مقدمه‌ای بر مسائل NP-Hard

۱ مسائل تصمیم گیری

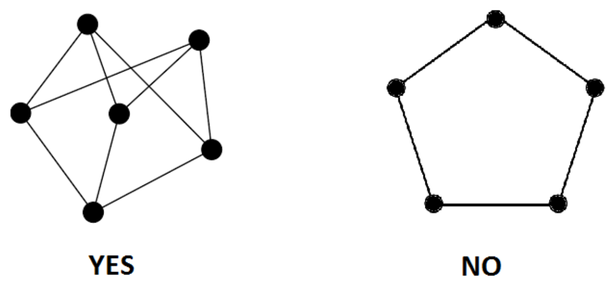
تعریف: تابع $f : \mathbb{N} \to \mathbb{N}$ را یک تابع چند جمله‌ای گویند اگر ثابت c وجود داشته باشد بطوریکه برای هر n داشته باشیم $f(n) \leq n^c$.

تعریف: طول رشته x را با $|x|$ نشان می‌دهیم.

تعریف: گوییم A یک مسئله تصمیم گیری (decision problem) است اگر برای هر ورودی جواب آن بله یا خیر باشد. در واقع می‌توانیم مسئله A را مانند یک مجموعه تصور کنیم که مسئله مورد نظر تشخیص عضویت در این مجموعه است.

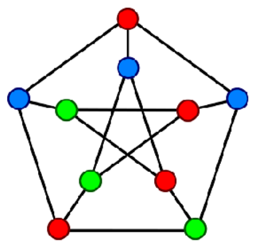
در زیر چند نمونه از مسائل تصمیم گیری آمده است. در این جزوه برای اسم مسائل از حروف بزرگ لاتین استفاده می‌کنیم.

- BIPARTITE: آیا گراف ورودی یک گراف دوبخشی است؟ به عبارت دیگر آیا می‌توان رئوس گراف را به دو قسمت افراز کرد بطوریکه هیچ یالی دو سرش در یک قسمت نباشد.

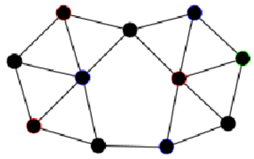


$$\text{BIPARTITE} = \{G \mid \text{یک گراف دوبخشی است } G\}$$

- TRI-PARTITE: آیا می‌توان با استفاده از ۳ رنگ، رئوس گراف ورودی را رنگ آمیزی کرد بطوریکه هیچ یالی دو سرش همرنگ نباشند.



YES



NO

$$\text{TRI-PARTITE} = \{G \mid \text{یک گراف ۳ رنگ پذیر است } G\}$$

- INDEPENDENT-SET: گراف G و عدد k داده شده. آیا G یک مجموعه رئوس با اندازه حداقل k دارد که هیچ یالی بین آنها نباشد.

$$\text{INDEPENDENT-SET} = \{(G, k) \mid \text{یک مجموعه مستقل با اندازه حداقل } k \text{ دارد } G\}$$

- VERTEX-COVER: گراف G و عدد k داده شده. آیا G یک مجموعه رئوس با اندازه حداقل k دارد که هیچ یالهای گراف را پوشش دهد. پوشش دادن یال به این معنی است که حداقل یکی از دو انتهای یال در مجموعه باشد

$$\text{VERTEX-COVER} = \{(G, k) \mid \text{یک پوشش راسی با اندازه حداقل } k \text{ دارد } G\}$$

- COMPOSITE: عدد صحیح و مثبت n داده شده. آیا n یک عدد مرکب است؟

$$\text{Yes Instances } = \{4, 6, 8, 9, 10, 12, 14, \cdots\}, \qquad \text{No Instances } = \{2, 3, 5, 7, 11, 13, \cdots\}$$

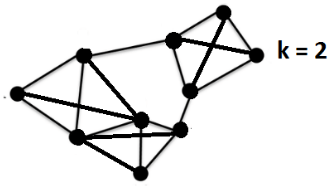
$$\text{COMPOSITE} = \{n \mid \text{یک عدد مرکب است } n\}$$

- FACTOR: زوج عدد (n, k) داده شده. آیا n یک مقسوم علیه اول کمتر از k دارد؟

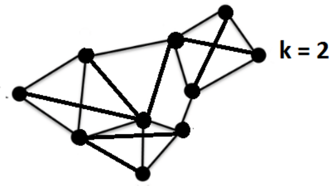
$$\text{Yes Instances } = \{(10, 4), (8, 5), (33, 12), \cdots\}, \qquad \text{No Instances } = \{(5, 3), (13, 13), (25, 4), \cdots\}$$

$$\text{FACTOR} = \{(n, k) \mid \text{یک مقسوم علیه کمتر از } k \text{ دارد } n\}$$

- MIN-CUT: گراف G و عدد k داده شده. آیا G یک برش با اندازه کمتر یا مساوی با k دارد؟



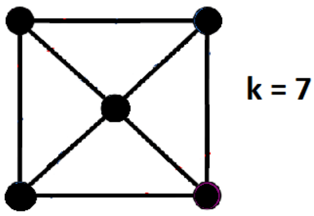
YES



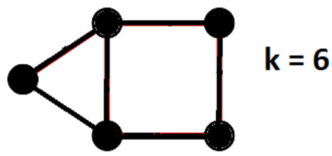
NO

MIN-CUT $= \{(G, k) \mid$ G یک برش با اندازه حداکثر k دارد$\}$

- MAX-CUT: گراف G و عدد k داده شده. آیا گراف ورودی یک برش با اندازه بیشتر یا مساوی با k دارد؟



YES

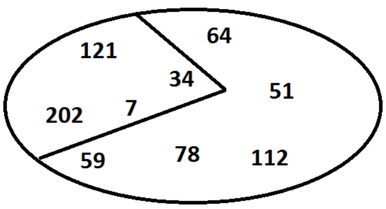


NO

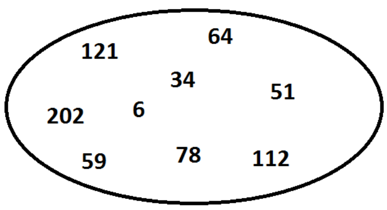
MAX-CUT $= \{(G, k) \mid$ G یک برش با اندازه حداقل k دارد$\}$

- PARTITION: مجموعه $A = \{x_1, \cdots, x_n\}$ شامل n عدد داده شده. آیا زیرمجموعه $S \subset [n] = \{1, \cdots, n\}$ وجود دارد بطوریکه

$$\sum_{i \in S} x_i = \sum_{i \in [n]/S} x_i$$



YES



NO

PARTITION $= \{A \mid$ A یک مجموعه شامل n عدد است که قابل افراز به دو زیرمجموعه با وزن مساوی است$\}$

- SAT: فرمول منطقی ϕ در قالب CNF (فرم نرمال عطفی) شامل n متغیر و m جمله داده شده است. فرمول یک ترکیب عطفی از جملات است و هر جمله ترکیب فصلی از متغیرها (و یا نقیضشان) است. به هر متغیر یا نقیضش یک لیترال گفته می‌شود.

$$\phi = \cdots \wedge \overbrace{(x_1 \vee \neg x_2 \vee \underbrace{\neg x_5}_{\text{یک لیترال}} \vee x_3)}^{\text{یک جمله}} \wedge \cdots$$

اینجا پرسش مربوطه این است: آیا می‌توان متغیرهای فرمول را با مقادیر True و False مقداردهی کرد بطوریکه ϕ ارزش True داشته باشد (صدق پذیر باشد)؟

$$\varphi = \left(\neg x_1 \vee \neg x_4\right) \wedge \left(x_1 \vee \neg x_2 \vee \neg x_3\right) \wedge \left(\neg x_2 \vee \neg x_3 \vee x_4 \vee x_1\right) \wedge \left(x_1\right)$$

YES

$$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2)$$

NO

$$\text{SAT} = \{\phi \mid \text{یک فرمول منطقی با فرمت } CNF \text{ است که صدق پذیر است } \phi\}$$

- UNSAT: مسئله UNSAT در واقع مکمل مسئله SAT است. اگر ϕ در مجموعه UNSAT باشد یعنی هیچکدام از مقداردهی ها برای متغیرهای ϕ ارزش کل عبارت را True نمی‌کند. تعریف زیر را داریم.

$$\text{UNSAT} = \{\phi \mid \text{یک فرمول منطقی با فرمت } CNF \text{ است که صدق پذیر نیست } \phi\}$$

- 2-SAT: حالت خاصی از مسئله SAT است با این محدودیت که تعداد لیترالهای داخل هر جمله دقیقا ۲ است.

$$\varphi = (\neg x_1 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2)$$

YES

$$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2)$$

NO

$$\text{2-SAT} = \{\phi \mid \text{یک فرمول منطقی با فرمت } 2CNF \text{ است که صدق پذیر است } \phi\}$$

۲ کلاس پیچیدگی P

تعریف: کلاس پیچیدگی P شامل همه مسائلی است که الگوریتمی با زمان چند جمله‌ای دارند. یعنی اگر n اندازه ورودی باشد، پیچیدگی زمانی الگوریتمی که مسئله را حل می‌کند بصورت $O(n^c)$ است وقتی c یک ثابت است.

در زیر نمونه‌هایی از مسائل داخل کلاس پیچیدگی P آورده شده است.

- BIPARTITE
- 2-SAT
- MIN-CUT
- COMPOSITE
- MAXIMUM MATCHING
- MAXIMUM FLOW
- SHORTEST PATH
- MINIMUM SPANNING TREE
- $\cdots$

۳ کلاس پیچیدگی NP

همانطور که گفتیم اینجا در مورد مسائل تصمیم گیری بحث می‌کنیم و هر مسئله تصمیم گیری را می‌توان مثل یک مجموعه به آن نگاه کرد. برای مثال مسئله BIPARTITE در واقع مجموعه همه گرافهای دوبخشی است.

$$\text{BIPARTITE} = \{G \mid \text{یک گراف دوبخشی است } G\}$$

بطور غیر رسمی مجموعه Z در NP است اگر برای هر $x \in Z$ یک **گواهی کوتاه و کارا** وجود داشته باشد. منظور از یک گواهی کوتاه و کارا برای اثبات وجود x در Z رشته y است که از روی آن بتوان سریعا (در زمان چند جمله‌ای) اطمینان حاصل کرد که x عضوی از Z است. سرشت این گواهی بستگی به مجموعه Z دارد.

- برای مثال اگر Z همه گرافهای همیلتونی باشد، گواهی اینکه گراف G عضو Z است می‌تواند ارائه یک دور همیلتونی در گراف G باشد. روشن است اگر یک دور ارائه شود می‌توان سریعا (در زمان چند جمله‌ای) راستی آزمایی کرد که آیا دور ارائه شده واقعا یک دور همیلتونی در گراف مورد نظر است یا نه.
- برای مثال دیگر اگر Z مجموعه همه گرافهای ۳ رنگ پذیر باشد، گواهی اینکه گراف G عضو Z است می‌تواند یک رنگ آمیزی برای گراف G باشد. روشن است که می‌توان در زمان چند جمله‌ای راستی آزمایی کرد که آیا رنگ آمیزی ارائه شده معتبر است یا نه.
- برای مثال دیگر، اگر Z مجموعه همه اعداد مرکب باشد، یک گواهی ساده برای عضویت n در مجموعه Z عددی مانند m است. می‌توان با انجام یک عمل تقسیم چک کرد که آیا m واقعا یک مقسوم علیه برای n است یا نه.

- برای مثال مجموعه‌ای که احتمالا در NP نیست، فرض کنید که Z مجموعه همه گرافهای غیر همیلتونی باشد. چگونه می‌توان یک گواهی برای همیلتونی نبودن گراف G ارائه کر که بتوان از روی آن همیلتونی نبودن G را سریعا راستی آزمایی کرد؟ اگر گواهی خود گراف باشد الگوریتمی نداریم که همیلتونی بودن را سریعا چک کند. اگر گواهی ارائه شده، همه جایگشتهای مختلف از رئوس گراف باشد، که تعداد آن خیلی زیاد خواهد بود و لذا یک گواهی کوتاه (با طول چند جمله‌ای) نخواهد بود. باور قوی بر این است که مجموعه همه گرافهای غیر همیلتونی عضو NP نیست.

حال یک تعریف رسمی برای کلاس NP ارائه می‌دهیم.

تعریف: مجموعه $Z \in NP$ اگر و فقط اگر تابع چندجمله‌ای $f : \mathbb{N} \to \mathbb{N}$ و الگوریتم A با زمان چند جمله‌ای وجود داشته باشد بطوریکه برای هر $x \in Z$ رشته y با طول حداکثر $f(|x|)$ وجود داشته باشد بطوریکه الگوریتم ورودی (x, y) را می‌پذیرد. اگر $x \notin Z$ آنگاه رشته y به طول حداکثر $f(|x|)$ وجود نداشته باشد که الگوریتم زوج (x, y) را بپذیرد.

همانطور که گفتیم اینجا y مانند یک شاهد، گواهی یا اثبات برای وجود x در مجموعه Z عمل می‌کند. در واقع رشته y کمک می‌کند که سریعتر اثبات کنیم که x عضوی از Z است. دقت کنید در تعریف کلاس NP این محدودیت اساسی را گذاشته‌ایم که الگوریتمی که وجود x در Z را با کمک y راستی آزمایی می‌کند، زمانش باید چند جمله‌ای باشد. روشن است با این محدودیت طول گواهی یعنی $|y|$ نیز باید کوتاه باشد و یک نسبت چند جمله‌ای با $|x|$ داشته باشد.

۴ نتایج تعریف کلاس NP

از بحثهایی که کردیم نتایج زیر بطور مستقیم بدست می‌آید. اثبات این نتایج نباید سخت باشد و بر عهده دانشجو است.

نتیجه: گزاره‌های زیر درست است.

- BIPARTITE $\in$ NP
- TRI-PARTITE $\in$ NP
- COMPOSITE $\in$ NP
- MIN-CUT $\in$ NP
- MAX-CUT $\in$ NP
- PARTITION $\in$ NP
- SAT $\in$ NP
- 2-SAT $\in$ NP

نتیجه: P $\subseteq$ NP

اثبات. فرض کنید $Z \in$ P. پس الگوریتم A با زمان چند جمله‌ای وجود دارد که تشخیص می‌دهد $x \in A$ یا خیر. توجه کنید این الگوریتم نیازی به کمک و ارائه گواهی برای عضویت x در A ندارد و خودش این کار را انجام می‌دهد. پس روشن است که $Z \in NP$ هم درست است.

مسئله باز: UNSAT $\subseteq$ NP ؟

مسئله باز: NP $\subseteq$ P ؟

۵ تقلیل چند جمله‌ای و مسائل NP-Complete

هر کلاس پیچیدگی مجموعه‌ای از مسائل تصمیم گیری است. در مباحثی که مطرح کردیم نمونه‌هایی از کلاس پیچیدگی NP و P را دیدیم. یک سوال اساسی اینجا این است برای یک کلاس پیچیدگی آیا مسئله‌ای وجود دارد که نقش محوری را داشته باشد؟ به عبارت دیگر آیا مسئله‌ای در کلاس NP یا P وجود دارد که سخترین مسئله در کلاس پیچیدگی خودش باشد؟ طبیعتا اینجا منظور از سخت بودن، یعنی سختی از نظر محاسباتی است. باید منظورمان را دقیقتر بیان کنیم. اینجا از مفهوم تقلیل با زمان چند جمله‌ای Polynomial time reduction استفاده می‌کنیم.

۱.۵ تقلیل در زمان چند جمله‌ای

تعریف: مسئله A قابل تقلیل به مسئله B است اگر الگوریتمی برای حل مسئله B داشته باشیم بتوانیم با استفاده از آن مسئله A را حل کنیم. در این حالت می‌نویسیم

$$A \leq B$$

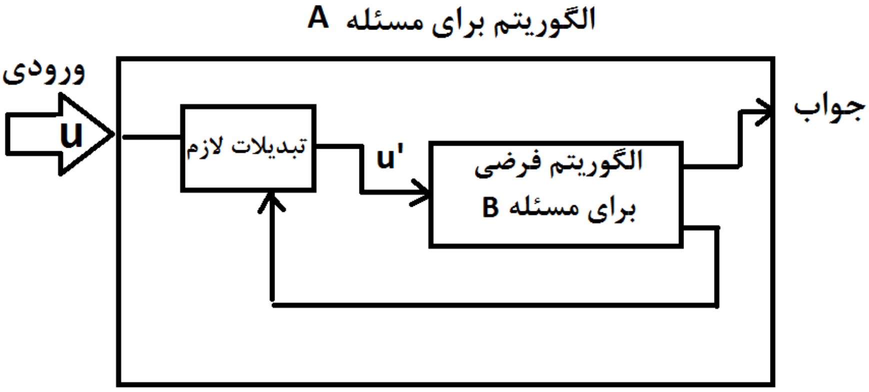
استفاده از نماد کوچکتر یا مساوی بدین معنی است اگر بتوانیم A را به B تقلیل دهیم، یعنی نشان داده‌ایم مسئله A از مسئله B سختتر نیست (سختی اش بیشتر نیست).

ما در این کلاسها قبلا مواردی از تقلیل داشته‌ایم. مثلا موقعی که می‌خواستیم تطابق بیشینه Maximum Matching در گراف دوبخشی را بدست بیاوریم مسئله را به شار بیشینه Maximum Flow تقلیل دادیم (تبدیل کردیم) و از آن کمک گرفتیم.

تعریف: مسئله A قابل تقلیل در زمان چند جمله‌ای به مسئله B است اگر بتوانیم با استفاده از یک الگوریتم که در زمان چند جمله‌ای کار می‌کند و فراخوانی هایی از الگوریتم برای حل مسئله B ، مسئله A را حل کنیم. در این حالت می‌نویسیم

$$A \leq_p B$$

دقت کنید که الگوریتم برای حل مسئله B فرضی است. ما اینجا فقط بصورت جعبه سیاه از آن استفاده می‌کنیم. علاوه بر این، تعداد دفعاتی که از این جعبه سیاه استفاده می‌کنیم باید یک تابع چند جمله‌ای باشد (نسبت به اندازه ورودی مسئله).



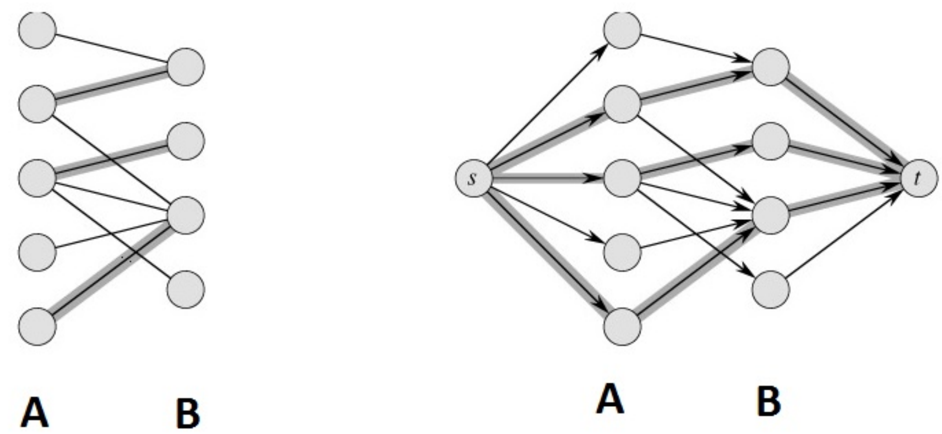
۲.۵ مثالهایی از تقلیل در زمان چند جمله‌ای

برای درک بهتر مفهومی که بیان کردیم چند مثال می‌آوریم.

لم : مسئله Maximum Matching در گرافهای دوبخشی قابل تقلیل در زمان چند جمله‌ای به مسئله Maximum s-t Flow است. به عبارت دیگر

$$\text{Bipartite-Maximum-Matching} \leq_p \text{Maximum-s-t-Flow}$$

اثبات: می‌خواهیم یک تطابق بیشینه در گراف دوبخشی $G = (A \cup B, E)$ پیدا کنیم. دو راس s و t را به گراف اضافه می‌کنیم. از راس s به همه رئوس A یک یال جهت دار قرار می‌دهیم (جهت به سمت A باشد.) همچنین از رئوس B به راس t یک یال جهت دار (جهت به سمت B) قرار می‌دهیم. جهت یالهای بین A و B هم همه به سمت B هستند. ظرفیت همه یالها را 1 تعریف می‌کنیم. حال الگوریتم شار بیشینه برای پیدا کردن بیشترین شار از s به t را فراخوانی می‌کنیم. طبق مشاهداتی که قبلا انجام دادیم، مقدار شار بیشینه از s به t دقیقا برابر با مقدار تطابق بیشینه بین A و B است. از طرف دیگر، یالهایی که شار مثبت دارند همان یالهای تطابق بیشینه هستند. تمام مراحلی که انجام دادیم در زمان چند جمله‌ای قابل انجام بود. همچنین فقط یک بار الگوریتم شار بیشینه را فراخوانی کردیم. پس نتیجه می‌گیریم که مسئله تطابق بیشینه در گراف دوبخشی قابل تقلیل در زمان چندجمله‌ای به مسئله شار بیشینه در شبکه است. □

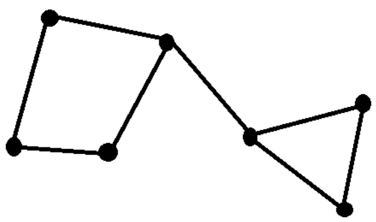


لم مسئله Global Min-Cut قابل تقلیل در زمان چند جمله‌ای به مسئله Maximum s-t Flow است. به عبارت دیگر

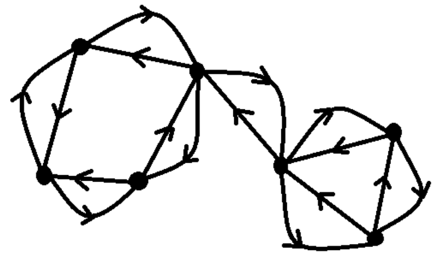
$$\text{Global-Min-Cut} \leq_p \text{Maximum-s-t-Flow}$$

اثبات: در مسئله Global Min-Cut می‌خواهیم در گراف داده شده $G = (V, E)$ که جهت دار هم نیست برشی را پیدا کنیم که کمترین تعداد یال را قطع کند. به عبارت دیگر، کمترین تعداد یال را از گراف G حذف کنیم بطوریکه گراف غیرهمبند شود. ابتدا نشان می‌دهیم اگر بخواهیم برای دو راس مشخص a و b برشی را پیدا کنیم که a را از b جدا کند و کمترین تعداد یال را قطع کند، کافی است که اول گراف جهت دار $\vec{G}$ از روی G بسازیم. برای این کار اگر یالی بین رئوس u و v باشد دو یال (در هر دو جهت بین u و v) قرار می‌دهیم. ظرفیت همه یالها را هم 1 تعریف می‌کنیم. توجه کنید که برش کمینه $a - b$ در گراف $\vec{G}$ دقیقا همان برش کمینه $a - b$ در گراف G است.

لذا کافی است که از الگوریتم شار بیشینه برای پیدا کردن برش کمینه $a-b$ در گراف $\vec{G}$ استفاده کنیم. برای پیدا کردن global min-cut کافی است که این برش کمینه را برای هر زوج رئوس a و b پیدا کنیم (در حقیقت تعداد تکرارهای کمتری لازم است اما فعلا همین برای ما کفایت می‌کند.) در نتیجه، با فرض اینکه گراف ورودی n راس دارد، با استفاده از یک الگوریتم در زمان چند جمله‌ای و تعداد حداکثر n^2 فراخوانی الگوریتم شار بیشینه توانستیم مسئله global min-cut را حل کنیم. نتیجه اینکه مسئله global min-cut در زمان چند جمله‌ای قابل تقلیل به مسئله شار بیشینه در شبکه است. □



G



$\vec{G}$

۶ مسائل NP-Complete

تعریف: A یک مسئله NP-Complete است اگر و فقط اگر

- $A \in NP$
- $\forall B \in NP,\ B \leq_p A$

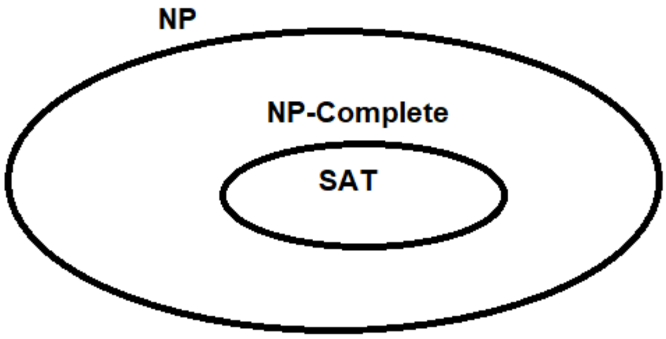
به عبارت دیگر مسئله A یک مسئله NP-Complete است اگر و فقط اگر A خودش عضوی از NP باشد و هر مسئله در NP در زمان چند جمله‌ای قابل تقلیل به مسئله A باشد.

استفان کوک Stephen Cook و لئونید لوین Leonid Levin بطور مستقل در اوایل دهه هفتاد میلادی اثبات کردند که مسئله SAT یک مسئله NP-Complete است. به عبارت دیگر، آنها نشان دادند که هر مسئله در کلاس NP را می‌توان در زمان چند جمله‌ای به مسئله SAT تقلیل داد. اثبات این قضیه از طریق تبدیل ماشین تورینگ غیرقطعی که در زمان چند جمله‌ای کار می‌کند به یک فرمول SAT انجام می‌شود و جزئیات آن بسیار و خارج از حوصله این کلاس است.

قضیه: کوک و لوین ۱۹۷۱.

$$\forall A \in NP,\ A \leq_p \text{SAT}$$

نتیجه: SAT $\in$ NP-Complete



با استفاده از قضیه کوک و لوین و ایده تقلیل چند جمله‌ای می‌توان مسائل دیگر را به جمع مسائل NP-Complete اضافه کرد.

قضیه: ۳-SAT $\in$ NP-Complete
اثبات: مسئله ۳-SAT حالت خاصی از مسئله SAT است با این محدودیت که در فرمول داده شده، تعداد لیترالهای داخل هر جمله ۳ است. روشن است که ۳-SAT یک مسئله NP است. نشان می‌دهیم

$$\text{SAT} \leq_p \text{۳-SAT}$$

برای این منظور فرمول ϕ که در آن ممکن است جملاتی وجود داشته باشد که بیشتر یا کمتر از ۳ لیترال دارند را به فرمول ϕ' تبدیل می‌کنیم بطوریکه ϕ' هر جمله‌اش دقیقا ۳ لیترال داشته باشد و علاوه بر این فرمول ϕ صدق پذیر باشد اگر و فقط اگر فرمول ϕ' صدق پذیر باشد.

فرض کنید در فرمول ϕ جمله c وجود دارد که ۲ لیترال دارد.

$$\phi = \cdots \wedge \overbrace{(x_1 \vee x_2)}^{c} \wedge \cdots$$

با اضافه کردن متغیر جدید z می‌توانیم جمله c را با عبارت c' جایگزین کنیم بطوریکه به صدق پذیر بودن لطمه‌ای وارد نکند.

$$\cdots \wedge \overbrace{(x_1 \vee x_2)}^{c} \wedge \cdots \qquad \Rightarrow \qquad \cdots \wedge \overbrace{(x_1 \vee x_2 \vee z) \wedge (x_1 \vee x_2 \vee \neg z)}^{c'} \wedge \cdots$$

با همین ترفند اگر جمله‌ای داشتیم که فقط ۱ لیترال داشته باشد، می‌توانیم آن را به جملاتی با ۲ لیترال تبدیل کنیم و سپس جملات حاصل را هر کدام به ۳ لیترال تبدیل کنیم.

$$\cdots \wedge \overbrace{(x_1)}^{c} \wedge \cdots \;\Rightarrow\; \cdots \wedge \overbrace{(x_1 \vee z_1) \wedge (x_1 \vee \neg z_1)}^{c'} \wedge \cdots \;\Rightarrow$$

$$\cdots \wedge \overbrace{(x_1 \vee z_1 \vee z_2) \wedge (x_1 \vee z_1 \vee \neg z_2) \wedge (x_1 \vee \neg z \vee z_3) \wedge (x_1 \vee \neg z \vee \neg z_3)}^{c''} \wedge \cdots$$

جملات با بیشتر از ۳ لیترال را بصورت بازگشتی به دنباله‌ای از جملات با ۳ لیترال تبدیل می‌کنیم. فرض کنید که یک جمله با k لیترال داریم بطوریکه $k \geq 4$.

$$\phi = \cdots \wedge \overbrace{(x_1 \vee x_2 \vee x_3 \cdots \vee x_{k-1} \vee x_k)}^{c} \wedge \cdots$$

متغیر جدید z را تعریف می‌کنیم

$$z \iff (x_3 \vee \cdots \vee x_k)$$

جایگزین می‌کنیم

$$\phi' = \cdots \wedge (x_1 \vee x_2 \vee z) \wedge (z \iff (x_3 \vee \cdots \vee x_k)) \wedge \cdots$$

معادل است با

$$\cdots \wedge (x_1 \vee x_2 \vee z) \wedge (z \to (x_3 \vee \cdots \vee x_k)) \wedge ((x_3 \vee \cdots \vee x_k) \to z) \wedge \cdots$$

معادل است با

$$\cdots \wedge (x_1 \vee x_2 \vee z) \wedge (\neg z \vee (x_3 \vee \cdots \vee x_k)) \wedge (\neg(x_3 \vee \cdots \vee x_k) \vee z) \wedge \cdots$$

معادل است با

$$\cdots \wedge (x_1 \vee x_2 \vee z) \wedge (\neg z \vee x_3 \vee \cdots \vee x_k) \wedge ((\neg x_3 \wedge \cdots \wedge \neg x_k) \vee z) \wedge \cdots$$

معادل است با

$$\cdots \wedge (x_1 \vee x_2 \vee z) \wedge (\neg z \vee x_3 \vee \cdots \vee x_k) \wedge ((\neg x_3 \vee z) \wedge \cdots \wedge (\neg x_k \vee z)) \wedge \cdots$$

معادل است با

$$\cdots \wedge (x_1 \vee x_2 \vee z) \wedge (\neg z \vee x_3 \vee \cdots \vee x_k) \wedge (\neg x_3 \vee z) \wedge \cdots \wedge (\neg x_k \vee z) \wedge \cdots$$

بدین ترتیب یک جمله با k لیترال تبدیل به ترکیب عطفی یک جمله با ۳ لیترال و یک جمله با $k-1$ لیترال و به تعداد $k-2$ جمله ۲ لیترالی شد. جملات ۲ لیترالی با استفاده از مشاهدات بالا قابل تبدیل به $2(k-1)$ جمله ۳ لیترالی هستند. اگر همین پروسه را بصورت بازگشتی ادامه دهیم در نهایت جملات حاصل همه ۳ لیترال خواهند داشت.

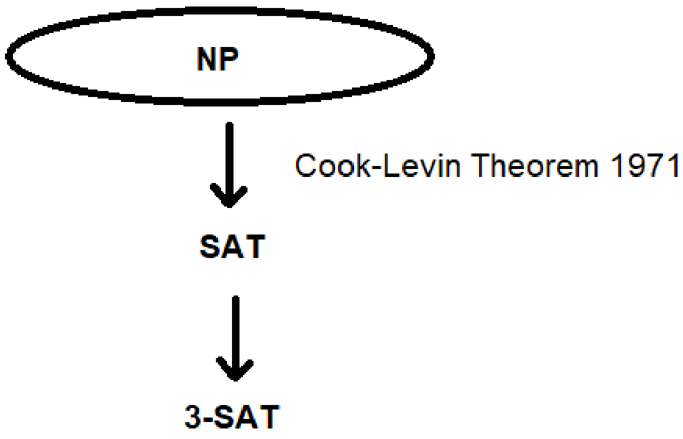
بطور کلی فرض کنید $C(k)$ در نهایت تعداد جملاتی باشد که از تبدیل یک جمله k لیترالی به جملات ۳ لیترالی بدست می آید. داریم

$$C(k) = \begin{cases} C(k-1) + 2(k-1) + 1 & k \geq 4 \\ 2 & k = 2 \\ 4 & k = 1 \end{cases}$$

می‌توان نشان داد که

$$C(k) = O(k^2)$$

با فرض اینکه فرمول اولیه ϕ به تعداد m جمله و n متغیر داشته باشد، فرمول نهایی ۳-SAT که بدست می‌آید حداکثر $O(mn^2)$ جمله خواهد داشت. دقت کنید هر جمله در ϕ حداکثر n لیترال دارد. نتیجه می‌گیریم که زمان تبدیل ϕ به ϕ' یک تابع چند جمله‌ای است و تقلیل در زمان چند جمله‌ای قابل انجام است. $\square$



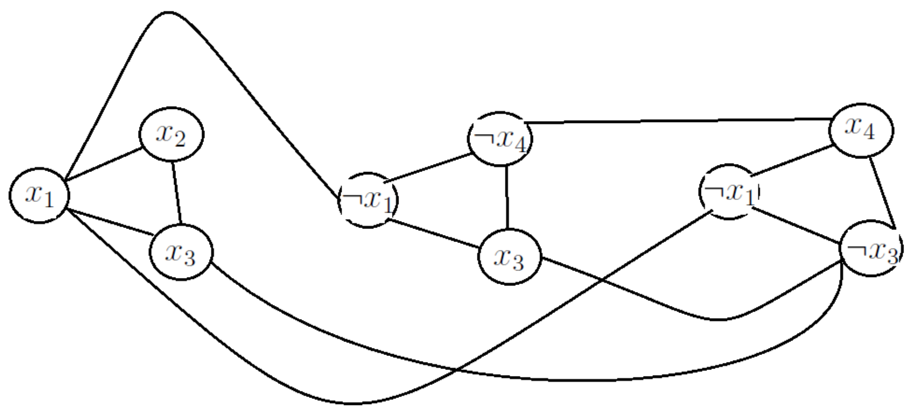
قضیه: INDEPENDENT-SET $\in$ NP-Complete

اثبات: روشن است که INDEPENDENT-SET $\in NP$. نشان می‌دهیم

$$\text{۳-SAT} \leq_p \text{INDEPENDENT-SET}$$

با داشتن فرمول منطقی ϕ که فرمت ۳-CNF دارد و به تعداد m جمله دارد، گراف G_ϕ را می‌سازیم بطوریکه اگر ϕ صدق پذیر باشد گراف G_ϕ یک مجموعه مستقل با اندازه m داشته باشد و اگر ϕ صدق پذیر نباشد گراف G مجموعه مستقلی با اندازه m نداشته باشد. برای ساختن چنین گرافی، در قدم اول برای هر لیترال در عبارت ϕ یک راس در گراف G_ϕ قرار می‌دهیم. در قدم دوم، برای هر جمله $(u \vee v \vee w)$ بین رئوس مربوط به لیترالهای آن یال قرار می‌دهیم. در واقع هر جمله در ϕ یک با یک مثلث در گراف G_ϕ داریم. در قدم سوم بین هر رئوس مربوط به هر لیترال و نقیض آن (در صورت وجود) یال قرار می‌دهیم.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_4 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee \neg x_3)$$

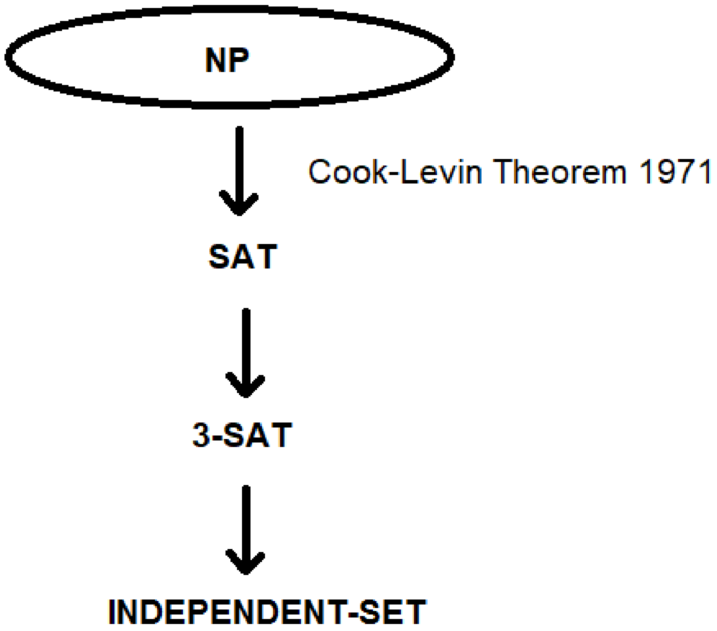


فرض کنید ϕ صدق پذیر باشد. نشان می‌دهیم که G_ϕ یک مجموعه مستقل با اندازه m دارد. چون ϕ صدق پذیر است پس در هر جمله آن باید حداقل یک لیترال با ارزش True وجود داشته باشد. در هر جمله، راس مربوط به یکی از لیترالهای True آن را انتخاب می‌کنیم. فرض کنید S مجموعه رئوس انتخاب شده باشد. چون m جمله

داریم پس $|S| = m$. ادعا می‌کنیم که S یک مجموعه مستقل است. از هر مثلث یک راس را انتخاب کرده‌ایم. پس اگر یالی در S باشد باید بین دو لیترال نقیض هم باشد. اما این امکان ندارد چون دو لیترال که نقیض هم باشند نمی‌توانند هر دو ارزش True داشته باشند.

حال فرض کنید که ϕ_G یک مجموعه مستقل با اندازه m داشته باشد. پس از هر مثلث در ϕ_G که متناظر با جملات هستند یک راس در مجموعه مستقل است (بدیهی است دو راس از یک مثلث نمی‌تواند در مجموعه مستقل باشد.) ارزش لیترالهای متناظر با رئوس مجموعه مستقل را True قرار می‌دهیم. این هر جمله ϕ را True می‌کند و یک ارزشدهی معتبر است چون دو لیترالی که نقیض هم هستند هر دو انتخاب نشده اند. نتیجه می‌گیریم که ϕ صدق پذیر است.

در پایان کافی است توجه کنیم که تبدیل ϕ به گراف ϕ_G در زمان چند جمله‌ای قابل انجام است. این ادعای ما را ثابت می‌کند. □



قضیه: VERTEX-COVER $\in$ NP-Complete

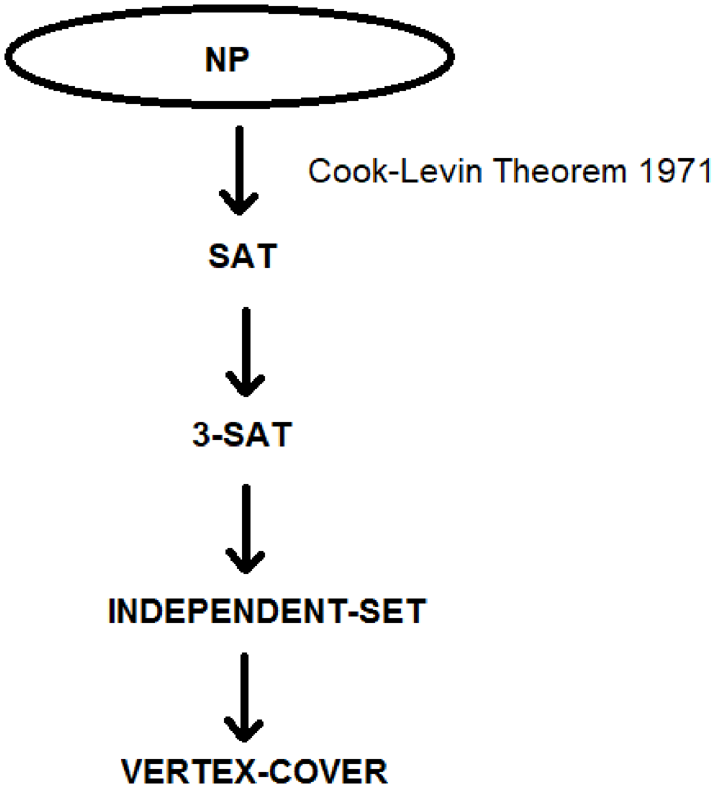
اثبات: روشن است که VERTEX-COVER $\in NP$. نشان می‌دهیم

$$\text{INDEPENDENT-SET} \leq_p \text{VERTEX-COVER}$$

برای این تقلیل لازم نیست کار خاصی انجام دهیم. در واقع یک رابطه نزدیک بین اندازه مجموعه مستقل یک گراف و اندازه پوشش راسی آن وجود دارد. گزاره زیر را داریم که اثبات آن را واگذار به خواننده می‌کنیم.

گزاره گراف G یک مجموعه مستقل با اندازه k دارد اگر و فقط اگر G یک پوشش راسی با اندازه $n-k$ داشته باشد.

با توجه به گزاره بالا، اگر بخواهیم جواب این سوال را بدانیم که آیا G یک مجموعه مستقل با اندازه حداقل k دارد یا نه کافی است بپرسیم آیا G یک پوشش راسی با اندازه حداکثر $n-k$ دارد یا نه. لذا یک الگوریتم برای مسئله VERTEX-COVER یک الگوریتم برای INDEPENDENT-SET را نتیجه می‌دهد. □


NP
Cook-Levin Theorem 1971
SAT
3-SAT
INDEPENDENT-SET
VERTEX-COVER