Automated Reasoning I*

Uwe Waldmann

Winter Term 2023/2024

Topics of the Course

Preliminaries

abstract reduction systems
well-founded orderings

Propositional logic

syntax, semantics
calculi: CDCL-procedure, OBDDs
implementation: Two watched literals

First-order predicate logic

syntax, semantics, model theory, ...
calculi: resolution, tableaux
implementation: sharing, indexing

First-order predicate logic with equality

term rewriting systems
calculi: Knuth-Bendix completion, dependency pairs

Emphasis on:

logics and their properties,

proof systems for these logics and their properties:
soundness, completeness, implementation

---

*This document contains the text of the lecture slides (almost verbatim) plus some additional information, in particular proofs of theorems that are presented on the blackboard during the course. It is not a full script and does not contain the examples and additional explanations given during the lecture. Moreover it should not be taken as an example how to write a research paper – neither stylistically nor typographically.

Parts of this document are based on lecture notes by Harald Ganzinger and Christoph Weidenbach.

1 Preliminaries

Literature:
Franz Baader and Tobias Nipkow: *Term rewriting and all that*,
Cambridge Univ. Press, 1998, Chapter 2.

Before we start with the main subjects of the lecture, we repeat some prerequisites from mathematics and computer science and introduce some tools that we will need throughout the lecture.

1.1 Mathematical Prerequisites

$\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of natural numbers (including 0).

$\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ denote the integers, rational numbers and the real numbers, respectively.

$\emptyset$ is the empty set.

If M and M' are sets, then $M \cap M'$, $M \cup M'$, and $M \setminus M'$ denote the intersection, union, and set difference of M and M'.

The subset relation is denoted by $\subseteq$. The strict subset relation is denoted by $\subset$ (i. e., $M \subset M'$ if and only if $M \subseteq M'$ and $M \neq M'$).

Relations

Let M be a set, let $n \geq 2$. We write M^n for the n-fold cartesian product $M \times \cdots \times M$.

In order to handle the cases $n \geq 2$, $n = 1$, and $n = 0$ simultaneously, we also define $M^1 = M$ and $M^0 = \{()\}$. (We do not distinguish between an element m of M and a 1-tuple (m) of an element of M.)

An n-ary *relation* R over some set M is a subset of M^n: $R \subseteq M^n$.

We often use predicate notation for relations:

Instead of $(m_1, \ldots, m_n) \in R$ we write $R(m_1, \ldots, m_n)$, and say that $R(m_1, \ldots, m_n)$ holds or is true.

For binary relations, we often use infix notation, so
$(m, m') \in {<} \quad \Leftrightarrow \quad {<}(m, m') \quad \Leftrightarrow \quad m < m'$.

Since relations are sets, we can use the usual set operations for then.

Example: Let $R = \{(0, 2), (1, 2), (2, 2), (3, 2)\} \subseteq \mathbb{N} \times \mathbb{N}$.
Then $R \cap {<} \;=\; R \cap \{\, (n, m) \in \mathbb{N} \times \mathbb{N} \mid n < m \,\} \;=\; \{(0, 2), (1, 2)\}$.

A relation Q is a *subrelation* of a relation R if $Q \subseteq R$.

Words

Given a non-empty set (also called *alphabet*) Σ, the set Σ^* of *finite words* over Σ is defined inductively by

(i) the empty word ε is in Σ^*,

(ii) if $u \in \Sigma^*$ and $a \in \Sigma$ then ua is in Σ^*.

The set of *non-empty finite words* Σ^+ is $\Sigma^* \setminus \{\varepsilon\}$.

The *concatenation* of two words $u, v \in \Sigma^*$ is denoted by uv.

The length $|u|$ of a word $u \in \Sigma^*$ is defined by

(i) $|\varepsilon| := 0$,

(ii) $|ua| := |u| + 1$ for any $u \in \Sigma^*$ and $a \in \Sigma$.

1.2 Abstract Reduction Systems

Throughout the lecture, we will have to work with reduction systems,

on the object level, in particular in the section on equality,

and on the meta level, i. e., to describe deduction calculi.

An *abstract reduction system* is a pair $(A, \rightarrow)$, where

A is a non-empty set,

$\rightarrow \subseteq A \times A$ is a binary relation on A.

The relation $\rightarrow$ is usually written in infix notation, i. e., $a \rightarrow b$ instead of $(a, b) \in \rightarrow$.

Let $\rightarrow' \subseteq A \times A$ and $\rightarrow'' \subseteq A \times A$ be two binary relations. Then the *composition of* $\rightarrow'$ *and* $\rightarrow''$ is the binary relation $(\rightarrow' \circ \rightarrow'') \subseteq A \times A$ defined by

$a \; (\rightarrow' \circ \rightarrow'') \; c$ if and only if there exists some $b \in A$ such that $a \rightarrow' b$ and $b \rightarrow'' c$.

For a binary relation $\rightarrow \subseteq A \times A$, we define:

$\rightarrow^0$	$= \{ (a,a) \mid a \in A \}$	*identity*
$\rightarrow^{i+1}$	$= \rightarrow^i \circ \rightarrow$	$i+1$-*fold composition*
$\rightarrow^+$	$= \bigcup_{i>0} \rightarrow^i$	*transitive closure*
$\rightarrow^*$	$= \bigcup_{i\geq 0} \rightarrow^i \;=\; \rightarrow^+ \cup \rightarrow^0$	*reflexive transitive closure*
$\rightarrow^=$	$= \rightarrow \cup \rightarrow^0$	*reflexive closure*
$\leftarrow$	$= \rightarrow^{-1} \;=\; \{ (b,c) \mid c \rightarrow b \}$	*inverse*
$\leftrightarrow$	$= \rightarrow \cup \leftarrow$	*symmetric closure*
$\leftrightarrow^+$	$= (\leftrightarrow)^+$	*transitive symmetric closure*
$\leftrightarrow^*$	$= (\leftrightarrow)^*$	*reflexive transitive symmetric closure* or *equivalence closure*

$b \in A$ is *reducible*, if there is a c such that $b \rightarrow c$.

b is *in normal form* (or *irreducible*), if it is not reducible.

c is a *normal form of* b, if $b \rightarrow^* c$ and c is in normal form.
Notation: $c = b{\downarrow}$ (if the normal form of b is unique).

A relation $\rightarrow$ is called

terminating, if there is no infinite descending chain $b_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \ldots$.

normalizing, if every $b \in A$ has a normal form.

Lemma 1.1 *If $\rightarrow$ is terminating, then it is normalizing.*

Note: The reverse implication does not hold.

1.3 Orderings

Important properties of binary relations:

Let $M \neq \emptyset$. A binary relation $R \subseteq M \times M$ is called

reflexive, if $R(x,x)$ for all $x \in M$,

irreflexive, if $\neg R(x,x)$ for all $x \in M$,

antisymmetric, if $R(x,y)$ and $R(y,x)$ imply $x = y$ for all $x, y \in M$,

transitive, if $R(x,y)$ and $R(y,z)$ imply $R(x,z)$ for all $x, y, z \in M$,

total, if $R(x,y)$ or $R(y,x)$ or $x = y$ for all $x, y \in M$.

A *strict partial ordering* $\succ$ on a set $M \neq \emptyset$ is a transitive and irreflexive binary relation on M.

Notation:
$\prec$ for the inverse relation $\succ^{-1}$
$\succeq$ for the reflexive closure $(\succ \cup =)$ of $\succ$

Let $\succ$ be a strict partial ordering on M; let $M' \subseteq M$.

$a \in M'$ is called *minimal in* M', if there is no $b \in M'$ with $a \succ b$.

$a \in M'$ is called *smallest in* M', if $b \succ a$ for all $b \in M' \setminus \{a\}$.

Analogously:

$a \in M'$ is called *maximal in* M', if there is no $b \in M'$ with $a \prec b$.

$a \in M'$ is called *largest in* M', if $b \prec a$ for all $b \in M' \setminus \{a\}$.

Notation:
$M^{\prec x} = \{\, y \in M \mid y \prec x \,\}$,
$M^{\preceq x} = \{\, y \in M \mid y \preceq x \,\}$.

A subset $M' \subseteq M$ is called *downward closed*, if $x \in M'$ and $x \succ y$ implies $y \in M'$.

Well-Foundedness

Termination of reduction systems is strongly related to the concept of well-founded orderings.

A strict partial ordering $\succ$ on M is called *well-founded (or Noetherian),* if there is no infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \dots$ with $a_i \in M$.

Well-Foundedness and Termination

Lemma 1.2 *If $\succ$ is a well-founded partial ordering and $\rightarrow \subseteq \succ$, then $\rightarrow$ is terminating.*

Proof. Suppose that $\rightarrow \subseteq \succ$ for some partial ordering $\succ$ and that $\rightarrow$ is not terminating. Then there exists an infinite descending chain $b_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \dots$. Since $\rightarrow \subseteq \succ$, we have an infinite descending chain $b_0 \succ b_1 \succ b_2 \succ \dots$, hence $\succ$ is not well-founded. □

Lemma 1.3 *If $\rightarrow$ is a terminating binary relation over A, then $\rightarrow^+$ is a well-founded partial ordering.*

Proof. Transitivity of $\rightarrow^+$ is obvious; irreflexivity and well-foundedness follow from termination of $\rightarrow$. □

Well-Founded Orderings: Examples

Natural numbers: $(\mathbb{N}, >)$

Lexicographic orderings: Let $(M_1, \succ_1), (M_2, \succ_2)$ be well-founded orderings. Define their *lexicographic combination*

$$\succ \; = (\succ_1, \succ_2)_{\text{lex}}$$

on $M_1 \times M_2$ by

$$(a_1, a_2) \succ (b_1, b_2) \quad :\Leftrightarrow \quad a_1 \succ_1 b_1 \text{ or } (a_1 = b_1 \text{ and } a_2 \succ_2 b_2)$$

(analogously for more than two orderings). This again yields a well-founded ordering (proof below).

Length-based ordering on words: For alphabets Σ with a well-founded ordering $>_\Sigma$, the relation $\succ$ defined as

$$w \succ w' \quad :\Leftrightarrow \quad |w| > |w'| \text{ or } (|w| = |w'| \text{ and } w >_{\Sigma,\text{lex}} w')$$

is a well-founded ordering on the set Σ^* of finite words over the alphabet Σ (Exercise).

Counterexamples:
$(\mathbb{Z}, >)$
$(\mathbb{N}, <)$
the lexicographic ordering on Σ^*

Basic Properties of Well-Founded Orderings

Lemma 1.4 *$(M, \succ)$ is well-founded if and only if every non-empty $M' \subseteq M$ has a minimal element.*

Proof. "$\Leftarrow$": Suppose that $(M, \succ)$ is not well-founded. Then there is an infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \ldots$ with $a_i \in M$. Consequently, the subset $M' = \{\, a_i \mid i \in \mathbb{N} \,\}$, does not have a minimal element.

"$\Rightarrow$": Suppose that the non-empty subset $M' \subseteq M$ does not have a minimal element. Choose $a_0 \in M'$ arbitrarily. Since for every $a_i \in M'$ there is a smaller $a_{i+1} \in M'$ (otherwise a_i would be minimal in M'), there is an infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \ldots$ $\square$

Lemma 1.5 *$(M_1, \succ_1)$ and $(M_2, \succ_2)$ are well-founded if and only if $(M_1 \times M_2, \succ)$ with $\succ = (\succ_1, \succ_2)_{\text{lex}}$ is well-founded.*

Proof. "⇒": Suppose $(M_1 \times M_2,\ \succ)$ is not well-founded. Then there is an infinite sequence $(a_0, b_0) \succ (a_1, b_1) \succ (a_2, b_2) \succ \dots$.

Let $A = \{\, a_i \mid i \geq 0 \,\} \subseteq M_1$. Since $(M_1, \succ_1)$ is well-founded, A has a minimal element a_n. But then $B = \{\, b_i \mid i \geq n \,\} \subseteq M_2$ can not have a minimal element, contradicting the well-foundedness of $(M_2, \succ_2)$.

"⇐": obvious. □

Monotone Mappings

Let $(M, \succ)$ and $(M', \succ')$ be strict partial orderings. A mapping $\varphi : M \to M'$ is called *monotone*, if $a \succ b$ implies $\varphi(a) \succ' \varphi(b)$ for all $a, b \in M$.

Lemma 1.6 *If φ is a monotone mapping from $(M, \succ)$ to $(M', \succ')$ and $(M', \succ')$ is well-founded, then $(M, \succ)$ is well-founded.*

Proof. Suppose that $(M, \succ)$ is not well-founded, then there exists an infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \dots$. Since $a_i \succ a_{i+1}$ implies $\varphi(a_i) \succ' \varphi(a_{i+1})$, we obtain an infinite descending chain $\varphi(a_0) \succ' \varphi(a_1) \succ' \varphi(a_2) \succ' \dots$, contradicting the well-foundedness of $(M', \succ')$. □

Well-founded Induction

Well-founded induction generalizes the usual induction over natural numbers or data structures.

Theorem 1.7 (Well-founded (or Noetherian) Induction) *Let $(M, \succ)$ be a well-founded ordering, let Q be a property of elements of M.*

If for all $m \in M$ the implication

> *if $Q(m')$ for all $m' \in M$ such that $m \succ m'$,*[1]
> *then $Q(m)$.*[2]

is satisfied, then the property $Q(m)$ holds for all $m \in M$.

Proof. Let $X = \{\, m \in M \mid Q(m) \text{ false} \,\}$. Suppose that $X \neq \emptyset$. Since $(M, \succ)$ is well-founded, X has a minimal element m_0. Hence for all $m' \in M$ with $m' \prec m_0$ the property $Q(m')$ holds. On the other hand, the implication which is presupposed for this theorem holds in particular also for m_0, hence $Q(m_0)$ must be true. Therefore m_0 cannot be in X, contradicting the assumption. □

[1]induction hypothesis
[2]induction step

Well-founded Recursion

Similarly, well-founded recursion generalizes the usual recursion over natural numbers or data structures. We will need this concept only once in this lecture (and once more in Automated Reasoning II), but in one of the main theorems.

Let M and S be sets, let $N \subseteq M$, and let $f : M \to S$ be a function. Then the *restriction* of f to N, denoted by $f|_N$, is a function from N to S with $f|_N(x) = f(x)$ for all $x \in N$.

Theorem 1.8 (Well-founded (or Noetherian) Recursion) *Let $(M, \succ)$ be a well-founded ordering, let S be a set. Let ϕ be a binary function that takes two arguments x and g and maps them to an element of S, where $x \in M$ and g is a function from $M^{\prec x}$ to S.*

Then there exists exactly one function $f : M \to S$ such that for all $x \in M$

$$f(x) = \phi(x, f|_{M^{\prec x}})$$

Proof. The proof consists of four parts.

Part 1: For every downward closed subset $N \subseteq M$ there is *at most one* function $f : N \to S$ such that $f(x) = \phi(x, f|_{N^{\prec x}}) = \phi(x, f|_{M^{\prec x}})$.

Proof: First observe that if $N \subseteq M$ is downward closed and $x \in N$, then $N^{\prec x} = M^{\prec x}$. Assume that there exist a downward closed subset $N \subseteq M$ and two *different* functions f_1 and f_2 from N to S with the property. Therefore, the set $N' := \{\, x \in N \mid f_1(x) \neq f_2(x) \,\}$ is non-empty. By well-foundedness, N' has a minimal element y. By minimality of y, $f_1|_{M^{\prec y}} = f_2|_{M^{\prec y}}$. Therefore $f_1(y) = \phi(y, f_1|_{M^{\prec y}}) = \phi(y, f_2|_{M^{\prec y}}) = f_2(y)$, contradicting the assumption.

Part 2: If N_1 and N_2 are downward closed subsets of M and the functions $f_1 : N_1 \to S$ and $f_2 : N_2 \to S$ satisfy $f_i(x) = \phi(x, f_i|_{M^{\prec x}})$ for all $x \in N_i$ $(i = 1, 2)$, then $f_1(x) = f_2(x)$ for all $x \in N_1 \cap N_2$.

Proof: Define $N_0 := N_1 \cap N_2$ and $f_i' = f_i|_{N_0}$ for $i = 1, 2$. Clearly N_0 is downward closed and for all $x \in N_0$ and $i = 1, 2$ we have $f_i'(x) = f_i(x) = \phi(x, f_i|_{M^{\prec x}}) = \phi(x, f_i'|_{M^{\prec x}})$. By part 1, there is at most one function from N_0 to S with this property, so $f_1' = f_2'$, and therefore $f_1(x) = f_2(x)$ for all $x \in N_1 \cap N_2$.

Part 3: For every $y \in M$ there exists a function $f_y : M^{\preceq y} \to S$ such that $f_y(x) = \phi(x, f_y|_{M^{\prec x}})$ for all $x \in M^{\preceq y}$.

Proof: We use well-founded induction over $\succ$. Let $y \in M$. By the induction hypothesis, for every $z \prec y$ there exists a function $f_z : M^{\preceq z} \to S$ such that $f_z(x) = \phi(x, f_z|_{M^{\prec x}})$ for all $x \in M^{\preceq z}$. By part 2, all functions f_z agree on the intersections of their domains. Define the function $f_y : M^{\preceq y} \to S$ by $f_y(x) = f_x(x)$ for $x \prec y$ and by $f_y(y) = \phi(y, f_y|_{M^{\prec y}})$. The function f_y has the desired property for $x = y$ by construction and for all $x \prec y$ by the induction hypothesis (since $f_y(x) = f_x(x)$ for $x \prec y$ and f_x has the desired property).

Part 4: There exists a function $f : M \to S$ such that $f(x) = \phi(x, f|_{M^{\prec x}})$ for all $x \in M$.

Proof: Define $f : M \to S$ by $f(x) = f_x(x)$.

The claim of the theorem follows now from part 1 (for $N := M$) and part 4. $\square$

The well-founded recursion scheme generalizes terminating recursive programs.

Note that functions defined by well-founded recursion need *not* be computable, in particular since for many well-founded orderings the sets $M^{\prec x}$ may be infinite.

1.4 Multisets

Let M be a set. A *multiset* S over M is a mapping $S : M \to \mathbb{N}$. We interpret $S(m)$ as the number of occurrences of elements m of the base set M within the multiset S.

Example. $S = \{a, a, a, b, b\}$ is a multiset over $\{a, b, c\}$, where $S(a) = 3$, $S(b) = 2$, $S(c) = 0$.

We say that m is an *element* of S, if $S(m) > 0$.

We use set notation ($\in$, $\subseteq$, $\cup$, $\cap$, etc.) with analogous meaning also for multisets, e. g.,

$$\begin{aligned}
m \in S \quad &:\Leftrightarrow \quad S(m) > 0 \\
(S_1 \cup S_2)(m) \quad &:= \quad S_1(m) + S_2(m) \\
(S_1 \cap S_2)(m) \quad &:= \quad \min\{S_1(m), S_2(m)\} \\
(S_1 - S_2)(m) \quad &:= \quad \begin{cases} S_1(m) - S_2(m) & \text{if } S_1(m) \geq S_2(m) \\ 0 & \text{otherwise} \end{cases} \\
S_1 \subseteq S_2 \quad &:\Leftrightarrow \quad S_1(m) \leq S_2(m) \text{ for all } m \in M
\end{aligned}$$

A multiset S is called *finite*, if the set $\{\, m \in M \mid S(m) > 0 \,\}$ is finite.

From now on we only consider finite multisets.

Multiset Orderings

Let $(M, \succ)$ be an abstract reduction system. The *multiset extension* of $\succ$ to multisets over M is defined by

$$
\begin{array}{l}
S_1 \succ_{\text{mul}} S_2 \text{ if and only if} \\
\qquad \text{there exist multisets } X \text{ and } Y \text{ over } M \text{ such that} \\
\qquad\qquad \emptyset \neq X \subseteq S_1, \\
\qquad\qquad S_2 = (S_1 - X) \cup Y, \\
\qquad\qquad \forall y \in Y\ \exists x \in X\colon x \succ y
\end{array}
$$

Lemma 1.9 (König's Lemma) *Every finitely branching tree with infinitely many nodes contains an infinite path.*

Theorem 1.10
(a) If $\succ$ is transitive, then $\succ_{\text{mul}}$ is transitive.

(b) If $\succ$ is irreflexive and transitive, then $\succ_{\text{mul}}$ is irreflexive.

(c) If $\succ$ is a well-founded ordering, then $\succ_{\text{mul}}$ is a well-founded ordering.

(d) If $\succ$ is a strict total ordering, then $\succ_{\text{mul}}$ is a strict total ordering.

Proof. see Baader and Nipkow, page 22–24. □

The multiset extension as defined above is due to Dershowitz and Manna (1979).

There are several other ways to characterize the multiset extension of a binary relation. The following one is due to Huet and Oppen (1980):

Let $(M, \succ)$ be an abstract reduction system. The *(Huet/Oppen) multiset extension* of $\succ$ to multisets over M is defined by

$$
\begin{array}{l}
S_1 \succ_{\text{mul}}^{\text{HO}} S_2 \text{ if and only if} \\
\qquad S_1 \neq S_2 \text{ and} \\
\qquad\qquad \forall m \in M\colon \big(S_2(m) > S_1(m) \\
\qquad\qquad\qquad\qquad \Rightarrow\ \exists m' \in M\colon m' \succ m \text{ and } S_1(m') > S_2(m')\big)
\end{array}
$$

A third way to characterize the multiset extension of a binary relation $\succ$ is to define it as the transitive closure of the relation $\succ_{\text{mul}}^1$ given by

$$
\begin{array}{l}
S_1 \succ_{\text{mul}}^1 S_2 \text{ if and only if} \\
\qquad \text{there exists } x \in S_1 \text{ and a multiset } Y \text{ over } M \text{ such that} \\
\qquad\qquad S_2 = (S_1 - \{x\}) \cup Y, \\
\qquad\qquad \forall y \in Y\colon x \succ y
\end{array}
$$

For strict partial orderings all three characterizations of $\succ_{\text{mul}}$ are equivalent:

Theorem 1.11 *If $\succ$ is a strict partial ordering, then*
(a) $\succ_{\text{mul}} = \succ_{\text{mul}}^{\text{HO}}$,
(b) $\succ_{\text{mul}} = (\succ_{\text{mul}}^{1})^{+}$.

Proof. (a) see Baader and Nipkow, page 24–26. (b) Exercise. □

Note, however, that for an arbitrary binary relation $\succ$ all three relations $\succ_{\text{mul}}$, $\succ_{\text{mul}}^{\text{HO}}$, and $(\succ_{\text{mul}}^{1})^{+}$ may be different.

1.5 Complexity Theory Prerequisites

A *decision problem* is a subset $L \subseteq \Sigma^*$ for some fixed finite alphabet Σ.

The function $\text{chr}(L, x)$ denotes the *characteristic function* for some decision problem L and is defined by $\text{chr}(L, u) = 1$ if $u \in L$ and $\text{chr}(L, u) = 0$ otherwise.

P and NP

A decision problem is called *solvable in polynomial time* if its characteristic function can be computed in polynomial time. The class of all polynomial-time decision problems is denoted by P.

We say that a decision problem L is in *NP* if there is a predicate $Q(x, y)$ and a polynomial $p(n)$ such that for all $u \in \Sigma^*$ we have

(i) $u \in L$ if and only if there is a $v \in \Sigma^*$ with $|v| \leq p(|u|)$ and $Q(u, v)$ holds, and

(ii) the predicate Q is in P.

Intuitively, a decision problem is in P, if we can *solve* it in polynomial time, and it is in NP, if we can *verify* a solution (namely the string v in the definition of NP) in polynomial time.

Reducibility, NP-Hardness, NP-Completeness

A decision problem L is *polynomial-time reducible* to a decision problem L' if there is a function g computable in polynomial time such that for all $u \in \Sigma^*$ we have $u \in L$ iff $g(u) \in L'$.

For example, if L is polynomial-time reducible to L' and $L' \in \mathrm{P}$ then $L \in \mathrm{P}$.

A decision problem is *NP-hard* if every problem in NP is polynomial-time reducible to it.

A decision problem is *NP-complete* if it is NP-hard and in NP.

The following properties are equivalent:

(i) There exists some NP-complete problem that is in P.

(ii) $\mathrm{P} = \mathrm{NP}$.

The question whether P equals NP or not is probably the most famous unsolved problem in theoretical computer science.

All known algorithms for NP-complete problems have an exponential time complexity in the worst case.