CS 577- Intro to Algorithms

Reductions (Part 2)

Dieter van Melkebeek

November 12, 2020

Outline

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Notation

$A \leq^p B$: A reduces to B in polynomial time.

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Notation

$A \leq^p B$: A reduces to B in polynomial time.

Example polynomial-time reductions

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Notation

$A \leq^p B$: A reduces to B in polynomial time.

Example polynomial-time reductions

- Bipartite Matching $\leq^p$ Integral Max Flow

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Notation

$A \leq^p B$: A reduces to B in polynomial time.

Example polynomial-time reductions

- Bipartite Matching $\leq^p$ Integral Max Flow
- Between different versions of Independent Set

Outline

Definition

A reduction from computational problem A to computational problem B is an algorithm for A that uses blackbox for B.

Running time

Time to run reduction discounting time for blackbox.

Notation

$A \leq^p B$: A reduces to B in polynomial time.

Example polynomial-time reductions

- Bipartite Matching $\leq^p$ Integral Max Flow
- Between different versions of Independent Set
- Satisfiability $\leq^p$ Independent Set

Independent Set Problems

Independent Set Problems

Definition

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that $E \cap S \times S = \emptyset$.

Independent Set Problems

Definition

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that $E \cap S \times S = \emptyset$.

Optimization problem

Input: graph G

Output: independent set S of G such that $|S|$ is maximized

Independent Set Problems

Definition

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that $E \cap S \times S = \emptyset$.

Optimization problem

Input: graph G

Output: independent set S of G such that $|S|$ is maximized

Search problem

Input: graph G, $k \in \mathbb{N}$

Output: independent set S of G such that $|S| \geq k$, or report that no such set exists

Independent Set Problems

Definition

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that $E \cap S \times S = \emptyset$.

Optimization problem

Input: graph G

Output: independent set S of G such that $|S|$ is maximized

Search problem

Input: graph G, $k \in \mathbb{N}$

Output: independent set S of G such that $|S| \geq k$, or report that no such set exists

Decision problem

Input: graph G, $k \in \mathbb{N}$

Output: whether independent set S with $|S| \geq k$ exists in G

Independent Set – Decision $\leq^p$ Search $\leq^p$ Optimization

Independent Set – Decision $\leq^p$ Search $\leq^p$ Optimization

- Decision $\leq^p$ Search

Independent Set – Decision $\leq^p$ Search $\leq^p$ Optimization

- Decision $\leq^p$ Search

1: **if** Search$(G, k) =$ "no solution" **then**
2:     **return** "no"
3: **else**
4:     **return** "yes"

Independent Set – Decision $\leq^p$ Search $\leq^p$ Optimization

- Decision $\leq^p$ Search
  1. **if** Search$(G, k) =$ "no solution" **then**
  2. **return** "no"
  3. **else**
  4. **return** "yes"
- Search $\leq^p$ Optimization

Independent Set – Decision $\leq^p$ Search $\leq^p$ Optimization

- Decision $\leq^p$ Search

```
if Search(G, k) = "no solution" then
    return "no"
else
    return "yes"
```

- Search $\leq^p$ Optimization

```
I ← Optimization(G)
if |I| ≥ k then
    return I
else
    return "no solution"
```

Independent Set – Optimization $\leq^p$ Search

Independent Set – Optimization $\leq^p$ Search

- Linear search for maximum size

Independent Set – Optimization $\leq^p$ Search

- Linear search for maximum size

```
k ← 0
while Search(G, k) ≠ "no solution" do
    k ← k + 1
return Search(G, k)
```

Independent Set – Optimization $\leq^p$ Search

- Linear search for maximum size

  $k \leftarrow 0$
  while Search$(G, k) \neq$ "no solution" **do**
  $\quad k \leftarrow k + 1$
  return Search(G, k)

- Binary search reduces number of queries from $O(|V|)$ to $O(\log |V|)$.

Independent Set – Search $\leq^p$ Decision

Independent Set – Search $\leq^p$ Decision

- Vertex v has to be in *every* independent set of size at least k $\Leftrightarrow$ Decision$(G - \{v\}, k) =$ "no"

Independent Set – Search $\leq^p$ Decision

- Vertex v has to be in *every* independent set of size at least k $\Leftrightarrow$ Decision$(G - \{v\}, k) =$ "no"
- Reluctant approach

Independent Set – Search $\leq^p$ Decision

- Vertex v has to be in *every* independent set of size at least k $\Leftrightarrow$ Decision$(G - \{v\}, k) =$ "no"
- Reluctant approach

```
if Decision(G, k) = "no" then
    return "no solution"
I ← V
for each v ∈ V do
    if Decision(G|_{I\{v}}, k) = "yes" then
        I ← I \ {v}
return I
```

Independent Set – Search $\leq^p$ Decision

- Vertex v has to be in *every* independent set of size at least k $\Leftrightarrow$ Decision$(G - \{v\}, k) =$ "no"
- Reluctant approach

  if Decision$(G, k) =$ "no" **then**
      return "no solution"
  $I \leftarrow V$
  for each $v \in V$ **do**
      if Decision$(G|_{I\setminus\{v\}}, k) =$ "yes" **then**
          $I \leftarrow I \setminus \{v\}$
  return I

- Considering vertices in lexicographical order results in independent set of size at least k with the lexicographically first characteristic vector.

Independent Set – Search $\leq^p$ Decision

Independent Set – Search $\leq^p$ Decision

- Vertex v can be in *some* independent set of size at least k $\Leftrightarrow$ Decision$(G - (\{v\} \cup G(v)), k - 1) =$ "yes"

Independent Set – Search $\leq^p$ Decision

- Vertex v can be in *some* independent set of size at least k $\Leftrightarrow$ Decision$(G - (\{v\} \cup G(v)), k - 1) =$ "yes"
- Eager approach

Independent Set – Search $\leq^p$ Decision

- Vertex v can be in *some* independent set of size at least k $\Leftrightarrow$ Decision$(G - (\{v\} \cup G(v)), k-1) =$ "yes"
- Eager approach

```
if Decision(G, k) = "no" then
    return "no solution"
I ← ∅; S ← V
while S ≠ ∅ do
    pick v ∈ S; S ← S \ {v}
    if Decision(G|_{S\G(v)}, k − 1) = "yes" then
        I ← I ∪ {v}
        S ← S \ G(v)
        k ← k − 1
return I
```

Independent Set – Search $\leq^p$ Decision

- Vertex v can be in *some* independent set of size at least k $\Leftrightarrow$ Decision$(G - (\{v\} \cup G(v)), k - 1) =$ "yes"
- Eager approach

```
if Decision(G, k) = "no" then
    return "no solution"
I ← ∅; S ← V
while S ≠ ∅ do
    pick v ∈ S; S ← S \ {v}
    if Decision(G|_{S\G(v)}, k − 1) = "yes" then
        I ← I ∪ {v}
        S ← S \ G(v)
        k ← k − 1
return I
```

- Considering vertices in lexicographical order results in independent set of size at least k with the lexicographically last characteristic vector.

Boolean Formulas

Boolean Formulas

Definition

Boolean Formulas

Definition

- Base case: variables x_1, x_2, ...

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

- Literal: variable x, negated variable $\overline{x}$

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

- Literal: variable x, negated variable $\overline{x}$
- Clause: disjunction of literals

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

- Literal: variable x, negated variable $\overline{x}$
- Clause: disjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

- Literal: variable x, negated variable $\overline{x}$
- Clause: disjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

Boolean Formulas

Definition

- Base case: variables $x_1, x_2, \ldots$
- Constructors:
  - Conjunction (AND): $\wedge$
  - Disjunction (inclusive OR): $\vee$
  - Negation: $\neg$

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

Restricted types

- Literal: variable x, negated variable $\overline{x}$
- Clause: disjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$$\wedge_{j=1}^{m} C_j \text{ where } C_j = \vee_{r=1}^{k_j} \ell_{jr} \text{ and } \ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$$

Satisfiability

Satisfiability

Search version

Input: Boolean formula φ

Satisfiability

Search version

Input: Boolean formula φ

Output: satisfying assignment of φ, i.e., a setting of the variables to true/false that makes φ evaluate to true; or report that no such setting exists

Satisfiability

Search version

Input: Boolean formula φ

Output: satisfying assignment of φ, i.e., a setting of the variables to true/false that makes φ evaluate to true; or report that no such setting exists

Decision version

Input: Boolean formula φ

Output: whether φ has a satisfying assignment

Satisfiability

Search version

Input: Boolean formula φ

Output: satisfying assignment of φ, i.e., a setting of the variables to true/false that makes φ evaluate to true; or report that no such setting exists

Decision version

Input: Boolean formula φ

Output: whether φ has a satisfying assignment

Restricted problems

Satisfiability

Search version

Input: Boolean formula φ

Output: satisfying assignment of φ, i.e., a setting of the variables to true/false that makes φ evaluate to true; or report that no such setting exists

Decision version

Input: Boolean formula φ

Output: whether φ has a satisfying assignment

Restricted problems

- CNF-SAT: φ is CNF

Satisfiability

Search version

Input: Boolean formula φ

Output: satisfying assignment of φ, i.e., a setting of the variables to true/false that makes φ evaluate to true; or report that no such setting exists

Decision version

Input: Boolean formula φ

Output: whether φ has a satisfying assignment

Restricted problems

- CNF-SAT: φ is CNF
- k-SAT for fixed $k \in \mathbb{N}$: φ is k-CNF, i.e., CNF with each clause containing at most k literals

CNF-SAT $\leq^p$ Independent Set

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

- Output: satisfying assignment for φ, or report none exists

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)
- Mapping reduction

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$$\wedge_{j=1}^{m} C_j \text{ where } C_j = \vee_{r=1}^{k_j} \ell_{jr} \text{ and } \ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)
- Mapping reduction, i.e., translation of CNF-SAT instance φ into Independent Set instance (G, k) with same decision:

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)
- Mapping reduction, i.e., translation of CNF-SAT instance φ into Independent Set instance (G, k) with same decision: φ is satisfiable $\Leftrightarrow$ G has independent set of size at least k.

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$$\wedge_{j=1}^{m} C_j \text{ where } C_j = \vee_{r=1}^{k_j} \ell_{jr} \text{ and } \ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)
- Mapping reduction, i.e., translation of CNF-SAT instance φ into Independent Set instance (G, k) with same decision: φ is satisfiable $\Leftrightarrow$ G has independent set of size at least k.
- Reduction runs in polynomial time.

CNF-SAT $\leq^p$ Independent Set

- Input: CNF-formula φ

$$(x_1 \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee x_2)$$

$\wedge_{j=1}^{m} C_j$ where $C_j = \vee_{r=1}^{k_j} \ell_{jr}$ and $\ell_{jr} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots x_n, \overline{x_n}\}$

- Output: satisfying assignment for φ, or report none exists
- Reduction makes one query: (G, k)
- Mapping reduction, i.e., translation of CNF-SAT instance φ into Independent Set instance (G, k) with same decision: φ is satisfiable $\Leftrightarrow$ G has independent set of size at least k.
- Reduction runs in polynomial time.
- Gadget reduction

CNF-SAT $\leq^p$ Independent Set – variable gadgets

CNF-SAT $\leq^p$ Independent Set – variable gadgets

Construction

For each variable x_i, include two new vertices, one labeled x_i and the other $\overline{x_i}$, and include the edge $(x_i, \overline{x_i})$.

CNF-SAT $\leq^p$ Independent Set – variable gadgets

Construction

For each variable x_i, include two new vertices, one labeled x_i and the other $\overline{x_i}$, and include the edge $(x_i, \overline{x_i})$.

Properties

CNF-SAT $\leq^p$ Independent Set – variable gadgets

Construction

For each variable x_i, include two new vertices, one labeled x_i and the other $\overline{x_i}$, and include the edge $(x_i, \overline{x_i})$.

Properties

- Maximum size of independent set is n.

CNF-SAT $\leq^p$ Independent Set – variable gadgets

Construction

For each variable x_i, include two new vertices, one labeled x_i and the other $\overline{x_i}$, and include the edge $(x_i, \overline{x_i})$.

Properties

- Maximum size of independent set is n.
- Bijection between
  - independent sets of maximum size and
  - assignments to variables $x_1, x_2, \ldots, x_n$.

CNF-SAT $\leq^p$ Independent Set – clause gadgets

CNF-SAT $\leq^p$ Independent Set – clause gadgets

Construction

For each clause C_j for $j \in [m]$, include a clique (complete graph) on k_j new vertices, where k_j = number of literals of C_j. Label each vertex of the clique with a unique literal of C_j.

CNF-SAT $\leq^p$ Independent Set – clause gadgets

Construction

For each clause C_j for $j \in [m]$, include a clique (complete graph) on k_j new vertices, where $k_j =$ number of literals of C_j. Label each vertex of the clique with a unique literal of C_j.

Properties

- Maximum size of independent set is m.

CNF-SAT $\leq^p$ Independent Set – clause gadgets

Construction

For each clause C_j for $j \in [m]$, include a clique (complete graph) on k_j new vertices, where k_j = number of literals of C_j. Label each vertex of the clique with a unique literal of C_j.

Properties

- Maximum size of independent set is m.
- Bijection between
  - independent sets of maximum size and
  - choices of literal in each clause C_j for $j \in [m]$.

CNF-SAT $\leq^p$ Independent Set – connections

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

Properties

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

Properties

- Max independent set size in G is at most $n + m$.

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

Properties

- Max independent set size in G is at most $n + m$.
- Independent set of size n in variable part can be extended with vertex in gadget of clause $C_j \Leftrightarrow$ assignment satisfies C_j.

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

Properties

- Max independent set size in G is at most $n + m$.
- Independent set of size n in variable part can be extended with vertex in gadget of clause C_j $\Leftrightarrow$ assignment satisfies C_j.
- Max independent set size in G is at least $k \doteq n + m$ $\Leftrightarrow$ φ has a satisfying assignment

CNF-SAT $\leq^p$ Independent Set – connections

Construction of G

- Disjoint union of all variable gadgets and clause gadgets.
- For each vertex of a variable gadget with label ℓ, and each vertex of a clause gadget with label $\overline{\ell}$, include the edge between them.

Properties

- Max independent set size in G is at most $n + m$.
- Independent set of size n in variable part can be extended with vertex in gadget of clause C_j $\Leftrightarrow$ assignment satisfies C_j.
- Max independent set size in G is at least $k \doteq n + m$ $\Leftrightarrow$ φ has a satisfying assignment
- Bijection between
  - independent sets of size $n + m$ in G and
  - satisfying assignments to $x_1, x_2, \ldots, x_n$ combined with choices of satisfying literal in each clause C_j for $j \in [m]$.