Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 13

Aufgabe H13.1. (*There are four colours! Four!*) 7 Punkte

Achtung: Geben Sie diese Aufgabe über das Dashboard ab.

Sei $G = (V, E)$ ein ungerichteter Graph. Für $k \in \mathbb{N}$ ist eine *k-Färbung* von G eine (totale) Funktion $f : V \to \{0, ..., k-1\}$ mit $f(u) \neq f(v)$ für alle $\{u, v\} \in E$. Es haben also alle benachbarten Knoten unterschiedliche Farben. Das *k-Färbbarkeitsproblem* (k-COL) ist folgendermaßen definiert, für $k \in \mathbb{N}$.

Eingabe: Ungerichteter Graph $G = (V, E)$
Ausgabe: Existiert eine k-Färbung für G ?

Unser Ziel ist es, zu zeigen, dass 3-COL ein NP-vollständiges Problem ist. Als Zwischenschritt betrachten wir eine Variante des Problems, bei dem eine Teilfärbung bereits vorgegeben ist. Für eine k-Färbung f ist eine *Teilfärbung* $f' : V \to \{0, ..., k-1\}$ eine beliebige partielle Funktion mit $f'(v) \in \{f(v), \bot\}$ für alle v. Das *partielle k-Färbbarkeitsproblem* (Partial-k-COL) ist nun

Eingabe: Ungerichteter Graph $G = (V, E)$, partielle Funktion $f' : V \to \{0, ..., k-1\}$
Ausgabe: Existiert eine k-Färbung f für G, mit Teilfärbung f' ?

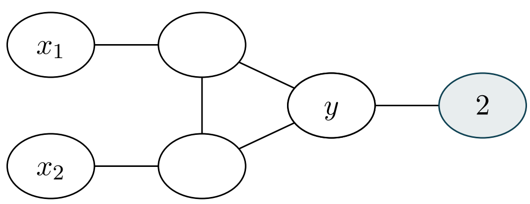
Dies werden wir hier nicht zeigen, aber Partial-k-COL $\leq_p$ k-COL und k-COL $\leq_p$ Partial-k-COL gelten – die Probleme sind also bezüglich Polynomialzeitreduktionen gleich schwierig.

Nun wollen wir SAT auf Partial-3-COL reduzieren, indem wir die Farben benutzen, um logische Gatter zu simulieren. Wir verwenden Farben $\{0, 1, 2\}$, wobei 0 und 1 jeweils für wahr und falsch stehen, und 2 eine Farbe ist, die wir intern in unserem Schaltkreis verwenden, allerdings nicht als Eingabe oder Ausgabe. Zusätzlich betrachten wir mehrdeutige Gatter, die bei bestimmten Eingaben sowohl 1 als auch 0 ausgeben können.

Seien nun $n \in \mathbb{N}$ und $g : \{0, 1\}^n \to \mathcal{P}(\{0, 1\})$ beliebig. Wir wollen eine Instanz von Partial-k-COL konstruieren, also einen partiell gefärbten Graphen, die einen Schaltkreis darstellt, der g simuliert. Dazu verwenden wir Knoten $x_1, ..., x_n$ und y, wobei $x_1, ..., x_n$ die Eingabe abbilden (also mit beliebigen Farben in $\{0, 1\}$ gefärbt werden können), und y dann eine Farbe entsprechend g erhalten muss.

Formal sei $G = (V \cup \{x_1, ..., x_n, y\}, E)$ ein Graph, und $f' : V \to \{0, 1, 2\}$ eine partielle Funktion. Wir sagen, dass (G, f') ein *g-Gatter* ist, wenn für jede Funktion $I : \{x_1, ..., x_n\} \to \{0, 1\}$ eine 3-Färbung f von G existiert, sodass f' und I Teilfärbungen von f sind. Die Menge der Farben, die solche f dem Knoten y zuordnen können, muss genau $g(I(x_1), ..., I(x_n))$ entsprechen, es soll also $\{f(y) : f\} = g(I(x_1), ..., I(x_n))$ gelten.

Die folgende Instanz ist z.B. ein `CHOICE`-Gatter, wobei `CHOICE`$(x_1, x_2) := \{x_1, x_2\}$ für $x_1, x_2 \in \{0, 1\}$. (Die Ausgabe entspricht also einer der Eingaben, darf sich aber aussuchen, welcher.)



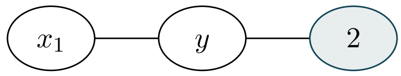
Konstruieren Sie ein AND-Gatter, mit $\texttt{AND}(x_1, x_2) := \{x_1 \wedge x_2\}$. Wie Sie dieses Gatter erzeugen, ist Ihnen überlassen, aber folgende Zwischenschritte mögen hilfreich sein:

(1) Konstruieren Sie ein NOT-Gatter, wobei $\texttt{NOT}(x_1) := \{\neg x_1\}$ für $x_1 \in \{0, 1\}$.

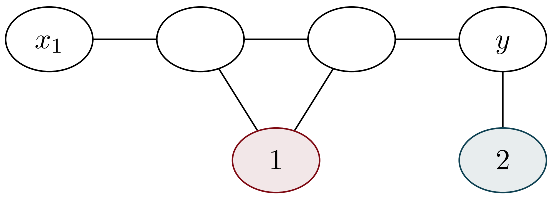
(2) Sei $\texttt{weakNOT}(x_1) := \texttt{NOT}(x_1) \cup \{1\}$. Konstruieren Sie ein weakNOT-Gatter.

Anmerkung: Über das NOT-Gatter und das AND-Gatter lässt sich die Reduktion SAT $\leq_p$ Partial-3-COL dann beweisen.

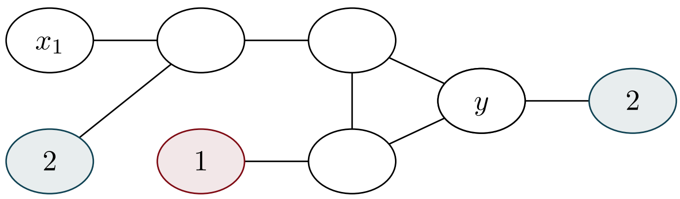
Lösungsskizze. **(1)**



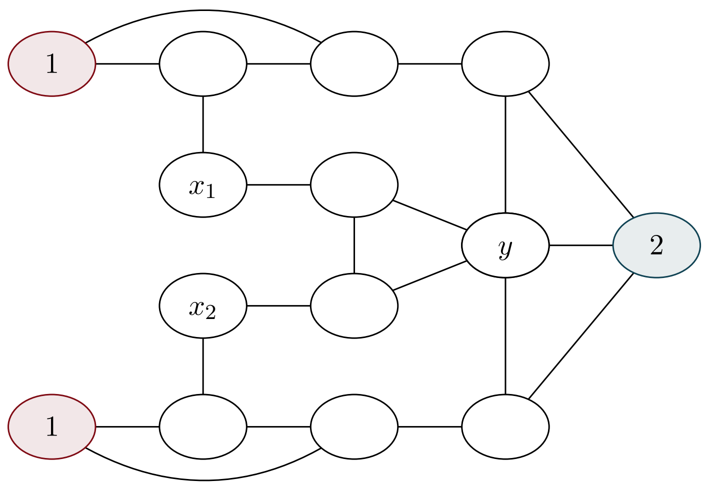
(2) Hier geben wir zwei mögliche Lösungen an. Zunächst kann man das Gatter folgendermaßen auf direktem Wege bauen:



Wenn $x_1 = 1$, dann muss der nächste Knoten 2 sein, der nächste 1 und y schließlich 0. Für $x_1 = 0$ können wir den nächsten Knoten als 1 wählen, den folgenden als 2 und y dann als 0 oder 1. Alternativ lässt es sich auch über das CHOICE-Gatter konstruieren, da $\texttt{weakNOT}(x_1) = \texttt{CHOICE}(\texttt{NOT}(x_1), 1)$, also als



AND Wenn wir nur $\texttt{CHOICE}(x_1, x_2)$ verwenden, sind wir für die Fälle $x_1 = x_2$ bereits fertig. Für die Fälle $x_1 \neq x_2$ müssen wir dann allerdings dafür sorgen, dass die Ausgabe 0 und nicht 1 ist. Hierfür können wir zwei weakNot-Gatter verwenden: Für x_1 und x_2 bauen wir jeweils ein solches Gatter und verbinden dessen Ausgabe mit der Ausgabe des CHOICE-Gatters. Wenn z.B. $x_1 = 1$, dann kann das weakNot-Gatter einen beliebigen Wert annehmen und beeinflusst die Ausgabe des CHOICE-Gatters nicht. Wenn allerdings $x_1 = 0$, dann hat das weakNot-Gatter Ausgabe 1, und das CHOICE-Gatter muss somit Ausgabe 0 haben.



Quizaufgabe H13.2. (*~~Auf Wiedersehen!~~ Tschüss!*) 18 Punkte

Theo hat bald die Grundschule erfolgreich abgeschlossen, und wechselt auf eine weiterführende Schule. Dora freut sich riesig für ihn, und hat ihm zur Feier des Tages eine Sammlung von Fragen aus allen seinen Prüfungen mitgebracht, damit Theo sich an die Dinge erinnern kann, die er gelernt hat. Der ist natürlich ganz außer sich vor Freude.

Anmerkung: Die folgenden Aufgaben sind aus den Klausuren des letzten Jahres übernommen, mit kleinen Änderungen.

(a) Geben Sie einen RE r an, für eine Sprache L mit $L = \{ab\}L \cup \{c\}$.

(b) Welche Sprachen können von kontextfreie Grammatiken über einem Alphabet Σ erzeugt werden, bei denen die rechte Seite jeder Produktion Länge 1 hat? Beachten Sie $|\varepsilon| = 0$.

(c) Seien $M_1 = (Q, \Sigma, \delta, q_0, F_1)$, $M_2 = (Q, \Sigma, \delta, q_0, F_2)$ zwei beliebige DFAs, die sich nur in ihren Finalzuständen unterscheiden, mit $F_1 \neq F_2$ und $L(M_1) = L(M_2)$. Gibt es solche M_1, M_2, in denen alle Zustände erreichbar sind? Wenn ja, geben Sie ein Beispiel an, wenn nein, begründen Sie dies kurz.

(d) Gibt es eine kontextfreie Grammatik G, sodass jeder PDA für $L(G)$, der über Finalzustände akzeptiert, mindestens 2 Zustände hat? Falls ja, geben Sie ein solches G an und begründen kurz, dass es die geforderte Eigenschaft erfüllt. Falls nein, begründen Sie kurz, wieso dies unmöglich ist.

(e) Sei $M \subset \mathbb{N}$ endlich mit $M \neq \emptyset$. Geben Sie den minimalen DFA für $L := \{a^n : n \in M\}$ über dem Alphabet $\Sigma := \{a\}$ formal an.

Entscheiden Sie, ob die folgenden Aussagen jeweils wahr sind, **unter der Annahme** $\mathsf{P} \neq \mathsf{NP}$. Falls ja, geben Sie eine *kurze* Begründung an, falls nein, ein Gegenbeispiel.

(f) Wenn $L \subseteq \{a, b\}^*$ entscheidbar ist, dann ist L^a entscheidbar. Hier bezeichnet $L^a := \{w : aw \in L\}$ die Residualsprache bezüglich a von L.

(g) Seien M_1, M_2 beliebige Turingmaschinen, und L eine Sprache mit $L(M_1) \subseteq L \subseteq L(M_2)$. Dann ist L semi-entscheidbar.

(h) Sei L entscheidbar und M eine NTM. Dann ist $L(M) \setminus L$ rekursiv aufzählbar.

Lösungsskizze.

(a) $(ab)^*c$ (s. Ardens Lemma).

(b) Genau die Sprachen in $\mathcal{P}(\Sigma) = \{S : S \subseteq \Sigma\}$.

(c) Nein, alle Zustände in $(F_1 \cup F_2) \setminus (F_1 \cap F_2)$ sind nicht erreichbar, denn falls $q = \delta(q_0, w)$ für einen solchen Zustand q und ein Wort w, wäre w nur in genau einer der Sprachen enthalten und diese damit unterschiedlich.

(d) Ja, gibt es: $S \to a$. Da $\varepsilon \notin L(G) = \{a\}$, darf der Startzustand nicht akzeptierend sein, aber da $L(G) \neq \emptyset$ muss es einen akzeptierenden Zustand geben.

(e) Mit $m := \max M$ haben wir Zustände $\{0, ..., m+1\}$, $\delta(q, a) := q + 1$ für $q \leq m$ und $\delta(m + 1, a) := m + 1$. Die Finalzustände sind genau M, Startzustand ist 0.

(f) Wahr, um $w \in L^a$ für ein Wort w zu entscheiden, müssen wir nur überprüfen, ob aw in L enthalten ist.

(g) Falsch, sei z.B. M_1 eine TM, die nie akzeptiert, M_2 eine TM, die immer akzeptiert, und L eine beliebige nicht semi-entscheidbare Sprache, beispielsweise $L := \mathcal{H}_{\Sigma^*}$. Dann gilt $L(M_1) = \emptyset$ und $L(M_2) = \Sigma^*$, aber natürlich $\emptyset \subseteq L \subseteq \Sigma^*$.

(h) Wahr. Rekursiv aufzählbar ist äquivalent zu semi-entscheidbar. Für eine Eingabe w überprüfen wir, ob $w \in L$; falls ja, gehen wir in eine Endlosschleife, falls nein, lassen wir M laufen.

Die Hausaufgabenzeit neigt sich mal wieder dem Ende — wir hoffen, Ihnen mit unseren Geschichten beim Bearbeiten der Aufgaben zumindest ein wenig Freude bereitet zu haben. Nun verabschieden wir uns und wünschen Ihnen fröhliche vorlesungsfreie Wochen!

—Ihr fiktionales THEO-Team

Theo & Dora