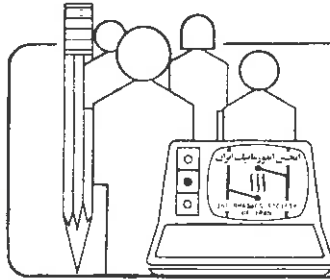


گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران



سال هفتم - شماره ۹

شماره پیاپی ۷۱

آذر ۱۳۶۴

تک شماره ۱۰۰ ریال - اشتراک سالانه ۸۰۰ ریال



اخبار انجمن

انتخاب هفتمین عضو هیأت اجرایی انجمن

با توجه به آن که در انتخابات هیأت اجرایی آقایان داریوش جوان تبریزی، محسن رستمی، و مهرداد دمدرنیا متفقا ۵۹ رأی آورده بودند در اولین نشست هیأت اجرایی جدید، برای انتخاب هفتمین عضو اصلی هیأت اجرایی انجمن و دو عضو علی البدل اول و دوم از میان آنان، بحث و تبادل نظر گردید و قرار شد این امر از طریق قرعه کشی انجام گیرد. در نتیجه، پس از قرعه کشی، آقای دکتر محسن رستمی به عنوان عضو اصلی، آقای مهرداد دمدرنیا به عنوان عضو علی البدل اول، و آقای داریوش جوان تبریزی به عنوان عضو علی البدل دوم انتخاب گردیدند.

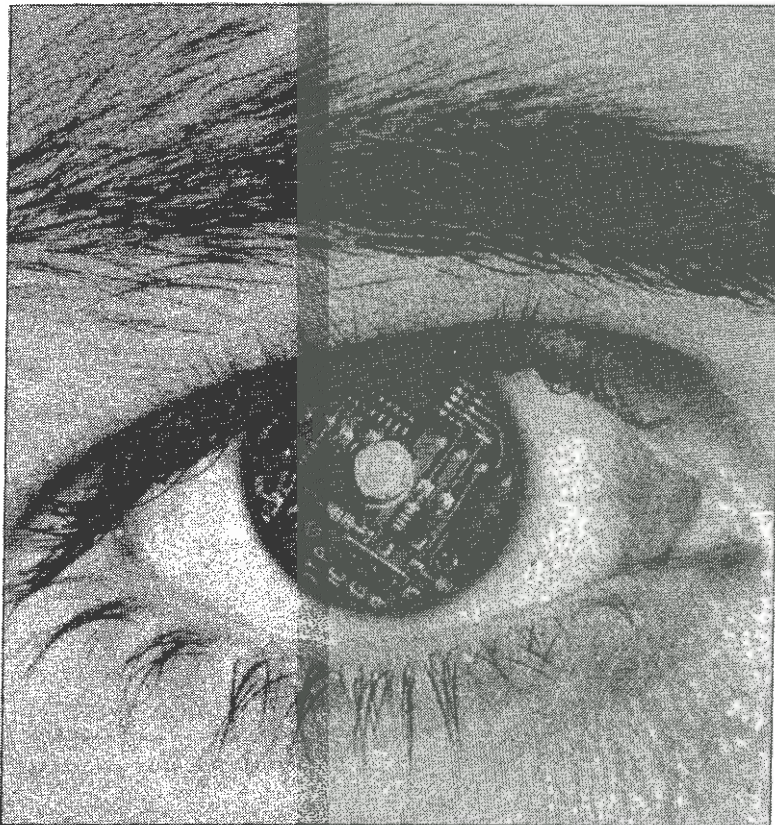
تعیین حدود وظایف و اختیارات مسئولین انجمن

پس از انتخاب هفتمین عضو اصلی هیأت اجرایی انجمن و مشخص شدن دو عضو علی البدل، هیأت اجرایی تصمیم گرفت که قبل از انتخاب مسئولین و سرپرستان کمیته های مختلف انجمن، کلیات مربوط به حدود وظایف و اختیارات آنها تعیین گردند تا افرادی که برای احراز هر سمت خود را نامزد می کنند، آشنائی بیشتری بسبب مسئولیتهای مربوطه داشته باشند. در آینده، حدود وظایف و اختیارات تعیین شده برای مسئولین انجمن، پس از تصویب نهائی به تفصیل به اطلاع اعضای گرامی انجمن خواهد رسید. در همین رابطه، به دلیل حجم زیاد کار، تصمیم گرفته شد که کمیته عضویت و روابط عمومی، به دو کمیته مستقل "عضویت" و "روابط عمومی" تقسیم گردد. لازم به تذکر است که در اساسنامه انجمن نیز به این دو کمیته به صورت جداگانه و مستقل اشاره شده است ولی بنا به دلائل تا کنون در هم ادغام شده بودند.

انتخاب مسئولین جدید انجمن

پس از تشبیه کلیات حدود وظایف و اختیارات مسئولین و سرپرستان کمیته های انجمن، برای تعیین آنها بحث و بررسی به عمل آمد و در هر مورد، بطور جداگانه رأی گیری شد. در نتیجه

ورق بزنید



در این شماره

- | | | | |
|----|---|----|-------------|
| ۱ | انتخاب هفتمین عضو هیأت اجرایی، انتخاب مسئولان جدید انجمن | ۱ | اخبار انجمن |
| ۱۲ | آئینیزکا میوشوردید، ریزپردا زنده، جدیداینتل | ۲ | اخبار رجهان |
| ۲ | خدا حافظی سردبیر | ۲۴ | اطلاعی |
| ۲۴ | جدول اعداد سه طرفه، شش ضلعی جادویی دستبرد رنست (گردآوری دکتر بهروز پرها می) | ۲۱ | سرگرمی |
| ۲۱ | پاسخ به سوالات علمی و فنی خوانندگان - آشنائی با هوش مصنوعی | ۷ | نامه ها |
| ۷ | ریزپردا زنده ای که می شود (ترجمه حسن کا تویان) | ۲۴ | نوآوری |
| ۲۴ | کا میپوترها همه چیز را درباره عدم قطعیت فرا می گیرند (ترجمه امیر منصور فردیهیانی) | ۳ | تکته |
| ۳ | مقدمه ای بر زبان منطقی پرولوگ و گاربردهای آن (ترجمه دکتر محسن رستمی) | ۸ | مقاله ۱ |
| ۸ | روشهای اشکال زدائی بر نامه ها (ترجمه ابراهیم نقیب زاده مشایخ) | ۱۴ | مقاله ۲ |
| ۱۴ | مقدمه ای بر نسل پنجم (ترجمه دکتر محمود نقیب زاده) | ۱۶ | مقاله ۳ |
| ۱۶ | مفاهیم اولیه سیستمهای عامل - قسمت آخر (ترجمه عباس مصطفی سبزواری) | ۲۵ | مقاله ۴ |
| ۲۵ | آشنائی با پیشتازان علم کامپیوتر - قسمت هجدهم | | مقاله ۵ |
| | هربرت الکساندر سیمون، بنیانگذار رشته هوش مصنوعی (ترجمه دکتر پروینا صر) | | |



خدا حافظی سردبیر

از فروردین ماه سال گذشته تا کنون کمیته سرپرستی کمیته انتشارات انجمن به عهده اینجانب محول گردیده بود. ۲۰ شماره از ماه هنام گزاریش کا مپیوتر بطور منظم منتشر شده است. علاوه بر این، از دیگر فعالیتهای که ظرف این مدت در کمیته انتشارات صورت گرفته اند، میتوان از برپائی جایزه خوانشی برای بهترین مقالات چاپ شده در ماههای گزاریش کا مپیوتر هر سال، در دو گروه دانشجوی و آزاد (برندگان اولین دوره، جوایز خود را در مهرماه سال جاری همزمان با برگزاری مجمع عمومی سالانه انجمن دریافت داشتند)، افزایش تعداد صفحات ماهنامه از ۱۶ به ۲۴ (از آبان ماه ۱۳۶۳) و چاپ دوماهنامه و بیژنه ۳۲ صفحه ای به مناسبت آغاز هفتمین و هشتمین سال فعالیت انجمن، ایجاد بانک مقالات علمی (جهت تحویل مقاله برای ترجمه به افرادی که مایل به همکاری با کمیته انتشارات بودند) و... نام برد. مجموعه این اقدامات و فعالیتهای در جلب همکاری اعضا انجمن و دیگر خوانندگان گرامی گزاریش کا مپیوتر با کمیته انتشارات سهم به سزائی داشته اند و خوشبختانه در خلال این مدت، نه تنها هیچگونه نگرانی از مواج شدن با کمبود مقالات و مطالب برای ماهنامه وجود نداشته بلکه هم اکنون، مقالات واصله به کمیته انتشارات، جهت چاپ بایدر

یک صفا انتظار ۲ تا ۳ ماهه سپریبرد و برای اولین بار امکان انتخاب از بین مقالات و در نتیجه، جهت دادن به موضوعات هر شماره از ماهنامه وجود آمده است. در صورت ادامه این همکاری و تقویت بنیه مالی انجمن، زمینه برای افزایش دیگری بر تعداد صفحات ماهنامه، کاملاً مهیا می باشد. همچنین، خوشبختانه در طول این مدت، فعالیتهای انجام گرفته در کمیته انتشارات مورد تأیید اعضای گرامی انجمن نیز قرار گرفته است و طی نظرخواهی به عمل آمده در مهرماه سال جاری، نزدیک به ۹۵٪ از اعضا از نحوه فعالیتهای این کمیته، پشتیبانی نموده اند.

بهر حال، اینک که پایان مسئولیت من در کمیته انتشارات انجمن فرا رسیده است، ضمن آرزوی موفقیت برای مسئول جدید این کمیته، وظیفه خود می دانم که از کلیه اعضای محترم انجمن و دیگر خوانندگان گرامی گزاریش کا مپیوتر که در خلال دو سال گذشته با پشتیبانی و تشویقهای خود، مرا مورد لطف و محبت قرار داده اند، صمیمانه قدردانی و سپاسگزاری کنم و از بابت کاستیها و کمبودها پوز بطلبم. همچنین، از کلیه کسانی که در طول این مدت با کمیته انتشارات همکاری نموده و مرا در انجام وظایف محوله یاری رساندند، و به ویژه خانم لیلی حاجی اسماعیلی و آقای حمیدرضا رهبر که همواره چون دویای پرتوان در خدمت کمیته انتشارات بودند، صمیمانه تشکر و قدردانی می نمایم.

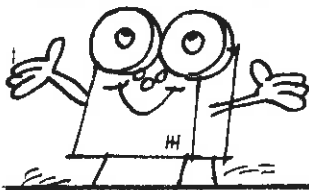
ابراهیم نقیب زاده، مشایخ



دنباله اخبار انجمن

مسئولین و سرپرستان کمیته های مختلف انجمن به شرح زیر انتخاب گردیدند:

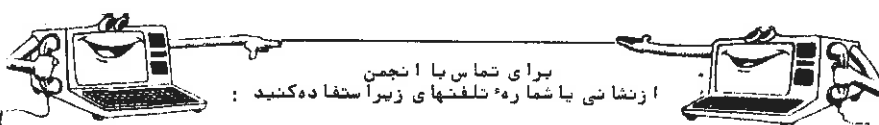
- آقای ابراهیم نقیب زاده، مشایخ: رئیس انجمن و سرپرست کمیته روابط عمومی
- آقای محمد میرزا عبداللّهی: نایب رئیس انجمن و سرپرست کمیته فعالیتهای علمی و فنی
- خانم لیلی حاجی اسماعیلی: دبیر انجمن
- آقای دکتر محسن رستمی: خزانه دار انجمن و سرپرست کمیته آموزش
- آقای مهدی فلاحتی: سرپرست کمیته عضویت
- آقای حمیدرضا رهبر: سرپرست کمیته انتشارات



لیست مقالات واصله

مقالات زیر تا پایان آذرماه ۱۳۶۴ به کمیته انتشارات واصل گشته اند:

- ۱- تجزیه مستقل از متن عبارتهای محاسباتی (ترجمه بهروزبنائی کاشانی)
- ۲- مخرج تحقیق و توسعه در سال ۱۹۸۴ (ترجمه لیلی حاجی اسماعیلی)
- ۳- بانک داده ها و زبانهای دستکاری داده ها (نوشته بهزاد منتظری)
- ۴- گامی در جهت افزایش توان استنتاج در کامپیوترها (تعریف ریاضی تشکیک) (نوشته سیدحسین کمالی)



برای تماس با انجمن از نشانی یا شماره تلفنهای زیر استفاده کنید:
نشانی: انجمن انفورماتیک ایران، صندوق پستی ۱۱۹۶-۱۴۱۵۵ شهران

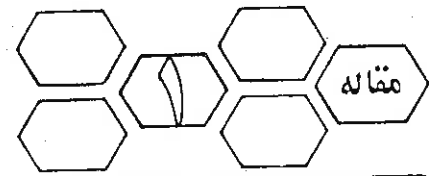
شماره تلفنهای مسئولان انجمن

رئیس	آقای ابراهیم نقیب زاده، مشایخ	 داخلی ۲۴۷ - ۲۲۹۰۷۱
نایب رئیس	آقای محمد میرزا عبداللّهی	 ۸۹۱۳۵۱
دبیر	خانم لیلی حاجی اسماعیلی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
خزانه دار	آقای دکتر محسن رستمی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
شماره تلفنهای سرپرستان کمیته ها		
انتشارات	آقای حمیدرضا رهبر	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
آموزش	آقای دکتر محسن رستمی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
فعالیتهای علمی و فنی	آقای محمد میرزا عبداللّهی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
عضویت	آقای مهدی فلاحتی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
روابط عمومی	آقای ابراهیم نقیب زاده، مشایخ	 داخلی ۲۴۷ - ۲۲۹۰۷۱
دیگر اعضای هیأت اجرایی		
عضو اصلی	آقای عبدالله کسرائی	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
عضو علی البدل	آقای مهرداد مدبرنیا	 ۸۹۱۴۲۹ و ۸۹۱۲۹۳
عضو علی البدل	آقای داریوش جوان تبریزی	 ۲۲۹۳۰۴



نشریات و کتابهای واصله به انجمن

- نشریات و کتابهای زیر تا پایان آذرماه ۱۳۶۴ به کتابخانه انجمن واصل گشته اند:
- ۱- راهنمای سمینارها، کنگره ها و سمپوزیومها برگزار شده در ایران (۱۳۶۲ - ۱۳۵۸)
 - مرکز اسناد و مدارک علمی - بهار ۱۳۶۴ (تهیه و تدوین: لیلا مرتضائی - محمود معتقد)
 - ۲- راهنمای نوشتن رساله های پژوهشی (وکتا بنیانی - توفیق روش تحقیق)
 - مرکز اسناد و مدارک علمی - ۱۳۶۴
 - نوشته: توراچی ورتن - ترجمه: محمدنقی مهدوی
 - ۳- تازه های کتاب (دوره ۴ شماره ۲)
 - مرکز اسناد و مدارک علمی - تیر ۱۳۶۴ (گردآوری و تنظیم: علی قبادی)



مقدمه‌ای بر زبان منطقی پرولوگ و کاربردهای آن

ترجمه آزاد*؛ دکتر محسن رستمی
(شرکت ایران ارقام)

بخش اول: مقدمه

در سال‌های اخیر با این که سخت‌افزار کامپیوتر و وسایل جانبی آن، شاهد تغییرات ریشه‌ای بسیاری بوده‌اند، ولی چنین تغییراتی در زمینه زبان‌های برنامه‌نویسی متداول، به وقوع نپیوسته‌بود. لزوم انجام چنین تغییرات ریشه‌ای، با در نظر گرفتن پیچیده و پرهزینه بودن برنامه‌نویسی و توسعه سیستم‌های نرم افزاری موجود، هر چه بیشتر ملموس و قابل توجه می‌شد. چنین تحول اساسی در زمینه زبان‌های برنامه‌نویسی با طرح و تولید زبان جدیدی به نام پرولوگ در اواسط دهه ۱۹۷۰ به وجود آمد. پرولوگ محصول جنبی تحقیقات دهه ۱۹۶۰ روی زمینه‌های هوش مصنوعی و اثبات اتوماتیک نظریه‌ها^۱ می‌باشد. دانشمندانی که در این سال‌ها روی زمینه هوش مصنوعی کار می‌کردند در ابتدا سعی داشتند که هوش را به‌طریقی به کامپیوتر القاء کنند. در این جهت، نخست طرق بیان و توصیف نظریه‌ها با استفاده از منطق صوری^۲ و اثبات اتوماتیک آنها مورد بررسی قرار گرفت. ایده کلی در این تحقیقات این بود که چون تمام تفکرات منطقی^۳ را می‌توان با استفاده از منطق صوری توصیف کرد، لذا کامپیوتری که بتواند این منطق را فهمیده و تجزیه و تحلیل کند بطور کلی دارای هوش خواهد بود.

در اواسط دهه ۱۹۶۰، الگوریتمی به نام "تفکیک کننده" را بینسون^۴ ارائه گردید. این الگوریتم نشان می‌داد که چگونه می‌توان نظریه‌های توصیف شده به وسیله منطق صوری را بطور اتوماتیک اثبات نمود. تا اواخر دهه ۱۹۷۰ این الگوریتم با بهره‌وری نسبتاً خوب روی بعضی از کامپیوترهای موجود آن روز پیاده و اجرا گردید.

در اواسط دهه ۱۹۷۰، پرفسور کوالسکی^۵ در آمریکا کالج دانشگاه لندن به این نتیجه رسید که جملات توصیف شده توسط منطق صوری را می‌توان به صورت جملات یک زبان برنامه‌نویسی در نظر گرفت. با چنین دیدی، اثبات یک نظریه که با این جملات توصیف می‌شود به منزله اجرای برنامه‌ای با همان جملات خواهد بود. نتیجه گیری مهم کوالسکی، منتج به خلق زبان پرو-لوگ توسط خود او گردید. در این زبان از منطق به عنوان یک زبان برنامه‌نویسی جدید و از تفکیک کننده را بینسون به عنوان مفسر^۶ این زبان استفاده می‌شود.

برای روشن شدن مفاهیم ذکر شده در این بخش، مثال زیر را در نظر بگیرید:

- ۱- انسان فانی است
- ۲- علی یک انسان است
- ۳- علی فانی است

در این مثال، جملات ۱ و ۲ مفروضات یک مسئله و جمله سوم، خود مسئله (یا نظریه) می‌باشد که می‌خواهیم حل کنیم. جملات ۱ و ۲ و همگی توسط جملات حساب گزاره‌ها^۸ قابل توصیف می‌باشند. الگوریتم تفکیک کننده را بینسون را در مورد این مسئله می‌توان بکار گرفت. این الگوریتم با استفاده از مفروضات بند ۱ و ۲، سعی در اثبات نظریه بند ۳ خواهد کرد. در مورد این مثال، الگوریتم حتماً به این نتیجه خواهد رسید که این نظریه درست بوده یعنی "علی فانی است".

برنامه‌نویسی با استفاده از منطق بطور معمول زبان پرولوگ به خصوص، نقش عمده‌ای در پیشبرد علم کامپیوتر در دهه ۱۹۸۰ و بطوریکه در دهه ۱۹۹۰ خواهد داشت. هم‌اکنون، پرولوگ بر روی بسیاری از ریز کامپیوترها با پردازنده‌های Intel 8088 و Z80 و همچون PDP-11 و مدل‌های گوناگون VAX پیاده شده است. سازندگان کامپیوترهای بزرگ، مثل IBM، ICL، و DEC بطور جدی بکارگیری این زبان را روی کامپیوترهای خود در این دهه و دهه بعد در نظر گرفته‌اند.

در بقیه بخش‌های این مقاله، ضمن آشنائی با مفاهیم اولیه و اساسی پرولوگ، بطور اجمالی آن را با زبان شناخته شده لیسپ^۹ مقایسه خواهیم نمود.

بخش دوم: چرا زبان پرولوگ؟

برای جواب به این سؤال، ابتدا زبان‌های معمول برنامه‌نویسی را مختصراً مرور می‌کنیم و مشکلات عمده آنها را در نظر می‌گیریم. سپس راه حل ارائه شده توسط پرولوگ را مورد بررسی قرار می‌دهیم.

برخلاف این که ممکن است این طور تصور شود که پرولوگ، زبان دیگری است که و بیش شبیه به سایر زبان‌های برنامه‌نویسی با قواعد نحوی^{۱۰} متفاوت، ولی پرولوگ زبانی است که با تمام زبان‌های برنامه‌نویسی معمول و شناخته شده کنونی، تفاوت‌های ریشه‌ای و اساسی دارد. یک زبان برنامه‌نویسی مثل فرترن، امکان‌ناهی را در اختیار برنامه‌نویس قرار می‌دهد که او با استفاده از آنها می‌تواند مسئله‌ای را به صورت مجموعه‌ای از دستورات پشت سر هم درآورده و سپس آنها را جهت تبدیل به کد ماشین^{۱۱} به کامپایلر زبان فرترن ارائه دهد. اجرای این دستورات توسط کامپیوتر، منجر به حل مسئله برنامه‌نویس خواهد شد. در این روش برنامه‌نویسی، مفروضات اولیه مسئله توسط دستورات زبان برنامه‌نویس داخل ساختارهای داده‌ها قرار داده می‌شوند. برنامه‌سپس با استفاده از ساختارهای دیگری به عنوان فضای کاری^{۱۲} جواب مسئله

مربوطه را پیدا کرده و آن را روی دستگاه خروجی مورد نظر می‌فرستد. این طریق برنامه‌نویسی دارای دو اشکال عمده و اساسی می‌باشد:

۱- بصیرت و آگاهی برنامه‌نویس با معلوماتی که او در مورد یک مسئله دارد، در لایه دستورات برنامه مورد استفاده، گم می‌شود. و در واقع، کامپیوتری که چنین برنامه‌ای را اجرا می‌کند، به جز دستوراتی که باید برای حل مسئله اجرا کند، چیز دیگری نمی‌داند.

۲- اگر قرار باشد برنامه‌ای مثلاً به زبان کوپول توسط برنامه‌نویسی غیر از برنامه‌نویسی که آن برنامه را نوشته توسعه پیدا کند، این امر بستگی کامل به مهارت او در خواندن و فهمیدن کدهای قدیم و اضافه کردن کدهای جدید خواهد داشت.

برخلاف یک برنامه مثلاً به زبان کوپول، برنامه پرولوگ مجموعه‌ای است از واقعیت‌ها و با مفروضات اولیه یک مسئله و قواعد استنتاجی مربوط به آن. بنا بر این، یک خط برنامه پرو-لوگ، شامل توصیف مفروضات و قوانین یک مسئله خواهد بود، نه دستوری که کامپیوتر باید آن را اجرا کند مثل جمع (ADD) یا تقسیم (DIVIDE). مفروضات و قواعد استنتاجی تعریف شده برای یک مسئله توسط مفسر زبان پرولوگ برای یافتن پاسخ به سؤال‌های مطرح شده در قلمرو آن مفروضات و قواعد مورد استفاده قرار می‌گیرند. با استفاده از پرولوگ، برنامه‌نویس فقط مشخص می‌کند که مسئله را می‌خواهد حل کند و چگونه حل مسئله به عهده خود پرولوگ می‌باشد. برنامه‌نویس در برنامه‌نویسی به زبان‌های متداول، برنامه‌نویس علاوه بر مشخص کردن مسئله، چگونه حل آن را هم باید از طریق کد کردن برنامه خاص مشخص نماید. بنا بر این، با استفاده از پرولوگ:

۱- مفروضات و بصیرت برنامه‌نویس در لایه کدهای برنامه همگ نمی‌شود زیرا هر خط برنامه پرولوگ به وضوح قواعد و واقعیت‌های مربوط به قلمروی یک مسئله را توصیف می‌کند و

۲- توسعه بعدی برنامه پرولوگ به آسانی امکان پذیر است زیرا معنای جملات این برنامه به آسانی برای هر کس قابل فهم می‌باشد.

مثال ۱

فرض کنیم مسئله مورد نظر ما پیدا کردن فاکتوریل عدد N باشد. با فرض این که فاکتوریل این عدد در یک کلمه حافظه کامپیوتر جا بگیرد، حل این مسئله با استفاده از زبان‌های فرترن و پرولوگ به طریق زیر خواهد بود.

برنامه فرترن:

```
INTEGER FUNCTION FACT(N)
FACT=1
IF(N.EQ. 0) THEN RETURN
DO 10 I=1,N
FACT=FACT*I
10 CONTINUE
RETURN
END
```

ورق بزنید

برنامه پرولوگ:

fact(0,1).
fact(N,Y):-fact(N-1,Z),multiply(Z,N,Y)

تفسیر برنامه پرولوگ بالا به شرح زیر است:

- ۱- فاکتوریل عدد صفر برابر یک بوده و
- ۲- فاکتوریل N عددی است مثل Y بطوری که فاکتوریل عدد N-1 برابر عدد Z بوده و حاصل ضرب اعداد Z و N برابر Y می باشد.

بخش سوم: اساس اولیه و کلی زبان پرولوگ

بطور کلی یک برنامه پرولوگ، توصیف کننده یک پایگاه بصیرت^{۱۴} می باشد. یک بصیرت حاکی از یک آگاهی از دنیای واقعی بوده و ممکن است از دو قسمت زیر تشکیل شود:

- ۱- یک واقعیت
- ۲- یک قاعده استنتاج

واقعیت، توصیف کننده یک نهاد و یا توصیف کننده یک رابطه بین دو یا چند نهاد می باشد. قواعد استنتاج، مکانیزمی هستند که به وسیله آنها می توان نهادها و روابط دیگر را با استفاده از نهادها و روابط تعیین شده، استنتاج نمود. مثلاً در برنامه پرولوگ مثال ۱ (بخش دوم)، عبارت اول بیانگر یک واقعیت و عبارت دوم یک قاعده استنتاج می باشد. یک عبارت^{۱۵} در برنامه پرولوگ، از یک یا چند لفظ^{۱۶} تشکیل می گردد. یک لفظ، توصیف کننده یک نهاد یا یک رابطه بوده و از یک گزاره^{۱۷} و یک یا چند نهاد تشکیل می شود. گزاره یک لفظ، صفت یک نهاد یا اسم رابطه ای است بین دو یا چند نهاد. یک نهاد به سه صورت ممکن است بیان شود:

- ۱- یک ثابت
 - ۲- یک متغیر
 - ۳- یک ساختار (مجموعه یک یا چند لفظ)
- نهاد متغیر در پرولوگ با یک حرف بزرگ^{۱۸} مشخص می گردد.

مثال ۲

در برنامه پرولوگ مثال ۲ که در همین صفحه آورده شده است:

- ۱- شش عبارت اول، توصیف کننده واقعیت و عبارت هفتم، بیانگر یک قاعده استنتاجی می باشد.

- ۲- شش عبارت اول، "تک لفظی" و عبارت هفتم "سه لفظی" می باشد.

- ۳- در این برنامه، کلمات colour، door، has، door-colour، گزاره بوده و کلمات pass-door، escape-door، red، floor-1، yellow، blue، door-1 و door-2 ثابت می باشند.

- ۴- نهاد اول عبارت ششم زنوع ساده و نهاد دوم آن از نوع ساختاری می باشد.

- ۵- در عبارت هفتم، کلمات D و C و T متغیرهای عبارت بوده و علامت (-) به معنای "اگر" می باشد.

عامل و یا رویدادی^{۱۹} که باعث اجرای یک برنامه پرولوگ می گردد، ورودی سوال به مفسر این زبان می باشد. مفسر پرولوگ پس از استفاده از واقعیتها و قواعد استنتاجی تعریف شده برای یک مسأله، نخست یک پایگاه بصیرت برای آن ساخته و سپس با کمک یک فرا روند تفکیک کننده^{۲۰}، سوال وارده را به اجزاء ساده تر تفکیک کرده و با جستجو در پایگاه بصیرت سعی در یافتن جواب برای این اجزاء خواهد کرد. هنگامی که جواب تمام اجزاء تفکیک شده پیدا شود، جواب سوال وارده به مفسر پیدا شده است و مفسر آن را ارائه می دهد. برای یک سوال وارده، امکان بیش از یک جواب وجود خواهد داشت. در این صورت، استفاده کننده با وارد نمودن علامت (؛) می تواند زمفسر پرولوگ بخواهد تا جوابهای متعدد را یکی یکی ارائه دهد.

جواب یک سوال که توسط مفسر پرولوگ ارائه می شود، ممکن است به یکی از صورتهای زیر باشد:

- ۱- بلی: تفسیر این جواب این است که اجزای هدفی سوال همگی در پایگاه بصیرت پیدا شده و امکان اثبات آنها وجود دارد.
 - ۲- خیر: این جواب به معنی آن است که برای مفسر پرولوگ، با توجه به معلومات ذخیره شده در پایگاه بصیرت، امکان یافتن جواب موجود نمی باشد.
 - ۳- بلی اگر: این جواب حاکی از وجود پاسخهای متعددی برای سوال وارده است، در این صورت، سوال کننده می تواند اندیک پاسخ را نپذیرفته و رد کند. در این موقع مفسر پرولوگ جواب دیگری را پیدا کرده و ارائه خواهد داد. وقتی جواب دیگری موجود نباشد، مفسر این موضوع را با کلمه "خیر" اعلام خواهد کرد.
- در زبان پرولوگ، یک سوال حاوی یک یا چند لفظ بوده و با علامت (-) در ابتدای عبارت مشخص می گردد.

مثال ۳

فرض کنید پایگاه بصیرت یک مسأله حاوی

واقعیتها و قواعد ذکر شده در مثال ۲ باشد. در این صورت با استفاده از مفسر پرولوگ می توان سوالهای زیر را پرسید. جوابهای پیدا شده پس از ورود هر سوال، نوشته می شوند (مثال ۳ در صفحه روبرو آورده شده است).

مثالهای ساده بالا البته بیاهمیت و نگرتمنا م قدرتها و امکانات موجود در این زبان نبوده و فقط به منظور آشنائی ابتدائی با این زبان مطرح شده اند. در یک محیط پرولوگ، فاصل استفاده کننده^{۲۱} می تواند نزدیک زبان طبیعی مثل زبان انگلیسی باشد. در این صورت، باز مفسر پرولوگ سوال وارده را تفسیر کرده و مثل قبل، جستجوی خود را برای یافتن پاسخ در پایگاه بصیرت ادامه خواهد داد.

بخش چهارم: زمینه های کاربردی پرولوگ

۴- الف: پروژه کامپیوترهای نسل پنجم

هیچ مقالهای درباره پرولوگ بدون اشاره به پروژه ساخت کامپیوترهای نسل پنجم و نقش این زبان در این پروژه، کامل نخواهد بود. اما کامپیوترهای نسل پنجم چه خصوصیات دارند و چرا به آنها احتیاج می باشد؟ جواب این سوال را باید در محدودیتهای موجود در کامپیوترهای فون نویماتی^{۲۲} یا کامپیوترهای نسل چهارم جستجو کرد. این کامپیوترها اگرچه ماشینهای محاسب^{۲۳} قدرتمند و سریعی می باشند ولی در بسیاری از زمینه های کاربردی مثل ساخت فاصلهای هوشمند بین انسان و ماشین و تولید نرم افزار در حجم زیاد^{۲۴} محدودیتهای خود را نشان می دهند.

شروع پروژه کامپیوترهای نسل پنجم که اولین بار توسط ژاپنیها اعلام گردید به منظور رفع این محدودیتها بود. هدف این پروژه ابتدا تعیین مشخصات سخت افزارها و نرم افزارهای لازم برای این کامپیوترها و سپس ساخت آنها می باشد. این پروژه، پنج زمینه تحقیقاتی زیر را برای ادامه کار تعیین کرده است:

- ۱- تعیین زبانهای جهت توصیف هوش مصنوعی
- ۲- شکل بخشیدن به داده ها و یا تجربیات داده ها و برنامه نویسی موضوع گرا (در مقابل روش گرا^{۲۵})



عبارات	توضیح
1: colour(escape-door,red).	۱- رنگ در خروج اضطراری قرمز است.
2: colour(pass-door,blue).	۲ و ۳- رنگ در خروج عادی آبی و یا زرد است.
3: colour(pass-door,yellow).	
4: door(door-1,escape-door).	۴- در شماره یک، در خروج اضطراری است.
5: door(door-2,pass-door).	۵- در شماره دو، در خروج عادی است.
6: has(floor-1,room(door-1,door-2)).	۶- در شماره یک و دوم متعلق به اتاق طبقه اول می باشد.
7: door-colour(D,C):-door(D,T),colour(T,C)	۷- رنگ در D زنوع رنگ C می باشد اگر ... در D از نوع T بوده و ... رنگ T از نوع C باشد.

مثال (۲)

گرا مثل کویول نوشته می‌شوند، ساخت.

۴ - پ: پرولوگ و پایگاه داده‌ها ۲۴

در زمینه پایگاه داده‌ها، علاقه به پرولوگ بیشتر به خاطر قدرتهای است که این زبان برای ایجاد داده‌های هوشمند دارد. البته برای این منظور در نظر است پرولوگ به قدرتهای یک سیستم مدیریت پایگاه داده‌ها نیز مجهز شود. مثلاً در آینده ترتیبی از داده‌ها بتواند بتواند ساختار داده‌ها را بطور مستقیم بر روی حافظه جانبی مثل دیسک قرار داده و از طریق مفسر پرولوگ به آنها دسترسی ۳۵ داشت. همچنین با امکانات بیشتری که برای این زبان در نظر گرفته شده می‌توان:

۱ - رکورد های یک پایگاه داده را تعریف و مشخص کرد،

۲ - محدودیت های لازم برای جلوگیری از مخدوش ساختن شما میت مفاهیم داده‌ها ۳۶ را تعریف نمود (مثلاً یک نوع از این محدودیتها می‌تواند از ورود سن منفی برای شخصی به پایگاه داده جلوگیری کند.)،

۳ - سوال های مورد نظری پایگاه داده‌ها را مشخص و تعریف کرد.

با تعبیه این امکانات جدید در زبان پرولوگ ایجاد سیستمهای مدیریت پایگاه داده‌ها به مراتب سریعتر از زمان حاضر امکان پذیر خواهد بود.

۴ - ت: پرولوگ و مترجمها ۳۷

پرولوگ بهترین وسیله برای تولید

ورق بزنید

از آن جدا شده و سیستم حاصل برای فروش به بازار عرضه می‌گردد. در یک سیستم خبره غیر-بسته‌ای، قسمت سوم سیستم روی آن با قسسی مانده و استفاده کننده مکان اجرای هرگونه تغییری را بر روی پایگاه بصیرت خواهد داشت. قسمت استنتاج کننده یک سیستم خبره، سوالهای فرستاده شده از طرف استفاده کننده را تفسیر کرده و با استفاده از پایگاه بصیرت سیستم، پاسخهای موجود را پیدا کرده و ارائه خواهد داد. یک سیستم خبره معمولاً برای یک محیط و کاربر خاص ساخته و نگهداری می‌شود. مثلاً سیستم خبره‌ای که برای عیب‌یابی دستگاه هوای محفظه فضا نوردان در یک سفینه فضاپیما نصب می‌شود، به مسئول تعمیرات سفینه این امکان را خواهد داد که در صورت بروز هر عیبی در این دستگاه، قسمت یا قسمت های معیوب را شناسایی کرده و نسبت به تعمیر یا تعویض آنها اقدام نماید. سیستمهای خبره حاضر و قابل استفاده امروزی در زمینه‌های مختلفی مانند نظامی، مالی، ارتباطات، الکترونیک، استخراج نفت، و معادن بکار گرفته شده‌اند. در مدیالائی از این سیستمها در بخش الکترونیک و برای عیب‌یابی قطعات الکترونیکی مورد استفاده قرار گرفته‌اند.

قواعد نحوی زبان پرولوگ و قدرتهای آن که در این مقاله به مختصری از آنها اشاره گردید، منجر به انتخاب آن توسط اکثریت سازندگان سیستمهای خبره شده است. با استفاده از این زبان می‌توان یک سیستم خبره را ۳ الی ۴ برابر سریعتر از سیستمهای مشابه که با یک زبان روش-

۳ - سیستمهای خبره ۲۷

۴ - ساخت کامپیوترهای با ساختار پردازش موازی مثل کامپیوترهای که بر اساس جریان داده‌ها ۲۸ کار می‌کنند.

۵ - ادامه تحقیقات در زمینه تکنولوژی مجتمع سازی مدارات در مقیاس بزرگ (VLSI) در زمینه نرم افزار، این پروژه سه هدف عمده زیر را دنبال می‌کند:

۱ - طراحی و ایجاد سیستمهای خبره

۲ - طراحی و ایجاد سیستمهای حلال مسائل و سیستمهای استنتاجی ۲۹

۳ - طراحی و ایجاد داده‌های هوشمند جهت ارتباط بین انسان و ماشین ۳۰

هر کدام از سیستمهای فوق، در سه سطح مختلف بررسی می‌شوند:

۱ - سخت افزار مناسب مورد نیاز

۲ - نرم افزارهای بنیادی

۳ - فاصله‌های خارجی نرم افزارهای بنیادی در سطح سخت افزار، بجای استفاده از پردازشهای ترتیبی از پردازشهای موازی و در زمینه نرم-افزار بجای استفاده از زبانهای روش گرا از زبانهای برنامه نویسی بر اساس منطق استفاده خواهد شد.

جدول شماره ۱، اجزای عمده نرم افزارها و سخت افزارهای کامپیوترهای نسل پنجم را نشان می‌دهد. در پروژه ساخت کامپیوترهای نسل بعد، زبان پرولوگ، هم نقش عمده‌ای در ساخت سیستمهای حلال مسائل و استنتاجی داشته و هم برای نوشتن بسیاری از نرم افزارهای سیستمهای خبره و فاصله‌های هوشمند بکار گرفته خواهد شد.

۴ - پ: پرولوگ و سیستمهای خبره

طراحی و ایجاد سیستمهای خبره، یکی از مهمترین هدفهای پروژه کامپیوترهای نسل پنجم می‌باشد. هم اکنون فعالیتهای زیادی در بخشهای صنعتی و دانشگاهی کشورهای پیشرفته برای ایجاد این سیستمها در جریان می‌باشد. بطور کلی یک سیستم خبره از سه بخش عمده زیر تشکیل شده است:

۱ - قسمت استنتاج کننده ۳۱

۲ - پایگاه بصیرت

۳ - قسمت توسعه دهنده و اصلاح کننده ۳۲

این بخشها و رابطه آنها در شکل ۱ نشان داده شده‌اند. سیستمهای خبره به دو دسته مختلف تقسیم می‌گردند:

۱- بسته‌ای ۳۳ که به آنها SHELL نیز می‌گویند

۲- غیر بسته‌ای

در یک سیستم خبره از نوع بسته‌ای، ابتدا پایگاه بصیرت توسط مهندس این پایگاه با استفاده از ویراستار مخصوص ساخته شده و درستی واقعیتهای قوانین استنتاجی آن و عملکرد آن در یافتن جوابهای صحیح برای سوالهای مشخص از نظر متان می‌شوند. پس از اطمینان از صحت عملکرد سیستم، قسمت سوم آن یعنی ویراستار مخصوص ساخت و اصلاح پایگاه بصیرت

سوال	توضیح
1- colour (pass-door, blue)? yes	1- سوال: آیا رنگ در خروج عادی آبی است؟ جواب: بلی
2- colour (escape-door, green)? no	2- سوال: آیا رنگ در خروج اضطراری سبز است؟ جواب: خیر
3- colour (pass-door, X)? X=blue; X= yellow; no	3- سوال: در خروج عادی چه رنگی است؟ جواب: آبی زرد جواب دیگری موجود نیست
4- door-colour (door-1, C)? C=red; no	4- سوال: در شماره یک چه رنگی است؟ جواب: قرمز جواب دیگری موجود نیست
5- has (floor-1, room(X, Y)), door-colour (X, C1), door-colour (Y, C2) X=door-1, Y=door-2, C1=red, C2=blue; X=door-1, Y=door-2, C1=red, C2=yellow; no	5- سوال: اطاق طبقه اول دارای چه درهایی است و رنگ آنها چیست؟ جواب: در شماره یک، در شماره دو، قرمز، آبی در شماره یک، در شماره دو، قرمز، زرد جواب دیگری موجود نیست

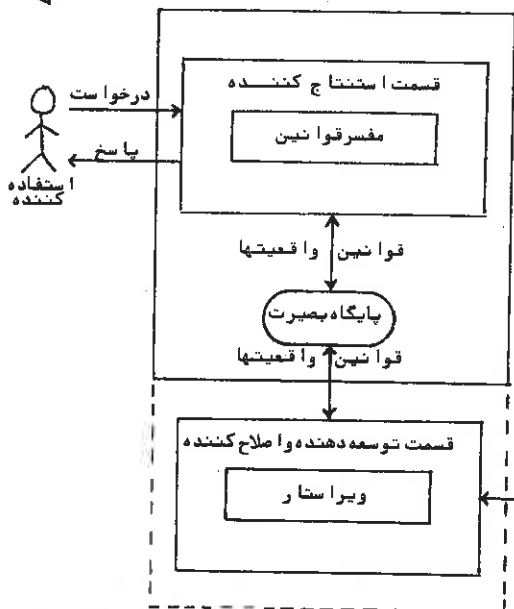
بیشتر قدرتها ی لیسپ را برای این گونه پردا - زنده ها دارا بوده و علاوه بر آن، فهم و خوانندگی جملات آن بسیار آسانتر از جملات ماشا به در زبان لیسپ می باشد، از این جهت، می توان را بطه پرولوگ با لیسپ را هما نندرا بطه یک زبان سطح بالا مثل پاسکال با زبان اسمبلی ماشیسن دانست.

بخش هفتم: نتیجه

مهمترین امتیاز پرولوگ به زبانهای متداول برنامه نویسی، رها ساختن برنامه نویس از نوشتن رویه ها ۴۲ و تست عملکرد آنها برای حل یک مساله می باشد. واقعیتها و قواعد استنتاجی که فضای کاری و پایگاه بصیرت لازم را برای یک مساله خاص تشکیل می دهند به مفسر پرولوگ داده شده و سپس مفسر با استفاده از این پایگاه، سعی در حل آن مساله خواهد کرد. مسائل معمولاً به صورت سوال به مفسر پرولوگ ارائه می گردند. برخلاف زبانهای برنامه نویسی متداول که احتیاج به تعریف کامل و جامع کلیه داده ها و رویه های لازم جهت حل یک مساله را دارند، پرولوگ می تواند با داشتن اطلاعات ناکامل راجع به قلمرو یک مساله نسبت به یافتن پاسخ برای سوالهای زارده، عمل نماید. این مهمترین امتیاز پرولوگ نسبت به زبانهای روش گرامی می باشد.

یادداشتها و منابع

- * An Introduction to Prolog, Peter S. G. Swinson, Computer Aided Design, Vol. 15, No. 4, July 1983
- * Logic Programming and Prolog, James Weiner, Computer and Electronics, January 1985.
- * Logic Programming in Prolog, Alex Goodall, Data Processing(dp), Vol. 26, No. 2, March 1984.



شکل (۱)

پیاپی شده است.

۳ - پرولوگ مجارستانی: این گونه یکی از قویترین گونه های پرولوگ از نظر داشتن تعداد زیادی گزاره داخلی ۴۱ می باشد. اسم تجاری این گونه، MPROLOG بوده و روی ماشینهای IBM و VAX تحت سیستم عامل VMS قابل اجرا می باشد.

۴ - میکرو پرولوگ: این یکی از گونه های محدود پرولوگ از نظر تعداد گزاره های داخلی بوده و روی پردا زنده های Z80 و تحت سیستم عامل CP/M قابل اجرا می باشد. در حال حاضر، فقط گونه های مختلف فوق، به با زار عرضه شده اند و لی هم اکنون، کار برای ساختن کامپیوتر این زبان که بتواند روی پردا زنده های ۳۲ بیتی اجرا شود، در جریان می باشد (تا سال ۱۹۸۲).

بخش ششم: مقایسه اجمالی پرولوگ و لیسپ

هما نظور که در بخش چهارم گفته شد، پرولوگ می تواند در زمینه توصیف و نشان دادن هوش مصنوعی مورد استفاده قرار گیرد. علاوه بر پرو-لوگ، زبان دیگری نیز به نام لیسپ برای این منظور استفاده می شود.

زبان لیسپ از حدود ۲۵ سال پیش مورد علاقه و استفاده محققین هوش مصنوعی بوده است. پس از پیدایش زبان پرولوگ و مخصوصاً پس از اعلام استفاده از این زبان در ساخت نرم افزارهای کامپیوترهای نسل پنجم توسط ژاپنیها، بسیاری از این محققین نسبت به برتری لیسپ در زمینه توصیف هوش مصنوعی به تردید افتادند. لیسپ اگرچه دارای قدرتها ی مناسبی برای نوشتن پردا زنده های زبانهای طبیعی مثل زبان انگلیسی می باشد ولیکن اشکال عمده آن پیچیدگی و سختی قوا عدنحوی آن است. پرولوگ

کامپیوترها و یا مترجمها می باشد. یک کامپیوتر یا مترجم، یک رشته از داده های ورودی را بر طبق قوا عدتبدیل ۳۸ معین، تبدیل به یک رشته از داده های خروجی می کند. در مورد کامپیوتر، رشته ورودی، جملات یک برنامه سطح بالا مثل فرترن و رشته خروجی، کد دستورات خاص یک ماشین می باشد. در مورد مترجم، رشته ورودی، جملات اسمبلی یک پردا زنده خاص مثل Z80 و رشته خروجی، معادل رشته ورودی بر حسب جملات اسمبلی پردا زنده دیگری مانند Intel 8086 می باشد. با استفاده از پرولوگ می توان قوا عدتبدیل لازم برای یک کامپیوتر یا یک مترجم را تعریف و توسط یک برنامه مفسر مولد ۳۹ به یک کامپیوتر یا مترجم اسمبلر به اسمبلر تبدیل نمود. با اتخاذ این روش، کامپیوتر یا مترجم تولید شده، بسیار انعطاف پذیر خواهد بود. هر تغییر در قوا عدتبدیل جدید به تبدیل منعکس شده و سپس قوا عدتبدیل جدید به برنامه پرولوگ مولد داده شده و گونه جدید کامپیوتر یا مترجم به وجود خواهد آمد.

بخش پنجم: گونه های مختلف زبان پرولوگ

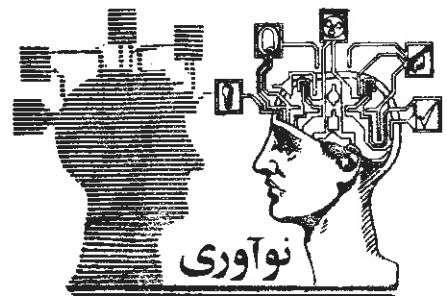
تاریخ شروع از دنیا د علاقه به این زبان و استفاده از آن در موارد مختلف، سال ۱۹۷۰ می باشد. در این سال، شخصی به نام دیوید وارن ۴۰، استاد دانشگاه دینبورگ انگلستان، یک نسخه از اولین گونه مفسر این زبان را از شهرماری فرانسه به دینبورگ برد. این مفسر روی کامپیوتر DEC10 آن دانشگاه پیاپی ده و سپس به بسیاری از محیطهای آکا دمیک در داخل و خارج انگلستان صادر گردید. امروزه، گونه های مختلفی از این زبان موجود می باشد که مهمترین آنها به قرار زیرند:

۱ - پرولوگ ادینبورگ: این گونه از پرولوگ بر روی کامپیوترهای DEC10 و DEC20 تحت سیستمهای عامل TOPS10 و TOPS20، بر روی کامپیوترهای PDP11 و VAX تحت سیستمهای عامل یونیکس و VMS و روی کامپیوتر ICL 2900 تحت سیستم عامل EMAS پیاپی ده شده است.

۲ - پرولوگ ماری: این گونه پرولوگ، قابل حمل بوده و بر روی ریز کامپیوترهای آپل

	Knowledge base management system	Problem solving and inference system	Intelligent interface system
External interface to basic software	High-level inquiry language	Core language (Prolog derivative)	Natural language speech, pictures etc.
Basic software	1) Knowledge representation language 2) Knowledge base support system 3) Knowledge acquisition system 4) External knowledge bases	1) Intelligent programming system 2) Meta-inference system 3) Distributed problem solving	1) Intelligent parser and meaning analyser 2) Continuous speech recognizer 3) Picture processing system
Hardware	1) Relational database machine 2) Parallel relational operations	1) Dataflow machine 2) Abstract data-type mechanism 3) Parallel inference mechanism	1) Image processing machine 2) Speech processing machine

جدول (۱)



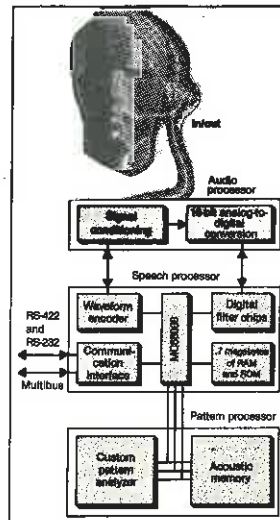
ریزپردازنده‌ای که می‌شنود!

ترجمه: حسن کاتوزیان

به تازگی یک سیستم شنوایی مصنوعی جهت اتصال به کامپیوترهای شخصی آی‌بی‌ام مدل‌های XT و AT طراحی گردیده است. در این دستگاه که به صورت واحدی مستقل به منظور درک مکالمات طراحی و ساخته شده، از ریزپردازنده "موتورولا ام - سی ۶۸۰۰۰" استفاده به عمل آمده است و از طریق سیستم‌های چندمسیره ویا با یکا رگیری استاندارد "آر - اس - ۲۳۲" می‌توان آن را به کامپیوتر متصل نمود. دقت دستگاه فوق به حدی است که برای کاربردهای متعددی می‌توان از آن به عنوان یک واحد ورودی داده‌های کامپیوتر استفاده به عمل آورد. همچنین می‌توان از آن در طراحی وساخت به کمک کامپیوتر ویا ضبط و نگهداری اطلاعات استفاده کرد. آزمایش بر روی سیستم فوق که دارای قابلیت پردازش ۹۰۰ لغت مختلف است تا ۹۷٪ موفقیت آمیز بوده است. البته میزان دقت تا بیرون تشخیص سیستم شنوایی مزبور در مورد لغاتی که شبیه هم تلفظ می‌گردند از این حد پایین‌تر است.

به منظور کار با سیستم لازم است تا ابتدا دستگاه را تحت آموزش قرار دهیم. بدین معنی

که ابتدا استفاده کننده از روی متنی می‌خواند و واحد پردازنده ۶۸۰۰۰ با توجه به تنوع آوازی که ادا می‌شود، یک مدل ریاضی از صدای گوینده درون دستگاه به وجود می‌آورد. این مدل، ریزپردازنده را قادر می‌سازد تا امواج صوتی رقمی شده، مزبور را فشرده سازد و آنها را در یک واحد حافظه ۲۹۰ کیلوبایتی با دستیابی مستقیم ذخیره نماید و سپس به مقایسه پردازد. وقتی سیستم در حالت درک آوازی و مکالمات



دریافتی قرائت می‌گیرد. آوازی را در زمان واقعی پردازش می‌کند. در ادامه این عمل، ریزپردازنده به گونه‌ای تکنواخت سیگنال‌ها یا علائم فوق را فشرده سازد و آنها را با ۳۰۰ الگوی از پیش تعیین شده رزداها و الفاظ مقایسه می‌نماید و در صورت تشابه یکی از آنها با کلمه ادا شده، مفهوم را به اصطلاح درک می‌کند. دستگاه یک فاصله ۲۵۰ میلی ثانیه‌ای بین

کلمات متوالی زمان می‌گیرد. بطوری که بین ادای یک جمله، اگر برای مدت زمان ۲۵۰ میلی ثانیه هیچ ورودی دیگری دریافت نشود، سیستم متوجه می‌گردد که آن جمله تمام شده و به سرعت آن را با اطلاعات حافظه مقایسه نموده و در صورت تشابه با یکی از الگوهای موجود، علامت تصدیق را می‌فرستد.

وقتی لغات بیان می‌گردند توسط دستگاه فهمیده می‌شوند. واحد ریزپردازنده به تصحیح و تغییرات احتمالی لازم در الگوهای حافظه اقدام می‌ورزد. در این کار تراشه‌های فیلترکننده و واحدهای برنامه پذیر کنترل آوازی که وظیفه دنبال کردن مسیر ادای کلمات را برعهده دارند، سیستم را یاری می‌کنند و در مرحله خاصی سیستم را قادر می‌سازد تا به تغییر و تصحیح مدل‌های غیر دقیق الفاظ و لغات بپردازد.

شرکت سازنده این دستگاه در ماچوست آمریکا میدوار است که به توان نظرفیت درک لغات سیستم را تا ۳۰۰۰ کلمه با لایبرد. بهای سیستم فوق همراه با فاضل لازم جهت ارتباط با کامپیوتر آی‌بی‌ام XT یا AT، ۶۵۰۰ دلار تخمین زده می‌شود.

به نقل از
IEEE Spectrum
Sept. 1985

اوسال مطالب برای گزارش کامپیوتر

برای همکاری با ما نامه گزارش کامپیوتر از طریق ارسال مطالب، لطفاً به نکات زیر توجه کنید:

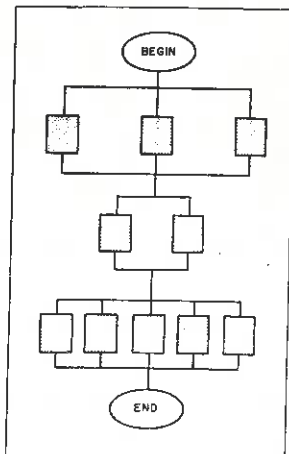
۱- ما را از اخبار و وقایع محل کار خود در زمینه انفورماتیک، حتی الامکان با همراه کردن اسناد و مدارک مربوطه، مطلع نمائید. مثالهایی از مطالب مناسب عبارتند از: انعقاد یا فسخ قراردادها، طراحی و پیاده کردن سیستمها، ابداعات و کاربردهای نوین، برگزاری سمینارها و سخنرانیها، ...

۲- اگر در روزنامه، مجله، کتاب یا نشریه‌ای دیگر به مطلب جالبی برخوردید، بریده یا فتوکپی تمیز و خوانای آن را با ذکر مشخصات کامل ماخذ (نام نشریه، تاریخ انتشار، ناشر، محل انتشار، شماره صفحه) برای ما بفرستید.

۳- مقالات خود را حتی الامکان به صورت ماشین شده در یک روز کاغذ (بسیار) ذکر مشخصات ماخذ و نام نویسنده و زمینه کردن نسخه‌ای از اصل مطلب در مورد مقالات ترجمه شده برای ما ارسال کنید. در صورتی که امکان ماشین کردن مقاله برای شما وجود ندارد، دست نوشت کاغذ "خوانا" با حواشی زیاد و رعایت حداقل ۲ سانتیمتر فاصله بین سطرها نیز پذیرفته می‌شود. ما نامه گزارش کامپیوتر در قبول یا رد مقالات رسیده، خلاصه کردن آنها، و هر نوع حکم و اصلاح دیگر آزاد است.

۴- مقالات و سایر مطالب رسیده مسترد نخواهند شد. نویسنده شخصاً مسئول نظرات ابراز شده در مقاله است و درج مقاله به معنی تأیید محتوای آن توسط انجمن انفورماتیک ایران نیست.

- 17: Predicate
- 18: Capital Letter
- 19: Event
- 20: Resolution Process
- 21: User Interface
- 22: Von Neuman
- 23: Arithmetic Machine
- 24: Software Productivity
- 25: Data Abstraction
- 26: Object-Oriented Versus Procedural-Oriented Programming
- 27: Expert Systems
- 28: Data-flow Computers
- 29: Problem Solving and Inference Systems
- 30: Intelligent Interface System
- 31: Inference Engine
- 32: Development Engine
- 33: Packaged
- 34: Database
- 35: Access and Manipulate
- 36: Integrity Constraint
- 37: Translators
- 38: Transformation Rules
- 39: Generating Program
- 40: David Warren
- 41: Built-in Predicate
- 42: Procedure
- * Programming in Prolog, W. F. Clocksin, C. S. Mellish, Springer-Verlag, 1981.
- * A Machine Oriented Logic Based on the Resolution Principle, J. A. Robinson, CACM, Vol. 12, January 1965.
- * Encyclopedia of Computer Science and Engineering, VNR, 1983 (page 852-Lisp and Llist processing languages).
- * Toward Automating the Software-Development Cycle, Karen A. Frenkel, CACM, Vol. 28, No. 6, June 1985
- 1: PROgramming in LOGic (PROLOG)
- 2: Automatic Theorem Proving
- 3: Formal Logic
- 4: Rational Thoughts
- 5: Robinson's Resolution Algorithm
- 6: R. A. Kowalski
- 7: Interpreter
- 8: Predicate Calculus
- 9: LList Processing Language (LISP)
- 10: Syntax Rules
- 11: Machine Codes
- 12: Data Structures
- 13: Work Space
- 14: Knowledge Base
- 15: Clause
- 16: Term



شکل ۱ - نمودار یک برنامه واحدی، جهت آزمایش کل برنامه به صورت یکپارچه باید ۳۰ (۳×۲×۵) مسیر مختلف بررسی گردند.

درواقع، یافتن خطا و تصحیح آن ممکن است دو عمل جداگانه باشند. به عنوان مثال، ممکن است شما از طریق بررسی محتوای یک متغیر در یک نقطه خاص از برنامه، متوجه شوید که آن متغیر باعث بروز خطا است. در این صورت تصحیح این خطا، خود مرحله جداگانه‌ای است. در روش علمی، شما با پذیرفته‌ای را بیابید که تمام واقعیات را دربرداشته باشد. به عنوان مثال، ممکن است فرضیه شما این باشد که "X باعث بروز خطا است" و در این صورت با تصحیح X فرضیه خود را می‌آزمایید. چنانچه خطا از بین رفت، فرضیه شما درست بوده است و شما توانسته‌اید منشأ خطا را یافته و آن را تصحیح نمایید. وگرنه (این نکته حائز اهمیت است)، اکنون یک اطلاع دیگر نیز به اطلاعات قبلی شما افزوده گردیده و آن این که "X باعث بروز خطا نبوده است". حال، فرضیه بعدی شما یا بدینا این اطلاع جدید نیز سازگار باشد.

فرضیات در اشکال زدائی

- یک خطا در برنامه ممکن است دارای چند منشأ مختلف باشد:
- ۱ - خطاهای نحوی
 - ۲ - خطاهای منطقی که باعث کارکرد اشتباه کامپیوتر می‌گردند (مانند پیش آمدن سرریز و امثال آن)
 - ۳ - خطاهای منطقی که باعث می‌شوند کامپیوتر با آن که به ظاهر کار خود را به پایان می‌رساند ولی جوابهای نادرست تولید نماید
 - ۴ - خطاهای تولید شده در خارج از برنامه مسه (به علت نرم افزار یا سخت افزار مورد استفاده).

بدان دلیل، هرگز نباید هیچیک از مخطو-ده‌های را که ممکن است دربرگیرنده منبع خطا

برنامه‌های خود را به صورت پیمانه‌ای یا واحدی طراحی کنید. برنامه‌ای که نامشودار اجرایی آن در شکل (۱) آمده است را در نظر بگیرید (هر بلوک، یک واحد یا پیمانه از برنامه است). چنانچه برنامه‌ها "به صورت یکپارچه" در نظر گرفته شود، جهت اشکال زدائی کامل آن، باید ۳۰ (۳×۲×۵) مسیر جداگانه، مورد ارزیابی قرار گیرند (این مسیرها، مسیرهایی هستند که برنامه می‌تواند از نقطه شروع تا نقطه خاتمه در پیش گیرد). حال آن که چنانچه هر واحد را جداگانه اشکال زدائی کنید، تنها نیاز به بررسی ۱۱ چیز خواهد داشت: هر یک از ۱۰ واحد بطور جداگانه، به اضافه برنامه‌ای که این ۱۰ واحد را به یکدیگر متصل می‌آورد.

هدف اصلی را "وضوح" و "خوانایی" برنامه قرار دهید و بهینه‌سازی آن را به بعد موکول کنید. معمولاً زمانی که صرف بهینه‌سازی برنامه می‌گردد، به زمان صرفه جویی شده نمی‌آورد. چنانچه شما "مجبورید برنامه را بهینه‌سازی کنید، ابتدا محو اصلی را کار کردن برنامه قرار دهید و سپس راه‌های بهینه‌سازی آن را آرزو-یابی نمایید.

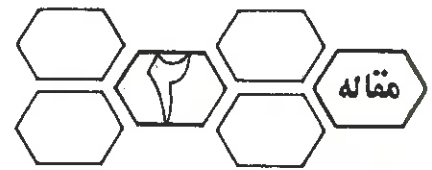
از یکا رگیری کدهای وابسته به سیستم (ماشین) تا حد امکان خود را از نوشتن آنها را از سایر قسمت‌ها جدا کرده و مستندات کافی برای آنها تهیه کنید. چنانچه در محیطی برنامه می‌نویسید که احتمال تعویض کامپیوتر و تبدیل سیستم‌ها وجود دارد، این امر اهمیت ویژه‌ای دارد. مستندات کافی برای برنامه به خود تهیه کنید. هم از جملات توضیحی در درون برنامه استفاده ننماید و هم مستندات نوشته شده‌ای برای برنامه فراهم سازید.

نظریه اشکال زدائی

بسیاری از ما، مدت زمانی بیشتر از آنچه که باید، صرف اشکال زدائی می‌نماییم. برای حصول کارائی بیشتر در این زمینه، همان گونه که در بالا آمد، باید نظام فکری دقیقی داشت. اما سوال این است که از کدام نظام باید پیروی کرد؟

تجربه شخصی من این است که در امثال -زدائی باید مهم‌روشهای علمی و هم راه‌های تجربی را یکا ر گرفت. روشهای علمی در مورد اشکال -زدائی را می‌توان به شرح زیر خلاصه کرد:

- ۱ - تشریح خطا
- ۲ - جمع آوری داده‌ها و اطلاعات در مورد رفتار برنامه
- ۳ - حدس زدن در مورد علت بروز خطا (به نحوی که با داده‌های جمع آوری شده سازگار باشد) و در پیش گرفتن راه‌های جهت تصحیح آن
- ۴ - آزمون حدس (تصحیح خطای فرضی و آزمون این که آیا اکنون برنامه درست کار می‌کند)
- ۵ - چنانچه حدس اشتباه بود برگشت به مرحله ۲ و در غیر این صورت خطا یافته و تصحیح شده است



روشهای اشکال زدائی برنامه‌ها

ترجمه: ابراهیم نقیب زاده، ماسیخ

مقدمه

هیچکس دوست ندارد برنامه‌های را اشکال -زدائی کند زیرا این کار، بسیار خسته کننده، وقت گیر، و گاهی اوقات سرگشته‌آور است. اشکال زدائی برنامه خود، بقدر کافی ناراحت کننده و دشوار است تا چه رسد به اشکال زدائی برنامه که دیگر (که معمولاً) هم برای سوال کردن در دسترس نیست). با این وجود، این کاری است که باید انجام پذیرد.

این مقاله، بر تجربیات شخصی نویسنده در زمینه اشکال زدائی برنامه‌ها و نیز متون و مستندات چاپ شده در این زمینه، مبتنی است. بطور خلاصه می‌توان گفت که هیچ روش معجزه آسائی برای اشکال زدائی برنامه‌ها وجود ندارد و این کار عمدتاً از طریق استفاده منظم و دقیقی از منطق و شعور انسانی امکان پذیر است. البته، روشهای مفیدی نیز وجود دارند که می‌توانند به عنوان ابزاری جهت این کار مورد استفاده واقع گردند. و ما در این مقاله به شرح آنها خواهیم پرداخت. ولی همچنان که گفته شد، شخصی که به اشکال زدائی برنامه می‌پردازد باید بخود، عقل و منطقش را به کار اندازد و ابزار مناسب کار خویش را برگزیند.

با وجودی که سعی نویسنده بر این بوده است که مقاله تا حد امکان عمومی‌تر باشد، ولی فرض بر این گذاشته شده است که برنامه‌ها به زبانهای برنامه نویسی سطح بالا نوشته شده‌اند و هیچگونه ابزار پیچیده‌ای نیز برای اشکال -زدائی در اختیار برنامه نویسنده قرار ندارد.

پیش از اشکال زدائی

چنانچه شما خود، نویسنده برنامه هستید بهتر است پیش از آنکه زیر بار بنشینید تا چنانچه در آینده ناچار به اشکال زدائی برنامه خود شدید، بتوانید این کار را راحت‌تر انجام دهید: از روش برنامه‌سازی ساختار یافته استفاده کنید. البته این صرفاً "بدان معنی نیست که از استفاده از جملات GOTO اجتناب کنید بلکه منظور عدم استفاده ناچار از آنها است. استفاده از روش برنامه‌سازی ساختار یافته، در هر زبان برنامه نویسی امکان پذیر است و این امر، منحصر به زبانهای که اصطلاحاً "ساختار یافته" نامیده می‌شوند (مانند پاسکال یا ماچولا-۲) نیست. برنامه‌سازی ساختار یافته، یک نظام است و یکا رگیری آن، بیش از هر چیز دیگری که در این مقاله ذکر خواهد گردید، به بیسای برنامه‌ها کمک می‌کند.

روش، وقت گیر و خسته کننده است اما غالباً خطاهای آشکارا میتوان از این طریق یافت. البته به منظور بکار بردن این روش باید با نحوه عمل جملات برنا به بطور دقیق آشنا گشت داشت و در شبیه سازی اجرای برنا مه، دقت کافی به عمل آورد، و گرنه شما نیز همان اشتباهی را که منجر به بروز خطا شده است، تکرار خواهید کرد. (کار دیگری که میتوان انجام داد این است که خطا را برای کسی دیگری تشریح نمود. حتی اگر این فرجه یزادی در مورد کامپیوتر نداشتند و تنها در اهلی پیشنها دکنند، غالباً این عمل با زگوشتی، منجر به کشف عواملی میشود که شما به هنگام کار فردی، آنها را نادیده می گرفتید.)

گاهی اوقات با مشاهده ساختار برنا مه، روند نما ۱۸، با هر نوع مستندات مشابه دیگر، میتوان خطای را کشف کرد. زیرا این مستندات در سطح بالاتری از کدهای برنا مه قرار دارند و مشاهده خطاهای منطقی طراحی برنا مه در آنها ساده تر است.

به کتا بچه مربوط به اشکال زدا ثیهایی قبلی خود نیز رجوع کنید. گاهی اوقات اشکالی که فعلاً با آن مواجه هستید، شبیه اشکالات رفع شده قبلی می باشد و یا حداقل این که اشکالات قبلی و روشهای که برای از بین بردن آنها بکار بسته اید، ممکن است ایده های در اختیار شما بگذارند.

قراردادن دستورات چاپ

دستورات چاپ، متداولترین ابزار ورق برزید

و بهتر قابل نگا هاری باشد. مسائل و مشکلات خاص زبان برنا مه نویسی مورد استفاده را در نظر داشته باشید (به عنوان مثال، استفاده اشتباه از اشاره گر ها ۱۶ در زبان C، و یا نحوه عمل خاص یک ثابت ۱۷ یا دستور العمل در زبان اسمبلی). احتمال استفاده از نمادهای شبیه به هم (حروف I، O، 1، و اعداد 0 و 1) و یا وجود فواصل خالی جا افتاده را بررسی کنید.

روشهای اشکال زدا ثی

روشهای اشکال زدا ثی را میتوان در سه رده قرار داد: روشهای از بین بردن خطاهای مشهود، روشهای یافتن خطاهای پنهان و بررسی صحت برنا مه، و پیشگیری از خطاهای آینده. روشهایی که ذیلاً تشریح خواهند گردید، در این سده قرار دارند و به ترتیب تحت نامهای "روشهای غیر مستقیم"، "آزمایش در مقابل اشکال زدا ثی"، و "اشکال زدا ثی پیشگیرنده" را را خواهند شد.

۱- روشهای غیر مستقیم

بسیاری از خطاهای برنا مه ها کلاً قابل رؤیت می باشد. شما می دانید چیزی اشتباه است زیرا می توانید آن را ببینید. اما علل بروز آنها این چنین نیستند. یک راه جهت کشف علت بروز خطا، "بازی کردن نقش کامپیوتر" است. به عبارت دیگر، با در دست داشتن لیست برنا مه و کما غنوم داد، برنا مه را بطور دستی اجرا کرده و مقادیر مختلف متغیرها را بر روی کاغذ، ثبت و پیگیری می کنید. این

با شند، از نظر دور داشت ولی از طرف دیگر، باید این نکته را در نظر داشت که سخت افزار کامپیوتر در ۹۹/۹٪ و نرم افزار سیستم (سیستم عامل، زبانها، و روالهای سودمند ۱۰) در ۹۹٪ اوقات، بی خطا می باشد. این واقعیات را میتوان در سه فرضیه زیر خلاصه نمود: الف- کامپیوتر هیچگاه خطا نمی کند.

ب- نرم افزار سیستم تقریباً هیچگاه خطا نمی کند.

پ- هر چیزی ممکن است باعث بروز خطا باشد (حتی آنچه در فرضیات الف و ب گفته شد، اما با احتمالات مختلف).

تجربه اشکال زدا ثی

در زیر، بطور خلاصه تذکراتی در مورد اشکال زدا ثی آورده شده است:

ه همواره رگور دن نوشته شده و سابقه ثبت شده ای از کارهای خود در زمینه اشکال زدا ثی تهیه نکنید. شکل ظاهری این رگور، اهمیت ندارد ولی باید شامل عباراتی در مورد نحوه بروز خطا، فرضیات، آزمایش و بررسی آنها و نتایج گرفته شده، سوالهای بدون پاسخ مانده، و هر چیز دیگری که به ذهن میرسد، باشد (جدول ۱، یک نمونه از چنین رگور دی را نشان میدهد). اگر این رگوردها بطور فعل و انفعالی ۱۱ در اختیار استفاده کننده باشند، مفید ترند.

ه همیشه با یک نسخه از برنا مه را قبل از اعمال هر گونه تصحیحاتی، نگا هاری کرد. در این صورت، چنانچه تصحیحات و تغییرات به عمل آمد، کار را خرابتر کرد، نیازی به تلاش در جهت بازگرداندن وضعیت اولیه نخواهد بود. بطور کلی، نگا هاری نسخه های مختلف برنا مه در مراحل مختلف اشکال زدا ثی، مفید است. ه به هنگام خواندن یک لیست چاپی، مطمئن باشید که آن لیست، آخرین لیست از برنا مه است. جهت اطمینان بیشتر، بد نیست بطور متناوب لیست جدیدی از برنا مه تهیه کنید. ه حالتی مختلف برنا مه را از زبانهای کتبی در هر یک، تنها یک چیز را تغییر دهید و نتایج را به دقت تجزیه و تحلیل کنید. این کار ممکن است در مورد طبیعت خطای که در مدد تصحیح آن می باشد، سرنخی در اختیار شما قرار دهد.

ه از نتایج منفی نیز اطلاعات لازم را کسب کنید. حتی آزمایشهایی که ظاهراً اطلاعاتی نمی رسند در واقع این چنین نیستند. ه احتمال نوشتن مجدد برنا مه به جای اشکال زدا ثی آن را از نظر خود دور سازید. چنانچه برنا مه به شکل بسیار بدی نوشته شده باشد، ممکن است نوشتن مجدد آن، سریعتر از اشکال زدا ثی صورت پذیرد و به علاوه، برنا مه حاصل، خوانا تر

جدول ۱ - یک نمونه از رگورده ثبت شده در اشکال زدا ثی. چنین رگوردهایی میتوانند در آینده به اشکال زدا ثی سایر برنا مه ها، کمک کنند.

محیط: به هنگام اشکال زدا ثی یک برنا مه به زبان BASIC 2.00 مایکروسافت پروری یک کامپیوتر "مک این تاش" با ظرفیت ۱۲۸ K، خطای "Type Mismatch" در اجرای SUB "get.field(new.item\$,1)" بروز می کرد. get.field به صورت "SUB get.field(msg.string\$,button.pressed)STATIC" تعریف شده بود. new.item\$ میتواند هر مقداری داشته باشد (در مثال مورد نظر، یک رشته خالی بود) و 1، مقدار معتبری برای پارامتر دوم (که میتواند ۱ یا ۲ باشد) بود.

اشکال زدا ثی:

با هر مقدار دیگری که در مکان دوم می گذارم (۲، ۳، و غیره)، خطای مشابهی بروز می کرد. برنا مه هنگامی که این مقدار را به متغیری نسبت میدادم و در جمله Call از نام آن متغیر استفاده می کردم، به خوبی کار می کرد، اما متغیر را بررسی کردم و متوجه شدم که مقدار آن، به دلیل نامعلومی تغییر پیدا نکرده است. برنا مه همچنین هنگامی که از مقدار "۳/۳" استفاده می کردم کار می کرد ولی این مطلب در مورد "۰ + ۱" صادق نبود. چرا؟

دلیل: متوجه شدم که در این گونه ۱۲ بپسپیک، "۱" از نوع صحیح ۱۳ و "button.pressed" از نوع "عقیدتی با دقت معمولی" ۱۴ است. این امر هنگامی که از مقدار ثابت استفاده می کردم، باعث بروز خطای "عدم تطابق گونه ها" می گشت ولی نه در موقعی که متغیر به کار می بردم. من حدس می زدم "۳/۳" به صورت یک مقدار ثابت با دقت معمولی محاسبه می گرد و نتیجه "۰ + ۱"، یک مقدار صحیح می باشد.

راه حل (ها):

- (۱) - مقدار را با بدقت از طریق متغیرها به برنا مه فرستادم.
- (۲) - با بدقت متغیر ساختگی ۱۵ را با استفاده از DEFINT یا با الحاق علامت "%" به انتهای آن (یعنی "button.pressed%") به یک متغیر صحیح تبدیل نمود.
- (۳) - با بدقت را ثابت را به یک عدد حقیقی با دقت معمولی تبدیل کرد (با وارد کردن آن به صورت "1.0")

نتیجه گیری:

در آینده باید مواظب گونه های متغیرها در زبان BASIC 2.00 مایکروسافت بر روی ماشین "مک این تاش" بود.

اشکال زدائی می‌باشد اما غالباً به‌نحوی شری بکار گرفته نمی‌شوند. از دستورات چاپ می‌توان به منظور جمع آوری اطلاعات دربارهٔ رفتار برنامه (مثل "تایم این متغیر، کاری غیرعادی انجام می‌دهد؟")، آزمایش یک فرضیه ("تایم مقدار این متغیر، اشتباه است؟") و یا اطمینان از عمل صحیح برنامه ("تایم این متغیر، حاوی مقدار صحیحی می‌باشد؟")، استفاده کرد.

دستورات چاپ اگر بطور مناسبتی بکار گرفته شوند، ابزاری کارآمد و قوی هستند. عامل در این رابطه نقش تعیین کننده‌ای دارند؛ از دستورات چاپ به چه منظوری استفاده می‌کنید، چه متغیرهای را مورد بررسی قرار می‌دهید، و در گجای برنامه آنها را بررسی می‌کنید. چنانچه در مورد هریک از دو عامل آخر، دچار اشتباه شوید، اطلاعات جمع آوری شده هیچگونه کمکی به شما نخواهند کرد و فقط با این کار، وقت خود را تلف کرده‌اید.

دستورات چاپ ممکن است به صورت غیر-شرطی مانند

```
6808 PRINT "At 6808, X=";X;
      "A$=";A$
```

یا شرطی نظیر

```
6808 IF A$="" THEN PRINT
```

```
"At 6808, A$ empty, X=";X
```

بکار گرفته شوند. دستورات چاپ غیرشرطی به سادگی قابل قرار دادن در برنامه هستند لیکن ممکن است باعث چاپ شدن تعداد بسیار جملات نسبتاً بی‌فایده گردند؛ مثلاً "اگر خط شماره ۶۸۰۸ فوق، در میان سه حلقه متداخل DO قرار داشته‌باشد. دستورات چاپ شرطی معمولاً مفید-ترند زیرا تنها در مواقعی که شرط مهمی (از نظر برنامه نویسی) برقرار شده‌باشد، اطلاعاتی را چاپ می‌کنند.

با بدینا داده‌اش که دستورات چاپ (و روشهای اشکال زدائی بر مبنای دستورات چاپ که در زیر ارائه خواهد گشت)، حداقل در مورد سه دستگاه قابل اجرا می‌باشد: صفحه نمایش، چاپگر، و دیسک. با استفاده از صفحه نمایش، بطور فعل و انفعالی رابطه بین علت (مقدار چاپ شده: متغیرها) و اثر آن (بر روی برنامه)، در اختیار شما قرار می‌گیرد لیکن ممکن است خروجی عادی برنامه در بر روی صفحه نمایش از بین بریزد. خروجی چاپ شده بر روی کاغذ، رکوردی را در اختیار می‌گذارد که بعدها نیز می‌توان آن را به همراه لیست برنامه مورد مطالعه قرار داد. قرار دادن خروجیها در بر روی دیسک، هنگامی سودمند است که تعداد دستورات چاپ بسیار زیاد باشد. در این صورت، تولید خروجی سریعتر از چاپگر صورت می‌گیرد و در ضمن، بسیار استفاده‌ایک کلمه پرداز ۱۹ می‌توان به سرعت، خروجی تولید شده را مورد جستجو و بررسی قرار داد. انتخاب هریک از روشهای فوق، باید با توجه به شرایط خاص هر مسأله صورت پذیرد. گاهی اوقات شما با یک خطای مشهود مواجه می‌شوید و ایده‌ای درباره علت بروز آن ندارید.

در این حالت می‌توانید با استفاده از دستورات چاپ، به روش زیر عمل کنید: ابتدا نقطه‌ای از برنامه را با یکدیگر هم چیدار آن روبرو است و اشکالی وجود ندارد و سپس نقطه دیگری که اینگونه نیست. پس از آن بطور مکرر دستورات چاپ را بین دو قسمت قرار دهید و ببینید که کدام نیمه حاوی خطا است. با تکرار این عمل، سرانجام به مرحله‌ای خواهید رسید که محدوده‌ای که غلط در آن وجود دارد بسیار کوچک خواهد شد و طبیعتاً "در چنین محدوده کوچکی، تعداد دعل و عوامل مختلفی که ممکن است باعث بروز خطا شده باشند نیز بسیار کم خواهد بود.

نقاط انفعال ۲۰

چنانچه بتوانید برنامه خود را در نقطه‌ای متوقف کنید، قادر به تغییرها را بررسی نموده و تغییر دهید، و اجرای برنامه را از نقطه متوقف شده از سر بگیرید، به چنین نقاطی، نقاط انفعال و به این توانائی، قابلیت ایجا نقاط انفعال در برنامه گفته می‌شود. در اغلب زبانهای بیسیک ما یک روش است، می‌توان توسط دکمه "control-c" از روی صفحه کلید و یا با دستور STOP در برنامه، اجرای برنامه را متوقف نمود و پس از بررسی و حتی تغییر مقدار متغیرها، به کمک دستور CONT، اجرای برنامه را از سر گرفت. در بسیاری حالات لازم است قادر به بررسی متغیرها قبل از توقف! اجرای برنامه، چاپ شوند.

نقاط انفعال، امکان اشکال زدائی برنامه را به طور فعل و انفعالی را فراهم می‌سازند. خصوصیات آنها غالباً شبیه دستورات چاپ است. به عبارت دیگر، هم می‌توانند به صورت شرطی و هم غیرشرطی اجرا گردند. همچنین می‌توانند جهت جمع آوری اطلاعات، تاء پیدیک حدس، و تعیین محل خطا بکار روند. میزان تاء ثیر آنها بستگی به نحوه استفاده‌شان دارد. بسیاری از برنامه‌نویسها از نقاط انفعال به عنوان مکانیسمی جهت به دست آوردن سرنخی در مورد خطا استفاده می‌کنند. بدین ترتیب که اجرای برنامه را در نقاط مختلف و متعددی قطع می‌کنند و پس از بررسی مقدار متغیرها می‌پردازند. این نحو استفاده از نقاط انفعال، صحیح و موثر نیست و شاید تنها در مواقعی مفید واقع گردد که شخص از انجام هر کاری عاجز ماند و دیگر نداند که باید چه کند.

با استفاده از نقاط انفعال، معمولاً امکان دستیابی به "وضعیت" ماشین - یعنی تمام داده‌های که جهت دوباره سازی کامل موقعیت کامپیوتر در یک لحظه خاص مورد نیازند - برای برنامه نویسی فراهم می‌گردد؛ البته ممکن است این اطلاعات مورد استفاده اوقات را بگیرند. به همین دلیل، نقاط انفعال، معیاری هستند که در بین دستورات چاپ و تصاویر لحظه‌ای (که در زیر مورد بحث قرار خواهد گرفت) واقع شده‌اند. نقاط انفعال و تصاویر لحظه‌ای از دو نقطه نظر با یکدیگر متفاوتند: ۱ - نقاط انفعال

همواره در دسترس نیستند. ۲ - استفاده از تصاویر لحظه‌ای سخت‌تر ولی اطلاعات حاصل از آن، کاملاً نرمی‌باشد.

تصاویر لحظه‌ای

تصویر لحظه‌ای، رکوردی چاپ شده از وضعیت ماشین است. به دلیل آن که تصاویر لحظه‌ای، وضعیت برنامه را در لحظه مربوطه کاملاً مشخص می‌سازد، بهترین ابزاری است که می‌توان به کمک آن از "محتویات" برنامه در آن لحظه خاص آگاهی یافت. این قابلیت، هم خوب است و هم بد. خوب است به دلیل آن که تمام اطلاعاتی که ممکن است مورد نیاز واقع شوند، در دسترس خواهد بود و بد است به این جهت که ممکن است نتوان اطلاعات مورد نیاز را از میان انبوه اطلاعات ارائه شده، یافت. تصویر لحظه‌ای نیز نوعی مدرک چاپ شده از وضعیت برنامه در یک لحظه خاص است و به دلیل آن که می‌تواند بعدها مورد رجوع قرار گیرد، نیاز به دوباره سازی وضعیت را از بین می‌برد. (گاهی در امثال اشکال زدائی، لحظاتی پیش می‌آید که نتایج فعلی شما ظاهر را "با آنچه از نتایج قبلی بیاد دارید، متناقض به نظر می‌رسد. در این لحظات نیز تصاویر لحظه‌ای، وسیله خوبی جهت رفع ابهام می‌باشد).

تصاویر لحظه‌ای معمولاً بطور فعل و انفعالی مورد استفاده قرار نمی‌گیرند (البته می‌توانند اینچنین هم باشند به شرطی که بخواهید منتظر چاپ اطلاعات بمانید). غالباً از آنها همراه با لیست تمیزی از برنامه، برای به دست آوردن سرنخی از علت بروز خطا استفاده می‌شود. همچنین تصاویر لحظه‌ای بسیار طولانی هستند و چاپ آنها مدت زمان زیادی بطول می‌انجامد (در بعضی حالات مجبورید آنها را روی دیسک بنویسید). بنا به دلایل فوق، تصاویر لحظه‌ای معمولاً به عنوان آخرین وسیله مورد استفاده قرار می‌گیرند. جهت به دست آوردن تصاویر لحظه‌ای در یک زبان سطح بالا، لیستی از تمام متغیرهای برنامه به سیستم به دست آورید. سپس زیرروالی ۲۲ برای چاپ آنها بنویسید (بهتر است ترتیب حروف الفبائی را رعایت کنید). برای استفاده از تصاویر لحظه‌ای، خطی به برنامه خود اضافه کنید که پس از چاپ توضیحات لازم، زیر-روال مربوطه را حاضر کند. برای مثال، در زبان بیسیک می‌توان نوشت:

```
3305 PRINT "Snapshot at end
of input routine":GOSUB 9950
```

به هنگام اشکال زدائی برنامه‌های به زبان اسمبلی، محتویات تمام حافظه را به دست آورید و نتایج تصاویر لحظه‌ای و مقدار دیسک شما خطای ۲۳ پردازنده و نیز مقدار دیرشها را ثبت نماید. در صورت امکان، محتویات حافظه را به صورت یک "روبردار" قالب بندی شده ۲۴ که با بهترها هم به صورت اعداد در مبنای شانزده و هم نویسه‌های اُسکی ۲۵ نشان می‌دهد، ضبط



وسپس پرونده‌ای از اطلاعات مفید برای اشکال زداشی را بر روی دیسک می‌نویسد. را شا ممل باشد.

بهترین موقع جهت افزودن چنین دستور-العمل‌های به برنا مه، هنگام نوشتن (وبعد، هنگام اشکال زداشی) آن می‌باشد زیرا در این هنگام، بیشترین بصیرت در مورد چگونگی کارکرد برنا مه وجود دارد. برای این منظور، سه‌الاتی ما ننداخته‌اند زیرا آمده است از خود بنما می‌سند: "بیشترین چیز خسارت زننده‌ای که ممکن است به هنگام اجرای برنا مه اتفاق افتد، کدام است؟ برنا مه برای کشف آن چه کاری می‌تواند انجام دهد؟ چه محدوده‌ای از داده‌ها هوارنا معتبر خواهند بود؟ استفاده کننده برای آن که برنا مه در مقابل این اتفاق حفاظی داشته‌باشد یا آن را تصحیح نماید، چه می‌تواند بکند؟" با برنا مه-ریزی و طراحی دقیق می‌توان دستورالعمل‌های خفته-اشکال زداشی را به نحوی به برنا مه اضافه نمود که خط‌های آتی برنا مه را به اطلاع برساند و کوشش لازم جهت اشکال زداشی را به حداقل ممکن کاهش دهد.

روش "حفاظ بندی"، نوعی اشکال زداشی پیشگیرنده است که می‌توان آن را در برنا مه‌های که کاملاً واحدی یا پیمانه‌ای نباشند، بکار بست. چنانچه طراحی واحدی برنا مه به صورت صحیحی انجام گرفته‌باشد، هروا حد کاملاً مستقل خواهد بود و تنها از طریق تعداد مشخصی از متغیرها با واحدهای دیگر ارتباط خواهد داشت. "حفاظ بندی"، نوع خاصی از دستورالعمل‌های خفته-اشکال زداشی است که اعتبار داده‌های که از یک واحد خارج می‌شوند را کنترل می‌نماید. ایده حفاظ بندی این است که چنانچه خط‌های در یک واحد بروز نمود، از انتشار رگسترش آن به سایر واحدها جلوگیری شود. بکارگیری این روش بیشتر در سیستم‌های بزرگی که داده‌ها از یک برنا مه به برنا مه دیگر انتقال می‌یابند، مفید واقع می‌شود اما ممکن است بتوانید آن را در برنا مه‌های کوچک خود نیز بکار ببندید.

نتیجه‌گیری

همان گونه که در ابتدای مقاله ذکر گردید، هیچ روش معجزه‌آسایی برای اشکال زداشی وجود ندارد اما برای این کار، ابزارهای در اختیار قرار دارند که قویترین آنها، مغز خود انسان است. بخشی از ما اشکال زداشی بکار بردن روش‌های اصولی، قسمتی از آن شانس، بخشی از آن هنر، و پیچیده‌ترین و مشکل‌ترین قسمت آن، "فکر کردن عمیق و سخت" در شرایط و موقعیتهای دشوار می‌باشد.

وسوهای که در انجام اشکال زداشی وجود دارد، ساده‌گرفتن کار، رد شدن از قسمتهای مشکل، و بازبینی مکرر برنا مه برای یافتن سرنخهای بیشتر به جای فکر کردن در مورد همان یک سرنخ یافته شده است و این درست مانند کسی است که به دنبال کلید گمشده، ما شیش در زیر بقیه در صفحه ۱۵

توجه به متغیرهای باقی‌مانده، مورد بررسی قرار می‌گیرد. با انجام این عمل برای متغیرهای مختلف و بررسی نتایج آن می‌توان پیاپی نکته‌های برده‌گذاشته متغیر با عث بروز خطا می‌گردد. پس از یافتن متغیر مربوطه، با بر متغیرهای که در رابطه با آن متغیر هستند، با مقادیر ثابت جایگزین می‌شوند و با مطالعه رفتار واقعی سیستم در مقابل به رفتار پیش-بینی شده برای آن، می‌توان به منشأ خطا پی برد.

با استفاده از این روش می‌توان نحوه عمل بخش‌های مختلف برنا مه را نیز آزمایش نمود. هنگامی که برنا مه دارای مسیرهای مختلفی از ابتدا تا انتها می‌باشد، با استفاده از این روش و وارد نمودن داده‌های آزمایشی می‌توان هر یک از مسیرهای احتمالی برنا مه را بطور جداگانه مورد بررسی قرار داد.

طراحی و انتخاب داده‌های آزمایشی، بر اعتبار عمل آزمایش برنا مه تا شیز زیادی دارد. بحث مفصل در مورد این موضوع، خارج از حوصله این مقاله است اما با بدخا طر نشان ساخت که استفاده از داده‌های آزمایشی، ابزار کارآمدی برای کشف و تصحیح خطاهاست. به عنوان مثال، هنگامی که برنا مه مورد نظریا محدوده‌ای از داده‌ها سروکار دارد، سعی کنید داده‌های ورودی آزمایشی را درست در نقاط انتهایی و خارج و داخل آن محدوده در نظر بگیرید. مثلاً، چنانچه برنا مه شما میانگین بین ۱ تا ۲۰ عدد را محاسبه می‌کند، آن را با ۵، ۱۰، ۱۹، ۲۰، ۲۱ عدد مورد آزمایش قرار دهید. عموماً، تعداد دفعاتی که برنا مه به درستی کار نخواهد کرد، زیاد و متعجب‌آور خواهند بود.

اشکال زداشی پیشگیرنده ۲۸

منظور از اشکال زداشی پیشگیرنده، تعبیه مکانیسمی در برنا مه جهت آگاه‌ساختن استفاده کنندگان بعدی از وقوع یک خطا می‌باشد. در این رابطه، دوروش وجود دارد. اولی که روش نسبتاً ساده‌ای است، استفاده از دستورالعمل‌های "خفته" ۲۹ اشکال زداشی می‌باشد و روش دوم به نام "حفاظ بندی" ۳۰ خوانده می‌شود. هر دو روش در دسترس کسانی است که در حال نوشتن برنا مه می‌باشند، گرچه در برخی موارد می‌توان با افزودن جملاتی به برنا مه‌های آماده نوشته شده قبلی نیز آنها را بکار بست. کاربرد این روش‌ها خصوصاً در برنا مه‌های تشخیصی و تصحیح سریع خطا در آنها اهمیت زیادی دارد، مفید است.

"دستورالعمل‌های خفته" اشکال زداشی، روال‌های تشخیصی و شرطی ۳۱ هستند که تنها هنگامی که یک شرط غیرعادی از پیش تعریف شده آنها را برپا نماید، فعال شده و اجرا می‌گردند. این روال‌ها می‌توانند از یک دستورالعمل ساده که وقوع خطای را به اطلاع می‌رسانند تا یک زیر-روال پیچیده که وضعیت برنا مه را تحلیل نموده

کنند. البته تنها ویرلحظه‌ای را می‌توان محدودتر نیز نمود و اطلاعات نواحی بخصوصی (مثلاً "حافظه فقط خواندنی ۲۶ سیستم) را در نظر گرفت و لسی با ایده‌ها راه متوجه این مسأله بود و چه بسا که علت خطا، در همان ناحیه‌ای قرار داشته‌باشد که در نظر گرفته نشده است!

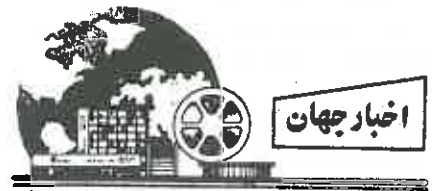
آزمایش کردن ۲۷ در مقابل اشکال زداشی

قبل از بحث در مورد یافتن خطاهای برنا مه و بررسی صحت برنا مه‌ها، لازم است تفاوت بین اشکال زداشی و آزمایش برنا مه را با طر نشان سازیم. "اشکال زداشی"، به فرا روند تصحیح تمام خطاهای کشف شده در یک برنا مه اطلاق می‌گردد و هنگامی که اشکال دیگری یافت نشود، خاتمه می‌یابد. اما "آزمایش" برنا مه عبارت است از فرا روند اثبات صحت رفتار برنا مه در تمام حالات، برای این منظور، معمولاً داده‌های ورودی آزمایشی به گونه‌ای انتخاب می‌شوند که تمام مسیرهای محتملی که برنا مه ممکن است از ابتدا تا انتها طی نماید، پیموده شوند و پس از بررسی و اطمینان از صحت نتایج حاصله، مستندات لازم جهت ملاحظات بعدی تهیه می‌گردد.

آیا می‌توان پس از کشف و تصحیح تمام خطاهای در برنا مه، ادعا کرد که دیگر برنا مه کاملاً صحیح است؟ متأسفانه جواب منفی است. زیرا آنچه یافته و تصحیح گشته، تنها خطاهای "مشهود" و آشکار بوده‌اند. به عبارت دیگر، اشکال زداشی برنا مه صورت گرفته ولی هنوز آزمایش آن انجام نشده است. اطمینان از بی‌خطا بودن برنا مه بطور کامل، تقریباً ناممکن است اما با آزمایش کردن آن می‌توان تا حد معقولی از صحت کارکرد آن اطمینان یافت. دوروش جهت یافتن خطاهای برنا مه و بررسی صحت برنا مه بکار می‌رود.

دوروش اول که Forcing نام دارد، برنامۀ واحدی که رابطه بین ورودی و خروجی آن مشخص گردید با سهولت قابل محاسبه‌باشد، ساده می‌شود. سپس برنا مه با داده‌های آزمایشی مورد امتحان قرار می‌گیرد و نتایج به دست آمده با آنچه پیش‌بینی می‌شد، مقایسه می‌گردد. در بعضی حالات، می‌توان مجموعه داده‌های ورودی را به طور قابل ملاحظه‌ای ساده‌تر از آنچه که باید باشد در نظر گرفت (به عنوان مثال، در یک برنا مه مربوط به تحلیل‌های آماری، می‌توان به جای ۱۰ یا ۱۰۰ داده ورودی، تعداد محدودتری را در نظر گرفت). در برخی موارد نیز می‌توان برنا مه را به گونه‌ای تغییر داد که در یک نقطه خاص بجای بکار بردن یک مقدار محاسبه شده در خلال برنا مه، از یک مقدار ثابت استفاده کند و نتیجه به دست آمده را مورد بررسی قرار داد.

روش دوم، بررسی برنا مه با داده‌های آزمایشی است. در این روش که در واقع گونه‌ای از روش اول است و بطور متداولی بکار گرفته می‌شود، برخی از متغیرهای برنا مه با مقادیر ثابت جایگزین می‌گردند و سپس رفتار برنا مه با



آپریزکا مپیوترا جدید

اپریزکا مپیوترا چندپردازنده ای P/15 که جدیداً به بازار عرضه گشته، تحت سیستم عامل یونیکس کامپیوتندوفا در دست راست تا استفاده کننده را پشتیبانی نماید. این کامپیوتر با ارتفاعی کمتر از ۶۵ سانتیمتر و وزنی در حدود ۳۵ کیلوگرم، ۱۴۴ مگابایت حافظه دیسک و پنچستر ۴۵ مگابایت نوار کارتریج پشتیبانی با دیسک لرن ۲ و ۴ مگابایت حافظه اصلی را در اختیار استفاده کننده قرار می دهد.

پردازنده این کامپیوتر تراشه ۳۲ بیتی MC68010 با سرعت ۱۰ مگاهرتز و بدون حالت انتظار ۳ تشکیل می دهد. واحد پردازنده مرکزی می تواند ۸ مگابایت حافظه I/O را نشانی دهد. یک پردازنده ۳۲ بیتی جداگانه نیز برای کارهای ورودی و خروجی به همراه این کامپیوتر می باشد.

این سیستم با نرم افزارهای تولید شده بر مبنای سیستم عامل یونیکس، هم ساز می باشد. قیمت P/15 با ۵/۵ مگابایت حافظه RAM، ۱۲ مگابایت حافظه دیسک سخت، و یک مگابایت دیسک لرن، معادل ۱۰۹۵۰ دلار است.

به نقل از

EDN, July 85

- 1: Supermicrocomputer
- 2: Floppy disk
- 3: Wait State
- 4: Compatible
- 5: Hard-disk

ریزپردازنده جدید اینتل

کمپانی اینتل هم اکنون فروش نمونه هائی از ریزپردازنده ۳۲ بیتی جدید خود به نام ۸۰۳۸۶ را همراه با تخته مدارهای CPU و ابزارهای نرم افزاری مربوطه، آغاز نموده است. بر طبق ادعای اینتل، این تراشه قادر به اجرای ۳ تا ۴ میلیون دستورالعمل در ثانیه می باشد. ریزپردازنده ۸۰۳۸۶، دارای ۴ گیگابایت فضای نشانی مستقیم و حافظه مجازی ۲ به گنجایش ۶۴ تریبلیون بایت می باشد.

معماری خط لوله ای ۸۰۳۸۶، امکان اخذ ۳ رمزگشائی و اجرای دستورالعملها را در یک زمان، فراهم می آورد. این تراشه، از نظر مدیریت حافظه، هم به بند شدن (هر بند تا ۴ گیگابایت) و هم صفحه بندی را پشتیبانی می نماید و طراحی آن به گونه ای است که با تمام نرم افزارهای نوشته شده برای ریزپردازنده

های قبلی ۱۸۰۳۸۶ اینتل (۸۰۳۸۶، ۸۰۳۸۸، ۸۰۱۸۸، ۸۰۲۸۶)، کاملاً هم ساز می باشد.

ریزپردازنده ۸۰۳۸۶، قادر است برنامه های کار بردی مختلف را تحت سیستمهای عامل متفاوت، بطور همزمان اجرا کند. با وجودی که تراشه ۸۰۲۸۷ می تواند به عنوان پردازنده عددی همراه ۸ برای ۸۰۳۸۶ بکار گرفته شود، ولی تولید یک پردازنده همراه سرهمتر به نام ۸۰۳۸۷ نیز برای آن برنامه ریزی گردیده است. قیمت این تراشه، برای فروش در کمیتهای نمونه، ۲۹۹ دلار تعیین گردیده و تولید کامل آن، برای انتهای سال ۱۹۸۶ در نظر گرفته شده است.

به نقل از

BYTE

Nov. 1985

- 1: Direct Address Space
- 2: Virtual Memory
- 3: Fetch
- 4: Decode
- 5: Segmentation
- 6: Paging
- 7: Fully Compatible
- 8: Numeric Coprocessor

رادیوگرافی به وسیله کامپیوتر

حدود سه سال پیش، شرکت هلندی فیلیپس شاخه آمریکای شمالی، قراردادی برای ساخت یک سیستم رادیوگرافی کامپیوتری با کمپانی ژاپنی "فوجی" منعقد نمود، کمپانی ژاپنی مزبور، پس از گذشت ۲۰ ماه موفق به ساخت این دستگاه بر طبق مشخصات ارائه شده از سوی فیلیپس گردید و ظرف ۱۶ ماه گذشته، این سیستم در چند کلینیک و بیمارستان آمریکا به آزمایش گذاشته شد. اکنون، نتیجه آزمایشات بسیار موفقیت آمیز بوده و برتری این روش بر روشهای فعلی رادیوگرافی در چند زمینه از جمله توموگرافی (عکس برداری مقطعی) به اثبات رسیده است. سیستم رادیوگرافی کامپیوتری فیلیپس که قرار است به زودی در مجمع عمومی سالانه انجمن رادیولوژیستهای آمریکا به معرض نمایش گذاشته شود، تنها ویری را که با اشعه ایکس از بیماران گرفته می شود، به صورت رقمی در آورده و ذخیره می نماید و در صورت لزوم، بر روی فیلم منتقل می سازد.

در صورت استفاده از این سیستم، به دلیل ضریب حساسیت بسیار بالایی که دار می توان جهت عکس برداری، اشعه ایکس رقیقتری را بر بدن بیمار تاباند و بدین ترتیب، از عوارض تشعشعات مزبور بر روی بدن بیماران به میزان قابل ملاحظه ای کاست. همچنین، این سیستم کامپیوتری به دلیل دقت زیاد، قادر است نقاط و زوایای را در عکس مشخص سازد که ممکن

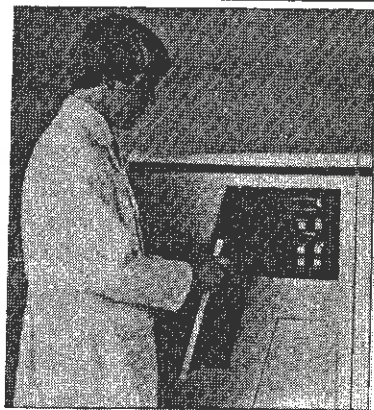
فوجیتسو تولید کامپیوترهای بسیار بزرگ

فوجیتسو با طرحی که در زمینه ادغام تکنولوژی کامپیوترهای بزرگ و بزرگ کامپیوترها دارد، خود را آماده یک رقابت جدی با آی بی ام و دیگر تولید کنندگان کامپیوترهای همساز با آی بی ام می نماید. به گفته "تاکوما یا ما - موتو" رئیس کمپانی فوجیتسو، محصول جدید "به زودی" به بازار عرضه خواهد شد.

بازاریابی و فروش این محصول در خارج از ژاپن، به عهده کمپانی "آمدال" که ۴۹٪ از سهام آن متعلق به فوجیتسو است، می باشد. انتظار می رود که قدرت این ماشین، نزدیک به قدرت ابر کامپیوتر VP فوجیتسو باشد. قدرت محاسباتی این ابر کامپیوتر، از ۱۴۰ میلیون عملیات با ممیزنا و در ثانیه (مگا فلاپ) تا ۱/۱۴ میلیارد عمل (گیگا فلاپ) است.

یا ما موتو می گوید: "ما با کمال تعجب، شاهد استفاده فزاینده از ابر کامپیوترها در محاسبات در مقیاس بزرگ، از سوی مشتریان خود هستیم." وی می افزاید: "ما فکر می کنیم که ابر کامپیوترها و کامپیوترهای بزرگ همواره - منظره ۷، بسیار با یکدیگر همسازند."

هم اکنون، ۴ سال از عمر بهترین کامپیوتر بزرگ فعلی فوجیتسو به نام M-382 می گذرد و



است در فیلمهای اشعه ایکس فعلی، منعکس نگردند. از جمله مزایای مهم و قابل ذکر دیگر، سهولت مقایسه عکسهای است که از یک موضع، قبل و بعد از درمان گرفته می شود. این کار که با استفاده از یک لامپ اشعه کاتدی با قدرت تفکیک پذیری بالا صورت می گیرد به پزشکان معالج امکان می دهد که بطور دقیق اثر درمان انتخاب شده را مطالعه و مشاهده نمایند. قیمت سیستم رادیوگرافی کامپیوتری فیلیپس، ۵۰۰ هزار دلار می باشد و از بهار سال مسیحی آینده به بازار عرضه خواهد شد.

به نقل از

Electronics
Nov. 1985

- 1: Philips Computerized Radiography
- 2: Digitization
- 3: High-Resolution Cathod-Ray Tube

حمل این سیستم کامپیوتری به آمریکا تا دوماه دیگر آغاز می‌گردد و به نظر می‌رسد در آینده نزدیک، وارد بازار اروپا گردد.

به نقل از

Infomatics
Feb. 85

- 1: Wang
- 2: Configuration
- 3: Floppy Disk Drive
- 4: Fixed Disk
- 5: Removable Disk
- 6: Workstation
- 7: Cache Memory

قدرتمندترین ابر کامپیوتر جهان

یک تیم تحقیقاتی در دانشگاه پرینستون، موفق به ساخت قدرتمندترین ابر کامپیوتر جهان شده‌اند. سرعت عملیاتی این ابر کامپیوتر در بالاترین حد، ۶۱/۴ میلیاردمال با ممیزشناور در ثانیه (کیگا فلاپ) و بطور متوسط ۵۱/۲ گیگا فلاپ می‌باشد. ابر کامپیوتر مزبور قادر است تا ۱۲۸ نقطه در داخل یک شبکه را سرویس دهد. در این حالت، سرعت عملیاتی هر یک از نقاط در بالاترین حد، ۴۸۰ میلیون عمل با ممیزشناور در ثانیه (مگا فلاپ) و بطور متوسط ۴۰۰ مگا فلاپ خواهد بود که این سرعت، سه‌برابر بیشتر از سرعت ابر کامپیوتر "کری ۴" می‌باشد.

به نقل از

Electronics
Nov. 1985

- 1: Supercomputer
- 2: Giga flops (billion Floating-point Operation Per Second)
- 3: Mega flops
- 4: Cray-1

شبکه قرار دهد. هر کامپیوتر می‌تواند تا سه دستگاه چاپگر نیز داشته باشد. سرعت انتقال اطلاعات بین کامپیوترها ۱۹/۲۸ بیت در ثانیه می‌باشد و به این منظور از درگاه‌های RS-232C استفاده شده است.

به نقل از

EDN, July 85

- 1: Compatible
- 2: Ports

کامپیوتر جدید کمپانی "ونگ"

کامپیوتر جدیدی که کمپانی "ونگ" عرضه نموده است، رقیبی جدی برای کامپیوترهای سیستم ۲۸۶ آی‌بی‌ام به حساب می‌آید. این سیستم که از سری کامپیوترهای VS "ونگ" است و تحت نام VS65 به بازار عرضه گردیده، جایگزین کامپیوتر VS45 این کمپانی خواهد شد. پیگیر بندی VS65 با یک مگا بایت حافظه اصلی، یک دستگاه گرداننده دیسک لیزر ۳۶۰ کیلوبایتی، ۱۴۷ مگا بایت حافظه دیسک ثابت و ۲۶ مگا بایت حافظه دیسک برداشتنی، چهار ایستگاه کار ۶ VS-4230، و سیستم عامل VS، ۵۴۳۰۰ دلار قیمت دارد. و این در حالی است که قیمت پیگیر بندی مشابه آی‌بی‌ام با ۵۱۲ کیلو بایت حافظه اصلی ۴۰۰ مگا بایت حافظه دیسک، ۷۳۰۰۰ دلار می‌باشد.

کامپیوتر VS65، دارای یک واحد پردازنده مرکزی ۲۲ بیتی اختصاصی است و حافظه اصلی آن تا ۴ مگا بایت قابل افزایش می‌باشد. این سیستم همچنین دارای ۱۶ کیلو بایت حافظه پنهانی بوده و قادر است تا ۳۰ استفاده کننده و ۲/۶ گیگا بایت حافظه دیسک را پشتیبانی نماید. یک پردازنده ۸۰۲۸۶ اینتل نیز جهت پردازش‌های ورودی و خروجی این کامپیوتر، در نظر گرفته شده است.

هنگامی که در اواصل امسال، آی‌بی‌ام تولید سری جدید کامپیوترهای بزرگ خود تحت عنوان سری ۳۰۹۰ را اعلام نمود، نیاز به تولید یک پردازنده جدید، نزد فوجیتسو بیش از پیش احساس گشت.

ماشین جدید فوجیتسو، ۳ تا ۴ برابر قدرتمندتر از ماشین ۳۰۹۰/۴۰۰ آی‌بی‌ام که با سرعت ۴۵ تا ۵۵ میلیون دستورالعمل در ثانیه کار می‌کند، خواهد بود. به علاوه، این امیدواری مهم برای فوجیتسو وجود دارد که محصول جدیدش در سال ۱۹۸۶ به تولید کامل خواهد رسید و این زمان، حداقل ۶ ماه زودتر از عرضه ماشین ۳۰۹۰ آی‌بی‌ام خواهد بود.

رقیب دیگر آی‌بی‌ام در زمینه کامپیوترهای بسیار بزرگ، کمپانی ژاپنی "تک" با ماشین Acos 1500 است. این ماشین، قادر به انجام ۱۳۰ میلیون دستورالعمل در ثانیه می‌باشد و بازاریابی و فروش آن در خارج از ژاپن، به عهده "هانیول" است.

کمپانی هیتاچی نیز تولید پردازنده جدید M-682H را که قادر به اجرای ۴۶ تا ۶۰ میلیون دستورالعمل در ثانیه می‌باشد، اعلام نموده است. این محصول از طریق NAS به فروش خواهد رسید. هیتاچی تولید کامل این محصول را برای سال آینده برنامهریزی کرده و هم‌اکنون، موفق به دریافت ۲۰ سفارش برای آن شده است.

در سال گذشته، "امدال" و NAS هر دو، تاخت و تاز عجیبی در بازار کامپیوترهای همساز با آی‌بی‌ام، در انگلستان داشته‌اند.

به نقل از

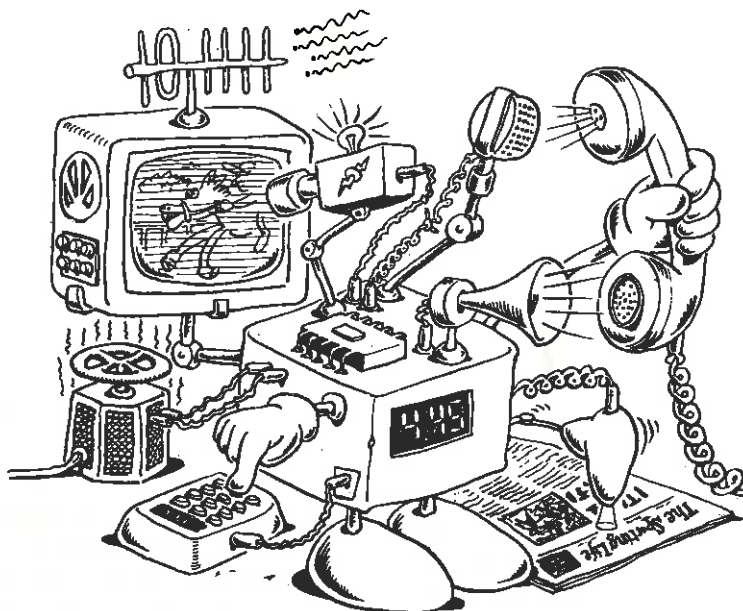
Computing

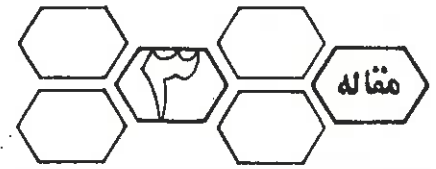
Oct. 1985

- 1: Super Mainframe
- 2: Supercomputers
- 3: Compatible
- 4: Takuma Yamamoto
- 5: Amdahl
- 6: Million Floating point Operation Per Second
- 7: General-purpose
- 8: NEC
- 9: National Advanced Systems

بسته نرم افزاری برای کار با شبکه‌ها

اخیراً یک بسته نرم افزاری مربوط به کار با شبکه‌ها به نام EasyLAN به بازار عرضه شده است. این بسته نرم افزاری که قیمت آن برای هر کامپیوتر ۱۰۰ دلار می‌باشد، امکان انتقال پرونده‌ها بین کامپیوترها و نیز استفاده از یک ایستگاه به عنوان مرکز چاپ، بدون ایجاد وقفه در کار سایر ایستگاه‌ها را فراهم می‌کند. این بسته نرم افزاری می‌تواند تا ۱۰ کامپیوتر شخصی آی‌بی‌ام از انواع XT، AT، و سایر کامپیوترها را با آنها را با یکدیگر در یک





مقدمه‌ای بر نسل پنجم

ترجمه: دکتر محمود تقی‌زاده

در ماه اکتبر سال ۱۹۸۱، در وزارت صنایع و تجارت بین‌المللی ژاپن (MITI)، جهت اعلام یک پروژه ملی جدید، کنفرانسی برگزار گردید. در این کنفرانس اعلام گشت که در کنار دیگر پروژه‌های ملی در مورد ایرکا مپیوترها^۱ و آدمکهای مصنوعی^۲، کوششی نیز به منظور ایجاد یک نسل جدید کامپیوترها (نسل پنجم به تشخیص آنها) به عمل خواهد آمد.

نسل جدید، کامپیوترهایی از انواع قبلی نخواهند بود. بلکه آنها ماشینهای استنباط گر نمادی^۳ هستند که قادر خواهند بود به طور منطقی و به سرعت را خود را از زمین توده‌ای از داده‌ها و آگایها پیدا کنند. آنها کامپیوترهای ثنی خواهند بود که می‌توانند به موزند، استنباط کنند، تصمیم بگیرند، و به عبارت دیگر، رفتاری داشته باشند که معمولاً "در حوزه منطق و استدلال خاص انسان قرار دارد. حتی نام آنها نیز نشانه‌ای از تغییرات است: سیستمهای پردازش آگای^۴ (KIPS). اینها ماشینهای جامع‌الاطلاع خواهند بود: کوچک، توانا، و ارزان. آنها به زودی به صورت ابزارهای جهانی در خواهند آمد و با طرز استفاده‌ای ساده مانند تلفن، در مکانهای عمومی قرار خواهند گرفت.

برنا مدریزی ده ساله، پروژه به سه مرحله تقسیم شده است. مرحله سه ساله اول به ساخت یک نمونه آزمایشگاهی از ماشین اختصاص دارد. این نمونه، شامل یک ایستگاه هکسار شخصی^۵ PROLOG است که دارای یک پایگاه آگای^۶ قابل مقایسه با سیستمهای خبره^۷ امروزی می‌باشد، ولی قدرت استنباط و نتیجه گیری آن، برابر یک میلیون استنباط منطقی در ثانیه^۸ (LIPS)، یعنی چند برابر بیشتر از کامپیوترهای بزرگ کنونی نظیر DEC 2060 است که زبان PROLOG بر روی آنها به صورت نرم افزاری پیاده شده است. این نمونه در سال ۱۹۸۴ به اتمام خواهد رسید و تولید تجاری آن برای حدود یک سال بعد پیش بینی گردیده است. انجام این مرحله برای ژاپن، فرصتی جهت آموختن و تجربه کردن می‌باشد و مشخصاً برای همین منظور نیز برنا مدریزی شده است.

مرحله چهار ساله بعدی برای تجربیات مهندسی، نمونه سازی، ادامه تجربه آموزی در کاربردهای مهم و تجربه اندوزی اولیه در امر مجتمع سازی سیستم می‌باشد. اولین برخورد با مسائل پردازش موازی در این سالها انجام

خواهد گرفت.

مرحله سه ساله آخر، بر روی مهندسی پیشرفته، ساخت نمونه مهندسی نهایی و کارهای دیگر در زمینه مجتمع سازی سیستم تمرکز خواهد داشت. هدف نهایی که برای اوایل سال ۱۹۹۰ برنا مدریزی شده است، حداقل یک ایرکا مپیوتر استنباطی خواهد بود که قادر است از یک میلیون تا یک میلیارد استنباط منطقی در ثانیه انجام دهد. به همراه این کامپیوتر، یک پایگاه آگای با دهها هزار قانون استنباط و دهها میلیون شی (عناصر اطلاعاتی) - تقریباً - به اندازه اطلاعات یک دانشمند المار ف بریتانیکا وجود خواهد داشت. در این پروژه، ژاپنیها به بیشتر به "خود را اندازی" متکی خواهند بود. به عنوان مثال، کاری که در مرحله اول پروژه بر روی طراحی به کمک کامپیوتر^۹ انجام می‌گیرد، بعداً "در طراحی سخت افزار لازم، مورد استفاده قرار خواهد گرفت.

برای این که طبق تصور ژاپنیها این ماشینها نقش مهمی در جامعه داشته باشند، باید به مراتب سهل‌الاستفاده تر از کامپیوترهای کنونی باشند. به این جهت، ماشینهای نسل پنجم قادر به درک بیان، نوشته، و نیز قبول ورودیه‌های گرافیک خواهند بود. ژاپنیها هم‌اکنون سرگرم تحقیقات بسیاری در زمینه رابطهای هوشمند^{۱۰}، شامل پردازش زبانهای طبیعی، درک بیان، و درک تما ویر و ورودیه‌های گرافیک می‌باشند. به عنوان مثال، تحقیقات در زمینه درک بیان، شامل تجزیه و تحلیل امواج صوتی، تجزیه و تحلیل معانی بیانات، و تجزیه و تحلیل عملی (که با استفاده از استخراج موضوعها و تشخیص تائیدهای یک جمله و چیزهای مشابه دیگر، فهم جملات را امکان پذیر می‌آورد) می‌باشد. نهایتاً انتظار می‌رود که ماشین بتواند بدین پیوسته انسانها را با واژه‌نامه‌ای حاوی ۵۰۰۰۰ کلمه و دقت ۹۵ درصد نماید. همچنین انتظار می‌رود که سیستم تشخیص بیان قادر باشد یک ماشین تحریر را که با صوت کار می‌کند، کنترل نماید و به ژاپنی یا انگلیسی با استفاده کنندگان محاوره کند.

تجزیه و تحلیل متنها نیز توسط ژاپنیها به عنوان بخشی از پردازش زبانهای طبیعی به شمار می‌آید. البته آنها آگای دارند که روشهای تجزیه و تحلیل متنهای بزرگ با روشهای هموار کردن راه برای یک استفاده کننده جهت صحبت با ماشین، متفاوت است. این کار شامل برنامهریزی ترجمه خودکار^{۱۱} (ابتدا بین انگلیسی و ژاپنی) نیز می‌شود که دارای واژه‌نامه‌ای حاوی ۱۰۰،۰۰۰ کلمه خواهد بود. هدف، به دست آوردن دقتی برابر ۹۰٪ است (۱۰٪ بقیه، توسط انسان پردازش خواهد شد). عمل ترجمه، محصول یک سیستم مجتمع خواهد بود که در تمام قسمتها از جمله ترجمه کردن متن و چاپ مستندات ترجمه شده، شرکت می‌کند.

پردازش اشکال و تما ویر دارای اهمیت بسیار است. برنا مدریزی زبان می‌باشد. بالاخص

که در طراحی وساخت به کمک کامپیوتر، تجزیه و تحلیل مؤثر تما ویر فاشی و ما هوارهای، و نیز عکسهای پزشکی و امثال آن، کاربرد دارد. نهایتاً، انتظار می‌رود که سیستم درک تما ویر، تقریباً ۱۰۰،۰۰۰ تصویر را ذخیره کند. در این مورد نیز ما به تشخیص و درک اصوات، ژاپنیها به نتایج تحقیقات انجام شده خود در خلال سالهای ۱۹۷۰ در مورد پروژه ملی سیستم پردازش اطلاعات الکترونی^{۱۲} (PIPS) متکی خواهند بود.

سازمان اداری پروژه نیز ما به اهداف آن عجیب و غریب است. نسل پنجم، کنسرسیومی از هشت شرکت (فوجیتسو، هیتاچی، نیپون الکتریک، میتسوبیشی، اوکی، شارب، و توشیبا) و دو آزمایشگاه ملی (آزمایشگاه دولتی تلگراف و تلفن ژاپن در موشیو و آزمایشگاه الکترو-تکنیک MITI) است. در آوریل ۱۹۸۲، تقریباً ۴۰ محقق دستچین شده از هر یک از این شرکتها و آزمایشگاهها، زیر یک سقف در توکیو و در محل انستیتیو جدید نسل تازه تکنولوژی کامپیوتر-تر^{۱۳} (ICOT) دور هم جمع شدند. رئیس آنها کا زوهیرو فوجی^{۱۴}، مدیر و دانشمندی بسیار لایق از آزمایشگاه الکترونیک می‌باشد که در واقع، روح هوشمندی است که در پشت پسرده نسل پنجم قرار دارد.

محققین جوان در ICOT با هم دیگر کار می‌کنند و بطور آزاد، ایده‌های خود را می‌آید. آنها به طور هفتگی به شرکتها می‌روند و به خود می‌گویند و گزارش تحقیقات انجام شده خود را ارائه می‌کنند و به این ترتیب، شرکتها ذینفع از تحقیقات در حال انجام، آگاهی می‌یابند. البته تمام تحقیقات در ICOT انجام نخواهد گرفت و با شرکتها ذینفع نیز بسته به علاقه و امکان آنها، قراردادها منعقد می‌گردد. اما این قراردادها، مطلقاً تعهد نیست. نظارت ICOT خواهد بود و از آن عجیب‌ترین که ICOT ونه MITI، کنترل بودجه را بر عهده خواهند داشت. این روش کار فوجی جهت حصول اطمینان از انجام تحقیقات ابتکاری و منحرف شدن به راههای محافظه کارانه دیگر است.

در حال حاضر، تمام بودجه این پروژه را MITI تأمین می‌کند. با وجودی که یک پروژه ملی معمولاً توسط سرمایه‌گذاری مشترک دولت و بخش خصوصی انجام می‌گیرد، شرکتها ذینفع در پروژه نسل پنجم این گونه استدلال کردند که برای آنها مقصور نیست که پروژه‌ای با ریسک زیاد را هم از نظر محققین و هم از نظر مالی حمایت کنند. MITI این استدلال را قبول کرد و در حال حاضر، بودجه لازم برای سه سال اول را تأمین نموده است. بودجه سال دوم ICOT، برابر ۱۳/۶ میلیون دلار است که خیلی بیشتر از بودجه ۲ میلیون دلاری سال اول آن می‌باشد. در طول دوره ده ساله پروژه، با فرض این که شرکتها به کمک خود ادامه دهند، بودجه کلی احتمالاً به ۲۰۰ میلیون دلار خواهد رسید.



دنیا له روشهای اشکال زدایی...

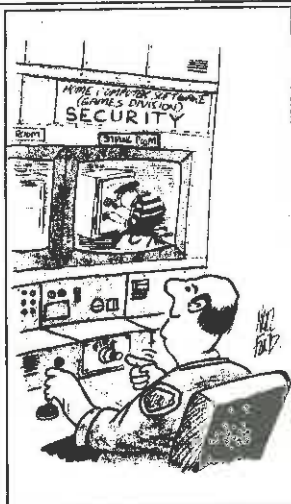
نور چراغ خیابان می‌گردد به این دلیل که در آنجا روشنائی بیشتر است! با وجودی که شانس هم در اشکال زدائی دخالت دارد، اما هیچ چیز جای تحلیل منطقی را نمی‌گیرد.

با ددا شتها و مرجع

* Debugging Techniques, Gregg

Williams, BYTE, June 1985.

- 1: Debugging
- 2: Structured Programming
- 3: Modularly
- 4: Monolithic
- 5: Optimize
- 6: Documents
- 7: Comments
- 8: Syntax Errors
- 9: Overflow
- 10: Utilities
- 11: Interactive
- 12: Type
- 13: Integer
- 14: Single-Precision Real
- 15: Dummy
- 16: Pointers
- 17: Register
- 18: Flowchart
- 19: Word Processor
- 20: Breakpoints
- 21: Snapshot
- 22: Subroutine
- 23: Flags
- 24: Formatted Dump
- 25: ASCII Characters
- 26: ROM
- 27: Testing
- 28: Preventive Debugging
- 29: Sleeping
- 30: Firewalling
- 31: Conditional Diagnostic Routines



سعی نکرده اند که سالهای دهه ۱۹۹۰ را پیش... یعنی کنند. در عوض آنها سعی کرده اند آینده‌ای ایده‌آل برای جامعه در نظر بگیرند و سپس، سیستمهای اطلاعاتی و کامپیوتری را که چنین آینده‌ای را می‌سازند، طراحی نمایند. به نظر من، از تصور آرمانی فوق بدور نیست که فرض کنیم نسل پنجم به منظور بدست آوردن جایگاهی ممتاز برای ژاپنیها در رشته کامپیوتر در سالهای ۱۹۹۰ نیز بوده است.

کا زوهیرو فوجی رئیس ICOT می‌گوید: در نتیجه توسعه زیا دوتخصصهای وسیع کامپیوتری کشورهایی که در حال حاضر از ژاپن جلو هستند قاندر نیستند. به دلیل وجود درختان زیاد، جنگل را رویت کنند! ولی خوشبختانه یا بدبختانه در ژاپن قاندر بودیم که منظره کلی را به چنگ آوریم.

آنها واقعا چنین منظره‌ای را به چنگ آورده اند. من بیا دمی آورم که تاکنون رشته کامپیوتر، چیزالها بیش، اصل، ویا حتی مهبجی به جهان عرضه کرده با دولی اینک ژاپنیها هر سه را عرضه کرده اند. آنها رشته کامپیوتر را از پرداختن به مسائل کوتاه مدت از قبیل سیستمهای عامل کمی بهتر، الگوریتمهای کمی سریعتر، پیشرفتهای مربوط به ما آینه و مسائل مشابه دیگر، به جلوه دل داده اند و به ما آورده اند که ما آنها به عنوان یک موجود، هنوز قاندریم که برای اهداف بزرگ ریسک کنیم. ما فقط به نیروی تصور، اراده، و جرات نیاز داریم.

با ددا شتها و مرجع

* Introduction to the Fifth Generation, Pamela McCorduck, Communication of the ACM, Sept. 1983.

- 1: Supercomputers
- 2: Robotics
- 3: Symbolic Inference Machines
- 4: Knowledge Information Processing Systems
- 5: Personal Workstation
- 6: Knowledge Base
- 7: Expert Systems
- 8: Logical Inference Per Second
- 9: Computer Aided Design (CAD)
- 10: Intelligent Interfaces
- 11: Automatic Translation
- 12: Pattern Information Processing Systems
- 13: Institute for new generation Computer Technology
- 14: Kazuhiro Fuchi
- 15: Edward Feigenbaum
- 16: Dense Chips
- 17: Tohru Moto-oka

"نسل پنجم"، افکار و تصورات متخصصین کامپیوتر در جهان را با خود جلب نموده و حتی برای افراد عادی اجتماع نیز جاذبه پیدا کرده است. به عنوان مثال، در ایالات متحده مقالات عمده‌ای در نیوزویک، تایم، بیزینس ویک، فورچون، و وال استریت جورنال (که همگی مجلات عمومی و معتبری می‌باشند)، در مورد نسل پنجم به چاپ رسیده است.

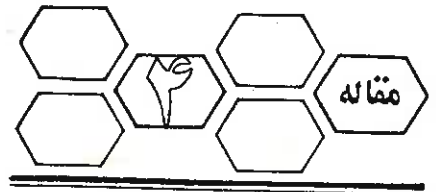
البته این بدان معنی نیست که همگان فکر می‌کنند که راهی که ژاپنیها در پیش گرفته اند، صحیح است. ایرادات فنی جدی و محکمی به کار آنها وارد شده است که با پدید آمدن پاسخ داده شود. از آن گذشته، حتی اگر فکر کنیم که راه ژاپنیها صحیح است، برای بعضی از ناظران رسیدن به این اهداف بلند - اگر آنها را گستاخانه بخوانیم - در یک دوره ۱۰ ساله بعید می‌نماید. (بعضی از ژاپنیها نیز این شک را دارند ولی به آن اهمیت نمی‌دهند. اهداف پروژه آقدر مهم است که تمدید مدت آن به نفع یا حتی یک دهه دیگر، غیر قابل قبول نیست.)

با توجه به ایرادات گرفته شده، نقل قولهای آورده شده در مورد عدم قابلیت ژاپنیها در ایجاد یا نوشتن نرم افزار، و امثال آن، شمار زیادی از متخصصان کامپیوتر اعلام کرده اند که "نسل پنجم"، غیر ممکن است. اما علیرغم اینسن اظهار نظرها، ادوار دفا یکن با م ۱۵ و من در کتاب "نسل پنجم" نوشته ایم که نسل پنجم نه فقط امکان پذیر، بلکه اجتناب ناپذیر است. کامپیوتر در آینده همان نقش مرکزیتی را در زندگی ما ایفا خواهد کرد که تا بحال توسط نوشته ها و کتابها ایفا شده است. استفاده از کامپیوتر ساده تر و حتی کار بر د آن وسیع تر خواهد شد. ولی از همه مهمتر این که کامپیوتر، هوشمند خواهد بود (به هر شکلی که هوشمند بودن را تعریف می‌کنید).

اما آنچه ذهن مردم را در مورد پروژه نسل پنجم جلب کرده است، نه قدرت میلیونها استنباط منطقی آنها در دقیقه، نه استفاده از تراشه های مترکم ۱۶، و نه قولها و وعده های درخشانی است که برای حل مسائل مربوط به پردازش موازی می‌دهند.

به نظر من، جذابیت نسل پنجم در تصویر آینده‌ای آرمانی است که ژاپنیها به همه ما وعده می‌دهند. آینده‌ای که در آن آگاهی و نه تنها اطلاعات محض، بلکه آگاهی در بالاترین کیفیت، شکل داده شده و قابل یافته برای رفع نیازهای فردی - در هر زمان، و هر کجا، با روشهای مفید، سریع، و قوی، در دسترس هر کس قرار خواهد داشت. آینده‌ای که در آن، آگاهی ثروت جدید ملتهاست.

آینده‌ای آرمانی به نوبتی آرمانی نیاز دارد. "توهرو موتو اوکا" ۱۷ پروفسور دانشگاه توکیو که رئیس کمیته تحقیق و مطالعه قبل از به وجود آمدن ICOT بود، به ما گفت: در برنامه ریزی برای نسل آینده کامپیوتر، ژاپنیها



مفاهیم اولیه سیستمهای عامل

(قسمت آخر)

ترجمه: عباس مصطفوی سبزواری
صنایع و خدمات کامپیوتری
(ایبیران)

یادداشت سردبیر

در شماره‌های گذشته ما هنام، اولین و دومین بخش از مقاله تفصیلی "مفاهیم اولیه سیستمهای عامل" که ترجمه بخش سیستمهای عامل دانشنامه Auerbach می‌باشد را خوانندگان گرامی گذشت. اینک توجهتان را به سومین و آخرین بخش از این مقاله جلب می‌نمایم.

کنترل ورود و خروج داده‌ها

انتقال داده‌ها عبارت است از جابجایی آنها بین حافظه اصلی و دستگاههای ورودی و خروجی. قبل از انتقال داده‌ها، باید محدودهای در حافظه اصلی به نام "میانگیر" در نظر گرفته شود.

روشهای تخصیص میانگیر در سیستمهای مختلف متنوعند. بعضی از سیستمها (مخصوصاً سیستمهای کوچک) تعدادی از زبانهای برنامه نویسی، تخصیص محلی از حافظه را برای نگهداری اطلاعات، جزء وظایف برنامه نویسنده را بر روی حساب می‌آورند. سه طریقه برای این کار مورد استفاده قرار می‌گیرند: "تک میانگیری"، "۲،" میانگیر" و "مخزن میانگیرها".

در روشهای اول و دوم، یک یا دونایه ثابت در حافظه اصلی به عنوان "میانگیر" برای هر پرونده در نظر گرفته می‌شود. این نوعی در ابتدای اجرای برنامه تخصیص یافته و صرف نظر از این که مورد استفاده قرار بگیرند یا خیر، تا انتهای اجرای برنامه در حافظه باقی می‌مانند. از طرف دیگر، در روش "مخزن میانگیرها" محل نسبتاً زیادی از حافظه به عنوان تعدادی میانگیر در نظر گرفته می‌شود. هر میانگیر می‌تواند در موقع لزوم برای پرونده‌های ورودی و یا خروجی مورد استفاده قرار بگیرد. در مواقعی که یک عمل ورودی و خروجی مورد نیاز باشد، استفاده کننده، میانگیر مورد نیاز را از مخزن اختیار کرده و آن را به پرونده مورد نظر تخصیص می‌دهد. سپس، بعد از تکمیل انتقال اطلاعات و پردازشهای بعدی، آن را به مخزن مسترد می‌دارد. بنابراین، این، چنانچه چند پرونده بطور همزمان فعال نباشند، کل فضای لازم جهت میانگیرها کاهش خواهد یافت.

پردازش ترمیمی^{۱۰}

این تکنیک به سیستم امکان می‌دهد که در مواقع بروز اشکال (که غالباً مربوط به برنامه نویسی نیست)، به نحوی آن را رفع نماید. این امر بسته به نوع خطا می‌تواند شامل اجرای مجدد برنامه، راه اندازی دوباره سیستم بطور کامل، و یا اجرای مجدد قسمتی خاص از برنامه می‌باشد. غالباً روش اجرایی اخیراً "مقابله نقاط و آغا ز مجدد" می‌نامند.

برای اجرای یک پردازش ترمیمی کامل، لازم است تا حداقل مکان شرایط محیطی به حالت قبل از شروع خطا برگردد. وظیفه ضبط و نگهداری این شرایط را روال "مقابله نقاط" برعهده دارد.

مقابله نقاط و آغا ز مجدد

این عمل عبارت است از ثبت اطلاعات لازم در مورد برنامه و محیط اجرای آن در یک حافظه فرعی به نحوی که برنامه می‌تواند از هر نقطه مورد نظر، دوباره آغا ز گردد. سیستمهای عامل محدود و کوچک، مخصوصاً سیستمهای پردازش ردیفی^{۱۲} غالباً برای تمام حافظه اصلی عمل مقابله نقاط را انجام می‌دهند ولی اغلب سیستمهای چندبرنامه‌ای، فقط بخش مشخصی از حافظه اصلی را برای این کار رزیت می‌نمایند. رکورد مقابله نقاط بر روی یک پرونده درون خطی^{۱۳} مخصوص (معمولاً) نوار مغناطیسی و یا حافظه‌های با دستیابی مستقیم ثبت می‌گردد. این رکوردها غالباً حاوی اطلاعات موجود در حافظه اصلی، محتویات تعدادی از شباهتهای مورد نظر، اطلاعات مربوط به وضعیت دستگاههای ورودی/خروجی، و اطلاعات مربوط به برخی پرونده‌های اساسی و مهم می‌باشد.

تشکیل رکوردهای مربوط به مقابله نقاط می‌تواند بر اساس درخواست واصله انجام پذیرد. این درخواست می‌تواند از طریق سیستم عامل، برنامه مورد اجرا، و یا متصدی سیستم صورت گیرد. بعضی از سیستمهای عامل، فقط برای برنامه‌های مشخصی عمل مقابله نقاط را پشتیبانی می‌نمایند. بسیاری از سیستمهای با اشتراک وقت^{۱۴} "این عمل را انجام نمی‌دهند ولی برخی از آنها به استفاده کننده اجازه می‌دهند تا وضعیت جاری برنامه‌اش را جهت بازگردن مجدد، ضبط نماید. این قابلیت را می‌توان به صورت یک مقابله نقاط محدود در نظر گرفت زیرا برنامه می‌تواند از آن نقطه مجدداً آغا ز گردد.

آغا ز مجدد، عبارت است از برگرداندن وضعیت سیستم درست به شرایط قبل از بروز اشکال. سه گونه اصلی قابلیت آغا ز مجدد که در اغلب سیستمهای عامل مشترک است، عبارتند از:

- اجرای مجدد از نقطه‌ای مشخص
- اجرای مجدد تکلیف^{۱۵}
- اجرای مجدد برنامه

در سیستمهای عامل پیچیده‌تر، از روشهای دیگری نظیر "میانگیری پویا"^{۱۶} و یا "میانگیری معادله‌ای"^{۱۷} استفاده می‌گردد.

در روش میانگیری پویا، اختصاص میانگیر به برنامه، در ضمن اجرا و در موقع نیاز توسط سیستم نامین می‌گردد. این روش کاملاً مشابه روش "مخزن میانگیرها" است با این اختلاف که در روش اخیر بجای برنامه نویسنده، خود سیستم مسئول تخصیص میانگیرهای مورد نیاز خواهد بود. در روش میانگیری معادله‌ای، نیازی به جابجایی اطلاعات بین میانگیر سیستم و محل‌های معرفی شده توسط استفاده کننده نیست. در این نوع پردازش، میانگیر سیستم از طریق پرونده‌های ورودی پرمی شود پس از آن بجای انتقال این اطلاعات به نامایه پردازشی استفاده کننده، سیستم به سادگی محدوده‌های میانگیر را بسا برنامه کاربردی معادله می‌نماید. همینطور موقعی که میانگیر استفاده کننده پر شود، نشانی آن با نشانی یک میانگیر خالی عوض می‌شود و سپس اجرای برنامه دنبال می‌گردد. در این روش ضمن اعمال کنترل بیشتر سیستم بر روی میانگیرها، مقدار عملیات سر بار سیستم نیز کاهش پیدا می‌کند.

موقعی که مسأله انتقال اطلاعات وجود دارد، ممکن است لازم باشد سیستم با استفاده از روالهای خاصی عمل ترجمه اطلاعات را نیز انجام دهد. این نیز مخصوصاً در مورد سیستمهای مخابراتی که در آنها کدهای ورودی با کدهای داخلی سیستم تفاوت زیادی دارد، مصداق پیدا می‌کند. وقتی مسأله ترجمه اطلاعات مطرح باشد، این وظیفه توسط فاصله سیستم و دستگاههای ورودی/خروجی انجام می‌گیرد و این اختلالات موجود در ساختار کدها غالباً برای برنامه نویسنده کاربردی مسأله‌ای ایجاد نمی‌شود و چگونگی این تبدیلات به استفاده کننده ارتباطی پیدا نمی‌کند. معذرت در تعدادی از سیستمهای کوچکتر که مجهز به این امکانات نیستند، عمل ترجمه کدها به برنامه نویسنده محول می‌گردد.

در اغلب سیستمهای اشتراک وقت یا پردازش از دور^{۱۸}، پایانه به منزله تنها سیستم ورودی و خروجی عمل می‌کند و این برای برنامه نویسنده امکان ارتباط مستقیم با متصدی سیستم از طریق صفحه کلید را اختیار دارد.

در تعداد زیادی از سیستمها نیز امکان ارتباط بین استفاده کننده با یکدیگر و یا ارتباط استفاده کننده با متصدی سیستم وجود دارد. در این گونه سیستمها، متصدی امکان ارتباط با یک، چند، و یا تمامی استفاده کننده‌ها را خواهد داشت و مشترکین سیستم نیز برای ارتباط با یکدیگر و یا متصدی، از امکانات مشابهی برخوردار خواهند بود. در تعدادی از سیستمها نیز استفاده کننده‌ها می‌توانند پیغام خاصی را برای دیگران در محل خاصی ذخیره کنند. در این صورت در آینده هرگاه طرف مورد نظر با سیستم ارتباط برقرار کنند، پیغامهای مربوط به خود را دریافت خواهند نمود.

محاسباتی و خطا در داده‌ها)، راه‌ساز و از طریق خودبرنامه‌ها اقدام نمودن خطاهای که بروز آنها به‌تأثیر می‌تواند سیستم مربوط باشد، به‌وسیله خودسیستم مورد بررسی قرار می‌گیرند. از جمله این خطاهای می‌توان خطاهای مربوط به حافظه را نام برد. وجود خطاهای جدی معمولاً به‌خاطر یافتن غیرطبیعی برنامه می‌تواند باشد.

خدمات زمانی

اغلب سیستم‌ها مجهز به یک دستگاه زمان‌سنج داخلی هستند که به‌وسیله آن، خدمات زمانی لازم را به برنامه‌های کاربردی ارائه می‌دهند. این خدمات از دو نوعند: زمان بندی فواصل ۲۴ و زمان واقعی.

زمان بندی فواصل برای تعیین برنامه‌ها به مدت معین و یا شروع مجدد آنها پس از گذشت فاصله زمانی مشخص، مورد استفاده قرار می‌گیرد. این روش می‌تواند برای زمان بندی بازرسی‌های متناوب به منظور اطلاع از وجود شرایط معین و یا به منظور جلوگیری از به وجود آمدن حلقه‌های بی‌انتها در برنامه‌ها، کاملاً مفید واقع شود.

از زمان واقعی غالباً به منظور در اختیار قرار دادن تاریخ وساعت روزانه به برنامه‌های در حال اجرا استفاده می‌گردد. همچنین می‌توان از آن برای زمان بندی وقفه‌ها در زمان‌های معین (بجای فواصل مشخص) استفاده جست و بنا بر این، زمان‌های شروع و ختم برنامه را برای ساعت مشخصی تنظیم نمود.

خدمات آزمایش و اشکال زدایی

تسهیلات موجود برای آزمایش و اشکال زدایی برنامه‌های کاربردی بسیار متنوعند. بعضی از روشهای پشتیبانی شامل مجموعه‌ای مستقل و کامل از برنامه‌های آزمایش و اشکال زدایی می‌باشد که به همراه سیستم عامل، عملیات لازم را انجام می‌دهد. در حالی که در تعداد دیگری از سیستم‌ها این برنامه‌ها به‌عنوان جزئی از سیستم عامل محسوب می‌گردند.

در اغلب سیستم‌ها از نقطه نظراتی که در گذشته در انواع حالت عملیاتی می‌تواند وجود داشته باشد، حالت عادی و حالت آزمایشی، موقعی که در حالت عادی اشکالی به وجود آید، برنامه به‌طور غیرعادی خاتمه می‌یابد در حالی که در حالت آزمایشی، با پیش آمدن هر نوع خطا، برنامه‌ها اشکال زدایی خاصی به منظور جمع آوری اطلاعات لازم در مورد آن خطا، فرا خوانده می‌شوند. مزیت حالت اخیر در این است که خطای حاصله قبل از خاتمه یافتن برنامه به‌دقت مورد بررسی و مطالعه قرار می‌گیرد.

عمل روبرواری از حافظه ۲۳ به‌استفاده کننده مکان می‌دهد که محتویات حافظه را قسمتی از آن را به‌صورت چاپ شده در اختیار داشته باشد. تقاضای روبرواری را می‌توان حین اجرای برنامه مه‌ویا پس از اتمام آن اعلام نمود. معمولاً ورق بزنید

اغلب سیستم‌ها مجهز به روال‌های جهت عیب‌یابی سخت‌افزاری می‌باشند. این اشکالات ممکن است مربوط به موارد زیر باشند:

- اشکال عدم توازن بینت ۴۰ در کلمات حافظه
- اشکالات مربوط به پردازنده
- اشکالات مربوط به کنترل عملیات ورودی و خروجی

• اشکالات منبع تغذیه
موارد فوق در عین حال که نمونه‌های استاندارد به نظر می‌رسند ولی ممکن است برای تعدادی از سیستم‌ها کافی نباشد.

به‌عنوان مثال، در سیستم‌های پیشرفته‌تر ممکن است موقعی که اشکالی در سیستم پیدا می‌شود، سیستم با استفاده از روال‌های خاصی سعی کند دلیل به‌وجود آمدن آن واحتمالاً درجه آسیب‌رسانی آن را بررسی نماید. در مواردی که اشکال جزئی باشد، سیستم با اجرای مجدد عملیات سعی می‌کند موفقیت یا ناموفقیت آن اشکال را تعیین کند. در بعضی موارد برای بررسی اشکالات ناموفق، دخالت متمدی سیستم ضروری است.

در صورتی که اشکال جزئی و گذرا باشد سیستم سعی خواهد نمود که با تحت نظر گرفتن اجزای مختلف و ثبت اشکالات مربوط به آنها، اطلاعات بیشتری را در خصوص عنصر مولد اشکال فراهم آورد. بررسی اطلاعات فراهم شده می‌تواند منجر به شناسایی عامل اصلی اشکال گردد و با اصلاح آن از مایعات مهمتر جلوگیری به‌عمل آید.

امروزه استفاده از پردازنده‌های جداگانه خاص و مستقل از پردازنده اصلی به منظور اعمال روال‌های خطایابی در بین سازندگان سخت‌افزار معمول گردیده است. با استفاده از این امکانات می‌توان از راه دور نیز آزمون‌های سنجش خطا را انجام داد. این روش نخستین بار به‌وسیله کمپانی‌های امدال ۲ و آیبیام معمول گردید. در سیستم‌های پیچیده‌تر این قابلیت وجود دارد که قطعات خراب شده را به‌طور پویا با قطعات جدیدی تعویض نمود. در این روش، قطعه معیوب از سیستم جدا شده و قطعه پشتیبان جایگزین آن می‌گردد. در صورتی که قطعه‌ای به‌عنوان ذخیره پیش‌بینی نشده باشد، در بعضی سیستم‌ها تکنولوژی "Fail-Soft" بکار گرفته می‌شود که طی آن، قطعه خراب به‌عنوان "معیوب" از مدار خارج می‌گردد و سیستم با سخت‌افزار کمتری بکار خود ادامه می‌دهد. از آنجا که اجرای این امر از کارآئی سیستم می‌کاهد، برای جبران آن، برنامه‌های کم‌اولویت یا اصولاً اجرا نمی‌گردند و یا تعداد دفعات سرویس‌دهی به آنها تقلیل می‌یابد.

تقریباً تمام سیستم‌های عامل، تسهیلاتی را برای تشخیص خطاهای مربوط به برنامه‌نویسی در اختیار دارند. البته نوع عکس‌العمل لازم برای رفع خطا بستگی به اهمیت خطای تولید شده دارد. معمولاً استفاده کننده مجاز است که برای تشخیص بعضی از خطاهای جزئی (نظیر خطاهای

در حالت اول، شرایط اجرای مجدد درست به حالت قبل از آخرین مقابله نقاط برمی‌گردد و برنامه از نشانه‌ای که در آنجا تعیین شده، اجرا می‌گردد. در حالت دوم، برنامه از ابتدای تکلیف مشخص شده و در حالت سوم، برنامه از ابتدای آغاز می‌شود. بجز در حالتی که عمل مقابله نقاط به‌وسیله سیستم و به منظور ارسال داده‌ها از حافظه اصلی به حافظه جانبی و یا بازگرداندن داده‌ها به حافظه اصلی انجام می‌گیرد، لازم است دستگاه‌های ورودی و خروجی نیز برگردان شوند. میزان بازگرداندن وضعیت این دستگاه‌ها متغیر است ولی در عین حال بیشتر سیستم‌ها حداقل قادرند وضعیت دستگاه‌های ردیفی را به‌حالت اولیه بازگردانند. تعداد دیگری از سیستم‌ها می‌توانند با برگرداندن نوک خواندن/نوشتن در دستگاه‌های با دستیابی مستقیم، به‌نوعی اطلاعات از بین رفته در خطوط ارتباطی را نیز بازیابی و بازیابی نمایند. در صورتی که عمل برگردان دستگاه‌ها توسط سیستم انجام نگردد، لازم است به‌وسیله برنامه‌های کاربردی یا متمدی سیستم انجام پذیرد.

پشتیبانی یا برنامه‌های راه‌دور

تمام سیستم‌های با اشتراک وقت، سیستم‌های پردازش تراکنش ۱۶، و سیستم‌های دسته‌ای ۱۷ با امکان ارسال برنامه‌ها از راه‌دور، روش‌های کنترلی خاصی را برای برنامه‌های راه‌دور اعمال می‌نمایند. این روش‌های کنترلی، عموماً به‌وسیله سیستم عامل اصلی و از طریق کنترل کننده‌های ارتباطی انجام می‌گیرد، منتها در سیستم‌های پیچیده‌تر، انجام آن به پردازنده درون خود یا برنامه محول می‌گردد. درجه‌ای که این کنترل‌ها بسته به نوع نرم‌افزار موجود در این پردازنده‌ها متفاوت است. این سیستم‌ها همچنین بطور نرم‌افزاری، ارسال عادی و بدون اشکال داده‌ها را نیز درون خطوط ارتباطی کنترل می‌کنند. بعضی از سیستم‌ها فقط در محدوده سازمان‌ها و یا برنامه‌های عمل می‌کنند و تعدادی دیگر قادرند سرعت‌های متفاوتی را پذیرا بوده و عملیات‌های نظیر "صفحه بندی ۱۸" و "فشرده سازی داده‌ها ۱۹" را نیز در برنامه‌های مجهز به لامپهای اشعه کاتدیک (CRT) انجام دهند.

پشتیبانی پردازشی

اغلب سیستم‌های عامل قادرند حداقل یکی از خدمات پشتیبانی مشروحه زیر (و معمولاً تمام آنها را) به‌عنوان پشتیبانی پردازشی انجام دهند:

- تشخیص خطا
- خدمات زمانی
- آزمایش و اشکال زدایی

تشخیص خطا

تشخیص خطا عبارت است از عیب‌یابی اشکالات سخت‌افزاری یا برنامه‌نویسی و بکارگیری روال‌های جهت رفع آنها.

این اطلاعات به خوبی قابل بندی ۲۴ شده اند تا حد اکثر اطلاعات لازم را راجع به برنامۀ مداخلت قرار دهند.

تسهیلات ردیابی ۲۵ می‌تواند مسیر اجرای برنامۀ تعقیب نموده و در حالت کلی، اطلاعات مربوط به اجرای هر دستورالعمل از قبیل نشانی، متغیرهای تاء، تیرپذیرفته، و محتویات آنها قبل و بعد از اجرای آن دستورالعمل را به دقت ثبت نماید. با وجود این که انواع گوناگونی از ردیابی وجود دارد، ولی اطلاعات ارائه شده توسط هر یک از آنها مشابیه بوده و فقط اختلاف آنها در مناسبت اعلام این اطلاعات می‌باشد. برخی از انواع متداول ردیابیها عبارتند از: ردیابی کامل دستورالعمل، ردیابی دستورالعمل در محدوده نشانیهای مشخص، ردیابی — روز وقفه‌ها، و ردیابی دستورالعملهایی که محتویات کلمات مشخصی را در حافظه صلاح می‌نمایند. ردیابی بیهاگی نیز وجود دارد که به موجب آنها می‌توان محتویات نواحی خاصی از حافظه را در صورت تغییر یافتن، نمایش داد.

در سیستمهای با اشتراک وقت و فعل و انفعالی ۲۶ استفاده از تسهیلات زمانی و شکل زداشی بسیار توسعه یافته است. استفاده کنندگان از آنها می‌توانند عناوین از برنامۀها را نظیر دستورالعملها، مقادیر عددی، و اطلاعات کد شده را به سادگی تغییر دهند، نقاط انفعالی ۲۷ را در برنامۀ مشخص سازند، و با لافرها اجرای برنامۀها را با تغییر مسیر اجرای شکل دستورالعملها کنترل نمایند.

ضبط وقایع و حسابداری ۲۸

اغلب سیستمها مجهز به امکاناتی برای ثبت وضبط وقایع و جمع آوری اطلاعات آماری لازم در مورد استفاده کنندگان از کامپیوتر و میسزانی استفاده از منابع موجود می‌باشد. معمولاً در خاتمه یک برنامۀ، اطلاعات زیر جمع آوری و ثبت می‌گردند:

ه اسم برنامۀ و موقعیت خاتمه یافتن آن
ه تعداد رکوردهای اضافه (یا حذف) شده به (از)
ه هر یک از پرونده‌های دائمی
ه زمان مصرفی پردازنده
ه زمان مصرف شده توسط هر دستگاه یا کانال ارتباطی

ه تعداد خطای چاپ شده
ه تعداد کارتهای پانچ شده
ه تعداد رکوردهای نوشته شده در واحدهای خروجی سیستم

مدیریت داده‌ها ۲۹

مدیریت داده‌ها شامل تسهیلات مدیریتی پرونده‌ها، پشتیبانی عملیات ورودی/خروجی، و سیستم مدیریت داده‌ها می‌باشد.

یک پرونده را می‌توان به صورت مجموعه‌ای از رکوردها تعریف نمود. تسهیلات مدیریت پرونده‌ها عبارت است از مجموعه برنامۀهایی که به منظور پشتیبانی پرونده‌ها (و نه رکورد داخل آنها)

به وسیله سیستم ارائه می‌گردد. پشتیبانی رکوردها غالباً به وسیله دو قسمت دیگر (تسهیلات پشتیبانی ورودی/خروجی و مدیریت داده‌ها) انجام می‌شود.

تسهیلات پشتیبانی ورودی/خروجی به برنامۀ اجازه می‌دهد تا به رکوردهای داخل پرونده‌ها دسترسی پیدا نموده و آنها را پردازش نماید. با بهره‌گیری از این امکانات می‌توان عملیات ورودی/خروجی را از طریق برنامۀها بکار گرفتن دستورالعملهای ماکرو ۳۰ انجام داد و نیاز برنامۀ نویسنده را به درگیری مستقیم با مسائل موجود در این راه و پردازش به نکات غالباً فنی آن، از بین برد.

سیستم مدیریت داده‌ها عبارت است از مجموعه روالهایی که به منظور ایجاد، نگهداری، و بازرسی مجموعه‌ای از اطلاعات وابسته به هم و کاملاً سازمان یافته که به نامهای یک داده‌ها ۳۱ خوانده می‌شود، بکار می‌رود. سیستم مدیریت داده‌ها می‌تواند از طریق منابع ورودی مختلف، پرونده‌ای را ایجاد و سپس با انجام عملیاتی از قبیل اضافه، حذف، و اصلاح، آن را نگهداری نماید. سیستم مدیریت داده‌ها همچنین می‌تواند پرونده‌های جدیدی را ایجاد کند، سازمان دهی پرونده‌های موجود را تغییر دهد ۳۲، محتویات آنها را در هم ادغام ۳۳ نماید، و اطلاعات موجود در آنها را در اختیار درخواست کننده قرار دهد. به علاوه، می‌تواند اطلاعات موجود را محاسبه و گزارشاتی با شکل دلخواه سیستم یا استفاده کننده، ارائه نماید. سیستم مدیریت داده‌ها می‌تواند هم برای کاربر سیستمهای دسته‌ای و هم در سیستمهای فعل و انفعالی، طراحی گردد.

سیستم مدیریت داده‌ها می‌تواند به استفاده کنندگانی که تجربه برنامۀ نویسی کمی دارند نیز امکاناتی را به منظور دسترسی به رکوردهای یک پرونده و نمایش آنها ارائه نماید. در حالت کلی، سیستمهای مدیریت داده‌ها از ملحقات سیستمهای عامل هستند و مشخصات آنها تابع معماری سیستم مورد نظر می‌باشد. طراحی داخلی سیستم مدیریت داده‌ها می‌تواند به گونه‌ای باشد که از دیگر امکانات داخلی سیستم عامل به منظور بهره‌وری بیشتر استفاده نماید.

تسهیلات مدیریت پرونده‌ها

تسهیلات مدیریت پرونده‌ها به منظور کنترل پرونده‌های مختلف موجود در سیستم طراحی شده اند و اعمالی که انجام می‌دهند عبارت است از یافتن محل قرار گرفتن پرونده‌ها، اجازه دادن یا منع کردن استفاده کننده از دستیابی به پرونده، و تهیه پرونده پشتیبان ۳۴ جهت بازیابی محتویات پرونده‌ها در صورت بروز اشکال.

در اغلب سیستمهای موجود، پرونده‌های اطلاعات دائمی از طریق برچسبهای ۳۵ شناسایی که توسط سیستم و یا استفاده کننده مشخص می‌شود شناخته می‌شوند. اطلاعات موجود در برچسبها شامل

نام پرونده، شماره ویرایش پرونده ۳۶، نام و شماره حساب صاحب پرونده، و احتمالاً کد حفاظتی پرونده می‌باشد. در بعضی از سیستمها به منظور ایجاد سطوح مختلف نمایندگی ۳۷ استفاده از ترکیبی از چند نام برای نام پرونده مجاز است.

در سیستمهای دسته‌ای و اشتراک و قسقی بزرگتر، محل تمام پرونده‌های دائمی شناخته شده توسط سیستم، در یک فهرست راهنما نگهداری می‌شود و به هنگام پردازش، جهت یافتن پرونده، این فهرست مورد جستجو قرار می‌گیرد. در سیستمهایی که مجهز به این فهرست نیستند برای پیدا کردن پرونده‌های درون خطی (متمل به سیستم) لازم است تمام برچسبها بطور ردیفی و متوالی مورد جستجو قرار گیرند و برای پرونده‌های بیرون خطی ۳۸ (غیر متمل) نیز دخالت متمادی سیستم برای ارائه نمودن آنها ضروری می‌باشد.

تسهیلات پشتیبانی ورودی/خروجی

پشتیبانی ورودی/خروجی در غالب سیستمها در دو سطح فیزیکی و منطقی اعمال می‌گردد. پشتیبانی فیزیکی که پشتیبانی اصلی است، شامل روالهایی است که به وسیله آنها انتقال اصلی اطلاعات انجام می‌گیرد و اطلاعات منتقل شده در دسترس برنامۀها قرار می‌گیرند.

در پشتیبانی منطقی، دسترسی به اطلاعات بدون توجه به ساختار فیزیکی آن مورد نظر است و در نتیجه، این حالت مانند واسطه‌ای است بین عملیات ورودی/خروجی که توسط استفاده کننده انجام می‌شود و خدمات ورودی/خروجی فیزیکی سیستم.

ساختار پرونده‌ها

معمولترین ساختار برای پرونده‌ها، ترتیبی، سلسله‌مراتبی، نمایه‌دار، لیستی، و به لافرها ۳۹ می‌باشد.

در ساختار ترتیبی، عناوین پرونده‌ها بطور ردیفی در داخل پرونده قرار می‌گیرند. در ساختار سلسله‌مراتبی، عناوین اطلاعاتی طبقه بندی شده بر اساس این طبقه بندی در پرونده ذخیره می‌شوند. در ساختار نمایه‌دار، قسمتی از پرونده برای قراردادن کلیدهایی که نشانگر محل اطلاعات در بخش دیگری از پرونده می‌باشد، در نظر گرفته می‌شود. در ساختار لیستی، هر عنصر موجود در پرونده، حاوی نشانی عنصر بعدی می‌باشد. ساختار حلقوی عیناً شبیه حالت لیستی است منتها در آن، همواره آخرین عنصر موجود در پرونده حاوی نشانی نخستین عنصر خواهد بود.

ساختار پرونده‌ها بستگی به روش دستیابی به آن که بعداً مورد بررسی قرار خواهد گرفت، اشتباه شود. روش دستیابی، مربوط به مجموعه‌ای از روالهاست که با استفاده از آنها برنامۀ نویسنده می‌تواند رکوردهای اطلاعاتی را در حافظه‌های جانبی ذخیره و بازیابی نماید. در حالی که ساختار پرونده به چگونگی سازماندهی پرونده



حافظه‌های فرعی ذخیره شده اند موجود است که از آن جمله می‌توان به منتقل نمودن اطلاعات از نوار به چاکریا از دیسک به چاکریا را نام برد. از آنجا که این عملیات برای کنترل چشمی محتویات مذکور انجام می‌شود، معمولاً اطلاعات چاپ شده، شامل قالب بندیهای لازم برای تسهیل کار استفا ده کننده نیز می‌باشند. قالب بندیهای مذکور، اگرچه غالباً به وسیله سیستم انجام می‌گیرند، ولی در تعدادی از سیستم‌ها می‌توان با استفا ده زبانه‌ها را مترهای لازم انجام این کار را به عهده استفا ده کننده واگذار نمود.

پشتیبانی دستگاه‌های جانبی عبارت است از موقعیت یابی^{۴۴}، تبدیل محیط داده‌ها، ویرایش داده‌ها، و پشتیبانی پرونده‌ها.

موقعیت یابی برای انجام کارهای نظیر راندن به عقب^{۴۵}، راندن به جلو^{۴۶}، برگرداندن نوار مغناطیسی، یافتن محل یک پرونده بر روی نوار مغناطیسی یا حافظه‌های با دستبای مستقیم، و به آخره یافتن رکوردی در داخل یک پرونده (از جمله رکورد انتهای پرونده) بکار می‌رود.

تبدیل محیط داده‌ها اجازت می‌دهد که داده‌ها از یک محیط اطلاعاتی به محیط دیگر منتقل شوند. معمولاً این انتقال می‌تواند بین هردو محیط اطلاعاتی که از نظر سیستم قابل پشتیبانی باشند، انجام گیرد. به عنوان مثال می‌توان از انتقال کارت به نوار، نوار به کارت، و نوار به دیسک نام برد. انتقال اطلاعات از یک محیط به محیط دیگر می‌تواند با تغییر قالب اطلاعات و احتمالاً ترجمه آنها همراه باشد، مخصوصاً در مواردی که محیط دریافت کننده اطلاعات نتواند قالب اولیه اطلاعات را پشتیبانی کند. برخلاف تسهیلات مربوط به نمایش اطلاعات، چگونگی این انتقال از نظر استفا ده کننده مشهود نبوده و عمل انتقال اطلاعات در محدوده فضای بین دو محیط انجام می‌گردد. به عنوان مثال، انتقال کارت به نوار عبارت است از کپی کردن محتویات کارت به صورت رکوردهای متوالی ۸۰ بوبه‌ای بر روی نوار مغناطیسی.

یکی از موارد قابل ذکر در استفا ده از این تسهیلات عبارت است از ذخیره محتویات حافظه فرعی و بار کردن دوباره آن. این عمل معمولاً به منظور تهیه پرونده‌های پشتیبان و فشرده سازی^{۴۷} حافظه صورت می‌گیرد.

امکانات بیشتری در زمینه داده گردانی در اغلب سیستم‌ها موجود است که در همه آنها بجای توجه به دستگاه‌های خاص، پشتیبانی کلی پرونده‌ها مورد توجه قرار گرفته است. برای مثال امکانات مربوط به ویرایش داده‌ها به استفا ده کننده مکان می‌دهد که پرونده مشخصی را با زبانی و اشکالات محتمل موجود در آن را کشف نماید (به عنوان مثال، اشکالاتی از قبیل خارج از ردیف بودن) و یا محتویات دو پرونده مختلف بر روی دو دستگاه جانبی متفاوت را با یکدیگر مقایسه نماید.

بقیه در صفحه ۲۲

حالی که از نظر کلاسیک این اطلاعات، پرونده مشخصی را به وجود نمی‌آورند، ولی تعداد زیادی از سیستم‌ها مجهز به روش‌های برای دستیابی و پردازش پیغام‌های قالب بندی شده‌ای که در میان نگیرهای سیستم انباشته می‌شوند، می‌باشند. این روش‌ها قادرند کلیه ارتباطات انجام شده بین پایانه‌ها و کامپیوتر مرکزی را پشتیبانی نمایند. اعمالی که انجام می‌گیرند عبارتند از: تخصیص حافظه برای انباشتن پیغام‌ها، بررسی پایانه‌ها به منظور تشخیص این که آیا پیغامی برای ارسال دارند، تعیین اولویت پیغام‌ها، تحلیل عناوین پیغام‌ها به منظور تشخیص مسیر ارسال آنها، شماره ترتیب پیغام‌های دریافتی، کنترل خطای ارسال، و تهیه صفی از پیغام‌های دریافتی و ارسال.

بلوکی کردن^{۴۸} و احوالت بلوکی در آوردن^{۴۹} داده‌ها

بلوکی کردن، عبارت از ترکیب دوبی چند رکورد به منظور ایجاد یک بلوک فیزیکی از اطلاعات و احوالت بلوکی در آوردن، عکس آن است. طول رکوردها می‌تواند ثابت، متغیر، و یا تعریف نشده باشد. رکوردهای بلوک بندی شده با وجود این که به میان نگیرهای ورودی/خروجی بزرگتری نیاز دارند، ولی میزان استفا ده مؤثر از حافظه را بالا می‌برند.

در سیستم‌های عاملی که فقط قابلیت پشتیبانی فیزیکی ورودی/خروجی را دارند عمل بلوکی کردن و احوالت بلوکی در آوردن رکوردها با یستی به وسیله استفا ده کننده انجام شود. در سیستم‌های مجهز به پشتیبانی منطقی، به استفا ده کننده مکان داده می‌شود که بدون توجه به ساختار فیزیکی بلوک‌ها، با تک تک رکوردها به صورت متفرک کار کنند. بلوکی کردن و احوالت بلوکی در آوردن معمولاً با انتقال اطلاعات بین میان نگیرهای ورودی/خروجی و محدوده پردازشی استفا ده کننده، و یا با استفا ده از راه‌گرهای خاص جهت جدا کردن و پردازش اطلاعات در داخل میان نگیرها، انجام می‌گیرد.

روالهای سودمند جهت داده گردانی^{۴۲}

برنامه‌های سودمند داده گردانی، از روالهای نسبتاً ساده‌ای تشکیل شده اند که عمدتاً به وسیله فراخوانی برنامه، کارتهای کنترلی سیستم، و یا درخواست متعدي سیستم (پرا تور) فعال گشته و مورد استفا ده قرار می‌گیرند. این روالها به ندرت با برنامه‌های در حال اجرا بطور مستقیم پیوند می‌یابند بلکه غالباً به صورت یک کام مستقل در برنامه‌های چندگانه^{۴۳} ظاهر می‌شوند. بطور کلی می‌توان تسهیلات فوق الذکر را در دو دسته مستقل تقسیم بندی نمود: نمایش، و پشتیبانی دستگاه‌های جانبی.

در اغلب سیستم‌ها امکاناتی برای نمایش محتویات حافظه اصلی، فهرستهای راهنمای سیستم، وجد اول وجود دارد، به علاوه معمولاً تسهیلات دیگری نیز در مورد داده‌های که در

ونحوه قرار گرفتن اطلاعات در داخل آن اشاره می‌کند. روالهای مربوط به مدیریت داده‌ها، تسهیلات لازم برای پیدا ده سازی هر نوع ساختمان پرونده، صرف نظر از روشهای دستیابی موجود را فراهم می‌آورند.

تخصیص حافظه برای پرونده

تکنیکهای موجود برای تخصیص محل لازم برای پرونده‌ها، از سیستمی به سیستم دیگر، متفاوت است. در سیستم‌های پیچیده، حافظه مربوطه بطور پویا تخصیص داده و آزاد می‌شود. در حالی که در تعداد زیادی از سیستم‌ها اتفا ذ تصمیم در این باره به استفا ده کننده واگذار می‌گردد و در این حالت، امور جنبی قضیه نظیر حفاظت پرونده‌ها از تخریبهای ناخواسته نیز با یستی تحت کنترل استفا ده کننده باشد.

روشهای دستیابی به پرونده

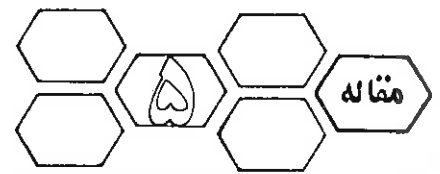
متمد و لبرترین روشهای دستیابی به پرونده‌ها عبارتند از: ترتیبی، نمایی، کلیدی، تصادفی، و سازماندهیهای مختلعاتی. هر روش می‌تواند امکانات ذخیره و یا یستی اطلاعات را برای هردو حالت فیزیکی و منطقی فراهم آورد. در روش دستیابی ترتیبی، خواندن و نوشتن اطلاعات در پرونده به طریق متوالی و ردیفی انجام می‌گیرد. دستیابی ترتیبی می‌تواند در مورد تمام حافظه‌های جانبی اعمال گردد ولی برای بعضی از آنها نظیر نوارهای مغناطیسی، استفا ده از این روش الزامی است.

روش دستیابی نمایی برای پرونده‌های بکار می‌رود که در آنها علاوه بر اطلاعات اصلی، نوعی فهرست از اطلاعات رکوردهای پرونده وجود داشته باشد. در این صورت برای پیدا کردن رکورد مشخص، بجای جستجو در تمام پرونده می‌توان فقط فهرست موجود را بررسی نمود. برای بکار گرفتن این روش دستیابی لازم است از شاخه‌های با دستبای سریع مانند دیسک استفا ده شود.

در روش دستیابی کلیدی، حوزه مشخصی از رکورد که در واقع معرف منحصر به فرد رکورد نیز هست، به عنوان کلید جستجو تعیین می‌شود. این روش برای آن دسته از حافظه‌های جانبی مفید است که دستورالعملهای جستجو به صورت سخت افزاری پیدا ده شده باشند. در این روش بکار جستجوی رکوردی مشخص، فقط کلیدهای رکوردها مورد جستجو قرار گرفته و از تلاف وقت پردازنده در جستجوی تمام پرونده جلوگیری می‌گردد.

در روش دستیابی تصادفی، عملیات خواندن و نوشتن رکوردها بدون توجه به ترتیب فیزیکی آنها روی پرونده صورت می‌گیرد. بنا بر این، رکوردهای ذخیره شده با یستی دارای نوعی کشش خاصی باشند با استفا ده از آن بتوان محل رکورد را مشخص نمود. معمولاً با استفا ده از الگوریتمهای مشخصی این کده نشانی شروع رکورد بر روی حافظه جانبی تبدیل می‌گردد.

اطلاعات مختلعاتی معمولاً از رشته‌های از علامات با طولهای متغیر تشکیل می‌گردند. در



"آشنایی با پیش‌تازان علم کامپیوتر"



هربرت الکساندر سایمون^۱
بنیان‌گذار رشته "هوش مصنوعی"

ترجمه: * دکتر پروین زامر

هربرت الکساندر سایمون که در سال ۱۹۱۶ در شهر میلواکی از ایالت ویسکانسین آمریکا به دنیا آمد، به خاطر کارهایش در زمینه هوش مصنوعی، شهرت به‌سزائی در رشته علوم کامپیوتر دارد. وی که بی‌شک یکی از بزرگ‌ترین نوابع علمی جهان معاصر می‌باشد، نه تنها در رشته علوم کامپیوتر بلکه در بسیاری از رشته‌های دیگر علوم نیز فردی شناخته‌شده و مورد احترام به‌شمار می‌آید.

سایمون در سال ۱۹۴۳، موفق به اخذ درجه دکتری علوم سیاسی از دانشگاه شیکاگو گشت. نظریات علمی و ارزشمند وی نه تنها سهم عمده و اساسی در پیشبرد این رشته از علوم داشته بلکه در زمینه‌های تحقیقاتی متعدد دیگری نظیر مطالعه سازمانها^۲، اداره جامعه، اقتصاد - سنجی، مدیریت، تحقیق در عملیات^۳، پایه‌های فلسفی علّیت و مکانیک نیوتن، طبیعت‌شناسی، علمی، و همچنین روانشناسی و علوم کامپیوتر نیز راهگشا و تعیین‌کننده بوده است.

با وجودی که احاطه علمی سایمون بر این تعداد از رشته‌های مختلف و متنوع علوم، اعجاب‌انگیز است ولی باید خاطرنشان ساخت که تمام کارهای علمی و تحقیقاتی وی، ریشه و موضوع مشترک و واحد دارند: "درک طبیعت رفتار منطقی و عقلانی در انسانها". اولین کتاب سایمون که در سال ۱۹۴۷ منتشر گردید^۴، تحلیلی بود در مورد رفتار افراد دارای در سازمانها رسمی دولتی. به اعتقاد او این افراد، مخلوق محدودیتهای مقرراتی، نهادی، اطلاعاتی، و محاسباتی می‌باشند و کارشان در چهار چوب یک "عقلانیت بسته"، صورت می‌پذیرد. در این کتاب، مدلی برای تممیم‌گیریهای اقتصادی ارائه گردید که "ملا" با مدلهای متداول و معمول، اختلاف داشت. بعدها این ایده عقلانیت بسته یا محدود، در مرکز کارهای تحقیقاتی سایمون در زمینه "درک عمیق چگونگی هوشمند بودن کامپیوترها و انسانها قرار گرفت".

در سال ۱۹۵۴، سایمون به همراه دو نفر دیگر کتابی در زمینه نحوه اداره جامعه نگاشت که شهرت فراوانی را برای او به ارمغان آورد. وی همچنین در سال ۱۹۶۰، به همراه سه نفر از همکارانش، کتاب جایی در مورد تحلیل مکانیسمهای کنترل خودکار تولید را نگاشت که نتایج آن منتشر شد. کتاب "تئوری سیستمهای ریاضیاتی" که در این زمینه

صورت می‌گرفت، نهاد، برخلاف تنوع و گستردگی فعالیتهای پژوهشی و تحقیقاتی سایمون، فعالیتهای شغلی و آکادمیک وی، بسیار ساده و مختصر است. او پس از اقامت کوتاهی در دانشگاه برکلی و سپس در انستیتوت تکنولوژی ایلینویز، در سال ۱۹۴۹ به دانشگاه کارنگی ملون^۵ پیوست. او در این دانشگاه، برپایه معلومات ارزنده‌ای که در رشته‌های اقتصاد، روانشناسی، و تحقیق در عملیات داشت، رشته دکتری مدیریت صنعتی را بنیان نهاد که این امر به نوبه خود، باعث تحول عظیمی در تحصیلات عالی در رشته تجارت گشت. او هم اکنون نیز در این دانشگاه مشغول به تدریس و تحقیق می‌باشد و از پانزده سال پیش، به درجه استادی روانشناسی و علوم کامپیوتر نائل آمده است.

اولین کارهای سایمون در زمینه علوم کامپیوتر، به همکاری او با جان شاول^۸ و آلن نیوول^۹ در ایجاد توسعه اولین برنامه‌های ذهنی و ابتکاری^{۱۰} ("نظریه پرداز منطق")^{۱۱} در سال ۱۹۵۶ و "حلال عمومی مسائل"^{۱۲} در سال



(۱۹۵۸) و اولین زبان پردازش لیستها^{۱۳} (زبان IPL در سال ۱۹۵۷)، اختصاص داشت. این تیم سه نفره به همراه جان مک کارتی، ماروین مینسکی^{۱۴}، و اولیور سلفریج^{۱۵} به عنوان بنیان‌گذاران "هوش مصنوعی" شناخته می‌شوند. برنامه‌های اولیه، مدلهای از چگونگی نحوه تفکر انسان را ارائه می‌کردند (بعدها به عنوان "شبیه‌سازی فراروندت فکر" خوانده شد). سپس سایمون در ادامه کار، دنیاهای از برنامه‌ها و تحلیلها را به وجود آورد و نظریات بسیار مهمی در مورد چگونگی بروز اعمال هوشمندانه به هنگام حل مسائل، یادآوری خاطرات، درک مطالب، و یادگیری ارائه نمود و ارتباطات مربوطه را مشخص و معین ساخت. در سال ۱۹۷۲، کارهای عمده او در زمینه حل مسائل توسط انسانها، در قالب کتابی با همکاری نیوول منتشر شد^{۱۶} و بعد در سال ۱۹۷۸، او ایده‌های خود در این زمینه را بطور گسترده تر و وسیعتر، در کتاب دیگری تحت عنوان "مدلهای تفکر" به رشته تحریر درآورد^{۱۷}.

در دوران حیات علمی پربار سایمون، جوایز علمی متعددی به وی تعلق گرفته است که از همه مهمتر، جایزه نوبل اقتصاد در سال

۱۹۷۸ می‌باشد. او همچنین در سال ۱۹۷۵ به همراه نیوول، به دریافت جایزه تیورینگ انجمن ماشینهای محاسبه^{۱۸}، نائل گشت. پیش از این، در سال ۱۹۶۹، جایزه مخصوص انجمن روانشناسی آمریکا نیز به خاطر مشارکت علمی ممتاز و همکاری برجسته در زمینه روانشناسی به او اختصاص یافته بود. علاوه بر اینها، جوایز علمی بسیار و دیپلمهای افتخار متعددی نیز به وی اهداء گردیده است.

پروفسور هربرت سایمون که عضو آکادمی ملی علوم آمریکا است، در بسیاری از جوامع علمی، فعال بوده و تاکنون در سطوح مختلفی از سوی حکومت آمریکا مورد مشورت و نظرخواهی قرار گرفته است. بازده فکری و کاری او، حتی بسا استنادهای افراد بزرگی نظیر او، شگفت آور و اعجاب‌انگیز می‌باشد. سایمون تا کنون ۱۲ کتاب و قریب ۵۰۰ مقاله علمی در زمینه‌های تحقیقاتی گوناگون از اقتصاد و علوم سیاسی گرفته تا روانشناسی و علوم کامپیوتر، ته‌لیف نموده است. سرچشمه فکرتوانای او تا این زمان (۱۹۸۰)، همچنان بطرز زوال ناپذیری جوشان و پرتوافشان مانده است.

با دداشها و مرجع

* Encyclopedia of Computer Science and Engineering, VNR, 1983.

1: Herbert Alexander Simon

2: Study of Organization

3: Operation Research

4: Simon, H. A., Smithburg, D. W.,

4: Administrative Behavior, New York: Macmillan

5: Simon, H. A., Smithburg, D. W., and Thompson, V. A., Public Administration New York: Knopf.

6: Holt, C. C., Modigliani, F., Muth, J. F., and Simon, H. A., Planning Production, Inventories, and Work Force. Englewood Cliffs, NJ: Prentice-Hall.

7: Carnegie Mellon University

8: John C. Shaw

9: Allen Newell

10: Heuristic Program

11: Logic Theorist

12: General Problem Solver

13: List Processing Language

14: Marvin Minsky

15: Oliver Selfridge

16: Newell, A. and Simon, H. A., Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall

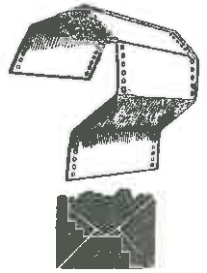
17: Simon, H. A., Models of Thoughts. New Haven, CT: Yale University Press.

18: ACM Turing Award

پاسخ

به

سؤالات



سوال: لطفاً بطور مختصر در مورد "هوش مصنوعی"، تاریخچه، زمینه‌های کاربرد آن توضیح دهید.

جواب: "هوش مصنوعی"، مطالعه‌ای پدیده‌های طبیعی است که کامپیوترها را قادر به انجام کارهای "هوشمندانه" می‌کند. اما نکته کلیدی در تعریف "هوش" نهفته است. آیا "هوش"، توانایی و قابلیت استدلال و استنتاج است؟ آیا قابلیت کسب و بکار بستن آگاهی است؟ آیا توانایی تکامل بخشیدن ایده‌هاست؟ مطمئناً همه این قابلیت‌ها بخشی از "هوش" را تشکیل می‌دهند ولی همه آن چیزی نیستند که می‌توان در این باره گفت و شاید راه یک تعریف جامع و کامل، ناممکن باشد. در مورد کامپیوترها نیز تلفظ "هوشمندی" دارای ابهامات و پیچیدگی‌های بسیاری است. چندی پیش که تحقیقات در زمینه این رشته علوم کامپیوتر آغاز گشت، مخالفت‌ها و مقاومت‌های زیادی را برانگیخت. برخی از ایراداتی که به "هوشمند شدن و توانایی تفکر" کامپیوترها گرفته می‌شد، صرفاً احساسی و ناشی از اضطراب و نگرانی از جایگزین شدن ماشین به جای انسان بود. گروهی می‌گفتند: "هوش پدیده‌ای مرموزی است که از جانب خداوند به انسان ارزانی شده است و کامپیوترها نمی‌توانند دارای آن شوند. و به همین دلیل هیچ کامپیوتری نخواهد توانست مانند شکسپیر بنویسد، مانند بتهوون موسیقی بسازد، و یا مانند نیوتن عالم باشد." و موافقان این امر در جواب می‌گفتند: "این برای همه آدمیان عادت شده است که فکر می‌کنند برتر از همه چیزند. زمانی زمین مرکز دنیا بود اما اکنون سیاره‌ها شناخته‌ای بیش نیست. امروزه تحقیقات در زمینه ساخت کامپیوترهای زیستی^۱ در جریان است و چنانچه به نتیجه برسد و چنین ساخته شود، شاید کاملاً با هوش باشد. و ثانیاً منطق حکم نمی‌کند که هوش کامپیوتر را با معیارهای فوق انسانی و نوابغ علم و هنر مانند نیوتن و بتهوون بررسیم زیرا در این صورت باید کاراکتر نویسنده، گان، موسیقیدانان، و دانشمندان بشر را نیز غیرهوشمندانه بدانیم." گروه دیگری از مخالفان می‌گفتند: "کامپیوترها فقط می‌توانند کارهایی را انجام دهند که برایش برنامه‌ریزی شده اند و لذا قادر بهبداع و ابتکار نیستند." و در جواب می‌شنیدند: "کامپیوترها چیزی را از مغز به وجود نمی‌آورند و به نظر می‌رسد که توانایی‌هایشان از انسان به آنها منتقل می‌شود ولی خود انسان هم

وابسته به ساختار و ژنتیکش است. انسان قادر است با پردازش اطلاعات، برداشته‌های خود بیاورد. پس چنانچه انسان بتواند میزبان هوش کامپیوتر را به این حد برساند، کامپیوترها هم بی‌شک خواهند توانست مانند انسان، سطح اطلاعاتشان را به وسیله شنیدن، گرفتن، تجربه کردن، پرسیدن، و گنجشک‌وی نمودن، بالا ببرند." در این رابطه، بجایست نظری به خصوصیات و مشخصات کامپیوترها بیا فکنیم و ببینیم که آیا اجزاء تشکیل دهنده آنها، امکان انجام اعمال هوشمندانه را به آنها می‌دهد.

کامپیوترهای همه منظوره^۲ و متعارف، از اجزاء زیر تشکیل شده اند:

- یک یا چند دستگاه ورودی، جهت کسب اطلاعات از محیط خارج
- یک یا چند دستگاه خروجی، برای ارسال نتایج حاصل از پردازش‌های انجام شده به محیط خارج
- یک یا چند واحد حفظ اطلاعات (حافظه)
- یک واحد حساب و منطق
- یک واحد کنترل
- واحدهای فرعی

در این میان، واحد کنترل از اهمیت خاصی برخوردار است. نحوه سیم‌بندی این قسمت به گونه‌ای است که آن را قادر می‌سازد تا یک سری "عملیات" و "دستورات" را تفسیر کرده و واحدهای دیگر را بکار اندازد. یک دستور مهم در بین این دستورات، "مقایسه و انتقال کنترل" است که کامپیوتر را قادر به انتخاب سادها بین دو چیز می‌کند. مثلاً اگر الف با ب برابر است اطلاعات بعدی را بخوان و وگرنه خط بعدی را بنویس. از طریق این مکانیسم ساده، تصمیم‌گیری‌های پیچیده‌تری را می‌توان سازمان داد.

نقش برنامه‌نویس در آن است که عملیات پیچیده‌تر از آنچه یک سخت افزار را برایش سیم‌بندی شده است را از طریق تنظیم منطقی یک سری دستورات ساده که سیم‌بندی سخت افزار قادر به انجام آنها می‌باشد، به صورت یک "برنامه" تهیه کند و به اجرا درآورد. به این ترتیب، کامپیوتر به ما شینی با قدرتهای شگرف بدل می‌شود. در یک لحظه، کار یک حسابدار را انجام می‌دهد، در لحظه بعد، برنامه‌های پرواز یک شرکت هواپیمایی را تنظیم می‌کند، سپس پرونده مشتریان شرکت تلفن را بررسی می‌نماید. با این زمینه کلی، این سوال می‌تواند بطور منطقی مطرح گردد که:

آیا کامپیوترها قادر به تفکرند؟

اگر فرض بر این باشد که تنها انسان قادر به تفکر (و انجام اعمال هوشمندانه) است و پس گفته شود تفکر (هوش) اساساً مکانیسم مرموزی است و نمی‌توان ماهیت آن را درک کرد، پاسخ به سوال فوق، منفی خواهد بود. اما اگر بپذیریم که تنها براساس مشاهدات و تجربیات به عمل آمده در مورد رفتار کامپیوترها و مقایسه آن با رفتار هوشمندانه انسان‌هاست که می‌توان در صورت تشابه، واژه "متفکرانه" یا "هوشمندانه" را به هر دو رفتار اطلاق کرد، در این صورت

می‌توان به سوال فوق، پاسخ مثبت داد. بهر حال باید به خاطر سپرد که اطلاق امکان تفکر در کامپیوترها به هیچ وجه به معنی آن نیست که کامپیوترها هم‌گنون قادر به تفکر و یا به میزان انسان‌ها متفکرند. هدف تحقیقات در زمینه هوش مصنوعی، تهیه نرم افزارهایی است که قادر به انجام عملیاتی باشند که در صورت انجام آنها از طرف انسان‌ها، "هوشمندانه" یا "متفکرانه" قلمداد خواهند شد (البته بدیهی است که چنین نرم افزارهایی، سخت افزارهای مناسب خود را می‌طلبند).

ولی آیا کامپیوترها تنها آن چیزی را انجام نمی‌دهند که به آنها دستور داده می‌شود و نه بیشتر؟

سایمون^۳، یکی از پیشگامان تحقیق در زمینه هوش مصنوعی، به این سوال آشنا این گونه پاسخ می‌دهد:

"این جمله که کامپیوترها فقط آنچه را انجام می‌دهند که برایش برنامه‌ریزی شده اند، مقدّمات واضح و بدون شک صحیح است. ولیکن هیچیک از نتیجه‌گیری‌هایی که برخی از افراد از آن به عمل می‌آورند را نتیجه نمی‌دهد. انسان قادر است فکر کند، یاد بگیرد، و خلق کند از آن رو که خواص بیولوژیکی وی این صفات را در او مهیا ساخته و همچنین از آن رو که بر خوردش با محیط باعث غنی‌تر شدن نظام اولیه وجود است و آن خصوصیات را شکوفا می‌کند. یک کامپیوتر هم خارج از این زمینه عمل نمی‌کند. از طریق یک برنامه اولیه است که کامپیوتر می‌تواند تفکر کند، یاد بگیرد، و خلق کند. اگر یک کامپیوتر بخواند و ندانسان عمل کند، این تنها از طریق یک برنامه ممکن خواهد شد و این برنامه، مسلماً ساده‌تر از برنامه اولیه وجود خواهد بود انسان نیست.

با یاد آفرین کردن این تصور که یک کامپیوتر نمی‌تواند رفتاری هوشمندانه‌تر از برنامه‌ش نویس داشته باشد و برنامه‌نویس کاملاً قادر به پیش بینی عملکرد برنامه‌اش خواهد بود، اشتباه است. زیرا که قدرت عظیم کامپیوترها را در مقایسه با پیچیدگی‌های وسیع اطلاعات نامیده گرفته ایم. برنامه‌نویس نمی‌تواند با ارائه یک نسخه کلی برای ابقاء یک رفتار مشخص، تمام امکانات را پس از اجرای میلیون‌ها دستور جزئی در فضای کلی یک مسئله در نظر آورد. و مهم‌تر از همه این که یک برنامه‌نویس نمی‌تواند همان دستورات را با سرعت و دقت کامپیوتر در مغز خودش به مرحله اجرا درآورد. مسائل اساسی در زمینه هوش مصنوعی کدامند؟

در رشته‌ای از تحقیقات به جوامع هوش مصنوعی، مسائل تحقیقاتی بسیاری وجود دارند. از مزه مهم‌ترین مسائل کلی در زمینه هوش مصنوعی می‌توان از مسائل زیر نام برد:

۱- مسائل مرحله‌ای

انواع بازی‌ها از قبیل دوز بازی، شطرنج و... جزء مسائل مرحله‌ای هستند که کامپیوتر می‌تواند بطور مرحله‌ای به حل آنها بپردازد. ورق بزنی

یافتن راه‌حلهای بهینه^۴ در جستجوی درخت با رویه^۵ و قطع شاخه‌های زائد آنها، از اهم موارد در این گونه مسائل است.

۲- منطق استقرایی

هوش مصنوعی در محدوده منطق استنتاجی نسبتاً قوی است اما هنوز در زمینه منطق استقرایی، نقاط ضعف بسیاری دارد. در حالی که این رابطه، اگر نه به یک نسبت، به‌رحال در آنها معکوس است. تقویت هوش مصنوعی در زمینه منطق استقرایی، زمینه مهمی از تحقیقات در این رشته را تشکیل می‌دهد. به عبارت دیگر، برنامه‌هایی باید تهیه گردند که به نحوی "مدلی‌ها" از محیط خارج را در داخل کامپیوتر ارائه دهند تا کامپیوتر بتواند بر اساس آنها، پیشگویی‌های مؤثری در مورد آینده و شرایط آن انجام دهد و قادر به ارائه فرضیات و نظریات گردد.

۳- درک زیادهای طبیعی

یکی دیگر از مسائل بسیار مهم، هم‌در زمینه عملی و هم‌نظری، تهیه برنامه‌هایی است که قادر به درک زبان طبیعی انسان‌ها باشند. به عبارت دیگر، مکالمه انسان و ماشین برپایه زبان انسان‌ها ایجاد گردد نه زبان ماشین. در این زمینه تا کنون پیشرفت‌های چشمگیری حاصل گشته است و هم‌اکنون سیستم‌هایی که بطور محدود قادر به درک زبان طبیعی می‌باشند، به‌بازار عرضه گردیده‌اند.

۴- علم آدمک‌های مصنوعی (هوشمند)

امروزه تحقیقات در زمینه آدمک‌های مصنوعی و میزان کارآئی آنها در شئون مختلف زندگی انسان‌ها از زمینه‌های بسیار مهم تحقیقاتی می‌باشد. با پیشرفت‌هایی که به عمل آمده است، بی‌گمان این آدمک‌ها نقش عمده‌ای را در زندگی فردای انسان‌ها بر عهده خواهند داشت. هم‌اکنون استفاده از آدمک‌های مصنوعی در صنایع و کارهای تولیدی گوناگون نظیر کارخانجات اتومبیل‌سازی، کارخانجات ساخت لوازم خانگی، صنایع بسته‌بندی، صنایع برش و دوخت لباس، سیستم‌های انبارداری جدید و امثال آن، رایج گشته است. در آینده آدمک‌های مصنوعی در انجام کارهای سنگین و خطرناک مانند کار در راکتورهای اتمی

و یا زیردریاها، جایگزین انسان خواهند شد.

۵- سیستم‌های خبره

سیستم‌های خبره، از نتایج پیشرفت‌های تئوریک در زمینه هوش مصنوعی است. مقصود از ایجاد یک سیستم خبره، ایجاد برنامه (سیستمی از برنامه‌ها) است که در زمینه خاصی از علوم و فنون، متخصص و خبره محسوب شود. برای نیل به این منظور، سیستم خبره به اطلاعات وسیع در موضوع تخصصی خود، به‌فراوان و فوق‌فراوان‌ها یعنی قاعده‌های حاکم بر نحوه کار برد قاعده‌های معمول، و قابلیت "یادگیری" نیاز دارد. مخزن اصلی اطلاعات کلی یک سیستم خبره، پایگاه بصیرت^۷ نامیده می‌شود که به‌فراوان پایگاه داده‌ها^۸ نامگذاری شده است. نکته جالب در مورد برخی از سیستم‌های خبره این است که پس از حل مسائل ساده‌ای، معین، اگر نتایج حاصله و تازه باشند، آن را بر پایگاه بصیرت خود می‌افزایند.

محدودیت‌های هوش مصنوعی کدامند؟

هنوز کسی پاسخ قطعی به این سوال نداده است. در هر حال، از نظر تئوری، امکانات پیشرفت این شاخه تحقیقات علمی، نامحدود به نظر می‌رسد. همان گونه که هوش و تفکر انسانی در طی قرون متما دی تکامل انسان‌ها و تکامل حیات، توسعه شگرفی یافته است و هنوز محدودیتی برای توسعه آن وجود ندارد، به نظریه‌پردازان در مورد کامپیوترها نیز حیطه عمل به همین گستردگی باشد. به‌رحال، علیرغم واکنش‌های اولیه، امروزه هوش مصنوعی از جذاب‌ترین و فعال‌ترین رشته‌های تحقیقاتی در بین همه علوم بشری است و دستاوردهای آن مسلماً نقش تعیین‌کننده‌ای در زندگی فردای انسان‌ها خواهند داشت.

یادداشت‌ها

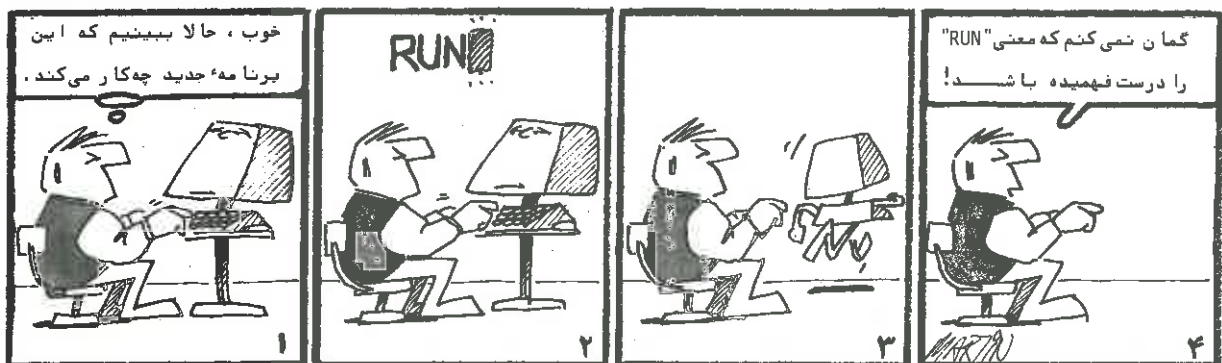
- 1: Biocomputers
- 2: General-purpose
- 3: Simon
- 4: Optimized
- 5: Game-trees
- 6: Expert Systems
- 7: Knowledge Base
- 8: Data Base

دنباله مفاهیم اولیه سیستم‌های عامله

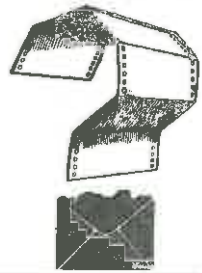
به‌علاوه، بعضی از سیستم‌ها امکان‌اتی را برای پشتیبانی پرونده‌های آزمایشی در اختیار دارند که از آنها می‌توان به‌منظور ایجاد پرونده‌های خصوصی جهت استفا ده در آزمایش برنامه‌های کاربردی، بهره‌گرفت.

یادداشت‌ها

- 1: Buffer
- 2: Single-buffering
- 3: Double-buffering
- 4: Dynamic buffering
- 5: Exchange buffering
- 6: Overhead
- 7: Interface
- 8: Remote Processing
- 9: Operator
- 10: Recovery Processing
- 11: Checkpointing/Restart
- 12: Serial Processing Systems
- 13: On-line
- 14: Task
- 15: Roll-in/Roll-out
- 16: Transaction Processing Systems
- 17: Batch Systems
- 18: Paging
- 19: Data Compression
- 20: Parity Errors
- 21: Amdahl
- 22: Interval Timing
- 23: Dump
- 24: Formatted
- 25: Tracing
- 26: Interactive Time-sharing Systems
- 27: Breakpoints
- 28: Logging and Accounting
- 29: Data Management
- 30: Macroinstructions
- 31: Data Base
- 32: Reorganize
- 33: Merge
- 34: Backup
- 35: Labels
- 36: File Edition Number
- 37: Indexing
- 38: Offline
- 39: Ring
- 40: Blocking
- 41: Deblocking
- 42: Data-handling Utilities
- 43: Multistep Job
- 44: Positioning
- 45: Backspace
- 46: Forward Space
- 47: Storage Compaction



پاسخ به سؤالات



سؤال : لطفاً بطور مختصر در مورد "هوش مصنوعی" تاریخچه، زمینه‌های کاربردی آن توضیح دهید.

جواب : "هوش مصنوعی"، مطالعه ایده‌ها و روشی است که کامپیوترها را قادر به انجام کارهای "هوشمندانه" می‌کند. اما نکته کلیدی در تعریف "هوش" نهفته است، آیا "هوش"، توانایی و قابلیت استدلال و استنتاج است؟ آیا قابلیت کسب و بکار بستن آگاهی است؟ آیا توانایی تکامل بخشیدن ایده‌هاست؟ مطمئناً همه این قابلیت‌ها بخشی از "هوش" را تشکیل می‌دهند ولی همه آن چیزی نیستند که می‌توان در این باره گفت و شاید راه یک تعریف جامع و کامل، ناممکن باشد. در مورد کامپیوترها نیز لفظ "هوشمندی" دارای ابهامات و پیچیدگی‌ها و ابهامات است. چندی پیش که تحقیقات در زمینه این رشته علوم کامپیوتر آغاز گشت، مخالفت‌ها و مقاومت‌های زیادی را برانگیخت. برخی از ایراداتی که به "هوشمند شدن و توانایی تفکر" کامپیوترها گرفته می‌شد، صرفاً احساسی و ناشی از اضطراب و نگرانی از جایگزین شدن ماشین به جای انسان بود. گروهی می‌گفتند: "هوش پدیده مرموزی است که از جانب خداوند به انسان ارزانی شده است و کامپیوترها نمی‌توانند دارای آن شوند. و به همین دلیل هیچ کامپیوتری نخواهد توانست مانند یک پسر بنویسد، مانند بنهون موسیقی بسازد، و یا مانند نیوتن عالم باشد." و موافقان این امر در جواب می‌گفتند: "این برای همه آدمیان عادت شده است که فکر می‌کنند برتر از همه چیزند. زمانی زمین مرکز دنیا بود اما اکنون سیاره‌ها شناخته‌ای نیست. امروزه تحقیقات در زمینه ساخت کامپیوترهای زیستی در جریان است و چنانچه به نتیجه برسد و چنین چیزی ساخته شود، شاید کاملاً با هوش باشد. و شاید منطق حکم نمی‌کند که هوش کامپیوترها با معیارهای فوق انسانی و نوابغ علم و هنر مانند نیوتن و بنهون بررسی کنیم زیرا در این صورت باید کارا کثرت‌یابند. گان، موسیقیدانها، و دانشمندان بشر را نیز غیر هوشمندانه‌ها بنیم، "گروه دیگری از مخالفان می‌گفتند: "کامپیوترها فقط می‌توانند کارهایی را انجام دهند که برای بیش برنا مریزی شده‌اند و لذا قادر بهبداع و ابتکار نیستند." و در جواب می‌شنیدند: "کامپیوترها چیزی را از صفر به وجود نمی‌آورند و به نظر می‌رسد که توانایی‌هاشان از انسان به آنها منتقل می‌شود ولی خود انسان هم

وابسته به ساختار و زنجیره است. انسان قادر است با پردازش اطلاعات، برداشته‌های خود بیاورد. پس چنانچه انسان بتواند میزبان هوش کامپیوترها به این حد برساند، کامپیوترها هم بی‌شک خواهند توانست مانند انسان، سطح اطلاعات را به وسیله شنیدن، گرفتن، تجربه کردن، پرسیدن، و کنجکاوی نمودن، بالا ببرند. در این رابطه، بجایست نظری به خصوصیات و مشخصات کامپیوترها بیا فک کنیم و ببینیم که آیا اجزاء تشکیل دهنده آنها، امکان انجام اعمال هوشمندانه را به آنها می‌دهد.

کامپیوترهای همه منظوره و متعارف، از اجزاء زیر تشکیل شده‌اند:

- یک یا چند دستگاه ورودی، جهت کسب اطلاعات از محیط خارج
- یک یا چند دستگاه خروجی، برای ارسال نتایج حاصل از پردازش‌های انجام شده به محیط خارج
- یک یا چند واحد حفظ اطلاعات (حافظه)
- یک واحد حساب و منطق
- یک واحد کنترل
- واحدهای فرعی

در این میان، واحد کنترل از اهمیت خاصی برخوردار است. نحوه سیم‌پیچی این قسمت به گونه‌ای است که آن را قادر می‌سازد تا یک سری "عملیات" و "دستورات" را تفسیر کرده و واحدهای دیگر را بکار اندازد. یک دستور مهم در بین این دستورات، "مقایسه و انتقال کنترل" است که کامپیوترها قادر به انتخاب ساده بین دو چیز می‌کنند. مثلاً اگر الف با ب برابر است اطلاعات بعدی را بخوان وگرنه خط بعدی را بنویس. از طریق این مکانیسم ساده، تصمیم‌گیری‌های پیچیده‌تری را می‌توان سازمان داد.

نقش برنا م‌نویس در آن است که عملیات پیچیده‌تر از آنچه یک سخت افزار را برایش سیم‌پیچی شده است را از طریق تنظیم منطقی یک سری دستورات ساده که سیم‌پیچی سخت افزار قادر به انجام آنها می‌باشد، به صورت یک "برنا م‌ه" تهیه کند و به اجرا درآورد. به این ترتیب، کامپیوتر به ماشین با قدرتهای شگرف بدل می‌شود. در یک لحظه، کار یک حسابدار را انجام می‌دهد، در لحظه بعد، برنا م‌های پرواز یک شرکت هواپیمایی را تنظیم می‌کند، و سپس پرونده مشتریان شرکت تلفن را بررسی می‌نماید. با این زمینه کلی، این سؤال می‌تواند بطور منطقی مطرح گردد که:

آیا کامپیوترها قادر به تفکرند؟

اگر فرض بر این باشد که تنها انسان قادر به تفکر (و انجام اعمال هوشمندانه) است و بهیچ گفته شود تفکر (هوش) اساساً مکانیسم مرموزی است و نمی‌توان ماهیت آن را درک کرد، پاسخ به سؤال فوق، منفی خواهد بود. اما اگر بپذیریم که تنها براساس مشاهدات و تجربیات به عمل آمده در مورد رفتار کامپیوترها و مقایسه آن با رفتار هوشمندانه انسان‌هاست که می‌توان در صورت تشابه، واژه "متفکرانه" یا "هوشمندانه" را به هر دو رفتار اطلاق کرد، در این صورت

می‌توان به سؤال فوق، پاسخ مثبت داد. به هر حال با پدیده‌ها طرپر درگاه اطلاق امکان تفکر در کامپیوترها به هیچ وجه معنی آن نیست که کامپیوترها همان‌گونه قادر به تفکر و یا به میزان انسان‌ها متفکرند. هدف تحقیقات در زمینه هوش مصنوعی، تهیه نرم افزارهایی است که قادر به انجام عملیاتی باشند که در صورت انجام آنها از طرف انسان‌ها، "هوشمندانه" یا "متفکرانه" قلمداد خواهند شد (البته پدیده‌ای است که چنین نرم افزارهایی، سخت افزارهای مناسب خود را می‌طلبند).

ولی آیا کامپیوترها تنها آن چیزی را انجام نمی‌دهند که به آن دستور داده می‌شود و نه بیشتر؟

سایمون آ، یکی از پیشگامان تحقیق در زمینه هوش مصنوعی، به این سؤال آشنا این گونه پاسخ می‌دهد:

"این جمله که کامپیوترها فقط آنچه را انجام می‌دهند که برای بیش برنا م‌ریزی شده‌اند، مقدّمات واضح و بدون شک صحیح است. ولیکن هیچیک از نتیجه‌گیری‌هایی که برخی از افراد از آن به عمل می‌آورند را نتیجه نمی‌دهد. انسان قادر است فکر کند، یاد بگیرد، و خلق کند از آن رو که خواص بیولوژیکی وی این صفات را در او مهیا ساخته و همچنین از آن رو که برخوردش با محیط، باعث غنی‌تر شدن نظام اولیه و جوداوست و آن خصوصیات را شکوفاتر می‌کند. یک کامپیوتر هم خارج از این زمینه عمل نمی‌کند. از طریق یک برنا م‌ه اولیه است که کامپیوتر می‌تواند تفکر کند، یاد بگیرد، و خلق کند. اگر یک کامپیوتر بخواد مانند انسان عمل کند، این تنها از طریق یک برنا م‌ه ممکن خواهد شد و این برنا م‌ه، مسلماً ساده‌تر از برنا م‌ه اولیه وجود خواهد داشت و انسان نیست.

با یادافه کرد که این تصور که یک کامپیوتر نمی‌تواند رفتارهای هوشمندانه‌تر از برنا م‌ه نویسنش داشته باشد برنا م‌نویس کاملاً قادر به پیش بینی عملکرد برنا م‌هاش خواهد بود، اشتباه است. زیرا که قدرت عظیم کامپیوترها را در مقایسه با پیچیدگی‌های وسیع اطلاعات نا دیده گرفته ایم. برنا م‌نویس نمی‌تواند بسیار آهسته یک نسخه کلی برای القاء یک رفتار مشخص، تمام امکانات را پس از اجرا برای میلیونها دستور جزئی در فضای کلی یک مسأله در نظر آورد. و مهم‌تر از همه این که یک برنا م‌ه نویسنمی‌تواند در همان دستورات را با سرعت و دقت کامپیوتر در مغز خودش به مرحله اجرا درآورد.

مسائل اساسی در زمینه هوش مصنوعی کدامند؟ در رشته‌ای از تحقیقات به جوامع هوش مصنوعی، مسائل تحقیقاتی بسیاری وجود دارند. از زمره مهمترین مسائل کلی در زمینه هوش مصنوعی می‌توان از مسائل زیرنا م‌برد:

۱- مسائل مرحله‌ای انواع بازیها از قبیل دوز بازی، شطرنج و... جزء مسائل مرحله‌ای هستند که کامپیوتر می‌تواند بطور مرحله‌ای بدلیل آنها بپردازد. ورق بزنند

دنباله مفاهیم اولیه سیستمهای عاملها

به علاوه، بعضی از سیستمها امکاناتی را برای پشتیبانی پروندههای آزمایشی در اختیار دارند که آنها میتوانند به منظور ایجاد پروندههای خصوصی جهت استفاذه در آزمایش برنامهای کاربردی، بهره گرفت.

یادداشتها

- 1: Buffer
- 2: Single-buffering
- 3: Double-buffering
- 4: Dynamic buffering
- 5: Exchange buffering
- 6: Overhead
- 7: Interface
- 8: Remote Processing
- 9: Operator
- 10: Recovery Processing
- 11: Checkpointing/Restart
- 12: Serial Processing Systems
- 13: On-line
- 14: Task
- 15: Roll-in/Roll-out
- 16: Transaction Processing Systems
- 17: Batch Systems
- 18: Paging
- 19: Data Compression
- 20: Parity Errors
- 21: Amdahl
- 22: Interval Timing
- 23: Dump
- 24: Formatted
- 25: Tracing
- 26: Interactive Time-sharing Systems
- 27: Breakpoints
- 28: Logging and Accounting
- 29: Data Management
- 30: Macroinstructions
- 31: Data Base
- 32: Reorganize
- 33: Merge
- 34: Backup
- 35: Labels
- 36: File Edition Number
- 37: Indexing
- 38: Offline
- 39: Ring
- 40: Blocking
- 41: Deblocking
- 42: Data-handling Utilities
- 43: Multistep Job
- 44: Positioning
- 45: Backspace
- 46: Forward Space
- 47: Storage Compaction

وبایزیدریها، جایگزین انسان خواهند شد.

۵ - سیستمهای خبره

سیستمهای خبره، از نتایج پیشرفتهای تئوریک در زمینه هوش مصنوعی است. مقصود از ایجاد یک سیستم خبره، ایجاد برنامه (پاسی سیستمی از برنامهها) است که در زمینه خاصی از علوم و فنون، متخصص و خبره محسوب شود. برای نیل به این منظور، سیستم خبره به اطلاعات وسیع در موضوع تخصصی خود، به قاعدهها و فوق قاعدهها یعنی قاعدههای حاکم بر نحوه کاربرد قاعدههای معمول، و قابلیت "یادگیری" نیاز دارد. مخزن اصلی اطلاعات کلی یک سیستم خبره، پایگاه بصیرت^۷ نامیده میشود که به قیاس پایگاه دادهها^۸ نامگذاری شده است. نکته جالب در مورد برخی از سیستمهای خبره این است که پس از حل مسألهای معین، اگر نتایج حاصله نو و تازه باشند، آن را بر پایگاه بصیرت خود میفزایند.

محدودیتهای هوش مصنوعی کدامند؟

هنوز کسی پاسخ قطعی به این سؤال نداده است. در هرحال، از نظر تئوری، امکانات پیشرفت این شاخه از تحقیقات علمی، نامحدود به نظر میرسد. همان گونه که هوش و تفکر انسانی در طی قرون متعادی تکامل انسانها و تکامل حیات، توسعه شگرفی یافته است و هنوز محدودیتی برای توسعه آن وجود ندارد، به نظر میرسد که در مورد کامپیوترها نیز حیطة عمل به همین گستردگی باشد. به هرحال، علیرغم واکنشهای اولیه، امروزه هوش مصنوعی از جذابترین و فعالترین رشتههای تحقیقاتی در بین همه علوم بشری است و دستاوردهای آن مسلماً نقش تعیین کننده ای در زندگی فردی انسانها خواهند داشت.

یادداشتها

- 1: Biocomputers
- 2: General-purpose
- 3: Simon
- 4: Optimized
- 5: Game-trees
- 6: Expert Systems
- 7: Knowledge Base
- 8: Data Base

یافتن راهحلهای بهینه^۴ در جستجوی درخت بازبها^۵ و قطع شاخههای زائد آنها، از اهم موارد در این گونه مسائل است.

۲ - منطق استقرایی

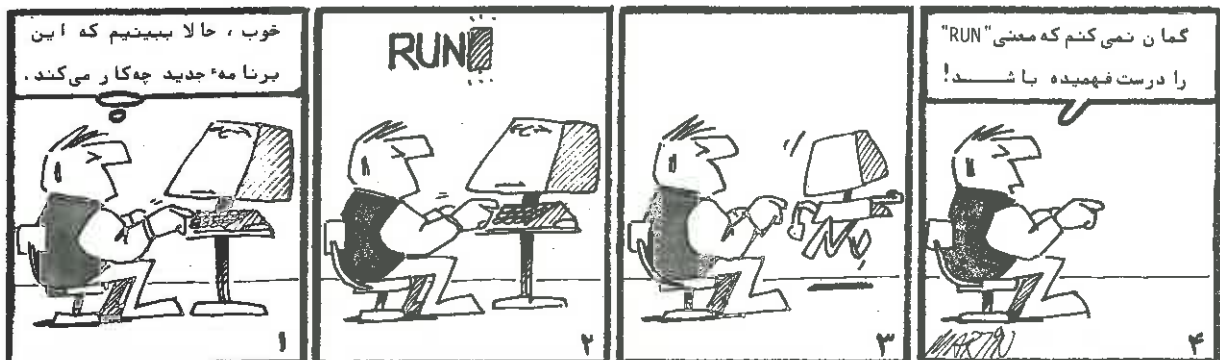
هوش مصنوعی در محدوده منطق استنتاجی نسبتاً قوی است اما هنوز در زمینه منطق استقرایی، نقاط ضعف بسیاری دارد. درحالی که این رابطه، اگر نه بیدک نسبت، بهر حال در انسانها معکوس است. تقویت هوش مصنوعی در زمینه منطق استقرایی، زمینه مهمی از تحقیقات در این رشته را تشکیل میدهد. به عبارت دیگر، برنامههایی باید تهیه گردند که به نحوی "مدلهائی" از محیط خارج را در داخل کامپیوترها ارائه دهند تا کامپیوتر بتواند بر اساس آنها، پیشگوییهای مؤثری در مورد آینده و شرایط آن انجام دهد و قاعدهها را به فرضیات و نظریات گردد.

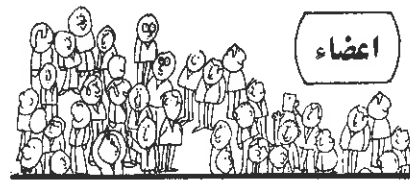
۳ - درک زبانهای طبیعی

یکی دیگر از مسائل بسیار مهم، هم در زمینه عملی و هم نظری، تهیه برنامههایی است که قادر به درک زبان طبیعی انسانها باشند. به عبارت دیگر، مکالمه انسان و ماشین برپایه زبان انسانها ایجاد گردیده زبان ماشین. در این زمینه تا کنون پیشرفتهای چشمگیری حاصل گشته است و هم اکنون سیستمهایی که به طور محدود قادر به درک زبان طبیعی میباشند، به بازار عرضه گردیده اند.

۴ - علم آدمکهای مصنوعی (هوشمند)

امروزه تحقیقات در زمینه آدمکهای مصنوعی و میزان کارائی آنها در شئون مختلف زندگی انسانها از زمینههای بسیار مهم تحقیقاتی میباشد. با پیشرفتهای که به عمل آمده است، بیگمان این آدمکها نقش عمده ای را در زندگی فردی انسانها بر عهده خواهند داشت. هم اکنون استفاده از آدمکهای مصنوعی در صنایع و کارهای تولیدی گوناگون نظیر کارخانجات اتوموبیل سازی، کارخانجات ساخت لوازم خانگی، صنایع بسته بندی، صنایع برش و دوخت لباس، سیستمهای انبارداری جدید و مثال آن، رایج گشته است. در آینده آدمکهای مصنوعی در انجام کارهای سنگین و خطرناک مانند کار در راکتورهای اتمی





اعضاء

تغییرات در آمار اعضای انجمن

افراد زیرتاپایان آذر ماه جاری به عضویت

انجمن انفورماتیک ایران درآمده اند:

۳۱۹۶	علی عسگری	مهدی	و
۳۱۹۷	پروین	مسعود	و
۵۵۲۷	خیمه دوزشیرازی	محسن	د
۵۵۲۸	منصوری شیرازی	نصرالله	د
۵۵۲۹	شمس	مجید	د
۵۵۳۰	شمس	کوروش	د
۳۱۹۸	فرحزاد	فرحذخت	و
۳۱۹۹	سپروس	مریم	و
۵۵۳۱	یوحنا	اسحاق	و
۱۶۵۴	وشو قنمند	مهدی	ع
۵۵۳۲	عراقی پور	نجفه	د
۵۵۳۳	احمدی	محمد رضا	د
۵۵۳۴	عربی شیرازی	سروش	د
۵۵۳۵	درمیشیان	فریده	د
۵۵۳۶	بیزدانی	علی اصغر	د
۵۵۳۷	نامدار	بهمنیار	د
۳۲۰۰	خانجانی	محمدجواد	و
۱۶۵۵	محسنی	محمود	ع
۱۶۵۶	بصیری	کرمان الله	ع
۳۲۰۱	مراد مند خورزوقی	ناهد	و
۳۲۰۲	کاج آبادی	ظاهره	و
۳۲۰۳	ایل بیگ بهرام آبادی	مژده	و
۱۶۵۷	پطروسیان	آلبرت	ع
۱۶۵۸	منصوری دارا	ایرج	ع
۳۲۰۴	اشراقی	مهتاب	و
۵۵۳۸	شهباز	وحید	د
۵۵۳۹	علوی فر	سید محمد رضا	د
۵۵۴۰	خورشیدی بنام	محمد قمر	د
۵۵۴۱	کوشان	سیامک	د
۵۵۴۲	یا قوتی	محسن	د
۵۵۴۳	شهبازی	ژورژیک	د
۵۵۴۴	قره باغی	ویدا	د
۵۵۴۵	کردنوه سی	مهدی	د
۵۵۴۶	محرم مرکب ساز	علیرضا	د
۵۵۴۷	قامیه	حمید	د
۵۵۴۸	کلهر	فضل الله	د
۵۵۴۹	کا کوشی	کاوه	د
۵۵۵۰	صادقی گرما رودی	سیامک	د
۵۵۵۱	موفقی	علی	د
۵۵۵۲	پوروطن	بهمن	د
۵۵۵۳	طیبا طیبائی کاشانی	ناصر	د
۵۵۵۴	مکیسان	نا هوشنگ	د
۵۵۵۵	مهرداری	مهران	د
۵۵۵۶	نهادندی	علیرضا	د
۵۵۵۷	هدائی امفها نی	کورین	د
۵۵۵۸	تغییر دگان شهرانی	محمد	د
۵۵۵۹	مسترشیدی	احمد	د
۵۵۶۰	ترا بلی لنگرودی	دیپیم	د
۵۵۶۱	ضیائی دوستان	محمد	د
۵۵۶۲	الفت	مسعود	د
۵۵۶۳	گلزاری	عبدالرضا	د
۵۵۶۴	شاپوری	گرینا	د
۵۵۶۵	غلامعلی	عبدالرضا	د
۵۵۶۶	رحیمی اقدام هریس	رحیم	د
۵۵۶۷	قامه سزاده	ابرج	د
۵۵۶۸	یوسفی فرد	چا نعلی	د
۵۵۶۹	محمدنژاد	غلامرضا	د
۵۵۷۰	کاظم فرح زادی	داود	د
۵۵۷۱	سعیدی نیا	بهمن	د
۵۵۷۲	چهارنگیری اسمعیلی	مرتضی (۶۴-۶۶)	د
۵۵۷۳	شکراللهی یا چشمه	بهزاد	د
۵۵۷۴	ستوده	مهدی	د
۱۶۵۹	عالمی	حسن	ع
۱۶۶۰	لری آمینی	محمد	ع
۳۲۰۵	مهاجر	محمود	و
۵۵۷۵	کلی بکی	حسن	و
۵۵۷۶	مصدق	کامییز	د

۵۵۹۷	امیراللهی	محسن	د
۵۵۹۸	قربانی	وحیدرضا	د
۵۵۹۹	حنیفی افشار	شرمین	د
۵۶۰۰	آشتی آشتیانی	ویولت	د
۵۶۰۱	تقشوانی	گیتا	د
۵۶۰۲	آریا	منصور	د

تجدید عضویتها

افراد زیرتاپایان آذر ماه جاری، عضویت

خود را در انجمن تجدید نموده اند:

۱۴۲۴	غیاثیان	احمد	ع
۳۰۸۴	رجبعلی پور	مهدی	و
۵۴۰۱	نوری	مهدی	د
۱۰۹۷	کامران	سعید	ع
۱۰۱۱	امامی	سیاوش	ع
۵۳۰۴	مرصاد	جواد	د
۵۱۰۳	فاطمی زاده	ساعت	د
۵۲۲۹	ابن یوسف	پیمان	د
۱۵۲۹	سلما سی زاده	محمود	ع
۵۲۷۹	عنصری	معصومه	د
۱۶۴۹	باقریان فرح آبادی	ابوالقاسم د به ع	د
۱۴۰۸	بهرچندی	خسرو	ع
۵۲۶۸	امساکی	مهناز	د
۱۶۵۰	فلاح	حسن	د به ع
۵۴۰۰	متولی علی آبادی	ها بیده	د
۵۴۵۵	طوبلی	رضا	د
۱۶۵۱	قاسم ثانی	غلامرضا د به ع	د
۵۴۴۶	حسینی نژاد	سید مهدی	د
۱۰۳۴	کامران پور	علیرضا	د
۵۲۶۵	شوشتریان	بهیژن	د
۳۱۷۸	رزاق منش	مسعود	و
۵۴۵۰	سرور	فریده	د
۵۴۴۸	نجفی	علی	د
۵۴۴۹	لطفیان	عظیبه	د
۵۲۴۴	بیوک آقا زاده	اکبر	د
۱۴۹۵	آکار	حسین	ع
۵۳۳۹	زندندان	حمید	د
۵۳۵۷	خواججه نصیری	مهرداد	د
۱۴۶۰	مسعودی	رجبعلی	ع
۵۴۴۴	سادات دربندی	کیهان دخت	د
۵۱۹۹	شلویری	گیتی	د
۵۳۷۱	فامیل سمواتی	فرامرز	د
۵۴۴۴	سقوطی	مریبه	د
۳۱۵۶	رایگان	شهرزاد	و
۱۵۴۹	عباسیان	عبدالاحسین	ع
۱۶۰۹	ناصری	مسعود	ع
۱۴۰۵	جذبی	محمد هاشم	ع
۳۱۲۵	خاکپور	نادر	و
۵۴۰۹	صادقی گاوگانی	رامین	د
۵۲۷۰	کورنگی	علی	د
۱۵۹۰	منوچهریان	فرهنگ	ع
۱۵۵۷	ملک	رحمن وفا	ع
۱۵۲۰	داعی حق	سعید	ع
۵۲۸۷	شهانقی	مسعود	د
۱۳۴۸	لوا سانی	سید محمد تقی	ع
۳۳۳۲	سعیدی	اعظم	د
۱۴۶۸	خسروی	حمید	ع

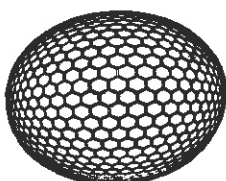
۳۱۵۰	فاضل پور	محمد علی	و
۱۲۲۵	داورپناه جزی	محمد	ع
۱۳۹۵	شریفی پور	علی	ع
۳۱۱۱	حشمتی	همایون (۶۴-۶۶)	و
۱۲۹۴	منتظری	بهزاد	ع
۱۵۶۵	امیرعلاتی	احمد	ع
۱۰۶۲	سلامی	عبداللہ	ع
۱۳۱۹	عباسی زاده	رحیم	ع
۱۵۵۳	پیمان	افسانه (۶۴-۶۶)	ع
۱۶۵۲	کوجمشکی	پروین د به ع	ع
۵۵۴۶	شنگانی خسرو شاهی	محمد و به د	و
۱۶۵۳	سینا می	ژوزف و به ع	و

درضمن در این ماه شرکت داده پردازی ایران با شما ره ۸۰۱۱ به عضویت حقوقی انجمن درآمد. بدین ترتیب، آمار اعضای انجمن تا پایان آذرماه ۱۳۶۴ به قرار زیر بود:

عادی وابسته	دانشجویی	برجسته	حقوقی	جمع کل
۳۱۴	۱۱۴	۴۲۱	۸	۸۵۸

حق عضویت سالانه: اعضا: ۱۵۰۰ ریال (شامل اشتراک ماهنامه) وابسته: ۱۵۰۰ " دانشجویی: ۵۰۰ " حق اشتراک یک ساله: گزارش کامپیوتر ۸۰۰ " هزینه پست برای اعضا: با مشترکین ۵۰۰ " مقیم خارج از کشور

لطفاً "رسید پرداخت وجه به حساب جاری شماره ۱۱۱۷ انجمن نزد بانک صادرات شعبه دانشگاه صنعتی شریف، تهران (شعبه شماره ۱۶۵۲) را به درخواست خود ضمیمه کنید. پرداخت وجه از طریق هر یک از شعب بانک صادرات در تهران و شهرستانها ممکن است. برای دریافت بروشور حاوی فعالیت های انجمن و تقاضا نامه عضویت، با ما مکاتبه کنید.



گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران

زیر نظر کمیته انتشارات انجمن

نشانی: صندوق پستی ۱۱۹۶-۱۴۱۵۵، تهران، ایران

سرپرست کمیته انتشارات و سردبیر: ابراهیم نقیب زاده، مشاور

آخرین مهلت قبول مطلب و آگهی: سومین چهارشنبه، ما قبل از انتشار نقل مطالب ما هنامۀ گزارش کامپیوتر با ذکر مأخذ آزاد است

COMPUTER REPORT, Monthly Publication of the Informatics Society of Iran (ISI)

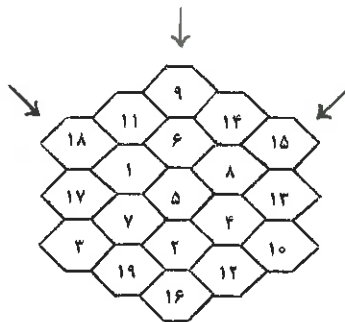
Address: P.O.Box 14155-1196, Tehran, IRAN--Editor: Ebrahim N. Mashayekh

قائم

- ۱ - رجوع کنید به ۳ سرپاشین
- ۲ - ۷ سرپالا برای عدد بخش پذیر است
- ۳ - مربع کامل
- ۴ - مکعب ۹ سرپاشین
- ۵ - با تعویض جای ارقام این عدد ۲ سرپالا بر آن بخش پذیر می شود.

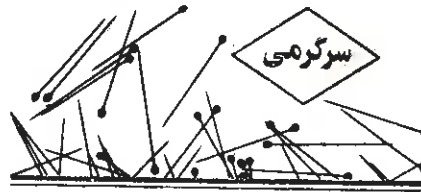
شش ضلعی جادویی دست بردار نیست!

شکل زیر، شش ضلعی جادویی را نشان می دهد که در آن اعداد ۱ تا ۱۹ به صورتی قرار گرفته اند که مجموع اعداد ردیفها در هر یک از سه جهت نشان داده شده برابر ۳۸ است:

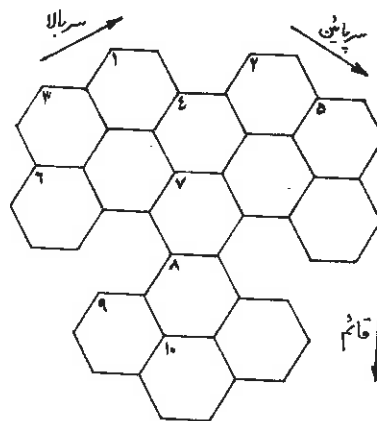


آیا می توانید اعداد ۱ تا ۱۹ را به گونه ای دیگر در خانه ها قرار دهید تا مجموع ردیفهای ۳ خانه ای برابر ۲۲، مجموع ردیفهای ۴ خانه ای برابر ۴۲، و مجموع ردیفهای ۵ خانه ای (قطرها) برابر ۶۲ گردد؟ ممکن است این معما در وهله اول مشکل به نظر برسد، اما راه ساده ای برای تبدیل پاسخ اصلی شش ضلعی جادویی به پاسخ مورد نظر در این حالت وجود دارد.

گردآوری از:
بهر روز پرها می



جدول اعداد سه طرفه



سرپالا

- ۳ - رجوع کنید به ۱۰ سرپالا
- ۶ - مکعب ۲ قائم
- ۷ - حاصل ضرب دو عدد اول
- ۹ - رجوع کنید به ۳ سرپاشین
- ۱۰ - دوسوم ۳ سرپالا

سرپاشین

- ۱ - ۲ قائم ضرب در ۳ سرپاشین تقسیم بر ۹ سرپالا ضرب در ۴ ملجم ارقام ۹ سرپالا
- ۲ - رجوع کنید به ۸ سرپاشین
- ۳ - ۹ سرپالا ضرب در ۵ قائم و ۱ قائم
- ۸ - چهار برابر ۲ سرپاشین
- ۹ - رجوع کنید به ۴ قائم

نقل قول

"پیشرفت صنعت الکترونیک، ساخت کامپیوتر، ترهای کوچک را با قیمتی ارزان امکان پذیر ساخته است. اما با وجود آن که تکنولوژی سخت افزار این ماشینها کاملاً امروزی است، تواناییها و امکانات نرم افزاری آنها یادآور تکنولوژی ۴۰ سال پیش است. ... کسانی که تجربه کامپیوتری خود را برایشان نوشتن برنامه های کوچک با زبانهای برنامه سازی ۴۰ سال پیش به دست می آورند، چیزی درباره جنبه های از کامپیوتر که به بحران نرم افزاری سالهای اخیر منجر گردیدند نخواهند آموخت و به احتمال زیاد، اشتباهات پیشینیان خود را تکرار خواهند کرد."

Shore, J. E. in *The Sachtortte Algorithm*, Viking Penguin, 1985.

هم اکنون بر اساس گزارشات منتشر شده، در اروپا بیش از یک دوجین دستگاههای کنترل کننده که بر اساس منطق نا دقیق ساخته شده اند جایگزین کارگران ما هرگشته اند. از آن جمله می توان کنترل کننده ای را ذکر کرد که پنج سال قبل در قالب سیستم طراحی شده توسط کمپانی F. L. Smith به بازار عرضه گردید. این کنترل کننده بر اساس استفاده از زیرپر-دا زنده ساخته شده و جهت کنترل کوره های سیمان بکار گرفته می شود. نمونه موفق دیگری از این کنترل کننده در صنایع چوب و کاغذ بکار گرفته شده است.

به نقل از

Chemical Engineering
June 1985

- 1: Uncertainty
- 2: Variations
- 3: Handle
- 4: Fuzzy Logic
- 5: Oxygen Concentration
- 6: Fuel Rate

تکنه

کامپیوترها همه چیز را درباره عدم قطعیتاً فرا می گیرند

ترجمه: امیر منصور فردبهبهانی

چنانچه محققینی نظیر دکتر لطفی زاده، استاد علوم کامپیوتر دانشگاه تبرکلی در کالیفرنیا به راه خود ادامه دهند، ممکن است به زودی کامپیوترها گزاره های دقیق ارسطویی (درست یا نادرست، این یا آن) را به کناری گذاشته، چندگانگیها ۲ و عدم قطعیتها را به حساب آورند. دکتر لطفی زاده، نظریه خود را که قائم به همبستگی و جذب گزاره های کیفی از قبیل "نسبتاً بلند"، "معمولی"، "کاهش یافته"، "کمی کاهش یافته"، "می باشد"، "منطق نا دقیق" می نامد. به دلیل آن که گزاره های فوق الذکر بیان نا دقیق شرایط موجود می باشد، دکتر لطفی زاده مدلسازی ریاضی جهت چنین ورودیهای گنگ و مبهمی را غیر ممکن می داند.

به عنوان نمونه، از آنجائی که مدلسازی ریاضی جهت فرآیند تولید سیمان مشکل می باشد، بکارگیری الگوریتم کنترل و نه پتانسیل کنترل تولید سیمان عمل دشواری است. علاوه بر این، هنگام کارکرد کوره، وضعیت داخل کوره از طریق اندازه گیری پارامترهای انگشت شماری نظیر غلظت اکسیژن ۵، مقدار آب و مقدار موجود، تغییرات در گشتا و رانندگی کوره و مقدار ریزش اکسید کربن در گازهای خروجی، معین می گردد. پاسخ وی جهت حل این معضل، تعریف عبارتهای گوناگون با استفاده از مجموعه های نا دقیق است. در این مجموعه ها، به هر متغیر یک ارزش درجه بندی شده ثانوی که بین صفر و یک تغییر می کند، داده می شود. به عنوان مثال، اگر غلظت اکسیژن در کوره سیمان، ۵٪ باشد، آنگاه این متغیر به قسمی تعریف می شود که ۰/۹ شایستگی (یا مناسبیت) تعلق به یک مجموعه نا دقیق "اکسیژن زیاد"، ۰/۷ شایستگی یا مناسبیت تعلق به یک مجموعه نا دقیق "اکسیژن متوسط"، و ۰/۳ شایستگی یا مناسبیت تعلق به یک مجموعه نا دقیق "اکسیژن کم" و از این دست را داشته باشد. با تصمیم گیری در مورد این که غلظت اکسیژن به کدامیک از مجموعه های نا دقیق با بدست تعلق داشته باشد و پس از بررسی مشا به دربار، وضعیت سایر متغیرهای مستقل، ممکن است سیستم کنترلی در قالب قائم به نظریه دستور العمل زیر ارائه گردد:

"اگر در صد اکسیژن زیاد است و مقدار آب آزاد موجود در گشتا و رانندگی کوره معمولی می باشد، آنگاه جریان گاز را کم کرده و به مقدار بسیار اندک نرخ سوخت مصرفی را کاهش دهید."

بنابراین اظهارات دکتر لطفی زاده، مفهوم مجموعه های نا دقیق مفهومی است که توسط آن می توان نحوه استدلال کامپیوترها را به نحوه استدلال انسانها نزدیک تر نمود.