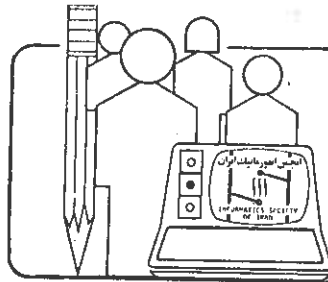


گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران



سال چهارم - شماره ۳
شماره پیاپی ۳۵

خرداد ۱۳۶۱

شماره ۵۰ ریال - اشتراک سالانه ۵۰۰ ریال



اخبار انجمن

اخبار کوتاه از فعالیتهای انجمن

همانطور که در شماره قبلی ماهنامه اعلان گردید، یک کلاس دو روزه با عنوان "تکنولوژی و سازمان حافظه کامپیوتر" برای روزهای اول و دوم تیرماه پیش بینی شده که چون تا زمان زیر چاپ رفتن این شماره از ماهنامه هنوز برگزار نشده است، گزارش آن در شماره آینده (که استثنائاً) مربوط به دوماء تیر و امرداد خواهد بود. بنظر اعضای گرامی خواهد رسید.

جلسه بعدی هیئت اجرایی انجمن انفورماتیک ایران از ساعت ۲ بعد از ظهر روز دوشنبه ۱۴ تیرماه ۱۳۶۱ در محل دانشکده ریاضی مجتمع دانشگاهی علوم (تهران، خیابان پاسداران، نرسیده به دولت) تشکیل خواهد شد. اعضای علاقمند می توانند برای مطرح نمودن سوالات و نظرات خود در این جلسه شرکت نمایند.

همچنین کمیته بازبینی واژه نامه مقدماتی کامپیوتر و انفورماتیک، که بخشی از نتایج فعالیت آن در همین شماره بنظر اعضای گرامی رسیده است، هر دوشنبه ساعت ۹ صبح در محل دانشکده ریاضی مجتمع دانشگاهی علوم جلسه دارد و شرکت در جلسات آن برای کلیه علاقمندان آزاد است.

شرکت باروز (وابسته به بنیاد مستضعفان) اخیراً موافقت نموده است که امکاناتی را در جهت دایر نمودن کتابخانه انجمن در اختیار ما قرار دهد. ضمن تشکر از مسئولان این شرکت، با اطلاع اعضای گرامی می رسانیم که از اوایل امردادماه ۱۳۶۱، خواهند توانست از کتابخانه انجمن انفورماتیک ایران در محل شرکت باروز (تهران، خیابان حافظ، تقاطع خیابان جمهوری، جنب پارک هتل) استفاده نمایند. با اطلاع علاقمندان بازدیدهای علمی انجمن می رساند که برنامه بازدیدها موقتاً به خاطر نزدیک شدن تابستان متوقف شده است و در اولین فرصت مجدداً برقرار خواهد گردید. ○



تصویر روی جلد: میناسبت اخبار مربوط به آدمکهای مصنوعی

در این شماره

- | | |
|----------------|---|
| اخبار انجمن .. | اخبار کوتاه از فعالیتهای انجمن |
| اخبار جهان ... | چند خبر درباره آدمکهای مصنوعی صنعتی - شرکتهای آمریکائی به جنگ ژاپن می روند - پشتیبانی دولت آمریکا از آی بی ام - و ... |
| واژه ها | تغییرات و اضافات در واژه نامه مقدماتی |
| مقاله ۱ | الگوریتم - قسمت دوم (ترجمه بهروز پرها می) |
| مقاله ۲ | مروری بر کاربرد تکنولوژی الکترونیک نوری در ساخت نمایشگرهای ... |
| مقاله ۳ | الفبا عددی (ترجمه بهروز پرها می و حسن کاشوریان) |
| مقاله ۴ | کامپیوترهای کوچک باروز سری ال (ترجمه و گردآوری محمدرضا حبیبیان) |
| مقاله ۵ | مسائلی در رابطه با ساخت فیزیکی مدارهای الکترونیک رقمی (فرهاد صاحبان) |
| مقاله ۶ | روشهای مختلف فراخوانی زیرروالها و نواقص هریک (رضا مومتمن) |
| مقاله ۷ | طرح چند زبان کردن چاپگرهای کامپیوتری "ال ای ۳۶" و "ال ای ۸۰" (حسین فیروزبخت و امیراسعد انزانی) |
| مقاله ۸ | آشنائی با زبان برنامه نویسی "پرولوگ" (ابراهیم مشایخ) |



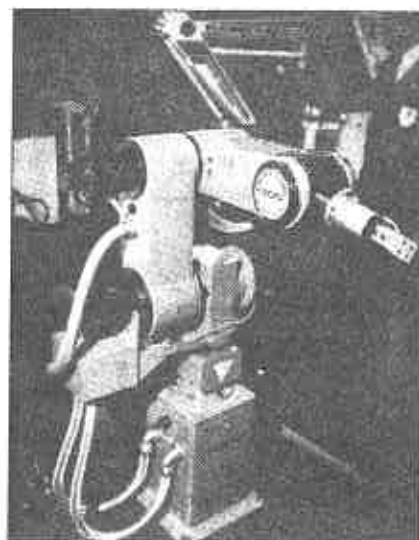
ژاپن و آدِمکهای مصنوعی

وزارت بازرگانی بین المللی ژاپن اخیراً "کمیته ویژه ای را" مأمور بررسی اثرات اقتصادی، اجتماعی و فنی آدِمکهای مصنوعی صنعتی نموده است. این کمیته، که قرار است گزارش مطالعات خود را در سال ۱۹۸۳ ارائه نماید، از اقتصاددانان، مهندسان، جامعه‌شناسان، نماینده‌هایی از بین استفاده‌کنندگان از کامپیوتر، و صاحبان امور کارگری و بهداشت تشکیل خواهد شد.

یک سخنگوی وزارت بازرگانی بین المللی ژاپن اظهار داشته است که جزئیات دستور کار این کمیته فعلاً فاش نخواهد شد ولی "کلاً" ماسک‌هایی در زمینه اثرات آدِمکهای مصنوعی صنعتی بر وضع روحی و استفاده‌ای کارگران، ساخت صنعتی ژاپن، و تکنولوژیهای ویژه در زمینه پزشکی و صنایع اتمی مورد بحث قرار خواهند گرفت.

تعداد آدِمکهای مصنوعی صنعتی در ژاپن حدود ۱۴ هزار برآورد می‌شود که این رقم نزدیک به هفتاد درصد تعداد کل آدِمکهای مصنوعی جهان را تشکیل می‌دهد. گفته می‌شود که ژاپن برای شرکت در نمایشگاه آدِمکهای مصنوعی که پائیز آینده در شهر لندن برگزار خواهد شد دعوت شده است. لازم به تذکر است که ژاپن اخیراً تعداد زیادی آدِمک مصنوعی برای ساخت اتومبیل را به شوروی صادر کرده است.

از "کامپیوتینگ"، ۱۴ ژانویه و ۱۸ فوریه ۱۹۸۲



آدِمکهای مصنوعی در انگلستان

گزارشهای منتشر شده نشان می‌دهند که تعداد آدِمکهای مصنوعی انگلستان طی سال ۱۹۸۱ حدوداً "دو برابر شده و از ۳۷۱ به ۷۱۳ رسیده است. این امر مستلزم تقریباً ۲۰ میلیون پوند سرمایه‌گذاری بوده است. در حال حاضر انگلستان از نظر تعداد آدِمکهای مصنوعی مقام پنجم را در جهان دارا است. پیش‌بینی می‌شود که تا سال ۱۹۸۳، تعداد آدِمکهای مصنوعی در انگلستان به حدود ۲۰۰۰ برسد و در این راه ممکن است علاوه بر پروژه‌های مشترک با شرکتهای ژاپنی که فعلاً در جریان هستند، همکاری با شرکتهای آمریکایی نیز مورد توجه قرار گیرد. بین کشورهای اروپایی، سوئد و آلمان غربی از نظر بکارگیری آدِمکهای مصنوعی پیشرفته‌تر از انگلستان هستند.

از "کامپیوتینگ"، ۱۱ فوریه ۱۹۸۲

آی بی ام و آدِمک مصنوعی ژاپنی

اخیراً "آی بی ام" با ارائه یک آدِمک مصنوعی صنعتی ساخت ژاپن به بازار، دست اندرکاران این رشته را دچار حیرت ساخته است. این آدِمک مصنوعی که "سیستم ساخت ۷۵۳۵" نام دارد، از طریق یک زبان برنامه نویسی جدید توسط کامپیوتر شخصی آی بی ام هدایت خواهد شد. این سیستم دارای یک دستگاه کنترل، یک تابلوی فرمان، و یک بازوی مکانیکی مفصل دار با قابلیت حرکت در چهار جهت است و ۲۸۵۰۰ دلار قیمت دارد. آنچه که تعجب برانگیز است ورود آی بی ام به بازار آدِمکهای مصنوعی در زمانی است که هنوز وسعت چندان ندارد (فروش این آدِمکها در آمریکا برای سال ۱۹۸۱ معادل ۱۵۰ میلیون دلار برآورد شده است)، آنهم با محصولی که خودش نساخته است. لازم به توضیح است که در حال حاضر آدِمک مصنوعی بسیار پیشرفته "آر اس ۱"، که ساخت خود آی بی ام است بصورت آزمایشی در اختیار بعضی از مشتریان آن شرکت قرار گرفته است.

از "کامپیوتینگ"، ۱۴ مارس ۱۹۸۲

شرکتهای آمریکایی به جنگ ژاپن می‌روند

نگرانی شرکتهای آمریکایی در مورد توسعه نفوذ ژاپن در بازارهای الکترونیک آن کشور سبب گردید که اخیراً "پنجگاه نماینده" این شرکتها در یک گردهم آیی مؤثر، امکانات همکاریهای پژوهشی را با تأسیس شرکتی بنام "نوآوریهای تکنولوژی ریزالکترونیک و کامپیوتر" مورد بررسی قرار دهند. آقای ویلیام نوریس، رئیس شرکت کنترل دیتا، که یکی از پیشگامان

اصلی اجرای این طرح است، عقیده دارد که اگر آمریکا در این زمینه به جلو حرکت نکند، در بازارهای داخلی و بین المللی مدمات جبران ناپذیری خواهد دید. در بین شرکت کنندگان در این گردهم آیی، نمایندگان یونیواک، شینال یسی - کانداکتور، هاریس، دیجیتال ایکوپیمنت، آدونسدماپیکرو دیوایس، باروز، زیراکس، و تگزاس اینسترومنتس دیده می‌شدند. شرکت آی بی ام، بدلیلی که هنوز مشخص نشده‌اند، در این گردهم آیی حضور نداشت.

از "کامپیوتینگ"، ۴ مارس ۱۹۸۲

پشتیبانی دولت آمریکا از آی بی ام

بدنبال پس گرفتن دعوی "غذ تراست" دولت آمریکا بر علیه آی بی ام، اینک برخی از مسئولان آن کشور به اروپا سفر کرده‌اند تا با وارد آوردن فشارهای پنهانی و آشکار، آی بی ام را در دعوی مشابهی که در اروپا مطرح است یاری نمایند. دولت ریگان بر این عقیده است که وارد آمدن هر نوع فشاری به آی بی ام در بازار اروپایی، به سایر شرکتهای آمریکایی نیز لطمه خواهد زد و میزان صادرات آمریکا را کاهش خواهد داد. ولی برخی از شرکتهای کامپیوتری آمریکا با دولت آن کشور هم عقیده نیستند و مثلاً "شرکتهای آمدال و ممورکس مدارکی را دال بر تخلف آی بی ام در زمینه جلوگیری از رقابت آزاد در بازار کامپیوتر، به مسئولان جامعه اقتصادی اروپا ارائه داده‌اند.

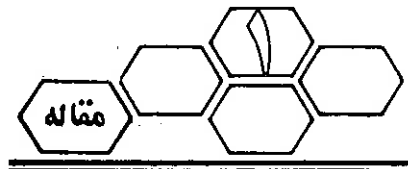
از "کامپیوتینگ"، ۲۵ فوریه ۱۹۸۲

جدیدترین آتِرا کامپیوتر

موسسه تحقیقاتی کری اخیراً "جدیدترین آتِرا کامپیوتر خود را بنام Cray X-MP معرفی نموده است. قابلیت داده‌پردازی این ماشین پنج برابر کامپیوتر "کری-۱" می‌باشد. کارخانه سازنده ادعا می‌کند که این ماشین جدید سریعترین سیستم کامپیوتری همه منظوره جهان است. زمان اجبرای دستورهای ساده در این ماشین ۹/۵ میلیاردیم ثانیه (سرعت بیش از صد میلیون دستورالعمل در ثانیه) و قیمت آن بین ۱۱/۴ تا ۱۴/۵ میلیون دلار است. اولین سیستم از این نوع در سال ۱۳۶۲ به بازار خواهد آمد.



فراموش کنید که انجمن انفورماتیک ایران به همکاری شما نیاز دارد.



الگوریتم (قسمت دوم)

یادداشت سردبیر: آنچه که در زیر بنظر خوانندگان گرامی می رسد قسمت دوم از یک مقاله توصیفی چند قسمتی است که توسط آقای دکتر بهروز پرها می، از مقاله‌ای با همین نام نوشته دکتر دانالد کنوت، به فارسی ترجمه شده است. در شماره گذشته پنج بخش اول مقاله را با عناوین زیر خواندیم:

- ۱- کلیات
 - ۲- جستجو در حافظه کامپیوتر
 - ۳- مزایای وجود ترتیب
 - ۴- تجزیه و تحلیل کتی
 - ۵- بهتر از بهترین!
- اینک مطلب را با بحثی در مورد روشهای جستجوی درختی دنبال می کنیم.

۶- جستجو در درختهای دودوئی

روش جستجوی دودوئی را می توان از راه دیگری نیز بهتر کرد و آن حذف نمودن فرضی است که هریک از کلیدهای موجود در جدول با احتمال مساوی بتواند کلمه مورد نظر باشد. اگر بدانیم که بعضی از کلیدها با احتمال خیلی بیشتری مورد جستجو قرار می گیرند، می توان الگوریتم را طوری طرح کرد که ابتدا این کلیدهای محتملتر را مورد بررسی قرار دهد.

قبل از آنکه بیشتر وارد این بحث شویم، بدنیست که ابتدا از دید دیگری به جستجوی دودوئی نگاه کنیم. مجموعه‌ای از ۳۱ کلمه را که بیش از کلمات دیگر در زبان انگلیسی بکار می روند درنظر بگیرید. اگر این کلمات با ترتیب الفبائی در جدولی بصورت KEY[۱]، KEY[۲]، ...، KEY[۳۱] قرار داشته باشند، الگوریتم B ابتدا کلمه X را با عنصر وسطی KEY[۱۶]، یعنی کلمه "I"، مقایسه می کند. اگر X در ترتیب الفبائی قبل از "I" قرار داشته باشد، مقایسه بعدی با KEY[A]، یعنی کلمه "BY"، صورت خواهد گرفت و اگر X بعد از "I" قرار داشته باشد، مقایسه بعدی با KEY[۲۴]، یعنی کلمه "THAT"، خواهد بود.

بعبارت دیگر، می توان گفت کلمه الگوریتم B جدول کلمات را بصورت یک سازمان درخت مانند وارونه درنظر می گیرد که از قسمت فوقانی آن شروع کرده و هرگاه X کوچکتر باشد، بطرف پایین و چپ و هرگاه X بزرگتر باشد، بطرف پایین و راست حرکت می کند (به شکل ۳ مراجعه کنید). بسادگی می توان دید که هر الگوریتم جستجوی

مقایسه‌ای در یک جدول ترتیب دار، بوسیله درخت دودوئی مشابهی قابل توصیف است.

تعریف درخت دودوئی در الگوریتم B بصورت ضمنی از طریق عملیات محاسباتی بر روی ۲۰، ۲ و ۱ صورت می گیرد. می توان این کار را بصورت صریح با ذخیره کردن اطلاعات لازم در خود جدول انجام داد. باین منظور، فرض کنید که LEFT[j] مکانی را نشان دهد که درحالتی که X کوچکتر از KEY[j] باشد برای عمل مقایسه بعدی انتخاب می شود. بهمین ترتیب فرض کنید کلمه RIGHT[j] مکان کلید مورد مقایسه بعدی را در حالتی که X بزرگتر از KEY[j] باشد مشخص می سازد.

بعنوان مثال، در مورد جستجوی دودوئی در جدولی با ۳۱ کلمه، مقدار LEFT[۱۶] مساوی ۸ و RIGHT[۱۶] مساوی ۲۴ خواهد بود، چون جستجو از KEY[۱۶] شروع شده و بعد از طریق KEY[A] یا KEY[۲۴] ادامه می یابد. درحالتی که جستجو موفقیت آمیز نباشد، پس از تعیین اینکه X کوچکتر از KEY[j] یا بزرگتر از آن است، مقیدار LEFT[j] یا RIGHT[j] مساوی صفر خواهد بود. در شکل ۲، این نوع صفرها بوسیله گره‌های کوچک مربع شکل در قسمت پایتین درخت نشان داده شده‌اند.

محل اولین کلید مورد مقایسه در درخت جستجوی دودوئی اصطلاحاً "ریشه" نامیده می شود. در مثال جدول ۳۱ کلمه‌ای که بررسی کردیم، ریشه درخت عنصر شماره ۱۶ بود. می توان الگوریتمهای جستجوی طرح نمود که مقایسه را لزوماً از عنصر KEY[۱۶] شروع نمی کنند و درحالتی که

بعضی از کلمات مورد جستجو خیلی بیشتر از سایرین مورد استفاده قرارگیرند، ممکن است این روشها از الگوریتم B بهتر باشند. یک روش کلی جستجوی درختی بصورت زیر است.

الگوریتم C: جستجوی درختی

C1 - [مقدار اولیه دادن] - مقدار j را مساوی شماره محل قرارگرفتن ریشه درخت جستجوی دودوئی قرار بده.

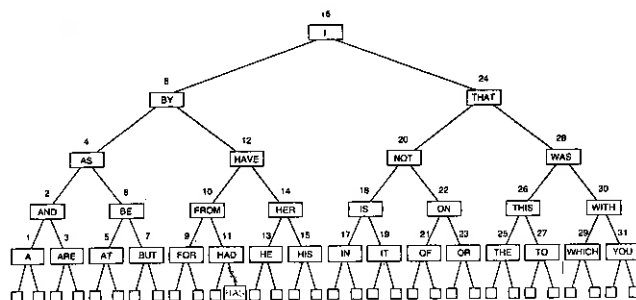
C2 - [آیا جستجو موفق است؟] - اگر j = ۰ است، مقدار j را در خروجی تحویل بده و عملیات را تمام کن.

C3 - [مقایسه] - اگر X = KEY[j] است، مقدار j را در خروجی تحویل بده و عملیات را تمام کن. اگر X < KEY[j] است، عمل LEFT[j] را انجام بده و بقیه C2 برگرد. اگر X > KEY[j] است، عمل RIGHT[j] را انجام بده و بقیه C2 برگرد.

نحوه کار الگوریتم C مشابه روش یک کتاب خودآموز است که در آن، بسته به پاسخی که خواننده به سوالات خاصی در هر صفحه می دهد، شماره صفحه بعدی برای مطالعه مشخص می گردد. این روش را می توان در مورد هر درخت دودوئی که در آن به ازای تمام مقادیر j، عناصر قابل دسترسی از طریق LEFT[j] همگی از KEY[j] کوچکتر و عناصر قابل دسترسی از طریق RIGHT[j] همگی از KEY[j] بزرگتر باشند بکار برد. چنین درختی را یک درخت جستجوی دودوئی می نامیم.

یکی از مزایای الگوریتم C نسبت به الگوریتم B این است که نیاز به عملیات

ورق بزنید



شکل ۳- الگوریتم B بصورت ضمنی با یک درخت جستجوی دودوئی کار می کند. این شکل مسیری را که الگوریتم B در جستجوی جدولی متشکل از متداولترین ۳۱ کلمه در زبان انگلیسی تعقیب می کند نشان می دهد. الگوریتم از "ریشه" یا نقطه فوقانی درخت شروع کرده و کلمه X را با عنصر میانی یعنی کلمه "I" مقایسه می کند. اگر X در ترتیب الفبائی قبل از "I" قرار داشته باشد، جستجو در شاخه سمت چپ درخت ادامه می یابد و اگر X بزرگتر باشد، جستجو در شاخه سمت راست تعقیب می شود. مثلاً اگر X کلمه "FROM" باشد، ابتدا معلوم می شود که X کوچکتر از "I" است، سپس معلوم می شود که X بزرگتر از "BY" است، بعد معلوم می شود که X کوچکتر از "HAVE" است، و بالاخره معلوم می شود که X مساوی "FROM" است. اگر X در جدول نباشد، جستجو در یکی از ۲۲ نقطه انتهائی درخت (گره‌های مربع شکل) متوقف می شود. اگر شاخه‌های درخت مستقیماً در حافظه کامپیوتر نمایش داده شوند (بجای آنکه مانند روش الگوریتم B بصورت ضمنی از طریق محاسبه نقطه میانی ناحیه مورد جستجو تعیین شوند)، عمل جستجو کمی سریعتر انجام خواهد گرفت و علاوه عمل اضافه کردن عناصر جدید به جدول ساده‌تر خواهد شد. مثلاً برای اضافه کردن کلمه "HAS" به درخت، کافی است آن را در ترتیب الفبائی که ذکر شد، بجای یکی از نقاط انتهائی (گره‌های مربع شکل) قرار دهیم.

کنیم و بنا بر این درخت مورد بحث نمی‌توانست بهترین درخت جستجو باشد.

بهترین ترتیب، در زیردرخت فوق، زیردرخت کوچکتری که در سمت راست کلمه "FOR" واقع شده است باید بهترین روش جستجوی ۱۱ کلمه "FROM"، "HAD" و غیره تا "NOT" را مشخص نماید. هر زیر درخت متناظر با مجموعه‌ای از کلمات مجاور صورت، $KEY[i]$ ، $KEY[i+1]$ ، ...، $KEY[j]$ است، که در آن رابطه $i \leq j \leq n$ برقرار است. پس می‌توان به ترتیب صعودی مقدار i - j پیش رفت و از روی بهترین زیردرختهای کوچکتر، بهترین زیردرختهای بزرگتر را پیدا کرد. در ازای هر انتخاب برای مقادیر i و j ، تعداد $j-i+1$ - انتخاب برای ریشه زیردرخت موجود است. چون برای هر یک از این انتخابها، بهترین زیردرختهای سمت چپ و راست آن در محاسبات قبلی تعیین شده‌اند، بسادگی می‌توان بهترین انتخاب را برای ریشه درخت بدست آورد.

با استفاده از این روش می‌توان بهترین درخت جستجوی دودویی را در مورد n کلید با بسامد معلوم با حدود n^2 عمل محاسباتی پیدا کرد. در واقع من توانسته‌ام کسبه روش فوق را از این هم بهتر بکنم و تعداد عملیات لازم را به n^2 کاهش دهم. بنا بر این در مورد ۳۱ کلمه متداول در انگلیسی، می‌توان بهترین درخت جستجوی دودویی را، بجای ۱۴ کوادرلیون قدم، فقط در ۹۶۱ مرحله بدست آورد.

باید متذکر شد که بحثهای فوق در مورد تعدادی الگوریتم است که تنها هدفشان بدست دادن بهترین درخت جستجوی دودویی است. بعبارت دیگر، نتیجه حاصل از این الگوریتمها خود الگوریتمی برای یک مسأله دیگر است. این مثال بخوبی نشان می‌دهد که چرا رشته علوم کامپیوتر با این سرعت بصورت یک نظام

از نقطه نظر ریاضی، پیدا کردن بهترین درخت جستجو برای مجموعه‌ای از n کلمه با بسامدهای معلوم کار ساده‌ای است، چون تعداد درختهای جستجوی ممکن محدود است و فقط کافی است از بین همه این درختها، آن را که از همه بهتر است انتخاب کنیم. ولی این روش عملی نیست، چون تعداد درختهای دودویی ممکن با n عنصر مساوی $(n+1)!/n!$ است. این فورمول نشان می‌دهد که تعداد درختهای دودویی ممکن واقعا "خیلی زیاد است؛ یعنی تقریبا" مساوی $n^n/\sqrt{n}$.

بعنوان مثال، اگر $n=31$ باشد، تعداد درختهای دودویی ممکن برابر $14'544'636'039'226'909$ می‌شود و هر یک از این ۱۴ کوادرلیون درخت می‌تواند به ازای مقادیر خاصی از بسامدها، بهترین درخت ممکن باشد. بنا بر این این سوال پیش می‌آید که چگونه می‌توان ثابت کرد که درخت شکل ۴ برای بسامدهای داده شده بهترین درخت ممکن است. حتی سریعترین کامپیوترهای امروزی هم دارای سرعت کافی برای بررسی تمام این ۱۴ کوادرلیون درخت نیستند، چون حتی اگر در هر میلیونیم ثانیه یکی از این درختها مورد بررسی قرار گیرد، درنظر گرفتن همه آنها نیاز به ۴۶۰ سال وقت خواهد داشت.

ولی یک اصل مهم وجود دارد که این کار را عملی می‌کند: هر زیردرخت از بهترین درخت باید خود، برای کلماتی که در آن ظاهر می‌شوند، بهترین درخت جستجو باشد. در مورد بهترین درخت برای متداولترین ۳۱ کلمه انگلیسی، زیردرختی که در سمت چپ کلمه "OF" قرار دارد باید متناظر با بهترین روش جستجوی جدولی حاوی کلمات "A"، "AND" و غیره تا "NOT" باشد، چون اگر روش بهتری برای جستجو در بین این کلمات موجود بود، می‌توانستیم درخت کلی را نیز بهتر

محاسباتی ندارد و در نتیجه توسط کامپیوتر سریعتر اجرا می‌شود. ولی مهمترین مزیت الگوریتم C این است که سازه درختی دارای قابلیت انعطاف بیشتری است و با استفاده از آن می‌توان عناصر جدول را به هر ترتیبی ذخیره نمود. دیگر لازم نیست که $KEY[1]$ حتماً از $KEY[2]$ کوچکتر باشد، و به همین ترتیب تا $KEY[n]$.

تا زمانی که اشاره‌گرهای ۱۹ LEFT و RIGHT هر کدام یک درخت جستجوی دودویی صحیح را مشخص کنند، ترتیب ذخیره کردن عناصر هیچ اهمیتی ندارد. این بدان معنی است که می‌توان بدون حرکت دادن عناصر موجود، عنصر جدیدی به جدول اضافه نمود. مثلاً برای اضافه کردن کلمه "HAS" به جدول ۳۱ کلمه‌ای، کافی است که عمل $KEY[32] \leftarrow KEY[31]$ را انجام داده و $RIGHT[j]$ را، که در آن j محل کلمه "HAD" است، از صفر به ۳۲ تغییر دهیم.

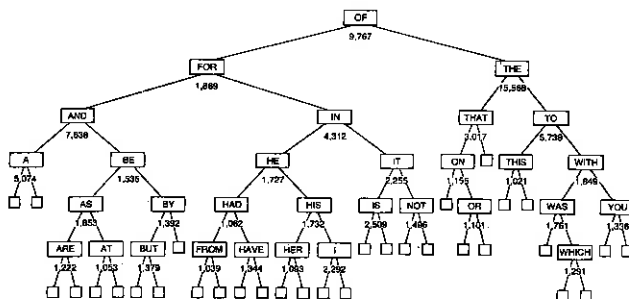
ممکن است فکر کنید که چنین اضافاتی به قسمت پائین درخت، توازن آن را از بین خواهد برد. ولی بسادگی می‌توان بسامد استفاده از استدلال ریاضی ثابت کرد که اگر عناصر جدید بصورت تصادفی به درخت اضافه شوند، درخت نتیجه شده با احتمال قریب به یقین بصورت متوازن باقی خواهد ماند.

۷ - بهترین درختهای جستجوی دودویی

چون الگوریتم C در مورد هر درخت جستجوی دودویی قابل استفاده است می‌توان درخت جستجو را طوری ترتیب داد که کلیدهایی که بیش از همه مورد استفاده قرار می‌گیرند قبل از سایرین بررسی شوند. چنین ترتیبی باعث خواهد شد که کامپیوتر بطور متوسط زمان کوتاهی را برای جستجو صرف نماید، ولی البته زمان جستجو در بدترین حالت را تقلیل نمی‌دهد.

شکل ۴ بهترین درخت جستجوی دودویی را برای متداولترین ۳۱ کلمه در زبان انگلیسی نشان می‌دهد که با کمک بسامدهای کلمات ساخته شده است. تعداد متوسط مقایسه‌های لازم برای پیدا کردن کلمه x در این بهترین درخت مساوی $1.2/447$ است، درحالی که این تعداد متوسط در مورد درخت جستجوی متوازن مساوی $4/393$ می‌باشد.

لازم به تذکر است که اگرچه این بهترین درخت از روی بسامد کلمات ساخته شده است، ولی اولین عمل مقایسه در آن با کلمه "THE" صورت نمی‌گیرد. با وجودی که کلمه "THE" متداولترین کلمه در زبان انگلیسی است، محل آن در ترتیب الفبائی آنقدر از نقطه میانی دور است که نمی‌توان آن را بهترین کلمه برای استفاده بعنوان ریشه درخت دانست.



شکل ۴ - این درخت جستجوی دودویی بهترین روش سازمان دادن متداولترین ۳۱ کلمه در زبان انگلیسی را از روی بسامد نسبی کلمات نشان می‌دهد. بسامد هر کلمه در شکل بوسیله عددی در زیر آن نشان داده شده است. این درخت مانند درخت جستجوی دودویی، که بطور ضمنی توسط روش جستجوی دودویی تعریف می‌شود (شکل ۳)، متوازن نیست و بنا بر این باعث می‌شود که در بعضی موارد جستجو طولانی‌تر شود. بعنوان مثال، یافتن کلمه "FROM" در این درخت در شش مرحله صورت می‌گیرد، در حالی که در مورد شکل ۳ فقط چهار مرحله لازم بود. ولی بطور متوسط جستجو در این درخت سریعتر است، چون کلمات متداولتر زودتر بررسی می‌شوند.

علامت دریافتی، می باشد. قبلاً از چنین روشی با لامپهای نئون استفاده می شده، که از نقاط ضعف آنها نیاز به ولتاژ کار بالا و قابلیت اطمینان تردید آمیزشان بوده است.

لامپهای ال ای دی تدریج در بسیاری زمینه ها جایگزین انواع نئون می گردند

و این بیشتر بدلیل طول عمر و دوام زیاد آنها (تا چهار هزار ساعت) است. از عوامل مؤثر دیگر در توسعه کاربرد این دیود های نوری، می توان پائین بودن ولتاژ کارشان و قابلیت تطابق با مدارهای منطقی تی تی ال را ذکر کرد. میزان خوانندگی بودن علائم هم (که بستگی به شدت نور ساطع شده از دیود دارد) نسبت به آغساز پیدایش اینگونه دیودها خیلی بهتر شده و به حدود ۱۰ برابر میزان اولیه رسیده است. در حال حاضر این لامپها در قطره های ۱/۹، ۳، ۵ و میلی متر و همچنین در انواع مربع شکل برای برخی کاربردها موجودند.

امروزه شرکت های سازنده دیودهای ساطع نور آنها را در رنگهای گوناگون (نارنجی، زرد، سبز، و انواع معمولی قرمز رنگ) می سازند. همچنین انواع دورنگ برای کاربردهای بخصوص، دیودهای بسا جریان ثابت، دیودهای نوری مقاومتی، انواع دیود چشمک زن همه و همه نتیجه سه ترکیب تکنولوژی ال ای دی با تکنولوژی سیستم جهت حداقل نمودن تعداد اجزای خارجی و ساده کردن کاربرد این نمایشگرها است.

دیودهای ساطع نور در ساخت نمایشگر های هفت قطعه ای^{۱۲} برای نشان دادن ارقام لاتین از صفر تا ۹ در انواع دستگاه های رقمی بکار می روند. اولین نمایشگرهای رقمی از نوع لامپهای رشته ای، با چند رشته در سطوح مختلف، بودند که بخاطر درخشندگی قابل ملاحظه، دامنه عملکرد وسیع از نظر درجه حرارت محیط، و همچنین ارائه الوان گوناگون و قابلیت کار با ولتاژ پائین جریان مستقیم، کاربردهای فراوانی یافتند. البته محدود بودن عمر رشته سیم یا فیلامن درون این لامپها از مشکلات اصلی کاربردشان محسوب می - گردید.

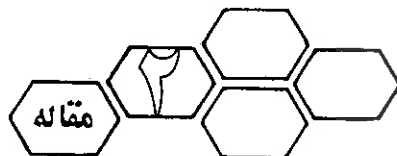
نوع دیگری از نمایشگرها عناصر بسا تخلیه گازی هستند که دارای درخشندگی زیاد ناشی از نور نارنجی گاز نئون درون لامپ هستند. این نور از پلاسمای احاطه کننده الکترودها تحت ولتاژ ۱۳۰ تا ۱۵۰ ولت تولید می گردد. همین نیاز به بالا بودن ولتاژ خود از معایب این عناصر بشمار می رود. البته در دستگاههای بزرگ این مشکل مهمی نیست، چرا که می توان با استفاده از محرکها و مدارهای ورق بزنید

۲۲/۹۰۷ عمل مقایسه برای هر جستجو لازم باشد. بنابراین حتی بدترین حالت الگوریتم C هم به بدی بدترین حالت الگوریتم A نیست. ●

(ادامه دارد)

یادداشت

۱۹ - اشاره گر Pointer



مروری بر کاربرد تکنولوژی الکترونیک نوری در ساخت نمایشگرهای الفبا عددی

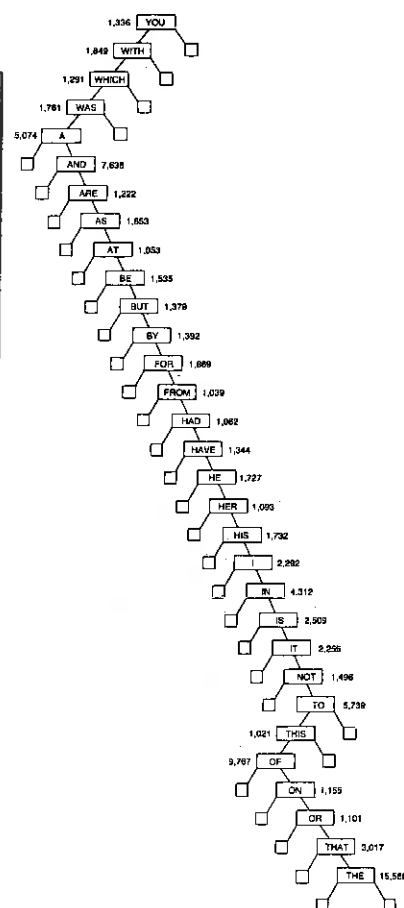
امروزه از روشهای متفاوتی برای نمایش اطلاعات خارج شده از کامپیوترها استفاده می شود. چراغهای علامت دهنده، نمایشگرهای عددی و الفبا عددی^۱، و بالاخره لامپهای با اشعه کاتدی از جمله وسایل رایج برای نمایش اطلاعات خروجی هستند. هر چند تکنولوژی ساخت لامپهای با اشعه کاتدی (با سی آر تی^۲) نسبتاً قدیمی بشمار می رود، ولی هنوز هم متداولترین و با صرفه ترین روش برای نمایش اطلاعات تغییرپذیر در حجم زیاد است. بعلاوه می - توان از سی آر تی برای نمایش نما و بر معمولی و رنگی و نیز رسم شکلها و نمودارهای چند رنگ با برچسبهای الفبا عددی در کاربردهای مختلف سود جست.

بهر حال، با وجود نفوذ و گسترش کاربرد های سی آر تی، مخصوصاً در پایانه کامپیوتر - های اداری و خانگی، رقباتی برای آن در عرصه الکترونیک نوری^۳ پدید آمده اند. نمایشگرهای الفبا عددی چند حرفی با کنترل ریزپردازنده ای، دارای نسبت کارآیی به هزینه بسیار بالایی هستند. تکنولوژیهای "ال ای دی"^۴ و "ال سی دی"^۵ خیلی سریع به بازار مصرف نفوذ کرده اند و نمایشگرهای بسیاری از ماشینهای حساب و وسایعهای رقمی امروزی از یکی از این دو نوعند.

البته علاوه بر دو شیوه نمایش رقمی فوق، که از سایر روشها شناخته شده ترند، روشهای دیگری نیز وجود دارند: نمایشگرهای پلاسمای^۶، با تخلیه گازی مستوی^۷، فلوشو - رمان^۸، انتهایی^۹، و الکترو لومینسان^{۱۰}.

در این مقاله روشهای مختلف موجود بررسی و با یکدیگر مقایسه می شوند. ساده ترین نوع نمایشگرهای الکترونیکی لامپهای تک نقطه ای هستند که تنها دارای دو وضعیت روشن و خاموش، بر حسب وضعیت

مستقل توسعه یافته است. در مطالعه اینکه چگونه از کامپیوتر بخوبی استفاده کنیم، مسائلی بروز می کنند که بخودی خود جالب هستند و حل بسیاری از این مسائل نیاز به مجموعه ای از مطالب و روشهای جدید و وابسته بهم دارد.



شکل ۵ - این درخت بدترین حالت در جستجوی دودویی را برای جستجو در میان متداولترین ۳۱ کلمه در زبان انگلیسی نشان می دهد. در این حالت، مزایای سازه درختی از بین رفته، چون یکی از شاخه های ممکن در مورد هر مقایسه مسدود است.

تفکر در باره بدترین درخت جستجوی دودویی ممکن برای متداولترین ۳۱ کلمه در انگلیسی هم جالب و هم آموزنده است، چون نشان می دهد که بدترین وضعیت الگوریتم C چیست. مانند حالت بهترین درخت، روشی نیز برای پیدا کردن بدترین درخت در n قدم وجود دارد. در مورد بسامدهای داده شده در شکل ۴، استفاده از بدترین درخت جستجوی دودویی باعث می شود که الگوریتم C بطور متوسط ۱۹/۱۵۸ عمل مقایسه برای هر جستجو انجام دهد. در مقام مقایسه، بدترین وضعیت قرار گرفتن کلیدها در مورد الگوریتم A سبب می شود که بطور متوسط

یادداشتها

- ۱ - الفباعدی Alphanumeric
- ۲ - لامپ با اشعه کاتدی CRT = Cathode-Ray Tube
- ۳ - الکترونیک نوری Optoelectronics
- ۴ - دیود ساع نور LED = Light-Emitting Diode
- ۵ - نمایشگر با کریستال مایع LCD = Liquid Crystal Display
- ۶ - نمایشگر پلاسمایی Plasma display panel
- ۷ - تخلیه گازی مستوی Planar gas discharge
- ۸ - فلئوئوسان Fluorescent
- ۹ - التهابی Incandescent
- ۱۰ - الکترو لومینسان Electroluminescent
- ۱۱ - تی تی ال TTL = Transistor-Transistor Logic
- ۱۲ - نمایشگر هفت قطعه ای Seven-segment display
- ۱۳ - ترمیونیک Thermionic
- ۱۴ - فاصل Interface

ترجمه آزاد: بهروز پیرهامی و حسن کاتوینان
از مجله "الکترونیک و قدرت"، ژوئن ۱۹۸۰



اوسال مطالب برای "گزارش کامپیوتر"

از اعضای انجمن انفورماتیک ایران و سایر افراد علاقمند به همکاری با "گزارش کامپیوتر" تقاضا داریم به نکات زیر توجه کنند:

۱ - اخبار و وقایع محل کار خود در زمینه انفورماتیک را حتی الامکان همراه با اسناد و مدارک مربوطه برای ما بفرستید. مثالهایی از مطالب مناسب عبارتند از: انعقاد یا نسخ قراردادهای، طرح و پیاده کردن سیستمها، کاربردهای نوین ویرگزار، سمینارها و سخنرانیها.

۲ - اگر در روزنامهها، مجلات، کتابها و سایر نشریات به مطالب جالبی برخوردید، بریده یا فتوکپی تمیز و خوانای آنها را با ذکر مشخصات کامل مآخذ (نام نشریه، تاریخ انتشار، ناشر، محل انتشار، شماره صفحه) برای ما بفرستید.

۳ - مقالات باید بصورت ماشین شده (با ذکر مشخصات مآخذ و نام نویسنده و ضمیمه کردن نسخه ای از اصل مطلب در مورد مقالات ترجمه شده) در یک روی کاغذ و در دو نسخه ارسال شوند. ماهنامه "گزارش کامپیوتر" در قبول یا رد مقالات رسیده، خلاصه کردن آنها و هر نوع حک و اصلاح دیگر آزاد است. در صورت تمایل میتوانید قبل از تهیه مقاله، تلفنی با ما در مورد آن مشورت کنید.

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

این کار ولتاژ تحریکی از ۴۰۰ تا ۶۰۰ ولت لازم دارد. ولی ساده بودن ساخت این وسایل همراه با به بازار آمدن مدارهای مجتمع محرک با قابلیت های لازم در آینده نزدیک، اهمیت نمایشگرهای الکترو لومینسان را افزایش خواهد داد.

در جایی که نیاز به نمایش دهنده های زیاد با قالبی فشرده بر روی صفحه وجود داشته باشد، لزوماً "مآخذ" کار بر برد لامپهای با اشعه کاتدی، یا سی آر تی، به میان می آید. لامپهای با پرده مسطح، که سیکتر و کوچکتر از انواع متداول در پایانه های نمایشی فعلی هستند، در آینده مزایای چشمگیری را ارائه خواهند داد. صفحات پلاسمایی با جریان متناوب یا جریان مستقیم نیز در این قبیل موارد قابل استفاده اند.

بطور کلی برای نمایش تعداد محدودی دهنه (تا ۱۰۰ عدد)، تکنولوژیهای دیود ساع نور، کریستال مایع، تخلیه گازی، و فلئوئوسان مناسب هستند. اکثر این نمایشگرها بصورت واحدهای یکپارچه کامل (شامل مدارهای فاصل ۱۴ و محرکهای لازم) ساخته می شوند و علائم را با استفاده از ماتریسهای نقطه ای ایجاد می نمایند. ساخت نمایشگرهای ساده "خودکفا" اخیراً با بکارگیری ریزپردازنده ها و مدارهای مجتمع با تراکم زیاد و خیلی زیاد، ممکن و مقرون به صرفه گشته است.

و اما در مورد قیمتها: معمولاً بهای عناصر مختلف الکترونیکی کاملاً وابسته به مناسب بودن و بازده عمل آنها می باشد. در عرصه الکترونیک نوری نیز چنین است. در حالی که یک نوع معمولی ال سی دی قرمز رنگ در اندازه استاندارد ۱۲/۵ میلیمتر مثلاً به قیمت ۲۰ پنس در انگلستان فروخته می شود، انواع مشابه و پیچیده تر آن تا ۱/۵ پوند قیمت دارند. بهای نمایشگرهای گازی - پلاسمایی ۲ تا ۵ پوند، دیود ساع نور ۱ تا ۳ پوند، و انواع کریستال مایع حدود یک پوند به ازای هر رقم یا علامت است.

هرگاه تعداد دهنه ها از ۱۰۰ تجاوز کند، استفاده از لامپهای با اشعه کاتدی مقرون به صرفه است، بطوری که مثلاً یک نمونه ۳۲x۱۶ دهنه ای حدود ۲۵۰ پوند هزینه برمی دارد. هرگاه تعداد دهنه ها بیشتر باشد، لامپهای با اشعه کاتدی باز هم با صرفه تر می شوند. ولی شکنندگی و حجیم بودن این لامپها نقاط ضعفی هستند که کاربرد آنها را در بسیاری از موارد محدود می سازند. در چنین شرایطی، شاید بتوان صفحات پلاسمایی با جریان متناوب یا جریان مستقیم را جانشین آینده آنها دانست.

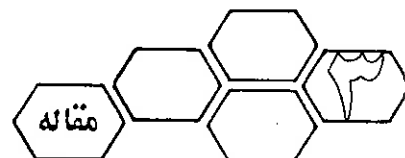
مجتمع ویژه، دستورات خارج شده از مدارهای منطقی را پس از تقویت کافی به ایمن لامپها رسانید.

همانطور که قبلاً ذکر گردید، دیودهای ساع نور، یا ال سی دی، امروزه متداول ترین تکنولوژی ساخت نمایشگرهای هفت قطعه ای را تشکیل می دهند. البته خواندن ارقام روی اینگونه نمایشگرها در مقابل نور مستقیم خورشید تقریباً غیرممکن است، هرچند که با روشهای نسبتاً پُر هزینه ای این نقص قابل رفع است.

از دیودهای ساع نور که بگذریم، می رسم به نمایشگرهای کریستال مایع، یا ال سی دی، که آنها نیز بتدریج جای خود را در بازار باز کرده اند. کم مصرف بودن اینگونه نمایشگرها اهمیت آنها را در ساخت وسایلی چون ساعت های مچی افزایش داده چون در این قبیل موارد مآخذ - لکه عمر باتری حائز اهمیت زیادی است. یک نقطه ضعف نمایشگرهای ال سی دی حساسیت آنها در مقابل دمای زیاد و رطوبت است. حداکثر درجه حرارت مجاز برای کار این نمایشگرها نوعاً ۵۰ درجه سانتیگراد است، ولی شرکت آمریکایی جنرال الکتریک انواعی از ال سی دی را ارائه داده که دامنه کارشان از ۱۵- تا ۸۵+ درجه سانتیگراد است و نیز می توانند تا حدود ۴۰ روز تحت دمای ۵۰ درجه سانتیگراد و رطوبت ۹۵ درصد مقاومت نمایند. دو مزیت عمده نمایشگرهای کریستال مایع ارزان آنها و خوانا بودن حتی در نور مستقیم خورشید است. کم بودن ولتاژ کار آنها این امکان را فراهم می سازد تا بتوان مستقیماً به مدارهای الکترونیکی از نوع "سی اس" و ولتان کرد.

اینک می رسم به تکنولوژی فلئوئوسان که اساس کار آن مشابه لامپهای ترمیونیک^{۱۳} می باشد. در لامپ فلئوئوسان با مهتابی، یک کاتد گرم شده، الکترونها را تولید می نماید که از میان یک میدان الکتریکی تا ۸۰ ولت، بسوی آند با پوشش فسفر جذب می شوند. یک توری یا شبکه مابین آند و کاتد قرار می گیرد که با تغییر ولتاژ بمیزان ۵ ولت، مشخصات و شدت تابش الکترونها را بر آند تحت کنترل قرار می دهد. این نمایشگرها با مصرفی کمتر از نصف مصرف دیودهای ساع نور، حروف و علائم را برنگ آبی مایل به سبز بسیار درخشان تولید می کنند و در واحدهای تا ۶۰ علامتی (با علائمی به ارتفاع ۵ تا ۲۰ میلیمتر) ساخته می شوند.

روش دیگری که به مرور مورد توجه بیشتری قرار خواهد گرفت استفاده از پدیده ای موسوم به الکترو لومینسانس است. در این روش، ماده بکار رفته تحت یک میدان الکتریکی متناوب و قوی، نورانی می شود.



کامپیوترهای کوچک با روز سری ال

پادداشت سردیس: کامپیوترهای کوچک و بزرگ بسیاری در گوشه و کنار کشور با حداقل استفاده مشغول کار هستند و با آنکه بکلی بدست فراموشی سپرده شده اند، بنا بر اظهار نویسنده مقاله: "تنها از سری ال کامپیوترهای کوچک با روز ۱۳۲ واحد توسط شرکتها و بانکها خریداری شده است، بسیاری از متدیان این دستگاهها، پس از سالها کار با آنها، هنوز با مفهوم برنامه و نحوه اجرای آن بیگانه اند و تنها رابط آنها با ماشین دیکمه‌هایی است که هر یک عمل بخصوصی را انجام می‌دهد. طبیعی است که این افراد در صورت بروز کوچکترین اشکال در سیستم و یا نیاز به اعمال جزئی ترین تغییرات در برنامه، با اشکالات فراوانی روبرو می‌گردند." این مقاله با هدف آشنا ساختن خوانندگان با کامپیوترهای کوچک با روز سری ال و فراهم آوردن زمینه‌ای برای فعال ساختن این ماشینها در سطح معقول بعنوان گامی در راه نیل به خودکفائی نسبی از نظر خدمات و فنون کامپیوتری نوشته شده است. امیدواریم خوانندگان آشنا با سایر ماشینهای موجود در ایران مقالات مشابهی را برای مسکن ارسال دارند.

۱ - مقدمه

کامپیوترهای کوچک با روز سری ال از سال ۱۹۶۸ به بازار عرضه گردیدند. مدلهای اولیه شامل سریهای ال ۲۰۰۰، ال ۳۰۰۰، ال ۴۰۰۰ و ال ۵۰۰۰ بودند. در حال حاضر این ماشینها از خط تولید خارج شده‌اند ولی تعداد زیادی از آنها به فعالیت خود ادامه می‌دهند. کامپیوتر کوچک ال ۷۰۰۰ با آنکه دارای امکانات بیشتری نسبت به مدلهای قدیمیتر بود، با استقبال چندانی مواجه نگردید و خیلی زود از خط تولید خارج شد.

در سال ۱۹۷۴، شرکت با روز تولید مدلهای جدید را با عرضه سیستمهای ال ۶۰۰۰ و ال ۸۰۰۰ آغاز نمود. مدل ال ۶۰۰۰ با اختلاف اندکی مشابه همان سیستمهای اولیه سری ال می‌باشد. کامپیوترهای سری ال ۸۰۰۰، که از حافظه "ماس" استفاده می‌کنند، علاوه بر دستیابی به سرعت زیاد در اجرای برنامه‌ها (۳۰ برابر سرعت سریهای قدیمی)، از امکانات فراوان دیگری منجمله دستگاههای جنبی متنوع برخوردارند. بالاترین مدل سیستمهای سری ال کامپیوترهای کوچک سری ال ۹۰۰۰ هستند که علاوه برداشتن کلیه امکانات سریهای قبیل، از تسهیلات دیگری مانند چاپ دوطرفه و چاپ ماتریسی سریع (۹۰ تا ۱۲۰ علامت در ثانیه) برخوردار می‌باشند.

زبان مورد استفاده سیستمهای ال ۲۰۰۰

تا ال ۶۰۰۰ زبان ماشین سری ال ۱۳ و زیرمجموعه‌ای از زبان تجاری کوپول می‌باشد. سریهای ال ۸۰۰۰ و ال ۹۰۰۰ علاوه بر زبان سری ال ۱۳، از زبان سری ال ۵ که دارای امکانات بکارگیری دستگاههای جنبی بیشتری می‌باشد، نیز استفاده می‌کنند.

هر یک از مدلهای سری ال ۸۰۰۰ و ال ۹۰۰۰ دارای ۸ تا ۱۶ هزار بایت حافظه برای ریزبرنامه‌ها ۲ و ۴ تا ۴۸ هزار بایت (بصورت پیمانه‌های ۳ دوازده هزار بایتی) برای برنامه‌های استفاده‌کننده است. این ماشینها را می‌توان با افزودن میانجیهای ارتباطی به پایانه کامپیوتر تبدیل نمود. داده‌ها بصورت همگام یا غیرهمگام با سرعت ۷۵ تا ۹۶۰۰ بیت در ثانیه ارسال می‌شوند. سرعت شباهت سریهای ال ۲۰۰۰ تا ال ۶۰۰۰، در دنباله مقاله تنها به ذکر مشخصات فنی کامپیوترهای سری ال ۶۰۰۰، ال ۸۰۰۰ و ال ۹۰۰۰ اکتفا می‌کنیم.

۲ - مشخصات فنی مشترک

کلمه‌ها دارای ۶۴ بیت درازا هستند. هر کلمه حافظه می‌تواند حاوی یک عدد دهدهی صحیح ۱۵ رتبی (با ضافه علامت)، ۸ علامت الفبا عددی ۴، و یا دستورالعمل باشد. ظرفیت حافظه با واحد بایت (هشت بیت) بیان می‌شود و هر هشت بایت تشکیل یک کلمه را می‌دهند.

هر عملوند از هشت علامت الفبا عددی با ۱۵ رقم صحیح با ضافه یک علامت تشکیل می‌شود. در سریهای ال هیچگونه تسهیلاتی برای محاسبات با ممیز شناور وجود ندارد. دستورالعملهای زبان سری ال ۳ دارای طولی برابر ۴ رقم و دستورالعملهای زبان سری ال ۵ دارای طولی برابر ۶ رقم هستند. اطلاعات داخلی در سیستمهای سری ال با کد آسکی نمایش داده می‌شوند.

از نقطه نظر فیزیکی، کامپیوترهای سری ال می‌توانند در محدوده وسیعی از درجه حرارت و رطوبت محیط کار کنند. این ماشینها بسیار مقاوم بوده و معمولاً نیاز به تهویه یا کف کاذب ندارند.

۳ - کامپیوترهای سری ال ۶۰۰۰

حافظه اصلی در سری ال ۶۰۰۰ بصورت یک گرده مغناطیسی سرامیک با نوکهای ثابت می‌باشد. زمان متوسط دستیابی برابر ۵ هزارم ثانیه است. ظرفیت حافظه حداکثر ۲۱۸۰ کلمه ۶۴ بیتی است که در ۵ بلوک ۲۵۶ کلمه‌ای سازمان داده شده‌اند. هر بلوک دارای ۸ لیه است و بر روی هر لیه یک نوک خواننده و نویسنده تعبیه شده است.

حافظه به سه ناحیه تقسیم می‌شود. ناحیه معمولی، ناحیه کنترل، و ناحیه

متغیر. ناحیه معمولی مختص برنامه‌های استفاده‌کننده است. در ناحیه کنترل تنها ریزدستورهای سیستم نگهداری می‌شوند. ناحیه متغیر در مواقعی که هر یک از ناحیه‌های معمولی یا کنترل نیاز به ظرفیت بیشتر داشته باشد مورد استفاده قرار می‌گیرد. یک انباشتگر ۶۴ بیتی (۱۵ رقم با ضافه

علامت) برای نگهداری موقت داده‌های عددی بکار برده می‌شود. این انباشتگر برای انتقال داده‌ها به محلهای مختلف حافظه و نگهداری موقت داده‌ها برای چاپ مورد استفاده قرار می‌گیرد.

چهار ثبات شاخه وجود دارند و شاخص دستورالعملها توسط یک دستورالعمل ویژه، که بلافاصله قبل از دستورالعملی که نشانی آن باید اصلاح گردد، قرار می‌گیرد، تعیین می‌شود. در سیستمهای سری ال ۶۰۰۰ نشانی دهی غیرمستقیم وجود ندارد.

این ماشینها دارای ۱۶۷ فرمان بسیار دستورالعمل هستند که هر یک ریزبرنامه مربوط به خود را فعال می‌کند. از این تعداد، ۶۷ دستورالعمل بصورت متعارف و ۹۱ دستورالعمل برای ارتباط داده‌ها و دستگاههای اختیاری دیگر مورد استفاده قرار می‌گیرند. زمان اجرای انواع دستورالعملها برای هر عملوند یک کلمه‌ای بصورت زیر است:

نقل مکان ۳۰ هزارم ثانیه

جمع و تفریق ۵۰ "

مقایسه و شرط ۶۰ "

در سیستمهای ال ۶۰۰۰ مکان نیز وقفه ۱۰ وجود ندارد. معیذا اشتباهات مربوط به صفحه‌کلید ماشین را بحالت توقف می‌برند. بروز اشتباه توسط یمدا در آمدن رنگ یا روشن شدن یک چراغ بروی صفحه‌کلید مشخص می‌گردد.

۴ - کامپیوترهای سری ال ۸۰۰۰ و ال ۹۰۰۰

حافظه اصلی این ماشینها از نوع "ماس" پویا ۱۱ با زمان دستیابی ۱/۵ میلی‌ثانیه است. ظرفیت حافظه استفاده‌کننده بین ۴ تا ۴۸ هزار بایت بصورت پیمانه‌های دوهزار بایتی و ظرفیت حافظه ریزبرنامه بین ۸ تا ۱۶ هزار بایت می‌باشد.

پردازنده اصلی و ممیز فرمان نیاز به هشت هزار بایت حافظه کنترل دارند. برای هر دستگاه جنبی دیگری که مورد استفاده سیستم قرار گیرد، مقداری حافظه کنترل بطور خودکار تخصیص داده می‌شود. دسترسی به ناحیه کنترل حافظه برای هیچ برنامه‌ای میسر نیست.

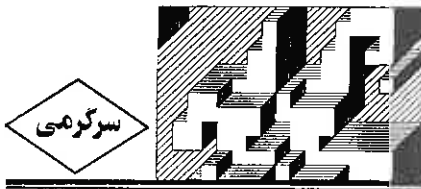
یک ویژگی جالب سیستمهای سری ال ۸۰۰۰ و ال ۹۰۰۰ مکان نیز خودکاری است که در موقع قطع برق یک دستگاه ضبط کارتریج را که با باتری کار می‌کند فعال ساخته و ورق بزنند

امکانات آنها را گسترش داده و برکارایی این سیستمها می افزایند. ●

ترجمه و گردآوری محمدرضا حسینیان
از "گزارشهای دیتا پرودریاس" ۱۹۸۰
کامپیوترهای کوچک

یادداشتها

- ۱- SL3 = System Language 3 ..
- ۲- ریزبرنامه Microprogram
- ۳- پیمانه Module
- ۴- الفبا عددی Alphanumeric
- ۵- کد آسکی ASCII
- ۶- لبه Track
- ۷- ریزدستور Micro-instruction
- ۸- انباشتگر Accumulator
- ۹- ثبات شاخص Index register
- ۱۰- وقفه Interrupt
- ۱۱- "ماس" پویا Dynamic MOS
- ۱۲- میانگیر Buffer
- ۱۳- کارت مدار Circuit card
- ۱۴- چرخاننده نوار کاسیت Cassette drive
- ۱۵- همگردانی Compilation
- ۱۶- برنامه مقصود Object program



پاسخ سرگرمی شماره قبل

مساءله این بود که اگر در عمل ضرب زیر هریک از ارقام ۱ تا ۹ یک بار و تنها یک بار بکار رفته باشد، ارزش عددی حروف A تا H را تعیین کنید.

$$3A \times BCD = EFGH$$

پاسخ مسأله بقرار زیر است:

$$۷۲۵۴ = ۱۸۶ \times ۳۹$$



انجمن انفورماتیک ایران
وماهنامه گزارش کامپیوتر
را به دوستان و همکاران
خود معرفی کنید.

میز فرمان در کلیه سیستمهای سری ال شامل صفحه کلید، حافظه، پردازنده، سیستم، دستگاه چاپ، کلیدهای کنترل، لامپهای نشانگر و یک خواننده کوچک نوار کاغذی (تنها برای بارکردن برنامهها) است. در بعضی مدلها، میز فرمان شامل چند چرخاننده نوار کاسیت ۱۴ نیز می باشد. صفحه کلید شامل مجموعه ای از کلیدهای انتخاب برنامه، چراغهای ارتباط با متمدی، کلیدهای مجزا برای وارد نمودن اعداد، و هشت کلید کنترل می باشد.

سیستمهای سری ال ۶۰۰۰ از مکانیزم گوی ۶۴ حرفی با سرعتی معادل ۲۰ حرف در ثانیه برای چاپ استفاده می کنند. مدلهای ال ۸۷۰۰ و ال ۸۸۰۰ نیز از مکانیزم گوی برای چاپ استفاده می کنند ولی سرعت چاپ در آنها ۳۰ حرف در ثانیه است. این دستگاهها می توانند از ۱۵۰ تا ۲۵۵ حرف را بر روی هر سطر چاپ کنند. تعداد نسخه های قابل تهیه به ۱۰ می رسد و شدت ضربه های گوی برای چاپ چند نسخه ای بصورت خودکار قابل تنظیم است.

سیستمهای سری ال ۹۰۰۰ دارای دستگاه چاپ ماتریسی ۹۴ حرفی (شامل حروف بزرگ و کوچک) هستند که سرعت چاپ آنها از ۶۰ حرف در ثانیه تا ۱۲۰ حرف در ثانیه است. تعداد نسخه ها در چاپ کننده های ماتریسی حداکثر مساوی ۶ است. یک ویژگی جالب سیستمهای سری ال ۶۰۰۰ به بالا امکان استفاده از کارتهای مغناطیسی است که داده ها را بر روی لبه خود ضبط می کنند. ظرفیت کارتهای مغناطیسی یک لبه ای معادل ۲۵۲ رقم و ظرفیت کارتهای دولبه ای معادل ۷۰۴ رقم است.

۶- تسهیلات نرم افزاری

کامپیوترهای کوچک سری ال دارای سیستم عامل نیستند. زبان برنامه نویسی اصلی برای سیستمهای سری ال ۸۰۰۰ و ال ۹۰۰۰ زیرمجموعه خاصی از زبان کوپول است. برنامه هایی که به این زبان نوشته شده باشند باید ابتدا بر روی یکی از ماشینهای باروز بی ۳۵۰۰، بی ۳۷۰۰ یا بی ۴۷۰۰ همگردانی شده و برنامه مقصود ۱۶ حاصل بر روی ماشینهای ال ۸۰۰۰ یا ال ۹۰۰۰ اجرا گردد.

استفاده کنندگانی که امکان همگردانی برنامه های خود را بر روی ماشینهای فوق الذکر نداشته باشند می توانند از زبان ماشین ال ۵، که همگردانی آن توسط خود کامپیوترهای ال ۸۰۰۰ یا ال ۹۰۰۰ انجام می گیرد، استفاده کنند. کامپیوترهای سری ال ۶۰۰۰ نیز می توانند بشرح فوق از زبان کوپول استفاده نمایند. برنامه های پیش نوشته شده بسیاری برای کامپیوترهای سری ال وجود دارند که

کلیه محتویات حافظه و وضعیت ثباتها را بر روی آن ضبط می کند. هنگامی که جریان برق مجدداً برقرار شود، محتویات نسوار کارتریج بطور خودکار وارد حافظه و ثباتها می گردد.

در صورت استفاده از زبان ال ۵، کامپیوترهای سری ال ۸۰۰۰ و ال ۹۰۰۰ دارای قابلیت نشانی دهی غیرمستقیم هستند. نحوه استفاده از ثباتهای شاخص در این ماشینها مانند ماشینهای سری ال ۶۰۰۰ است. در این ماشینها، تقریباً ۶۵۰ دستور العمل در سطح سخت افزاری و ریزبرنامه های برای قریب به ۷۰ نوع عمل مورد استفاده قرار می گیرند. دستورات عملهای سخت افزاری با ۱۹۰ فرمان ال ۳ یا ۲۵۷ فرمان ال ۵ در سطح استفاده کننده فعال می شوند.

زمان اجرای انواع دستورات عملها برای هر عملکرد یک کلمه ای بصورت زیر است:

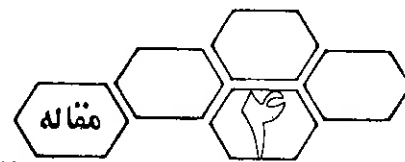
نقل مکان	۵/۷۵ هزارم ثانیه
جمع	۱/۵۵ "
تفریق	۱/۶۳ "
ضرب (۱۶ در یک رقم)	۱۸/۲۵ "
تقسیم (۱۶ بر ۵ رقم)	۲/۷۰ "
مقایسه	۱/۷۷ "

سیستم وقفه ای که در کامپیوترهای سری ال ۸۰۰۰ و ال ۹۰۰۰ بکار برده می شود از چهار سطح تشکیل شده است. سه نوع وقفه توسط دستگاههای جنبی تولید شده و منجر به قطع برنامه یا مشخص شدن اشکال توسط لامپ یا زنگ می گردد. وقفه نوع دیگر اجازه استفهام در مورد وضع دستگاهها را به پردازنده می دهد.

۵- دستگاهها و کنترل ورودی و خروجی

سیستمهای سری ال دارای یک میانگیر ۱۲ ۳۲ حرفی برای ورودیهای صفحه کلید و یک میانگیر ۳۲ حرفی برای خروجیها هستند که انجام عملیات همزمان ورودی و خروجی را ممکن می سازند. برای ثبت اطلاعات علاوه بر صفحه کلید می توان از پایانه های محلی که دارای یک صفحه نمایش ۲۵۶ تا ۴۸۰ حرفی و یک صفحه کلید می باشند استفاده نمود.

تعداد دستگاههای جنبی و یا پیمانه های حافظه که می توانند بر روی سیستمهای سری ال ۸۰۰۰ و ال ۹۰۰۰ نصب گردند بستگی کامل به شمارهای موجود روی محفظه پردازنده دارد. محفظه پردازنده می تواند حداکثر ۱۰۰ کارت مدار ۱۲ را در خود جای دهد. از این تعداد ۵۰ محل برای خود پردازنده و ۱۶ هزار بایت حافظه استفاده کننده تخصیص یافته و ۵۰ محل باقیمانده را می توان برای حافظه های اضافی و یا کنترل کننده های ورودی یا خروجی مورد استفاده قرار داد.



مسائلی در رابطه با ساخت فیزیکی

مدارهای الکترونیک رقمی

دکتر فرهاد صاحبان
استادیار دانشگاه ملی ایران

با داشتن سردبیر: برخلاف آنچه که ممکن است در مرحله اول بنظر برسد، طراحی مدارهای منطقی تنها بخش کوچکی از فرآیند طراحی و ساخت یک محصول یا سیستم جدید الکترونیکی را تشکیل می دهد. در این مقاله سعی شده است که خواننده با مرألی که بین طراحی مدارهای منطقی و تولید مدارهای الکترونیکی مربوطه در کارخانه می آیند آشنا شود. مقاله دیگری با عنوان "مقدمه ای بر روشهای مکان گزینی و مسیریابی در طراحی مدارهای چاپی یکمک کامپیوتر" که برای یکی از شماره های آینده ما هنامه در دست ترجمه است، به بررسی دقیقتر مسائل سوم و چهارم از مسائل شش گانه، تشریح شده در این مقاله می پردازد.

پس از تکمیل طراحی منطقی یک مدار الکترونیک رقمی، باید طرح فیزیکی آن را مشخص نمود. جزئیات نحوه عمل در مشخص ساختن این طرح برای تکنولوژیهای مختلف متفاوت است. در این مقاله، فقط مسائل طرح فیزیکی را در رابطه با مدارهای مجتمع الکترونیکی و کارتهای مدار چاپی در نظر خواهیم گرفت. در این تکنولوژی، قسمتها با عناصر اصلی عبارتند از تراشه های مدار مجتمع (یا بطور ساده، مدارهای مجتمع).

قسمتهای مستقل یک مدار مجتمع را اجزای آن می نامند. هر یک از اجزای اصلی یک سیستم رقمی (عناصر منطقی، فلیپ فلاپها، و غیره) باید بوسیله یک مدار مجتمع یا جزئی از آن ساخته شود. پس اولین قسمت مسأله عبارت از انتخاب مجموعه ای از مدارهای مجتمع است که برای ساختن همه اجزای سیستم کافی باشد. بعنوان مثال، توجه کنید که یک عنصر منطقی NAND با سه ورودی می تواند بجای یک عنصر منطقی NAND با دو ورودی بکار رود (در صورتی که به ورودی اضافی آن مقدار ثابت منطقی ۱ را متصل کنیم). بنابراین هیچ لزومی ندارد که عناصر مدار اصلی و اجزای مدارهای مجتمع بکار رفته کاملاً همان باشند.

پس مدارهای مجتمع را بر روی کارتهای مدار چاپی قرار می دهند. هرگاه یک کارت گنجایش تمام مدارهای مجتمع را نداشته باشد، باید از چندین کارت استفاده شود. این کارتها بر روی یک صفحه بنام "مفحه پشتیبان" قرار می گیرند.

در مرحله بعدی باید اتصالیهای الکتریکی بین قسمتهای مختلف مدارهای

مجتمع برقرار شود. این اتصالات بر سه نوعند:

الف - اتصال بین دو جزء در داخل یک مدار مجتمع.

ب - اتصال بین مدارهای مجتمع بر روی یک کارت.

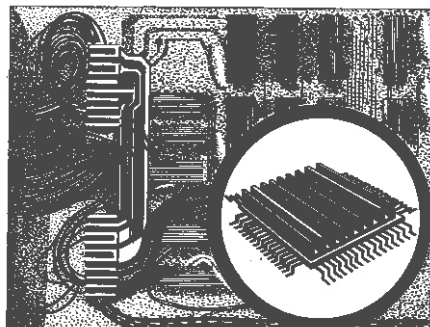
پ - اتصال بین مدارهای مجتمع در کارتهای مختلف.

اکنون رشته مسائلی را که در طرح فیزیکی مدارهای الکترونیک رقمی با استفاده از مدارهای مجتمع مطرحند با جزئیات بیشتری بررسی می کنیم.

۱ - انتخاب و تخصیص مدارهای مجتمع: برای ساختن مدار مورد نظر، مجموعه ای از مدارهای مجتمع را از بین مدارهای در دسترس انتخاب می کنیم. این انتخاب عموماً با هدف بدست آوردن یک مجموعه کم هزینه یا مجموعه ای با کمترین تعداد مدارهای مجتمع صورت می گیرد. روش انتخاب در این مورد نظیر مسأله "پوشش" در تعیین ساده ترین شکل یک عبارت یا مدار منطقی است.

۲ - تخصیص کارتها: عموماً "مجموعه تمام مدارهای مجتمع لازم برای ساختن یک سیستم را نمی توان بر روی یک کارت قرار داد. بنابراین، این مجموعه به زیرمجموعه های مجزا تفکیک شده و به هر زیرمجموعه یک کارت اختصاص می یابد. مسیهای الکتریکی بین کارتها از طریق اتصالات لبه ای برقرار می شوند. تخصیص کارتها با هدف حداقل کردن تعداد کارتها، بطریقی که تعداد اتصالیهای لبه ای هر کارت از عدد ثابت P و تعداد مدارهای مجتمع بر روی یک کارت از عدد ثابت M تجاوز نکند، صورت می گیرد. بطور کلی باید مجموعه مدارهای مجتمع را بطریقی تقسیم کرد که تعداد اتصالیهای الکتریکی بین کارتها به حداقل برسد.

۳ - قراردادن مدارهای مجتمع بر روی کارت (مکان گزینی): مدارهای مجتمعی که به یک کارت تخصیص یافته اند باید بصورتی در مکان مشخص بر روی آن کارت قرار گیرند که مسیهای اتصال الکتریکی بین زوجهای از این مدارها از یک طرف و بین این مدارها و اتصالات لبه ای کارت از طرف دیگر



حتی الامکان ساده شوند. محدودیتهای بسیاری باید در تعیین مسیر این اتصالیها در نظر گرفته شوند. این محدودیتهای شامل حداکثر طول هر اتصال، حداقل فاصله ای که دو مسیر را از هم جدا می کند، و عبور تقاطع اتصالیها هستند. ممکن است برای رعایت شرط آخر، یک کارت نیاز به لایه های مختلفی داشته باشد که مسیها، با استفاده از اتصالیهای بین لایه ای، بر روی آنها قرار گیرند. تعریف یک تابع هدف ساده که دقیقاً تمام شرایط ذکر شده را در بر داشته باشد غیرممکن است. تابع هدفی که اکثراً برای حل مسأله قرار دادن مدارهای مجتمع بر روی کارت مورد استفاده قرار می گیرد و نتایج خوبی نیز می دهد، مستلزم حداقل ساختن طول کلی اتصالیهای موجود بر روی یک کارت است.

۴ - تعیین اتصالیهای داخلی کارت (مسیریابی): مسیر هندسی تمام اتصالیهای الکتریکی روی کارت بر حسب شرایط فوق الذکر تعیین می شود. تعیین مسیر بصورت گام به گام (با تعیین مسیر برای یک اتصال در هر مرحله) صورت می گیرد. بنابراین، حل مسأله تعیین اتصالیهای داخلی کارت مستلزم حل مسائل فرعی زیر است: (الف) تعیین ترتیب - تعیین مسیر اتصالیها به چه ترتیبی باید صورت گیرد؟ یک قاعده متداول این است که در هر مرحله کوتاهترین اتصال در نظر گرفته شود، ولی مؤثر بودن این قاعده قابل سوال است. (ب) لایه گذاری - چند لایه بکار می رود و بر روی هر لایه چه اتصالیهای قرار می گیرند؟ (پ) تعیین مسیر - مسیر هندسی هر اتصال بر روی کارت باید مشخص شود. متداولترین روش در این زمینه تعیین کوتاهترین مسیر برای هر اتصال یکمک الگوریتم موسوم به "الگوریتم سرگردان" است.

۵ - قراردادن کارتها بر روی صفحه پشتیبان: برای مدارهایی که به بیش از یک کارت نیاز دارند، کارتها باید با هدف ساده کردن مسیر اتصالیهای الکتریکی بین آنها بر روی صفحه پشتیبان قرار گیرند. این مسأله مشابه مسأله قرار دادن مدارهای مجتمع بر روی کارت است و برای حل آن می توان از همان روشها و الگوریتمها استفاده نمود.

۶ - اتصالیهای داخلی صفحه پشتیبان: مسیهای هندسی برای اتصالیهای بین تمام کارتها باید تعیین شوند. این مسأله مشابه مسأله اتصالیهای داخلی بر روی یک کارت است و برای حل آن می توان از همان روشها و الگوریتمها استفاده نمود.

مسائلی که بر شمردیم دارای این خاصیت مشترک هستند که در حالت کلی، در نظر گرفتن تمام راه حل های ممکن آنها غیر عملی

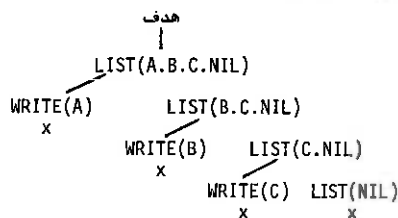
زیر جهت تشخیص این حالت بکار می روند:
 $SUBLIST(NIL, *X).$
 $SUBLIST(*U, *X, *U, *Y) + SUBLIST(*X, *Y).$
 $SUBLIST(*X, *U, *Y) + SUBLIST(*X, *Y).$

یادداشتها

- ۱ - پرولوگ $PROLOG=PROgramming\ in\ LOGic$
- ۲ - نحو Syntax
- ۳ - منطق تفکیک پذیری Resolution logic
- ۴ - پردازش اشبات .. Proof procedure
- ۵ - بترتیب ژرفا Depth-first
- ۶ - بترتیب پهنا Breadth-first
- ۷ - عبارت Term
- ۸ - شالوده Skeleton
- ۹ - لفظ Literal
- ۱۰ - محمول Predicate
- ۱۱ - سوراخ فراگیر Universally quantified
- ۱۲ - درخت ایجاب Implication tree
- ۱۳ - ردیابی Backtracking
- ۱۴ - جانشینی Substitution
- ۱۵ - مورد Instance
- ۱۶ - یکسان سازی Unification
- ۱۷ - نشان گذاری میانوندی Infix notation
- ۱۸ - محمول توکار .. Built-in predicate
- ۱۹ - زیرلیست Sublist

(تاریخ دریافت: ۱ اردیبهشت ۱۳۶۱)

دنیاله مقاله ۷: پرولوگ ...
 محمول توکار ۱۸ WRITE در زبان پرولوگ
 برای نوشتن بکار می رود.
 جمله هدف $LIST(A.B.C.NIL) +$ بصورت
 زیر عمل می نماید:



خروجی حاصل بر روی پایانه عبارت
 خواهد بود از:

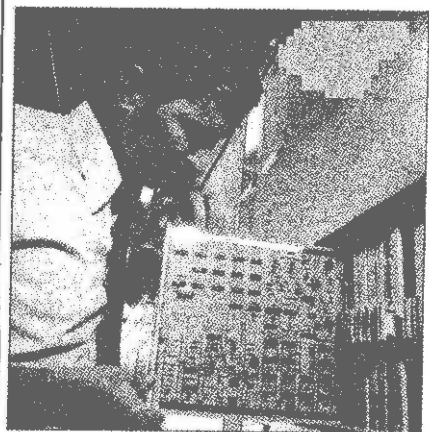
A.
B.
C.

حال به مثال دیگری می پردازیم. اصل
 موضوعه INTERSECT را، برای تشخیص
 اینکه دو مجموعه دارای عنصر مشترکی
 هستند یا نه، در نظر بگیرید:
 $INTERSECT(*A, *B) + ELEMENT(*X, *A)$
 $\& ELEMENT(*X, *B).$

بعنوان آخرین مثال، مسأله زیر را
 در نظر بگیرید: "لیست A زیرلیستی ۱۹ از
 لیست B است اگر تمام عناصر A بهمان
 ترتیب، ولی نه لزوماً بطور همجوار، در
 B وجود داشته باشند." طبق این تعریف،
 ab زیرلیستی از acb است. اصول موضوعه

است. برای مثال، اگر بخواهیم M مدار
 مجتمع را بر روی یک کارت که دارای N محل
 است قرار دهیم ($N \geq M$)، تعداد $M!(N!)$ راه
 ممکن وجود دارد و بنابراین بررسی تمام
 جوابها فقط برای مقادیر کوچک N و M عملی
 است.

هدف بیشتر اینگونه مسائل رانمی توان
 دقیقاً از نظر ریاضی تعیین کرد و تابع
 هدفی که معمولاً بکار می رود فقط تخمینی
 از هدف اصلی است. مثلاً هدف صحیح در
 مرحله مکان گزینی بر روی یک کارت، ساده
 کردن مسیرهای اتصال بر روی آن است، در
 صورتی که عموماً حداقل کردن طول کلی
 اتصالات بعنوان تابع تقریبی هدف در نظر
 گرفته می شود.



معمولاً بنا بر دلایل فوق الذکر، تلاشی
 برای یافتن بهترین راه حل این مسائل صورت
 نمی گیرد و در عوض روشهایی بوجود آمده اند
 که هدف آنها پیدا کردن جوابهای نسبتاً
 خوب از طریق انجام محاسبات برای تابع
 هدف تقریبی است. اینگونه روشها را
 روشهای "آزمایش و تصحیح" می نامند.
 با کاربرد روشهای مختلف "آزمایش و
 تصحیح" و با بکار گیری یک روش بخصوص برای
 چندین بار، می توان راهحلهای مختلفی
 نسبتاً خوبی برای یک تابع هدف تقریبی
 بدست آورد و با ارزیابی دقیق آنها،
 بهترین راه حل را با توجه به هدف واقعی
 انتخاب نمود.

مراجع و یادداشتها

1. Breuer, M.A., Design Automation of Digital Systems, -- Vol I: Theory and Techniques, Prentice-Hall, 1972.
2. Friedman, A.D., Logical Design of Digital Systems, Computer Science Press, 1975.

۲ - پوش Covering

(تاریخ دریافت: ۱ اردیبهشت ۱۳۶۱)



کمیته انتشارات

انجمن انفورماتیک ایران

تقدیم می کند

واژه نامه مقدماتی کامپیوتر
 و انفورماتیک (انگلیسی به فارسی)

بها ۲۰۰ ریال

حای ۶۰۰۰ واژه کامپیوتری

ماهنامه گزارش کامپیوتر

بها به ریال
 دوره های جلد شده اعفاء * دیگران
 سال اول (۱۲۰ صفحه) ۲۰۰ ۴۰۰
 سال دوم (۱۷۶ صفحه) ۳۰۰ ۶۰۰
 سال سوم (۱۸۴ صفحه) ۴۰۰ ۷۰۰

تک شماره برای هر ماه ۵۰ ۵۰

لطفاً "نشریات فوق را با ارسال
 رسید پرداخت وجه به حساب جاری
 شماره ۱۱۱۷ انجمن نزد بانک
 صادرات شعبه ۱۶۵۲ (دانشگاه
 صنعتی شریف)، تهران، و درج نشانی
 کامل پستی خود سفارش دهید.

* تذکر: تخفیف مخصوص اعفاء فقط
 در نسخه اول برای استفاده
 شخصی منظور می شود.

برای تماس

برای تماس با انجمن از نشانی
 یا تلفنهای زیر استفاده کنید:
 نشانی: انجمن انفورماتیک ایران،
 صندوق پستی ۲۵/۱۲۶، آریا شهر، تهران ۱۴، ایران.

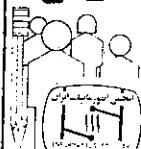
شماره تلفن مسئولان انجمن:

رئیس انجمن (دکتر بهروز پرهامی) ۹۱۸۲۶۹
 نائب رئیس (دکتر خلیل فضل الهی) ۶۲۱۷۸۱-۲
 دبیر انجمن (آقای ابراهیم مشایخ) ۲۴۹۰۱۸
 خزانه دار (آقای علی اکبر نهودی) ۶۲۸۴۱۳

شماره تلفن سرپرستان کمیته ها:

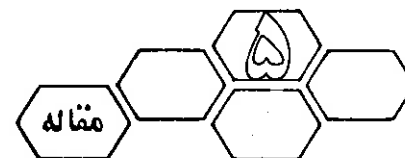
انتشارات (دکتر بهروز پرهامی) ۹۱۸۲۶۹
 فعالیتهای علمی و فنی (دکتر فرهاد ماحیان) ۲۱۴۱۶۱۵
 آموزش (دکتر علیرضا کاویانپور) ۹۱۸۲۵۰
 عضویت و روابط عمومی (خاتم ویدا دانی) ۹۱۸۲۶۹

گزارش کامپیوتر



ماهنامه انجمن انفورماتیک ایران
 زیر نظر کمیته انتشارات
 سرپرست کمیته انتشارات و
 سردبیر: دکتر بهروز پرهامی
 نشانی: صندوق پستی ۲۵/۱۲۶
 آریا شهر، تهران ۱۴

آخرین مهلت قبول مطالب و آگهی دومین یکشنبه
 ماه انتشار است. نقان مطالب ماهنامه
 گزارش کامپیوتر با ذکر ماه خذ آزاد است.



روشهای مختلف فراخوانی زیرروالها و نواقص هر یک

رضا مؤتمن
دانشگاه مشهد

فراخوانی یک زیرروال^۵ را، برحسب نحوه انتقال پارامترها، می توانیم بصورت های مختلفی انجام دهیم. این تفاوتها غالباً موجب بروز اختلاف در نتایج اخذ شده می گردند که ممکن است در مواردی باعث گمراهی برنامه نویس باشند. از مهمترین روشهایی که مورد استفاده همگردانه های مختلف بوده اند می توانیم فراخوانی با مقدار^۶، فراخوانی با نتیجه^۷، فراخوانی با ارجاع^۸، و بالاخره فراخوانی با نام^۹ را ذکر کنیم. هدف از تنظیم این مقاله معرفی روشهای مزبور و نیز یادآوری برخی از نواقص آنها می باشد. این نواقص را غالباً "توسط مثالها" بسیار کوتاه و برنامه هایی کاملاً ابتدائی بیان کرده ایم تا مطلب حتی الامکان خلاصه بوده و مرورش برای خواننده آسانتر باشد. این سادگی در مثالها سبب می گردد که بعضی از اشکالات مورد نظر بسیار بدیهی جلوه نمایند. ولی در برنامه های طولانی و پیچیده، تشخیص اشکالاتی از این قبیل می تواند خیلی دشوار باشد.

۱- فراخوانی با مقدار

در این نوع فراخوانی، قبل از آغاز اجرای برنامه فراخواننده شده، مقادیر پارامترهای واقعی در ناحیه پارامتر یا ناحیه داده روال فراخواننده شده رونویسی می گردد. سپس، درحین اجرای آن روال، رونوشت پارامترهای مزبور بعنوان متغیرهای موضعی روال فراخواننده شده تلقی گردیده و تغییرات بعدی آنها در متغیرهای واقعی روال فراخواننده منعکس نمی شود.

از نواقص این روش اتلاف وقت و اتلاف حافظه در ردوبدل ساختن پارامترها، بخصوص پارامترهای آرایه ای، می باشد. همچنین عدم توانائی این روش در تغییر دادن پارامترهای واقعی قابل بحث است. این عدم توانائی در برخی موارد بعنوان یک مزیت محسوب گردیده است، زیرا از تغییر ناخواسته پارامترهای واقعی جلوگیری می کند.

۲- فراخوانی با نتیجه

این روش مشابه فراخوانی با مقدار است، با این تفاوت که در خاتمه اجرای روال

فراخواننده شده، با استفاده از نشانی پارامترهای واقعی مربوطه که در هنگام دخول در اختیار روال فراخواننده شده قرار گرفته اند، مقادیر پارامترهای صوری متقابلاً در محل پارامترهای واقعی رونویسی می گردند. نقص عمده این روش نیز همانا عدم کارآئی در خصوص پارامترهای آرایه ای می باشد.

بمنظور روشنتر ساختن تفاوت میان فراخوانی با مقدار و فراخوانی با نتیجه، به دو برنامه شبوالگول زیر توجه می کنیم:

```

مثال ۱-
begin integer Y;
  procedure P(X); integer X;
    begin X:=X+1; Print(X,Y);
  endproc P;
Y:=1; P(Y); Print(Y);
end

```

در این برنامه چنانچه انتقال Y بعنوان مقدار باشد، اعداد خروجی بترتیب عبارت خواهند بود از ۲، ۱، و اگر انتقال Y بعنوان نتیجه باشد، اعداد چاپ شده در خروجی بترتیب عبارت خواهند بود از ۲، ۱ و ۲.

مثال ۲- فرض کنید که بخواهیم زیرروالی بنام SWAP برای شویف مقدار دو متغیر بنویسیم.

```

procedure SWAP(X,Y); real X,Y;
  begin T:=X; X:=Y; Y:=T;
endproc SWAP;

```

در این مورد ملاحظه می شود که فراخوانی با مقدار ناموفق است، در حالی که فراخوانی با نتیجه منظور ما را برآورده می سازد: یعنی در بازگشت از SWAP، متغیر A دارای مقدار اولیه B و متغیر B دارای مقدار اولیه A خواهد بود.

۳- فراخوانی با ارجاع

فراخوانی با ارجاع را فراخوانی توسط نشانی نیز می گویند. این نوع فراخوانی بطور کلی ساده ترین و سریعترین روش برای ردوبدل ساختن پارامترها است و بیش از همه مورد استفاده قرار می گیرد. بطور خلاصه می توان گفت که نشانی محل ذخیره شدن پارامتر واقعی تعیین شده و به روال فراخواننده شده ارسال می گردد. روال فراخواننده شده این نشانی را بعنوان پست اشاره گر بکار برده و به محل پارامتر واقعی دسترسی پیدا می کند.

هر تغییری در پارامتر صوری خود بخود در پارامتر واقعی منعکس می گردد، زیرا در واقع هر دو به یک محل در حافظه وابسته هستند. نتیجتاً، در این موارد هیچ لزومی به تهیه رونوشت از پارامترهای واقعی وجود ندارد. این ویژگی سبب شده است که فراخوانی با ارجاع از نظر کارآئی

مورد توجه قرار گیرد. در جوار ایسین ویژگی مثبت، گاه نواقصی وجود دارند که در مثالهای ۳ و ۴ به شرح دو نمونه از آنها خواهیم پرداخت.

بعنوان یک مورد ساده از این نسوع فراخوانی، مثال ۱ را دوباره در نظر می گیریم. چنانچه فراخوانی p با ارجاع باشد، اعداد ۲، ۲، و ۲ در خروجی چاپ می شوند.

مثال ۲- برنامه فراخواننده:

```

10 M = 1
  ...
20 CALL SUB(A,5)
  ...
30 M = M + 5
  ...
  SUBROUTINE SUB(X,L)
    ...
200 L = L + 3
    ...
300 RETURN

```

در این حالت چنانچه مقدار ثابت ۵ نیز مانند یک متغیر تلقی گردد، نشانی حافظه مربوط به آن به متغیر صوری L وابسته می شود. دستور ۲۰۰ در زیرروال SUB مقدار A را در حافظه مربوط به کمیت ثابت ۵ ثبت می کند.

می توان حدس زد که از این بیعد عدد ۵ در برنامه فراخواننده صورتی انحرافی خواهد داشت، زیرا ظاهراً ۵ است و باطناً ۸. برنامه نویس که از خطای مرتکب شده در زیر روال SUB آگاهی ندارد، و پیاپی هم در ارتباط با آن دریافت نکرده، انتظار دارد که مقدار M پس از اجرای دستور شماره ۳۰ برابر ۶ باشد، در حالی که واقعا ۴ چیزی دیگری است. این اشکال در زبان فرترن ۲ عملاً وجود داشته ولی در همگردانه های بعدی این زبان مرتفع شده است.

روش ساده ای که در اغلب همگردانه ها برای رفع نقیصه فوق بکار می رود استفاده از حافظه موقت برای قرار دادن مقادیر مربوط به کمیت های ثابت، و عباراتی که نمی توانند بعنوان یک متغیر محسوب شوند، می باشد. باین ترتیب که اگر پارامتر واقعی، که قرار است توسط ارجاع منتقل شود، یک ثابت یا مثلاً یک عبارت محاسباتی باشد، در لحظه ای قبل از انتقال به زیر روال مقدارش تعیین شده و در یک محل حافظه موقت قرار می گیرد. سپس نشانی این محل موقت به متغیر صوری مربوطه وابسته می شود. چنین متغیر موقتی را اصطلاحاً "یک نشانگر ساختگی" می نامند.

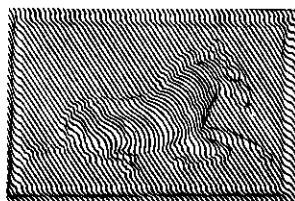
مثال ۴- فرض کنید که در برنامه فرترن زیر، که از یک زیرروال چندخولی وری بنزد

بعضی فقط یک نوع ثابت فراخوانی فرض شده است، در برخی دیگر، مانند بی ال وان یا آلگول، برنامه نویس می تواند برای انتقال هر پارامتر، یکی از طرق معین را مطابق قراردادها و قوانین مربوطه انتخاب نماید. آنچه که مهم است توجه برنامه نویس و آگاهی وی از روش بکار رفته (بسیار انتقال پارامترها) در همگردانی است که با آن سروکار دارد. ممکن است یک برنامه نویس، در عین داشتن مهارت، فقط بخاطر بی اطلاعی از نوع فراخوانی بکار رفته، درگیر خطاهایی شود که خلاصی از آنها حداقل سبب اتلاف وقت وی گردد. البته برخی از اشکالات فوق الذکر، از قبیل پایین بودن کارآیی نسبی یک روش بخصوص، مسأله یک برنامه نویس نبوده و لزومی ندارد که وی چندان نگران اینگونه مسائل باشد؛ بویژه آنکه افزایش روزافزون سرعت و گنجایش حافظه کامپیوترها بتدریج سبب کم اهمیت تر شدن چنین مسائلی می گردد. در خاتمه، بد نیست باین نکته نیز توجه داشته باشیم که گاه اسامی ذکر شده برای انواع فراخوانی در نوشته های مختلف متفاوتند. بعنوان نمونه، برای فراخوانی با ارجاع اسامی گوناگون "فراخوانی بسا نشانی"، "فراخوانی با مکان"، "فراخوانی بانام ساده"، و حتی "فراخوانی بانام" بکار رفته اند.

مراجع و یادداشتها

1. Pratt, T.W., Programming Languages: Design and Implementation, Prentice-Hall, 1975.
2. Stone, H.S. and D.P. Siewiorek, Introduction to Computer Organization and Data Structures (PDP-11 Edition), McGraw-Hill, 1975.
3. Elliott, D.G., "FORTRAN 77 and Structured Design," ACM SIGPLAN Notices, Vol. 16, No. 12, Dec. 1981.
4. Gries, D., Compiler Construction for Digital Systems, Wiley, 1977.
- ۵ - زیرروال Subroutine
- ۶ - فراخوانی با مقدار ... Call by value
- ۷ - فراخوانی با نتیجه .. Call by result
- ۸ - فراخوانی با ارجاع ... Call by reference
- ۹ - فراخوانی بانام Call by name
- ۱۰ - شاخص دار Indexed
- ۱۱ - فراخوانی با مکان ... Call by location

(تاریخ دریافت: ۲۱ اردیبهشت ۱۳۶۱)



در نظر می گیریم، که در آن زیرروال F به هر یک از دو کمیت داده شده یک واحد می افزاید.

```
begin integer I; integer array A[1:2];
  procedure F(X,Y); integer X,Y;
    begin X:=X+1; Y:=Y+1;
  endproc F;
  I:=1; A[1]:=5; A[2]:=10;
  F(I,A[1]); Print(I,A[1],A[2]);
end
```

در این مورد چنانچه انتقال پارامترهای واقعی I و A[1] توسط نام باشد، پیکره F در لحظه ای قبل از اجرا بصورت زیر خواهد بود:

```
begin I:=I+1; A[1]:=A[1]+1;
endproc F;
```

مقادیری که در اجرای برنامه فوق درخروجی چاپ می شوند بترتیب عبارتند از ۲، ۵، و ۱۱. در حالی که اگر فراخوانی با ارجاع باشد، مقادیر خروجی بترتیب ۲، ۶، و ۱۵ خواهند بود.

بطور کلی وقتی پارامترهای واقعی در فراخوانی متغیرهای ساده یا شاخص دار یا شاخص ثابت باشند، نتایج حاصل از فراخوانی با نام و فراخوانی با ارجاع یکسانند. ولی اگر بعضی از پارامترهای واقعی متغیرهای شاخص دار یا شاخص متغیر باشند، چنانچه دیدیم ممکن است نتایج متفاوت باشند.

بنظر می رسد که در فراخوانی با نام، وقتی پارامترها متغیرهای شاخص دار باشند، در موارد زیادی ممکن است نتایجی دور از انتظار یا نامطلوب حاصل گردد. بعنوان مثال ۲ را برای فراخوانی SWAP(I, A[1]) با در نظر گرفتن فراخوانی بانام مورد بررسی قرار دهید.

این نوع فراخوانی اشکال دیگری نیز دارد که به متغیرهای شاخص دار مربوط می شود. در مثال ۵ توجه می کنیم که بمحض تغییر مقدار شاخص I، نشانی پارامتر واقعی مربوطه تغییر می کند. بعبارت دیگر، نشانی پارامتر واقعی می تواند بطور پویا در متن روال فراخوانده شده متغیر باشد. بنابراین باید برای هر مراجعه به هر پارامتر صوری در روال فراخوانده شده، نشانی مربوطه مجدداً محاسبه گردد، که این امر باعث نقصان کارآیی است. ولی بخاطر می آوریم که در فراخوانی با ارجاع، نشانی پارامتر واقعی فقط یک بار، آنهم قبل از فراخوانی روال مربوطه، تعیین می شود.

نتیجه گیری

بطور کلی، زبانها و همگردانه های مختلف، برای انتقال پارامترها، قواعد و قراردادهای مخصوص بخود را دارند. در

استفاده می کنند، فراخوانی توسط ارجاع انجام شود. زیرروال ENT1 مجموع N عنصر اول بردار داده شده X را تعیین کرده و در متغیر ساده SUMA قرار می دهد. زیرروال ENT2 مجموع اعداد منفی را، که در بین K عنصر اول بردار مفروض Y هستند، محاسبه نموده و در متغیر ساده SNEG قرار می دهد. برنامه اصلی میانگین ریاضی هر دسته از عناصر را، با تقسیم کردن هر یک از مجموعها بر تعداد مربوطه، تعیین و در خروجی چاپ می کند.

برنامه فراخواننده:

```
DIMENSION A(100), B(100)
READ N, K, A, B
CALL ENT1(A, N, SUMA)
CALL ENT2(B, K, SNEG, M)
AV1 = SUMA / N
IF (M.EQ. 0) ...
AV2 = SNEG / M
WRITE ... AV1, AV2, ...
```

...

زیرروال چنددخولی:

```
SUBROUTINE ENT1(X, I, SUMA)
DIMENSION X(100), Y(100)
SUMA = 0
DO TO J = 1, I
  TO SUMA = SUMA + X(J)
RETURN

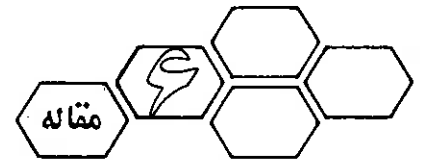
ENTRY ENT2(Y, K, SNEG, M)
SNEG = M = 0
DO 20 I = 1, K
  IF (Y(I).GE. 0) GO TO 20
  SNEG = SNEG + Y(I)
  M = M + 1
20 CONTINUE
RETURN
END
```

فکر می کنید که در این برنامه چه اشکالی پیش خواهد آمد؟ فراخوانی ENT1 سبب می گردد تا متغیر صوری I به پارامتر واقعی N اشاره نماید. در ENT2 برنامه نویس بدون توجه به نقشی I، از آن بعنوان یک متغیر موضعی استفاده کرده است. فراخوانی ENT2 باعث می شود که تغییرات حاصل در I، مقدار N را در برنامه اصلی خراب کند. نتیجتاً در برنامه اصلی، بعضی آنکه SUMA بر مقدار N مورد نظر برنامه نویس تقسیم شود، بر K تقسیم گردیده و نتیجه مورد نظر حاصل نخواهد شد.

۴ - فراخوانی با نام

انجام این نوع فراخوانی معادل آن است که در متن روال فراخوانده شده در یک لحظه قبل از اجرای آن، تمام متغیرهای صوری را توسط پارامترهای واقعی نظیرشان (ونه مقدار این پارامترها) جایگزین کنیم. بهمین سبب آن را فراخوانی توسط جایگزینی هم می گویند. مطلب را همراه با یک مثال ساده دیگر ادامه می دهیم.

مثال ۵ - برنامه شبیه الگول زیر را



طرح چندزبانه کردن چاپگرهای کامپیوتری "لای ۳۶" و "لای ۱۸۰"

حسین فیروزیخت و امیراسعد انزانی
دانشکده علوم، دانشگاه تهران

یادداشت سردبیر: در شماره ۴۴ ماهنامه گزارش کامپیوتر (اردیبهشت ۱۳۶۱)، ضمن ارائه گزارش برگزاری سمینار شورای عالی انفورماتیک کشور، اشاره ای هم به چاپگر فارسی که توسط جهاد دانشگاهی دانشگاه تهران ارائه شده بود داشتیم. آنچه که در زیر بنظر خوانندگان گرامی ماهنامه می رسد خلاصه ای از مشخصات فنی طرح فوق الذکر از زبان مجریان آن است.

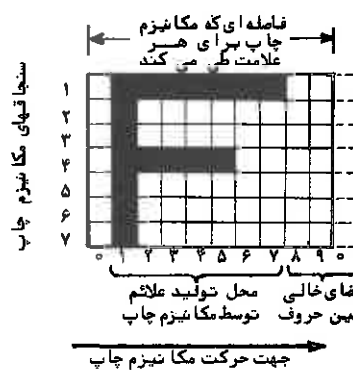
پایانه چاپگر "لای ۳۶" یکی از دستگاه های ورودی و خروجی ساخت شرکت دیجیتال اِکوپیمنت^۱ است که می تواند با سرعت ۳۰ علامت در ثانیه، تا ۱۲۲ علامت را در یک سطر چاپ کند. چاپ علامت در این دستگاه توسط شاخکهای نوشتن در یک ماتریس نقطه ای ۷x۷ صورت می گیرد.

چاپگر ستری "لای ۱۸۰"^۲ یکی از چاپگرهای سریع شرکت دیجیتال اِکوپیمنت است که می تواند با سرعت ۱۸۰ علامت در ثانیه، تا ۱۲۲ علامت را در یک سطر چاپ نماید. چاپ حروف و علامت توسط شاخکهای نوشتن در یک ماتریس ۷x۷ و با استفاده از روش ماتریس ضربه ای صورت می گیرد.

کنترل سخت افزار در چاپگرهای "لای ۳۶" و "لای ۱۸۰"، برعهده واحد کنترل ریزبرنامه نویسی شده^۴ و ریزبرنامه های موجود در حافظه کنترل فقط خواندنی است. این ریزبرنامه ها، که شامل روالهای مختلفی بنا بهای "ورودی"، "چاپ"، "سرعت"، و غیره هستند، از ۵۱۲ دستورالعمل هشت بیتی تشکیل می شوند. هر دستورالعمل در ۱/۱۸۴ میلیونیم ثانیه اجرا می شود و در آن چهار بیت به داده و چهار بیت برای مشخص کردن نوع دستورالعمل اختصاص یافته است.

واحد کنترل ریزبرنامه نویسی شده با توجه به کُدر هر علامت، شکل آن را از یک حافظه فقط خواندنی مخصوص^۵ بدست آورده و در یک ماتریس ۷x۷ چاپ می کند. حرکت شاخک نوشتن با کدهای مختلفی که از واحد کنترل به سِرُموتور محرک داده می شود به چپ و یا راست خواهد بود. بعد از چاپ شدن یک سطر، علامتی به یک موتور پله ای^۶ داده می شود که کاغذ را بالا می برد و با برگشت شاخک نوشتن به ستون اول، دوباره چاپ از ستون اول آغاز می گردد.

از آنجا که حروف لاتین از چپ به راست تحریر می گردند، بنا براین حرکت مکانیزم چاپ از چپ به راست انجام می پذیرد، و چون مابین حروف لاتین فاصله وجود دارد، لذا واحد کنترل پس از هر علامت، بطور خودکار ۳ ستون ماتریسی خالی رد می نماید تا این فاصله ایجاد شود (شکل ۱). مراحل اجرائی این طرح مشتمل بر دو فاز بود که در فاز اول، دوزبانه کردن چاپگرهای مذکور (فارسی و لاتین)، و در فاز دوم چندزبانه کردن این چاپگرها (فارسی، لاتین، یونانی، و غیره بطور آمیخته) مورد نظر بود.



شکل ۱ - نحوه تشکیل حروف و علامت لاتین در ماتریسهای ۷x۷.

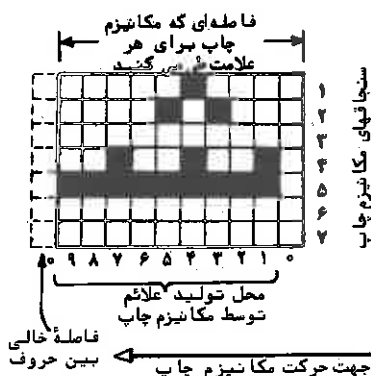
در فاز اول سه موضوع اساسی مورد توجه قرار گرفت:

الف - تغییر جهت حرکت مکانیزم چاپ بطوری که علامت فارسی را از راست به چپ تولید نماید: با توجه به اینکه تمامی سیستم تحت نظارت واحد کنترل ریزبرنامه نویسی شده کار می کند، لذا برای تغییر جهت حرکت مکانیزم چاپ، که در حالت اولیه (لاتین) از چپ به راست است، تغییراتی در ریزبرنامه های موجود در حافظه کنترل داده شد. واحد کنترل ریزبرنامه نویسی شده در وضعیتهای مختلف با فرستادن کدهای مناسبی، که در روال "سرعت" تولید می شوند، سِرُموتور را قادر به حرکت در جهت مثبت یا منفی می نماید. لذا با تغییر الگوریتم روال "سرعت" در ریزبرنامه ها، می توانیم باین هدف برسیم. لازم به تذکر است که چون سِرُموتور در هر مرحله از حرکت، وضعیت خود را به واحد کنترل اطلاع می دهد، بنا براین می توانیم با تغییر دادن این خروجی و پاره ای تغییرات دیگر، بدون اصلاح ریزبرنامه ها نیز حالت ماشین را عکسوار^۷ بنماییم.

ب - طراحی حروف و علامت فارسی و انتخاب مناسبترین و اقتصادی ترین حافظه که قابل تعبیه در سیستم باشد: بمنظور ایجاد هماهنگی در تبادل اطلاعات فارسی،

کود استاندارد^۸ "اینسپک"۷ سازمان برنامه و بودجه (گزارش کامپیوتر، شماره ۱۵، شهریور ۱۳۵۹، صفحات ۳ و ۴) در گذداری و طراحی حروف مورد توجه واقع شد. البته بعلمت محدودیت ماتریس چاپ، پساره ای تغییرات در این گذداری صورت گرفت. طراحی حروف در یک حافظه مناسب، که با کمترین هزینه در سیستم قابل تعبیه باشد، نیز حائز اهمیت بود. برای این منظور حافظه از نوع "شی پرآم"^۸ در نظر گرفته شد. مزیت حافظه های شی پرآم انتخاب شده در قابلیت برنامه ریزی مجدد برای استفاده از انواع نگارشها می باشد. بنا براین موسساتی که از نتیجه این طرح استفاده می نمایند خواهند توانست کد مورد نظر و شکل انواع حروف و علائم را بطور دلخواه تعیین نمایند.

ج - استفاده از ستونهای خالی مابین حروف لاتین بمنظور فراهم آوردن امکان چسباندن حروف فارسی مجاور از قبیل "ب"، "پ"، "ج"، "ع"، و غیره: چون در خط لاتین مابین حروف مجاور فاصله وجود دارد، بنا براین واحد کنترل بطور خودکار مابین حروف ۳ ستون ماتریسی فاصله می گذارد. از آنجائی که در فارسی حروف مجاور به یکدیگر می چسبند، استفاده از این ستونهای خالی در فارسی لازم می آمد. لذا با یک مدار ویژه و ایجاد تغییراتی در ریزبرنامه ها، ماشین از کنترل خودکسار خارج گردید و کنترل دو ستون از سه ستون خالی در حالت فارسی میسر گشت. با این تغییرات، حروف مجاور در ماتریس ۷x۹ طراحی شدند (شکل ۲).



شکل ۲ - نحوه تشکیل حروف و علامت فارسی در ماتریسهای ۷x۹.

در فاز دوم از این طرح، عملکرد سه دو صورت زیر مورد توجه قرار گرفت:

الف - در شروع کار ماشین بطور خودکار در حالت "لاتین" قرار می گیرد و ماشین وضعیت سابق خود کار می کند.

ورق بزنید

Analog transmission	فرارفتی قیاسی، مخابره قیاسی	شده و یا تقدّم و تاخّر معادلهای پیشنهادی تغییر یافته است.	
Analogy	قیاس، تشابه	۴ - علامت "+" واژه‌هایی را مشخص می‌کند که بلافاصله پس از آنها در واژه‌نامه	
Analysis	تحلیل، آنالیز	مقدمه‌ای، واژه‌ها و اصطلاحات ترکیبی	
Analyst	تحلیل گر، آنالیزگر	آمده‌اند. تغییرات پیشنهادی برای این	
Analytic	تحلیلی، آنالیزیک	واژه‌ها باید در مورد ترکیبات آنها نیز اعمال شوند.	
Analytical	تحلیلی، آنالیزیک		
Analyze	تحلیل کردن، آنالیز	۵ - بدیهی است که بمرور زمان ممکن است تغییرات و اضافات دیگری نیز پیشنهاد	
Analyzer	تحلیل کننده، آنالیزگر	شوند که نتایج مربوطه بتدریج در این ستون اعلام خواهند شد.	
Ancestor	نیا	اعضای گرامی انجمن و دیگر خوانندگان ما هم‌اکنون می‌توانند برای ابراز نظرات خود و یا دانستن علل تغییرات اعمال شده کتباً یا شفاهاً با کمیته تماس حاصل نمایند. کمیته بازمینی واژه‌نامه امیدوار است که از نظرات سازنده ماحصلان و علاقمندان برخوردار شود. ○	
* Annex	ضمیمه، پیوست	کوتاه نویسی، مختصر کردن	
* Append	پیوستن	Abbreviation	کوتاه نوشت، اختصار، مختصر
Applicable	کاربردی، قابل اعمال	* Abort	واپس زدن، واپس زدگی
Application-oriented	کاربردی	* Aborted	واپس زده
Apply	بکار بستن، اعمال کردن، درخواست دادن	* Abortion	واپس زدگی
Approach	نشی، نزدیک شدن، راه	Acceptability	مقبولیت، پذیرفتگی، میزان پذیرفتگی
* Arbitrary	دلخواه	Acceptable	مقبول، پذیرفتنی، قابل پذیرش
Arithmetic expression	عبارت حسابی	Acceptor	پذیرنده، پذیرا
Arithmetic/logic unit	واحد حساب و منطق، واحد عملیات حسابی و منطقی	Accessories	لوازم تکمیلی، لوازم بدکی
Arm	بازو	Accessory	تکمیلی، مکمل، فرعی، بدکی
* Aspect	جنبه	Adapt	وفق دادن، تطبیق دادن
Asset	دارائی، سرمایه	* Adaptation	وفق، تطبیق، انطباق
Assign	منسوب کردن، نسبت دادن، تخصیص دادن	* Adaption	وفق، تطبیق، انطباق
Assignment	انتساب	Additional	اضافی، دیگر
Assignment statement	حکم انتسابی	Address modification	اصلاح نشانی
Assistant	دستیار	Address part	بخش نشانی
Associate	همبسته، همبسته کردن، مربوط کردن	Adequacy	کفایت، بسندگی
Association	همبستگی، همبسته سازی، انجمن	Adjacency	مجاورت، همجاری، نزدیکی
* Associative	همبسته، انجمنی، شرکت پذیر	Adjacent	مجاور، همجوار، نزدیک
Astable multivibrator	تکانه ساز ناپایا	Adjust	تعدیل کردن، تنظیم کردن، میزان کردن
Asterisk	ستاره	Administrative	اداری، مدیریتی
Asynchronous	ناهمگام، ناهمزمان	Admissible	مجاز، قابل قبول
Asynchronous transmission	فرارفتی ناهمگام، مخابره ناهمگام	Aggregate	متراکم، انباشته، کلّ
Attach	الماق کردن، ضمیمه کردن	Aggregation	تراکم، انباشت، جمع
Attachment	الماق، ضمیمه، انضمام	Air gap	فاصله هوایی
Attenuate	میراشدن، خفیف شدن	Alarm	هشدار، هشدار دادن، هشداردهنده، اعلان خطر
Attenuation	میراشی، تخفیف	Alert	گوش بزرگ، هوشیار
Attribute	صفت خاصه، نسبت دادن	+ Alphameric	القیاعدی، الفعددی
Audible alarm	هشدار شنیدنی، آژیر	+ Alphanumeric	القیاعدی، الفعددی
Audible signal	علامت شنیدنی	* Alphanumeric	القیاعدی، الفعددی
Automatic exchange	تبادل خودکار، مرکز تبادل خودکار	Alter	تغییر دادن
Automatic testing	آزمودن خودکار	+ Alternation	تغییر
Auxiliary operation	عمل کمکی، عملیات کمکی	Alternate	متناوب بودن، متناوب کردن، دیگر
Availability	دسترسی پذیری، در دسترس بودن		
Available	دسترسی پذیر، در دسترس، فراهم		
Average seek time	مدت متوسط پیگردی		
Axis	محور		
Axle	محور چرخ، میله		



ب- در شروع ماشین بطور خودکار در حالت "فارسی" قرار می‌گیرد. در این حالت ما قادر خواهیم بود که با ارسال برخی از علائم کنترل، مانند CTRL A، CTRL B، و غیره، به انواع خطا و علائم، از جمله فارسی، یونانی، لاتین (لاتین در شی پروم جداگانه طراحی شده)، و غیره در یک سطر چاپ دسترسی یابیم. البته با توجه به قابلیت حافظه شی پروم، می‌توانیم با محدود کردن تعداد حروف هر زبان (خط)، حروف فارسی، لاتین، و یا علائم ریاضی را با کدهای غیراستاندارد داشته باشیم. بمبارت دیگر مثلاً می‌توان حروف لاتین را بجای علائم کنترل قرارداد (با فشار دادن CTRL A، علامت A، با فشار دادن CTRL B، علامت B، و غیره چاپ گردد). ●

یادداشتها

1. LA36 hard copy terminal
2. DEC = Digital Equipment Corp.
3. LA180 line printer
4. MPC = Micoprogrammed Controller
5. ROM character generator
6. Stepper motor
7. INSIC = Iranian National Standard Information Code
8. EPROM = Erasable Programmable Read-Only Memory

(تاریخ دریافت: ۱۶ خردادماه ۱۳۶۱)

واژه‌ها



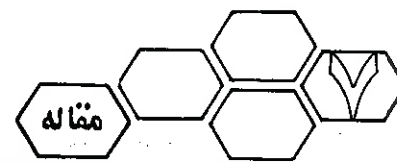
تغییرات و اضافات واژه‌نامه مقدّماتی

کمیته بازمینی واژه‌نامه درنشتهای متعدد خود، بازمینی واژه‌هایی را که با حرف A شروع می‌شوند به پایان رسانده است. ذیلاً نتیجه این بازمینی و تغییرات و اضافاتی که توسط کمیته صورت گرفته است جهت اطلاع علاقمندان درج می‌شود. قبل از ارائه فهرست واژه‌ها، یادآوری نکاتی را لازم می‌دانیم:

۱ - تعدادی از واژه‌های پرکاربرد جداگانه به نظرخواهی گذاشته شده‌اند (گزارش کامپیوتر، شماره ۳۴، اردیبهشت ۱۳۶۱) و لذا تکرار آنها را اینجا جایز ندیدیم.

۲ - واژه‌های جدیدی که در واژه‌نامه مقدّماتی وجود ندارند با علامت ستاره مشخص شده‌اند.

۳ - معادل بعضی از واژه‌ها کلاً تغییر یافته است و برای بعضی دیگر، معادل یا معادلهای جدیدی به معادلهای قبلی اضافه



آشنائی با زبان برنامه‌نویسی "پرولوگ"

ابراهیم تقی‌زاده، مباحث
مربی مجتمع دانشگاهی علوم

یادداشت سردبیر: در مقاله "تحول بازار زبان لیسپ" (گزارش‌کا مپیوتر، شماره ۳۲، بهمن و اسفند ۱۳۶۰)، اشاره‌ای باین نکته شده بود که موفقیت تجاری زبان لیسپ مبنای طراحی زبانهای سطح بالاتری در زمینه هوش مصنوعی، و از آن جمله زبان پرولوگ، گشته است. هدف از درج مقاله‌ای که در زیر می‌خوانید آشنا نمودن خوانندگان گرامی ما هنام با زبان برنامه‌نویسی پرولوگ، بعنوان نمونه‌ای از زبانهای برنامه‌نویسی ویژه برای کاربردهای هوش مصنوعی و علمی-الخصوصی حل مسائل بصورت خودکار، است. مقاله حاوی تشریح مقدماتی نحوه زبان پرولوگ و مثالهای ساده‌ای از برنامه‌ها و نحوه اجرای آنها است.

در زبان پرولوگ، جملات عمدتاً "بر مبنای منطق تفکیک‌پذیری" بنا شده و از نوع تشریحی شکل "X درست است" می‌باشند، حال آنکه جملات زبانهای متعارف برنامه‌نویسی از نوع امری و شکل "عمل X را انجام بده" هستند. برای بدست آوردن یک زبان برنامه‌نویسی بر مبنای منطق تفکیک‌پذیری، جملاتی امری بصورت "ثابت کن که X درست است" را به آن می‌افزاییم. جملاتی باین شکل را "جمله هدف" می‌نامیم.

یک برنامه "پرولوگ شامل مجموعه‌ای از جملات هدف و نیز مجموعه‌ای از اصول موضوعه است. اصول موضوعه از نوع جملات تشریحی هستند و لیستی از "واقعیات" را تشکیل می‌دهند. هر جمله هدف بشکل امری است و باعث می‌شود که (در صورت امکان) از اصول موضوعه برای اثبات واقعیت مشخصی استفاده گردد.

پردازش اثبات در زبان پرولوگ از اصل تفکیک‌پذیری در روش جستجوی به‌ترتیب ژرفا استفاده می‌نماید. این پردازش اثبات کامل نیست زیرا ممکن است شاخه‌ای را در درخت جستجو بصورت نامحدود تعقیب نماید و هیچگاه شاخه دیگری را، که ممکن است به اثبات قابل قبولی بیانجامد، در نظر نگیرد. این روش ساده جستجو، با وجود ناکام بودن، دارای مزایای بسیاری نسبت به روشهای جستجوی عمومی تر (جستجو به ترتیب پهنا) می‌باشد که یکی از عمده‌ترین این مزایا، نیاز کمتر به حافظه است. سادگی روش جستجوی پرولوگ به برنامه‌نویس امکان می‌دهد تا به آسانی آن را فهمیده و تحت کنترل قرار دهد.

اساسی‌ترین واحد نحوی در زبان پرولوگ "عبارت" است، که می‌تواند یکی از موارد زیر باشد:

الف - یک مقدار ثابت، مشتمل بر دنباله دلخواهی از حروف و ارقام (مانند ABC یا X2).

ب - یک متغیر، که توسط علامت ستاره همراه با دنباله‌ای از حروف و ارقام نشان داده می‌شود. (مانند A1 یا *Y).

پ - یک شالوده^۱، که توسط نام آن و لیستی از یک یا چند پارامتر (بین دوکمان) نشان داده می‌شود. (مانند F(X2,*Y) یا (G(*B,A,F(3)).

جملات هدف و اصول موضوعه پرولوگ، از "الفاظ" تشکیل شده‌اند. هر لفظ یا مقداری ثابت و یا یک شالوده است. "محمول" نامی است که به یک لفظ تخصیص می‌یابد. چنانچه لفظ شالوده باشد، محمول همان نام شالوده است.

شکل کلی یک اصل موضوعه پرولوگ چنین است:

<بدنه اصل موضوعه> <سر اصل موضوعه>

سر اصل موضوعه یک لفظ و بدنه آن یا یک لفظ و یا ترکیبی عطفی از الفاظ (که با علامت & از یکدیگر جدا شده‌اند) می‌باشد. بعنوان مثال معنی اصل موضوعه

$A + B \& C.$

چنین است: "A از B و C نتیجه می‌شود" و یا عبارت دیگر، "برای اثبات A، باید ابتدا B و سپس C را اثبات نمود". اصل موضوعه‌ای را که بدنه نداشته باشد "واحد" می‌نامند. برای مثال، $F(M).$ یعنی $F(M)$ درست است.

شکل کلی جمله هدف پرولوگ عبارت است از:

<ترکیب عطفی هدف> +

ترکیب عطفی هدف یا یک لفظ و یا ترکیبی عطفی از الفاظ است. بعنوان مثال:

$+ P.$

$+ Q(R) \& F.$

پرولوگ باید جملات هدف را "اثبات" نماید. کلیه متغیرها در اصول موضوعه و نیز در عبارات، بصورت "سوردار فراگیر"^{۱۱} در نظر گرفته می‌شوند. یعنی اصل موضوعه‌ای که شامل متغیر باشد برای تمام مقادیر آن متغیر معتبر است. بعنوان مثال، معنی

$FATHER(*X,*Y) + SON(*Y,*X)$

چنین است: "برای تمام مقادیر X و Y، پدر Y است اگر Y فرزند X باشد".

اجرای برنامه پرولوگ با یک جمله هدف آغاز می‌شود. جمله هدف در واقع درخواستی برای یک اثبات است.

برای تشریح مراحل پیشرفت اثبات در یک برنامه پرولوگ، می‌توان از نمودار زیر

هایی که "درخت ایجاب"^{۱۲} نامیده می‌شوند استفاده کرد. هر درخت ایجاب شامل یک یا چند گره برچسب‌دار می‌باشد. گره‌ای که در رأس نمودار واقع است با برچسب "هدف" مشخص می‌شود و برچسب سایر گره‌ها از الفاظ است. هر گره می‌تواند در یکی از سه وضعیت قرار داشته باشد:

۱ - باز: هنوز هیچ کوششی جهت اثبات لفظی که برچسب‌گره می‌باشد صورت نگرفته است. چنین گره‌ای دارای فرزند نیست.

۲ - بسته: لفظی که برچسب‌گره است، با استفاده از اصل موضوعه مربوط به آن لفظ، اثبات گردیده است. چنین گره‌ای را با علامت "x" مشخص می‌سازیم. گره بسته نیز فرزندی ندارد.

۳ - فعال: لفظ برچسب‌گره، با استفاده از اصول موضوعه در دست‌اثبات است. برچسبی که به چنین گره‌ای داده می‌شود، لفظ سر اصل موضوعه است. برچسب‌هایی که به فرزندان این گره داده می‌شود نیز بترتیب از چپ به راست الفاظ بدنه اصل موضوعه می‌باشند.

برای روشن شدن مفاهیم فوق، اصول موضوعه و جمله هدف زیر را در نظر بگیرید:

$A + B \& C.$

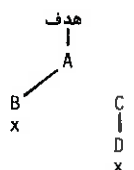
B.

C + D.

D.

+ A.

اثبات هدف فوق توسط درخت ایجاب زیر نشان داده شده است:



نمودار فوق یک درخت ایجاب کامل است، زیرا تمام گره‌های آن فعال یا بسته هستند. گره‌هایی که دارای برچسبهای B و D هستند بترتیب با استفاده از اصول موضوعه "B." و "D." بسته شده‌اند. گره‌ای که با برچسب A مشخص شده فعال است و با استفاده از اصل موضوعه "A+B&C" اثبات شده است.

حال به ارائه مثال دیگری می‌پردازیم. اصول موضوعه و جمله هدف زیر را در نظر بگیرید:

$A + B \& C.$

$B + D \& F.$

$B + E \& F.$

C + G.

E + G.

F + H.

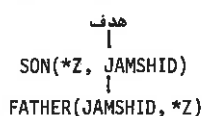
G.

H.

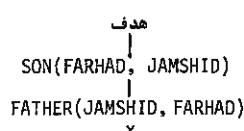
+ A.

وضعیت اولیه اثبات، توسط نمودار زیر ورق زیند

SON(*Z, JAMSHID) ← FATHER(JAMSHID, *Z)
در این مرحله شکل درخت بصورت زیر است:



یافتن جانشینهایی برای مشابه ساختن دو عبارت را "یکسان سازی" می نامیم. حال می خواهیم اصل موضوعه ای نیابیم که سر آن با FATHER(JAMSHID, *Z) یکسان گردد. چنانچه FARHAD را جانشین *Z سازیم، اولین اصل موضوعه برای FATHER مطابقت خواهد کرد و درخت ایجاب زیر نتیجه خواهد شد:



مثالهای زیر نحوه عمل با ساخت داده های پیچیده تری را در زبان پرولوگ نشان می دهند. مجموعه ای از عناصر را توسط یک عبارت با استفاده از نماد سازنده S و نشانه انتهایی NIL نشان می دهند. بعنوان مثال، مجموعه ای بسا عناصر، A، B، و C بصورت S(A, S(B, S(C, NIL))) نمایش داده می شود. مجموعه تهی را فقط با NIL نشان می دهند. با توجه به نحوه نمایش فوق برای مجموعه، رابطه "عضویت در مجموعه" را می توان بصورت زیر تعریف کرد:

ELEMENT(*X, S(*X, *Y)).

ELEMENT(*X, S(*Y, *Z)) +
ELEMENT(*X, *Z).

این دو اصل موضوعه را می توان باین شکل تفسیر کرد: "X عضوی از مجموعه است اگر یا اولین عنصر مجموعه باشد و یا عضوی از مجموعه عناصر باقی مانده".

مجموعه را می توان به شکل ساده تری با استفاده از نشان گذاری میانوندی^{۱۷} نمایش داد. اگر از علامت "." نیز بعنوان نماد سازنده استفاده نمائیم، در این صورت مجموعه ای با عناصر A، B، و C بصورت A.B.C.NIL نشان داده می شود. اصول موضوعه ELEMENT برای این نحوه نمایش مجموعه بصورت زیر می آیند:

ELEMENT(*X, *X.*Y).

ELEMENT(*X, *Y.*Z) + ELEMENT(*X, *Z).

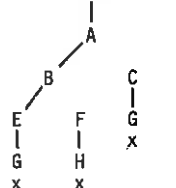
از اصول موضوعه زیر می توان برای نوشتن تمام عناصر یک مجموعه استفاده کرد:

LIST(*X.*Y) + WRITE(*X) & LIST(*Y).

LIST(NIL).

دنباله در صفحه ۱۵

هدف



اثبات نهایی توسط درخت ایجاب کاملی نمایش داده شده است. البته چنانچه اثبات موفقیت آمیز نباشد، درخت ایجاب هیچگاه کامل نمی شود. برای نمونه چنانچه در این مثال، اصل موضوعه C + G را حذف نمائیم، اثبات به نتیجه نخواهد رسید. از طرف دیگر چنانچه اصل موضوعه دیگری مانند D + D را اضافه کنیم، اثبات کننده سعی در ساختن یک "شاخه نامتناهی" در درخت خواهد کرد.

اصول موضوعه پرولوگ می توانند حاوی متغیر نیز باشند. برای مثال، اصل موضوعه

SON(*X, *Y) ← FATHER(*Y, *X).

جمله "X فرزند Y است اگر Y پدر X باشد" را نمایش می دهد.

در هر اصل موضوعه و یا جمله هدفی می توان عمل "جانشینی" را انجام داد. با این عمل، یک عبارت جانشین تمام رویدادهای یک متغیر می شود. عبارت جانشین می تواند یک مقدار ثابت، یک شالوده، و یا متغیر دیگری باشد. هنگامی که عمل جانشینی یک بار یا بیشتر بر روی عبارت با اصل موضوعه ای صورت گیرد، نتیجه را یک "مورد" از آن عبارت با اصل موضوعه می نامند.

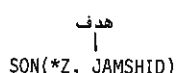
برای روشنتر شدن مفهوم جانشینی،

مثال زیر را در نظر بگیرید:

SON(*X, *Y) ← FATHER(*Y, *X).
FATHER(JAMSHID, FARHAD).
FATHER(JAMSHID, HASSAN).
FATHER(ALI, BEHZAD).
FATHER(HASSAN, ALI).

می خواهیم هدف SON(*Z, JAMSHID) +

را حل کنیم. منظور ما از "حل کردن یک هدف" یافتن موردی از هدف است که بتوان آن را اثبات نمود. مراحل اثبات ذیل^{۱۸} توسط درختهای ایجاب نشان داده شده اند. درخت اولیه چنین است:



حال، نیاز به یافتن موردی از یک اصل موضوعه داریم که بتوان از آن در اثبات SON(*Z, JAMSHID) استفاده نمود. مورد مطلوب از روی

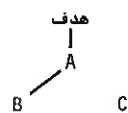
SON(*X, *Y) + FATHER(*Y, *X)

و جانشین ساختن *Z بجای *X و JAMSHID بجای *Y بدست می آید. عبارت حاصله چنین است:

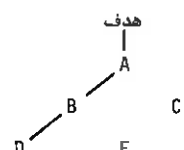
نشان داده می شود:



اولین اصل موضوعه برای A انتخاب می شود:



اثبات کننده همواره به ترتیب ژرفا و از چپ به راست عمل می کند. لفظ بعدی که باید اثبات شود B است. اصل موضوعه B + D & F انتخاب می شود:



اثبات کننده باید اکنون D را اثبات نماید، ولی اصل موضوعه ای برای D موجود نیست. در نتیجه اثبات کننده برای یافتن راههای دیگر به عقب باز می گردد. ترتیب انتخاب اصول موضوعه همواره بهمان ترتیبی است که در برنامه ظاهر شده اند. بنابراین در مثال فوق، B + D & F همیشه قبل از B + E & F امتحان می شود.

در این مثال، پردازش اثبات در محل انتخاب اصل موضوعه B + D & F متوقف گشته است. اثبات تا این نقطه ردیابی^{۱۹} شده و اصل موضوعه دیگری (B + E & F) انتخاب می شود. اثبات به ترتیب زیر ادامه می یابد:

