۰)	۳۲ ۱۰۹	مرجع ارتباط انسان و کامپیوتر (معرفی کتاب)	۴ ۱۰۷
زبانهای برنامه‌سازی به مثابه مدل‌های تفکر (۱)	۷ ۱۱۰	حرف E و F (واژه‌نامه)	۲۱ ۱۰۷
آرایه‌های تپنده	۱۵ ۱۱۰	نامزدهای انتخابات هیأت اجرایی (معرفی)	۲۸ ۱۰۷
فوتونیک	۲۰ ۱۱۰	معرفی کتاب	۳۴ ۱۰۷
مجموعه‌های مشکک	۸ ۱۱۱	حرف G و H (واژه‌نامه)	۲۴ ۱۰۸
زبانهای برنامه‌سازی به مثابه مدل‌های تفکر (۲)	۱۴ ۱۱۱	دو انجمن کامپیوتری در بریتانیا (معرفی)	۲۵ ۱۰۸
نفرماه افسانه‌ای (۱۱)	۲۸ ۱۱۱	سمینار ادواری فارابی (معرفی)	۲۳ ۱۰۹
فهرست ترتیبی گزارشها	شماره صفحه	ترازنامه	۲۴ ۱۰۹
دیسکهای نوری	۲۳ ۱۰۶	حرف I و J (واژه‌نامه)	۲۹ ۱۰۹
سمینار نشر کامپیوتری	۲۴ ۱۰۶	گریس هاپر (درگذشتگان)	۳۰ ۱۰۹
المپیاد انفورماتیک	۲۳ ۱۰۷	حساب و کتاب (سرمقاله)	۴ ۱۱۰
اهداف رشته کامپیوتر	۲۵ ۱۰۷	راست و دروغ (سرگرمی)	۶ ۱۱۰
سئوالات المپیاد انفورماتیک	۱۹ ۱۰۸	ترتیب (سرگرمی)	۲۴ ۱۱۰
کنفرانس چندرسانه‌ای	۲۲ ۱۰۸	حرف K و L (واژه‌نامه)	۲۴ ۱۱۰
ائتلاف دو رقیب	۲۳ ۱۰۸	علم و اخلاق	۳۲ ۱۱۰
بمبی به نام ویروس	۲۶ ۱۰۹	حرف M (واژه‌نامه)	۲۲ ۱۱۱
فعالتهای یکساله انجمن	۲۷ ۱۰۹	سالی نویدبخش (سخن رئیس)	۲۶ ۱۱۱
سمینار آموزش و انفورماتیک	۲۵ ۱۱۰		۷ ۱۱۱



در انجمن انفورماتیک ایران عضو شوید

اهداف انجمن

انجمن انفورماتیک ایران در سال ۱۳۵۷ به عنوان یک انجمن مستقل و غیرانتفاعی تأسیس شد. هدفهای انجمن انفورماتیک ایران مطابق اساسنامه عبارتند از: ترویج کاربرد انفورماتیک از طریق وضع استانداردهای علمی، فنی و معنوی، بررسی اثر تکنولوژی کامپیوتر بر وضع اجتماع و محیط زیست، آشنا کردن افراد اجتماع با محاسن و خطرات استفاده از کامپیوتر، کمک در وضع قوانین حمایتی و حفاظتی مربوط به انفورماتیک و کاربردهای آن، بالا بردن سطح دانش در انفورماتیک از طریق کمک به بهبود کیفیت آموزش، توسعه و تشویق فعالیتهای پژوهشی، ایجاد همکاری بین مراکز علمی، صنعتی و اجرایی، و حفظ ارتباط بین متخصصین این رشته به منظور همکاری و تبادل افکار.

اساسنامه انجمن برای نیل به این هدفها چاپ نشریات، تشکیل مجامع علمی، طرح ریزی و بررسی برنامههای آموزشی، ارزیابی تخصصها، تشویق فعالیتهای پژوهشی، توسعه مبادلات علمی فنی و نیروی انسانی، کمک در وضع استانداردها و بالاخره ایجاد یک کتابخانه را پیشبینی کرده است.

مزایای عضویت

کلیه اعضای انجمن از خدمات و امتیازات زیر برخوردار می شوند:

- دریافت رایگان گزارش کامپیوتر و استفاده از تخفیف در بهای سایر نشریات انجمن.
- حق شرکت در سخنرانیها، سمینارها و دورههای مختلف علمی و فنی به صورت رایگان و یا با تخفیف قابل ملاحظه نسبت به افراد غیر عضو.
- دریافت اخبار و اطلاعات مربوط به انفورماتیک، امکان آشنایی و تبادل افکار با سایر کارشناسان این رشته و قرار گرفتن در جریان رویدادهای ملی و بین المللی در انفورماتیک.
- حق استفاده رایگان از کتابخانه و مرکز اسناد انجمن.

شرایط عضویت

تقاضای عضویت در انجمن انفورماتیک ایران باید با معرفی دو نفر اعضای برجسته یا عادی انجمن، مدارک لازم و حق عضویت سالانه همراه باشد. در صورت نداشتن معرف، ارائه معرفی نامه دانشجویی از دانشگاه برای سال تحصیلی جاری (برای دانشجویان) و یا فتوایی آخرین مدرک تحصیلی برای سایرین کافی است. شرایط انواع عضویت در انجمن به شرح زیر است:

عضو عادی: داشتن درجه لیسانس یا بالاتر در رشته انفورماتیک و یا درجه لیسانس همراه با حداقل سه سال سابقه کار مفید در این رشته.

عضو برجسته: داشتن حداقل پنج سال سابقه عضویت عادی در انجمن، سوابق درخشان در کمک به پیشبرد هدفهای انجمن و فعالیتهای برجسته (پژوهشی، علمی،

اجرایی یا آموزشی) در رشته انفورماتیک.

عضو افتخاری: سابقه فعالیتهای سازنده و کمکهایی عمده در پیشبرد هدفهای انجمن (فقط با دعوت).

عضو وابسته: فعالیت در رشته انفورماتیک و یا رشته های وابسته. عضو وابسته می تواند طبق ظوابطی که در آیین نامه های انجمن پیشبینی شده، به عضو عادی تبدیل شود.

عضو دانشجویی: اشتغال به تحصیل در یکی از مراکز معتبر آموزش عالی داخل یا خارج کشور با گواهی مؤسسه مربوطه.

عضو حقوقی: مؤسسه یا سازمان علاقه مند به هدفها و فعالیتهای انجمن.

درخواست (تمدید) عضویت و دریافت نشریات

برای عضویت یا تمدید عضویت و دریافت نشریات انجمن، اطلاعات زیر را تکمیل کرده و به نشانی انجمن بفرستید.

نام و نام خانوادگی: _____

مؤسسه: _____ تحصیلات: _____

نشانی پستی: _____

کد پستی: _____ تلفن: _____

شماره عضویت (برای تمدید عضویت): _____

حق عضویت (برای سال ۷۱-۷۰): _____

☐ عادی، وابسته، برجسته (۲۵۰۰ ریال)

☐ دانشجویی (۱۰۰۰ ریال)

☐ حقوقی (حداقل ۵۰ هزار ریال)

حق عضویت شامل اشتراک رایگان گزارش کامپیوتر (از زمان عضویت تا شهریور ماه آینده) نیز هست. اعضا و مشترکین خارج از کشور که از ایران اقدام می کنند، باید برای هزینه پست مبلغ ۲۵۰۰ ریال علاوه بر مبالغ فوق پرداخت کنند.

☐ اشتراک یکساله گزارش کامپیوتر (بدون عضویت در انجمن، ۲۰۰۰ ریال)

سایر نشریات موجود: _____

☐ کتاب «آزمونی برای طراحان نرم افزار» (۵۰۰ ریال برای اعضا، ۹۰۰ ریال دیگران)

☐ دوره جلد شده گزارش کامپیوتر مربوط به سال ۱۳۶۴ (۱۰۰۰ ریال)

☐ دوره جلد شده گزارش کامپیوتر مربوط به سال ۱۳۶۵ (۱۰۰۰ ریال)

☐ دوره جلد شده گزارش کامپیوتر مربوط به سال ۱۳۶۹ (۲۰۰۰ ریال)

جمع _____ ریال

وجوه مورد نظر را به حساب جاری ۱۱۱۷ بانک صادرات شعبه ۱۶۵۲ واریز کرده، فیش بانکی را برای انجمن بفرستید.

برای عضویت و کسب اطلاعات بیشتر با نشانیهای زیر مکاتبه کنید:

داخل کشور:

انجمن انفورماتیک ایران

صندوق پستی ۱۱۹۶-۱۴۱۵۵

تهران

خارج کشور:

Informatics Society of Iran

c/o Anoosh Hosseini

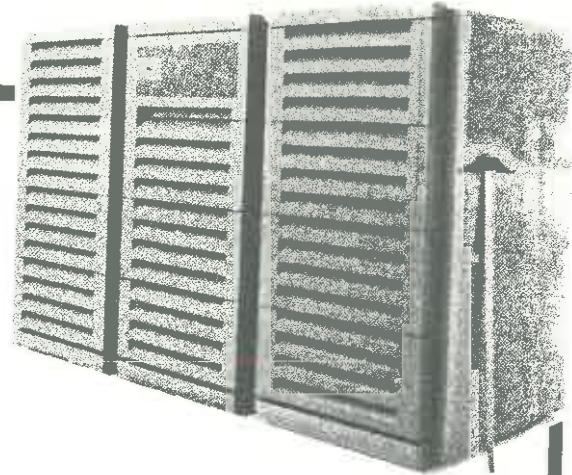
P.O.Box 61622

Sunnyvale, CA 94088

U.S.A

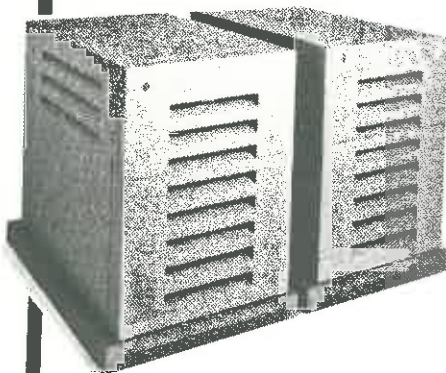


sun microsystems



بدعت گزار در صنعت انفورماتیک جهان

- تکنولوژی و طراحی برتر
- سیستم عامل UNIX بر اساس پردازنده RISC/SPARC
- پیشرو در شبکه بندی OPEN SYSTEMS
- پیشرو در سیستم های CLIENT/SERVER
- پیشرو در سیستم های کامپیوتری به لحاظ PRICE/PERFORMANCE
- پردازنده مرکزی SPARC با طراحی RISC با سرعت 110 MIPS و امکان پردازش موازی (PARALLEL PROCESSING)
- پیشرو در سیستم های توزیعی (DISTRIBUTED)
- OPEN LOOK واسطه گرافیکی کاربران (GRAPHIC USER INTERFACE)
- برای استفاده ساده از کامپیوتر
- بیش از ۳۸۰۰ بسته نرم افزاری برای کاربردهای علمی / مهندسی / ...
- و اداری / مالی



sun تولید کننده سیستم های کامپیوتری سوپر مینی :

microsystems

- ۱ - ایستگاه کار (WORKSTATION) برای کاربردهای ENGINEERING ...
 - ۲ - SERVER برای محیط های CLIENT/SERVER , ELECTRONIC PUBLISHING, MANUFACTURING, SCIENTIFIC ...
 - ۳ - چند استفاده کننده (MULTIUSER) برای کاربردهای آموزشی ، اداری ، DATA BASE SERVER WORKGROUP COMPUTING و
- مالی و با قابلیت اتصال ۳۶ تا ۱۳۰ ترمینال .



شرکت نگاره عرضه کننده و پشتیبان سیستم های کامپیوتری کمپانی

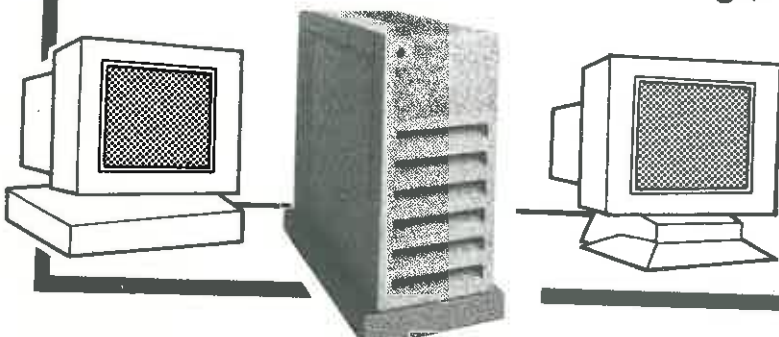
SUN MICROSYSTEMS در ایران .

تهران - خیابان استاد مطهری - خیابان جم (فجر) - حبان تانده

(شهید لطفی) - کوچه بهرامی - پلاک ۷

تلفن: ۸۳۸۶۲۸ - ۴۴۰۸۴۵۰ تلکس: ۲۱۲۴۴۱

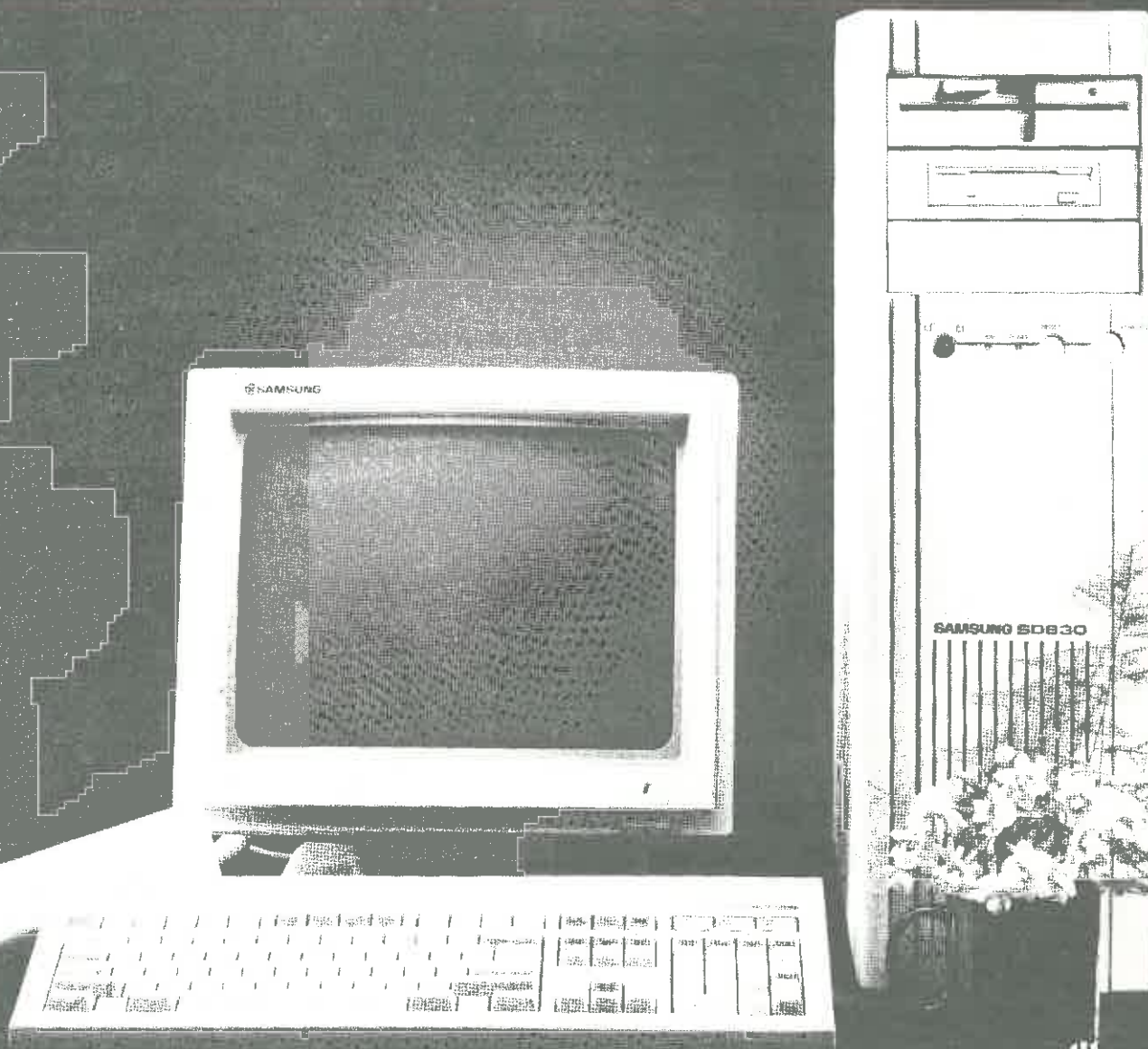
فاکس: ۸۳۵۹۹۷ صندوق پستی: ۱۵۸۷۵/۱۴۱۴



 **SAMSUNG**
Electronics

پویا 

طراحی و ساخت سیستمهای کامپیوتری



خیابان قائم مقام فراهانی، کوچه بیستم، شماره ۶۲، صندوق پستی: ۵۷۴، ۱۴۳۳۵، تهران

تلفن: ۸۶۶۶۷۹-۸۶۳۳۶۷-۸۶۱۷۴۲-۸۲۹۲۳۵-۸۲۸۳۲۰

فاکس: ۸۶۱۸۵۸

موج پویا