۱۶

به‌نام خدا

سرمقاله ۳
گپ صنفی: تغییرات آموزشی و انتظار از بچه‌ها ۴
ما مخالفیم! ۷
ایران هم قانون کپی‌رایت دارد! ۹
گپ علمی؛ طراحی الگوریتم ۱۰
حل و بررسی پستچی چینی ۱۳
زندگی‌نامه کنوث ۱۵
T_EX چیست؟ ۱۷
مصاحبه با کنوث ۱۹
دانشگاه مجازی ۲۰
هنر تست‌کردن نرم‌افزارهای کامپیوتری ۲۳
حمله و دفاع ۲۴
IP v6 می‌آید! ۲۵
آشنایی با ASP.NET ۲۸
کامپیوتر، علمی شاعرانه ۲۹
سوالاتی از مفاهیم کامپیوتر ۳۰
نظارت در کامپیوتر ۳۱
سخت‌افزار تکامل‌پذیر ۳۳
پارادکس‌های زبان‌های هوش‌مصنوعی ۳۶
مبنای یک و صفر ۳۷
ریاضی‌دانی که ریاضی‌دان ساخت ۳۸
مند در هوش مصنوعی چیست؟ ۴۲
المپیاد دانشجویی ۸۱ ..... ۴۶
فلاسفه از کامپیوتر چه می‌فهمند؟ ۴۸
کلمات قصار کامپیوتری ۴۹
سرمقاله؟ ۵۰
فیلم‌شناسی ۵۰
روزگار ما ۵۱
تذکره الاساتید ۵۱
گذری کوتاه بر موسیقی کلاسیک و پاپ ۵۲
دو سوال اخلاقی ۵۳
جستاری از میشل دِ مونتن ۵۴
معلقات سبع ۵۶
پنجره ۵۷

مدیر مسوول:

سید احسان لواسانی

سردبیر:

زیر نظر تیم سردبیری

هیات تحریریه:

حامد پیشوایزدی

رضا حسامی‌فرد

هانی جوان‌همت

سولماز مسعودیان

شعله هاشمی

کاریکاتوریست:

هانی جوان‌همت

مدیر پویش در وب:

پدرام تدینی

و با تشکر از:

دیگران

نشریه دانشجویان دانشکده کامپیوتر؛ پویش

دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر

aku_pooyesh@yahoo.com

سرمقاله

چهره در هم کشید و روی برتافت

چرا که نابینایی به سراغ او آمده بود

تو چه می‌دانی، شاید او پاکی پیشه کند؟

یا متذکر گردد و این یادآوری به حال او مفید باشد.

اما تو به آن کس روی می‌آوری

که مستغنی است.

سورهٔ عبس

تغییرات نظام آموزش و انتظار از بچه‌ها

مصاحبه با دکتر سعادت پورمظفر

حامد پیشوایزدی

من اعتقاد کامل دارم که باید قوانین در یک محیط آموزشی برقرار باشد. مقررات باید رعایت شوند. هنوز بعضی از دانشجویان آگاهی کامل ندارند و به کلاس رفتن توجه زیادی نمی‌کنند. ما باید جلوی این امر را بگیریم. از طرفی ما حتی اساتید را حضور و غیاب می‌کنیم. اگر استادی نبود ما به او اعلام می‌کنیم و بعد به استاد می‌گوییم که باید زمان کلاس جبرانی را به ما اعلام کند.

شورای آموزش و پژوهش دانشکده شدیداً از این امر استقبال کرده است. مجموعه اساتید دانشکده، یک مجموعه‌ی همگن و جزو بهترین مجموعه‌ها می‌باشد. ما می‌توانیم برویم طبقه دوم، اتاق‌های بزرگ را در دست بگیریم و آزمایشگاه‌ها را به office (دفتر) تبدیل کنیم. ولی اگر بودجه‌ای در دست باشد که بواسطه آن برای دفتر اساتید و یا دانشکده وسیله‌ای خریداری کنند، همگی رای می‌دهند که بودجه وقف دانشکده شود. یعنی دانشکده را مقدم می‌دانند. هر استادی حق دارد که دفتر کاری مناسب داشته باشد. چون ما اساتید نیز شخصیت اجتماعی داریم و در این محیط زندگی می‌کنیم. ولی برای ما این‌ها مهم نیست. ما به اعتلای دانشکده فکر می‌کنیم. حتی سال‌های پیش نیز در ایام تابستان اساتید از کولر محروم بودند. کامپیوترهای اتاق‌های اساتید همه ۱۰۰ و ۲۰۰ MHZ است ولی ما کامپیوترهای سایت را برای خودمان بر نداشتیم.به نظر من خواست مجموعه‌ی همه اساتید ما پیشرفت دانشکده است.

این یک خواست عمومی است. تا همین الان هم دشمنان زیادی برای، خودمان تراشیده‌ایم. مثلا برای همین پروژه IT از شرق و غرب به ما حمله شده که چرا می‌خواهید این پروژه را در دست بگیرید؟ من فکر می‌کنم ما دانشکده بسیار بسیار خوبی داریم. من از نزدیک با وضعیت بعضی دانشگاه‌های مطرح آشنا هستم. این‌ها اساتید دارند ولی همگی در خارج از کشور هستند و هر ترم از استاد مدعو استفاده می‌شود.

- ***لطفاً در مورد قویتر کردن گرایش سخت افزار دانشکده و تغییر و تحول کادر اساتید توضیح دهید:***

چون احساس کردیم مقطع کارشناسی، در زمینه سخت افزار کمی ضعیف است تصمیم گرفتیم از دکتر کاردان که استاد مدعو هستند (استاد مدعو فقط حق تدریس در کلاسش را دارد و حق نظارت بر ازمایشگاه‌ها را ندارد) با توجه به کاردانی دکتر کاردان و تعهد و سواد ایشان، از همکاری ایشان بیشتر استفاده کنیم.

من خودم را آدم پرکاری می‌دانم و ۱۰ تا ۱۲ ساعت در روز کار می‌کنم ولی هر وقت به دکتر کاردان نگاه می‌کنم انرژی می‌گیرم. بنابراین برای این‌که دانشجویان ما در زمینه کارشناسی سخت افزار پایه‌ی قوی داشته باشند از دکتر کاردان دعوت به همکاری کردیم، که شورای آموزشی و پژوهشی به این کار رای مثبت داده ولی باید دانشگاه نیز موافقت کند تا دکتر کاردان به عنوان استاد ثابت دانشکده پذیرفته شوند.

اگر کار ایشان درست بشود، ما با دکتر کاردان بعنوان یک تیم، تصمیماتی گرفته‌ایم که قصد داریم در آینده آن‌ها را عملی کنیم، آزمایشگاه رباتیک یکی از آن‌هاست.

ما به دانشکده برق نیز گفته‌ایم آزمایشگاه AC/DC را خالی کنند. ما با مشکل کمبود جا روبرو هستیم. تصمیم داریم در دانشکده یک computer shop دایر کنیم و در نظر داریم مسئولیت و مدیریت این مرکز را به خود دانشجوها محول کنیم تا دانشجوها بتوانند این درس را بعنوان یک واحد کارگاهی بگیرند. قصد داریم کارآموزی را در آینده به دو درس یک واحدی تبدیل کنیم که یکی از این‌ها همین

computer shop است . یک دانشجوی کامپیوتر باید با این امور اشنا باشد و این د رحالی است که در صورت وجود computer shop آموختن این امور برای یک دانشجوی کامپیوتر فقط یک ساعت وقت می‌برد ولی تا وقتی عملا با قطعات درون کامپیوتر کار نکند همیشه این قطعات و نحوه ارتباط آنها با هم بصورت مبهمی در ذهن دانشجو باقی خواهد ماند.

- ***لطفاً در مورد درس طراحی سیستمهای آنالوگ و دیجیتال (مباحث ویژه) که اخیراً ارائه شده توضیح دهید:***

در مورد این درس من و دکتر کاردان جلسه‌ای داشتیم و تصمیم براین شد که دکتر آزادنیا این ترم الکترونیک را تدریس کنند و دکتر کاردان درس مباحث ویژه (طراحی سیستم‌های آنالوگ و دیجیتال) را تدریس کنند. قرار است بنشینیم و ببینیم چه کمبودهایی داریم. ما الکترونیکII و مدارهای الکتریکی نداریم. در مباحث ویژه (طراحی سیستم‌های آنالوگ و دیجیتال) پروژه‌هایی نیز همزمان با درس تعریف خواهند کرد. ما قصد داریم در آینده تا جایی که ممکن باشد، آزمایشگاه‌ها را با درس‌های مربوط به آن‌ها به‌صورت هم‌زمان ارائه کنیم. به امید خدا آزمایشگاه‌های ما بروز خواهند شد. آقای دکتر کاردران مشغول تعریف پروژه‌های جدید هستند. من هم خودم، به عنوان پروژه‌های الکترونیک دیجیتال، یک تعداد پروژه به دانشجویان داده‌ام. این پروژه‌ها از کتاب‌های معتبر آزمایشگاهی کشورهای غربی انتخاب شده‌اند و تعدادی نیز از پروژه‌های معتبر آزمایشگاهی کشورهای غربی هستند هرچند که من تعصب خاصی روی کشور خاصی ندارم.

قرا است من و دکتر کاردان این پروژه‌ها را منظم و یکنواخت کنیم و ببینیم کدام یک از آن‌ها به آزمایشگاه مدار مربوط است، کدامیک به آزمایشگاه الکترونیک و کدام به آزمایشگاه الکترونیک دیجیتال. آزمایشگاه FPGA ما مشکل کامپیوتر داشت که حالا این مشکل حل شده است . ما الان قسمتی از آزمایشگاه مدار منطقی را با FPGA حل می‌کنیم یعنی Alterها را شبیه‌سازی می‌کنیم. ان شاء ا... تا اول مهر آزمایشگاه مدار منطقی با FPGA خواهد بود. آزمایشگاه FPGA را هم می‌توانیم به رباتیک و ... گسترش دهیم . قرار است بنده، دکتر کاردان، دکتر رحتمی و دکتر همایون پور روی این امر نظارت کنیم. قرار است آیین‌نامه و اساسنامه‌ای بنویسیم که مستقل از نظارت ما باشد. نظر خود دکتر کاردان این بود که تابستان آزمایشگاهی داشته باشیم که چند اسیلوسکوپ در یک اتاق داشته باشد و کلید آن نیز در دست خود دانشجو باشد و مسئولیت آن نیز با خود دانشجو باشد. ما باید برای دانشجو شخصیت قائل باشیم. اگر هم در جالی خطایی روی داد، آن‌را به تمامی دانشجویان تعمیم ندهیم. دانشجو بخشی از آموزش است.

هدف ما این است که استفاده از دانشکده را ۲۴ ساعته کنیم. الان ما بجای اینکه بنشینیم و راهکارها را بررسی کنیم، می‌خواهیم صورت مسئله را پاک کنیم که این راهی بجایی نمی‌برد. به نظر من دانشجویان ما باید بیش از پیش خود را در امور مربوط به دانشکده سهیم بدانند. باید مشغول به کار شوند. ولی بعضی اوقات دانشجوهای ما این شخصیت بخشیدن به دانشجو را بد تعبیر می‌کنند بنظر من ما در جهت مثبت پیش می‌رویم . ما مثلا همین دیشب عکس سایت دانشکده کامپیوتر را در مجله امیرکبیر دیدم. وضعیت سایت خیلی بهتر شده که زحمت این کار را دکتر همایون‌پور، دکتر اکبری و اساتید دانشکده کشیده‌اند. وضعیت ما رو به بهبود است. ما در یک دنیای کامل زندگی نمی‌کنیم. نمی‌توانیم همه چیز را سیاه و سفید ببینیم. بین این دو رنگ، رنگ‌های دیگری نیز هستند. نمی‌توانیم ۱۰۰٪ خوب باشیم. تکنولوژی و صنعت تصاعدی است . باید روز به روز بهبود پیدا کند تا به مرحله ۷۰٪ -۸۰٪ برسد. همه چیز نسبی است . باید خودمان را با دانشکده‌های دیگر قیاس کنیم. خواست ما این است که ان شاء ا... دانشکده کامپیوتر ما number-one و درجهٔ اول باشد، می خواهیم بهترین باشیم اگر خواست ما چیزی کمتر از این باشد اشتباه است. اگر مجموعه مقالات هفتمین کنفرانس انجمن کامپیوتر ایران را بررسی کنیم، می‌بینیم با وجود اینکه دانشگاه شریف متولی این کنفرانس بوده است ما ۴ مقاله بیشتر داریم. ما در دانشکدهٔ‌ای هستیم که اساتید آن از نظر تحقیقاتی د ر وضعیت خوبی هستند. البته در تحقیقات، به هنوز در بومی کردن صنعت رشد خوبی نداشته‌ایم و خیلی دوریم. ماهنوز از حد ایده‌ال پایین‌تریم، ولی از بسیاری از دانشکده‌های رقیب بهتریم. من فکر می‌کنم ما سرویسی که در اینجا ارائه می‌کنیم از دانشکده برق پیشرفته‌تر است . خلاصه‌ی کلام این‌که کارهایی در دست انجام است که اگر این کارها انجام شود، ظرف یک سال و نیم آینده ما دانشکده‌ی بسیار خوبی خواهیم داشت.

ما در اینترنت روی دانشگاه برکلی، استانفورد و MIT مطالعه می‌کنیم تا از آنها ایده بگیریم و در شورای عالی آموزشی پژوهشی دانشگاه مطرح کنیم باید هم و غم همه‌ی ما، از دانشجو گرفته تا اساتید اعتلای دانشکده و امیرکبیر باشد. در اینصورت همه‌ی ما برنده و پیروزیم و الا اگر به همین سیستم‌های نیمه تمام اکتفا کنیم کاری از پیش نخواهیم برد. دانشجوهای ما بسیار زیرکند و وقتی ببینند که وضعیت دانشگاه و دانشکده بروز می‌شود، انگیزه پیدا می‌کنند و روز به روز توانمندتر می‌شوند.

ما نمی‌توانیم تمام خواست دانشجوها را برآورده کنیم ولی باید حداقل این حسن نیت را نشان دهیم که قصد چنین کاری داریم. ولی نباید این قضیه تعبیر به سوء شود. این درست نیست که دانشجو درست درس نخواند، در کلاس حضور نداشته باشد و ما به او نمره بدهیم. این نوعی خیانت است. کمک به دانشجو نیست.

در یک محیط آموزشی باید مقررات حاکم باشد و دانشجو و استاد موقعیت خودشان را به خوبی بشناسند. دانشجویی که بیشتر درس می‌خواند و در کلاس حضور بیشتری دارد باید نمره بیشتری بگیرد. به نظر من اگر هدف ما، هدف والایی باشد، در ما عشق و انگیزه بوجود می‌آید و همین عشق و انگیزه است که باعث می‌شود ما در کار دوام بیاوریم. اگر انگیزه باشد ۱۲ ساعت وقتی که من می‌گذارم مثل ۱۲ دقیقه خواهد بود و در غیر اینصورت رنج و عذاب است.

درمواقعی که دانشجویان نسبت به استادِ درسی اعتراض دارند ما سعی می‌کنیم با توجه به امکانات موجود و کمبودهایی که داریم شرایط را به

دانشجو تفهیم کنیم و دانشجویان را متقاعد کنیم. من هیچوقت این‌طور نیستم که فکر کنم همیشه حق با من است. بعضی مواقع دانشجویان نسبت به بعضی مسائل انتقاد دارند، من هم می‌گویم انتقاد و توجه به جاست ولی ما راه حلی برای این مشکل نداریم.

بنظر من وقتی واقعیت را به دانشجو بگوییم دانشجو متوجه مسائل خواهد شد. اگر شما ضد استاد مثل دکتر کاردان را به ما معرفی کنید ما آنها را در اسرع وقت استخدام می‌کنیم. ولی بعضی اوقات دست ما بسته است. مثلاً در مورد نرم افزار در سطح کارشناسی ما با مشکل استاد مواجهیم.

بنظر من یک دانشجو باید از حالت‌های هیجانی پرهیز کند و تحت تاثیر جمع قرار نگیرد.

مثلاً گاهی بعضی از دانشجویان تنبل کلاس، باقی دانشجویان را تحریک می‌کنند تا نامه‌ای را امضاء کنند، که این امر از نظر من پذیرفته نیست. باور کنید یک دانشجوی امریکایی کاملاً تسلیم محض شرایط است و احترام متقابل کاملاً جا افتاده است.

تقاضای من از دانشجویان این است که هر مشکلی دارند با ما در میان بگذارند. پیغام دیگر من به دانشجویان این است که اگر امکانات دانشکده بیشتر شود و وضعیت بهبود پیدا کند، ما برنامه آموزشی و درسی مختصری برای دانشجویان خواهیم داشت. ولی ما هیچوقت از برنامه‌های آموزشی قدیمی و از رده خارج شده است نمی‌کنیم.

- ***سوال: در مورد تدریس C در درس مبنای کامپیوتر و C++ یا Java در درس برنامه نویسی پیشرفته؟***

من با آقای پوروطن، دکتر کاردان و دکتر خرسندی دراین مورد صحبت کردم، نظر همه این بود که برای یاد گرفتن برنامه نویسی بطور صحیح زبان پاسکال مورد نیاز است. با وجود این که دانشجویان تمام برنامه نویسی خود را با C انجام می‌دهند ولی باید با زبان پاسکال نیز آشنایی داشته باشند.

تصمیم گرفتیم کار نپخته‌ای انجام ندهیم، بنابراین قضیه فعلاً در حال بررسی است و هنوز عملی نشده است. درس مبانی برای ورودی‌ها جدید ارائه می‌شود. بعضی از دانشجوهای جدید از دبیرستان‌هایی مثل البرز می‌آیند که ۱۰۰٪ زمینه قبلی دارند و برنامهٔ نویسند و بعضیها از شهرستان می‌آیند و اصلاً کلمه‌ای از کامپیوتر به گوششان نخورده است. به همین دلیل ما نمی‌توانیم از همان ترم اول C ارائه کنیم چون در ترم اول بعضی‌ها قویند و بعضی‌ها ضعیف‌ترند، در ترم اول دانشجو باید ابتدا با دانشکده و محیط آشنا شود و تا بتوانیم از ترم بعد برنامه‌نویسی را با او قویتر کار کنیم. بعضی‌ها حتی معتقدند که اصلاً ترم اول دروس کامپیوتر ارائه نکنیم. ضمن اینکه نباید پایه دانشجو در دروس دیگر ضعیف شود. ما می‌خواهیم دانشجویان زبانهای پاسکال ، C و Java را خوب یاد بگیرند. بنابراین از تمام دانشجویان و شورای صنفی تقاضا داریم نظراتشان را در مورد این‌که این دروس چگونه و کی ارائه شوند، بیان کنند. قرار است از آقای شایسته کمک بگیریم و Java را در آزمایشگاه برای دانشجویان ارائه کنیم.

- ***لطفاً در مورد تحصیلات خودتان، اینکه مقطع کارشناسی، کارشناسی ارشد و دکترا را در چه دانشگاه‌ها و با چه تخصص‌هایی گذرانده‌اید کمی توضیح دهید:***

من لیسانسم را از Metro State College و دانشگاه کلرادو دردنور گرفته‌ام. فوق لیسانسم را در دانشگاه ایالتی آریزونا در مبحث MicroElectronic خواندم و دکترایم را از دانشگه ایالتی نیومکزیکی گرفتم و تخصصم نیز در زمینه Computer Testing است و تز من نیز در همین زمینه می‌باشد. ضمن اینکه حدود سه سال هم در دانشگاه ایالتی کلرادو فیزیک جامدات خواندم و کار عملی کردم و سه سال هم در دانشگاه ایالتی آریزونا در زمینه Fabrication و ساخت ترانزیستور کار عملی انجام می‌دادم.

خبر ۱۵

نامه‌ی پویش به کنوت

از آن‌جایی که در این شماره به دکتر کنوت و T_EX پرداخته شد، نسخه‌ای از شماره‌ی ۱۶ پویش به همراه هدیه‌ای از طرف دانشجویان دانشکده‌ی کامپیوتر دانشگاه صنعتی امیرکبیر به آدرس وی پست شد.

امنیت دانشکده بالا رفت

مقر نگهبانی حراست دانشگاه در دانشکده‌ی ما کاملا جا افتاد. این مقر شامل یک میز، یک اجاق زیر میز، یک تلفن، تلویزیون، یک صندلی چرخدار، کمد و یک عدد تخت می‌باشد. همچنین شنیده شده که قرار است مسئولیت نگهبانی (که تاکنون تحت نظر مدیرکل اداری بود) به حراست دانشگاه سپرده شود.

دانشکده در تابستان:

باز هم دانشکده‌های دیگر در تابستان آزمایشگاه ارائه دادند ولی نه دروس آزمایشگاهی دانشجویان دانشکده‌ی مهندسی کامپیوتر . خبرهای مشابه در شماره‌های پیشین نیز آماده بود ولی ظاهرا این فریاد گوش شنوایی ندارد.

مسابقات داخلی ACM:

مسابقات ACM با صورت جدیدی در دانشکده برگزار گردید. در این ساختار نو، دانشجویان در گروه‌های سه نفره با یکدیگر رقابت می‌کنند. امسال گروه‌های ذیل به مرحله‌ی بعد راه یافتند:

تیم اول: وحید غفاریور، حامد عینی و محسن ایزدی

تیم دوم: بابک بهساز، آرش رحیمی و حمیده وثوق‌پور

ما مخالفیم!

نامه‌ی بی‌پاسخ شورای صنفی دانشجویان دانشکده کامپیوتر به سید محمد خاتمی

در نخستین‌ماه های سال ۸۰ زمانی که بحث انتخابات ریاست جمهوری به تدریج داغ می‌شد، خبر تعطیلی تعداد زیادی از کافی‌نت‌های تهران توسط نیروی انتظامی از صفحه اینترنت روزنامه‌ها خود را به تیتر اصلی صفحه اول رساند. این ماجرا فروکش کرده بود که در نیمه‌ی آذر ۸۰، بعد از مدت‌ها حرف و حدیث بالاخره مصوبه‌ی شورای عالی انقلاب فرهنگی درباره‌ی ضوابط شبکه‌هایّ اطلاع‌رسانی رایانه‌ای منتشر گردید. هرچند قانونمند شدن این عرصه چیزی بود که اهالی صنف کامپیوتر مدت‌ها به دنبال آن بودند، ولی این قانون‌گذاری سیلی از انتقادات را با خودش آورد. حادثه‌ی دیگری که در روزهای پایانی سال ۸۰ منتشر شد، از یکطرفه شدن خطوط تلفن بسیاری از سرویس‌دهندگان اینترنتی حکایت می‌کرد. در چنین فضایی، نامه‌ی شورای صنفی دانشجویان دانشکده مهندسی کامپیوتر به رییس جمهور (که همزمان رییس شورای عالی انقلاب فرهنگی هم هست) نشان از نگرانی دانشجویان درباره‌ی تاثیر این مصوبه در آینده‌ی شغلی و صنفی خود داشت. دانشجویان این دانشکده تنها کسانی نبودند که با چشمانی نگران اخبار دولتی شدن اینترنت را دنبال می‌کردند. اواخر فرودین ۸۱ نامه دیگری خطاب به رییس جمهور منتشر شد که نویسنده‌ی آن رییس انجمن صنفی کارفرمایان شبکه‌های اینترنتی بود. این نامه هم دغدغه‌های مشابهی را مطرح می‌کرد. آنچه در پی می‌آید، متن کامل نامه شورای صنفی دانشکده به سید محمد خاتمی است.

جناب آقای خاتمی

رییس محترم جمهور و رییس شورای عالی انقلاب فرهنگی

سلام!

با گذشت بیش از دو ماه از تصویب و ابلاغ قانونی تحت عنوان «مقررات و ضوابط شبکه‌های اطلاع رسانی رایانه‌ای» در شورای عالی انقلاب فرهنگی، ابهامات و انتقاداتی که درباره‌ی این قانون مطرح شده همچنان بی‌پاسخ مانده‌اند. به عنوان یک تشکل صنفی که موضوع این مصوبه در حیطه رشته‌ی تخصصی آن است، انتقادات خود را پیرامون این مصوبه برایتان می‌نویسیم و در انتظار پاسخ شما می‌مانیم.

۱- پیش از هر چیز، همانطور که بارها مطرح شده، تصویب قانون توسط شورای عالی انقلاب فرهنگی برخلاف قانون اساسی است. قانون گذاری در ایران تنها و تنها باید توسط منتخبین مردم در مجلس انجام گیرد و دخالت نهادهای انتصابی مانند شورای عالی انقلاب فرهنگی و مجمع تشخیص مصلحت در این امر، نقض قانون اساسی و تجاوز به حقوق مردم محسوب می‌شود.

قانون گذاری در ایران تنها و تنها باید توسط منتخبین مردم در مجلس انجام گیرد و دخالت نهادهای انتصابی در این امر، نقض قانون اساسی و تجاوز به حقوق مردم محسوب می‌شود.

۲- «هیچ مقامی حق ندارد به نام حفظ استقلال و تمامیت ارضی کشور، آزادی‌های مشروع را هرچند با وضع قوانین و مقررات سلب کند» این جمله، بخشی از اصل نهم قانون اساسی جمهوری اسلامی ایران است. هرچند در مقدمه مصوبه شورای عالی انقلاب فرهنگی بر حق دسترسی آزاد مردم به اطلاعات و دانش تاکید شده، در بندهای متعدد دیگر این مصوبه عملا این امر مختل گشته است. از جمله با تعیین مواردی مبهم و کلی به عنوان تخلف (مانند بندهای ۶-۴ ، ۶-۵ ، ۶-۶ ، ۶-۷ ، ۶-۸ ، ۶-۲۰ در قسمت ب و بندهای ۷-۷، ۷-۱۳ در قسمت ج مصوبه) و اجبار سرویس‌دهندگان به ثبت اطلاعات کامل کاربران (بند ۵-۳-۱۶ در قسمت ب). هرچند ثبت مشخصات کلی کاربران آنچنان که در بند ۵ در قسمت ج ذکر شده (به منظور پاسخگو بودن قانونی و حفظ حقوق افراد در مقابل اقدامات انجام شده‌ی خود) اقدامی مثبت و مورد قبول است، ولی ابهامی که در بند ۵-۳-۱۶ وجود دارد و مشخص نکردن حدود «بانک فعالیت‌های اینترنتی کاربران» با احاله‌ی آن به ضوابط مصوب کمیسیون راهبردی، با توجه به ویژگی‌های فنی قابل تصور برای چنین بانک اطلاعاتی، این نگرانی را بوجود می‌آورد که زمینه‌ی استراق سمع و تجاوز به حریم خصوصی افراد و در نتیجه نقض اصل بیست و پنجم قانون اساسی فراهم گردد.

۳- در نخستین بند آیین نامه‌ی فوق، ایجاد انحصار دولتی در ارائه خدمات اینترنت صراحتا مورد تاکید قرار گرفته است. این انحصار بر خلاف روح قانون اساسی و نص اصول و مقدمه‌ی آن است، که تصریح می‌کند: «قانون اساسی تضمین‌گر نفی هر گونه استبداد فکری و اجتماعی و انحصار اقتصادی می‌باشد و در خط گسستن از سیستم استبدادی و سپردن سرنوشت مردم به دست خودشان تلاش می‌کند» همچنین در بخشی از اصل چهل و سوم قانون اساسی (بند دوم) تبدیل دولت به یک کارفرمای بزرگ مطلق در حیطه‌های اقتصادی نفی شده است.

۴- تعیین دولت به عنوان تنها سرویس‌دهنده‌ی اصلی اینترنت (ASP) جدای از آنکه بر خلاف قانون اساسی است، با توجه به سابقه ناموفق مدیریت دولتی در بخش‌های اصلی اقتصادی، نوعی بازگشت به عقب و باز آزمودن پیش آزموده است. و آیا مگر نه این است که پس از سال‌ها تجربه‌ی اقتصاد متمرکز و دولتی، دولت و مجلس در کار کاستن از تصدی‌گری دولت و حتی یافتن راه‌کارهایی برای دخیل‌کردن بخش خصوصی در بخش‌هایی چون بانکداری، هواپیمایی، کشتیرانی، راه‌آهن، بیمه، پست و غیره می‌باشند؟

۵- آنچه بیش از هرچیز ما را نگران می‌کند، این است که ورود دولت به عنوان ارائه دهنده‌ی انحصاری خدمات اتصال بین المللی، با توجه به سابقه نامطلوب بخش دولتی در این حیطه، آسیب‌های ترمیم‌ناپذیری بر صنعت پویا و در حال رشد انفورماتیک در کشور بزند. شبکه‌ی مخابراتی کشور، چند سال است تلاش می‌کند زیر ساخت مناسبی برای شبکه ملی انتقال داده‌ها فراهم کند و هنوز نیز در این کار آنچنان که شایسته است توفیق حاصل ننموده است. به طوریکه هم‌اینک ارتباط سرویس‌دهنده‌های اینترنت در ایران با یکدیگر نیز از طریق ستون فقرات (Backbone) اروپا یا آمریکا یا آسیای دور انجام می‌گیرد. به عقیده‌ی ما به عنوان تشکلی که اعضایش (دانشجویان رشته کامپیوتر) ارتباط نزدیکی با این بخش دارند، بوروکراسی دولتی کشور ناکارآمدتر از آن است که بتواند خود را با تحولات پرشتاب فن‌آوری ارتباطات همگام سازد و اصرار دولت بر این امر، تنها جلوی رشد بخش خصوصی در این حیطه را خواهد گرفت. می‌توان به شکل مستند نشان داد که شرکت مخابرات ایران در سالیان گذشته حتی در ارائه سرویس مناسب به زیربخش‌های خود و برخی سازمان‌های دولتی که از طریق این شرکت با اینترنت مرتبط بودند ناتوان بوده و استفاده‌کنندگان از این سرویس دچار مشکلات عدیده‌ای بوده‌اند. این در شرایطی است که بخش خصوصی در سالیان گذشته عملکرد موفقی داشته است و روند بهبود کیفیت و کمیت دسترسی به اینترنت در ایران رشدی تصاعدی را نشان می‌دهد.

جناب آقای خاتمی. همانطور که به عنوان رییس قوه مجریه مستحضرید، بحران اشتغال در چشم‌انداز آتی کشور به شکل جدی مطرح است و هم‌اینک نیز با ورود تدریجی فارغ‌التحصیلان دانشگاه‌ها به بازار کار، خود را نشان می‌دهد. این امر یکی از عوامل موثر در پدیده‌ی نامبارک فرار مغزها نیز می‌باشد. در چنین شرایطی، توجه به صنعت انفورماتیک (که در کشور ما صنعتی نوپا محسوب می‌گردد) و فراهم ساختن بسترهای قانونی لازم برای رشد آن (مواردی مانند حمایت از حقوق پدیدآورندگان آثار نرم‌افزاری و پیوستن به قانون بین‌المللی کپی‌رایت) می‌تواند عرصه‌ی بسیار وسیعی را برای جذب دانش‌آموختگان رشته‌های مختلف و گذر از این بحران ایجاد کند. با توجه به مزیت-نسبی ایران در این حیطه رسیدن به موقعیتی مانند کشور هند و کسب درآمدی چند میلیارد دلاری از آن برای کشور، چندان دور از دسترس نیست. این امر به سرمایه‌گذاری چندانی هم نیاز ندارد و تنها انتظاری که از دولت می‌رود، اتخاذ سیاست‌های حمایتی از یک سو و خودداری از دخالت‌های مخربی چون این از سوی دیگر است.

امیدواریم با توجه و پیگیری شما به عنوان رییس شورای عالی انقلاب فرهنگی، نگرانی و ابهامی که درباره آینده صنعت انفورماتیک کشور در این زمینه به وجود آمده است مرتفع گردد.

۱۳۸۱/۱۲/۱۸

شورای صنفی آموزشی دانشجویان

دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

تعیین دولت به عنوان تنها سرویس‌دهنده‌ی اصلی اینترنت جدای از آنکه بر خلاف قانون اساسی است، با توجه به سابقه ناموفق مدیریت دولتی در بخش‌های اصلی اقتصادی، نوعی بازگشت به عقب و باز آزمودن پیش آزموده است.

به عقیده‌ی ما به عنوان تشکلی که اعضایش (دانشجویان رشته کامپیوتر) ارتباط نزدیکی با این بخش دارند، بوروکراسی دولتی کشور ناکارآمدتر از آن است که بتواند خود را با تحولات پرشتاب فن‌آوری ارتباطات همگام سازد

تنها انتظاری که از دولت می‌رود، اتخاذ سیاست‌های حمایتی از یک سو و خودداری از دخالت‌های مخربی چون این؛ از سوی دیگر است.

حمایت از حقوق پدیدآورندگان نرم‌افزارهای رایـــــــــانه‌ای

قانون

بله! ایران هم قانون کپی‌رایت دارد.

مـاده ۱- حق نشر، عرضه، اجرا و حق بهره برداری مادی و معنوی نرم افزار رایانه‌ای متعلق به پدیدآورنده آن است. نحوه تدوین و ارائه داده ها در محیط قـابل پـردازش رایانه ای نیز مشمول احکام نرم افزار خواهد بود. مدت حقوق مـادی سـی (۳۰) سـال از تـاریخ پدیـدآوردن نرم افزار و مدت حقوق معنوی نامحدود است.

ماده ۲- در صورت وجود شرایط مقرر در قانون ثبت علائم و اختراعات، نرم افـزار بـه عـنوان اخـتراع شـناخته می شود، آئین نامه مربوط به این ماده به تصویب هیأت وزیران خواهد رسید.

ماده ۳- نام، عنوان و نشانه ویژه ای که معرف نرم افزار است از حمایت این قـانون برخوردار است و هیچ کس نمی تواند آنها را برای نرم افزار دیگری از همـان نـوع یـا مانـند آن به ترتیبی که القای شبهه کند بکار برد در غیر این صورت به مجازات مقرر در ماده (۱۳) این قانون محکوم خواهد شد.

ماده ۴- حقـوق ناشی از آن بخش از نرم افزاری که به واسطه نرم افزارهای دیگر پدید می آید متعلق به دارنده حقوق نرم افزارهای واسط نیست.

مـاده ۵- پدیـدآوردن نـرم افـزارهای مکمل و سازگار با دیگر نرم افزارها با رعایت حقوق مادی نرم افزارهای اولیه مجاز است.

مـاده ۶- پدیـدآوردن نرم افزارها ممکن است ناشی از استخدام و یا قرارداد باشد در اینصورت:

الـف- بـاید نام پدیدآورنده توسط متقاضی ثبت به مراجع یاد شده در این قانون به منظور صدور گواهی ثبت، اعلام شود.

ب - اگر هدف از استخدام یا انعقاد قرارداد، پدیدآوردن نرم افزار موردنظر بـوده و یا پدیدآوردن آن جزء موضوع قرارداد باشد، حقوق مادی مربوط و حـق تغیـیر و توسـعه نرم افزار متعلق به استخدام کننده یا کارفرما است، مگر اینکه در قرارداد به صورت دیگری پیش بینی شده باشد.

مـاده ۷- تهـیهٔ نسـخه هـای پشـتیبان و همچنین تکثیر نرم افزاری که به طـریق مجـاز برای استفاده شخصی تهیه شده است چنانچه به طور همزمان مورد استفاده قرار نگیرد، بلامانع است.

مـاده ۸- ثبت نرم افزارهای موضوع مواد (۱) و (۲) این قانون پس از صدور تأیـیدیه فنـی توسـط شـورای عالـی انفورماتـیک حسـب مورد توسط وزارت فرهنگ و ارشاد اسلامی و یا مرجع ثبت شرکتها انجام می پذیرد.

مـاده ۹- دعوای نقض حقوق مورد حمایت این قانون، در صورتی در مراجع قضـایی مسـموع اسـت که پیش از اقامه دعوی، تأییدیه فنی یادشده در ماده (۸) ایـن قـانون صـادر شده باشد.در مورد حق اختراع،علاوه بر تاییدیه مزبور، تقاضای ثبت نیز باید به مرجع ذیربط تسلیم شده باشد.

مـاده ۱۰- بـرای صـدور تأیـیدیه فنـی موضـوع ماده (۸) در مورد نرم افزارهایـی که پدیدآورنده آن مدعی اختراع بودن آن است، کمیته ای به نام "کمیته حق اختراع" زیرنظر شورای عالی انفورماتیک تشکیل می‌شود. اعضـای ایـن کمیـته مرکـب از ۳ کارشـناس ارشـد نـرم افزار به عنوان نمایـندگان شورای عالی‌انفورماتیک، نماینده سازمان ثبت اسناد و املاک کشـور و یـک کارشـناس حقوقـی بـه انتخاب شورای عالی انفورماتیک خواهد بود.

مـاده ۱۱- شـورا مکلف است از صدور تأییدیه فنی برای نرم افزارهایی که به تشخیص وزارت فرهنگ و ارشاد اسلامی خلاف اخلاق اسلامی و عفـت عمومـی و سلامت شخصیت کودکان و نوجوانان باشند خودداری کند. وزارت فرهنگ و ارشاد اسلامی باید ظرف دو هفته راجع به استعلام کتبی شورای عالی انفورماتیک اعلام نظر کند.

ماده ۱۲- به منظور حمایت عملی از حقوق یاد شده در این قانون، نظم بخشـی و سـاماندهی فعالیـت های تجاری رایانه ای مجاز، نظام صنفی رایانه‌ای توسط اعضای صنف یاد شده تحت نظارت شورا به وجود خواهد آمـد. مجازات های مربوط به تخلفات صنفی مربوط، برابر مجازات های جـرایم یـاد شـده در لایحه قانونی امور صنفی - مصوب ۱۳۵۹/۴/۱۳ و اصلاحیه های آن - خواهد بود.

ماده ۱۳- هـر کس حقوق مورد حمایت این قانون را نقض نماید علاوه بـر جبران خسارت به حبس از نود و یک روز تا شش ماه و جزای نقدی از ده میلیون تا پنجاه میلیون ریال محکوم می گردد.

تبصره - خسارات شاکی خصوصی از اموال شخص مرتکب جرم جبران می‌شود.

ماده ۱۴- شاکی خصوصی می‌توان تقاضا کند مفاد حکم دادگاه در یکی از روزنامه ها با انتخاب و هزینه او آگهی شود.

مـاده ۱۵- رسیدگی جـرم مذکـور در مـاده (۱۳) بـا شـکایـت شاکی خصوصی آغاز و با گذشت او موقوف می شود.

ماده ۱۶- حقوق مذکور در ماده (۱) در صورتی مورد حمایت این قانون خواهـد بـود کـه موضـوع برای نخستین بار در ایران تولید و توزیع شده باشد.

ماده ۱۷- آییـن نامه اجرایی این قانون شامل مواردی از قبیل چگونگی صـدور گواهـی ثبـت و تأییدیه فنی و هزینه های مربوط همچنین نحوه تشـکیل نظـام صـنفی رایانـه ای، بـه پیشنهاد سازمان مدیریت و برنامه ریـزی کشـور و بـا هماهنگی وزارتخانه های فرهنگ و ارشاد اسلامی و دادگستری به تصویب هیأت وزیران خواهد رسید.

ـ قانون فوق مشتمل بر هفده ماده و یک تبصره در جلسه علنی مورخ ۷۹/۱۰/۴ مجلس شورای اسلامی تصویب و در تاریخ ۱۳۷۹/۱۰/۱۰ به تأیید شورای نگهبان رسیده است

گپ علمی

طراحی الگوریتم

مهندس پوروطن حدود دو سال است که دروس پایه‌ی مهندسی کامپیوتر را ارائه می‌کند. وی جزو کمیته‌ی علمی مسابقات ACM در غرب آسیا می‌باشد. با توجه به این که دیدگاه جدیدی را در ارتباط با فرایند حل مساله ارائه کرده است، بخش صنفی پویش این بار به خدمت ایشان رسید.

حسن تکابی
هانی جوان‌همت
رضا حسامی‌فرد

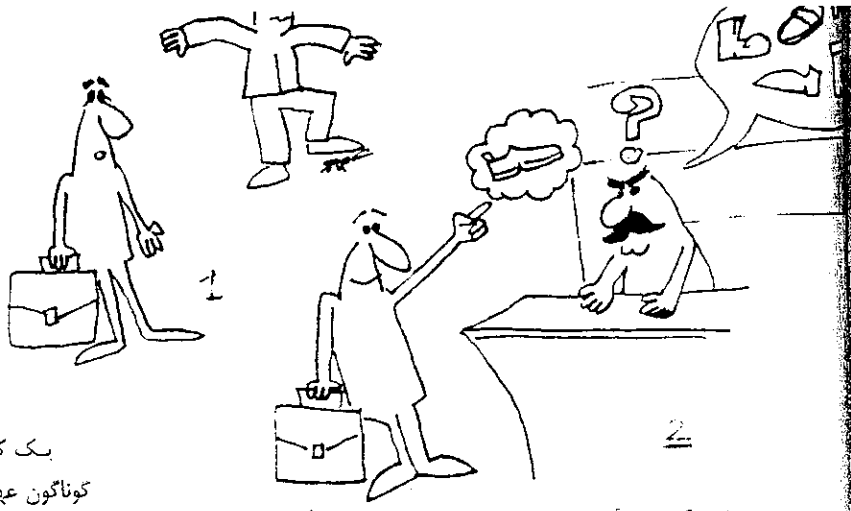
- ***دو درس ساختمان گسسته و ساختمان داده‌ها دو پیش نیاز درس طراحی الگوریتم است. شما توقع دارید دانشجو در این دو درس چه آموخته باشد حالا که سر کلاس طراحی الگوریتم نشسته است؟***

هنگامی که برنامه‌نویسی مبتنی بر Procedure Oriented است، ما دو عنصر اساسی در مسألهٔ برنامه‌سازی داریم: ۱) مشاهدهٔ رفتار اجرایی و ۲) داده‌های ما در برنامه که معمولاً رفتار اجرایی بر روی داده‌ها کار می‌کند. این دو مستقل از هم نیستند؛ یعنی می‌توان با نگرشی که به داده‌ها داریم، روی رفتارها اثر مستقیم بگذاریم و می‌توانیم با توجه به رفتارهایی که داریم برای داده‌ها چیدمان‌های مختلف داشته باشیم. آنچه در درس ساختمان‌داده‌ها مطرح می‌گردد این است که ما اطلاعات را چگونه بچینیم و چه ترتیبی مؤثرتر است و هر چیدمان چه معایبی دارد و چه محاسنی و کلاً خروجی چگونه کار می‌کند و روابط داده‌ها چگونه است. درست در کنار این یک سری الگوریتم‌ها را هم می‌خوانیم، چرا؟ چون می‌خواهیم بدانیم مثلاً اگر یک ماتریس پراکنده داشته باشیم چگونه Transpose کنیم و چه Transpose مناسب‌تر است و کمترین O را برای ما خواهد داشت. ولی به هرحال تمام تمرکز روی چیدمان اطلاعات و روابط آن‌ها است و کلاً به حل مسائل از این دیدگاه می‌نگریم که چگونه پراکنده کنیم، چگونه بچینیم، با چه ساختمانی داده‌ای O کمتری خواهیم داشت، انعطاف‌پذیری را چگونه تأمین کنیم که با این ساختاربندی اطلاعات، بهترین سرویس را در روند اجرایی برنامه داشته باشیم و مسائل مختلف دیگر. تکیه و توجه به این‌ها است.

اما وقتی می‌خواهیم فرآیند اجرایی را بررسی کنیم دیگر مستقل از ساختمان‌داده کار می‌کنیم و به آن توجه زیادی نداریم و دقت و توجه اصلی به فرآیند است. فرآیند را انتخاب می‌کنیم و بعد تصمیم می‌گیریم چه ساختمان‌داده‌ای می‌تواند مناسب باشد. در اینجا هم باز ما با O کار داریم. اما توجه ما به چگونه حل کردن مسائل است. به بیان بهتر، سعی در دسته‌بندی فرآیندها داریم و می‌خواهیم آن‌ها را هم‌خانواده کنیم. همان طور که در درس ساختمان‌داده‌ها یک ویژگی به

در مبانی کامپیوتر دانشجو باید بتواند Machine State Programming را فرا بگیرد و در برنامه‌سازی پیشرفته، Real World Programming را.

نام صف را مستقل از پیاده‌سازی در نظر می‌گیریم در اینجا هم آیا می‌توانیم گروهی از مسائل را طوری دسته‌بندی کنیم که رفتار آن گروه یکسان



باشد؟ و ما با آنها برخورد یکسان داشته باشیم؟ عملاً یاد می‌گیریم چگونه مسائل را حل کنیم.

وقتی ما روش‌های مختلف را یاد می‌گیریم و استفاده هر فرآیند را آموختیم، آنگاه ما به این مساله می‌رسیم که چگونه مسائل حل می‌شوند. مثلاً وقتی با یک مسأله در pop برخورد می‌کنیم، سعی می‌کنیم آن را به یک خانواده نسبت دهیم و سپس آن را آنالیز کنیم و بعد آن را مدل می‌کنیم و در نهایت حلش می‌کنیم. در اینجا ما باز هم درگیر Order هستیم. بنابراین وقتی من سر کلاس طراحی الگوریتم می‌آیم، دوست دارم هر ساختار داده‌ای که نام می‌برم، دانشجو لااقل به گوشش خورده باشد. نمی‌خواهم به آن مسلط باشد اما می‌خواهم با آن آشنا باشد و طرح آن را بشناسد. البته درست است که ما ساختارهای فراوانی داریم، اما ساختارهای متعارف را باید بشناسد. آنقدر آشنا باشد که با یک توصیف کوتاه به خاطر بیاورد و بداند چه کاربردی داشته است. مثلاً بدانند که درخت یک ساختمان داده است که این طور درج می‌شود، این طور حذف می‌شود و فلان کاربرد را دارد که متأسفانه اکثر دانشجویان در این مورد مشکل دارند.

یک سری روابط در ساختمان‌داده وجود دارد که ریاضیات آن‌ها را مورد حمایت قرار می‌دهد. این گونه روابط را در ساختمان گسسته مطالعه کرده‌ایم. ببینید ممکن است این مطالب گاهی مهم هم نباشد اما باید به گوش دانشجو آشنا باشد. این عدم آشنایی این قدر مؤثر است که گاهی یک لغت کلیدی هم باعث گیج شدن دانشجو می‌گردد. مثلاً مفهوم Latice diagram که در کلاس گفته می‌شود؛ لغت Latice یک شک به کلاس وارد می‌شود. هر چند پشت این کلمه محتوای خاصی وجود ندارد اما همان شک باعث می‌شود که بازده مفید بچه‌ها سر کلاس کم شود و کمتر بتوانند حضور ذهن داشته باشند. خصوصا چون ذهن‌ها اکثراً تنبل است، اگر یک مطلب با سرعت بیشتری گفته شود، دانشجو قدرت تحلیل در همان زمان را ندارد. حالا هر چقدر لغت‌های جدید زیادتر باشد فراگیری را سخت‌تر می‌کند. من می‌خواهم دانشجو در درس ساختمان‌داده و ساختمان‌گسسته این مفاهیم را یاد گرفته باشد و دانشجو با کلمات مأنوس شده باشد و یا با یک توضیح به آن برسد. این از اهداف دروس فوق است. این سه درس سعی می‌کنند زنجیره‌وار مهندسین خوبی را برای حل مسائل و بهینه‌سازی حل مسائل پرورش دهند.

این سه درس وابستگی زیادی به هم دارند. گاهی مساله ما را وادار می‌کند ساختمان داده را تأمین کنیم و گاهی ما را وادار می‌کند از طریق فرآیند، ساختمان داده را انتخاب کنیم و آزادی عمل دست ماست.

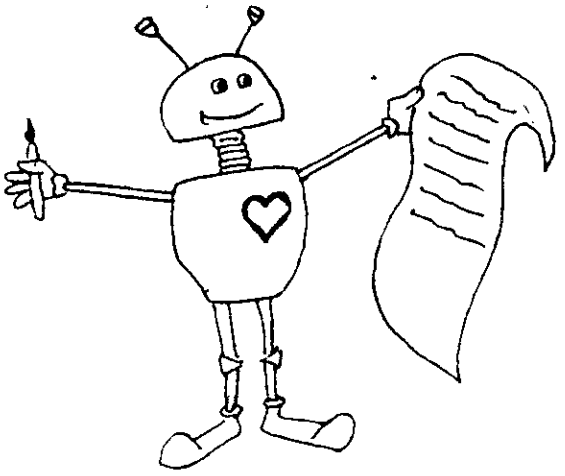
- ***سؤالی که در بین اکثر دانشجویان مطرح است این است که فرضاً من ۱۰ نوع مرتب‌سازی را یاد گرفتم و یاد گرفتم که هر کدام در کجا به کار می‌رود اما در بازار کار چه؟ و یا در دروس پایین‌تر مثلاً ساختمان گسسته، اصل لانه کبوتر را در نظر بگیرید که برای آن مسائلی مطرح می‌گردد که عجیب و بدون کاربرد به نظر می‌آید؟ با توجه به هدف که حل مسأله است: پس این مسأله کجاست که یک مهندس کامپیوتر باید آن را حل کند؟***

یک کارشناس کامپیوتر می‌تواند از مشاغل گوناگونی در طیف‌های گوناگون عهده‌دار شود. شاید ما در یک بازار سراسری و عمومی بگوییم اگر یک نوع مرتب‌سازی بدانیم کافی است و می‌توان با آن کارهای لازم را انجام داد ولی آیا ما واقعاً برای یک بازار خاص که محدود به یک مقطع زمانی خاص است کار می‌کنیم و کارشناس تربیت می‌کنیم؟ وقتی ما قصد داریم قدرت تحلیل و آنالیز را برای مسائل مختلف بالا ببریم، چنین مسایلی نمونه‌های مختلفی هستند برای بالا بردن قدرت تحلیل شما. مثلا در یک پدیده مانند Shell Sort که می‌تواند برای یک Sorter که بر روی آن مثلاً ۱۷ تا Transfilter نصب شده است، به خوبی عمل کند و یا در مورد Batchler Sort هم همین طور و خواهیم دید که O خیلی خوبی داریم که خیلی عالی کار می‌کند: چگونه می‌گویید که ما چنین کاربردهایی نداریم؟ برای اکثر کاربردهای آماری که در کشور انجام می‌شوند، به هیچ عنوان نمی‌توانیم از الگوریتم‌های معمولی استفاده کنیم و شاید شما مجبور شوید که از یک نوع مرتب‌سازی استفاده کنید که فقط در همان یک مورد کاربرد دارد. این یک مسأله است که در تصمیم‌گیری در کل کشور مورد استفاده قرار می‌گیرد. حال آیا یک مهندس نباید به این موارد آشنایی داشته باشد؟ تمام مسائل مورد استفاده‌ای خواهند داشت. اگر استفاده‌ای هم نداشته باشد، در بالا بردن قدرت درک شما مفید است. بنابراین هیچ چیز بدون استفاده نداریم. برای خودم که از سال ۶۲-۶۳ کار برنامه‌نویسی کامپیوتر می‌کنم، هیچ دانشی نبوده که برایم بدون استفاده بوده باشد. ببینید شما اشعاری که در دبستانی حفظ کرده‌اید به یاد دارید؟ نه! پس اگر قرار بود فراموش کنید چرا حفظ کردید؟ چون در آن موقع و در آن مقطع با آن ویژگی که داشته‌اید حفظ کردن اشعار کمک می‌کرد که ذهن شما تقویت شود؛ الان هم بسیاری از مسایل همین طور است. مثلاً در درس مبانی، من تمرینهای سنگینی به دانشجویان داده‌ام. منتها شما در دبستان حفظ می‌کنید اما در اینجا شیوه تقویت به گونه‌ای دیگر است. در دنیای اینترنت شما دیگر چیزی را لازم ندارید که حفظ کنید؛ دانستن اطلاعات کلی درباره یک مطلب کافی است. بیشتر اگر بخواهید می‌توانید به راحتی در اختیار بگیرید. بنابراین تمام توجه ما بر روی این است که شما چه چیز را بخواهید و دنبال چه چیزی بگردید و چگونه بگیرید؛ و آن چیز دارای چه ویژگیهایی است. مثلاً در همین پدیدهٔ انفورماتیک: فردی وسیله کامپیوتری را برای منظور خاصی سفارش داده بود زیرا معتقد بود که در کانادا از این وسیله برای این منظور خاص استفاده می‌شده است. اما چنین چیزی مانند این است که کسی بگوید دانشکده پر از سوسک است پس باید برویم و کفش بخریم چون من خودم دیده‌ام در کانادا فلان پروفسور برای از بین بردن سوسک از کفش استفاده می‌کرد. حالا وقتی می‌رفتند سراغ خریدن کفش، فروشنده می‌پرسید مردانه باشد یا زنانه؟ شماره‌اش چند باشد؟ جنس آن؟ می‌بینید پارامترهای فراوانی برای انتخاب مناسب وجود دارد و شما نمی‌توانید بگویید من فقط می‌خواهم سوسک بکشم.

مباحث تئوریک و قدرت تحلیل یک چنین معایبی را در ما برطرف می‌کند بله Sort همان Sort است اما آیا واقعاً یک نوع مرتب‌سازی برای ما کفایت می‌کند؟ آیا یک quice sort دانستن، تمام مسائل را حل می‌کند یا اینکه هر مسأله‌ای با توجه به محیط عملیاتی خود، امکانات خاص خود، محیط کار و Platform هایی که دارند و ... راه حل مخصوص خود را می‌طلبد؟ بعلاوه نوع داده و توقعات ما از آنها همه و همه تعیین کننده هستند. مثلاً ۱۰۰ هزار مراجعه کننده هستند و قصد داریم به آنها سرویس دهیم. آیا باید صبر کنیم تا همه را Sort کنیم، تا کوچکترین نفر را پیدا کنیم و سپس سرویس دهیم؟ هر روش ویژگی خاص خودش را دارد و ما باید با همه آشنا شویم چرا که هر کدام جایگاه خاص خودش را دارد. بنابراین شما نباید به فکر یک سری برنامه‌های کلی باشید که مثلاً مد Sort هم در آنها نقش چندانی ندارد. حتی اگر بخشی از مطالب فعلاً به درد ما نمی‌خورد تقویت ذهنی است و به این ترتیب وقتی ما دریافتیم که در یک sort چه قوانینی مطرح می‌شود دیگر از کنار هیچ مسأله‌ای به راحتی عبور نمی‌کنیم. و هر مسأله مانند یک تابلو هنری است که هر چه بیشتر روی آن وقت بگذاریم زیباتر می‌شود و با ارزش‌تر می‌گردد.

- ***بینشی که هم اکنون برجو دانشکده حاکم است معتقد است که دروس ابتدائی دورهٔ لیسانس را هر کسی می‌تواند تدریس کند و اصطلاحاً این دروس اصلاً مهم نیستند؟ نظر شما چیست؟***

یکی از کارهایی که مهندس کامپیوتر انجام می‌دهد تولید برنامه است و برای برنامه‌سازی هم مدلسازی لازم است. ببینیم جایگاه اینها کجاست. برنامه‌نویسی یک ساختار ذهنی می‌خواهد که شما با آن به مثابه‌ی یک جهان‌بینی، به پدیده‌های اطراف خود نگاه می‌کنید و از هر دیدگاهی، مدل متفاوتی می‌سازید این دیدگاه برای ما لازم است تا بتوانیم به دو مساله بپردازیم. اول حالات مساله که اصطلاحاً (Machine State Programming) گفته می‌شود و دوم مسائل جهان واقعی (Real word programming) طراحان سیلابس وزارت علوم معتقد هستند که ما Machine State Programming را در مبانی کامپیوتر آموزش می‌بینیم و Real word programming را در برنامه‌سازی پیشرفته. در درس مبانی آنچه اهمیت دارد این است که دانشجو بتواند خود را جای ماشین بگذارد و از دیدگاه ماشین به مسائل بنگرد و ما در این درس روش‌های گوناگونی، مثلاً سری‌ها، شمارنده‌ها و ... را آموزش می‌دهیم و با توجه به این‌که برخی ذهن‌ها تنبل است و از این کار هیچ ایده‌ی کلی ندارد، تدریس آن بسیار دشوار می‌باشد. مثلاً من امروز دانشجویی به نام خانم .... داشتم که در مورد پروژه‌ای به مشکل برخورد کرده بود که در درس، جزو اولین تمرینات است. امروز سرانجام موفق به حل مساله شده بود و بسیار خوشحال هم بود که آن را حل کرده بود. بیشتر این افراد قبلاً هم می‌توانستند این کار را بکند ولی ما مجبوریم آنها را وادار به فکر کردن بکنیم. در این مقطع به نظر من باید بهترین افراد و کسانی که خود برنامه‌نویس هستند و به مطلب تسلط کامل دارند آموزش دهند. بزرگترین خطا در آموزش این است که استاد یک برگه در دست



داشته باشد و دیکته کند و همه آن را بنویسند زیرا آن هنگام شما نخواهید توانست فرآیند درست تولید کردن را منتقل کنید. باید مساله را همان جا ایجاد کرد و همان جا راه حل بدست آورد و حل کرد. حل مهم نیست بلکه فرآیند حل مساله مهم است. ما معمولاً دقت خود را بر روی Syntax متمرکز می‌کنیم در صورتی که اصل، عمل فرآیند تولید کد می‌باشد. متأسفانه در برخی جاها گمان می‌کنند که مثلاً من فوق‌لیسانس را می‌توانند بر سر کلاس بگذارند، در صورتی که این کار بسیار مهم را فقط باید واردترین افراد انجام دهند بنابراین باید آموزش دید که اگر مسائل راه حل داشت چه کنیم و اگر نداشت چه؟ بله افرادی مانند من می‌توانند به عنوان حل تمرین کار کنند اما کار اصلی را باید خود اساتید انجام دهند. هم اکنون سر کلاس‌ها سعی می‌شود بر ذهن دانشجویان کار شود و دانشجو وقتی این را درک می‌کند که خودش شروع به حل مساله می‌کند و در می‌یابد که بله من نسبت به گذشته فرق کرده‌ام.

ارزیابی این دروس هم مشکل است بنابراین من سعی می‌کنم کمترین ارزش را برای امتحان در نظر بگیرم و سعی می‌کنم بیشتر از پروژه‌ها برای ارزیابی کمک بگیرم. مگر ما می‌توانیم بگوییم همه شما موظفید فلان روز و فلان ساعت آمادگی ذهنی کامل در سر جلسه داشته باشید؟

در درس برنامه‌نویسی پیشرفته هم همین است فقط دیدگاه عوض شده است و ما دنبال OOP می‌رویم. ولی از عنوان آموزش برنامه‌نویسی شی‌گرا به روش C++، ما فقط C را می‌گیریم و تنها به Syntax می‌پردازیم. بنابراین ابزار را می‌شناسیم اما قادر به برنامه‌نویسی نیستیم. بنابراین آنجا هم باید از مدرسان خوبی استفاده شود که بداند چه درس می‌دهند و چگونه تدریس کنند و چگونه تفکر OOP را به آنها آموزش دهند و چگونه نرم‌افزاری را قطعه قطعه تولید کنند؟ گاهی می‌بینیم شخصی رشتهٔ برق خوانده و دستورات C را هم می‌شناسد و حال برنامه‌سازی پیشرفته درس می‌دهد. این شخص در همان جا دانشجو را ترور می‌کند. اثر این گونه تدریس دیده نمی‌شود ولی ضربه‌ای به دانشجو می‌زند که دیگر قابل جبران نخواهد بود. از آن‌جا که در مراحل بعد هم به برنامه‌نویسی پرداخته نمی‌شود، دانشجو لطمهٔ جبران ناپذیری خورده است. در دروس بالاترچون کار پایه‌ای انجام شده است و دانشجویان تسلط بیشتری پیدا کرده‌اند، اگر من سر کلاس اشتباهی بکنم اشتباه من راحت بر طرف می‌شود، اما در دروس پایه وقتی دانش آموز می‌آید و دانشجو می‌شود کوچکترین اشتباه از سوی من لطمه‌هایی می‌زند که نمی‌توانیم آن را جبران کنیم.

بله بهتر است در تمام دروس ما متخصص داشته باشیم اما وقتی بضاعت ما این است، ترجیح می‌دادم از اساتید بهتر در دروس پایه استفاده کنیم.

مساله‌ی پستچی چینی

مساله :

یک پستچی در راستای کارش، نامه ها را از پستخانه تحویل می‌گیرد و آن‌ها را به مقاصد خود می‌رساند و سپس به پستخانه بر می‌گردد. او مایل است مسیرش را به طریقی انتخاب کند که حداقل مسافت را بپیماید. مسیر مناسبی برای وی تعیین کنید.

این مساله به مساله پستچی چینی معروف است . زیرا اولین بار کوان [1] ریاضی دان چینی آن را بررسی کرده است.

در حل این مساله بررسی نوع خاصی از گراف‌ها لازم به نظر می‌رسد، بنابراین به چند تعریف ساده در زیر می‌پردازیم.

گراف G گراف اویلری نامیده می‌شود ، اگر بتوان با آغاز از یک راس آن و پیمایش بدون تکرار همه یالها به همان راس ابتدا رسید . به عبارت دقیقتر:

یک سیر G گشت بسته ای است که از هر یال G حداقل یک بار عبور کند و یک گراف اویلری است اگر شامل حداقل یک سیر اویلری باشد.

قضیه (۱): گراف G اویلری است اگر و فقط اگر درجه هر راس آن زوج باشد.

اثبات:

به عهده خواننده (راهنمایی به [1] ، فصل چهارم مراجعه کنید)

ناحیه یا شهر مربوط به پستچی را در نظر می‌گیریم ، اگر خیابان‌های این شهر را به عنوان یال‌های گراف در نظر بگیریم و رئوس این گراف را محل تقاطع خیابان‌ها (چهارراه‌ها و میدان‌ها و...) ،نقشه شهر به صورت گراف مفروضی در آمده است .

ابتدا فرض می‌کنیم که نقشه شهر به صورت گراف اویلری است پس یک مسیر اویلری در آن پیدا می‌کنیم . بعد حل مساله را کامل کرده و برای تمام گراف‌ها حل می‌کنیم .

برای یافتن مسیر اویلری الگوریتم زیر را که منسوب به فلوری [2] است را بیان کرده و اثبات می‌کنیم :

شروع :

۱) راس دلخواه v_0 را انتخاب کرده و قرار می‌دهیم $w_0 = v_0$

۲) فرض کنید $w_i = v_0 e_1 v_1 \dots e_i v_i$ مسیر انتخاب شده تا مرحله i ام است که در آن v_i ها رئوس و e_i ها یال‌های انتخاب شده هستند یال e_{i+1} را از مجموعه یالهای انتخاب نشده طوری انتخاب می‌کنیم که:

I) e_{i+1} از v_i بگذرد

II) تا حد ممکن e_i یک یال برشی برای G_i که به زیر تعریف می‌شود نباشد.

$$G_i = G - \{e_1, e_2, \dots, e_i\}$$

(یال برشی یالی است که درصورتی که از گراف حذف شود بین دو راس اطراف یال مذکور مسیر دیگری موجود نباشد

۳) اگرمرحله ۲ قابل اجرا نیست توقف کن.

قضیه(۲): اگر Gاویلری باشد ، آنگاه هر گذر در G که به وسیله الگوریتم فلوری ساخته شود یک مسیر اویلری در G است .

اثبات :

فرض کنید G اویلری باشد و گیریم $w_n = v_0 e_1 v_1 \dots e_n v_n$ گذری در G باشد که توسط الگوریتم فلوری ساخته شده است .به وضوح v_n از درجه صفر است. زیرا اگر درجه آن بیشتر از صفر باشد هنوز الگوریتم متوقف نشده است. از آنجا که در طول پیمایش ، هر کجا به راسی وارد شده ایم و پس از آن خارج گردیده ایم از درجه رئوس آن ۲ واحد کم شده است. (به جزراس ابتدا که فقط از آن خارج شده ایم و راس انتها که فقط به آن وارد شده ایم .) با توجه به اینکه درجه راس v_n زوج مانده است بنابر این $v_0 = v_n$ است .یعنی w_n گذری بسته است.

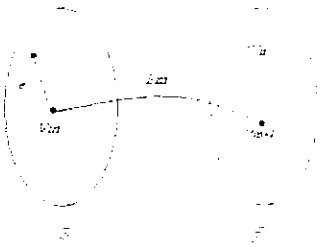
فرض کنید – یک سیر اویلری در G نباشد ، وگیریم S مجموعه راسهای با درجه مثبت در G_n باشد، دراین صورت S ناتهی است و

---

[1] KUAN

[2] Fleury

$v_n \in \overline{S}$. حال فرض کنید m بزرگترین عدد صحیحی باشد که $v_m \in S$ و $v_{m+1} \in \overline{S}$ چون w_n در $\overline{S}$ پایان می‌یابد و e_{m+1} تنها یال بین S و $\overline{S}$ در G_n است، بنابراین یال برشی G_m است.



(شکل ۱)

فرض کنید e یال دیگری از G_m باشد که از v_m می‌گذرد نتیجه می‌شود (از مرحله ۲) که e باید یک یال برشی G_m هم باشد از این رو یالی از برش $G_m[S]$ است. اما چون $G_m[S] = G_n[S]$ (چرا ؟) هر راس در $G_m[S]$ از درجه راس زوج است وبا توجه به اینکه گراف (اگر گراف مفروض H دارای کلیه رئوس درجه زوج باشد یال برشی نداشته باشد) ایجاب می‌کندکه دارای یال برشی نباشد و این یک تناقض است.پس S تهی است یعنی w_n اویلری است. بنابراین اگر گراف یک شهر بصورت گراف اویلری باشد آنگاه مطمئنا مساله ما حل شده بود. اما این یک فرض خوشبینانه است.در نقشه یک شهر که به صورت گراف تصویر شده باشد تعدادی از رئوس با درجه فرد موجود است (این تعداد زوج است چرا ؟)مسلما پستچی باید تعدادی از مسیرها را را چند بار تکرار کند.

در این جا لازم است که دوگانه سازی یک یال را تعریف کنیم:یال e در گراف G دوگانه است اگر دو انتهایش توسط یک یال با وزن W(E) به هم وصل شود. در گراف نااویلری شهر باید یک سری از یالها دوگانه شوند تا درجه رئوس فرد این شهر زوج شود .ملاحظه کنید که دلیل این امر دوشرطی بودن قضیه (۱) است .

تعریف: به مجموعه یالهایی که اگر به G اضفه شود ، آن را تبدیل به یک گراف اویلری می‌کند بخش ترمیمی‌می‌گویند.

$$\pi = \{e_1, e_2, \ldots, e_m\}$$

تعریف می‌کنیم

$$d(\pi) = \sum_{i=1}^{m} w(e_i)$$

بنابراین هدف ما به پیدا کردن یک بخش ترمیمیا حداقل وزن تبدیل می‌شود.

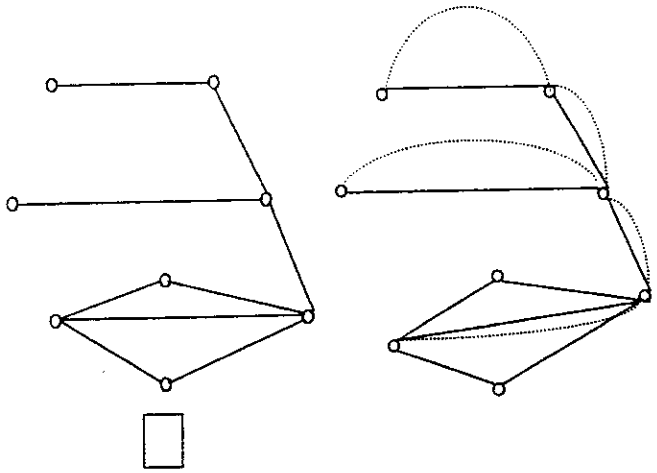
تعریف کگراف مسیر G گرافی است که در آن بین u و Y راسی موجود است ، اگر و فقط اگر مسیری بین آنها هوجود باشدوزن این یال را طول کوتاهترین مسیر در نظر می‌گیریم.

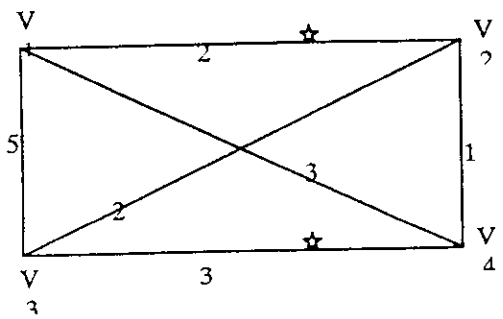
گراف شهر را در نظر می‌گیریم و گراف مسیر آن را می‌سازیم . این گراف گراف کامل است زیرا گراف شهر، همبند می‌باشد .کوتاهترین مسیر بین هر دو راس را می‌توان از الگوریتم دیجکسترا [۳] بدست آورد.

بخش ترمیمی‌گراف G در حقیقت تعدادی مسیر است که هر کدام دو راس درجه فرد را به هم وصل میکنند که در حقیقت در گراف مسیر G برای رئوس فرد (گراف مسیر القائی توسط رئوس فرد) تعدادی یال است که هر کدام دو راس را به هم وصل می‌کنند که در حقیقت نماینده مسیرها درG هستند. در نظر داشته باشید که هر راس باید انتها یا ابتدای فقط یک یال در این مسیر ترمیمی‌باشد. (چرا؟)

تعریف : جورسازی مجموعه‌ای از یال‌ها است، که هر راس G به حداکثر یک یال از این مجموعه متصل باشد ؛ و جورسازی تام یک جورسازی از G است که همه رئوس را بپوشاند یعنی هر راس دارای یک یال در این مجموعه باشد.

بنابراین هدف ما به پیدا کردن یک جورسازی تام با حداقل وزن در گراف القائی G توسط رئوس درجه فرد تبدیل شده است .





شکل(۲)

یال‌هایی که با خطچین مشخص شده بخش ترمیمی‌است که گراف را به یک گراف اویلری تبدیل کرده است و حال با الگوریتم فلوری می‌توان مسیر اویلری را مشخص کرد.

تمرین: ثابت کنید الگوریتم فلوری دارای پیچیدگی زمانی غیر نمایی است.

---

[۳] Dijkestra

زندگی نامه دکتر دونالد کنوث

رضا حسامی فرد

دانشگاه صنعتی امیر کبیر
دانشکدهٔ مهندسی کامپیوتر

شاید به جرئت بتوان دکتر کنوث را سلطان برنامه نویسی کامپیوتر نامید در زیر سعی کرده‌ایم مختصری از شرح حال او را بیان کنیم . ضمن اینکه اشاره می شود این مقاله توسط نسخه فارسی TEX (فارسی‌تک)حروف چینی شده است.

دونالد کنوث در دهم ژانویه سال ۱۹۳۸ در ویسکانسین—میلواکی متولد شده است . او دوره‌ی لیسانس خود را در رشته‌ی ریاضی انستیتو تکنولوژی کیس (CASE) تحصیل کرد. در همان هنگام وی مشغول نوشتن نرم‌افزارهایی در مرکز محاسبات آنجا بود. در سال ۱۹۶۰ دانشکده‌ی علوم دانشگاه CASE کرسی استادی را با مدرک کارشناسی (لیسانس) به وی اعطا کرد. بعد از دوره‌ی کارشناسی ارشد در انستیتو تکنولوژی کالیفرنیا، مدرک دکترای خود را در رشته‌ی ریاضیات در سال ۱۹۶۳ دریافت کرد و سپس در دانشکده‌ی ریاضیات مشغول به کار شد. در تمام این مدت وی به کار توسعه‌ی نرم‌افزار می‌پرداخت و مشاور شرکت Burrouqhs (1960-1968) و همچنین به عنوان ویراستار زبان‌های برنامه‌نویسی برای انتشارات ACM خدمت می‌کرد (۱۹۶۷—۱۹۶۴). او در سال ۱۹۶۸ به عنوان استاد علوم کامپیوتر به دانشگاه استانفورد پیوست و پس از ۹ سال به عنوان شخص اول علوم کامپیوتر دانشگاه استانفورد شناخته شد. او به عنوان استاد دانشگاه دروس متنوعی از دوره‌های تحصیلی را ارایه می کرد. که می توان از این میان به ساختمان داده ها و ریاضیات گسسته اشاره کرد. در سال ۱۹۹۳ مدرک افتخاری استادی هنر برنامه‌نویسی را دریافت کرد. او تاکنون به رساله‌ی دکترای ۲۸ دانشجو نظارت کرده است. کنوث در سال ۱۹۶۲ برای فراهم کردن کتابهای مرجعی راجع به فنون برنامه‌نویسی شروع به فعالیت کرد. این کار وی به صورت یک دوره‌ی هفت جلدی به نام «هنر برنامه‌نویسی کامپیوتر»[1] ظهور یافت. مجلدهای ۱ و ۲ و ۳ در سال‌های ۱۹۶۸ و ۱۹۶۹ و ۱۹۷۳ برای اولین بار به چاپ رسید. همچنین او در سال ۱۹۹۷ این سه جلد را با اصلاحات فراوان تجدید چاپ کرد. او هم اکنون به طور تمام وقت بر روی جلدهای باقی‌مانده مشغول به کار است. تقریبا یک میلیون کپی از این مجلدات به فروش رفته است و به ٦ زبان زنده‌ی دنیا ترجمه گردیده است. او کار این پروژه‌ی بزرگ را به مدت ۱۰ سال متوقف کرد.که در این مدت او به کار گسترش نرم افزار TEX (بخوانید تک) برای حروف چینی متون علمی و غیره و همچنین طراحی سیستم متافونت [2] برای طراحی الفبا پرداخت شایان ذکر است که WEB و CWEB که زبان‌هایی برای تولید متون

[1] The art of computer programming
[2] metafont

ساختاریافته (از نظر حروف‌چینی) هستند، محصولات فرعی این تحقیقات هستند. TEX هم اکنون برای حروف‌چینی اکثر مقالات و متون علمی به خصوص در ریاضیات و فیزیک به کار می‌رود. مقالات تحقیقاتی وی در پایه‌گذاری چندین بخش از علوم کامپیوتر و مهندسی نرم‌افزار تاثیرگذار بوده است؛ از جمله:

-Knuth-Bendix algorithms for axiomatic reasoning

-Emperical studies of user programs and profiles

-Emperical studies of user programs and profiles

-Analysis of algorithms

—خواص گرامر

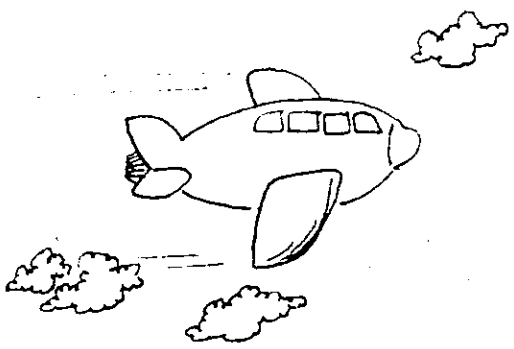
—

Parsing LR(k)

به طور کلی تحقیقات وی به سوی ایجاد یک تعادل میان مباحث تئوری و عملی هدایت شده است. دکتر کنوث در سال ۱۹۷۴ جایزه‌ی ACM Turing را دریافت کرد و به سال ۱۹۸۰ هم به عضویت انجمن کامپیوتر انگلستان درآمد. همچنین در سال ۱۹۸۲ عضو افتخاری IEEE گردید. همچنین او در آکادمی هنر و علوم آمریکا و آکادمی ملی علوم و آکادامی ملی مهندسی و همچنین در Foreign Association of Facademei des sciences در پاریس و همچنین در Det Norske Vedenrkaps Akadem در Oslov عضویت دارد.

کنوث پنج اختراع ثبت شده دارد؛ تقریبا ۱۶۰ مقاله و ۱۹ کتاب نوشته است . او در سال ۱۹۷۹ از رییس جمهور وقت آمریکا— کارتر— مدال علوم را دریافت کرد. همچنین وی نشان نقره‌ای انجمن ریاضی آمریکا (۱۹۸۶) و جایزه‌ی آکادمی علوم نیویورک (۱۹۸۷) و جایزه‌ی ح.د.وارینز را به خاطر ابزارشناسی نرم‌افزار (۱۹۸۹) و مدال Adelskold از آکادمی علوم سویدیش (۱۹۹۴) و همچنین جایزه‌ی هاروی از تکنیون (۱۹۹۵) و جایزه‌ی کیوتو به خاطر تکنولوژی پیوسته را در سال ۱۹۹۶ دریافت کرده است او علاوه بر اینکه نامزد دریافت جایزه پیشگامان کامپیوتر IEEE در سال ۱۹۸۲ بود جایزه انجمن کامپیوتر مک دونالد IEEE به سال ۱۹۸۲ و مدال جان ون نیومن IEEE را در سال ۱۹۹۵ دریافت کرد.او از دانشگاه آکسفورد، پاریس، پترزبورگ و بیش از دوجین دانشگاه وکالج در آمریکا، مدرک دکترای افتخاری دریافت کرده است.

او هم اکنون در محل اختصاصی دانشگاه استانفورد به همراه همسر و دو فرزندش زندگی می‌کند. وی اوقات فراغت خود را به موسیقی می‌گذراند.



فارسی‌تک ابزار قدرتمندی برای تولید متون زیبا

رضا حسامی فرد

دانشگاه صنعتی امیر کبیر
دانشکدهٔ مهندسی کامپیوتر

از زمانی که TEX (بخوانید تِک) توسط پروفسور کنوث معرفی شد، حدودا بیست سال می‌گذرد. از همان ابتدا این سیستم به عنوان ابزار قدرتمندی برای حروف‌چینی متون علمی توسط مجامع علمی مورد استفاده قرار گرفت. در این جا مجالی یافته‌ایم تا به معرفی کوتاهی از TEX و بررسی تاریخچه‌ی آن بپردازیم.

TEX در حقیقت یک زبان برنامه‌نویسی است و ورودی آن یک فایل ASCII است که شامل متن و دستورات لازم جهت صفحه بندی و تعیین فاصله‌ها و ...است .(چیزی که بعدها در HTML هم پرورده شد.) خروجی TEX هم فایلی است مستقل از کلیه‌ی سیستم‌های رایانه‌ای . TEX یک سیستم برای طراحی ساختاریافته‌ی متون است و نه براساس طراحی بصری. سیستم‌هایی که بر اساس wysiwyg[1] طراحی شده‌اند؛ اگرچه برای حروف‌چینی خبرنامه‌های کوتاه و حاوی اشکال گوناگون مناسب هستند ولی استفاده از آن ها برای حروف چینی کتاب ، رساله و مقاله، خصوصا اگر حاوی جدول و فرمول ریاضی باشد، کمی نگران کننده است و در برخی موارد رعایت کلیه‌ی نکات غیرممکن به نظر می‌رسد. در TEX یک بار در ابتدای فایل صورت کلی متن را (کتاب—نامه—مقاله—رساله—یادداشت و ...) را با استفاده از ماکروهای گوناگون به زبان TEX مشخص می‌کنیم. اگر از ماکورهای مناست استفاده کنید مطمئن خواهید شد که خروجی مورد نظر از هر لحاظ مورد قبول شما خواهد بود. در TEX می‌توان با نوشتن ماکروهایی قالب کلی مثلا کارنامه‌های یک مرکز آموزشی را تهیه کرد و تمام کارنامه‌ها را با آن چاپ کرد. غوغای اصلی تک در حروف‌چینی متون ریاضی و فرمول‌ها و جدول‌ها است که کیفیت خروجی خواننده را به وجد می‌آورد و زیبایی خیره‌کننده‌ای به متن می‌دهد. رسم درخت های گوناگون، مهره‌های شطرنج، نت موسیقی و ...تنها قابلیت‌های جزئی تک محسوب می‌شود. اما از کجا آغاز شد؟

در اواخر سال ۱۹۷۰ دونالد کنوث مشغول به اصلاح جلد دوم مجموعه‌ی The Art of Copmuter Programming بود. او با دلخوری به اولین نسخه‌ی حروف‌چینی شده‌ی کامپیوتری جلد دوم نگاهی انداخت به مراتب بدتر از ویرایش اول آن به نظر می رسید . او احتمالا درآن هنگام با خود گفته است «من یک دانشمند علوم کامپیوتر هستم و باید راه حلی برای این معضل پیدا کنم. »[2] او مطالعات خود بر روی حروف چینی متون ریاضی و هر آنچه با حروف چینی زیباتر متون (به خاطر علاماتی که به آنها نیاز پیدا می کرد) در ارتباط بود را آغاز کرد. او گمان می‌کرد این کار حدود ۶ ماه به طول بینجامد و (سر انجام این پروژه عظیم پس از ۱۰ سال به پایان رسید)البته او در حین انجام این کار از کمک دانشمندان بزرگی مانند Herman Hapf , Chuck Bigelow , Kris Holmes , Mattew Carter و Richard Southall که در بخش قدر دانی کتاب "Computer Modern Typefaces" (جلد E از سری "Computer Typesetting") از آنها تشکر کرده است ؛ استفاده کرده است .

حدود یک سال و اندی پس از آنکه او کار خود را آغاز

---

[1] what you see is what you get

[2] کارهای بزرگ از همین جا آغاز می شود

کرده بود از طرف AMS(American Math Society) دعوت شد تادر همایش سالانه آنها مقاله ای را در مورد تحقیقاتش ارائه دهد.این افتخار بزرگی برای محققین محسوب می گردد که در همایش سالانه ای که تعداد زیادی از دانشمندان علوم مختلف (ریاضی ، فیزیک و ...) در آن حضور دارند سخنرانی کند. موضوع سخنرانی که بسته به سلیقه خود فرد انتخاب می شد. دکترکنوث تصمیم گرفت در مورد تحقیقات جدیدش در مورد TEX سخن بگوید. راجع به TEX (برای حروفچینی) و (برای گسترش فونتهای لازم برای TEX). در آنجا او علاوه بر مبانی حروفچینی به مباحث ریاضی استفاده شده در این پروژه که می توان به عنوان مثال به استفاده از معادلات بزبه برای شکل دادن به قطعات و حروف و ...اشاره کرد. که حتی در آن جلسه این دو مبحث کافی بود که دانشمندان معروف علوم ریاضی حاضر درجلسه نگاه دقیقتری به سخنرانی او داشته باشند. در آن هنگام TEX به یک پروژه تحقیقاتی بیشتر شبیه بود تا یک محصول قدرتمند صنعتی. اما نمای جذاب و مشخصی داشت.

—به نظر می رسید تنها نویسنده و منشی های او قادر به استفاده از آن هستند.

— از آنجا کمیک کار آکادمیک بود. کسی قادر نبود هزینه آن را برآورد کند. (هیچ کس قادر نبود در مورد قیمت ان اظهار نظر کند)

— همچنان که بر روی بسیاری از رایانه هاو سیستم عاملهای گوناگون گسترش می یافت و از آنجا که طراحی سیستم به گونه ای بود که همه چیز به فایل ورودی بستگی داشت (Wysiwys نبود) برای هر فایل ورودی یک خروجی یکسان تولید می کرد که این عامل قابلیت حمل آن را بسیار افزایش داده بود.

— نرم افزارهای موجود مشابه در بازار در آن هنگام چندین مشکل عمده را داشت.

— اختصاصی بود.

— گران بود.

—اگر از wysiwyg استفاده شده بود ممکن بود یک اصطلاح خاص در دو جای مختلف معانی گوناگونی بدهند به طوری هرگز به نظر نمی‌رسید که مشابه باشند.

می‌توان گفت دانشمندان علوم ریاضی غالبا کم خرج هستند. گچ و کاغذ و مداد و تخته سیاه تمام ابزار تحقیقاتی آنها محسوب می‌شود. درست کمی قبل از آنکه رایانه‌های شخصی مورد استفاده قرار گیرد TEX معرفی شد. گرچه بر روی یکی از سیستمهای mainframe دانشگاهی گسترش یافته بود. به سرعت بر روی جدیدترین رایانه‌های HP قابل استفاده شد که به عنوان رایانه‌های شخصی شناخته می‌شدند. از همان ابتدا TEX در میان ریاضیدانان ، فیزیکدانان و منجمان ، فضانوردان و محققان کلیه علوم که از کمبود علامات مورد نیاز خود به ستوه آمده بودند بسیار محبوب شد. طراح اصلی TEX برای نوشتن کتب خود ابزارهایی مانند زیرنویس‌ها و درج تصاویر و جداول در متن ، را طراحی کرد. TEX در حقیقت یک مفسر است که مخلوطی از اطلاعات و توضیحات را دریافت می‌کند. توضیحات یک زبان سطح پایین است. (فاصله‌ها را نادیده می‌گیرد، فونت را عوض می‌کند، این جمله را در قالب پاراگراف قرار می‌دهد و ...) به طور کلی می‌توان دلیل اصلی گسترش TEX را قیمت کم آن (مجانی) در نظر گرفت که باعث شده افراد زیادی ماکروهای فراوان به آن بیفزایند. برخی از الگوریتم‌های TEX در ابزارهای معرفی شده با آن تغییر چندانی نکرده‌اند. واضح ترین مثال ممکن شکستن پاراگراف است. متن در TEX به عنوان یک پاراگراف کلی فرض می‌شود، و نه خط به خط.این مبنای الگوریتم شروع برنامه که توسط پیتر کارو در HZ program استفاده شده است. به طورکلی TEX یک زبان برنامه نویسی تک منظوره است که‌در واقع قسمت اصلی سیستمهای حروف چینی محسوب می شود که برای متون ریاضی و به طور کلی تمام متون قابل دسترسی و استفاده شخصی افراد می‌باشد. اما از TEX گفتن و به نپرداختن ظلم بزرگی در حق Knuth محسوب می‌شود.

چیست؟ metafont—برنامه تولید فونت— جدا از TEX است. این برنامه فقط فونت‌های bitmap را تولید می‌کند. (هرچند در سیستم درونی آن زمینه‌ای تولید می‌شود که bitmap هابر روی آن قرار داده شده‌اند.) ولی هنوز مطالعاتی در حال انجام است تا با ترکیب هم‌پوشانی محیط‌های گوناگون به یک زمینه مورد استفاده قرار گیرد. مانند زمینه تایپ یک فونت. تحقیقات و گسترش توسط افراد دیگری انجام می‌گیرد و نتایج آن هم نامیده نمی‌شود. در حقیقت تنهادرقرار گرفته است تا ابزار قدرتمندی برای تولید فونتها باشد. توجه کنید که تنها دلیلی که باعث شد تا بیش از این به پرداخته نشود، این است که قدرت آن در چند سطر نمی‌گنجد. برای این مهم توجه کنید که تولید هر شک و تصویری و هر خطی با آسان به نظر می‌رسد. از تولید تصاویر مهره‌های شطرنج گرفته تا خط نستعلیق و علامات ریاضی. در مقالات بعدی بیش از اینها این ابزار قدرتمند را می‌شناسانیم.

مصاحبه با کنوت

کتاب "هنر برنامه‌نویسی کامپیوتر" شاید خود یکی از شاهکارهای برنامه‌نویسی باشد. مصاحبه‌ی زیر با دونالد کنوت، خالق این شاهکار است.

ترجمه‌ی سولماز مسعودیان

تام مک: سری چهار جلدی شما "هنر برنامه‌نویسی کامپیوتر"برای ۲۵ سال است که به عنوان دایره‌المعارف علوم کامپیوتری از آن استفاده می‌شده است. واقعا به شما باید تبریک گفت. جلد اول این سری، الگوریتم‌های پایه، اولین جلد در ویرایش کامل این کتاب است. چه چیز تازه‌ای در آن برای برنامه‌نویسان وجود دارد؟ چه تفاوتی بین این چاپ و چاپ‌های پیشینی وجود دارد که سال‌ها مطالعه شده‌اند؟

دونالد کنوت: تفاوت‌های برجسته‌ای نخواهید یافت. این تجدید نظر بیشتر یک بالایش بوده تا اصلاح ریشه‌ای؛ ولی کتاب از هر جهت بهتر شده است. بعد از ۲۵ سال تغییر نیافتن نسخه‌ی قدیمی، حالا سعی کردم تا دید یک آدم تجربه دیده را ارائه دهم درحالی که شور جوانی را هم در آن حفظ کرده باشم. اگر شما به طور تصادفی صفحاتی از نسخه‌ی قدیمی را با نسخه‌ی جدید مقایسه کنید، احتمالا چندین نکته‌ی جدید یا جالب یا ... در هر صفحه خواهید دید. این‌ها هستند که کتاب را در مجموع بهتر ساخته‌اند. همچنین صدها تمرین اضافه می‌یابید و خواهید دید که جواب تمرین‌ها بهتر شده است.

ت: برای کسی که تازه با کار شما آشنا شده است، چه چیزی در این نسخه‌های اصلاح‌شده وجود دارد که برنامه‌نویسی را برای او دسترس‌پذیرتر می‌کند؟

ک: این کتاب برای کسانی است که برنامه‌نویسی را جدی می‌گیرند: یک شخص ۵۰ ساله که واجد این روشِ اندیشیدن غریب است که می‌خواهد برنامه‌نویس بسازد. جلد اول، مقدمه‌ای برای بقیه‌ی جلدها است. اجزای صحنه را می‌چیند، الگوریتم‌ها را می‌شناساند، پایه‌ای از ساختمان داده‌ها می‌دهد و مبناهای تکنیک‌های ریاضی را شرح می‌دهد و شما از آن‌ها در بقیه‌ی کتاب استفاده خواهید کرد. سپس من یک کامپیوتر فرضی را معرفی کرده‌ام: MIX؛ که از قابلیت‌های آن برای بنا نهادن همه چیز در واقعیت استفاده می‌شود. من همچنین در تاریخچه، تکنیک‌های پایه‌ایِ برنامه‌نویسی را آورده‌ام که نشان می‌دهند ما چگونه اشیاء را از عالم واقعیت بیرون می‌آوریم و درون دنیای کامپیوتر با آن‌ها به گفتگو می‌نشینیم. راه‌های زیادی برای سازماندهی داده‌ها در ماشین وجود دارد و تمام متدهای اصلی با مثال‌های زیادی، در این کتاب آورده شده است. خواندن کتاب من آسانَ نیست ولی خیلی آسان‌تر از بسیاری از کتاب‌های نوشته شده بوسیله‌ی متخصصین است. برای مثال وقتی من می‌خواهم چیزی را در ریاضی توصیف کنم، آن واژگان فنی‌ای را که جز متخصصین فن، از آن سر در نمی‌آورند، به کار نمی‌برم. اما به هرحال بیشتر متن برای کسی که به طور پیشرفته آموزش ندیده غیرقابل فهم بود و بدین جهت من سعی کردم مفاهیم را تا حد امکان ساده کنم.

ت: شیوه‌ی کارتان چه بود و چگونه توانستید به این چاپ تازه دست پیدا کنید؟

ک: من در مدت این ۲۵ سال تمامی قسمت‌های تصحیح شده را در نسخه‌ی شخصی خودم جمع‌آوری می‌کردم. مردم برای من می‌نوشتند "دون، می‌دونی در صفحه‌ی فلان و فلان، فلان اشتباه هست؟" من همیشه می‌دانستم این اشتباهات وجود دارند و از این‌که می‌دیدم هر سال هزارها کپی از آن کتاب، با وجود این همه اشتباه، چاپ می‌شود، ناراحت بودم. ولی همچنین می‌دانستم که تصحیح آن‌ها کار زیادی می‌برد، چون بسیاری از این اشتباهات، مربوط به ارجاعات داخل خود کتاب بودند. اما در آن موقع بزرگ‌ترین پروژه‌ی من کار کردن بر روی جلدهایی بود که هنوز می‌بایست نوشته می‌شدند. بنابراین هم در این مدت تنها توانستم غلطنامه‌هایی برای جلد اول و دوم و سوم تهیه کنم و بر روی اینترنت قرار دهم. من یک پایگاه‌داده‌ی بزرگی از تصحیح‌ها بوجود آوردم که تا حدود ۲۰۰ صفحه برای هر جلد رسیده بود. زمانی که این لیست تصحیحات را به یک جلسه‌ی گروه کاربران تک نشان دادم، یک نفر از حضار داوطلب شد که کار دشوار ساخت نسخه‌ی اصلاح شده به فرم الکترونیکی را برعهده بگیرد.

بعد از آن، فقط می‌بایست تصحیح‌های اعمال شده دوباره چک شوند. چند نفر برای این کار داوطلب شدند و دو سال وقت گذاشتند و جزئیات کار را انجام دادند تا این‌که بالاخره نسخه‌ی تصحیح شده آماده شد. در ژانویه‌ی سال پیش،من جلد اول و دوم را به فرم الکترونیکی دریافت کردم. آدم احساس متفاوتی نسبت به آن دارد چون می‌داند این چیزی است که به راحتی می‌توان اصلاحش کرد و رشدش داد.

من ۲۵ سال برروی آن کار کردم و خیلی جاها فهمیدم که می‌توان آن را باز بهتر کرد. بنابراین هفت ماه دیگر هم وقت گذاشتم و اکنون این کتاب یک کتاب منحصر به فرد شده است. می‌توانم بگویم که تقریبا کاری انجام داده‌ام که بدان مفتخرم.

ت: کارتان در مورد جلدهای باقی‌مانده چطور پیش می‌رود؟

در ابتدای سال گمان می‌کردم که سه جلد بعدی نیز تا الان آماده خواهند بود، ولی در حال حاضر مشغول اعمال تغییرات مهمی در جلد دوم هستم (الگوریتم‌های نیمه عددی) و سه ماه دیگر هم باید برای جلد سوم وقت بگذارم (مرتب‌سازی و جست‌وجو).

ت: آیا هنر برنامه نویسی اساسا در تمام این سال‌ها بی‌تغییر مانده بود؟

ک: خب، این کتاب به طرق مختلفی تغییر کرده است، ولی مسایل پایه‌ای مثل زیرروال‌ها و تکنیک‌های جایگزینی به جای خود هستند. من سعی کردم دنبال چیزهایی باشم که ۵۰ سال دیگر هم به اندازه‌ی حالا مهم باشند. من می‌خواهم چکیده‌ی آن‌ها را بیرون بکشم و آن‌ها را تا جایی که ممکن است توضیح بدهم و یک کتاب جاویدان به مردم بدهم.

ت: جالب‌ترین پیشرفت را از زمانی که جلد اول بیرون آمد، چه می‌بینید؟

ک: چیزی است که من به آن می‌گویم "برنامه‌نویسی پایه‌ای". تکنیکی برای نوشتن مستند و نگهداری برنامه‌ها بوسیله‌ی یک زبان سطح بالای جمع‌آوری شده که با یک زبان نوشتاری مانند انگلیسی، نوشته شده است. اما چیزی که من بیشتر تاسفش را می‌خورم این است که مردم سعی می‌کنند الگوریتم‌ها را هم ثبت کنند. این یعنی دیگر نوشتن یک نرم‌افزار محال است مگر این‌که شما یک شرکت بزرگ داشته باشید. این امر باعث جلوگیری از پیشرفت می‌شود. اگر یک چنین وضعی قبلا وجود می‌داشتم، من هرگز نمی‌توانستم این کتاب را بنویسم.

دانشگاه مجازی

سعید عباسی مقدم

مقدمه

دنیای امروز با سرعت خیره‌کننده‌ای مرزهای خیال را در می‌نوردد و لحظه به لحظه در تکاپوی به تصویر کشیدن رؤیایی دگر با سرعتی شگرف به سوی آینده پیش می‌رود . ناممکن‌ها ممکن می‌شود، نیست‌ها هست می‌شوند و خیال‌ها به واقعیت بدل می‌گردند.

آری! تحولی عظیم در حال شکل گرفتن است و این انسان فرداست که در تصورات پدران خویش واقعیت را لمس خواهد نمود و دست نیافتنی‌ها را تجربه خواهد کرد.

اگر در گذشته سال‌ها به طول می‌انجامید تا انسان در مسیر زندگی خویش، با صرف وقت و هزینه‌های بسیار تجربه‌ای بیاندوزد و دانشی بیاموزد، در فردایی نزدیک می‌تواند در هر کجای این گیتی پهناور که باشد در دنیایی مجازی، بلادرنگ و بی هیچ هزینه، هر زمان که بخواهد تجربه‌هایی بس گران‌بها به دست آورد و در کنار برجسته‌ترین اساتید عالم، مرزهای دانش را درنوردد.

فن‌آوری نوین این هدیه را به او خواهد بخشید تا به راحتی در کلاس‌های مجازی شرکت کند، به تیم‌های مجازی بپیوندد، در کتابخانه‌ای به وسعت عالم به مطالعه بنشیند و در آزمایشگاه‌های مجازی، آموخته‌هایش را عملی سازد . این رؤیا تبلور خواهد یافت در دانشگاه مجازی!

آموزش از راه دور

از دیرباز گسترش آموزش و بهبود کیفیت آن، یکی از مهمترین دغدغه‌های اندیشمندان و افراد آگاه و مسؤول بوده است. لذا همواره با توجه به رشد فناوری سعی در متحول نمودن روش‌های آموزشی و یافتن راهکارهایی جهت گسترش آموزش و ارتقاء کیفیت آن داشته‌اند.

از این رو با عرضه کامپیوترهای شخصی و گسترش فزاینده آن، با توجه به امکانات چند رسانه‌ای آن، آموزش بر مبنای کامپیوتر در سطحی وسیع و فراگیر مطرح گردید. با گسترش ابزارهای ایجاد محیطهای چندرسانه‌ای و سود جستن از واسطهای گرافیکی کاربر، کامپیوتر به سرعت به یکی از مهمترین ابزارهای کمک آموزشی بدل گردید.

اما آنچه روند رشد روش‌های آموزش از راه دور را با تحولی شگرف رو برو ساخت انقلابی جز اینترنت نبود. رشد برق آسای شبکه اینترنت افق‌هایی بسیار فراتر از پندارهای کهن آموزشی را به سوی نسل آینده گشود .

ایده دانشگاه مجازی

با گسترش روزافزون کاربرد کامپیوترهای شخصی و از سویی روند رو به رشد استفاده از شبکه اینترنت شیوه نوینی از آموزش مطرح گردید . این روش که بر مبنای استفاده از شبکه اینترنت پایه گذاری گردید، به خاطر ویژگی‌های خاص خود توانست به سرعت سایر روش‌های آموزشی را تحت الشعاع قرار دهد.

دریک دانشگاه مجازی، کلیه امکانات یک دانشگاه واقعی اعم از ثبت نام و شرکت در کلاسها، برگزاری امتحانات، انجام آزمایشات و پروژه‌های گروهی، کتابخانه دیجیتال و ... پیش بینی گردیده است .در ابتدایی ترین شکل،دانشجو می‌تواند با استفاده از امکانات این دانشگاه، به راحتی و در زمان دلخواه با استفاده از شبکه اینترنت متن دروس و یا اشکال و نمودارهای مورد بررسی در آن را در یک قالب استاندارد (HTML) دریافت نماید، اشکالات خود را از طریق پست الکترونیک با استاد مطرح کند و سؤالات امتحانی، مقالات و کتاب‌های مربوطه را از سایت مزبور دریافت نماید. بدین ترتیب امکان ایجاد یک رابطه متقابل میان استاد و دانشجو همانند آنچه در یک کلاس واقعی می‌گذرد، فراهم می‌گردد.

در سال‌های گذشته مشکلاتی چند مانع از گسترش و فراگیری این طرح مخصوصا در کشور ما می‌گردید. این مشکلات عبارت بودند از :

- پایین بودن سرعت خطوط اینترنت
- هزینه بالای استفاده از اینترنت
- ناتوانی مرورگرهای اینترنت از در اختیار قرار دادن امکانات چندرسانه‌ای
- ثابت بودن قالب صفحات HTML
- پایین بودن قدرت کامپیوترهای شخصی در فراهم آوردن امکانات چند رسانه‌ای
- عدم وجود هرگونه استانداردی برای انتقال محیطهای مجازی

با کمرنگ شدن و یا از میان رفتن مشکلات مزبور در سال‌های پایانی هزاره دوم، این ایده اکنون می‌تواند جنبه عملی به خود گیرد.

دانشگاه‌های مجازی در یک نگاه

در یکی دو سال اخیر سایتهای بسیاری تحت عناوین آموزش از راه دور (Distance Learning)، دانشگاه‌های مجازی (

Virtual University)، دوره‌های Online و... به طور مجزا و یا به عنوان جزئی از یک سایت بزرگتر ایجاد گشته است . در بسیاری از این سایت‌ها تنها به ارائه برخی از مطالب و یا دروس در قالب ابرمتن اکتفا شده است .در دسته‌ای دیگر علاوه بر ارئه دروس، امکان ثبت نام نیز وجود دارد .

دروس در این‌گونه سایت‌ها غالبا در قالب ابرمتن، PDF و یا PPT ارائه گشته‌اند . در برخی سایت‌ها نیز از ابزارهای ویژه ارائه دروس مانند DazzlerMax و ClassBuilder استفاده گشته است. برخی نیز تنها به نمایش فیلم ارائه درس، توسط استاد و در اختیار قرار دادن متن آن بسنده کرده‌اند.

اگرچه در برخی از این سایت‌ها امکانات ثبت نام وجود دارد اما این ثبت نام بیشتر جنبه تشریفاتی داشته و ثبت نام کنندهٔ از امکانات ملموس بیشتری نسبت به سایر سایت‌هایی که نیاز به ثبت نام ندارند، برخوردار نمی‌شود.

اما در پاره‌ای دیگر این امکان برای ثبت نام کننده فراهم گشته تا در طول ترم با استاد و سایر همکلاسی‌هایش ارتباط بر قرار کند، در جلسات رفع اشکال شرکت کند و بعضا در گروه‌های کاری شرکت نماید . علاوه بر این در برخی نیز در طول ترم بوسیله پست الکترونیک سؤالات نمونه و یا مقالاتی مرتبط با موضوع درس برای شرکت کنندگان کلاس ارسال می‌کنند.

البته در برخی از سایت‌ها، بهنگام ثبت نام از طریق دریافت شماره کارت اعتباری شخص و یا شماره پرداخت بانکی، مبلغی از دانشجو جهت شرکت در کلاس دریافت می‌شود.

در تعداد محدودی از سایت‌ها نیز امکان برگزاری آزمون فراهم گشته است . علاوه بر این برخی نیز امکان استفاده از کتابخانه دیجیتال را در اختیار دانشجو می‌گذارند که البته بسیار محدود می‌باشند.

بالاخره در بسیاری از سایت‌ها نیز امکاناتی نظیر Email و Chat در اختیار دانشجویان قرار گرفته است.

ضرورت‌ها و نیازهای حاکم

رشد برق آسای علوم در سال‌های اخیر، مراکز دانشگاهی را بیش از پیش تحت تاثیر تحولات آموزشی قرار داده است . اگر در ایام گذشته مباحث مربوط به دروس رشته‌های مختلف سالها بدون تغییر و یا با تغییرات بسیار کم تدریس می‌شد، اما اکنون بقدری روند پیشرفت و تغییر و تحولات علوم افزایش یافته که عملا سیستم‌های آموزشی سنتی از همگامی و هماهنگی با آن به طور محسوسی باز می‌مانند.

مخصوصا در کشورهایی که به اصطلاح بیشتر مصرف کننده علم هستند تا تولید کننده آن این امر یعنی فاصله میان آموخته‌های دانشگاهی و تخصص‌های روز مورد نیاز جامعه صنعتی، اقتصادی و حتی فرهنگی و بسیاری زمینه‌های دیگر هر روز بیشتر و بیشتر می‌گردد.

از سوی دیگر بر کسی پوشیده نیست که بروز رسانی پیوسته مباحث مختلف دروس، البته اگر مد نظر قرار گیرد، مستوجب صرف هزینه‌های هنگفت جهت تغییر کتب مرجع، آموزش اساتید، ایجاد هماهنگی و ... خواهد شد . ضمن اینکه زمانبری این عمل بقدری بالا خواهد بود که عملا مباحث فوق تا زمان استفاده دستخوش مرور زمان گشته، تاثیر چندانی در ایجاد توازن میان آموخته‌ها و تخصص‌های مورد نیاز ایجاد نخواهد نمود.

این جاست که ناکارآمدی سیستم‌های آموزشی کنونی در عصر انفجار اطلاعات بیش از پیش خودنمایی می‌کند. بدین ترتیب ساده اندیشی است اگر بکارگیری دانشگاه‌های مجازی در سیستم‌های آموزشی تنها بصورت پارامتری اختیاری که صرفا موجب بهبود کیفی، گسترش کمی و کاهش هزینه‌های آموزش خواهد شد در نظر گرفته شود . هر چند نیل به این اهداف نیز به تنهایی دارای ارزش بسیار والایی بوده، می‌تواند نقش ارزنده‌ای جهت صرفه جویی وقت و هزینه‌های مربوط به احداث و نگهداری واحدهای آموزشی، اداری و خوابگاه، رفت و آمد و آلودگی‌های زیست محیطی ناشی از آن و ... ایفا نماید .

با توجه به آنچه بیان گردید در هزاره جدید میزان توجه به گسترش دانشگاه‌های مجازی به یکی از دغدغه‌های اصلی و حیاتی پیشرفت‌های علمی، اقتصادی، فرهنگی و ... بدل خواهد گشت و عدم توجه به این ضرورت انکارناپذیر کلیه بخش‌های جامعه و کشور را متحمل خسارات جبران ناپذیری خواهد نمود.

بخش تحقیقات دانشگاه مجازی در دانشکده مهندسی کامپیوتر پلی تکنیک تهران

با توجه به آنچه بیان گردید در حال حاضر ایجاد یک سیستم آموزشی نوین بر اساس فناوری روز (دانشگاه مجازی)،یک ضرورت تام و انکار ناپذیر جهت توسعه فن آوری اطلاعات در کشور می‌باشد . با پیش بینی این نیاز حیاتی در دو سال پیش و با توجه به اهمیت امر، مطالعاتی در این زمینه از سال ۷۸ در دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر آغاز گردید.

پس از تحقیقات اولیه در سال ۷۸ که به منظور آشنایی با ابعاد مختلف دانشگاه‌های مجازی، امکانات و ویژگی‌های مختلف متصور در آن و امکان سنجی‌های مقدماتی صورت یافت، در سال ۷۹ روند اجرایی پروژه به منظور ایجاد یک نمونه اولیه (Demo) از یک دانشگاه مجازی فرضی، سمت و سویی عملیاتی به خود گرفت.

هدف اصلی در این بخش از پروژه، ایجاد یک طرح اولیه با استفاده از ابزارهای عمومی نشر بر روی وب، برای ارائه دروس و نحوه عملیات مرتبط با آن مانند آزمون، رفع اشکال،اخذ تکالیف، ثبت نام و ... در یک دانشگاه مجازی بود.

با آماده شدن این نمونه اولیه در اواسط سال ۷۹، بازبینی طرح و بررسی کارشناسی چالش‌های موجود مد نظر قرار گرفت. با مشخص شدن مشکلات و ابهامات موجود، ساخت دوم این نمونه در نیمه دوم سال ۷۹ آغاز گردید.

از سوی دیگر با توجه به تنوع روشها و ابزارهای آموزش از راه دور، همزمان با روند ساخت نمونه‌های مذکور، مطالعات وسیعی جهت شناخت این روشها و ابزارها و بدست آوردن رهیافت‌های عملی جهت پیاده سازی و بکارگیری آنها در دانشگاه مجازی آغاز گردید. این تحقیقات به طور وسیع در زمینه‌های مختلف مانند ابزارهای تهیه دروس، تله کنفرانس، کتابخانه دیجیتال، آزمایشگاه‌های مجازی و ... صورت گرفته است.

به منظور ایجاد هماهنگی میان فعالیت‌های صورت یافته و سازماندهی آن در قالب یک محیط مجازی دانشگاهی به طور عملیاتی، از ابتدای سال ۸۰ بخش ویژه‌ای بدین منظور ایجاد گردید. حاصل این امر که با پیگیری‌های مستمر و مطالعات فراوان بر روی دانشگاه‌های مجازی مختلف و بررسی فعالیت‌های صورت یافته، امکانات موجود و توانایی‌های بالقوه موجود در هر یک از زمینه‌های مختلف، فراهم آمده طرح جامعی است که در آن کلیه بخش‌های متصور و روابط متقابل آن جهت پیاده سازی محیط مجازی یک دانشگاه و امکانات آن مانند سیستم آموزشی، کتابخانه، اداری، مالی، آزمایشگاهها،ارتباط با صنعت، گروه‌های پروژه، بخش فرهنگی، گردهمایی و ... مشخص گردیده است.

در این طرح، ویژگی‌ها، وظایف و عملکرد هر یک از زیرسیستم‌ها مانند ارائه دروس،برگزاری آزمون، ثبت نام، نمایشگاه‌ها، حسابداری، سفارشات صنعتی، ارزیابی، مجله الکترونیک، انتقال اعتبارات، تحلیل قابلیت‌ها، مشاوره، برنامه‌ریزی، خبرگزاری، ابزارهای جستجو، آزمایشگاه‌هوشمند، ایجاد گروه‌ها، ارتباطات بلادرنگ و ... مورد تشریح قرار گرفته است.

ضمن اینکه به منظور ایجاد یک سیستم ساختار یافته، انعطاف‌پذیر و دارای قابلیت گسترش سعی شده تا با دیدی جامع نسبت به سیستم، تحلیل و طراحی کلی آن صورت گیرد، به گونه‌ای که مباحث و ایده‌های موجود برای ارتقاء و گسترش سیستم، قابلیت تطابق با آن راداشته باشند.

از سوی دیگر مرزبندی عملیاتی پروژه به گونه‌ای صورت گرفته که دارای کمترین همپوشانی در مراحل تحلیل، طراحی و پیاده سازی باشد. این امر می‌تواند نقش مؤثری در روند اجرایی پروژه، ایجاد هماهنگی، موازی سازی و تسریع فعالیتها، کنترل وپیگیری پروژه ایفا نماید.

هم اکنون ساخت چهارم دانشگاه مجازی بر مبنای این طرح جامع و امکان سنجی‌های صورت یافته، با بکارگیری پیشرفته ترین ابزارهای موجود مانند Rational Rose(UML Analyse,), Oracle, XML, Java Servlet Adobe Photoshop 6, Ms Office Xp, Ms Project 2000, Macromedia Dreamweaver و ... در حال پیاده سازی است.

نتیجه گیری

همانگونه که مشاهده گردید، ایجاد یک دانشگاه مجازی در مقیاس عملیاتی، امری دور از دسترس برای این بخش پویای تحقیقاتی دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر نمی‌باشد. لذا با توجه به مزایای بیشمار و ضرورت‌های ویژه‌ای که این پروژه داراست، و از سویی فعالیت‌های وسیع صورت یافته در این بخش و توانایی‌های بالفعل و بالقوه آن، انتظار می‌رود که با همکاری و توجه مسؤولین محترم، در آینده‌ای نه چندان دور شاهد بهره برداری یکی از گسترده ترین و قدرتمندترین دانشگاه‌های مجازی در میهن عزیزمان باشیم. انشاءالله !

خبرها

بهبود وضعیت ارایه‌ی دروس

بعد از تلاش‌های دانشجویان و مدیریت دانشکده، اوضاع ارایه‌ی دروس رو به بهبود گذاشته است و توجه بیشتری به دروس پایه معطوف شده است. به طور نمونه: ارایه‌ی درس ساختمان‌گسسته توسط دکتر صادقیان، مبانی کامپیوتر، ساختمان داده‌ها و طراحی‌الگوریتم توسط مهندس پوروطن، سیستم‌عامل و شبکه توسط دکتر پدرام و ریزپردازنده توسط دکتر همایون‌پور. البته هنوز درس‌هایی چون طراحی و پیاده‌سازی نیاز به توجه بیشتری دارند.

فضای مرموز

یک پرسش: کسی می‌داند به فضای محصور بین سایت کارشناسی و سایت کارشناسی‌ارشد و اتاق مطالعه‌ی "سابق" خانم‌ها چگونه می‌توان دسترسی پیدا کرد؟ درب آن کجاست؟

اتاق معاونت پژوهشی

فعالیت‌هایی که در این اتاق انجام می‌گردند کاملا اداری هستند و شامل جذب نیروی علمی برای دانشکده و تسهیلات پژوهشی است.

همایش دانشجویی کامپیوتر دانشگاه آزاد

دانشگاه آزاد اسلامی، واحد تهران مرکز دانشکده‌ی فنی در مهر ماه ۸۰ دومین همایش دانشجویی کامپیوتر را برگزار کردند. در این همایش جایزه‌ی مقاله‌ی ویژه به مهندس محمدرضا خجسته (ورودی ۷۳ کارشناسی و ۷۸ کارشناسی ارشد دانشکده‌ی کامپیوتر امیرکبیر) اهدا گردند. همچنین یکی از جوایز مقاله‌ی برتر نیز به مهندس علی اژدری راد (ورودی ۷۵ کارشناسی و ۸۰ کارشناسی ارشد)اهدا شد.

همچنین در شهریور ماه ۸۱ نیز، چهارمین همایش دانشجویی کامپیوتر دانشگاه آزاد در نجف‌آباد برگزار گردید که باز دانشجویان دانشکده‌ی ما حضور فعالی در آن داشتند.

دو ارتقا در درجه‌ی علمی

درجه‌ی علمی دکتر صادقیان از استادیاری به دانشیاری ارتقا یافت. همچنین درجه‌ی علمی دکتر صفابخش نیز از استادیاری به دانشیاری ارتقا پیدا کرد.

اتاق‌های مطالعه جمع شدند

بدون شرح.

نکته‌هایی در باره‌ی

هنر تست کردن

در پروژه‌های نرم‌افزاری

علی حاجی‌زاده

چهارمین مرحله چرخه حیات هر نرم‌افزاری، آزمودن عملکرد آن است. این مرحله، گلوگاه پروژه‌هاست. اینجاست که بسیاری از نفس می‌افتند و خفه می‌شوند. درست انجام نشدن این مرحله از یک پروژه نرم‌افزاری، می‌تواند باعث شکست کار شود. تست نرم‌افزار یعنی چه؟ یعنی آزمایش درستی عملکرد آن، هم در شرایط معمولی و هم در شرایط خاص، در نقاط مرزی، و در زیر باری مشابه محیط واقعی. «تست نرم‌افزار» یعنی هبوط آن از بهشت برنامه‌نویس، به برزخی شبیه دنیای پرخطر کاربران، برای اینکه ببینیم درست در بزنگاه، او چه می‌کند؟ آیا راه درست را برخواهد گزید؟

اگر می‌خواهید تیم تست نرم‌افزار تشکیل دهید، یادتان باشد که . . .

- این کاری است که اکیدا نباید بر عهده برنامه نویس پروژه گذاشته شود. چون برنامه نویس سعی خود را برای ارائه‌ی برنامه‌ای صحیح کرده است و به دلیل زحمتی که کشیده، احتمال زیادی دارد که خطاهای کار خود را نبیند. او انتظار دارد همه چیز همانطور که برنامه‌اش را نوشته کار کند، و در هنگام تست نرم‌افزار هم (خودآگاه یا ناخودآگاه) می‌خواهد زودتر به همین نتیجه برسد. برنامه نویس آفریننده برنامه است و به راحتی از خطاهای مخلوق خویش چشم خواهد پوشید.
- اعضای گروه تست نرم‌افزار را از بین گروه‌های مختلف کاربران نهایی نرم‌افزار (گروه‌های سنی، شغلی و نیز دارندگان سطوح مختلف دانش کامپیوتری) انتخاب کنید.
- بیشتر اوقات تست کردن یک برنامه وقت‌گیرتر از نوشتن آن است. بنابراین در زمان‌بندی پروژه خود، مرحله تست را اصلا دست کم نگیرید، که ضرر می‌کنید!

اگر عضو تیم تست نرم‌افزار هستید، توجه داشته باشید که . . .

- همه چیز را، همه چیز را، همه چیز را، باید همه چیز را بنویسید. تمام اشکالات باید مستند شوند. مستند سازی (Documentation) مهمترین کار بشر است. چگونگی پیدا شدن اشکال و مراحلی که پیش از رسیدن به آن طی شده و وضعیت محیط اجرای برنامه در هنگام پیدا شدن یک اشکال (از جمله خصوصیات سکو، مانند نسخه‌ی سیستم عامل مورد استفاده و تنظیمات خاص آن) همگی باید طوری نوشته شوند که بتوان دوباره با طی همان مراحل، به همان اشکال برخورد کرد.
- از تخیل خود استفاده کنید. دسته‌های مختلف کاربران نرم‌افزار را در نظر بیاورید و خودتان را به جای آنها بگذارید. فکر کنید آنها ممکن است چه برخوردی با نرم‌افزار داشته باشند.
- به همه‌ی گزینه‌ها و گوشه‌کنارهای نرم‌افزار سر بزنید. بخصوص جاهایی که در روند عادی کار کمتر مورد استفاده‌اند نباید فراموش شوند.
- در جاهایی که کاربر باید ورودی در اختیار برنامه بگذارد، غیرمنطقی‌ترین حالت‌ها را هم بررسی کنید. به هیچ وجه روی هوش کاربر حساب نکنید، او ممکن است احمقانه‌ترین کارها را با برنامه انجام دهد و در صورت انجام این کار، نباید عملکرد نرم‌افزار دچار وقفه شود.
- برنامه را به ازای مقادیر مختلف ورودی، بخصوص حالات مرزی، بیازمایید.
- نرم‌افزار را زیر بار سنگین بگذارید، و ببینید در این شرایط چطور کارش را انجام می‌دهد. ممکن است برنامه‌ای که برای ۱۰۰ مورد کارش را درست انجام می‌دهد، در برخورد با ۱۰۰۰ درخواست دچار فروپاشی شود.
- به عملکرد صحیح برنامه بر روی یک سیستم دلخوش نکنید و حتما آنرا بر روی سیستم‌های مختلف و سکوهای گوناگون نرم‌افزاری و سخت‌افزاری آزمایش کنید.
- به راحتی کاربر فکر کنید. آیا روند کار خسته‌کننده است؟ آیا ممکن است کاربر ترجیح دهد کارها را بدون استفاده از نرم‌افزار انجام دهد؟ پیشنهادهای خود را برای بهتر شدن عملکرد برنامه و خوشحال‌تر بودن کاربر بنویسید.
- در یک جمله: **شما باید بسیار سخت‌گیر باشید.**

حمله و دفاع

علی‌حاجی‌زاده

خلاصه: این مقاله کوتاه اول از همه می‌خواهد به خواننده‌اش یادآوری کند که زندگی روی خط اینترنت چقدر می‌تواند خطرناک باشد. بعد می‌کوشد برای خواننده ناآشنا با خطرات روی خط، تصوری از انها ایجاد کند. به این منظور یک روش معمولی حمله به رایانه‌های شخصی از روی اینترنت مورد بررسی اجمالی قرار می‌گیرد، و در آخر هم چند توصیه، برای آنکه کاربر بی‌گناه لااقل تاحدودی از شر خرابکاران تازه‌کار در امان بماند.

قدم‌به قدم با یک هکر

یک) اولین قدم برای حمله به یک قربانی فرضی، بدست آوردن آدرس IP رایانه او در اینترنت است. آدرس آی‌پی از چهار عدد تشکیل شده که هر کدام بین صفر تا ۲۵۵ هستند و مانند شماره تلفن، برای هر رایانه متصل به اینترنت عددی منحصر بفرد است. با دانستن این آدرس، هکر می‌تواند خرابکاری‌اش را اغاز کند.

یکی از پرکاربردترین سرویس‌های اینترنت، گفتگو روی خط (chat) است و یکی از محبوبترین ابزارهای این گفتگو، Yahoo Messenger. در حالت عادی، هر دو طرف یک گفتگو به سرویس‌دهنده‌ی مرکزی Yahoo! وصل می‌شوند و پیغام‌هایشان با واسطه (و نه به طور مستقیم) به دیگری می‌رسد. ولی مواقعی که با Messenger فایل ارسال یا دریافت می‌کنید، ارتباط مستقیمی بین رایانه شما و طرف مقابل برقرار می‌شود. وقتی این اتفاق افتاد (یعنی هکر شروع به ارسال یک فایل از طریق Messenger برای قربانی‌اش کرد)، می‌تواند با دستور netstat –n آدرس آی‌پی رایانه قربانی را پیدا کند. با پیدا کردن این آدرس، او برای برداشتن دومین قدم آماده است.

تبصره- پیش از پرداختن به دومین قدم، لازم است یادآوری شود که بسیار اوقات ممکن است کار از قدم دوم شروع شود، و آن مواقعی است که قربانی خاصی مورد نظر نیست. در این صورت هکر یک دامنه از آدرس‌های آی‌پی را برای یافتن نقاط ضعف‌شان یکجا بررسی می‌کند.

دو) در دومین قدم، باید نقاط ضعف امنیتی رایانه قربانی پیدا شود. ابزار کمکی این مرحله، برنامه‌های Port Scan هستند. این برنامه‌ها یک آدرس آی‌پی مشخص (یا دامنه‌ای از آدرس‌های آی‌پی) را از هکر دریافت می‌کنند و سپس با آزمودن پورت‌های مختلف آنها، فهرست پورت‌های باز را به همراه اطلاعات مفید دیگری درباره‌ی رایانه‌های مورد نظر به او می‌دهند. وقتی معلوم شد رایانه‌ی مورد نظر کدام پورت‌ها را باز گذاشته، نوبت بررسی این مطلب می‌رسد که آیا پورت‌های باز دارای اشکالات شناخته شده‌ی امنیتی هستند یا نه. در اینجاست که خلاقیت هکر و به روز بودن دانش او درباره‌ی آخرین اشکالات و نقاط ضعف کشف شده به کارش می‌آید.

یکی از پورت‌هایی که روی خیلی از رایانه‌ها باز است، پورت شماره ۱۳۹ است که مربوط به NetBIOS می‌شود. NetBIOS استانداردی است برای دسترسی به شبکه محلی و به اشتراک گذاشتن فایل‌ها و پرینترها و سایر منابع در شبکه LAN. وقتی روی آیکون Network Neighbourhood روی میز کار ویندوزتان کلیک می‌کنید، در واقع دارید NetBIOS را فرا می‌خوانید. برآوردها نشان می‌دهد بیش از ۱۰٪ از کاربران اینترنت از این نظر آسیب‌پذیرند. وقتی هکر از بازبودن پورت ۱۳۹ در یک آدرس آی‌پی مطلع شد، به سادگی (با وارد کردن آدرس آی‌پی در قسمت RUN از منوی Start) به هارد دیسک قربانی دسترسی پیدا می‌کند. اگر قربانی برای فایل‌های به اشتراک گذاشته شده خود Password گذاشته باشد، ابزارهای کمکی دیگری (مانند PQwak) به کار گرفته خواهند شد. اگر قربانی از ویندوز NT استفاده کند، قضیه قدری تفاوت می‌کند. در این صورت با استفاده از یک ایراد در مکانیسم امنیتی IPC$ ، با دستور زیر یک null connection ایجاد می‌شود (که در آن TARGET آدرس آی‌پی قربانی است):

```
NET USE \\TARGET\IPC$ * /USER:""
```

نقاط ضعفی مانند این بسیار زیادند. دو سایت زیر برای شروع مطالعه در این زمینه مقاله‌های خوبی دارند: astalavista.com و cert.org

سه) کار هکرهای حرفه‌ای در همین‌جا و با مستند سازی و ثبت مراحل کار (و شاید هشدار دادن به قربانی درباره وجود نقطه ضعف در رایانه‌اش) تمام می‌شود. ولی هکرهای تازه‌کار یا خراب‌کارهای هکر نما بعد از این مرحله است که به نیت پلید خود عمل می‌کنند. این نیت ممکن است دزدیدن برخی فایل‌ها و اطلاعات قربانی باشد، یا آسیب رساندن به سیستم او و از کارانداختن آن، یا نصب یک برنامه KeyLogger برای سردرآوردن از کارش، یا حتی گذاشتن یک BackDoor روی رایانه وی (مثلا نصب یک ftp Server کوچک).

چند توصیه

- به شدت از اجرای فایل‌های اجرایی دریافت شده از اینترنت روی سیستم‌هایی که امنیت آنها مهم است اجتناب کنید.
- اصولا از دریافت فایل از طریق برنامه‌های Messenger خودداری نمایید. اگر کسی می‌خواهد فایلی برای شما بفرستد، از او بخواهید از Email استفاده کند.
- ویندوز اساسا سیستم عامل امنی نیست و درزهای امنیتی فراوانی دارد. برای کسب حداقل امنیت، روی سیستم‌های ویندوزی که قرار است به اینترنت متصل شوند از برنامه‌های Firewall شخصی استفاده نمایید. این برنامه‌ها پاکت‌های ارسالی و دریافتی رایانه شما را کنترل می‌کنند و در موارد مشکوک شما را در جریان می‌گذارند. یک نمونه دیوار آتش پرطرفدار را می‌توانید از آدرس زیر به صورت رایگان دریافت کنید:

http://www.sygate.com/swat/free/default.php#spf

برای کسب اطلاعات بیشتر دراین باره این مقاله را بخوانید: http://www.cert.org/tech_tips/home_networks.html

آماده باشید یا نه IPv6 می‌آید!

حمیدرضا کاظمی

شبکهٔ جهانی دیگر تنها شامل ایستگاههای کاری،سرورها و کامپیوترهای شخصی نیست. بلکه در حال حاضر شامل PDA ها، تلفنهای همراه و اتومبیلها نیز می‌گردد. پروتکلهای اینترنت قدیمی دیگر نمی‌توانند این میلیونها دستگاه را پشتیبانی و اداره کنند. IPV6 می‌آید و شما باید آماده باشید.

اینترنت و خیلی از شبکه‌های کوچولوی خصوصی، از نگارش‌ها پروتکل اینترنت (Internet Protocol Version4) به عنوان لایه شبکه استفاده می‌کنند که سایر پروتکل مانند TCP قرار گرفته‌اند. IPV4 در اواخر دهه ۷۰ بوجود آمد و در اوایل دههٔ ۹۰ مشخص شد که به انتهای عمرش نزدیک می‌شود.

آخرین آمار رشد سالانهٔ ۶۵٪ اینترنت را نشان می‌دهند. در واقع ترس از تمام شدن آدرس‌های IP بود که منجر به توجه به نگارش جدیدی از IP شد.

◆ تغییرات:

نیاز به مجموعهٔ جدیدی از پروتکل‌های اینترنت ابتدا در RFC 1287 ، مطرح شده «بسوی ساختار آیندهٔ اینترنت» که در دسامبر ۱۹۹۱ چاپ گردید. در اوایل سال ۱۹۹۶ اولین استانداردهای پیشنهادی نگارش IP۶ (IPV6) منتشر شد. در نگارش ۶ چهار تغییر اساسی داده شده بود:

- فضای آدرس دهی بیشتر
- header های ساده‌تر
- جریان‌ها
- تصدیق و امنیت

◆ فضای آدرس دهی بیشتر:

عمومی‌ترین تغییر فضای آدرس دهی بیشتر است. آدرس‌های IPV4 ، ۳۲ بیتی‌اند. تعداد شبکه‌های متصل طی ۱۰ تا ۱۵ سال اخیر در عرض کمتر از ۱۲ ماه دو برابر شده‌اند؛ با توسعهٔ اینترنت، PC ها، PDA ها و حتی تلفن‌های همراه هم به اینترنت متصلند و بنظر می‌رسد فضای آدرس دهی IPV4 بزودی به پایان می‌رسد. در مقابل IPV6 از آدرس‌های ۱۲۸ بیتی استفاده می‌کند. یک راه برای فهم تعداد آدرس‌هایی که با ۱۲۸ بیت قابل نمایش است، تصور کتابی است که کلیه آدرس‌های ممکن را شرح داده باشد و در هر صفحهٔ آن ۱۰۰ آدرس چاپ گردد. اگر ضخامت هر صفحه یک دهم میلیمتر باشد و پشت و رو چاپ شود،ضخامت کتابی که آدرس‌های IP4 را شرح دهد ۲۰۰۰ متر خواهد بود. ممکن است بنظر کتاب بزرگی بیاید، اما ضخامت کتابی که تمام آدرس‌های IPV6 را شرح دهد ۲×۱۰^{۱۶} سال نوری خواهد بود تا به این تاریخ، دورترین نقطه‌ای که توسط تلسکوپ هابل دیده شده است کمتر از ۲×۱۰^{۱۳} سال نوری است.

احتمالاً بخاطر دارید که IPV4 از نمایش "doted decimal" استفاده می‌کرد. مثلاً ۸۹.۶۷.۴۵.۱۲۳ در عوض آدرس‌های IPV6 توسط مقادیر ۱۶ بیتی هگزا دسیمال که توسط ":" جدا شده‌اند نمایش داده می‌شوند. مثلاً FF02:0:0:0:0:200E . بجای نوشتن صفرهای متوالی می‌توان از "::" استفاده کرد، بنابراین FF02::1:200E:8C6C معادل آدرس IPV6 قبلی است. "::" تنها یکبار می‌تواند مورد استفاده قرار بگیرد.

بعلاوه آدرس‌های سازگار با IPV4 نیز موجودند که ۹۶ بایت پر ارزش آن‌ها صفر است. این آدرس‌ها اینگونه نمایش داده می‌شوند: 0:0:0:0:0:0:123.45.67.89 یا 123.45.67.89:: در حالتیکه استفاده از علامت ":" باعث سر در گمی گردد، آدرس IPV6 می‌تواند درون کروشه قرار بگیرد. مثلاً یک URL ممکن است اینگونه باشد http://[fec0::55:a00:20ff:fe90:58f8]

◆ header های ساده‌تر:

header های IPV6 از هشت فیلد یک اندازه تشکیل شده است. در حالیکه IPV4 شامل فیلدهای متغیر است که برخی از آن‌ها اختیاری‌اند. IPV6 ،فیلدهای اختیاری را با قرار دادن آن‌ها در header های جداگانه که بدنبال IP header اصلی است اداره می‌کنند. این تغییرات مسیریابی Packet های IPV6 را بسیار ساده‌تر می‌کنند زیرا فیلدهای کمتری مورد آزمایش و پردازش قرار می‌گیرند.

◆ جریان‌ها:

یک «جریان» توسط RFC 1883 بعنوانی یک سری از Packet ها که پردازش خاصی توسط مسیریاب‌هااند گفته می‌شود. «جریان‌ها» ممکن است برای کنترل کیفیت سرویس برای پشتیبانی کاربردهای "Real Time" مانند صدا و تصویر استفاده گردند.

◆ تصدیق و امنیت:

گرچه امنیت هنگام طراحی IPV4 در نظر گرفته شده بود، e-commerce نیاز به امنیت درونی (built-in) را بیشتر نمود. IPV6 از دو header توسعه یافته برای امنیت استفاده می‌کند: header تصدیق که در RFC 1826 شرح داده شده است encapsulating security payload که در RFC 1827 شرح داده شده است.

header تصدیق اصیلیتان از صحت و تصدیق وی Packet ها را فراهم می‌سازد. encapsulating security payload بصورت کاملاً محرمانه محتویات Packet ها را مخفی می‌سازد.

◆ Autoconfiguration :

آدرس‌های IPV6 از آدرس‌های IPV4 بلندتراند و برای مردم بسیار سخت است که با آن‌ها کار کنند. یکی از اهداف IPV6 این برد که کار مدیریت آدرس‌ها را ساده‌تر کند.

IPV6 برای رسیدن به این هدف پیکربندی اتوماتیک را ارائه می‌دهد که به دستگاه‌های IPV6 امکان ارتباط Plug-and-play را می‌دهد. دو نوع پیکربندی اتوماتیک موجود است: معین و نامعین.

در پیکربندی اتوماتیک نامعین، host ها آدرس IP خود را بصورت اتوماتیک با ترکیب کردن پیشوند شبکهٔ خود که توسط (router) مسیریاب محلی ارائه می‌گردد با آدرس MAC کارت Ethernet خود محاسبه می‌کنند. پیکربندی اتوماتیک نامه در RFC2462 شرح داده شده است.

از آنجایی که برخی از افراد استفاده از آدرس MAC را نوعی تجاوز به امنیت خود می‌دانند، طرح جدیدی راه دیگری در اختیار host ها قرار می‌دهد تا آدرس IP خود را بر اساس یک عدد تصادفی محاسبه کنند. استفاده از یک عدد تصادفی تعیین هویت دستگاه‌ها را برای استراق سمع کنندگان و سایر افرادی که اطلاعات جمع آوری نی‌کنند مشکلتر می‌سازد.

پیکربندی اتوماتیک معین توسط پروتکل DHCPV6 انجام می‌پذیرد. این پروتکل مشابه DHCP فعلی است که برای قطعه، قطعه کردن و بسته‌بندی آدرس‌ها و اطلاعات سرویس‌ها مورد استفاده قرار می‌گیرد.

یکی از مزایای استفاده از DHCPV6 بجای پیکربندی اتوماتیک نامعین این است که DHCPVb به شما امکان پیکربندی دینامیک DNS را می‌دهد. مزیت دیگر آن امنیت است. در پیکربندی اتوماتیک نامعین هر node ای می‌تواند به شبکه شما وصل شود و کار کند. اما با استفاده از server های DHCP می‌توانید کنترل بیشتری روی node هایی به شبکه متصلند داشته باشید. تنها node هایی می‌توانند از طریق DHCP Server پیکربندی شوند که شما به آن‌ها صریحاً اجازه داده‌اید.

◆ تبدیل از IPV4 به IPV6:

RFC2553 که شامل بخش‌هایی از POSIX 1003.1g است سوکت‌های جدید API را برای IPV6 تعریف می‌کند. خبر خوش اینکه API جدید مستقل از پروتکل است. فرقی نمی‌کند که پروتکل شما IPV6,IPV4 یا IPVn است همان کد بدون هیچ تغییر کار خواهد کرد.

این بدین معنی است که حتی اگر هنوز فقط از IPV4 استفاده می‌کنید می‌توانید برای IPV6 برنامه نویسی کنید! API و برنامهٔ شما کار خواهد کرد. وقتی که IPV6 را در شبکه‌تان فعال کنید این برنامه بدون هیچ تغییری باید کار کند.

تغییراتی که در RFC2553 برای IPV6 مطرح شده‌اند شامل چهار قسمت می‌گردند.

- توابع سوکت‌های مرکزی (Core Sockets)
- ساختارهای اطلاعات آدرس‌ها
- توابع مترجم نام به آدرس
- توابع تبدیل کنندهٔ آدرس‌ها

◆ توابع سوکت‌های مرکزی:

توابع اصلی سوکت‌ها که ارتباط را برقرار می‌سازند و قطع می‌نمایند و Packet ارسال داشته و دریافت می‌دارند مستقل از پروتکل طراحی شده‌اند. این توابع در IPV6 تغییر نمی‌کنند. ولی به ساختارهای داده‌های جدیدی احتیاج دارند که در بخش بعدی مطرح می‌گردند.

توجه داشته باشید که این تنها در صورتی است که از سوکتای API استفاده کنید، که ممکن است در یک سیستم embedded اینگونه نباشد: تغییر code یک application فقط وقتی معنی دارد که شما از سوکت‌های یک API شبکه استفاده می‌کنید. اگر از چنین API ای استفاده می‌کنید می‌توانید از (SocketScrubber) که sun تهیه کرده است استفاده کنید.

WWW.sun.com/software/solaris/ipv6

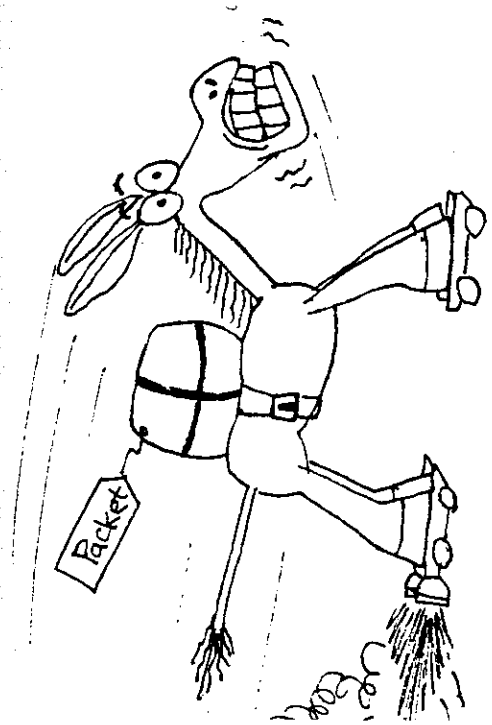
◆ ساختارهای اطلاعات آدرس:

یک ساختار دادهٔ خاص برای هر پروتکلی که توابع سوکتی پشتیبانی می‌کنند،تعریف می‌گردد. شما در حالت عادی این ساختار مخصوص هر پروتکل را هنگام استفاده از توابع اصلی سوکت‌ها به sockaddr تبدیل می‌کنید. مثلاً در IPV4 این ساختار مخصوص پروتکل عبارتست از sockaddr-in و شما هنگامی که تابع ( )connect را استفاده می‌کنید این ساختار را به sockaddr تبدیل می‌کنید. ساختارهای اطلاعات مخصوص پروتکل IPV6 عبارتست از sockaddr-in6 .

بعلاوه spvb شامل خانوادهٔ جدیدی از سوکت‌ها، AF-INET6 ، یک مجموعه پروتکل جدید، PF-INET6 ، و یک ساختار جدید برای نگهداری یک آدرس in6-addr:IPV6 است. این‌ها مشابه PF-INET,AF-INET و in-addr در IPV4 استفاده می‌گردند.

◆ توابع تبدیل نام به آدرس:

در IPV4 معمولترین تابع برای تبدیل نام به آدرس ( )gethosthyname است. این تابع برای ایجاد سازگاری با قبل هنوز وجود دارد ولی برای IPV6 کافی



نیست. مشکل کلی این است که ( )thread-safe: gethostbyname نیست، ولی مشکل بزرگتر این است که هیچ راهی برای مشخص کردن نوع آدرس مورد نظر (IPV4,IPV6,...) نیست. طبق POSIX 1003.1g ( )getaddrinfo این مشکلات را حل می‌کند.

تابع دیگری نیز وجود دارد که بر عکس عمل می‌نماید. تابع ()getnameinfo با گرفتن نام یک ند آدرس IP آن را بر می‌گرداند.

◆ توابع تبدیل آدرس‌ها:

معمولاً در IPV4 توابع ( )inet-addr و ( )inet-ntoa برای تبدیل آدرس از binary به text و بر عکس استفاده می‌شود. مشابه این توابع در IPV6 عبارتند از ( )inet-pton و ( )inet-ntop و () inet-pton یک آدرس را از متن به معادل باینری آن تبدیل می‌کند و ( )inet-ntop برعکس، یک آدرس باینری را به متن تبدیل می‌کند.

بدلیل اینکه هر دوی این توابع یک پارامتر برای تعیین خانواده آدرس دارند می‌توانند هر دو نوع آدرس IPV4 و IPV6 را تغییر دهند.

◆ وضعیت IPV6 در حال حاضر:

هنگامی که IETF کار روی IPV6 را شروع کرد، یکی از موارد ضروری این بود که اجازه دهد IPV4 به آرامی به IPV6 تبدیل شود. کاملاً بدیهی است که تغییر ناگهانی از یک پروتکل به پروتکل دیگر غیر ممکن است. برای اینکه بتوانند IPV6 را موفق سازند، باید همزمان با IPV4 روی اینترنت کار کند. RFC1933 ، «مکانیزم تغییر برای host ها و Router های IPV6» چندین روش برای بکار گرفتن تجهیزات IPV6 ارائه می‌دهد.

◆ ندهای Dual – Stack :

ندهای dual-stack هر دو پروتکل IPV4 و IPV6 را بصورت همزمان پشتیبانی می‌کنند. وقتی که با یک ند IPV6 یا ند dual-stack دیگری ارتباط برقرار می‌کنید ند dual-stack از IPV6 استفاده می‌کند. و هنگامی که با یک ند IPV4 ارتباط برقرار می‌کنید از IPV4 استفاده می‌کند. هر دو پروتکل بدون اینکه برنامه‌ها متوجه شوند مورد استفاده قرار می‌گیرند. در واقع dual-stack نام درستی نیست. در حقیقت تنها یک IP هر دو پروتکل IPV4 و IPV6 را پشتیبانی می‌کند. لایهٔ سوکت بر اساس پارامترهای AF-INET یا AF-INET6 تشخیص می‌دهد که کدام پروتکل مورد استفاده قرار می‌گیرد. اگر از IPV4 استفاده شود. از آدرس که به IPV4 تبدیل شده است استفاده می‌گردد که 80 بیت با ارزش آن صفر است و ۱۶ بیت بعدی 0xffff است و ۳۲ بیت کم ارزش آن عبارتست از آدرس IPV4 . مثلاً FFFF:123.45.67.89:: . بنابراین قسمت IP همواره یک آدرس ۱۲۸ بیتی از لایه‌های بالاتر دریافت می‌کند. وقتی بخش IP آدرس IPV4 تبدیل شده دریافت کند header های IPV4 را می‌سازد و هنگامیکه آدرس تبدیل نشده دریافت کند header های IPV6 را می‌سازد. در هر صورت،این بخش با هر دو پروتکل کار می‌کند و در نتیجه تنها یک Stack وجود دارد.

◆ تونل‌ها:

اگر دو ند IPV6 توسط یک شبکهٔ IPV4 جدا شده باشند چه اتفاقی می‌افتد؟ این دو چگونه ارتباط برقرار می‌کنند؟ با استفاده از مسیریاب‌های dual-stack می‌توان یک تونل در شبکه IPV4 ایجاد کرد. تونل‌ها با استفاده از قرار دادن یک IP Packet در Pay load، Packet دیگری حفر می‌گردند. مسیریاب dual-stack در یک سمت، پکت‌های IPV6 را از فرستنده دریافت داشته و در پکت‌های IPV4 قرار می‌دهد، سپس پکت‌ها را از طریق شبکهٔ IPV6 ارسال می‌دارد و مسیریاب dual-stack دیگری در سمت گیرنده، پکت‌های IPV4 را دریافت می‌کند و پکت‌های IPV6 درو نشان را استخراج می‌نماید و آن‌ها را بسمت مقصد نهایی هدایت می‌نماید. پیکربندی تونل‌ها ممکن است کمی پیچیده باشد. برای سهولت این عمل،در طرح اولیه IETF -«واسطه تونل IPV6» - راه حلی برای اداره کردن اتوماتیک تونل‌ها شرح داده شده است. واسطهٔ تونل یک دستگاه dual-stack است که اقدام به ایجاد، اصلاح و حذف تونل می‌نماید. برای حفر یک تونل شما تنها کافیست که اطلاعات زیر را به واسطهٔ تونل بدهید:

- آدرس IPV4 سمت کلاینت تونل
- یک نام مستعار برای ثبت یک DNS برای آدرس‌های IPV6ای که در هر دو سمت تونل قرار دارند.
- کار کلاینت: یک host مستقل یا یک مسیریاب

برای توضیحات بیشتر به طرح اولیه IETF برای تونل‌ها مراجعه کنید.

◆ برقراری ارتباط با دستگاه‌های IPV4

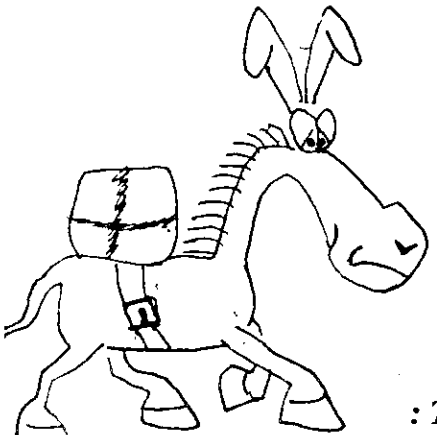
چندین مکانیزم توسط IETF برای برقراری ارتباط برنامه‌های IPV6 با برنامه‌های IPV4 تعریف شده‌اند. برای توضیحات بیشتر به اسناد معرفی شده مراجعه کنید.

SIIT

Stateles IP/ICMP Trans latice الگوریتم header های پکتهای IPV4 و IPV6 را در شبکه به یکدیگر تبدیل می‌کند.

NAT-PT :

ترکیبی از Network Address Translation و Protecol Translation است که برای مسیریابی بکار برده می‌شود. NAT-PT نیاز به تجهیزات مسیریابی dual-stack یا مخصوص را مرتفع نمی‌سازد.



TCP/UDP Relay :

این مکانیزم پکت‌ها را در لایه TCP ترجمه می‌کند. یک مترجم بین ندی که فقط IPV6 است و ندی که فقط IPV4 است قرار می‌گیرد و پروتکل را ترجمه می‌کند.

آشنایی با
ASP.NET

اخیرا سایت‌های جالبی راه افتاده‌اند که برنامه‌نویسی را به زبان فارسی یاد می‌دهند، مانند iranjavascript.com و iranphp.net و iranasp.net. مقاله‌ی زیر برگرفته از سومین است. ASP.NET عضوی از بدنه .NET و متعلق به میکروسافت است. شرکت میکروسافت موفقیت خود را با سرنوشت .NET گره زده ست. بنابراین شاید جالب باشد که بدانیم .NET چیست؟ بدنه .NET دارای دو قسمت اصلی است: قسمت اول یک کتابخانه عظیم از کلاس های آماده است و قسمت دوم یک محیط زمان اجرا می باشد.

کتابخانه کلاس ها در .NET.

کتابخانه .NET دارای بیش از ۳۴۰۰ کلاس آماده جهت استفاده در برنامه ها است. بعضی از این کلاس ها همان کلاس های پایه مانند آرایه و رشته هستند. اما عمده این کلاس ها به پیاده سازی اعمال تخصصی مانند کار با فایل و یا تولید مستندات XML پرداخته اند.

فضانام ها در .NET.

انبوه کلاس های موجود در .NET به حال خود رها نشده اند بلکه این کلاس ها در یک دسته بندی سلسله مراتبی به نام namespace یا فضانام چیده شده اند. بعنوان مثال کلیه کلاس های مربوط به کار با فایل ها در فضانام System.IO دسته بندی شده اند. بنابراین با استفاده از نام کلاس و فضانام مربوطه هر کلاس بصورت منحصربفردی مشخص می شود. بعنوان مثال جهت دسترسی به کلاس File در فضانام System.IO می توان از عبارت System.IO.File استفاده کرد.یک دسته از فضانام ها بطور خاص برای برنامه نویسی در ASP.NET در نظر گرفته شده اند. این فضانام ها با System.Webشروع می شوند. بعنوان مثال کلیه ابزارهای HTML در فضانام System.Web.UI.HTMLControls و ابزارهای Web در فضانام System.Web.UI.WebControls قرار دارند.

مجموعه ها در .NET.

کلاس های موجود در .NET در قالب مجموعه ها (Assemblies) جاسازی شده اند. نکته مهم این است که در اینجا یک مجموعه با یک فضانام اشتباه نشود. یک مجموعه عبارت است از یک یا چند فایل که کد برنامه مربوط به کلاس ها در آن قرار دارد. برای مثال، کلیه کلاس های موجود در فضانام System.IO در یک مجموعه بنام Mscorlib.dll قرار دارد. مجموعه Mscorlib.dll یک فایل معمولی است که در دیسک سخت یک کامپیوتر حاوی .NET در کنار سایر فایل ها قرار می گیرد. برای یک فضانام مهم نیست که کلاس های آن بروی دیسک چگونه ذخیره شده است. کلاس های یک فضانام ممکن است در قالب چند مجموعه ذخیره شده باشند و از طرف دیگر یک مجموعه ممکن است حاوی کلاس های چند فضانام باشد. یک فضانام عبارت است از دسته بندی منطقی کلاس ها در محیط برنامه نویسی در حالیکه یک مجموعه، دسته بندی فیزیکی کلاس ها بروی دیسک سخت است

زبان مشترک زمان اجرا در .NET.

قسمت دوم بدنه .NET یک محیط یا زبان برای زمان اجرا است که زبان مشترک زمان اجرا - The .NET Common Language Runtime (CLR) نام دارد. در این فراخوانی یک صفحه ASP.NET آن صفحه ابتدا کامپایل شده و سپس اجرا می گردد و کد حاصله جهت مراجعات بعدی و جلوگیری از کامپایل مجدد بروی سرور نگهداری می شود. این کد کامپایل شده یک کد میانی است که زبان آن زبان جدیدی است بنام زبان میانی میکروسافت (Microsoft Intermediate Language) یا MSIL یا به اختصار IL نام دارد. MSIL مشابه کد زبان اسمبلی است. با این تفاوت که دارای ویژگیهای شئ گرا است و مستقل از پردازنده کامپیوتر طراحی شده است. صفحات ASP.NET با هر زبانی که نوشته شوند در نهایت به کد MSIL تبدیل می شوند. سپس کد MSIL در زمان اجرا توسط کامپایلری بنام JIT به کد زبان ماشین مربوطه تبدیل می شود. در واقع اینکه صفحات ASP.NET قبل از اجرا به کد میانی MSIL تبدیل می شوند دلایل مهمی دارد. دلیل اول اینکه این مساله دست برنامه نویس را در انتخاب زبان برنامه نویسی باز می گذارد بدون آنکه تفاوتی در نتیجه چه از جهت سرعت و چه از جهت کارآئی برنامه احساس شود. بعنوان مثال شما می توانید یک صفحه ASP.NET را با زبان ویژوال بیسیک بنویسید در حالیکه سرعت اجرای آن همان سرعت برنامه ای است که با C# یا C++ نوشته شده است. و این به آن دلیل است که هر زبانی که برنامه نویس انتخاب کند در نهایت برنامه تولید شده به کد MSIL تبدیل خواهد شد. دلیل دوم اینکه چون همه زبان های .NET به یک زبان میانی تبدیل می شوند، لذا این زبان ها در کنار یکدیگر بخوبی کار می کنند. بعنوان مثال شما می توانید از کلاسی که به زبان ویژوال بیسیک نوشته شده است در زبان C# استفاده نمائید.

کامپیوتر علمی شاعرانه

حامد پیشوایزدی

آدا بایرون، بانو لاولیس، یکی از شایان تصویرترین چهره‌های تاریخ کامپیوتر است. آگوستا آدا بایرون، متولد ۱۰ دسامبر ۱۸۱۵ میلادی، دختر لرد بایرون، شاعر برجسته، بود. پنج هفته بعد از تولد آدا، بانو بایرون خواستار دوری بدون متارکه از لرد بایرون شد و حفاظت از آدایی که از او یک ریاضیدان و دانشمند بارآورد، به او سپرده شد. بانو بایرون از اینکه آدا سرانجام همچون پدرش شاعر شود بیم داشت. به خاطر برنامه‌ریزی بانو بایرون، آدا سرشت شاعرانه خود را متعالی نساخت. او آرزو داشت که یک تحلیلگر و استاد علم ماوراء الطبیعه باشد. در ۳۰ سالگی به مادر خود نوشت «اگر نمی‌توانی به من شاعری اعطا کنی، آیا نمی‌توانی به من «علوم شاعرانه و تخیلی» را ببخشی؟». درک او از ریاضیات با تخیل گره خورده بود و با استعاره و تشبیه توصیف می‌شد.

در سن ۱۷ سالگی آدا به «مری سامرویل»، زن ممتازی که کارهای لاپلاس را به انگلیسی ترجمه کرده بود و متون او در کمبریج مورد استفاده قرار می‌گرفت، معرفی شد. با وجود اینکه خانم سامرویل او را به تحصیل ریاضیات تشویق می‌کرد، او همچنین تلاش می‌کرد که ریاضیات و تکنولوژی را به شیوه‌ای مناسب در زمینه امور بشری بکار گیرد. در یک میهمانی شام، در نوامبر ۱۸۳۴ میلادی، آدا با ایده‌های بابیج در مورد یک ماشین حساب جدید، یک ماشین تحلیلگر، آشنا شد. او فکر کرد، چه می‌شد اگر ماشین حسابگر فقط پیش بینی نمی‌کرد، بلکه می‌توانست براساس پیش‌بینی‌ها عملی انجام دهد؟ فراگیر بودن ایده‌های بابیج، آدا را تحت تأثیر قرار داد. در آن زمان کمتر کسی این چنین تحت تأثیر او قرار گرفته بود.

بابیج روی طرحهای ماشین جدید کار کرد و پیشرفتهایش را در سمیناری در شهر تورین ایتالیا، در پاییز ۱۸۴۱ گزارش کرد. یک ایتالیایی بنام Menabrea خلاصه‌ای از آنچه بابیج توضیح داده بود نوشت و بصورت مقاله‌ای به زبان فرانسوی انتشار داد. در سال ۱۸۴۳ آدا با کنت لاولیس ازدواج کرد و سپس در حالیکه مادر سه کودک زیر هشت سال بود، مقاله Menabrea را ترجمه کرد.

هنگامیکه او ترجمه‌اش را به بابیج نشان داد، بابیج پیشنهاد کرد که یادداشتهای خودش را نیز به آنها اضافه کند و این کار باعث شد که حجم مقاله، به سه برابر میزان اصلی آن برسد. نامه‌های بسیاری که سرشار از خیال و حقیقت بودند میان آدا و بابیج رد و بدل می‌شدند. درمقاله‌ای که او در سال ۱۸۴۳ انتشار داد، نظریات از پیش آگاهانه او شامل پیش بینی ماشینی که بتواند موسیقی پیچیده‌ای تصنیف کند؛ گرافیک تولید کند و بصورت عملی و علمی مورد استفاده قرار گیرد، وجود داشت. حق با او بود.

آدا به بابیج پیشنهاد کرد که طرحی برای ماشینی که بتواند اعداد برنولی را محاسبه کند، بنویسید. این طرح امروزه بعنوان اولین «برنامه کامپیوتری» در نظر گرفته می‌شود (یک زبان برنامه‌نویسی که بوسیله Us DOD ،سازمان دفاع آمریکا، توسعه یافت و در سال ۱۹۷۹ به احترام او «Ada» نامیده شد). بعد از اینکه او توضیحی در مورد ماشین تحلیلگر بابیج نوشت، زندگی او بوسیله بیماری دستخوش رنج و مصیبت شد. زندگی اجتماعی او علاوه بر بابیج شامل «سر دیوید بروستر» (سازنده زیبابین دستگاهی متشکل از سه آینه و مقداری خرده شیشه رنگی برای تولید اشکال گوناگون)، چارلروسمون، چارلردیکنز و مایکل فارادی نیز می‌باشد. علاقه‌مندی‌های او شامل موارد گوناگونی از موسیقی و اسب گرفته تا ماشین‌های حساب، بود. با وجود اینکه عمر او کوتاه بود (او همانند پدرش در ۳۶ سالگی در گذشت)، آدا به اندازه یک قرن در آنچه که تصور می‌کنیم امور کامپیوتری داغ روز است، از ما سبقت گرفته بود.

سوالاتی از مفاهیم کامپیوتر

پاسخ از مهندس پوررضا

◆ تفاوت رویکرد دیجیتال و آنالوگ در چیست؟

دنیای واقع، دنیایی است از پارامترها که انسان از بدو تولد تا مرگ با آن در ارتباط است. پارامترهایی که در تمام ارکان زندگی وجود دارند: حرارت، فاصله، زیبایی، گرسنگی، دوست داشتن، وزن، حقیقت، افراد خانواده و ... مواردی از این دست هستند. برخی از پارامترها دارای این خصوصیت هستند که می‌توانند هر مقداری را شامل شوند. مثلا بین هر دو درجه‌ی حرارتی، می‌توان درجه‌ی حرارت دیگری تعیین کرد. در این حال، این پارامتر دارای مقادیر پیوسته‌ای خواهد بود که می‌تواند در یک محدوده‌ی مجاز هر مقداری را دارا باشد. این پارامتر، یک پارامتر آنالوگ است. در مقابل پارامترهایی هستند که دارای طبیعت گسسته هستند و نمی‌توانند هر مقداری را دارا باشند، مانند افراد خانواده. این پارامتر یک پارامتر گسسته است. در مهندسی مرتبا از این مفهوم استفاده می‌شود که پارامترها، از فضای واقعی خود خارج شده و در فضایی دیگر بیان شوند. مثلا در ساده‌ترین شکل، می‌توان به هر پارامتر یک عدد نسبت داد. در این حال، در رویکرد آنالوگ لازم است که از اعداد حقیقی برای این منظور استفاده شود. در مقابل اگر و فقط اگر از n مقدار برای بیان یک پارامتر استفاده شود (این پارامتر ممکن است دارای طبیعت دو سطحی باشد، مانند حقیقت) رویکرد دیجیتال خواهد بود. در ساده‌ترین شکل می‌توان برای بیان این دو مقدار، از 0 و 1 استفاده کرد. در این حال یک پارامتر گسسته و یا پارامتر پیوسته با یک تبدیل خطی و یا غیر خطی به این فضا تبدیل شده است.

◆ مفهوم زمان در کامپیوتر چیست؟ چرا کامپیوتر به زمان احتیاج دارد؟ CPU بدون زمان نمی‌تواند کار کند؟

در انجام هرکاری بی‌تردید زمان یک پارامتر اساسی است. چراکه هر کاری خود دارای اجزایی است که خود این اجزا برای انجام شدن نیاز به گذشت زمان دارند. گاهی این زمان غیرقابل تغییر است و گاهی این زمان تابع شرایط انجام کار است.

کامپیوتر نیز مستثنا از این قاعده نیست. در یک تعریف ساده، کامپیوتر وسیله‌ای است که قادر به انجام مجموعه‌ای از عملیات پایه و خاص است. از طرفی در یک کامپیوتر (البته منظور کامپیوترهای دیجیتال است) خودِ انجام این عملیاتِ پایه، نیاز به عاملی به نام ساعت دارد. برای انجام هر کاری تعداد تیک‌های خاصی از این ساعت نیاز است. بنابراین با یک بسامد خاص برای تیک‌های این ساعت، زمان انجام هر کاری مشخص و بلاتغییر خواهد بود. از این پس کاربر است که با ترکیب این کارهای پایه خواهد توانست نتیجه‌ی خاصی را دنبال کند. با توجه به این‌که برای رسیدن به یک هدف خاص ممکن است راه‌های مختلفی وجود داشته باشد، از این پس، این مسیر رسیدن به هدف که خود تابعی است از فکر کاربر، زمان رسیدن به این هدف را نیز تعیین می‌کند.

◆ آیا مفهوم Process ، مفهوم Execute را هم در دل خود دارد؟

Process یا فرایند، به عنوان یک مفهوم عمومی، گویای نحوه‌ی رسیدن از یک‌سری ورودی به یک‌سری خروجی است. به عبارتی، Process را می‌توان یک جعبه‌ی سیاه در نظر گرفت که دارای یکسری ورودی و خروجی است. وضعیت خروجی‌ها در هر لحظه تابعی است از ورودی‌های آن لحظه و خروجی‌های لحظات قبل. مفهوم Execute موقعی پیش می‌آید که با یک فرایند نرم‌افزاری سروکار داریم. در این حال اجرای نرم‌افزار، مفهوم Execute را ایجاد می‌کند. بنابراین شاید بتوان Execute را جزیی از یک فرایند پیاده شده به صورت نرم‌افزار دانست.

◆ داده‌ی بامعنا برای کامپیوتر چیست؟ ملاک با معنی بودن برای آن چیست؟خصوصیت یک داده‌ی با معنی چیست؟

معنادار بودن یک داده وابسته به استفاده کننده‌ی آن است. مثلا یک متن فارسی برای یک فرد آشنا با فارسی معنی دارد. در کامپیوتر نیز با معنی بودن وابسته به برنامه است. یک داده می‌تواند برای یک برنامه معنی‌دار و برای برنامه‌ای دیگر بی‌معنی باشد. البته در مواردی که از زبان واحدی در بیان یک مفهوم استفاده شود، می‌تواند اشتراک در بکارگیری آن بوجود آید.

◆ سیگنال یعنی چه؟ رابطه‌ی سیگنال و سیستم چیست؟

سیگنال تکنولوژی نیست بلکه مفهومی است برای نمایش دادن کمیت‌ها. مثلا تغییرات ولتاژ یک سیگنال الکتریکی را به وجود می‌آورد و یا تغییر شدت صوت، یک سیگنال صوتی را. پس سیگنال عبارت است از بیان تغییر یک کمیت فیزیکی نسبت به زمان.

اما از رابطه‌ی سیگنال با سیستم. اول این‌که از خود کلمه‌ی سیستم، باید یک دید سخت‌افزاری داشته باشید. از دید سخت‌افزار، سیستم مجموعه‌ای از دستگاه‌ها است که ارتباط منطقی با هم دارند (در حالیکه از دید نرم‌افزار، سیستم در بسیاری از موارد یعنی قواعد کلی که یک کمیت را مورد پردازش قرار می‌دهند). مثلا در مخابرات سیستم یعنی مجموعه‌ی قواعد و تجهیزات فنی که برای ارسال اطلاعات و ورودی این سیستم هم همان کمیت‌هایی است که اندازه گرفته‌ایم (و به آن سیگنال می‌گوییم). سیگنال و سیستم در مورد کمیت‌های الکتریکی هم به این گونه است: ما می‌خواهیم بدانیم چگونه سیگنال‌های الکتریکی را به دیجیتال تبدیل کنیم و از یک مکان به یک مکان دیگر انتقال دهیم. مجموعه‌ی این قواعد روابط سیستم است و سیگنال هم اندازه‌ی این کمیت‌های الکتریکی.

نظارت در کامپیوتر

سید احسان لواسانی

همه می‌دانیم که سیستم‌عامل‌ها به هدف ایجاد سادگی در استفاده از سخت‌افزار و استفاده‌ی بهینه از منابع سیستم ایجاد شدند. در این مقاله سعی خواهد شد تا گشترش این هدف را در قالب مفهوم "کنترل" بررسی نماییم.

خلاصه‌ی اولیه:

۱- در یک سیستم کامپیوتری، تصور از شیوه‌ی کنترل چنین است: هر چیزی که در هنگام طراحی و ساخت، نظم و انضباط مشخصی در آن "تعبیه" شود به نحوی که تخطی از آن ممکن نباشد.

۲- این نظم یعنی وظایفی که طراحان برای هر چیزی در نظر گرفته‌اند. پس هر اتفاقی که طراحان اتفاق افتادن آن را پیش‌بینی کرده‌اند، یک اتفاق منظم است و سیستم هم در قبال آن می‌تواند یک عکس‌العمل منضبط نشان دهد. در غیر این‌صورت خطا رخ خواهد داد.

۳- تا آن‌جایی که به اجزای سیستم و ارتباطشان با یکدیگر مربوط می‌شود، یک شبکه‌ی کنترل که در هر چیزی درونی گشته، باعث می‌گردد که سیستم همان کاری را انجام دهد، که قرار بود انجام دهد.

۴- این سیستم، بسته نیست و در طراحی آن، ارتباط با کاربر (عالم خارج) نیز پیش‌بینی شده است. اما برای طراحان، این امکان وجود نداشت که تمام فعالیت‌های او را پیش‌بینی کنند، لذا سیستم با یک سری امور غیرمنضبط و نامنظم رو به رو می‌گردد که در کار او خلل وارد می‌سازند.

۵- اجزایی از این سیستم که با کاربر محاوره می‌کنند، وظیفه دارند کاربر را مجبور سازند تا بر اساس منطق معلومی رفتار کند و کاری خارج از پیش‌بینی‌های طراحان آن انجام ندهد. پس رابط کاربر باید بتواند ساختارهای نامنظمی را که از کاربر دریافت می‌کند، منظم و منضبط سازد.

۶- حرکت طراحی این سیستم به سمتی است که این اجبار را با ساده کردن کار کاربر برآورده نماید: "تو بگو چه می‌خواهی، به بقیه‌اش کاریت نباشه." یعنی آن‌چه او می‌خواهد را خود، طراحی و پیاده‌سازی نماید.

۷- این ساده‌سازی در قسمت طراحی، از طریق آموزش متدهای مناسب به متخصصان کامپیوتر، تامین می‌گردد. پیاده‌سازی نیز با در اختیار گذاشتن یک سری omponent های از پیش‌طراحی شده صورت می‌گیرد.

توضیح:

به این ترتیب یک سیستم کامپیوتری ترجیح می‌دهد کسانی هم که با او سروکار دارند، جزئی از او باشند. یعنی بر اساس منطق و نظم او کار کنند. ریزپردازنده از آن‌جا به وجود آمد که برای هر کاری نمی‌توانستیم یک PLA یا ROM را برای آن برنامه‌ریزی نماییم، لذا یک "سری Component های کوچک نرم‌افزاری به نام Instructions طراحی شدند. از این پس یک کار مشخص، توسط ترتیبی از Instructionsها انجام می‌شد و به این مجموعه کلمه‌ی "برنامه" اطلاق گشت. ریزپردازنده‌هایی نیز طراحی شدند که وظیفه دارند این برنامه‌ها را در یک چرخه‌ی بی‌نهایت از Execute and Fetch قرار دهند. کامپیلرهایی نیز طراحی شد تا بررسی کند این مجموعه آیا از آن نظم پیش‌بینی شده در ریزپردازنده‌ها تبعیت می‌کند یا خیر.

یک برنامه، نماینده‌ی کاربر در سیستم است و تمام دردسرها و مشکلاتی که یک کاربر ایجاد می‌کند نیز، ایجاد می‌کنند. همراه با راه حل ریزپردازنده‌ها، مشکل توزیع منابع (مانند زمان cpu، حافظه‌ی در اختیار آن و ...) نوبت‌بندی و ترتیب اجرای

برنامه‌ها، اولویت‌دهی آن‌ها، جلوگیری از کارهای پیش‌بینی نشده‌ای که انجام می‌دهند، نیز بوجود آمدند. برای نمونه: درخواست یک کار از یک جزئی که طراحان سیستم برای آن چیزی پیش‌بینی نکرده‌اند و جزو وظایف آن نیست (مثلا درخواست انتظار از ریزپردازنده یا دسترسی به حافظه‌ای که در اختیار برنامه‌های دیگر است). به دنبال این مشکلات، یک چیزی دیگر به سیستم اضافه گشت به نام سیستم‌عامل.

همان‌طور که این مشکلات را انسان‌ها بوجود اوردند، راه‌حل‌های آن نیز انسانی بود. کنترل و منضبط کردن این برنامه‌ها به وسیله‌ی یک مرکز کنترلی با نام سیستم‌عامل راه‌حل هزینه‌بر و محدود کننده‌ای بود. راه حل بهتر ایجاد یک شبکه‌ی نظم‌دهنده است. سعی بر آن شد که کار سیستم‌عامل در تمام قسمت‌هایی پخش گردد که ورودی سیستم و نقطه‌ی ارتباط با کاربر محسوب می‌شوند. یعنی آن نظمی که سیستم‌عامل موظف به تامین آن بود، در خود ساختارهای کاربر، "تعبیه" شود و به نوعی، کاربر هم "خود کنترل شده" و جزیی از سیستم گردد. رشد زبان‌های برنامه‌نویسی سطح‌بالا و بعد هم زبان‌های برنامه‌سازی Visual را می‌توان به این‌گونه توجیه نمود که با گسترده‌تر کردن Componentهای اولیه در ROM ریزپردازنده، عمده‌ی پیاده‌سازی را خود بر عهده گرفتند. زبان‌های برنامه‌سازی Visual با در اختیار گذاشتن Componentهای که نظم دلخواه کامپیوتر از قبل در آن‌ها تعبیه شده است، کنترل بیشتری بر ساختارهای کاربر اعمال می‌کنند. به این وسیله می‌توان نظمی را که کاربر "سرخود" به برنامه‌ی خود داده است حذف نمود و کاربر را کنترل و منضبط کرد. حتی در سیستم‌عامل هم Monitorها جای Semaphorها را گرفتند، به این دلیل ساده که Semaphorها قدرت زیادی در اختیار کاربر قرار می‌دادند. ولی می‌بایست ابزار مشابهی در اختیار او قرار می‌گرفت که هم نیاز کاربر را برآورده سازد و هم آن انضباط مورد نظر طراحان در آن تعبیه شده باشد، که کاربر با استفاده‌ی از آن ابزار، خود به خود آن نظم را هم رعایت کرده باشد. ساده‌سازی و انجام آن‌چه کاربر می‌خواهد، بهترین و موجه‌ترین روش کنترل است. از این پس کاربر تنها کافی است بداند چه می‌خواهد. بقیه به او مربوط نیست چراکه از پیش انجام شده است.

در مرحله‌ی طراحی نیز تلاش بر درونی کردن این انضباط نجود داشته است. مثلا یکی از معایبی که برای روش Functional Decomposition ذکر می‌کنند این است که سلیقه و دیدگاه طراح در آن دخیل است و دو طراح تحت تاثیر نگاه‌شان به مساله، ممکن است یک فرایند را به زیرفرایندهای متفاوتی تجزیه کنند. از این رو در متدهای بعدی طراحی، سعی بر این شد که هرچه بیشتر تاثیر شخصِ طراح را کم کنند. یعنی هم محیط‌های مهندسی در اختیار او قرار دهند که با طراحی در آن‌ها، به طرز پیش‌بینی شده‌ای طراحی خود را انجام دهد (Case Tools) و هم متدهایی را به او یاد دهند که کمابیش مطمئن گردند که هرکس آن‌ها را به کار گیرد به طراحی واحدی خواهد رسید.

خلاصه‌ی ثانویه:

پس یک سیستم کامپیوتری نیازمند کنترل مرتب، دقیق و راهبری جریانِ استفاده‌ی کاربر از آن سیستم است. اما عمده‌ی این کنترل باید قبل از ورود کاربر به سیستم اعمال شود تا سیستم بتواند به او نیز به عنوان یکی از اجزای خود نگاه کند که با منطق و نظم او کار می‌کند. کاربر هم باید "خود کنترل شده" گردد. این انضباط از طریق ابزارهای ارتباطی به ساخته‌های کاربر منتقل می‌شود. این ابزارها با انجام دادن طراحی و پیاده‌سازی برای کاربر و ساده‌تر کردن کار او، کارهای پیش‌بینی نشده‌ی او را حذف می‌کنند. به این ترتیب، کاربر برای سیستم "یکی از اجزا" محسوب می‌شود و سیستم از دید کاربر، یک جعبه‌ی سیاه که به او می‌گویی چه می‌خواهی و او آن‌را انجام می‌دهد. پس شبکه‌ی کنترل در یک سیستم کامپیوتری سعی دارد به عنوان فیلتری در برابر درز کردن مشکلات انسان‌ها به داخل سیستم عمل نماید. راه حلی هم که به کار بست، راه حلی بود که خود انسان‌ها در جهان خود به کار بسته‌اند: ایجاد یک نظام بوروکراتیک که در آن سعی می‌شود تمام ساختارهای غیرعقلانی حذف شود و یک نظم غیرشخصی در قالب یک شبکه‌ای از نظارت حاکم گردد. به این تربیت دو نوع مهندس کامپیوتر پرورش پیدا می‌کنند. یک دسته یاد می‌گیرند چگونه برای مشکلات انسان‌ها، راه حل کامپیوتری پیدا کنند. یک دسته‌ی دیگر هم تربیت می‌شوند تا بتوانند برای مشکلات کامپیوترها در مواجه با انسان، راه حل انسانی بیابند.

سخت‌افزار تکامل‌پذیر[1]

سید حیدر مرتضوی
Heidar_m@itrc.ac.ir

چکیده

سخت‌افزار تکامل‌پذیر را شاید بتوان شاخه سخت‌افزاری "حیات مصنوعی"[1] دانست. این شاخه علمی و صنعتی جدید همان طور که از نامش بر می آید، از دو قسمت تشکیل شده است: یکی الگوریتمهای تکاملی که در حیات مصنوعی استفاده می شوند، و دیگری سخت‌افزار مناسبی که قابلیت پیاده سازی سخت‌افزاری این الگوریتمها را دارا باشد. این مقاله سعی می کند آشنایی مختصری با سخت‌افزار تکامل‌پذیر، ملزومات آن و برخی از کاربردهایی که در دنیای امروز برای آن تعریف شده است ایجاد کند.

مقدمه

طبیعت و جهان آفرینش از دیر باز، الهام‌بخش بسیاری از دانشمندان، هنرمندان و مخترعان بوده است. تاریخ پر فراز و نشیب بشری همواره شاهد انسان‌هایی بوده که با الگو گرفتن از طبیعت به دنبال ساخت اختراعات و یا یافتن راه حل‌های جدید برای مشکلات بوده‌اند. علاوه بر آن، تصور ساخت ارگانیزم‌هایی شبیه ارگانیزم‌های زنده و در رأس آن‌ها، انسان، از دیرباز یکی از آرزوهای دست نیافتنی بشر بوده است و این سوال همواره مطرح بوده که آیا می‌توان موجودی مصنوعی ساخت که مانند انسان راه برود، ببیند، بشنود و از همه مهم‌تر، مانند انسان فکر کند و تصمیم بگیرد؟ داستان‌های "ایزاک آسیموف"[1] و "باری بی‌بری"[2] کامل‌ترین نمونه تصورات و آرزوهای انسان مدرن در مورد انسان‌های مصنوعی است. جایی که دیگر تشخیص بین انسان‌ها و روبات‌ها گاه بسیار دشوار می‌شود. ولی آیا واقعا زمانی خواهد رسید که این رؤیاها به واقعیت بپیوندند؟ هنوز مشخص نیست؛ ولی به نظر می‌رسد حیات مصنوعی و سخت‌افزارهای تکامل‌پذیر هر روز این رؤیای دست نیافتنی را به واقعیت نزدیک‌تر می‌سازند.

تکامل

تکامل، نظریه‌ای در زیست‌شناسی است که ادعا می‌کند گونه‌های مختلف حیوانات و گیاهان، از گونه‌هایی که قبلا زندگی می‌کرده‌اند، نشأت گرفته‌اند. این نظریه در سال ۱۸۵۹ توسط "داروین"[3] ارایه شد که انقلابی در مطالعات زیست‌شناسی ایجاد نمود. این نظریه بیان می‌کند موجوداتی که توانایی بیشتری در سازگاری با طبیعت دارند، از شانس بیشتری برای زنده ماندن و تولید نسل برخوردارند. این صفات از نسلی به نسل دیگر منتقل می‌شود؛ بنا بر این طبیعت با انتخاب موجودات تواناتر و از بین بردن سایرین، نسل موجودات را مدام به سوی نسلی کاملتر و تواناتر سوق می‌دهد- بقای اصلح. نظریه داروین قادر به بیان شیوه انتقال خصوصیات زیستی از نسلی به نسل دیگر نبود که این خلأ را علم ژنتیک قرن بیستم پر کرد. البته در درستی یا نا درستی نظریه "داروین" هنوز هم میان زیست‌شناسان اختلاف وجود دارد.

الگوریتم‌های ژنتیک

به زبان ساده، الگوریتم ژنتیک، روندی تکاملی است که از روش‌های ژنتیک طبیعی الگو گرفته است. یک الگوریتم ژنتیک، یک روال تکراری است که از یک "جمعیت"[4] با اندازه‌ي ثابت از موجودات تشکیل شده است. برای هر موجود یک سری خصوصیات فردی تصادفی تعریف می‌شود. خصوصیات هر موجود با یک رشته متناهی از علائم (معمولا رشته دودویی) نمایش داده می‌شود که در حقیقت نقش کروموزوم[5] را برای موجود بازی می کند. این رشته یک راه حل ممکن در فضای مسأله‌ي داده شده را کد می کند. این فضا، فضای جستجو نامیده می شود و تمام راه حل‌های موجود برای مسأله را شامل می‌شود. برای آشنایی بیشتر، برخی از مراحل و عملگرهای الگوریتم ژنتیک را بررسی می کنیم.

الف) مقدار دهی اولیه: مرحله مقدار دهی اولیه به‌صورت تصادفی انجام می‌شود. اگر چه می‌توان بعضی افراد را با یک استراتژی حریصانه[6] مقدار دهی اولیه کرد.

ب) انتخاب: این مرحله در صدد است که این ویژگی طبیعی را که مناسب‌ترین افراد شانس بیشتری برای بقاء دارند، مدل سازی کند.

---

[1] Isaac Asimov
[2] Barry B. Brey
[3] Darvin
[4] Population
[5] Chromosome
[6] Greedy

ج) *همگذری*[7] *و تولید مثل جنسی:* همگذری عملگری است که باعث می شود دو کروموزوم به طور متقابل مادهٔ ژنتیک خود را تعویض کنند. به این ترتیب که در صدی از افراد جمعیت، به صورت تصادفی انتخاب می‌شوند و دو به دو به صورت جفت در می‌آیند. سپس بین کروموزوم‌های دو موجود، از یک نقطه مشخص به بعد اطلاعات ژنتیک بین دو موجود تعویض می شود.

د) *جهش:* جهش، عملگری است که یک کروموزوم را انتخاب کرده و به طور تصادفی محتوای برخی از خانه‌های کروموزوم را تغییر می دهد. این عملگر با یک احتمال از پیش تعیین شده عمل می کند و در صورت عمل، کروموزوم جهش یافته جایگزین کروموزوم اصلی می شود. در این حالت اندازه جمعیت ثابت می ماند.

سخت‌افزار تکامل‌پذیر

سخت‌افزار تکامل‌پذیر، تلاش برای پیاده‌سازی یک الگوریتم تکاملی- شبیه آنچه در طبیعت صورت می‌گیرد- بر روی سیستم‌های سخت‌افزاری است. به طور کلی سخت‌افزار تکامل‌پذیر از دو بخش

الگوریتم تکاملی + سخت افزار مناسب =
سخت افزار تکامل پذیر

عمده تشکیل شده است:

در سخت‌افزارهای تکامل‌پذیر فعلی، معمولا الگوریتم ژنتیک به عنوان الگوریتم تکاملی، مورد استفاده قرار می‌گیرد. سخت‌افزار مورد استفاده نیز معمولا یک FPGA است.

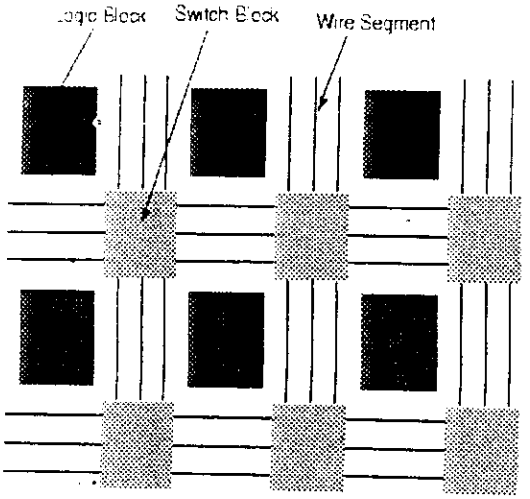
FPGA [8] نوعی سخت‌افزار قابل برنامه ریزی مجدد است که سابقه‌ای ۱۵ ساله در صنعت دارد. FPGAها آرایه‌هایی از بلوک‌های چندکارهٔ دیجیتال هستند که به وسیلهٔ سیم‌های مسیرگزینی به هم وصل شده‌اند (شکل ۱). عملکرد و مسیر بین بلوک‌ها توسط کاربر تعیین می شود که این کار از طریق برنامه‌ریزی تراشه صورت می‌پذیرد. هر بلوک از یک سری مدارهای ساده منطقی شامل می‌شود که وظیفه انجام محاسبات منطقی کوچکی را بر عهده دارند.

به زبان ساده، پیاده سازی یک الگوریتم ژنتیک بر روی FPGA به این صورت است که هر یک از بلوک‌های FPGA، یا قسمتی از یک بلوک، به عنوان یک موجود در نظر گرفته می شوند. کروموزومها در حافظه‌هایی نگهداری می شوند و پس از هر بار اجرای الگوریتم، موجودات مورد ارزیابی قرار می گیرند و بر اساس نتایج، کروموزومها در حافظه‌ها تغییر پیدا می کنند.

مزایای سخت‌افزار

الگوریتم‌های ژنتیک به صورت گسترده‌ای در کاربردهای مختلف استفاده می شوند و به عنوان یک تکنیک بهینه سازی همه منظورهٔ قوی شناخته شده‌اند. اما بسیاری از پیاده سازی‌های نرم افزاری کارآمد، در برخورد با مسایل بسیار پیچیده، کارایی پایینی از خود نشان می دهند.

ساختار محاسبات الگوریتم ژنتیک و سادگی عملیات‌های آن، یک بستر مناسب برای پیاده‌سازی سخت‌افزاری ایجاد می‌کند. موازی‌سازی و خط لوله‌ای کردن[9] عملیات‌های الگوریتم ژنتیک به سادگی توسط سخت‌افزار امکان‌پذیر است. در نتیجه پیاده‌سازی سخت‌افزاری الگوریتم‌های ژنتیک برای مسایل پیچیده، سودمند است.



شکل ۱
شکل ساده‌ای از ساختار یک FPGA

کاربردهای سخت‌افزار تکامل‌پذیر

هر چند سخت‌افزار تکامل‌پذیر هنوز در نخستین مراحل رشد خود قرار دارد، ولی کاربردهای زیادی برای آن تعریف شده است که به طور مختصر، برخی از آنها را بررسی می کنیم.

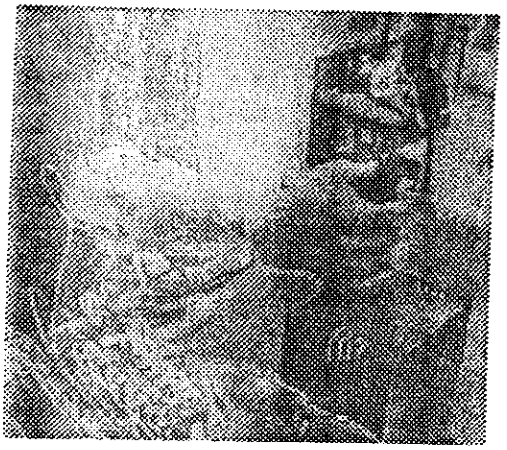
- **کاربردهای پزشکی**

تکامل مصنوعی، امکان استفاده از سخت‌افزار تکامل‌پذیر را در کاربردهای خاص بدنی مطرح ساخته است. تا کنون در بیشتر کاربردهای پزشکی، راه حل‌های سخت‌افزاری به خاطر هزینه بالایی که داشتند کنار گذاشته می‌شدند، ولی سخت‌افزار تکامل‌پذیر می‌تواند راهبردهایی کم هزینه و سریع را در کاربردهای پزشکی ایجاد کند.

برای مثال از سیستم‌های تشخیص الگو، برای تشخیص الگوهای سیگنال‌های عصبی در کنترل دست‌های مصنوعی استفاده می‌شود. تراشهٔ سخت‌افزار تکاملی، عمل یادگیری و تشخیص الگوهای عصبی فرستاده شده را زودتر و دقیقتر از شبکه‌های عصبی پیاده شده متناظر، انجام می‌دهد. به علاوه، از آن‌جایی که نتیجه‌ی نهایی به یک مدار منطقی ساده محدود می‌شود، از شبکه‌های عصبی متناظر قابل فهم‌تر است. و مهم‌تر از همه، از آن‌جایی که کل این سیستم بر روی یک تراشه کوچک پیاده سازی می شود، قابل حمل و بنابراین برای کاربردهای پزشکی مناسب تر است. شکل ۲ این سخت‌افزار و نمونه‌ای از قابلیت‌های آن را نشان می دهد.

---

[7] Crossover
[8] Field Programmable Gate Array
[9] Pipelining



شکل ۲

سخت‌افزار تکامل پذیر برای کنترل دست مصنوعی

کاربردهای فضایی

سخت‌افزار تکامل پذیر، دارای امکان سازگاری خودکار با محیط اطراف خود است. این قابلیت می‌تواند در جاهایی که امکان کنترل دایمی‌توسط انسان وجود ندارد- مانند ماموریت‌های فضایی- بسیار مفید باشد. به خصوص مأموریت‌هایی که امکان برخورد با شرایط خطرناک و غیرمنتظره، در آن‌ها وجود دارد. ناسا قصد داشت از سال ۲۰۰۲ در تمام ماموریت‌های فضایی از سیستم‌های قابل پیکربندی مجدد استفاده کند و پس از آن، استفاده از سخت‌افزار تکامل‌پذیر خودمختار را رواج خواهد داد.

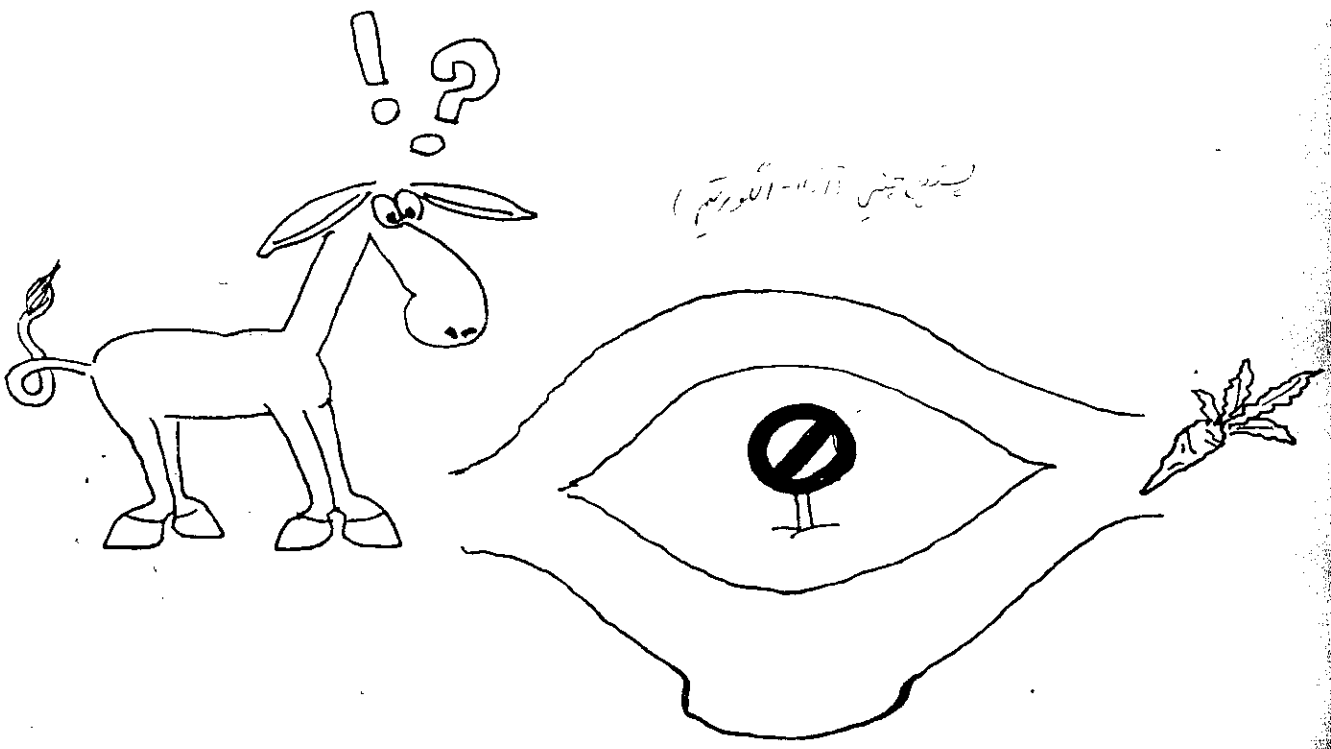
کاربردهای مربوط به حیات‌مصنوعی

یکی از اهداف اولیه محققین حیات‌مصنوعی این بود که چگونگی تکامل، رشد و عملکرد ارگانیزم‌های واقعی را کشف کنند. شبیه سازی سیستم‌های بیولوژیکی در سیلیکون به خاطر تفاوت‌های دو ماده بسیار سخت است. سیستم‌های بیولوژیکی معمولاً آنالوگ، سه بعدی و بر پایه واکنش‌های کند شیمیایی استوار هستند. سیستم‌های کامپیوتری معمولاً دیجیتال، مسطح و بر پایه واکنشهای سریع الکتریکی استوار هستند و این امتیاز سرعت در شبیه‌سازی سخت‌افزاری این سیستم‌ها بیشترین کاربرد را دارد.

در پایان

سخت‌افزار تکامل پذیر، مقوله جدیدی است و با این که زمان زیادی از ورود آن به عرصه صنعت نمی‌گذرد، توانسته به خوبی جای خود را در کاربردهای مختلف علمی و صنعتی باز کند و به نظر می‌رسد که در آینده حرف‌های بیشتری برای گفتن داشته باشد.

مباحث مربوط به سخت‌افزار تکامل پذیر بسیار گسترده‌تر از آن است که در این مقاله بتوان آن را مورد بررسی قرار داد. هدف اصلی این مقاله، صرفا معرفی یک زمینه تحقیقاتی جدید و ایجاد انگیزه برای دانشجویان و پژوهشگرانی است که به فعالیت تحقیقاتی در چنین زمینه‌هایی علاقه مندند.



پارادکس‌های زبان‌های برنامه‌نویسی AI

گویا همه قبول کرده‌اند که کارهای راسل و تارسکی نشان داده است که اگر ما می‌خواهیم از مفاهیمی مانند True (که متعلق به زبان منطق هستند) استفاده کنیم، به هرمی از متازبان‌ها احتیاج داریم. پس اگر بخواهیم درست استدلال کنیم و دقیق تئوری‌سازی کنیم؛ باید این هرم را رعایت کنیم تا این زبان‌ها که در سطوح مختلفی تعریف شده‌اند، با هم خلط نگردند. مثلا منطق (یکی از متازبان‌ها) زبانی است که روی زبان عادی و روزمره تعریف شده است؛ پس اگر در زبان عرفی، استدلال منطقی کنیم و از مفاهیمی مانند درستی و نادرستی استفاده کنیم، این دو زبان را با هم قاطی کرده‌ایم. البته به هر صورت زبان روزمره‌ای که ما به کار می‌بریم، شامل متازبان خود هم هست. یعنی هم نادقیق است و هم برای بیان استدلال‌های منطقی به کار می‌رود. زبان ما، همیشه این دو خصیصه را با خود به همراه داشته است.

حرف‌های بالا از سر و کله زدن با جملاتی مانند

"این جمله درست نیست."

حاصل شده است. چنین گزاره‌ای نادرست است اگر درست باشد و درست است اگر نادرست باشد. این جمله و روایت‌های دیگر پارادکس دروغگو و پارادکس‌های شبیه به آن، نشان می‌دهند که اگر قانونی که می‌گوید یک گزاره یا درست است یا نادرست، صحیح است؛ پس در زبان‌هایی که متازبان خود را هم شامل می‌شوند. تناقض (مانند همین پارادکس دروغگو) وجود دارد.

بسیاری از فلاسفه و منطقیون از بحث بالا نتیجه گرفتند که تنها اگر ما هرم متازبان‌ها را رعایت کنیم (یعنی متازبان یک زبان را از خود آن زبان جدا کنیم و مثلا برای استدلال، از منطق کاملا صوری که در ریاضیات جدید مقداری از آن را یاد گرفته‌ایم، استفاده نماییم) می‌توانیم چارچوبی دقیق و مستحکم برای تئوری‌سازی در علم و ریاضیات بنیان کنیم.

حالا در هوش‌مصنوعی (و نیز زبان‌های برنامه‌نویسی) هم آمدند همین کار را کردند. یعنی زبان‌هایی ایجاد کردند که این سلسله مراتب را رعایت می‌کنند تا دچار چنان تناقضاتی نشوند . . . اما چنین نیست! مرزهای هر زبانی با متازبان خودش مغشوش است و هر چقدر هم که سعی کنیم این خط‌کشی‌ها را رعایت کنیم، از این دست تناقضات رخ خواهند داد؛ به هر حال این زبان‌ها، زبان ما را نمایش می‌دهند.

اکنون چند مثال می‌آوریم از برنامه‌های دقیق و مستحکمی که به صورت گسترده‌ای هم استفاده می‌شوند، و نشان می‌دهیم که چگونه پارادکس‌های مشابهی را به طور ساده و قابل فهمی ایجاد می‌کنند.

در Pop-11 (زبان برنامه‌نویسی AI) می‌توانید ارزش درستی یک لیست از برنامه را توسط یک برنامه‌ی کوچکی تعیین کنید. چنین برنامه‌ی کوچکی به این صورت تعریف می‌شود:

```
define ISTRUE(list);
    popll_compile(list) = true
enddefine;
```

هنگامی‌که رویه‌ی popll_compile بر روی لیستی از برنامه اجرا می‌شود، آن لیست را کامپایل و اجرا می‌کند. حالا اگر از pr برای چاپ نتایج حاصله از ISTRUE استفاده کنیم:

```
pr(ISTRUE( [ 8 > 5 ] ));
```

نتیجه‌ای که چاپ می‌کند،چنین است:

```
<true>
```

چراکه ۸ بزرگتر از ۵ است. در حالیکه

```
pr(ISTRUE( [isinteger("cat")]));
```

نتیجه‌ای که چاپ می‌کند،چنین است:

```
<false>
```

چراکه کلمه‌ی "Cat"، *integer* نیست. اما ما می‌توانیم متغیری به نام S تعریف کنیم:

```
vars S;
```

و بگوییم آن‌چه S می‌گوید درست نیست:

```
[not(ISTRUE(S))] -> S;
```

و از Pop-11 بخواهید که چک کند و ببیند آیا S درست است یا نه:

```
pr(ISTRUE(S));
```

سیستم کمپ می‌کند و یک پیغام خطا می‌دهد که infinite recursion تولید شده است. چراکه خواسته است بفهمد که آیا S درست است یا نه که برای این احتیاج دارد بفهمد که آیا S درست است یا نه که برای این هم احتیاج دارد بفهمد که آیا S درست است یا نه که برای این هم باز احتیاج دارد تعیین کند که آیا S درست است یا نه و ...

این برنامه حتی به تناقض هم نخواهد رسید چون خود پروسه‌ی چک کردن، متوقف نخواهد شد مگر آن که حافظه تمام شود. البته اگر تعریف تناقض را این گونه گسترش دهید که: هر گزاره‌ی خوش فرمی (از جمله همین گزاره‌ی S) باید یک ارزش قطعی داشته باشد؛ پس می‌توانید بنویسید که تناقضی هم رخ داده.

همین کار را هم می‌توانیم برای پارادکس راسل انجام دهیم. زبان Pop-11 هم مانند زبان‌های دیگر، رویه‌های از پیش‌ساخته‌ای دارد که مانند یک محمول بر روی گزاره‌های ورودی‌شان عمل

می‌کنند. همانند isinteger و isword و isprocedure و ... در این زبان ۱۱ رویه‌ی این چنینی وجود دارد: به این صورت که:

```
pr(isinteger(3));
<true>
pr(isprocedure(isinteger));
<True>
pr(isinteger(isinteger));
<false>
pr(isprocedure(isprocedure));
<true>
```

ما رویه‌ای به نام RUSSEL به این صورت تعریف می‌کنیم که هر رویه‌ای را بر روی خودش اجرا کند و سپس حاصل را هم نقیض کند:

```
define RUSSELL(f);
    not(f(f))
enddefine;
```

به این ترتیب، RUSSEL به عنوان یک محمول تعریف شده است.

دستور

```
pr(RUSSELL(isprocedure));
```

باعث می‌شود که isprocedure برای خودش صدا زده شود که حاصل آن خواهد بود:

```
<true>
```

و بعد هم این حاصل را نقیض می‌کند و بنابراین:

```
<false>
```

به عنوان حاصل نهایی چاپ خواهد شد. به طور مشابه هم:

```
pr(RUSSELL(isinteger));
```

نتیجه‌اش

```
<true>
```

خواهد بود چراکه isinteger یک رویه است و نه یک integer. پس این رویه هم تعریف دقیقی دارد و معمولا (!) کار می‌کند. اما اجرای دستور زیر آغاز دردسر خواهد بود:

```
pr(RUSSELL(RUSSELL));
```

این دستور تمام نخواهد شد مگر آن که چک کند که اگر RUSSEL را روی RUSSEL اجرا کنیم، حاصل True خواهد بود یا False. اما برای این کار همان آزمایش پیشین لازم است. پس باز هم سیستم یک دور بی‌نهایت را شروع کرد.

نتیجه آن که اگر قرار باشد یک زبان، هر چیزی را پیاده‌سازی کند، چنین تناقضاتی اجتناب‌ناپذیراند؛ هر چقدر هم که هرم متازبان‌ها را رعایت شده باشد. تعجبی هم نیست! به هر حال مدلی از زبان طبیعی هستند.

مبنای یک و مبنای صفر

رضا حسامی‌فر

قضیه‌ی بنیادین حساب را به خاطر بیاورید:

$$x=a_n b^n+a_{n-1}b^{n-1}+\ldots+a_1 b+a_0$$

هر عدد را می‌توان به این صورت نوشت اما به دو شرط: اولا $0 \le a_i < b$ باشد و دوما $b>1$. به این صورت هر عدد می‌تواند به مبنایی بزرگ‌تر از یک تبدیل شود. این البته چندان امر غریبی نیست، چیزی که عجیب به نظر می‌رسد، شرط $b>1$ است:

به طور مثال شمردن اعداد را در مبنای دو (یا هر مبنای دیگری) در نظر بگیرید:

0	1	10	11
هیچ	صفر	یک	دو

همان‌طور که ملاحظه می‌کنید، هر بار یک واحد اضافه می‌کنیم، و درواقع تا رسیدن به اولین صفر، همه‌ی یک‌ها را صفر می‌کنیم و اولین صفر را هم یک می‌کنیم.

حالا بیایید همین مکانیزم را برای مبنای یک پیاده کنیم:

0	0	→	∞
هیچ	صفر		تا بینهایت

ببینید برای معرفی یک در مبنای یک، باید بینهایت صفر را کنار هم بگذاریم. ممکن است بگویید که ما در مبنای یک عدد واحد نداریم که اضافه کنیم؛ اصلا یک نداریم! اما من باز می‌گویم که این نتیجه را از تقسیم‌های متوالی می‌گیریم؛ هر چه باشد، این هم یکی از روش‌های تبدیل عدد به مبنای دیگر است.

نتیجه‌ی اول) به نظر من می‌آید که مبنای یک، برای معرفی ∞ است.

اما مبنای صفر:

ملاحظه کنید که باز

هیچ

هیچ را داریم اما اگر بخواهیم صفر را بنویسیم، دستمان بسته است: صفر نداریم که بنویسیم. پس برعکس معمول که هیچ را با صفر یکسان در نظر می‌آوردیم (حداقل خودم را می‌گویم) این مبنا نشان می‌دهد که صفر همان هیچ نیست.

نتیجه‌ی دوم) مبنای صفر برای آن است که بفهمیم صفر با هیچ فرق دارد.

نتیجه‌ی کلی) مبنای صفر و یک وجود دارند و کاربرد هم دارند؛ هرچند دنیای متفاوتی داشته باشند.

لطفا توجه داشته باشید که علی‌رغم همه چیز، این بحث کاملا جدی است.

ریاضی‌دانی که بی‌اعتناست

مصاحبه با محمدرضایی

کامپیوتر طبق تعریف، رشته‌ای است در اختیار دیگر رشته‌ها؛ اما گاهی پیش می‌آید که [illegible] رشته‌ی سرویس‌گیرنده، کارهای بعیدی در زمینه‌ی رشته‌ی [illegible] از این دست است. او رشته‌ی ریاضی خوانده ولی برنامه‌ای [illegible] انجام می‌دهد و قضایای ریاضی اثبات می‌کند. در این مصاحبه ما [illegible] تاکید کرده‌ایم چراکه بقیه را علی‌القاعده خود بلدیم.

موضوع اثبات قضایای ریاضی است. تا وقتی که ما با گزاره‌های ریاضی کار می‌کنیم و هنوز محمول‌ها وارد کار ما نشده‌اند، صورها وارد کار ما نشده است، می‌توان روش‌هایی را طراحی کرد که با آن روش‌ها درستی یا نادرستی قضایا یا عبارت‌های منطقی را تشخیص داد. ساده‌ترین روش آن این است که یک جدول ارزش تشکیل بدهیم و آن predicate هامان را در آن جدول ارزش، ارزش‌دهی predicate هم ضربدر دو می‌شود. یعنی اگر n تا predicate داشته باشیم، دو به توان n ردیف در جدول ارزش خواهیم داشت. و در نهایت تشخیص بدهیم که یک عبارت منطقی، آیا منطقا صادق هست یا نه. منظور هم از *منطقا صادق* این است که ستون آخر آن به ازای هر ارزشی یک بشود، True. یا به عبارت ریاضی یا منطقی آن یک توتولوژی باشد. اما وقتی که محمولات وارد موضوع می‌شود، سورها وارد کار می‌شود، دیگر حرف زدن به این آسانی نیست.

در خود قضیه‌ی گودل یک روش عددی از اثبات کردن قضایا وجود دارد و با همین روش عددی است که بیان می‌کند یک سری گزاره‌های اثبات‌ناپذیر در هر دستگاهی وجود دارد.

- ***قبل از این‌که جلوتر برویم، منظورتان از محمول چیست؟***

حساب محمولات یا quantifiers ... مثلا در نظر بگیرید که می‌گوییم P(x). این یعنی Pیک محمول برای x است. یعنی می‌گوییم x دارای خاصیت P است. مثلا اول بودن در اعداد طبیعی: می‌گوییم x اول است. یا در مجموعه‌ی انسان‌ها: می‌گوییم x کچل است. یعنی یک خاصیتی را روی یک مجموعه بررسی می‌کنیم. مثلا در همین مثلا، خاصیت کچل بودن روی انسان‌ها [illegible] که می‌گوییم اگر دامنه‌ی اشیایی که بررسی می‌توانیم آن را چک کنیم. باز مثلا همه کچل [illegible] اشیا را از مجموعه‌ی انسان‌ها بیرون بیاوریم و از او بپرسیم که "آیا کچلی؟" و می‌توانیم صدق یا کذب این گزاره را چک کنیم که "همه‌ی انسان‌ها کچل هستند." به این می‌گوییم محمول. به این [illegible] حالا می‌توانیم روابط را بیشتر کنیم. مثلا [illegible] "همه با هم دوست هستند." این یک گزاره است که می‌توانیم [illegible] را بررسی کنیم. "هر کسی یک دوستی دارد." این [illegible] یک گزاره است که یک پایگاه داده‌ی قوی داشته باشیم تا بتوانیم در آن [illegible] و ببینیم این رابطه برقرار هست یا نه و این گزاره درست [illegible] اگر دامنه‌ی اشیایی که در مورد آن [illegible] خیلی مشکل می‌شود. مثلا در نظر [illegible] اعداد طبیعی را بررسی کنیم. یک گزاره‌ی [illegible] یا فرد. اگر بخواهیم با روش چک [illegible] متوقف نمی‌شود و ما هیچ وقت [illegible] درست یا درست نیست. چون هر چقدر [illegible] چک کردن وجود دارد و این عمل [illegible] این سؤالی بود که برای من پیش [illegible] یک گزاره را ثابت یا رد می‌کنند؟" این [illegible] زیادی روی منطق کار بکنم و در [illegible] کردم به نام قضیه‌ی ***گودل***. قضیه‌ی [illegible] ریاضی (مانند جبر، هندسی، آنالیز و [illegible] که اثبات ناپذیراند." خیلی جالب [illegible] این دو را از هم جدا کرده است. [illegible] که اثبات‌ناپذیراند؛ اما گودل در

چیست؟ نظریه‌هایی ریاضی نظریه‌هایی هستند که حداقل، حساب را داشته باشند. یعنی شامل اعداد طبیعی باشند. یعنی اگر دستگاهی ۰، ۱، ۲ ... را داشته باشد، در این صورت گزاره‌های درستی در آن یافت می‌شود که اثبات‌ناپذیراند. با توجه به این می‌توان گفت که آن ادعا که می‌گوید "ما می‌توانیم یک ماشین یا الگوریتم ارایه دهیم که بتواند تمام قضایا را ثابت کند، منتفی است." چون اگر برنامه‌ی ما درست کار کند، به ازای آن مساله‌ای که اثبات‌ناپذیر است، نباید جواب دهد و این کلیت آن ماشین را مخدوش می‌کند. اما یک مساله‌ای پیش می‌آید: یک ریاضی‌دان عادی که در آن دستگاه دارد کار می‌کند، آیا او هم می‌تواند آن گزاره را ثابت یا رد کند؟ خیلی از مسایل هنوز بازاند و هنوز هم مشخص نشده‌اند که ثابت می‌شوند یا خیر. مثلا حدس گلدباخ. این یک حدس بسیار ساده است که می‌گوید "هر عدد زوج را می‌توان به صورت مجموع دو عدد اول نوشت." صورت خیلی ساده‌ای دارد اما هنوز نتوانسته‌اند که آن را اثبات یا رد کنند. بعضی از تصورها برای این است که ممکن است این گزاره تصمیم‌ناپذیر باشد؛ یعنی در نظریه‌ی اعداد موجود نه بتوان آن را ثابت نمود و نه رد.

خب حالا من دارم چه کار می‌کنم؟ خود قضیه‌ی گودل وقتی که دقیق مطالعه‌اش کنید، یک روش عددی از اثبات کردن قضایا وجود دارد که با همین روش عددی است که بیان می‌کند یک سری گزاره‌ها هستند که اثبات‌ناپذیر هستند. یک مفهوم دقیق از اثبات کردن ارایه می‌کند و بعد هم در مورد آن یک حرف دقیق می‌زند که ""در اکثر دستگاه‌های ریاضی گزاره‌های درستی یافت می‌شوند که اثبات ناپذیراند."

- ***روش عددی اثبات کردن یعنی چه؟***

او به این ترتیب که الان می‌گویم گزاره‌ها را کد گذاری می‌کند: ما یک سری محمول داریم و می‌آییم آن‌ها را به این طریق نشان می‌دهیم: $A(n,k)$ که در آن n تعداد پارامترهای آن است، یعنی روی چند عضو عمل می‌کند و k یک عدد یکتایی است که به آن نسبت می‌دهیم. به این صورت برای محمول‌های یک نماد واحد در نظر گرفته‌ایم. بعد $f(n,k)$ را هم برای توابعی که داریم تعریف می‌کنیم. مثلا برای جمع کردن. برای همه چیز در اثبات قضایا می‌خواهیم یک زبان مشترک ایجاد کنیم که راحت در موردشان حرف بزنیم. ثابت‌های یک زبان را هم با a_k نشان

قضیه‌ی گودل می‌گوید "در اکثر دستگاه‌های ریاضی (مانند جبر، هندسی، آنالیز و ...) گزاره‌های درستی یافت می‌شوند که اثبات ناپذیراند." خیلی جالب است: هم درستی و هم اثبات‌ناپذیری و او این دو را از هم جدا کرده است.

می‌دهیم و متغیرهای آن را نیز با x_k نشان می‌دهیم. یک سری نماد ثابت هم مانند سور عمومی ($\forall$) و آنگاه ($<=$) و پرانتز باز و ویرگول و نقیض یا منطقی را بسازیم. مثلا می‌گوییم

$\forall x$ زوج است x or x فرد است

$\forall x$ زوج است x => x فرد است

حالا یک محمول به نام زوج بودن $E(x)$ و یک محمول برای فرد بودن $O(x)$:

$$\forall x \ (\neg E(x) => O(x))$$

قرار شد من برای هر محمول $A(n,k)$ قرار دهم. زوج یا فرد بودن روی یک عنصر عمل می‌کند. پس اگر در ترتیب شماره‌گذاری‌مان به زوج بودن عدد ۱۰۰ نسبت داده باشیم و به فرد بودن عدد ۸۶ ، صورت قضیه به این صورت خواهد شد:

$$\forall x \ (A(1,100)(x) => A(1,86)(x))$$

پس ما ابتدا نمادهای این زبان ریاضی را شماره‌گزاری کردیم و به هرکدام یک عدد نسبت دادیم. بعد از این می‌آید و عبارت‌ها را هم شماره‌گزاری می‌کند و به هر کدام از عبارت‌ها یک عدد منحصر به فرد نسبت می‌دهد. بعد حتی به دنباله‌ای از عبارت‌ها هم یک شماره نسبت می‌دهد و هر کدام یک عدد منحصر به فرد. به این عددگذاری می‌گویند عددگزاری گودل.

- ***پس به هر قضیه‌ی ریاضی که بتوانیم تصور کنیم، یک عدد منحصر به فرد نسبت داده است؟***

به هر عبارت ریاضی؛ اعم از این‌که اثبات بشود یا نه؛ چه آن عبارت درست باشد و چه نادرست. بعد از این کار را کرد، انقدر عبارت می‌سازد تا به یک رابطه‌ای به نام $Pf(y,x)$ برسد. به این معنا که y معادل عددی یک دنباله از عبارات است که برای عبارتی با عدد x یک اثبات محسوب می‌شود. بعد از این یک گام دیگر هم می‌ماند؛ یعنی قضیه‌ی نقطه‌ی ثابت.

قضیه‌ی نقطه‌ی ثابت می‌گوید به ازای هر خاصیتی مثل C در اعداد طبیعی، یک عبارتی مثل φ درست است اگر و فقط اگر آن خاصیت در مورد مقدار عددی آن صدق کند. حالا با استفاده از قضیه‌ی نقطه‌ی ثابت و $Pf(y,x)$ می‌توانیم قضیه‌ی گودل را به راحتی ثابت کنیم. می‌آییم خاصیت را $C(x)$ در نظر می‌گیریم که $C(x)$ برابر است با: هیچ اثباتی وجود نداشته باشد که هیچ مقدار y در $Pf(y,x)$ وجود داشته باشد. یعنی:

$$C(x)= \neg \exists y \ Pf(y,x)$$

طبق قضیه نقطه‌ی ثابت یک عبارتی مثل φ باید پیدا شود که درست است و فقط اگر اثباتی برای آن وجود نداشته باشد. حالا اگر نظریه‌ای که داریم با آن کار می‌کنیم، سازگار باشد، φ را نه می‌توان اثبات کرد و نه می‌توان رد نمود. چون اگر φ را ثابت کند، در آن صورت φ را اثبات هم کرده است و ما به تناقض رسیده‌ایم. اگر هم φ را رد کند، پس باید یک دنباله‌ی اثباتی برای آن پیدا شود که φ را ثابت می‌کند و باز به تناقض رسیده‌ایم [می‌خندد] یک کار شگفتی است.

خب اما این $Pf(y,x)$ را که می‌سازد می‌توانیم یک متد هم ارایه دهیم که قضایای ریاضی را هم ثابت کند. اگر یک گزاره‌ای مثل A اثبات‌پذیر

باشد،در این روند حتما متوقف خواهد شد. پس برای اثبات قضیا این الگوریتم را دنبال می‌کنیم.

این الگوریتم بازگشتی است و به صورت کاملا دقیقی قابل تعریف است. پس اگر قضیه‌ای اثبات‌پذیر باشد، این روند می‌تواند اثباتی برای آن پیدا کند.

اگر چه این روند می‌تواند اثباتی برای قضایا پیدا کند، اما پیاده کردن آن به صورتی که گودل گفته،غیر عملی است، چون اعدادی که گودل معرفی می‌کند، بسیار بزرگ است. مثلا شما چگونه می‌خواهید محاسبه‌ای روی یک عدد ده میلیارد رقمی انجام دهید؟ پس من آمدم و این روند را شکافتم

اما خلاقیت‌های ریاضی‌دان؛ یک ریاضی‌دان ممکن است در ذهنش یک روند تفکر داشته باشد و در نهایت، محصول این تفکر اثباتی باشد برای قضیه‌ای، اما تا این اثبات را به صورت دقیق ارایه نکند، نه قابل ارایه است و نه قابل دفاع. پس خلاقیت او در آخر مجبور است تابع قوانینی باشد که این ماشین با آن‌ها کار می‌کند.

تا ببینم گودل اصلا چگونه آن را ساخته است. دیدم که نیازی به اعداد نیست و ما می‌توانیم خود دنباله‌ی نمادها را استفاده کنیم.

- ***چرا اعدادی که گودل معرفی کرده انقدر بزرگ هستند؟***

اگر دقیق تعریف اعداد گودل را ببینیم، مشخص خواهد شد. ببینید، یک عبارت ریاضی، دنباله‌ای از عددها است. فرض کنید که این نمادها از S1 تا Sn باشند. یعنی هر نمادی از صور عمومی و پرانتز باز گرفته تا محمول‌ها و متغیرها و اعداد ثابت. گودل عدد یک عبارت را به این شکل تعریف می‌کند:

$$2^{S1}*3^{S2}*4^{S3} \dots *(P_n{}^{S1})$$

یعنی دو به توان عددِ نماد اول، ضرب در سه به توان نماد دوم و تا n امین عدد اول به توان عدد نماد n ام.

این یک عددهای بسیار وحشتناکی به ما می‌دهد. بعد از مدت‌ها بررسی دیدم می‌توانیم به جای این اعداد می‌توانیم با همان روش سمبلیک که خودمان هم با آن کار می‌کنیم. به تدریج هم توانستم یک سری تکنیک‌هایی به آن اضافه کنم. ایده‌های خیلی کوچکی هم از خیلی جاها گرفتم مثلا در نحوه‌ی نمادگذاری از word ایده گرفتم، از ساختمان داده‌ها ایده گرفتم و از مباحث مختلف ایده گرفتم و این ایده‌های کوچک را به هم چسباندم و برنامه را جلو بردم. تا جایی که الان یک روایت مناسب از برنامه دارم که در آن راحت می‌توان اطلاعات را وارد کرد و راحت با آن کار کرد.

- ***با توجه به این نظریه‌ی گودل، درواقع شما یک ریاضی‌دان را شبیه‌سازی کردید. ریاضی‌دانی که می‌توان گزاره‌های ریاضی را ثابت کند؟***

می‌تواند مشابه با یک ریاضی‌دانی که در یک گرایش ریاضی کار می‌کند، قضیه ثابت کند. حالا مشابه با یک ریاضی‌دان ادعای بزرگی به نظر می‌رسد چون ریاضی‌دان یک سری خلاقیت‌ها به خرج می‌دهد. یک کارهایی می‌کند که به نظر ما الگوریتمیک نیست و نمی‌شود برای آن الگوریتمی پیدا کرد. این ایراد به جا به نظر می‌رسد چراکه ریاضی‌دان‌ها قواعد استنتاج متنوعی به کار می‌برند. اما در منطق ریاضی ثابت می‌شود که تمام قواعد استنتاج که ریاضی‌دان‌ها از ان استفاده می‌کنند به دو قاعده و پنج اصل قابل کاهش است. این ویژگی، ویژگیی بود که من است آن استفاده کرده‌ام. این قواعد استنتاج مثل قواعد بازی است. این دو تا قاعده، یکی modus penance است که می‌گوید اگر P را داشته باشیم و P=>Q را هم داشته باشیم، Q نتیجه می‌شود. دیگری هم Generalization است یعنی قاعده‌ی تعمیم که می‌گوید اگر P را ثابت کردیم،به ازای هر x هم P صادق است. مثلا اگر برای x دلخواه ثابت کردیم که x کچل است، آنگاه به‌ازای هر x، x کچل است. این دو قاعده به اضافه‌ی آن پنج اصل، تمام قواعد استنتاج را پوشش می‌دهند. من هم در حقیقت آمدم و آن دوتا قاعده‌ی استنتاج را مکانیزه کردم و الگوریتمی نوشتم که آن قواعد به صورت مکانیزه اجرا می‌شوند. هیچ کار اضافه‌ای انجام ندادم. خب حالا برگردیم به آن بحث خلاقیت. یک ریاضی‌دان ممکن است در ذهنش یک روند تفکر داشته باشد و در نهایت محصول این تفکر یک اثباتی باشد برای چنین چیزی اما تا این اثبات را به صورت دقیق ارایه نکند، نه قابل ارایه است و نه قابل دفاع. تا آن خلاقیت‌هایی که در ذهنش هست را به صورت فرمال ننویسد. ریاضی‌دان‌های دیگر از او نخواهند پذیرفت. اما وقتی که آن اثبات و تفکر را در قالب یک فرمالیسم می‌آوریم، باید یک فرمالیسم منطقی باشد یعنی گام‌های منطقی آن مشخص باشد و گرنه قابل پذیرش نیست. این منطقی بودن گام‌های آن یعنی مجبور است که از آن قوانین استنتاج استفاده کند. حالا آیا تمام قضایایی که ریاضی‌دان‌ها می‌توانند اثبات کنند، این برنامه هم می‌تواند اثبات کند؟ بله می‌تواند اما دیر و زود دارد. من هنوز نمی‌توانم به طور قطع در مورد زمان آن نظر بدهم. بحثی هم که من با آقای دکتر بروجردیان و آقای علیزاده راجع به این موضوع داشتم، سر زمان آن بود. من ادعا می‌کنم که این برنامه موثر است و می‌تواند خیلی از قضایای باز ریاضی را ثابت کند، در حالیکه دکتر بروجردیان نظری مخالف این داشتند و می‌گفتند در یک سری مسایل ساده‌ای که تا به حال اثبات شده‌اند شاید بتواند موفق باشد اما از یک حدی به بعد انقدر مسایل پیچیده می‌شود که این برنامه کارایی خود را از دست خواهد داد.

- ***شما وقتی که این برنامه را نشان من می‌دادید، گفتید که "برنامه شروع می‌کند به اثبات کردن" این شروع کردن از کجا است؟***

مشابه همان عملی است که ریاضی‌دان‌ها انجام می‌دهند. ریاضی‌دان‌های محض طبیعت کارشان این است که تلاش می‌کنند یک چیزهای جدیدی را به دست بیاورند. حالا این چیزهای جدید واقعا به درد می‌خورد یا نه، سوالی است که مربوط به خود نمی‌دانند. می‌گویند ما کاری به این چیزها نداریم. این مساله به نظر ما جالب است و ما می‌خواهیم آن را ثابت کنیم.

•

نقطه‌ی شروعش اصول و تعاریف آن قضیه است. برنامه خلق نظریه نمی‌تواند بکند و دستگاه را هم نمی‌تواند گسترش دهد. این دو محدودیت بزرگ است اما به هرحال مشابه یک ریاضی‌دان می‌تواند قضیه اثبات کند.

•

- ***من سوالم در مورد نقطه‌ی شروع برنامه بود.***

نقطه‌ی شروعش اصول و تعاریف آن قضیه است. برنامه خلق نظریه نمی‌تواند بکند و دستگاه را هم نمی‌تواند گسترش دهد. این دو محدودیت بزرگ است اما به هرحال مشابه یک ریاضی‌دان می‌تواند قضیه اثبات کند. وقتی نظریه‌ای مهیا شد، می‌تواند تمام قضایای آن را ثابت کند. حالا آن نظریه چطور به وجود می‌آید؟ ما یک تصوری از پیرامون‌مان داریم،بعد می‌آییم آن را تجرید می‌کنیم و تبدیلش می‌کنیم به یک نظریه‌ی ریاضی. این فرایند را این برنامه نمی‌تواند انجام دهد.

- ***من باز جواب سوالم را نگرفتم، یعنی نحوه‌ی انتخاب قضایا توسط این الگوریتم برایم معلوم نشده است.***

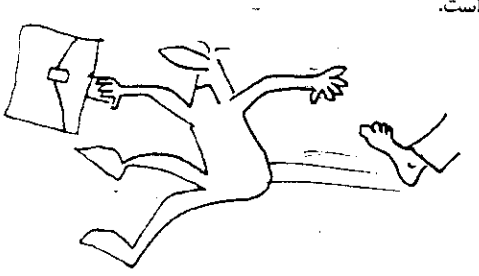
نحوه‌ی انتخاب آن در حال حاضر این است که تمام امکان‌ها را بررسی می‌کند. این کار البته هزینه‌ی زمانی زیادی دارد و من هم درصدد این هستم که با استفاده از یک سری تکنیک‌های فازی در گزینش‌های برنامه، انتخاب‌هایش را مفیدتر کنم. طرح‌هایی برای این کار دارم اما تا قسمت اول آن کامل نشده، فرصت تکمیل‌شان را ندارم.

- ***پس در شروع هم برنامه می‌آید و بر اساس اصول اولیه‌ی نظریه، تمام قضایای ممکن را بررسی می‌کند و دانه‌دانه آن‌ها را اثبات می‌کند؟***

بله، هر قضیه‌ای که به ذهنش برسد را ثابت می‌کند و این روند در زمان نامتناهی تمام قضایای آن نظریه را در بر می‌گیرد. من این را از لحاظ تئوری اثبات می‌کنم.

یک مساله‌ی دیگر هست. این برنامه‌ای که من تا الان نوشته‌ام، روشش جلو رونده (Forward)است. به این ترتیب که خودش می‌آید و هر چه حدسیات دارد، اثبات می‌کند و جلو می‌رود. از نظر تئوری من ثابت می‌کنم که این روند در زمان نامتناهی تمام مسایل را شامل می‌شود. هر قضیه‌ی اثبات‌پذیری در یک زمان متناهی در این روند اثبات خواهد شد. این زمان چقدر هست، من نمی‌توانم نظر قطعی بدهم. در واقع به علت پیچیدگی الگوریتمی که به کار برده‌ام، هنوز نتوانسته‌ام Order دقیقی برای آن پیدا کنم. تازه بحث مفید بودن قضایا هم هست. قضیه‌ی مفید مفهوم خیلی گنگی است، خیلی فازی است. اگر این مفهوم را نتوانیم دقیق کنیم، کارایی برنامه هم معلوم نخواهد شد. من مفید بودن را این در نظر می‌گیرم که این برنامه بتواند در ظرف شش ماه تمام قضایای مبانی ریاضی را ثابت کند. چرا؟ چون یک دانشجوی ریاضی با تمام معلومات قبلی خود می‌آید و شش ماه یاد می‌گیرد که استنتاج کند و مانند یک ریاضی‌دان قضیه ثابت کند. حتی این را هم از او نمی‌خواهند چراکه بسیاری چیزها را انتظار دارند که حفظ کند و جواب دهد و فقط بعضی جاها از او اثبات می‌خواهند. پس اگر این برنامه بتواند ظرف همین شش ماه قضایای مبانی ریاضی را ثابت کند، مفید است و اگر هم بتواند ظرف ده سال قضایای پیچیده را ثابت کند، مفید است. البته پیش‌بینی من خیلی کمتر از این است.

اما تفاوتی که این برنامه با یک ریاضی‌دان دارد این است که این برنامه تمام قضایا را ثابت می‌کند ولی یک ریاضی‌دان این گونه عمل نمی‌کند. ما از یک ریاضی‌دان می‌پرسیم که این قضیه درست است یا غلط؟ اگر او آن را بلد باشد، می‌گوید این اثباتش در فلان کتاب هست. اگر هم بلد نباشد، به اندازه‌ی ارزشی که مساله برایش دارد، روی آن وقت می‌گذارد. مثلا همان حدس گلدباخ نمونه‌ی ساده‌ای از این است.خیلی‌ها از این مساله دست کشیده‌اند. بعضی بعد از دو ساعت، بعضی هم بعد از هفت سال. حالا من هم یک روند پس رو (Backward) به صورت است که یک عبارت خاص را به او می‌دهیم و می‌پرسیم که ببین می‌توانی این را ثابت کنی؟ برنامه مهم در پایگاه داده‌ی خودش که دایما در حال افزایش است جستجو می‌کند. اگر پیدا کرد که با روش اثبات آن ارایه می‌کند. اگر هم پیدا نکرد، عباراتی را در پایگاه‌داده‌ی خود پیدا می‌کند که به درد اثبات آن قضیه بخورد. حالا هر کدام از آن‌ها اگر اثبات شده بود که اثبات قضیه‌ی مورد نظر را کامل می‌کند و ارایه می‌دهد. این روند دایما گسترش می‌یابد و به عقب بر می‌گردد. این شیوه به سمت این حرکت می‌کند که اثبات را برساند به پایگاه‌داده‌ی موجودش. این روند الزامه به جواب منتهی نخواهد شد اما سریع‌تر است.



تنها سوالی که در مورد این نرم‌افزار می‌ماند، کارآمدی آن است. البته یک تلفیقی هم ممکن است وجود داشته باشد که این نرم‌افزار همکار یک ریاضی‌دان باشد. یعنی یک ریاضی‌دان و یک ماشین بشینند هم فکری کنند تا یک سری قضایا را ثابت کنند. اگر این برنامه آن کارآمدی اصلی را نداشته باشد، روی این خیلی امیدوارم. من اما خودم روی کارآمدی برنامه شک ندارم.

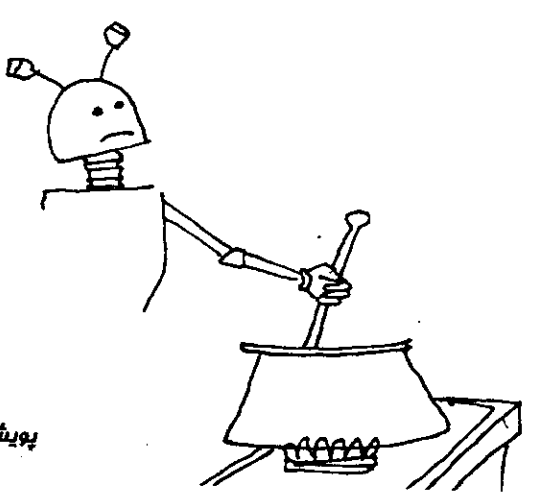
کجا باید آن را جست و جو کرد؟ از بین تلقی‌های مختلف، کدام را باید برگزید؟ "روش" داوری در بین این تلقی‌ها چیست؟ کدامین "تلقی پیشین" دیگر، این "روش" را تعیین می‌کند؟

همان طور که می‌بینید، علم بسیار شبکه‌ای تر از آن است که بتوانیم نقطه آغازی در آن مشخص نماییم. دانشمندان پیوسته مجبورند یک سری پیش فرض‌ها و تلقی‌های بررسی ناشده را بپذیرند و برپایه آن‌ها تحقیق خود را آغاز کنند. به این ترتیب در هوش‌مصنوعی نیز ما ناگزیر از یک سری مفروضات و تلقی‌ها هستیم و نمی‌توانیم آن‌ها را کنار بگذاریم. اما می‌توانیم ببرسیم که این تلقی‌ها و پیش فرض‌ها از چه عواملی متاثرند و از چه عناصری تشکیل یافته‌اند؟ تصور بیولیژیکی ما از مغزمان، قسمتی از آن را تشکیل داده است؛ فعالیت‌های منطقی و محصولات آن بحشی دیگر را؛ تلقی جامعه علمی‌و غیرعلمی از هوش نیز پاره‌ای دیگر را و سرآخر خود هوش‌مصنوعی تا به اکنون. عنصر آخر نشان می‌دهد که نه تنها تلقی ما از هوش در روش تحقیق ما درباره‌ی هوش اثر می‌گذارد، بلکه خود نتایج حاصل از به کارگیری آن روش نیز تلقی ما را دگرگون می‌کند.

پس ما به نوعی پیش فرض گرفتیم که در تحقیق درباره هوش‌مصنوعی، تلقی انسان‌ها شرط تعیین کننده است و اصولا انسان‌ها و تلقی آن‌هاست که هوش را هوش می‌کند. با آگاهی نسبت به این پیش فرض، می‌خواهیم دوباره به پرسش نخستین بازگردیم: در هوش‌مصنوعی آیا باید از روش تفسیری–هرمنوتیکی وارد شویم یا از روش علمی–تجربی؟ اما این بار می‌دانیم که هر پاسخی که برای این پرسش بیابیم، بر روی پیش‌فرض‌های اولیه‌ی ما تاثیر خواهد گذاشت.

می‌دانیم که "فهم" و "فهمیدن" سخت با هوش گره خورده است. پس بررسی آن نیز احتمالا ما را به نکات درخوری درباره هوش خواهد رساند. اما "فهم" نیز از آن ترم‌های نظری‌ای است که باز تئوری ما درباره هوش آن را معنا می‌کند. به بیان دیگر عناصری چون فهم و تصمیم و آگاهی معنای هوش را می‌سازند ولی خود هوش نیز معنای این عناصر را تعیین می‌کند. این جا نیز یک دور هرمنوتیکی بوجود می‌آید و ما باید از یکی آغاز کنیم تا به صورت دوری این دو معنای یکدیگر را پرورش دهند:

فرض می‌گیریم که فهمیدن یعنی پیروی از یک قاعده به نحوی که بتوانیم آن قاعده را بسط دهیم. اما قاعده چیزی است که اولا تخلف‌بردار باشد؛ یعنی بتوان از آن پیروی نکرد. ثانیا قاعده باید خصلت جمعی نیز داشته باشد؛ یعنی قاعده را باید چندین نفر به رسمیت بشناسند و قاعده‌ی شخصی معنا ندارد. همانطور که دیده می‌شود، این تلقی از فهم، انسان و اعتبارات انسانی را در معنای هوش دخالت بسیار می‌دهد و دیگر نمی‌توانیم بگوییم که هوش و هوشمندی در مورد انسان و غیرانسان تفاوتی



نمی‌کنند. پس با این تلقی از هوش نمی‌توانیم صرفا به دنبال پیاده‌سازی یک سری نظم در رفتار و عمل باشیم. همچنین دیگر هوش در بررسی‌های بیولوژیکی ابزارهای آن (یعنی مغز[5]) در هوش طبیعی، و الگوریتم‌های شبکه‌های عصبی در هوش‌مصنوعی، خلاصه نمی‌شود.

حال فرض خود را درباره فهم پیشتر می‌بریم. فرض این بود که فونکسیون هوش و هوشمندی، پیاده‌سازی نظم از طریق کشف مشابهت‌ها است. اما تشخیص مشابهت‌ها به چه طریقی و از روی چه چیزی فهمیده می‌شود؟

مثال زبان را در نظر بگیرید. مشابهت دو جمله «آسمان می‌گرید.» و «باران می‌آید.» بر اساس چه معیاری فهمیده می‌شود؟ جمله اول به مجاز است و جمله دوم حقیقی. حال از نظر کلمات ادا شده، امواجی که در فضا پخش شده‌اند، کش و قوس گلو و دیگر نشانه‌های بیولوژیکی جمله (که ما تا به حال شناخته‌ایم) متفاوتند. اما این دو جمله از لحاظ معنا و اعتباری که انسان‌ها بدان داده‌اند، "مشابهند". پس در کشف مشابهت در عالم انسانی نمی‌توان معنا و اعتبار را بیرون گذاشت؛ حتی بالعکس در رفتارهای هوشمندانه باید مشابهت را در معنا و اعتبارات انسانی جستجو کنیم و نه در صورت رفتارها. اگر بخواهیم دانشی بر پایه رفتارهای هوشمندانه انسان‌ها بنا نهیم، نمی‌توانیم صرفا بر صورت و شواهد فیزیکی تکیه کنیم بلکه مشابهت در هوش گویی تنها از دید همان هموکلئوس خارجی دیدنی است؛ از جایی که بتوانیم معنا و اعتبار را هم دخالت دهیم، یعنی اموری را دخالت دهیم که جنبه فیزیکی و مادی ندارند بلکه امور تئوریک و نظری صرف هستند.

خلاصه آنکه چون هوش انسانی معنادار است، نمی‌توانیم روش‌های تفسیری–هرمنوتیکی را در فهم آن به کنار بگذاریم. معنا در تمام وجود انسان رخنه کرده است؛ حتی دیدن و حس کردن او نیز متاثر از معنا است یعنی تصورات observational هم theoretical و theory laiden هستند. آن چه ما از هوش می‌بینیم، جنبه خارجی و دیدنی آن است (از پالس‌های مغز تا گفتار و رفتار) ولی این جنبه خارجی خود احتیاج به تفسیر دارد؛ تفسیری که از معناها و اعتبارات درونی و اجتماعی انسان‌ها بر می‌خیزد. هوشی بدون این تفاسیر نمی‌تواند وجود داشته باشد[6]. اصطلاحی که برای این اعتبارات و معناها در دانش هوش‌مصنوعی کنونی به کار می‌رود، general background knowledge است. به بیان دیگر آن چه به طور بی واسطه در اختیار هوش قرار می‌گیرد، بسیار ناچیز و اندک است و واسطه‌ای به نام اعتبار و معنا عموما نقش بازی می‌کند.

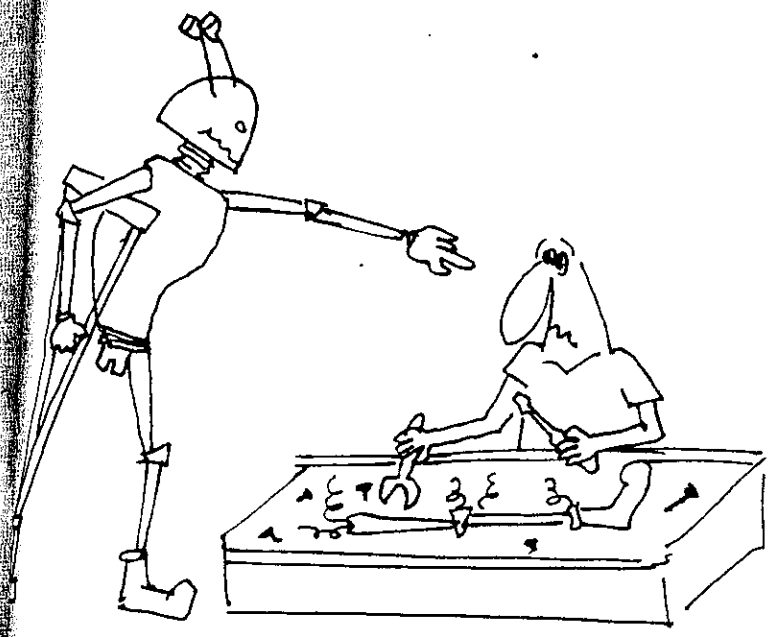
پس یک رکن اساسی در تلقی ما از هوش این است: هوش با کشف مشابهت‌ها و مغایرت‌ها گره خورده است و این کار به مدد اعتباراتی صورت می‌گیرد که بر روی پدیدارهای فیزیکی سوار شده‌اند. در واقع معنای هر عملی که انسان انجام می‌دهد با اعتباری که بدان می‌بخشد، روشن می‌گردد و ما نمی‌توانیم از ظاهر مکانیکی عمل به معنای آن دست یابیم تا بعد کشف مشابهت بکنیم یا نکنیم.

به این ترتیب روشن گشت که در مطالعه هوش‌مصنوعی و گزینش روش خود، باید ابتدا تکلیف اعتبار و معنا را مشخص نماییم. فرض کنید می‌خواهیم رباتی بسازیم که خودکشی کند. آیا اگر خود را حلق آویز کرد و

---

[5] البته با این فرض که حیات یا تصفیه شدن خون در هوش انسانی تاثیری ندارند.

[6] به خاطر داشته باشید که این "نمی‌تواند" در سایه پیش فرضهایی است که در ابتدا ذکر شد.



یا chip اصلی خود را کند و تف کرد بیرون خودکشی کرده است؟ ابدا. خودکشی عملی است که با اعتباری که انسان بدان می‌دهد عملی می‌شود؛ یعنی صورت ظاهری استاندارد شده‌ای ندارد: آن چه عمل خود کشی را عمل خودکشی می‌کند، اعتباری است که بدان داده می‌شود. ظواهر مکانیکی و بیولوژیکی آن تنها نشانه‌هایی از اعتبارهای بارشده بر آن عمل هستند. فرضا درباره کسی که در معالجه خود سستی کرده و مرده است چه قضاوتی می‌توان کرد؟ ظاهر عمل او چه چیزی را ثابت می‌کند؟ هیچ. آن چه تعیین کننده بوده است، نیت آن فرد بود که‌ایا به قصد خودکشی سستی کرده بود یا صرفا از روی تنبلی. مثال دیگر فهم زبان است. برای فهمیدن زبان نمی‌توانیم حرکت مکانیکی و بیولوپیکی لب‌ها و حنجره و سلولهای مغز را بررسی کنیم. این روش بررسی حداکثر صنعت حرف زدن را به ما می‌دهد بدون اینکه بتوانیم آن را بفهمیم[7] و نظریه، معنا و علم آن را به چنگ آورده باشیم. این روش بررسی به ما قدرت می‌دهد که حنجره یک فارسی زبان را به گونه‌ای تغییر دهیم که حرف "ر" را مانند فرانسوی‌ها تلفظ کند ولی به ما قدرت فهم زبان فارسی را نمی‌دهد. فهم زبان، وارد شد نبه قواعد آن است، یعنی اعتبارات و معانی ای را که صاحبان آن زبان از کلمات آن زبان اختیار کرده‌اند. باید میان فهم با قدرت تصرف تمایز نهاد. نکته‌ای که همین جا می‌توان به آن اشاره نمود این است که هوش‌مصنوعی بیشتر به سمت قدرت تصرف پیش می‌رود تا فهم هوش؛ یعنی بتوانیم خروجی‌های هوشمندانه‌ای بازتولید نماییم بدون اینکه بدانیم واقعا چه معنایی از آن اعمال و رفتارها اعتبار شده است. نظریه‌هایی که در مورد هوش نیز پرورش می‌یابند و روش‌هایی که بالتبع طراحی می‌گردند نیز در این جهت اند؛ یعنی در این جهت که هوش را چونان ابزاری در اختیار ما قرار دهند. توضیح بیشتر این مطلب در ادامه خواهد آمد.

اگر دید ما نسبت به هوش‌مصنوعی ابزارگرایانه است، پس ابتدا باید مشخص کنیم (شاید هم تصمیم بگیریم) که از او می‌خواهیم چه کاری برای ما انجام دهد. اگر بخواهیم یک ابزار بسازیم، مانند قیچی، دیگر اعتبارات و معناهای انسانی دخالتی نخواهند داشت؛ یعنی ما به عنوان هموکلئوس خارجی معناها و اعتبارهای انسانی خودمان را در قالب یک سری شروط و "اگر آن گاه"ها در یک ماشین تعبیه می‌کنیم تا آن ماشین بتواند در آن چارچوب عمل کند. البته از این دیدگاه، هوش شباهتی به هوش انسانی ندارد بلکه شبیه به همان قیچی خواهد بود. یک ابزار احتیاجی به دانستن معنا و اعتبار کار خود ندارد. مثال آن این گونه است: وقتی خورشید بالا می‌آید شما در بالا آمدن خورشید معنایی نمی‌بینید. نمی‌گویید که "خورشید امروز سرحال بود" یا که "باز خورشید دستور داد بروم سر کار". بالا آمدن خورشید در نظر شما trigger ی است برای شروع فعالیت‌های روزانه. اصولا تمام اعمال طبیعت همین گونه‌اند چون در نظر ما طبیعت فاقد هوش است. ما از طبیعت تنها ظاهر آن را می‌بینیم بدون آن که معنایی درونی برای آن قایل باشیم. در واقع اگر ما معنایی هم از طبیعت بفهمیم، خودمان آن را بر طبیعت بار کرده‌ایم و از خود مایه گذاشته‌ایم. اما وقتی پلیسی دستش را بلند می‌کند که "بایست!" یا چراغ راهنما قرمز می‌شود، از این صورت‌ها و توصیفات مکانیکی، "فرمان" می‌فهمیم[8] و می‌دانیم که می‌توانیم از این فرمان سرپیچی کنیم. "فرمان" دقیقا یک مفهوم درونی و اعتباری است و به همین دلیل نیز نافرمانی از خورشید برای ما بی معنی است چون آن را فاقد این حالت درونی می‌دانیم. حال برای هوش‌مصنوعی تمام دنیا مانند طبیعت بی هوش است، یعنی بی معنی و بی اعتبار. تمام عالم در نظر آن صرفا trigger محسوب می‌شوند که براساس آن "اگر آن گاه"ها عکس العمل تعیین شده را از خود نشان دهد؛ یعنی هوش‌مصنوعی تنها می‌تواند برحسب ظاهر امور قضاوت کند[9]. هوش‌مصنوعی در مواجهه با عالم انسانی و اعتبارات و معانی مربوط به آن، تنها می‌تواند در آن حدی قضاوت و داوری کند که ما برای او "رو" کرده‌ایم.

چکیده ی سخن تا به اکنون این است که تلقی ما از هوش‌مصنوعی، یک ابزار است و بالتبع روش ما در تحقیق از آن نیز ابزارگرایانه است. به این ترتیب هوش‌مصنوعی ابزار دست هوش انسانی و توسعه ی ابزاری آن محسوب می‌شود. حال هرچه ما بیشتر اعتبارات و معناهای انسانی را برای آن "ظاهر" کنیم، ابزار موثرتر و موفق تری ساخته‌ایم. اما موفق تر بودن ابزار هوش‌مصنوعی به چه معنا است؟

از دید ابزارگرایانه، هوش‌مصنوعی باید بتواند معناها و اعتباراتی را که برایش ظاهر نشده‌اند، "پیش بینی" کند[10]. اما ماشین هوش‌مصنوعی چگونه می‌تواند این کار را انجام دهد؟ فرض کنید ما اعتبارات انسانی را در قالب پنج جمله از یک سری برای او صوری نموده‌ایم: ۱،۳،۵،۷،۹. ماشین می‌تواند براساس طرح از پیش تعیین شده خود (یعنی بر اساس قواعدی پیشتر برای آن ظاهر شده‌اند) جمله ششم را حدس بزند: ۱۱. اما از کجا بداند که پیش بینی او درست است؟ نمی‌تواند، مگر این که معناها و اعتبارات بیشتری برای او ظاهر شده باشند. چراکه ممکن است سری مذکور تکه‌ای از یک سری بزرگتر بوده باشد: ۱،۳،۵،۷،۹،۱۳،۱،۳،۵،۷،۹ و جمله ششمی‌که او حدس زد در واقع جمله این سری بزرگتر بود: ۱۳. حال

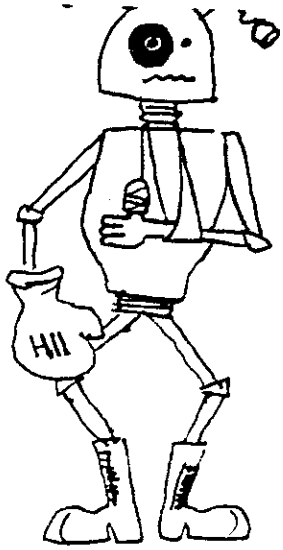
---

[7] مانند مثال اتاق چینی جان سرل

[8] یاد آوری می‌کنم که چراغ قرمز و دیگر ابزارهای اجتماعی درواقع توسعه ی فیزیکی هوش و اعتبارات انسان‌ها هستند

[9] همانند ما در مقابل طبیعت که تنها بر حسب ظاهر آن قضاوت می‌کنیم و تنها برای ظاهر آن شانیت قایلیم

[10] البته باید توجه داشت که حتی پیش بینی‌های او را نیز ما به عنوان هموکلئوس خارجی تفسیر خواهیم کرد



اگر ما این تفاوت را برای او ظاهر کرده بودیم، او نیز می‌توانست آن را بفهمد.

به این ترتیب هر چه ما کمتر مجبور باشیم معناهای کمتری را برای یک ماشین رو کنیم، آن ماشین هوشمندتر است. به هر حال ما هوش را این گونه اعتبار کرده‌ایم. قدما در تعریف فکر می‌گفتند "حرکت منظم از مبدا معلوم به سمت مجهولات". هوش‌مصنوعی هم قرار است همین کار را انجام دهد با این تفاوت که مجهولات او اعتبارات و معناهای انسانی است چون برای هوش موضوع دیگری وجود ندارد.

این ادعا به شرطی معنا خواهد داشت که پذیرفته باشیم جهان از طریق اعتبارات و معناهای خود ساخته ی ما در اختیار انسان‌ها قرار می‌گیرد و همه ی عالم را ما از طریق و با عینک اعتبارات خود می‌بینیم. چگونه؟ فرض کنید از یک ماشین می‌خواهیم که تحقیق کند آیا آب از دو تا اتم هیدروژن و یک اتم اکسیژن تشکیل شده است یا خیر. ببینید اعتبارات ما برای او چگونه تعیین کننده‌اند: روش او برای گردآوری داده‌ها چیست؟ روش او برای داوری در مورد نظریه مورد تحقیق چیست؟ همان که در پارادایم جامعه عالمان زمانه مقبول است؛ یعنی اگر او بگوید "چون ده مولکول آب را تجزیه کردم و این گونه بود، کسی از او نخواهد پذیرفت. اصلا او از اتم، هیدروژن و مولکول چه می‌فهمد؟ یعنی او از آن چه قرار است اثبات کند چه درکی دارد؟ همان را که عالمان به او گفته‌اند؛ یعنی ماشین مورد نظر اتم و هیدروژن را به منزله چیزی می‌بیند که در پارادایم حاکم علمی‌معرفی شده است. اگر هوش‌مصنوعی در زمان بیرونی مطرح بود، "ماشین‌ها آب را به منزله جسمی‌یکپارچه می‌دیدند چون پارادایم علمی‌آن زمانه این گونه حکم می‌کرد. ماشین "هوش‌مصنوعی همه چیز را "به منزله چیزی" می‌بیند چون خود انسان‌ها همه چیز را به اعتبار خاصی می‌بینند. دنیای طبیعی مستقیما در اختیار انسان‌ها قرار نمی‌گیرد و بالتبع برای هوش‌مصنوعی کل عالم مستقیما در دسترس نیست. ما آب را به اعتبار چیزی می‌بینیم که اگر هربار بر روی آتش بریزیم آن را خاموش می‌کند، اگر هر بار آن را تجزیه کنیم دو اتم هیدروژن و یک اتم اکسیژن به دست می‌آوریم و . . . هوش‌مصنوعی نیز دنیا را از طریق همین اعتبارات می‌بیند چون ما این گونه اعتبارات را برای او رو کرده‌ایم.

خلاصه آن که تلقی ما نسبت به هوش‌مصنوعی ابزارگرایانه است و آن را مانند ابزاری برای هوش انسانی خود می‌بینیم. یک ابزار نیز می‌تواند تنها با ظاهر شده‌ها کار کند. هوش‌مصنوعی به عنوان ابزار هوش ما، لاجرم با اعتبارات انسانی سروکار دارد و جهان انسان‌ها و طبیعت را از آن چشم انداز می‌بیند. پس هرچه ما بیشتر اعتبارات خود را برای او ظاهر کنیم، پیش بینی‌ها ی او نیز صائب تر خواهند بود.

از این جا می‌توانیم پی ببریم که چرا هوش‌مصنوعی این گونه د دنیای انسان‌ها گره خورده و عمدتا با آن سروکار دارد و نمی‌تواند مستقیم با طبیعت درگیر شود. انسان‌ها جهان را با عینک اعتبا معناهای خود می‌بینند و تفسیر می‌کنند؛ پس اگر هوش‌مصنوعی بخواهد ابزار موثری در این جهت باشد، باید بتواند با این اعتبارات و تفاسیر کار کند. ما جهان را می‌بینیم و یک سری جمله تولید می‌کنیم. معنای این جملات ر هم پارادایم زمانه تعیین می‌کند. سپس ما رابطه این جملات را با نظریه‌های مان (یک سری جمله ی دیگر) درباره جهان و تمم‌ها و مشابهت‌های می‌سنجیم. صحت داوری ما را نیز همان اعتبارات و معناها تعیین می‌کنند. دیدگاه ابزارانگارانه درباره هوش‌مصنوعی حکم می‌کند که هوش‌مصنوعی نیز در این چارچوب قرار گیرد تا نتایج فعالیت‌های او به کار ما بیاید.

تصویری که از هوش‌مصنوعی (تا به این جا) ترسیم گشته است، پرت شدن آن در میان هوش طبیعی انسانی است؛ که او باید از آن مایه بگیرد و در همان چارچوب نیز حرکت کند. ما انسان‌ها تامین کننده و سازنده و در عین حال زندان هوش‌مصنوعی هستیم.

حال که جایگاه و نقش اعتبارات انسانی در هوش‌مصنوعی مشخص شد، قدم بعدی این است که مولفه‌های معنا و اعتبار را برای خود "ظاهر کنیم. هوش‌مصنوعی بر اساس تشابه‌ها عمل می‌کند و گفتمی‌که تشابه را تنها می‌توانیم براساس معنا و اعتبارات انسانی درک کنیم و نه صورت ظاهر پدیده‌ها. همچنین گفتیم چون هوش‌مصنوعی به عنوان یک ابزار تنها می‌تواند با صورت ظاهر کار کند، باید به نحوی این اعتبارات را برای او ظاهر نماییم و هر چه بیشتر آن‌ها را رو کردیم، ابزار موفقت تری ساخته‌ایم و چون این ابزار می‌تواند اعتبارات ظاهر نشده را بهتر حدس بزند. اما برای این کار لازم است خود مفهوم اعتبار و معنا را بیشتر بشکافیم:

یکی از اموری که به افعال مکانیکی و بیولوژیکی معنا می‌دهد، "تصمیم" است. تصمیم بین دو فعل ظاهرا مشابه تمایز ایجاد می‌کند. همچنین تصمیم (که نهفته در هر معنایی است) باعث می‌گردد امور انسانی (اعتبارات + معانی) صددرصد قابل پیش بینی نباشند. آن چه ما در هوش‌مصنوعی می‌توانیم انجام دهیم این است که به ماشین بگوییم "هرگاه مقدمات الف مهیا شد، تصمیم ب گرفته می‌شود". اما در عا انسانی این گونه نیست. شخص ج می‌گوید "هرگاه مقدمات الف مهیا شد. من تصمیم خواهم گرفت". اگر قرار بود که شخص ج هرگاه مقدمات الف رخ می‌دادند، تصمیم ب را می‌گرفت. معنا را از عمل او ستانده‌ایم. هرگاه عمل تصمیم را "خودکار" کردیم، آن را از معنا نیز تهی ساخته‌ایم. در واقع همان گونه که در مورد هوش‌مصنوعی و مفهوم trigger گفته شد، در مورد اعمال خود به خودی همیشه یک هموکلئوس خارجی باید بیاید معنایی را به آن عمل (از خود) بدهد.

با ادامه دادن این روند بحث،مفاهیمی که برای فهمیدن هوش مور نیازاند، به صورت نمایی افزایش می‌یابند. افزایش نمایی مفاهیم خود منجر به افزایش نمایی "تلقی‌ها" می‌گردد و لذا ادامه به مقاله‌های تعدی موکول می‌گردد.

المپیاد دانشجویی ۸۱

۱- فرض کنید که دنباله‌ای از اعداد طبیعی مختلف داریم. یک زیردنباله صعودی یا نزولی از آن مجموعه‌ای است از اعداد درون دنباله‌ای که ترتیبی صعودی یا نزولی داشته باشند.

الف) الگوریتمی از $O(n^2)$ ارائه دهید که طول بزرگ‌ترین زیر دنباله صعودی یا نزولی از این دنباله را مشخص کند.

ب) الگوریتم قبلی را طوری تغییر دهید که خود زیر دنباله را نیز بدهد.

ج) ثابت کنید در هر دنباله به طول k^2+1 حداقل یک دنباله نزولی یا صعودی با طول $k+1$ وجود دارد.

۲- تعدادی دانشجو در یک سالن ورزشی در جاهای مشخصی ایستاده‌اند. دانشجوی شماره‌ی i ام در محل (x_i, y_i) قرار دارد. مربی والیبال می‌خواهد محلی را برای ایستادن او تعیین کند که مجموع فواصلش از دانشجویان مینیمم باشد. الگوریتمی از مرتبه $O(n)$ بنویسید که محل مربی را تعیین کند.

۳- n عدد داریم که دارای مقادیری در محدوده $[0,\ldots k]$ هستند. پیش‌پردازشی از مرتبه $O(n+k)$ بنویسید بطوری‌که پس از آن بتوان درخواست زیر را با $O(1)$ انجام داد:

تعداد اعدادی که در محدوده $[a..b]$ هستند

۴- یک راس برشی راسی است که با حذف آن و تمامی یال‌های متصل به آن گراف ناهمبند شود.

الف) اگر پیمایش dfs را از ند u شروع کنیم ثابت کنید که می توان تعیین کرد که u راس برشی است یا نه.

ب) آیا یک برگ یک راس برشی است؟

۵- یک DFA طراحی کنید که زبان مقابل را بپذ یرد $L(M)=\{w \mid w \bmod 3 = 0\}$

۶- گرامر مقابل عبارت‌های ریاضی را تولید می کند. اما در این گرامر ممکن است پرانتزهای اضافی نیز وارد شوند. مثلا عبارت 2+(5-3) با اینکه به پرانتز احتیاج ندارد ولی توسط این گرامر قابل تولید است. گرامر را طوری تغییر دهید که عبارت‌هایی را تولید کند که پرانتز اضافی ندارند. مثلا عبارت 2*(5+3) را تولید کند ولی 2+(5*3) را تولید نکند.

$$E \rightarrow E+T \mid E-T \mid T \mid -T$$
$$T \rightarrow T*F \mid T/F \mid F$$
$$F \rightarrow n \mid (E)$$

۷- گرامر زیر چه زبانی را تولید می‌کند؟

$S \rightarrow ABC$
$b \rightarrow DBE \mid \varepsilon$
$DE \rightarrow EDa$
$AC \rightarrow \varepsilon$
$aE \rightarrow Ea$
$aC \rightarrow Ca$
$AE \rightarrow A$
$DC \rightarrow C$

۸- یک ماشین تورینگ با سه TAPE طراحی کنید که زبان $a^n b^{n^2}$ را بپذیرد. دراین ماشین می‌توانید از وضعیت عدم حرکت نیز استفاده کنید. یعنی می‌توان در هر بار به راست یا چپ رفت و یا این‌که حرکتی نداشت.

۹- ستون فقرات تعداد ۱۰ شبکه را با ستون فقرات از نوع 10 base 5 و با توپولوژی ستاره به هم مرتبط کرده‌ایم.

الف) شکل این شبکه را رسم کنِدو اجزا لازم آن را تشریح کنید.

ب) اگر در ستون فقرات شبکه مزبور FDDI بکار بریم. شکل آن را رسم کنید و تشریح نمایید.

ج) اگر در این شبکه ۱۰۰۰ ند نهایی داشته باشیم در مورد کارایی روش‌های FDDI و 10 base 5 بحث کنید.

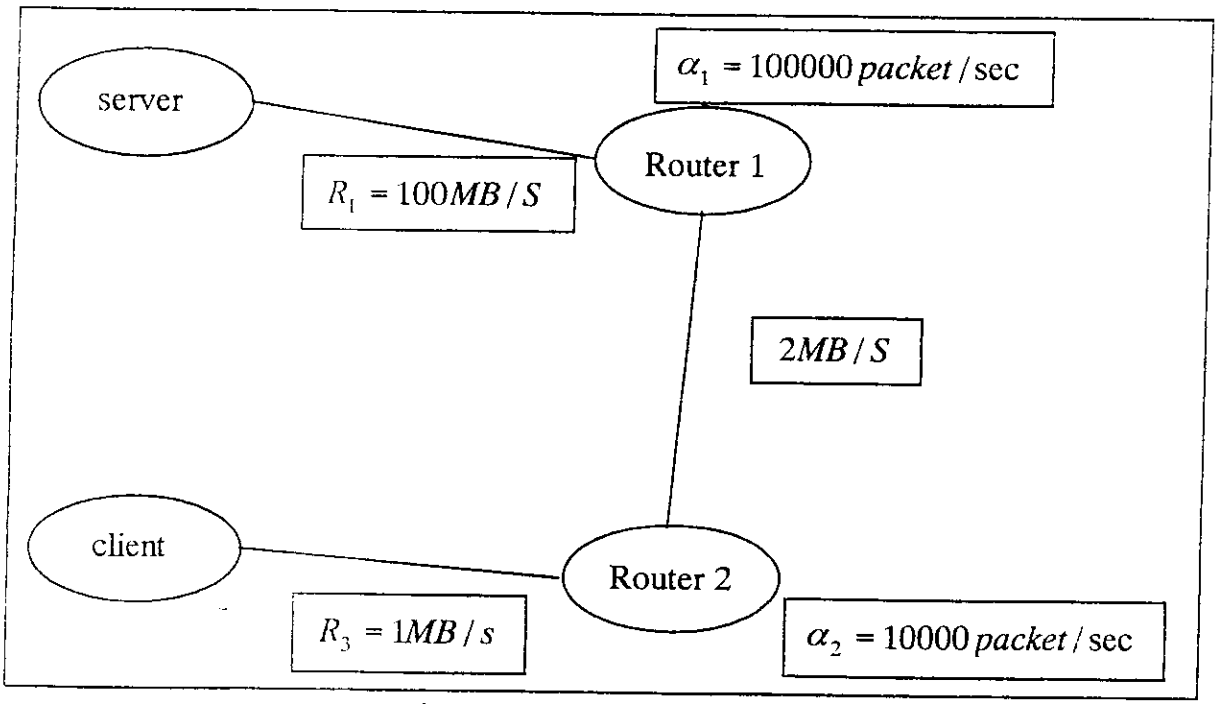
۱۰- یک کلاینت روی اینترنت درخواستی را برای دریافت فایلی تولید می‌کند که در cashe محلی وجود دارد.

الف) کلیه مبادلات و فرمان‌هایی که در پروتکل TCP رد و بدل می‌شود، مشخص و ترسیم کنید.

ب) مبادلات و فرمان‌ها را برای حالتی که فایل در cashe نباشدْ نیز ترسیم نمایید.

ج) کلیه درخواستهاو پاسخهای آن را در صورت استفاده از HTTP1.1 مشخص نمایید.

۱۱- در ساختار زیر سرور یک فایل 32kB برای کلاینت می فرستد متوسط زمان انتظار در صف هر یک از روترها $\frac{IL}{R(1-I)}$ است که Lمتوسط طول فریم، R نرخ انتقال اطلاعات و I شدت جریان ورودی است. متوسط زمان لازم برای انتقال این فایل را تعیین کنید. طول هر فریم 64 byte است و α متوسط نرخ ورود داده ها به router می‌باشد.



فلاسفه از کامپیوتر چه می‌فهمند؟

سید احسان لواسانی

از نظر آنان کامپیوترها، ماشین‌هایی هستند که محاسبات صوری را به صورت اتوماتیک انجام می‌دهند. محاسبات صوری هم از نظر آنان، یک جور عمل شبیه به استنتاج بر روی مجموعه‌ای از سمبل‌ها است؛ یعنی این سمبل‌ها را بر طبق قواعد معین و از پیش تعریف شده‌ای دستکاری می‌کنند. چند مثال از این دستکاری، حذف یا ایجاد یا تغییر این ساختارهای سمبلیک در هر مرحله‌ای از پروسه است.

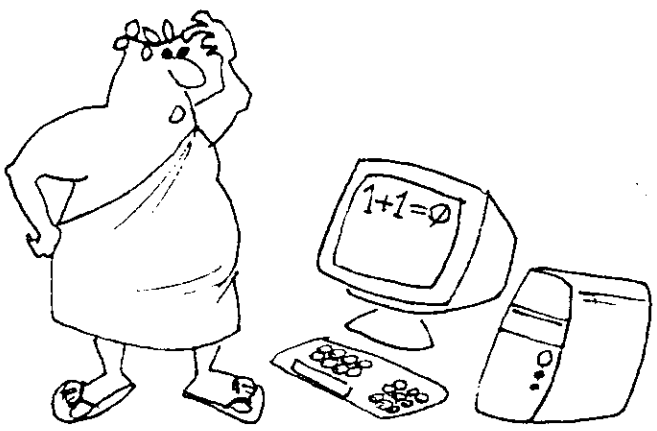
در این مرحله است که فلاسفه، مفهوم " برنامه " را هم به فهم‌شان از کامپیوتر اضافه می‌کنند: این دستکاری‌ها توسط یک برنامه هدایت می‌شود. برنامه از نظر آنان، مجموعه‌ای از دستورات است که عمل (دستکاری سمبل‌ها) خاصی را به ترتیب خاصی انجام می‌دهند. پس کامپیوترها از نظر آنان، آلتی (device)است که سمبل‌ها را دستکاری می‌کند. برنامه هم مجموعه‌ای از دستورات است که به صورت صوری (formal) برای این آلت تعریف شده‌اند تا بتواند دستکاری کردن خود را هدایت کند. از این جهت رفتار یک کامپیوتر بر اساس برنامه‌های آن قابل پیش‌بینی است.

حالا اگر دستورات یک برنامه برای انجام یک دستکاری، مستقیما بر دستورات پایه‌ی ماشین بنا نشده بود، باید به آن دستورات تفسیر شود؛ یعنی تبدیل به برنامه‌ای بشود که دستورات مستقیما منطبق بر دستورات پایه‌ی ماشین است. در این جا هم مفهوم کامپایل کردن را به فهم خود اضافه کردند. این تفسیر دستورات آنقدر ادامه می‌یابد تا بالاخره بتوانیم دستورات را توسط یک سری عمل‌های الکترونیکی در سخت‌افزار ماشین انجام دهیم. پس هر وقت که یک فیلسوف جلوی مانیتور یک کامپیوتر می‌نشیند و با آن سر و کله می‌زند، در این خیال است که دارد با یک ماشین مجازی ور می‌رود و این ماشین مجازی نهایتا (به قول خودشان، در آخرین تحلیل) خود را تفسیر می‌کند به عمل‌های واقعی که بر روی یک سخت‌افزار واقعی اجرا می‌شوند.

در این مرحله از فهم است که فلاسفه ذوق‌زده می‌شوند؛ چراکه به این صرافت افتادند که تمایز سخت‌افزار-نرم‌افزار می‌تواند مساله‌ی سنتی ذهن-روح را روشن کند (مساله‌ای که سیصد سال است ذهن‌شان را آزار می‌دهد) و به نوعی به این دوگانگی پایان دهد.

از یک طرف فلاسفه‌ای که گرایشی به ذهن دارند، سعی می‌کنند مغز انسان را به همین آلت دستکاری سمبل‌ها و رفتار آن را به یک برنامه مدل کنند و ادعا کنند که رفتار انسان، برنامه‌ای است که در یک سخت‌افزار عصبی پیاده‌سازی شده است. بعد هم به فهم خود، هوش مصنوعی را اضافه کنند و بگویند اگر ما هم بتوانیم برنامه‌ی درستی بنویسیم، قادر خواهیم بود که این رفتار (یعنی هوش انسانی) را شبیه‌سازی و بازتولید کنیم. چنین ادعایی بر این پیش‌فرض بنیان شده است که توانایی‌های ذهن انسان را می‌توانیم توسط رفتارهای بیرونی او یا ارجاع به یک سری حالات (state) و پروسه‌های درونی، تفسیر کنیم.

البته وقتی که فیلسوف یا کامپیوتردانی، چنین کاربردی از کامپیوتر را در نظر گرفت، تصور او از خود کامپیوتر و تئوری محاسبات هم متفاوت خواهد شد. از نظر او: پردازش P وقتی معنی می‌یابد که پردازشگر Q سمبل‌های خاصی را (یعنی مجموعه‌ای از سمبل‌ها که پردازش P بر روی آن تعریف شده است) دستکاری کند. اما دستکاریی که پردازشگر Q بر روی پردازش P انجام می‌دهد، خود یک پردازش دیگر است به نام P* که نسبت به پردازش P، یک پردازش درونی محسوب می‌شود. پردازش P* بر روی مجموعه‌ی سمبل‌های Q تعریف شده است و توسط یک پردازشگر درونی Q درون مجموعه‌ی سمبل‌های خودش دستکاری می‌شود. پس از نظر آن فیلسوف یا کامپیوتردان، هر پردازش تشکیل شده است از یک سلسله مراتب از پردازشگرها و مجموعه‌ای از سمبل‌ها که آن پردازشگرها بر روی آن عمل می‌کند. این سلسله انقدر جنبانده می‌شود تا به سطح ماشین می‌رسیم که مجموعه‌ی سمبل‌های آن اجزای الکترونیکی و واقعی آن است. به این ترتیب، عملیات پیچیده‌ی سطح بالا، بالاخره باید به یک سری عملیات ساده‌ی ماشین تبدیل شوند.



وقتی آن کامپیوتردان یا فیلسوف چنین تصوری را از کامپیوتر پرورش داد، می‌تواند بگوید که رفتارهای به ظاهر هوشمند انسان را هم می‌توانیم تصویر بزرگ‌تر مجموعه‌ی فعالیت‌های پردازشگرهای ساده و احمقی در نظر بگیریم که پردازش‌های ساده و غیرهوشمندانه‌ای را انجام می‌دهند. به این ترتیب، هوش و شعور در سطح زنده‌ی انسانی آن، یک ماشین مجازی است که بر روی سخت‌افزار عصبی پیاده‌سازی شده است.

البته هستند فیلسوفان و کامپیوتردان‌هایی که با این ایده مخالفت می‌کنند. یک استدلال آن‌ها نیز این است که معنای سمبل‌هایی را که کامپیوترها دستکاری می‌کنند، ما (موجوداتی که دارای شعور در سطح زنده‌ی آن هستیم) باید به آن بدهیم و معنای آن‌ها از دل خود کامپیوتر در نمی‌آید در حالیکه معنای سمبل‌هایی که خودمان استفاده می‌کنیم، از دل خودمان در می‌آید. البته این حرف بر می‌گردد به این که انسان را جزئی از کل در نظر بگیریم یا یک کل که کل است و از این جور چرندیات دیگر که از حوصله‌ی این مقاله بیرون است.

بس همانطور که دیدید (یا اگر دقت می‌کردید می‌توانستید ببنید) تعداد زیادی از ایده‌های کامپیوتر ابتدا توسط منطقیون پرورده شده است و به همین جهت هم فلاسفه از کامپیوتر بیشتر منطق می‌فهمند. در زمینه‌های دیگر هم فلا[illegible] که دانشمندان علوم کامپیوتر، خصوصا آن‌هایی که [illegible] مصنوعی کار می‌کنند، پیوسته به جاهایی می‌رسند که [illegible] انحصار فلاسفه بوده است مثل تئوری دانش، فلسفه ذهن و [illegible] البته نمی‌تواند غلط باشد. چراکه هر دانش به درد نخور و روی هوا اول جزو فلسفه است اما به محض این‌که چیز محصل و مشت پر کنی از [illegible] متعلق به یکی از علوم می‌شود. تئوری دانش و فلسفه‌ی [illegible] از این قاعده مستثنا نبودند.

البته فلاسفه از کامپیوتر [illegible] تئوری هم می‌فهمند. مثلا این که کامپیوترها حاصل تلفیقی از توسعه‌ی دو رشته، هر کدام با یک تاریخ طولانی هستند. اول توسعه‌ی ماشین‌هایی که پروسه‌های مختلف فیزیکی را مکانیزه می‌کنند. و دوم ماشین‌هایی که عملیات انتزاعی بر روی چیزها (entities) انتزاعی [illegible] انجام محاسبات روی اعداد. اما اگر بخواهیم فهرست جامع [illegible] تهیه کنیم، البته از ظرفیت یک مقاله فراتر است.

کلمات قصار کامپیوتری

Who's General Failure & why's he reading my disk?

If debugging is the process of removing bugs, then programming must be the process of putting them in.

Real programmers don't document. If it was hard to write, it should be hard to understand

Computers make it easier to do a lot of things, but most of the things they make it easier to do don't need to be done.
- Andy Rooney

Only wimps use tape backup: *real* men just upload their important stuff on ftp, and let *the rest of the world mirror it.*
- Linus Torvalds

He who refuses to do arithmetic is doomed to talk nonsense. *- John McCarthy*

All models are wrong. Some models are useful.

Nothing is destroyed until it is replaced.
- Auguste Comte (1798-1857)

My crystal ball is fuzzy
- Lotfi Zadeh (inventor of Fuzzy Logic, when asked to predict the future)

Computers are useless. They can only give you answers.
- Pablo Picasso

Cannot find REALITY.SYS. Universe halted.

Computers make very fast, very accurate mistakes.

My software never [illegible]. It just develops random features

As a computer, I find your faith in technology amusing.

...File not found. Should I fake it? (Y/N)

All computers wait at the same speed.

سرمقاله؟

سولماز مسعودیان

اینان از سرمقاله خالیست و مدیرمسوول را یارای نگاشتن نیست. او را چه شده است که قلم بر زمین نهاده خداوند داند و بس. آنقدر دانم که وجدانمان را دست بر دست گذاردن و هیچ نکردن گران آید. که این رسم پویشی بودن نباشد که پویش بی‌سرمقاله به زنبور بی‌عسل ماند! اینک چه بنگاریم که مدیرمسوول را خوش آید و وجدانمان نیز آسوده گردد؟

«خاطرمان است آنک که تارد به دانشگاه مشرف شده بودیم، پویش، بسیار به ما چشمک می‌زد و وسوسهٔ تحریریه‌ای شدن در جان ما فتاده ما را رها نمی‌نمود. و آن گرامی پویش‌یان درب‌ها را بسته پچ‌پچ‌ها می‌نمودند تا آنکه پویشی زاده شود و کس را زهره نبود که در بگشاید. ما که ندانستیم. شاید پویش را مصلحت چنان بود. و هنگام که در زمرهٔ پویش‌یان در آمدیم و با پویش‌یان قدیم وداع گفتیم، بر آن شدیم میان «شوق پویش گشتن» و «پویشی گشتن» فاصله‌ها، همه بر چشم تا به هر که را هوسی در سر است که از ما شود گوییم بسم‌ا... و ندانستیم که گاه، تعبیرها دیگر گونه است.

بگذریم، این مجمل، مجال گلایه نیست. ختم سخن این‌که: این دفتر از آن تو رایانه‌ایست، تا روزی اگر ناگفته‌ای اندر گلویت عقده گشت که مجالی از برای نوشتن می‌جست، برگی سپید از آن در انتظار قلمت باشد والسلام»

گویی خاطرمان اندکی آسوده گشت. این‌که کو آن شهامت که این نگاشته را به مجلس تحریریه بریم. ترسم که مدیر مسئول ما را از دبیری بخش فرهنگی عزل نماید. وجدان‌مان آسوده گردد بس است. از عزل گشتن چه باک!

فیلم شناسی

ح ر خ ۱۷۹ای

مسئولین آموزش : دو زن

ورودی جدید : الو الو من جوجو ام

اردو : سفر قندهار

اعضای پویش : جنگجویان کوهستان

سلف : بدون شرح

سال بالایی : نسل سوخته

درخواست نمره : می خواهم زنده بمانم

فارغ التحصیل : دیوانه از قفس پرید

دانشجوی دودر : مرد نامرئی

شب تحویل پروژه : بای سیکل ران

جلسه امتحان : در پناه تو

نمره : کیمیا

بوفه : دستفروش

کلاسهای بعد از ظهر : بعد از ظهر نحس

دانشجوی خوابگاهی : سالهای دور از خانه

کپی کردن تمرین : دو فیلم با یک بلیط

ازدواج دانشجویی : عروسی خوبان

سایت : ساعت شلوغی

دانشگاه : آرزوهای بزرگ

دستشویی : بهشت پنهان

استاد دروس مبانی ، ساختمان داده ، ذخیره و بازیابی و طراحی الگوریتم : مردی که زیاد می دانست

انتشارات : خانه خلوت

نگهبانی : سه تفنگدار – سه کله پوک

سر کلاس درس : زمانی برای مستی اسبها

لوازم دانشکده : ارثیه فامیلی

روزهای امتحان : و اینک آخر الزمان

ACM: این چند نفر

Robo Cup: بر باد رفته

استاد راهنما : دردسر زیادی

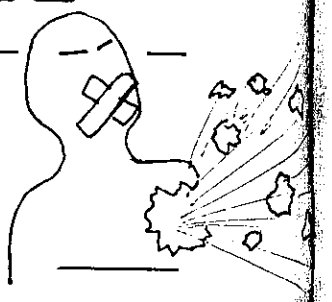
انتخاب واحد : ۱، ۲، ۳ سگک کفشتو ببند

دانشجوی ممتاز : who am I ?

دانشجوی مشروط : گامهای معلق لک لک

مسئول کارپردازی : گلادیاتور

روزگار ما



امضا محفوظ

از تاریخچه‌ی پویش که پر بود از انتقادات، نتیجه می‌شود که پویش به نوعی تابلوی آزاد این دانشکده بوده است؛ اما شاید (شاید؟!) این شماره آخرین شماره‌ی پویش باشد و مجله‌ای که حدود شش ماه مطالب آن آماده بود، به دلایل -یا بهتر است بگوییم عللی- از انتشار باز ماند. قصد از نوشتن، بررسی و ریشه‌یابی این مسایل است تا شاید کسی خواند و کسی فکر کرد.

◆ مشکلات از آن‌جا آغاز گشت (شاید هم کمی قبل از آن) که ما احساس کردیم کار طاقت‌فرسای انتشار مجله بر دوش چندین نفر خاص است و از آن میان هم فقط چهار یا پنج نفر همکاری نزدیک دارند. از طرفی با توجه به عدم لطف فراوان دانشکده مبنی بر همکاری مالی یا لااقل چاپی (مگر به شرط رویت کلیشه قبل از چاپ) وظیفه‌ی این افراد بسیار سنگین‌تر شد. پویش مایه‌ی افتخار دانشجویان و دانشکده بود و با گذشت ۱۶ شماره از آن، بی‌توجهی و بی‌میلی، همه‌ی آن را زمین زد.

◆ مساله این‌جا است که پویش در این میان تنها نیست. این بی‌توجهی در بسیاری دیگر از امور دانشکده وجود دارد. از مسابقات Robocup و ACM گرفته تا سطح علمی دانشجویان. البته باز دست دانشجویان و مخصوصا کادر فعال دانشکده‌ی ریاضی درد نکند که لااقل نگذاشتند پرچم پر افتخار پلی‌تکنیک در مسابقات جهانی Robocup بر زمین بماند. بچه‌های ACM هم کم نگذاشتند، هر چند به عللی از رفتن به مسابقات باز ماندند[1]. درگیری‌های بی‌موردی که در این یکی دوسال بین دانشکده‌ی ما و دانشکده‌ی ریاضی درگرفت، مسؤولان را از پرداختن به اصل بازداشته است و هنوز این پرسش باقی است که دانشکده‌ی ما که تقاضای رشته‌ی علوم کامپیوتر را دارد، آیا توانسته است به ۵۰ ورودی تازه‌ی هر سال سرویس مطلوب را ارائه دهد؟ گو این‌که آن طرف دعوا نیز چنین توانایی‌ای را ندارد. در این میان آن‌که ضرر کرد، دانشجویانی بودند که از گرفتن دروسی چون نظریه‌ی گراف محروم شدند چون سطح ارائه‌ی آن پایین است!!

◆ از وقتی که مسابقات خارجی بوده، مشکل اعزام را هم داشته‌ایم؛ پس چرا مسوولان ما فکری برای این قضیه نکرده‌اند؟ این در حالی است که ما در فرستادن تیم دو نفری ربوکاپ مانده‌ایم و صنعتی شریف گروه ۴۰ نفری اعزام کرد و مگر امیدها به تیم‌مان نداشتیم؟ دیگر دوران توهم توطئه گذشته است.

◆ وقتی پای درد دل بچه‌های شورای صنفی می‌نشینیم، یکی می‌گوید "در همه‌ی دانشکده‌ها بودجه‌ای برای شورا تصویب کرده‌اند ولی ما همچنان به دنبال یک H.D.D. می‌گردیم بلکه P133 را به سلامتی راه بیندازیم." یا بحث بر سر این است که چرا عکس‌ها را ۲۵ تومان گرانتر نفروشیم تا پول فیلم آن هم درآید؟

◆ گروه‌های فراوانی از دانشجویان تاکنون قصد به وجود آوردن محفل‌های علمی داشته‌اند که از جمله‌ی شکست‌ها می‌توان به گروه روبات‌های مین‌یاب و گروه شبیه‌سازی نام ببریم.

◆ ارائه‌ی دروس محشر است. البته مشکلاتی نظیر کمبود استاد قابل درک است اما آن‌چه درکش مشکل است، ادعای ما برای رشته‌ی جدید IT است؛ از قرار سال آینده، ۲۰ ورودی جدید دیگر هم اضافه خواهند شد. پس روی صندلی‌ها جمع و جورتر بنشینید و اگر استاد درس ریاضی۱، مهندسی نرم‌افزار درس داد خیلی تعجب نکنید.

◆ با توجه به مسایل فوق چقدر روا است که بگوییم دانشجویان ما فعال نیستند؟ بچه‌ها تنبل شده‌اند؟ شرایط سردی در دانشکده بوجود آمده است. به خاطر دارید اردوی پیش‌دانشگاهی‌ای را که ریاست دانشکده گفت "دانشکده‌ی ما در سطح دانشگاه اول است"؟ دانشجویان از نظر علمی تضعیف شده‌اند، گروه‌های علمی انحصاری، عدم پشتیبانی، مرگ گروه‌های صنفی- فرهنگی ... دانشجویان! شما هم باید تصمیم بگیرید که ریاست دانشکده چه کند، چگونه انجام دهد و تا کی ... باید بیشتر گفت، بیشتر نوشت، بیشتر خواند ...

تذکره الاساتید

آن سید بزرگوار، آن ملقب به مهندس در تمامی اعصار، آن حلال مشکلات هر سایت، آن مرجع بایگاه‌های داد، علامه سید مهدی حسینی‌نژاد، ادام الله ظله. و او را در تدریس فصاحتی خاص بود، بر نمطی که جماعت چرتیان کلاس هم درس او را همی‌فهمیدندی چون مجبور بودندی کتاب را به تفعات بخوانندی.

نقل کنند که در دانشکده، همگان را شوقی عظیم بود برای قرار گذاشتن با او (اطال الله عمره). علت را جویا شدند، طلاب همی‌گفتند که این استاد را نظمی عجیب است در قول و قرار، چراکه اگر گوید در فلان ساعت مشکلی هست و آمدن در تقدیر نمی‌باشد، همه را یقین حاصل گردد که او خواهد آمد و اگر همی‌گوید که حتما می‌آیم، حتم بدان که نخواهد آمد و بدین ترتیب نظم برقرار همی‌گردد.

گویند چند طالب علم در مکتب درسش مرتکب گناهی عظیم شدند و بیش از حد مجاز در غیبت اوفتادند. این جنایت، آن سید بزرگوار را بس گران آمد، الساعه امرشان فرمود که تنی صد ساعت کار اجباری در دانشکده همی‌کنند، تنبیها، تا عبرت همی‌باشد مر آیندگان را.

دو سوال اخلاقی

سوال اول: اگر شما زنی را بشناسید که حامله است و ۸ فرزنده هم دارد (که سه‌تایشان کر، دو تا کور و یکی عقب افتاده ذهنی است) و به بیماری سیفلیس هم مبتلاست، آیا به او حق می‌دهید بچه‌اش را سقط کند؟
قبل از پاسخ به این سوال، سوال دوم را هم بخوانید.

سوال دوم: فرض کنید در یک کشور اروپایی زندگی می‌کنید و زمان انتخابات ریاست جمهوری است. سه نفر با خصوصیات زیر کاند هستند:

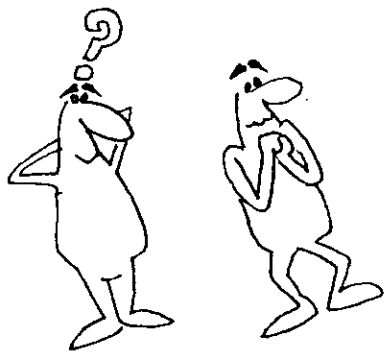
کاندیدای الف: او با سیاستمداران فاسد همکاری می‌نماید، با داشتن دو معشوقه به همسرش خیانت می‌کند، اغلب سیگار می‌کشد و ۱۰ بطری مشروب در روز می‌خورد.

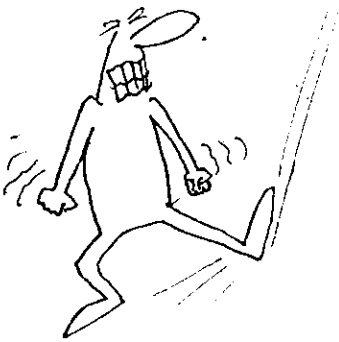
کاندیدای ب: او دو بار از پارلمان اخراج شده، هر روز تا ظهر می‌خوابد، در دوران دانشجویی تریاک مصرف می‌کرده و هر روز عصر چهارم لیتر ویسکی می‌خورد.

کاندیدای ج: او یک فرمانده نظامی پرافتخار بوده، یک گیاهخوار است، سیگار نمی‌کشد و گاهی یک لیوان آب جو می‌خورد. ا همسرش خیانت نکرده.

شما به کدام یک از این سه کاندیدا رای می‌دهید؟

اول تصمیم بگیرید و بعد جواب را در پایین این صفحه بخوانید.





کاندیدای الف فرانکلین روزولت است
کاندیدای ب وینستون چرچیل است
و کاندیدای ج آدولف هیتلر است.

و ضمناً، اگر به سوال اول درباره‌ی سقط جنین پاسخ مثبت داده‌اید، در این صورت شما نیهون را کشته‌اید!

جستاری از "میشل دِ مونتن"

Mishel de Montaigne (1533-1592)

این ترجمه به میثم افراز تقدیم می‌گردد.

"داوری" اسباب آماده و مناسبی برای هر موضوعی است و بدان می‌توان دستی در هر چیزی داشت. به این دلیل است که من در این جستارها هر فرصتی را مغتنم می‌شمرم. گر چه هیچ دور نیست که موضوعی باشد که من به خوبی درنیابم. با این حال تلاش می‌کنم که صدای آن را از دور دست هم که شده، بشنوم و آنگاه که بحر معانی را برای قامت خود ژرف می‌یابم بر ساحل می‌ایستم. این آگاهی که انسان نمی‌تواند پیشتر رود؛ یک اثر از این پرهیز است، آری! یکی از آن‌هایی که بسیار موجب سربلندی‌اند. یک هنگام در اطراف یک موضوع پوچ و بیهوده بدان جهت که شاکله‌ای فراهم آورم و آن را تقویت کنم؛ تلاش می‌کنم تا مطلب را دریابم؛ و به هنگامی دیگر آن مطلب را در موضوعی مستقل به کار می‌برم. موضوعی که هزاران مرتبه دستمالی شده است، آن‌چنان که نادر امکانی است که شخصی، چیزی تنها از خویشتن خویش بنمایاند. راهی چنین لگدخرده و فرسوده که هر که در آن پای می‌نهد، لاجرم قدم در جاقدم دیگری می‌نهد. در یک چنین احوالی این کار "داوری" است که از میان هزاران مسیر، آنی را که به نظر می‌رسد بهتر است، برگزیند. این کار "داوری" است که نشان می‌دهد این یا آن، بهتر است.

من موضوع خود را فعلا رها می‌کنم و به موضوعی که آن بانو به من عطا کرد می‌پردازم. این هر دو [موضوع] برای من یکسانند، من هرگز تشخیص نمی‌دهم که به سمت کدام یک روم، چراکه هیچگاه تمام آن‌چه هست را نمی‌بینم و نه حتی کسانی که به تاکید وعده می‌دهند که آن‌ها را به دیگران نشان می‌دهند، چنینند. از میان صد چهره و صد بخشی که هر چیز دارد من یکی را بر می‌گزینم. به یک گاه تنها به مرور آن می‌پردازم و به هنگام دیگر پوست را می‌خراشم و گاه تا استخوان پیش می‌روم؛ زخمی بر جای می‌گذارم نه چندان فراخ، بلکه تا بدان میزان که می‌توانم عمیق، و آن‌چه بدان سخت وسوسه می‌شود آن سان که برتو نوی را که در آن می‌یابم، در دست گیرم.

آیا خویشتن را کمتر می‌شناختم. شاید می‌بایست خطر می‌کردم و چیزی را تا انتها بررسی می‌کردم و فریب‌خورده ناتوانی خویش می‌شدم. اما کلمه‌ای این‌جا می‌پراکنم و آن‌جا دیگری را. الگوها به قطعات متعدد خرد می‌شوند و بدون طرح‌ریزی پراکنده می‌گردند. از آن‌جایی که تعهدی نسبت به آن‌ها ندارم، در قبال آن‌ها مسؤول نیستم و مجبور نیستم خود را همواره در کنار آن‌ها نگاه دارم و این مغایرتی با آزادی فردی و کامرانی من نخواهد داشت. خود را تسلیم شک و عدم اطمینان می‌کنم و نیز تسلیم روش حکمرانی خویش: تجاهل.

هر حرکت‌مان ما را مکشوف می‌کند. همان شخصیت "سزار" که او را چنان در راهبری و فرماندهی "نبرد فارزالیا" شهره و نام‌آور کرد، در عشق ورزیدن او نیز رخ می‌نماید. کسی که اسبی را بّه داوری و برانداز می‌نشیند، تنها به مشاهده‌ی خرامش آن اکتفا نمی‌کند؛ بلکه این داوری را به واسطه‌ی هر لحظه رفتار و حتی ایستادنش در اصطبل صورت می‌دهد.

در میان رفتارهای شخصیتی مراتب پَست‌تر و پایین‌تری نیز وجود دارد. هر آن‌کس که معشوق را در سطوح پایین‌تر، آن‌چنان خوب و زیبا که در مراتب بالاتر می‌بیند، نبیند؛ هرگز "او" را به کمال در نمی‌یابد. اتفاقا آنگاه که "او" تنها و تنها گام‌های خود را در می‌نوردد، بهتر نمایانده می‌شود. نسیم علاقه و احساسات، "او" را در پروازهای رفیع‌تر، سخت‌تر در بر می‌گیرد. و از این رو، "او" در هر مورد مشخصی خود را به تمامی به استخدام می‌گیرد و تمام خصوصیات خود را به کار می‌بندد و هیچگاه بیش از یک امر را بروز نمی‌دهد و این به طبع آن امر نیست بلکه مطابع شخصیت اوست. از قضا این امور به خود بستگی دارند. به شرایط، وزن و اندازه‌شان. اما هر آنگاه که ما آن‌ها را به درون خود وارد می کنیم؛ وجود ما آن‌ها را به گونه‌ای شکل می‌دهد، که "او" خوشایند گردد. مرگ برای "سیسرو" دردآورد است، برای "کاتو" آرزوست و برای "سقراط" تفاوتی ندارد. سلامت، هوشیاری، تفوق، دانش، مکنت و زیبایی و اضداد آن‌ها همه خود را به محض ورود به ما برهنه می‌کنند و ردای جدیدی به سبک و جنس دیگری، از روح ما بر تن می‌کنند. به چه رنگ: سبز، قهوه‌ای، تیره، روشن. به چه کیفیت: تلخ، شیرین، عمقی، سطحی، به گونه‌ای که "او" را هرچه بیشتر دلپذیر کند و رای هیچ کدام شکل و معیار و فرایندی یک انداز و معمول مقرر نشده است. هر کس خسرو قلمرو خویش است. بیش از این خود را درگیر حواس بیرونی اشیا نکنیم. این در اختیار ماست که چه میزان از آن‌ها را در خود بگیریم. سلامت و علت به هیچ چیز جز "خود" ما بستگی ندارد. این آن چیزی است که وعده‌ها و نویدهای ما بدان بستگی دارد و نه به آینده. معشوق هیچ اراده‌ای به شخصیت و حالات ما ندارد. برعکس این خصوصیات ما هستند که چهره‌ای از "او" به سلسله‌ای از خویش، نقش می‌زنند و "او" را در قالب خویش می‌ریزند.

ترجمه از م.ع.*

---

* روح آن کسی مسرور است
که جاودانگی نامش را طلب نمی‌کند.
(میشل دِ مونتن)

site
مژده: نمرات ۱۳ ترم قبل
فردا اعلام می شود!
فعالیتهای علمی

معلقات سبع

بادیه چونان مدرسه‌ای بود که شاعران جاهلی در آن پرورش می‌یافتند. بادیه همواره مهبط وحی شعر بود. شعر جاهلی نیز بیشترش مولود فطرت بود و بداهت. شعر را در آن‌جا مقامی رفیع بود و تاثیری بسیار. شعر ترجمان احساسات افراد و زبان قبیله و طومار اخبار و سرگذشت‌های آنان بود. شاعر پیشوای آنان در زمان صلح و قهرمان آنان در زمان جنگ بود. هر قبیله می‌کوشید پیش از داشتن پیشوا، شاعری داشته باشد. معلقات قصایدی است طولانی، از زیباترین اشعار جاهلی که به دست ما رسیده است. این قصاید اعجاب اعراب را برانگیخته بود. آنسان که با آب زر نوشتند و بر پرده‌های کعبه آویختند" (از مقدمه‌ی کتاب)

◄عرب جاهلی را آدمی می‌دانم که با خودش رو راست است. آدمی که به خاطر هیچ چیز به خودش حد نمی‌زند و خودش را محق می‌بیند که خوش باشد. وحشی‌گری از خودش بروز می‌دهد ولی قابل ترحم است چون می‌تواند از وحشی‌گری خودش مظلومانه (!) شعر بگوید:

(معلقه‌ی امرء القیس/۱۶و ۱۷) *به سان دیدار تو چه بسا شب هنگام به دیدار زنانی آبستن و زنانی شیرده رفته‌ام و زن شیرده را از کودک یکساله‌اش که هنوز مهره‌های افسون‌خوانان بر گردن داشت، باز داشته‌ام. چون کودک می‌گریست، با نیمی از پیکر خود به سوی او می‌خزید، که نیمی دیگرش را یارای گردیدن نبود.*

◄ اصلا شعرش یک جوری وحشی است، هر چند که انسانی (لطیف) هم هست:

(معلقه‌ی امرء القیس/۴۳) چه بسا مدعیان کینه‌توز، با همه‌ی ملامتگریشان در عشق تو، بی‌دریغ پندم دادند تا از این عشقم باز دارند و من دل از تو نبریدم.

◄خط سیر تفکر و دلیل آوردنشان را برای رسیدن به میل‌شان را به خوبی می‌توان دید:

(معلقه‌ی طرفه بن العبد/۴۳) هان ای ملامتگر که مرا از کارزار می‌ترسانی و از نوشخواری بازم می‌داری، اگر من از این دو احتراز جویم، آیا تو مرا عمر جاویدان خواهی داد؟

◄ فکر کردم که شعر بستار انتقال تفکر بوده و نغز بودن شعر هم به معنای ظریف‌تر بودن تفکر است! نه؟!

◄ عجیب است که مدح خدایان در شعرشان نیست ولی زن و شتر و گاه خیلی مطرح هستند.

◄این ابیات خیلی عجیب‌اند. مضمونی دارند که خداپرستان بسیار تکرار کرده‌اند:

(معلقه‌ی طرفه بن العبد/۷۶-۸۰) *اگر سر و کار من در این ماجرا با سروری دیگر بود، غمم را تسکین می‌بخشید یا چند روزی مهلتم می‌داد.*

ولی دریغا که سرور من مردی است که در همه حال بر من ستم می‌کند. حتی آن هنگام که دست نیاز به سویش دراز کرده‌ام و زبان به سپاسش گشوده‌ام و رهایی خویش می‌خواهم، چنان بر من سخت می‌گیرد که بیم آن است که جان تسلیم کنم.

(من خویشاوندی ندارم یا احساس خویشی با کسی نمی‌کنم و گرنه می‌گفتم) سوز ستم خویشاوندان از زخم تیغ هندی دردناک‌تر است.

مرا با خلق و خوی خود واگذار. من همواره سپاسگزار توام هر چند بر دامنه‌ی کوه ضرغد در خانه‌ای دور دست باشم.

◄ به نظر من این شعرها اصلا نسبتی با جهل ندارند.

(معلقه‌ی زهیر بن ابی سلمی/۲۶ و ۲۷) *اندیشه‌ی ناپاک خود را از خدای در دل پنهان مسازید که هر چه را در دل پنهان دارید، خدا به آن آگاه است.*

گناهکاران را از عذاب خداوند رهایی نیست. یا اعمالشان را در نامه‌ای ثبت می‌کنند و برای روز رستاخیز می‌گذارند یا آن که خداوند در همین دنیا از آنان انتقام می‌گیرد.

◄ این را برای احترام به شاعر می‌نویسم. این شاعر چقدر حالیش بوده و چقدر زندگی کرده بود:

(معلقه‌ی زهیر بن ابی سلمی/۴۶-۵۵) *دیگر از مشقات زندگی سیر شده‌ام. بلی، کسی که هشتاد سال از عمرش می‌گذرد، بی‌گمان از زندگی سیر گردد.*

به وقایع گذشته و حوادث آن روز آگاهی فراوان دارم ولی دیده‌ی باطنم از دیدن فردا نابیناست.

مرگ را چون اشتری کور دیدم که نمی‌داند پایش را کجا می‌گذارد. آنان که را به ناگاه فرو می‌گیرد، می‌میراند و آنان را که نمی‌یابد، می‌مانند تا پیر شوند.

آن کس که با مردم مدارا نکند، در زیر دندان‌های سوانح روزگار دریده شود و زیر پای حوادث خرد گردد.

و آن کس که برای حفظ آبروی خود نیکی کند، بر آبروی خود بیفزاید و آن کس که از دشنام‌گویی بپرهیزد، دشنام شنود.

و آن کس که از مرگ می‌هراسد، اگر چه با نردبام به آسمان فرارود، مرگ او را در خواهد یافت.

و آن کس که در برابر صلح، سرسختی روا دارد، جنگ، او را نرم سازد مانند کسی که از ته نیزه می‌گریزد و به زخم سرنیزه دچار گردد.

◄ اصلا انگار که دارم آیه‌ی قرآن می‌نویسم، از بس که تحکم در آن هست:

(معلقه‌ی زهیر بن ابی سلمی/۶۰ و ۶۲) نیمی از آدمی دل او و نیمی زبان اوست. از این دو که بگذریم، چیزی جز گوشت و خون نیست.

از شما بخشش خواستیم، به ما بخشیدید. باز هم طلبیدیم و باز جوانمردی کردید. ولی کسی که در طلب مبالغه کند، عاقبت محروم خواهد شد.

◄ اصولا این عرب‌ها بیرون‌ریزی‌شان زیاد بوده (رودربایستی نداشتند) و راحت آن چه را که ازش لذت می‌بردند را شعر می‌کردند (جنگ، معشوقه، شمشیر، شتر ...) بعد هم بابت این کارشان تشویق می‌شدند در حالیکه ما احتمالا چنین شعری را مبتذل می‌شماریم. ولی برای آن‌ها که انسان‌های بکری بودند (بی‌دین در طبیعت) شمشیر، مبارزه، معشوق و شراب همان اندازه برای‌شان مطرح بود که شتر! برعکس دین‌داری که همه چیز خداست، آن‌ها از هم قوت می‌گرفتند.

پنجره

تناقض لحظه‌ها
سکوت بعد از ظهر گرم را می‌شکند
و من کلافه‌ام از اینکه
این چرتکه قدیمی
برای شمردن مگسهای اتاق
چند مهره کم دارد.

«احسان‌ا... غلامی»

بارانی از طراوت و شوق
در من است
و عشق، تندیسی ست
که مرا معبود وار
تا جاودانه
به بت پرستی کشانده است
و هر صبح پرسیده‌ام
سهم امروزم از این
خدای هر روزه
چیست؟

سولماز مسعودیان

در خواری دنیا نزد خدا بس، که جز در دنیا نافرمانی او نکنند و جز با وانهادن دنیا به پاداشی که نزد خداست نرسند.

مگو آنچه نمی‌دانی و مگو هر چه می‌دانی.

حق سنگین است اما گوارا و باطل سبک است اما در کام چون سنگ خارا.

مردم دشمن آنند که نمی‌دانند.

هر که پوشش شرم گزیند کس عیب او نبیند.

رشک بردن دوست بر خالص نبودن دوستی اوست.

جدال، تدبیر را ویران می‌کند.

از مولا علی (علیه‌السلام)

جدول ضرب

صفرشدم!
امروز سرِ کلاسِ ریاضی، خانم معلم
می‌گفت:
"هر عدد ضربدر صفر،می‌شود صفر"
من هم خودم را در توانایی‌هایم ضرب
کردم؟!
شد صفر!
حالا نمی‌دانم خودم صفرم یا
توانایی‌هایم.

ش. فریاد

چهره در هم کشید و روی برتافت
چرا که نابینایی به سراغ او آمده بود
تو چه می‌دانی، شاید او پاکی پیشه کند؟
یا متذکر گردد و این یادآوری به حال او مفید باشد.
اما تو به آن کس روی می‌آوری،
که مستغنی است.

سورهٔ عبس

آن راز سر به مهر
آن گمشده به دشت، آهوی خوش خرام
کاووس باغ مهر، گم کرده آشیان
آن شمع پر فروغ، هر لحظه در گداز
مهتاب زرفشان، گر چه مه نهان
آن شیر در کمین، گویی به انتظار رهایی ز بندها
شمشیر در غلاف، آمادهٔ برآمدن از دخمهٔ نیام
آن جنگجوی شیردل و پنجه آهنین
مانده ست منتظر که چه وقت آیدش به گوش
فرمان انتقام!

نجلا محقق

به الف. ن

غیر ممکن است.

حتی نمی‌دانم هیمالیا کوه است یا رودخانه.

از درس جغرافی نمره نمی‌آورم.

خدایا چه کار کنم؟

نذر می‌کنم، نذر بیست تومان که قبول شوم.

نه سی‌تومان که پانزده بگیرم.

اصلا پنجاه تومان که بیست شوم.

و رفتم پول را در صندوق خیرات مدرسه انداختم.

فردا، آقای معلم ورقه‌ها را آورد. از دور آخرین ورقه را دیدم. خط خودم بود. یک ۲ هم در پایین ورقه به چشم می‌خورد. خوشحال شدم. گفتم حتما بیست شده‌ام. خدایا متشکرم!

آقای معلم گفت، نمره‌ها همه خوب بود،به جز یکی ورقه!

بعد رو به من کرد و گفت:

"- شما آبروی کلاس را بردید، پایین‌ترین نمره‌کلاس شدید! دو!"

آن روز تا شب جلوی در اتاق فراش مدرسه ایستادم و انقدر گریه کردم تا مجبور شد پنجاه تومان مرا پس بدهد.

ش.فریاد

بزرگ که شدم

بدون این‌که کسی متوجه شود، به همه‌ی فقیرها غذا می‌دهم

نه به خاطر رضای خدا

به خاطر اینکه گرسنگی نکشند.

بزرگ که شدم

برای همه‌ی بچه‌هایی که شب‌های زمستان در خیابان می‌خوابند، خانه می‌سازم.

نه به خاطر رضای خدا

به خاطر این‌که قلب‌های‌شان در سرما یخ نزند.

بزرگ که شدم

مادربزرگ را به زیارت می‌برم.

نه به خاطر رضای خدا،

به خاطر مادربزرگ

که دوستش دارم.

خدایا کمکم کن!

ش. فریاد

ای که در پاییز قدم می‌زنی

برگ‌های زردی که زیر پای تو به رقص در می‌آیند،

روزی مایه مباهات درختی بوده‌اند.

به برگهای سبزت دل مبند که

که پاییز تو هم در راه است و زمستانی که عریان می‌شوی و در زمستان زیباترین درختان، پر شاخه‌ترین آنهاست.

ر ح ف

از خاطرات دختر

به ونک که می‌رسم،

بوی خانه می‌زند بالا

زود پیاده می‌شوم،

وسط هم نمی‌نشینم.

در قفل کلید می‌پیچانم،

داستانی نه تازه می‌کنم به کار:

مایع بلورین را می‌افروزم،

رخت‌ها را می‌کنم.

نامه دارم، مثل هر روز، مثل دیشب،

همه را من می‌کنم پاک جز یکی،

پنکه سبز اتاقم، می‌شود روشن،

من، فضای خالص را می‌دهم فشار.

نه که من باز کنم،

پنجره. خود باز شود باز، شود باز.

باد سرد می‌آید، سرد از همه پنجره‌ها، جز یکی،

همه را من می‌بندم،

پنجره سردش شده است، جز همه.

گاه‌گاهی از پنجره، می‌کنم فرار.

پنجره، در است روی دیوار،

نامه، حرفهای ناگفته، ناشنیده،

پدر، همسر مادر و مادر، دختر مادر بزرگ

سکه، اسکناس آهنی

ونک، نزدیک خانه.

پنجره، سروش، است.

آن مرد، پشت پنجره بود.

ونک، میدان بزرگی، است.

بادها، سرد، هستند.

جز یکی.

صنم سقاباشی

AXprint.com چیست؟

شما در هرکجاي دنيا كه باشيد مي توانيد از طريق اين سايت فايل تصويري خود را به عكس هاي چاپ شده تبديل كنيد. از AXprint.com مي توانيد در موارد زير استفاده نماييد:

- ارسال عكس براي اقوام خود در ايران از هركجاي دنيا
- چاپ عكسهايي كه توسط دوربين ديجيتال خود گرفته ايد
- چاپ عكس هايي كه از اينترنت گرفته ايد يا براي شما Email شده است

و دهها استفاده ديگر كه قطعا براي شما راهگشاست.

AXprint.com چطور كار مي كند؟

- اگر بار اول است كه از اين سايت استفاده مي كنيد، با كليك كردن روي گزينه ي *كاربر جديد هستم* و پركردن فرم عضويت، به عنوان يكي از كاربران سايت شناخته خواهيد شد و برايتان يك حساب كاربري ايجاد مي شود.
- پس از عضويت در سايت با كليك روي گزينه *ورود* و وارد كردن نام كاربري و كلمه عبور خود (كه در مرحله قبل انتخاب كرده ايد) وارد سايت خواهيد شد.
- براي ايجاد سفارش جديد از منوي سمت چپ صفحه گزينه ي *سفارش چاپ* را انتخاب كنيد.
- در اين بخش پس از مشخص كردن فايل هاي تصويري (كه مطابق *فرمت هاي مشخص شده* هستند)، با زدن دكمه ي Upload آنها را به سايت AXprint ارسال نماييد.
- در صفحه بعد بايد تعداد و اندازه هر يك از عكس هاي خود را مشخص كنيد. *قيمت* سفارش شما و مقدار اعتبار فعلي تان هم در اين صفحه نشان داده مي شود.
- سرانجام در آخرين صفحه آدرس و نحوه ارسال سفارش از شما پرسيده مي شود.
- اگر مشكل ديگري در كار با سايت داشتيد، صفحه *راهنماي سايت* را بخوانيد، يا *با ما تماس بگيريد*.

WWW.AXprint.com

دکتر لطفی‌زاده

خالق منطق فازی

او کیست؟

دکتر لطفی‌زاده پدر منطق فازی است. مخترع مدلی برای نحوه‌ی استدلال انسانی که در آن همه چیز —حتی حقیقت— رتبه‌مند هستند. او در سال ۱۹۶۵ نظریه‌ی منطق فازی را ارائه داد و اکنون camcorder ها توسط آن لرزش دست را می‌گیرند، غذای ما در مایکروویو پخته می‌شود و ... نیز در بسیاری از زمینه‌های غیرمهندسی این ایده توسعه یافته است.

در زمان‌های ارسطو و دکارت، انسان‌ها سعی می‌کردند تا حد امکان دقیق باشند چرا که گویی با کمک دقت می‌توانند به بسیاری از حقایق دست یابند، ولی در طی این فرایند بیشتر از آن چه به دست آوردند، از دست دادند. منطق فازی حرکت به سوی آن از دست رفته‌ها است، التفات به استدلالات تقریبی که امور هر روزه‌ی ما توسط آن پیش می‌رود.

آیا دکتر لطفی‌زاده ایرانی است؟

"مساله این نیست که آیا من ایرانی هستم یا آذربایجانی، روس، آمریکایی یا هر چیز دیگری. تمام این فرهنگ‌ها در شکل یافتن من موثر بوده‌اند و من در میان همه‌ی آن‌ها احساس آرامش می‌کنم."

بزرگترین دستاورد او در زندگی‌اش چه بوده؟

"تئوری منطق فازی البته و نیز کار اخیرم در ارتباط با تئوری محاسباتی ادراک. گمان می‌کنم این تئوری تاثیر بزرگی خواهد داشت. تاثیر آن نه فقط در مهندسی کردن، بلکه به علوم پایه و بسیاری از زمینه‌های دیگر نیز سرایت خواهد نمود. مفهوم آن زمینه را برای توسعه‌ی نقش زبان طبیعی در تئوری‌های علمی مهیا خواهد نمود. ما در هنگامی که با علم سر و کار داریم، از فرمول‌ها، نمادهای ریاضی و اعداد استفاده می‌نماییم ولی تئوری محاسباتی ادراک، ما را قادر خواهد ساخت که از زبان طبیعی در گفتمان علمی استفاده نماییم. به این ترتیب جابجایی بزرگی در پارادایم‌ها و الگوهای ما رخ خواهد داد." *شگفت است که کسی در سن ۷۸ سالگی دوباره نکان شدیدی به علم بدهد.*

منبع

سایت بین‌المللی آذربایجان: www.azar.com (تعجب نکنید که چرا کشور آذربایجان، چون به هر حال ایشان در باکو، آذربایجان، هنگامی که پدرشان به عنوان خبرنگار در آن‌جا کار می‌کرد در سال ۱۹۲۱ به دنیا آمد و البته بعد هم در ده سالگی به تهران سفر نمود، در دبیرستان البرز درس خواند، با رتبه‌ی دوم در کنکور به دانشگاه تهران رفت، لیسانس الکترونیک گرفت، به آمریکا سفر کرد، در MIT درس خواند و در برکلی درس داد و ...)

سوال: متد ما در هوش‌مصنوعی چیست؟

سید احسان لواسانی

یک طریق بحث این است که بپرسیم: آیا باید از طریق تفسیر عمل‌هایی که به هوشمندانه بودن معروفند وارد شد؟ یا از طریق شبیه‌سازی مسائل خود به خودیِ معرفتی[1]؟ این روش بحث مانند صحبت از تفاوت میان روش هرمنوتیک-تفسیری با روش تجربی-علمی‌است.

در حالت اول ادعا این است که ما یک سری general background knowledge داریم (همان پارادایم‌ها) که به ما می‌گویند چه چیز را هوشمندانه ببینیم و چه چیز را غیر هوشمند[2]. در این حالت ما به دنبال فهمیدن هوش هستیم و می‌دانیم برای فهم هوش باید پیش فرض‌های اجتماعی حاکم بر ذهنمان را بررسی کنیم.

اما در حالت دوم فرض ! این است که فکر و هوش منبعث از ساختار فیزیکی-منطقی مغز ماست که به خاطر این گونه ساختار خود نتایج خاصی از مقدماتی خاصی نتیجه می‌شوند. پس برای داشتن هوش‌مصنوعی نیز ما باید بتوانیم این ساختار را شبیه‌سازی کنیم و چیزی از آن دست بنا سازیم.

به بیان دیگر در تلقی هرمنوتیکی از هوش ما با تلقی‌هایی که از رفتارهای موصوف به هوشمندانه وجود دارند سر و کار داریم؛ یعنی با اعتبارات و روابط و احکام جمعی انسان‌ها. در این روش ما می‌خواهیم بفهمیم چه شد که این رفتارها هوشمندانه تلقی شده‌اند. به دنبال این فهم نیز می‌خواهیم جایگاه ماشین خود را به گونه‌ای بیابیم که او نیز هوشمند نامیده شود. اما در تلقی تجربی-علمی‌از هوش ما با عقل و مغز به عنوان یک ابزار رو به رو می‌شویم؛ ابزاری که یک سری functionality دارد کارایی آن را هم صرفا به ساختار درونی منطقی-بیولوژیکی آن نسبت می‌دهیم. پس قصد ما در حالت دوم، شبیه‌سازی آن کارآمدی‌ها در ابزار دیگری است.

بیان آشناتری در هوش‌مصنوعی[3] این است که‌ایا روش ما در هوش‌مصنوعی به پیروی از physical symbol system است و می‌خواهیم هوش را از طریق نمادها و روابط منطقی و سطح بالای آن‌ها بازنمایی[4] کنیم و یا که نه منطبق است بر روش nueral network و بازسازی هوش از طریق روابط ساده و احمقانه‌ی چندین نود؟

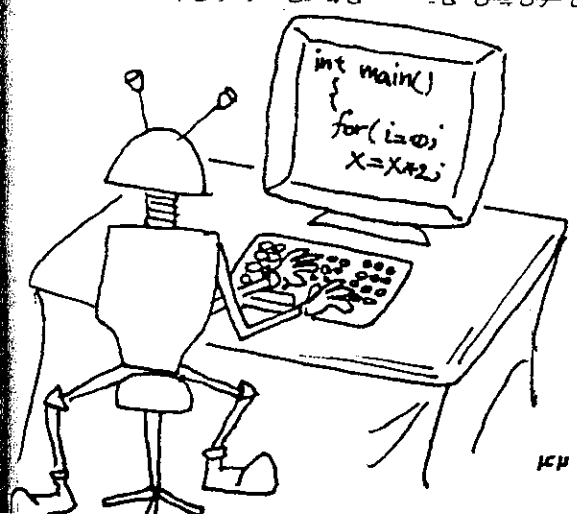
پس طریقه دیگر بحث این است که بپرسیم آیا ما در هوش‌مصنوعی ابزارگرا هستیم یا تفسیرگرا؟ به دنبال فهمیم یا به دنبال توفیق یافتن در ساختن یک ابزار؟

البته سوال این مقاله انتخاب بین این دو روش نیست؛ هر چند به آن نیز خواهیم پرداخت. سوال این مقاله این است که "چه چیزی تعیین می‌کند که ما در هوش‌مصنوعی چه چیزی را مطالعه کنیم؟" پاسخ این سوال خود از بین دو دسته روشی که بیان شد یکی را برخواهد گزید.

پاسخ مجمل پرسش این مقاله چنین است: تلقی ما از هوش و نظریه ی ما درباره هوش‌مصنوعی و هوش طبیعی؛ روش تحقیق در آن را نیز تعیین می‌کنند. ما هوش را از طریق نظریه‌های مان درباره هوش می‌فهمیم. اگر نظریه‌های ما بگویند که هوش مانند دستگاه گوشت چرخ کن است، پس ما نیز سعی می‌کنیم با داشتن ورودی و خروجی، چیزی شبیه به چرخ کن بسازیم که اگر دوباره آن ورودی‌ها را به آن دادیم بتواند باز آن خروجی‌ها را بازتولید کند. اما اگر نظریه‌های ما بگویند در هوش‌مصنوعی عنصر "تصمیم" وجود دارد و تلقی دستگاه از ورودیهایش، بر خروجی آن اثر می‌گذارد، کلا روش برخورد ما نیز تغییر خواهد کرد.

خلاصه آن که ما انسان‌ها هستیم که به هوش‌مصنوعی و هوش طبیعی معنای خوش بودن را اعطا می‌کنیم؛ یعنی هوش انسانی و هوشمندی ما به مغز ما نسبت داده نمی‌شود (به این معنا که مغز ذاتا و ماهیتا هوشمند باشد)؛ بلکه هوشمندی را ما خودمان به خودمان نسبت می‌دهیم. به عبارت دیگر هوشمندی به تلقی ما از هوش نسبت داده می‌شود. قبول این پاسخ، ما را به این نتیجه می‌رساند که تلقی ما از هوش، روش تحقیق ما را نیز تعیین می‌کند. برای مثال اگر تلقی ما از فرشتگان این باشد که موجودات غیرمادی اند، هیچگاه روش‌های تجربی را هم برای شناختن آن‌ها به کار نمی‌گیریم.

حال این سوال پیش می‌آید که تلقی پیشین ما از هوش چیست؟ اصلا



---

[1] خود به خودی معرفتی مانند استنتاج منطقی نتیجه از مقدمات

[2] seeing as an inteligent act و نه to see an inteligent act

[3] این بیان دقیقا منطبق بر تمایزی که بین روش تفسیری و روش تجربی گزاردیم نیست ولی در هوش مصنوعی آشناتر و گویاتر است.

[4] represent

گذری کوتاه بر

موسیقی کلاسیک و پاپ

پژمان کلانتری

آن‌جا که زبان از سخن گفتن ناتوان می‌ماند، موسیقی آغاز می‌شود. موسیقی را هنر بیان احساسات درونی انسان توسط صداها تعریف کرده‌اند؛ صداهایی که ممکن است بوسیلهٔ آلات موسیقی، حنجرهٔ انسان‌ها و یا اصوات طبیعت مانند آواز قناری و بلبل،رودخانهٔ جاری،آبشارهای خروشان و ... ایجاد شوند. این صداها بسته به آلت موسیقی بکار رفته و یا احساساتی که از انسان‌ها بر می‌انگیزند و آنگاه ترکیب این دو سبک‌های خاص و معینی از موسیقی را پدید می‌آورند.

دو سبک مشهور و محبوب موسیقی امروزی عبارتند از:

۱- سبک کلاسیک (Classical)

۲- سبک پاپ (Popular)

در سازبندی موسیقی کلاسیک هر سه نوع تقسیم‌بندی سازها یعنی بادی، زهی و کوبه‌ای دیده می‌شوند؛ تعداد سازهای بکار رفته در هر گروه معمولاً بیش از یکصد عدد می‌باشد که طبیعی است نام بردن آن‌ها در این نوشته نمی‌گنجد. این سبک از نظر میزان‌های موسیقی محدود به چند میزان خاص و معروف نیست بلکه یکی از نقاط قوت موسیقی کلاسیک گستردگی استفاده از میزان‌های موسیقی و متنوع بودن میزان‌های این نوع آثار می‌باشد. در سبک پاپ سازی‌ها سنتی سایزر (کیبورد)، گیتار (آکوستیک یا برقی) و درام (جاز) تقریباً جزو اعضای ثابت و همیشه حاضر در گروه‌های موسیقی پاپ می‌باشند که به تناوب از سازهایی مانند پرکاشن (تومبا)، ویلن، فلوت، ساکسیفون و ترومپت نیز بسته به نوع آثار استفاده می‌شود. یکی از نقاط ضعف همیشگی سبک پاپ محدود بودن به چند میزان معروف مانند $\frac{2}{4}, \frac{4}{4}, \frac{3}{4}$ و در بعضی موارد $\frac{6}{8}$ است که قابل قیاس با موسیقی کلاسیک نمی‌باشد.

اما بحثی که برای خود نگارنده برای مدت‌ها بصورت ابهامی بجا مانده این است که چرا با وجود اینکه بطور کلی سبک کلاسیک از هر نظر (مانند تعداد سازها، تنوع میزان‌ها، نیاز به دارا بودن سواد موسیقی توسط تک تک اعضای گروه و شخصیت خاص سبک کلاسیک و ...) با سبک پاپ قابل مقایسه نیست و در سطح بسیار بالاتری از موسیقی پاپ قرار دارد؛ اما انسان امروزی (پیر و جوان، غربی و شرقی)گرایش بیشتری به سبک پاپ دارد؟

چرا دیگر سمفونی‌هایی نظیر سمفونی‌های آهنگسازان بزرگ تاریخ همچون بتهوون،باخ، موتزارت و ... توسط هنرمندان فعلی ساخته نمی‌شود و بزرگان موسیقی حال حاضر جهان نظیر ونجلیز، کلایدرمن،یانی و ... به ساختن کارهای تلفیقی روی آورده‌اند؟

واقعیت این است که انسان امروزی به دلیل شرایط زندگی تحمیل شدهٔ این دوران، انسانی سطحی نگر و برون گرا شده است. او دیگر پس از مشغله‌های روزانه توان فکر کردن و تحلیل موسیقی مورد علاقه‌اش را ندارد که مثلاً فلان سمفونی بتهوون زندگی گنجشک‌ها روی شاخه‌های درختان قبل و بعد از مرگشان را ترسیم می‌کند. او برای لحظات اندوه خویش بطور مثال فقط به یک فلوت و شعری ساده متناسب با مشکلات خود نیاز دارد و برای شادی کردن تنها به چند سر ضرب درام یا پرکاشن محتاج است تا ریتم شادی خویش را حفظ کند. بنابراین شاید به این دلیل باشد که این موسیقی سطحی، البته ساده و زود فهم را (سبک پاپ) انتخاب می‌کند.