Logic in Computer Science

LiCS

Pamela Fleischmann

fpa@informatik.uni-kiel.de

October 14, 2019

Kiel University

Dependable Systems Group

Decision-SAT

Definition

Given a formula in CNF, the **decision-SAT-problem** is to decide whether there exists a satisfying substitution for the variables

Decision-SAT

Definition

Given a formula in CNF, the **decision-SAT-problem** is to decide whether there exists a satisfying substitution for the variables

Theorem (Cook)

SAT $\in$ NPC

Proof. omitted

3 – SAT

Definition

3 – SAT is the restriction of SAT to 3 literals per clause

3 – SAT

Definition

3 – SAT is the restriction of SAT to 3 literals per clause

Theorem

3 – SAT $\in$ NPC

3 – SAT

Definition

3 – SAT is the restriction of SAT to 3 literals per clause

Theorem

3 – SAT ∈ NPC

Proof:

- polynomial-time verifier can check the satisfiability

3 – SAT

Definition

3 – SAT is the restriction of SAT to 3 literals per clause

Theorem

3 – SAT ∈ NPC

Proof:

- polynomial-time verifier can check the satisfiability
- reduction-plan: SAT $\leq_p$ 3 – SAT

Proof of 3 – SAT $\in$ NPC

Given $C_i = \bigvee_{i \in [k]} y_i$ in F with $y_i \in \{x_i, \overline{x}_i\}$ and $a_i = \bigvee_{j=i+1}^{n} y_j$

- $f : \Sigma^* \rightarrow \Sigma^*; F \mapsto F_3$ with

Proof of 3 – SAT ∈ NPC

Given $C_i = \bigvee_{i\in[k]} y_i$ in F with $y_i \in \{x_i, \overline{x}_i\}$ and $a_i = \bigvee_{j=i+1}^{n} y_j$

- $f : \Sigma^* \to \Sigma^*; F \mapsto F_3$ with
- $F_3 = \bigwedge_{i\in[n]} C_3^i$

Proof of 3 – SAT ∈ NPC

Given $C_i = \bigvee_{i \in [k]} y_i$ in F with $y_i \in \{x_i, \overline{x}_i\}$ and $a_i = \bigvee_{j=i+1}^{n} y_j$

- $f : \Sigma^* \longrightarrow \Sigma^*; F \mapsto F_3$ with
- $F_3 = \bigwedge_{i \in [n]} C_3^i$
- $C_3^i =$
  $(y_1 \vee y_2 \vee a_1) \wedge \left(\bigwedge_{\ell \in [k-3]} (\overline{a}_\ell \vee y_{\ell+1} \vee a_{\ell+1}) \right) \wedge (\overline{a}_{k-3} \vee y_{k-1} \vee y_k)$

Clique

Definition

$G = (V, E)$ undirected graph, a $k \in \mathbb{N}$, decide whether G has a clique on k nodes, i.e. k of G's nodes form a complete subgraph.

Clique

Definition

$G = (V, E)$ undirected graph, a $k \in \mathbb{N}$, decide whether G has a clique on k nodes, i.e. k of G's nodes form a complete subgraph.

Theorem

Clique $\in$ NPC

Clique

Definition

$G = (V, E)$ undirected graph, a $k \in \mathbb{N}$, decide whether G has a clique on k nodes, i.e. k of G's nodes form a complete subgraph.

Theorem

Clique $\in$ NPC

- plan: $3-\mathrm{SAT} \leq_p$ Clique

Clique

Definition

$G = (V, E)$ undirected graph, a $k \in \mathbb{N}$, decide whether G has a clique on k nodes, i.e. k of G's nodes form a complete subgraph.

Theorem

$\mathit{Clique} \in \mathrm{NPC}$

- plan: $3-\mathrm{SAT} \leq_p \mathrm{Clique}$
- $x_1, \ldots, x_n$ boolean variables, $C = (C_1, \ldots, C_m)$ with $C_i = z_i^{(1)} \vee z_i^{(2)} \vee z_i^{(3)}$ for the literals $z_i^{(j)}$

Clique

Definition

$G = (V, E)$ undirected graph, a $k \in \mathbb{N}$, decide whether G has a clique on k nodes, i.e. k of G's nodes form a complete subgraph.

Theorem

Clique $\in$ NPC

- plan: $3 - \mathrm{SAT} \leq_p \mathrm{Clique}$
- $x_1, \ldots, x_n$ boolean variables, $C = (C_1, \ldots, C_m)$ with $C_i = z_i^{(1)} \vee z_i^{(2)} \vee z_i^{(3)}$ for the literals $z_i^{(j)}$
- $G := (V, E)$ with $V = [m] \times [3]$ and $E = \{\{(i,j),(k,\ell)\} \mid i \neq k \wedge z_i^{(j)} \neq \overline{z}_k^{(\ell)}\}$

Hamilton path

Definition

$G = (V, E)$ undirected graph, decide whether there exists a Hamilton path in G, i.e. a path visiting all nodes extactly once

Hamilton path

Definition

$G = (V, E)$ undirected graph, decide whether there exists a Hamilton path in G, i.e. a path visiting all nodes extactly once

Theorem

$\mathit{HamiltonPath} \in \mathrm{NPC}$

Travelling Salesman Problem

Definition

Consider n cities which are connected via streets. Taking a street s costs toll $c(s)$ for a cost-function c. Is there a path for a travelling salesman visiting all cities with costs at most m?

Travelling Salesman Problem

Definition

Consider n cities which are connected via streets. Taking a street s costs toll $c(s)$ for a cost-function c. Is there a path for a travelling salesman visiting all cities with costs at most m?

Theorem

$TSP \in \mathrm{NPC}$

Graph-Colouring

Definition

$G = (V, E)$ undirected graph, $f : V \longrightarrow [k]$ valid coloring function iff $f(u) \neq f(v)$ for all $\{u, v\} \in E$; decide whether such an f exists for a given k or not.

Graph-Colouring

Definition

$G = (V, E)$ undirected graph, $f : V \longrightarrow [k]$ valid coloring function iff $f(u) \neq f(v)$ for all $\{u, v\} \in E$; decide whether such an f exists for a given k or not.

Theorem

GraphColouring $\in$ NPC

Graph-Colouring

Definition

$G = (V, E)$ undirected graph, $f : V \longrightarrow [k]$ valid coloring function iff $f(u) \neq f(v)$ for all $\{u, v\} \in E$; decide whether such an f exists for a given k or not.

Theorem

$\mathit{GraphColouring} \in \mathrm{NPC}$

- plan: $3 - \mathrm{SAT} \leq_p$ graph-colouring

Graph-Colouring

Definition

$G = (V, E)$ undirected graph, $f : V \longrightarrow [k]$ valid coloring function iff $f(u) \neq f(v)$ for all $\{u, v\} \in E$; decide whether such an f exists for a given k or not.

Theorem

GraphColouring $\in$ NPC

- plan: $3-\mathrm{SAT} \leq_p$ graph-colouring
- given $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$

Graph-Colouring

Definition

$G = (V, E)$ undirected graph, $f : V \longrightarrow [k]$ valid coloring function iff $f(u) \neq f(v)$ for all $\{u, v\} \in E$; decide whether such an f exists for a given k or not.

Theorem

GraphColouring $\in$ NPC

- plan: $3 - \mathrm{SAT} \leq_p$ graph-colouring
- given $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$
- $V = \{v_1, \ldots, v_n, x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n, C_1, \ldots, C_m\}$ and $E = \{(v_i, v_j) | \, i \neq j\} \cup \{(v_i, x_j), (v_i, \overline{x}_i) | \, i \neq j\} \cup \{(x_i, \overline{x}_i) | \, i \in [n]\} \cup \{(x_i, C_j) | \, x_i \notin C_j\} \cup \{(\overline{x}_i, C_j) | \, \overline{x}_i \notin C_j\}$

Easy Instances

Lemma

2-Colouring is in P*, i.e. find a* 2*-colouring in polynomial time*

Lemma

$2-\mathrm{SAT}$ *is in* P*, i.e. find a satisfying substitution in polynomial time*

Knapsack-Problem, Rucksack-Problem

Definition

Consider k goods with values $a_1, \ldots, a_k$ and weight $g_1, \ldots, g_k$, decide whether there exists a $M \subseteq [k]$ with $\sum_{i \in M} g_i \leq G$ and $\sum_{i \in M} a_i \geq A$ for given $A, G \in \mathbb{N}$

Knapsack-Problem, Rucksack-Problem

Definition

Consider k goods with values $a_1, \ldots, a_k$ and weight $g_1, \ldots, g_k$, decide whether there exists a $M \subseteq [k]$ with $\sum_{i \in M} g_i \leq G$ and $\sum_{i \in M} a_i \geq A$ for given $A, G \in \mathbb{N}$

Theorem

$RSP \in \mathrm{NPC}$

Proof of RSP $\in$ NPC

- plan: $3-\mathrm{SAT} \leq_p \mathrm{RSP}$

Proof of RSP $\in$ NPC

- plan: $3 - \text{SAT} \leq_p \text{RSP}$
- given: $X = \{x_1, \ldots, x_n\}, C = \{C_1, \ldots, C_m\}$

Proof of RSP $\in$ NPC

- plan: $3 - \mathrm{SAT} \leq_p \mathrm{RSP}$
- given: $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$
- w.l.o.g. $A = G$, $a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)

Proof of RSP $\in$ NPC

- plan: $3-\mathrm{SAT} \leq_p \mathrm{RSP}$
- given: $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$
- w.l.o.g. $A = G$, $a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)
- $A := \underbrace{4\ldots4}_{m \text{ times}}\underbrace{1\ldots1}_{n \text{ times}}$

Proof of RSP $\in$ NPC

- plan: $3-\text{SAT} \leq_p \text{RSP}$
- given: $X = \{x_1, \ldots, x_n\}, C = \{C_1, \ldots, C_m\}$
- w.l.o.g. $A = G$, $a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)
- $A := \underbrace{4 \ldots 4}_{m \text{ times}} \underbrace{1 \ldots 1}_{n \text{ times}}$
- $a_i = \alpha_1 \ldots \alpha_m 0_1 \ldots 0_{i-1} 1_i 0_{i+1} \ldots 0_n$ with
  $$\alpha_j = \begin{cases} 1 & \text{if } x_i \in C_j, \\ 0 & \text{if } x_i \notin C_j \end{cases}$$

Proof of RSP $\in$ NPC

- plan: $3 - \text{SAT} \leq_p \text{RSP}$
- given: $X = \{x_1, \ldots, x_n\}, C = \{C_1, \ldots, C_m\}$
- w.l.o.g. $A = G, a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)
- $A := \underbrace{4\ldots4}_{m \text{ times}}\underbrace{1\ldots1}_{n \text{ times}}$
- $a_i = \alpha_1 \ldots \alpha_m 0_1 \ldots 0_{i-1} 1_i 0_{i+1} \ldots 0_n$ with
$$\alpha_j = \begin{cases} 1 & \text{if } x_i \in C_j, \\ 0 & \text{if } x_i \notin C_j \end{cases}$$
- $b_i = \alpha_1 \ldots \alpha_m 0_1 \ldots 0_{i-1} 1_i 0_{i+1} \ldots 0_n$ with
$$\alpha_j = \begin{cases} 1 & \text{if } \overline{x}_i \in C_j, \\ 0 & \text{if } \overline{x}_i \notin C_j \end{cases}$$

Proof of RSP $\in$ NPC

- plan: $3-\text{SAT} \leq_p \text{RSP}$
- given: $X = \{x_1, \dots, x_n\}, C = \{C_1, \dots, C_m\}$
- w.l.o.g. $A = G, a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)
- $A := \underbrace{4 \dots 4}_{m \text{ times}} \underbrace{1 \dots 1}_{n \text{ times}}$
- $a_i = \alpha_1 \dots \alpha_m 0_1 \dots 0_{i-1} 1_i 0_{i+1} \dots 0_n$ with
  $$\alpha_j = \begin{cases} 1 & \text{if } x_i \in C_j, \\ 0 & \text{if } x_i \notin C_j \end{cases}$$
- $b_i = \alpha_1 \dots \alpha_m 0_1 \dots 0_{i-1} 1_i 0_{i+1} \dots 0_n$ with
  $$\alpha_j = \begin{cases} 1 & \text{if } \overline{x}_i \in C_j, \\ 0 & \text{if } \overline{x}_i \notin C_j \end{cases}$$
- $c_j = 0_1 \dots 0_{j-1} 1_j 0_{j+1} \dots 0_m 0_1 \dots 0_n$

Proof of RSP $\in$ NPC

- plan: $3-\mathrm{SAT} \leq_p \mathrm{RSP}$
- given: $X = \{x_1, \ldots, x_n\}$, $C = \{C_1, \ldots, C_m\}$
- w.l.o.g. $A = G$, $a_i = g_i$ (this problem NPC $\Rightarrow$ general, too)
- $A := \underbrace{4 \ldots 4}_{m \text{ times}} \underbrace{1 \ldots 1}_{n \text{ times}}$
- $a_i = \alpha_1 \ldots \alpha_m 0_1 \ldots 0_{i-1} 1_i 0_{i+1} \ldots 0_n$ with
  $$\alpha_j = \begin{cases} 1 & \text{if } x_i \in C_j, \\ 0 & \text{if } x_i \notin C_j \end{cases}$$
- $b_i = \alpha_1 \ldots \alpha_m 0_1 \ldots 0_{i-1} 1_i 0_{i+1} \ldots 0_n$ with
  $$\alpha_j = \begin{cases} 1 & \text{if } \overline{x}_i \in C_j, \\ 0 & \text{if } \overline{x}_i \notin C_j \end{cases}$$
- $c_j = 0_1 \ldots 0_{j-1} 1_j 0_{j+1} \ldots 0_m 0_1 \ldots 0_n$
- $d_j = 2c_j$

Partition-Problem

Definition

Given $I = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Partition-Problem

Definition

Given $I = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Theorem

Partition $\in$ NPC

Partition-Problem

Definition

Given $I = \{a_1, \dots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Theorem

Partition $\in$ NPC

Proof:

- plan: RSP $\leq_p$ Partition

Partition-Problem

Definition

Given $I = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Theorem

Partition $\in$ NPC

Proof:

- plan: RSP $\leq_p$ Partition
- $I = (a_1, \ldots, a_n, a_1, \ldots, a_n, A, A)$ RSP-instance

Partition-Problem

Definition

Given $I = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Theorem

Partition $\in$ NPC

Proof:

- plan: RSP $\leq_p$ Partition
- $I = (a_1, \ldots, a_n, a_1, \ldots, a_n, A, A)$ RSP-instance
- $S := \sum_{i \in [n]} a_i$, $a_{n+1} := S - A + 1$, $a_{n+2} := A + 1$

Partition-Problem

Definition

Given $I = \{a_1, \ldots, a_n\} \subseteq \mathbb{N}$ decide whether there exists $T \subseteq [n]$ with $\sum_{i \in T} a_i = \sum_{i \in [n] \setminus T} a_i$

Theorem

Partition $\in$ NPC

Proof:

- plan: RSP $\leq_p$ Partition
- $I = (a_1, \ldots, a_n, a_1, \ldots, a_n, A, A)$ RSP-instance
- $S := \sum_{i \in [n]} a_i$, $a_{n+1} := S - A + 1$, $a_{n+2} := A + 1$
- Partition-instance $(a_1, \ldots, a_{n+2})$

Binpacking-Problem

Definition

Given $a_1, \ldots, a_n, b, k \in \mathbb{N}$ decide whether there exists $\dot\bigcup_{i \in][k]} I_i = [n]$ with $\sum_{i \in I_j} a_i \leq b$

Binpacking-Problem

Definition

Given $a_1, \dots, a_n, b, k \in \mathbb{N}$ decide whether there exists $\dot{\bigcup}_{i \in][k]} I_i = [n]$ with $\sum_{i \in I_j} a_i \leq b$

Theorem

Binpacking $\in$ NPC

P-Problems on Graphs

Lemma

Determining the minimal-spanning-tree or the shortest-path-tree in a given undirected graph is in P

P-Problems on Graphs

Lemma

Determining the minimal-spanning-tree or the shortest-path-tree in a given undirected graph is in P

Proof:

- the Kruskal- resp. the Dijkstra-algorithm solve the problem

NP-Problems on Graphs

Definition

Given an undirected graph $G = (V, E)$, find a minimal

- vertex cover, i.e. find $C \subseteq V$ such that $\forall \{u, v\} \in E : \{u, v\} \cap C \neq \emptyset$
- set cover, i.e. find $C \subseteq E$ such that $\forall v \in V \exists e \in C : v \in e$

NP-Problems on Graphs

Definition

Given an undirected graph $G = (V, E)$, find a minimal

- vertex cover, i.e. find $C \subseteq V$ such that $\forall \{u, v\} \in E : \{u, v\} \cap C \neq \emptyset$
- set cover, i.e. find $C \subseteq E$ such that $\forall v \in V \exists e \in C : v \in e$

VertexCover, SetCover, Clique, IndependentSet (complement of Clique) are in NPC