P vs NP

دکتر رسول رمضانیان

در کاری که انجام شده است دو مفهوم اساسی وجود دارد که این دو مفهوم به نحوی در هم تنیده‌اند:

۱- از پیش تعیین شده نبودن معنای یک واژه (یک کد ماشین تورینگ)،

۲- تغییر پایای یک معنا.

اثبات چیست؟

ما می‌خواهیم مساله دشوار P vs NP را حل کنیم. یعنی یا اثباتی در رد آن یا اثباتی در موافقت آن ارائه دهیم. اما یک اثبات چیست؟ اثبات یک گزاره عبارت است از آغاز کردن از یک سری اصول بدیهی واضح (که بدون اثبات پذیرفته می‌شوند؛ چون اصل استقرا، اصل لانه کبوتری و ...) و استفاده از قواعد استنتاج منطقی و رسیدن به گزاره مورد نظر. از اصول بدیهی می‌توان به اصل لانه کبوتری و اصل استقرا اشاره کرد (که بدون اثبات پذیرفته می‌شوند و دلیل این است که برای انسان کنونی این اصول واضح هستند، یعنی برای ما واضح است! که اگر سه کبوتر را بخواهیم در دو لانه بنشانیم حتما در یکی از دو لانه دو کبوتر باید بنشیند). انسان ۱۰ هزار سال پیش غار نشین را لحاظ کنید که هنوز توان تکلم نداشت. آیا برای او نیز اصل لانه کبوتری واضح بوده است؟ در واقع واضح و بدیهی شدن یک اصل در نتیجه تکامل انسان رخ می‌دهد!!! حال فرض کنید ما می‌خواهیم یک گزاره را ثابت یا رد کنیم. اما این گزاره از هیچیک از اصولی که برای ما واضح و بدیهی به نظر می‌رسند قابل استخراج نیست. بلکه لازم است که اصلی بدیهی پیدا شود تا از آن اصل بتوان مساله را ثابت یا رد کرد. نحوه اندیشیدن در این نوشتار در مورد مساله P vs NP به همین نحوه است.

فرض کنید که یک جعبه سیاه داریم که دارای دو نوار ورودی و خورجی است. روی نوار ورودی یک عدد طبیعیa می‌نویسم و جعبه سیاه روی نوار خروجی عدد طبیعی b را می نویسد.

گوییم جعبه سیاه خوش‌تعریف است هرگاه اگر در مرحله‌ای از زمان عدد طبیعی c را به جعبه وارد کردیم، و جعبه عدد d را در خروجی نوشت، آنگاه هرگاه در آینده دوباره همان عدد c را به جعبه وارد کنیم، جعبه دوباره همان عدد d را در خروجی چاپ کند.

گوییم جعبه سیاه ایستا است هرگاه خروجی‌ها به ترتیب ورودی‌ها وابسته نباشد.

پرسش: آیا اگر یک جعبه خوش‌تعریف عمل کند، آنگاه حتما ایستا است؟

پاسخ: خیر . کد زیر را در داخل جعبه سیاه قرار دهید.

1. Input x;
2. Check to see whether there exists a pair (x,y) in MEMORY, for some y, if there exists output y, goto line 1;
3. If [(x=5 or there exists a pair (5,2) in Memory) and there exists no pair (13,3) in MEMORY] output 2, save (x,2) in Memory, goto line 1;

4. If [(x=13 or there exists a pair (13,3) in Memory) and there exists no pair (5,2) in MEMORY] output 3, save (x,3) in Memory, goto line 1;

5. If (x≠5 and x≠13) output 1, save (x,1) in MEMORY, goto line 1;

جعبه خوش‌تعریف عمل می‌کند، اما وابسته به اینکه شما عدد ۱۳ را قبل از عدد ۵، یا عدد ۵ را قبل از عدد ۱۳ به جعبه وارد کنید، جعبه سیاه دارد توابع متفاوتی را محاسبه می‌کند. اگر تابع محاسبه شده توسط کد بالا را معنای[۱] کد در نظر بگیریم، کد بالا یک معنای از پیش‌تعیین‌شده[۲] ندارد، و معنای کد در اثر کنشگری با کاربر در طی گذر زمان شکل می‌گیرد!! معنای کد به اراده کاربر وابسته است (این همان نقطه کلیدی اثبات مساله اصلی است)

برای بهتر روشن شدن موضوع مثالی دیگر را در نظر بگیرید. فرض کنید E یک زبان برنامه نویسی است و من یک کامپایلر C برای زبان برنامه نویسی E طراحی کردم که این زبان را به زبان ماشین تبدیل می‌کند. به این ترتیب که، نحوه ترجمه آن بطور پایستاری (خوش‌تعریفی) تغییر می‌کند. یعنی اگر شما پیش‌تر دستوری در زبان E را توسط کامپایلر اجرا کرده باشید و از نتیجه آن آگاه شده باشید، هرگاه دوباره همان دستور را اجرا کنید، نتیجه همان چیزی خواهد بود که پیش‌تر از آن آگاه شده بودید. اما ترتیبی که شما دستورات را اجرا می‌کنید، سبب می‌شود که کامپایلر بطور پایایی تغییر کند. به این ترتیب، برنامه‌هایی که شما در زبان E می‌نویسید روی کامپایلر C معنایی از پیش‌تعیین شده ندارند! هر چند اگر شما به نحوه کارکرد کامپایلر توجه نکنید، هیچگاه از این موضوع آگاه نخواهید شد. (در مقاله [۱] ، نشان داده شده است که می‌توان یک زبان برنامه نویسی E و یک کامپایلر C برای آن معرفی کرد، بطوریکه بتوان در این محیط همه ماشین‌های تورینگ را پیاده‌سازی کرد، و به علاوه برنامه‌ای نوشت که زبان آن متعلق به NP هست ولی متعلق به P نیست)

اصل عدم تمییز: اگر معنای یک واژه (در یک زبان) برای انسان بطور پایستاری (با حفظ خوش-تعریفی) تغییر کند، انسان نمی‌تواند تفاوتی بین اینکه آیا معنا از پیش-تعیین‌شده است، یا معنا دینامیک است و در اثر کنشگری در طی زمان شکل می-یابد، قائل شود. (مقاله [۲] را ببینید)

در واقع این شبکه عصبی مغز ما است که به کدهای ماشین‌های تورینگ معنا می-بخشد. کدهای ماشین تورینگ همانند کدهای زبان برنامه‌نویسی E در مثال بالا هستند، اما برای آنکه مغز ما آنها را درک کند، باید به زبان شبکه عصبی (زبان ماشین مغز انسان) تبدیل شوند، حال چگونه می‌توان مطمئن بود که این فرآیند کامپایل کردن به طور پایستاری تغییر نمی‌کند؟ در واقع اگر شبکه عصبی مغز را نادید بگیریم، هیچ راهی برای تمییز قائل شدن بین ایستایی و تغییر پایا نخواهیم داشت، زیرا ما نمی‌توانیم به گذشته برگردیم و نمی-توانیم ترتیب‌های گوناگون را آزمون کنیم که ببینیم آیا یک ماشین تورینگ، معنایش وابسته به نحوه اجرا شبکه عصبی مغز ما است یا نه! در واقع ما با دو زبان روبرو هستیم، زبان برنامه نویسی، و زبان ماشین. یک کامپایلر زبان اول را به دومی ترجمه می‌کند. تز چرچ تورینگ می‌گوید، کدهای ماشین تورینگ زبان برنامه نویسی شبکه عصبی مغز ما هستند. زبان ماشین مغز ما برای ما ناشناخته است. هیچ دلیلی هم نداریم که فرض کنیم که کامپایلری که کدهای تورینگی را به زبان ماشین ترجمه می‌کند، بطور پایستاری تغییر نمی-کند.

بنابراین

۱. کدینگی که تورینگ ارائه می‌دهد، در واقع یک زبان برنامه‌نویسی است که CPU آن شبکه عصبی مغز انسان است. این زبان برنامه‌نویسی باید به زبان ماشین توسط یک کامپایلر ترجمه شود، حال ممکن است که کامپایلر مغز انسان، بطور پایستاری تغییر کند، زبان و تابعی که (معنایی که) یک ماشین تورینگ آن را توصیف و نمایندگی می‌کند، از پیش تعیین شده نیست!

۲. اگر زبان یک ماشین تورینگ از پیش تعیین شده نیست، اراده آزاد انسان در این که کدام برنامه را اجرا کند، سبب می‌شود که او بتواند معناهای ممکن بسیاری را (البته تنها یکی از آنها را، چون نمی تواند به گذشته برگردد) تحقق بخشد.

پس دو جهان ممکن پیش رو داریم

۱. معنای یک واژه (ماشین تورینگ) از پیش تعیین شده است

۲. معنای یک واژه (ماشین تورینگ) از پیش تعیین شده نیست.

---

[۱] semantics

[۲] Predetermined

در حالت دوم ثابت می‌شود که P مساوی NP نیست (مقاله [۱] را ببینید). بنابر اصل عدم تمییز ، چون نمی‌توان بین دو حالت ۱ و ۲ تمییز گذاشت، پس نمی‌توانیم هیچگاه ثابت کنیم .P=NP

مراجع

[1] R. Ramezanian, Computation Environment 1, An interactive Semantics for Turing Machines (which P is not equal to NP considering it), arXiv, 1205.5994v1. 2012.

[2] R. Ramezanian, Computation Environment 2, Persistently Evolutionary Semantics, arXiv, 1207.0051v1. 2012.

[3] R. Ramezanian, Computation Environment 3, P=NP Conflicts with the Free Will, http://sharif.edu/ ramezanian/cela.pdf