به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار ۱۴۰۱

تمرین شماره 12

دستیار آموزشی این مجموعه: آوا میرمحمدمهدی

avamir80@gmail.com

تاریخ تحویل :1401/10/14

---

1) درستی یا نادرستی هر یک از عبارات زیر را با ذکر دلیل مشخص کنید. (16 نمره)

الف) اگر اثبات شود که یک مساله‌ی NP-complete درون دسته‌ی P قرار می‌گیرد، آنگاه تمامی مسائل NP درون دسته‌ی P قرار می‌گیرند.

ب) اگر در آینده ثابت شود مساله‌ی 3-SAT در زمان $o(n^6)$ قابل حل است، تمام مسائل NP در زمان $o(n^6)$ قابل حل هستند.

ج) اگر یک مساله‌ی NP در زمان چندجمله‌ای حل شود، تمام مسائل NP نیز در زمان چندجمله‌ای حل می‌شوند.

د) مساله‌ی A را در نظر بگیرید؛ اگر بتوان در زمان چندجمله‌ای مساله‌ی 3-SAT را به مساله‌ی A کاهش داد و همچنین بتوان در زمان چندجمله‌ای مساله‌ی A را به مساله‌ی 3-SAT کاهش داد، مساله‌ی A در دسته‌ی NP-complete قرار دارد.

پاسخ:

الف) صحیح - تمام مسائل NP را می‌توان به مساله‌ی NP-complete کاهش داد پس درصورتی که آن مساله عضو P باشد، تمامی مسائل NP را نیز می‌توان به مساله‌ی P کاهش داد پس عضو P می‌شوند.

ب) غلط - تمام مسائل NP قابل کاهش به مساله‌ی 3-SAT هستند و با این فرض، تمام مسائل NP در زمان چندجمله‌ای قابل حل هستند ولی این لزوما به این معنا نیست که مرتبه‌ی زمانی آن کمتر یا مساوی مرتبه‌ی زمانی مساله‌ای است که به آن کاهش یافته‌اند.

ج) غلط - اگر یک مساله‌ی NP-complete در زمان چندجمله‌ای حل شود، می‌توان نتیجه گرفت تمام مسائل NP نیز در زمان چندجمله‌ای حل می‌شوند ولی با حل شدن یک مساله‌ی NP نمی‌توان این نتیجه‌گیری را کرد.

د) صحیح - با کاهش مساله‌ی 3-SAT در زمان چندجمله‌ای به مساله‌ی A، از آنجایی که 3-SAT در دسته‌ی NP-complete است، پس مساله‌ی A در دسته‌ی NP-hard قرار می‌گیرد؛ همچنین با کاهش مساله‌ی A به مساله‌ی 3-SAT در زمان چندجمله‌ای می‌توان نتیجه گرفت که مساله‌ی A در دسته‌ی NP قرار دارد چون همه‌ی مسائل NP را می‌توان به مساله‌ی 3-SAT کاهش داد؛ از دو اثبات بالا نتیجه می‌شود که مساله‌ی A در دسته‌ی NP-complete قرار می‌گیرد.

2) گراف G را در نظر بگیرید؛ ثابت کنید تعیین این که گراف ’G زیرگراف G است یا خیر، در کلاس پیچیدگی NP-COMPLETE قرار دارد. (15 نمره)

پاسخ:

ابتدا NP بودن مساله را اثبات می‌کنیم؛ اگر تعداد رئوس را n و تعداد یال‌ها را m در نظر بگیریم، بررسی اینکه تمام رئوس و یال‌های گراف ’G در گراف G قرار داشته باشد در زمان $o(n + m)$ قابل انجام است که چندجمله‌ای است پس این مساله در دسته‌ی NP قرار می‌گیرد.

برای اثبات NP-hard بودن این مساله، مساله‌ی Clique را به آن کاهش می‌دهیم؛ فرض کنید <Graph, k> ورودی مساله‌ی Clique باشد، برای تبدیل این ورودی به ورودی مساله‌ی داده شده، گراف ’G را یک گراف کامل با k راس در نظر می‌گیریم و G (که ورودی مساله‌ی داده شده است) را برابر با Graph(که ورودی مساله‌ی Cliqueاست) در نظر می‌گیریم؛ تبدیل ورودی Clique به سماله‌ی مورد نظر در زمان چندجمله‌ای قابل انجام است چون تشکیل گراف ’G به اندازه‌ی $o(k^2) = o(n^2)$ زمان می‌برد که چندجمله‌ای است. حال نشان می‌دهیم Graph دارای یک clique به اندازه‌ی k است اگر و تنها اگر گراف ’G، زیرگراف G باشد؛ اگر گراف ’G، زیرگراف G باشد، از آنجایی که ’G یک گراف کامل با k راس است پس قطعا در G یک clique با اندازه‌ی k وجود دارد؛ همچنین اگر در Graph یک clique با اندازه‌ی k وجود داشته باشد، با توجه به نحوه‌ی تشکیل ’G واضح است که ’G زیرگراف G می‌شود؛ طبق توضیحات داده شده، توانستیم یک مساله‌ی NP-hard را به مساله‌ی داده شده کاهش دهیم پس مساله‌ی ما NP-hard است.

در بالا ثابت شد که مساله‌ی ما هم NP و هم NP-hard است پس در دسته‌ی NP-complete قرار می‌گیرد.

3) ثابت کنید مکمل زبان REG-EQ در دسته زبان‌های NP قرار دارد. (20 نمره)

REG-EQ = {⟨L_1 , L_2⟩|

زبان‌های L_1 و L_2 دو زبان منظم و معادل هم هستند به طوری که در تعریف آنها به شکل عبارت منظم، از علامت * استفاده نشده باشد.}

پاسخ:

می‌توان پاسخ‌های قابل قبول برای مکمل زبان REG-EQ را به طور زیر دسته‌بندی کرد:

الف) حداقل یکی از دو زبان L_1 و L_2 منظم نباشد.

ب) هر دو زبان L_1 و L_2 منظم باشند ولی حداقل در تعریف یکی از آنها به شکل منظم از علامت * استفاده شده باشد.

ج) هر دو زبان L_1 و L_2 منظم باشند و در تعریف هر دو به شکل منظم از علامت * استفاده نشده باشد ولی L_1 و L_2 معادل هم نباشند.

مورد الف را می‌توان به راحتی با توجه به الگوریتم‌های ارائه شده در درس در زمان چندجمله‌ای accept یا reject کرد. مورد ب را نیز می‌توان با پیمایش رشته‌ی ورودی در زمان چندجمله‌ای فهمید. برای فهمیدن مورد ج باید بفهمیم آیا رشته‌ای وجود دارد که فقط یکی از زبان‌ها بپذیرد یا خیر؛ از طرفی می‌دانیم در تعریف هیچ یک از زبان‌ها به شکل عبارت منظم از علامت * استفاده نشده است و درنتیجه طول رشته‌هایی که این دو زبان می‌پذیرند محدود است و به اندازه‌ی طول عبارت منظمی است که این زبان‌ها را توصیف می‌کنند؛ پس می‌توان به طور غیرقطعی و در زمان چندجمله‌ای رشته‌هایی را حدس زد و بررسی کرد که در کدام یک از این زبان‌ها است. از توضیحات بالا نتیجه می‌شود که توسط یک ماشین تورینگ غیرقطعی می‌توان عضویت رشته‌ها در مکمل زبان REG-EQ را بررسی کرد و ورودی ادعا شده را در زمان چندجمله‌ای accept یا reject کرد.

4) گزاره‌های زیر را در صورت درست بودن، اثبات کنید و در غیر این صورت برای آن مثال نقض بیاورید. (14 نمره)

الف) اگر A و B دو زبان NP-complete باشند، $A \cap B$ نیز عضو NP-complete است.

ب) اگر A و B دو زبان NP-complete باشند، $A \cup B$ نیز عضو NP-complete است.

پاسخ:

الف) نادرست؛ زبان‌های A و B را که در دسته‌ی NP-complete هستند به صورت زیر تعریف می‌کنیم:

$A = \{1\#x : x \in SAT\}, \ B = \{0\#x : x \in SAT\}$

از آنجایی که $x \in SAT$ پس این زبان‌ها قطعا NP-complete هستند؛ اشتراک این دو زبان تهی است در نتیجه عضو NP-complete نیست.

ب) نادرست؛ زبان‌های A و B را که در دسته‌ی NP-complete هستند به صورت زیر تعریف می‌کنیم:

$A = \{1\#x : x \in SAT\} \cup \{0\#x : x \in \{0, 1\} *\}, \ B = \{0\#x : x \in SAT\} \cup \{1\#x : x \in \{0, 1\} *\}$

$A \cup B = \{0, 1\} *$

همانطور که در بالا می‌بینیم، اجتماع این دو زبان، شامل تمامی رشته های باینری است پس در دسته زبان‌های NP-complete نیست.

5) مجموعه‌ی M را در نظر بگیرید؛ مجموعه‌ی D شامل تعدادی از زیرمجموعه‌های دوعضوی Mاست. مجموعه S را یک زیرمجموعه‌ی تکمیل می گوییم هرگاه $S \subseteq M$ وهمچنین این زیرمجموعه با تمام اعضای D حداقل در یک عضو مشترک داشته باشد. زبان Common-Subs شامل تمام ورودی‌ها به شکل $\langle D, M, k\rangle$ است به طوری که به ازای مجموعه‌های D و M حداقل یک زیرمجموعه‌ی تکمیل با k عضو وجود داشته باشد. ثابت کنید این زبان از نظر پیچیدگی زمانی در دسته‌ی NP-COMPLETE قرار دارد. (15 نمره)

پاسخ:

ابتدا ثابت می‌کنیم که این زبان عضو دسته‌ی NP است؛ فرض می‌کنیم مجموعه‌های M , D (با mعضو) و مجموعه‌ی S با k عضو برای حل این سوال ادعا شده‌اند. ابتدا تمامی اعضای D را بررسی می‌کنیم که دوعضوی باشند و همچنین در مجموعه ی M وجود داشته باشند؛ سپس تمام مجموعه‌های دوعضوی D را بررسی می‌کنیم که حداقل یکی از آنها در S موجود باشد؛ اگر در هر مرحله از بررسی یکی از شروط برقرار نباشد، reject می‌کنیم و اگر همه‌ی شروط برقرار باشند، accept می‌کنیم؛ این کار در زمان$o(m \times n)$ قابل انجام است که چندجمله‌ای است پس این زبان در دسته‌ی NP قرار دارد.

برای اثبات NP-HARD بودن این مساله، مساله‌ی vertex cover را به آن کاهش می‌دهیم. گراف $G = (V,E)$ و عدد k ورودی های مساله‌ی vertex cover هستند که از روی آن باید ورودی مساله‌ی خودمان را بسازیم. مجموعه‌ی M را برابر با V (رئوس گراف) و D را برابر با E (یال‌های گراف) در نظر می‌گیریم و همچنین عدد k نیز همان عدد k در vertex cover است؛ در واقع در مجموعه‌ی E، دوتایی هایی قرار دارند که دو راس تشکیل دهنده‌ی یال‌ها هستند؛ حال برای اثبات طرف اول اثبات می‌کنیم که پاسخ مساله‌ی vertex cover همان پاسخ مساله ی ماست چون با

k عضو تمامی یال‌ها پوشش داده شده‌اند و این بدین معناست که با تمامی اعضای D حداقل یک عضو مشترک دارد و به طورمشابه طرف دیگر اثبات نیز بدیهی است.

6) گراف G را با V راس در نظر بگیرید به طوری که هر راس نماینده‌ی یک فرد در یک مدرسه است. دو فرد درصورتی به یکدیگر یال دارند که با یکدیگر رابطه‌ی دوستی داشته باشند. می‌خواهیم ببینیم که آیا می‌شود مجموعه رئوس گراف G را به حداکثر k دسته‌ی $T_1, T_2, \dots, T_k$ افراز کرد به طوری که هر فرد در یک دسته با تمامی افراد آن دسته رابطه‌ی دوستی داشته باشد و همچنین هر فرد دقیقا در یک دسته وجود داشته باشد. ثابت کنید زبان L که در واقع توصیف کننده‌ی این مساله است، یک زبان NP-complete است.

$L = \{\langle G , k\rangle |$

گراف G دارای افرازی با حداکثر k دسته است.}

راهنمایی: برای حل این مساله می‌توانیدی از NP-complete بودن مساله ی [k-colouring] استفاده کنید. (20 نمره)

پاسخ:

ابتدا اثبات می‌کنیم که زبان L در دسته‌ی NP قرار دارد. گواهی ای که برای این مساله داده می‌شود، افرازی روی گراف G است. Verifier بررسی می‌کند که اولا تعداد دسته های این افراز حداکثر k باشد، ثانیا هر راس در یک دسته به تمامی رئوس دسته‌ی خود یال داشته باشد و ثالثا هر راس تنها در یک دسته قرار داشته باشد؛ بررسی هر سه مورد ذکر شده در زمان چندجمله‌ای امکان پذیر است پس زبان L در دسته‌ی NP قرار می‌گیرد.

برای اثبات NP-hard بودن این زبان، مساله‌ی k-colouring را به آن کاهش می‌دهیم؛ ورودی مساله رنگ آمیزی، <G, k>است. G گرافی است که می‌توانیم یک رنگ (عددی بین 1 تا k) به هریک از راس های آن اختصاص دهیم به طوری که دو سر هیچ یالی با یک رنگ، رنگ آمیزی نشده باشند. برای تبدیل ورودی‌های مساله k-coloring به مساله‌ی داده شده، مکمل گراف G و عدد k را به عنوان ورودی به زبان L می‌دهیم. گراف G را می‌توان با حداکثر k رنگ، رنگ آمیزی کرد اگر وتنها اگر گراف مکمل G دارای یک افراز با حداکثر k دسته باشد؛ برای اثبات جهت برگشت این ادعا، می‌دانیم در مساله‌ی رنگ آمیزی، راس‌های همرنگ قطعا به یکدیگر متصل نیستند و از هم مستقل اند پس درواقع ما گرافمان را به دسته‌هایی افراز کردیم به طوری که هر دو راس در دو گروه متفاوت از یکدیگر مستقل‌اند و به هم یالی ندارند. اگر از گراف گفته شده مکمل بگیریم، دسته های تشکیل شده در واقع همان خواسته‌ی زبان L است. جهت رفت اثبات نیز با همین استدلال قابل اثبات است پس زبان L در دسته مسائل NP-hard قرار دارد.

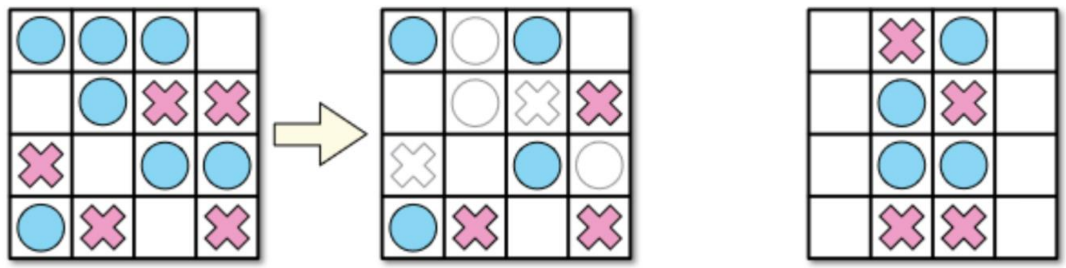
از NP و NP-hard بودن زبان L نتیجه می‌شود که این زبان در دسته مسائل NP-complete قرار دارد.

7) جدولی با n سطر و m ستون داریم به طوری که در هر خانه از این جدول علامت x یا o قرار گرفته است و یا هیچ علامتی قرار نگرفته است. هدف این است که با حذف برخی از علامت‌ها جدول را به شکل ایده‌آل درآوریم که تعریف آن در زیر آمده است:

الف) در هر سطر حداقل یکی از علامت‌های x یا o قرار دارد.

ب) در هیچ ستونی دو نوع علامت وجود ندارد.

ثابت کنید فهمیدن اینکه می توان با داشتن یک جدول اولیه به جدول ایده‌آل رسید یا خیر، در دسته مسائل NP-hard قرار دارد.(قطعا برای برخی حالات نمی‌توان به جدول ایده‌آل رسید.) (10 نمره امتیازی)



A solvable puzzle and one of its many solutions.   An unsolvable puzzle.

پاسخ:

برای اثبات NP-hard بودن مساله، کافی است مساله‌ی 3-SAT را به مساله‌ی داده شده کاهش دهیم. فرض کنید Φ یک 3CNF با m متغیر و n عبارت باشد؛ برای تبدیل این عبارت به ورودی مساله‌ی داده شده برای هر خانه‌ی جدول در سطر i و ستون j به صورت زیر عمل می‌کنیم:

- اگر متغیر a_j در عبارت iام از Φ وجود داشته باشد علامت x را در خانه (i, j) قرار می‌دهیم.
- اگر متغیر a_j در عبارت iام از Φ وجود داشته باشد علامت o را در خانه (i, j) قرار می‌دهیم.

اگر دو حالت بالا نبود، در خانه (i, j) هیچ علامتی قرار نمی‌دهیم.

ثابت می‌کنیم که می‌توان به جدول ایده‌آل رسید اگر و تنها اگر Φ دارای جواب باشد.

اگر Φ دارای جواب باشد دو حالت برای متغیر a_j وجود دارد: اگر a_j = True باشد آنگاه تمام علامت‌های o را از جدول پاک می‌کنیم و اگر a_j = False باشد، تمام علامت‌های x را از جدول پاک می‌کنیم؛ هر متغیر در حداقل یک عبارت حضور دارد پس هر ستون حداکثر یکی از انواع علامت‌ها را دارد و همچنین به دلیل اینکه هر عبارت حداقل یک متغیر که True باشد وجود دارد، هر ردیف حداقل دارای یک علامت است.

برای طرف دوم اثبات اگر بتوانیم به جدول ایده‌آل برسیم، مقدار متغیر a_j مطابق زیر تعیین می‌شود:

- اگر ستون j دارای علامت x باشد، a_j = True را قرار می‌دهیم.
- اگر ستون j دارای علامت o باشد، a_j = False را قرار می‌دهیم.

اگر حالات بالا نبود a_j می‌تواند هر مقداری داشته باشد.