An alternative proof of SAT NP-completeness

Bruno Escoffier*, Vangelis Th. Paschos*

Résumé

Nous donnons une preuve de la **NP**-complétude de SAT en se basant sur une caractérisation logique de la classe **NP** donnée par Fagin en 1974. Ensuite, nous illustrons une partie de la preuve en montrant comment deux problèmes bien connus, le problème de MAX STABLE et de 3-COLORATION peuvent s'exprimer sous forme conjonctive normale. Enfin, dans le même esprit, nous redémontrons la **min NPO**-complétude du problème de MIN WSAT sous la stricte-réduction.

Mots-clefs : logique du second ordre, **NP**-complétude, réductions.

Abstract

We give a proof of SAT's **NP**-completeness based upon a syntaxic characterization of **NP** given by Fagin at 1974. Then, we illustrate a part of our proof by giving examples of how two well-known problems, MAX INDEPENDENT SET and 3-COLORING, can be expressed in terms of CNF. Finally, in the same spirit we demonstrate the **min NPO**-completeness of MIN WSAT under strict reductions.

Key words : **NP**-completeness, reductions, second order logic.

1 Proof of Cook's theorem

According to Fagin's characterization for **NP** ([3]), any $\Pi \in \mathbf{NP}$ can be written in the following way. Assume a finite structure $(U, \mathcal{P})$ where U is a set of variables, called the *universe* and $\mathcal{P}$ is a set of predicates $P_1, P_2, \ldots, P_\ell$ of respective arities $k_1, k_2, \ldots, k_\ell$.

* LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. `{escoffier,paschos}@lamsade.dauphine.fr`

Pair $(U, \mathcal{P})$ is an instance of Π. Solving Π on this instance consists of determining a set $\mathcal{S} = \{S_1, S_2, \dots S_p\}$ of predicates on U satisfying a logical formula of the form: $\Psi(\mathcal{P}, S_1, S_2, \dots, S_p)$. In other words, an instance of Π consists of the specification of $P_1, P_2, \dots, P_\ell$ and of U; it is a *yes*-one if one can determine a set of predicates $\mathcal{S} = \{S_1, S_2, \dots S_p\}$ satisfying $\Psi(\mathcal{P}, \mathcal{S})$.

As an example, consider 3-COLORING, where one wishes to answer if the vertices of a graph G can be legally colored with three colors. Here, finite structure $(U, \mathcal{P}) = (V, G)$, where $V = \{v_1, \dots, v_n\}$ is the vertex set of G. This graph is represented by predicate G of arity 2 where $G(x, y)$ iff vertex x is adjacent to vertex y. A graph G is 3-colorable iff:

$$\begin{aligned}\exists S_1 \exists S_2 \exists S_3 \quad & \Big(\forall x S_1(x) \vee S_2(x) \vee S_3(x)\Big) \\ \wedge \ & \Big(\forall x \Big(\neg S_1(x) \wedge \neg S_2(x)\Big) \vee \Big(\neg S_1(x) \wedge \neg S_3(x)\Big) \vee \Big(\neg S_2(x) \wedge \neg S_3(x)\Big)\Big) \\ \wedge \ & \Big(\forall x \forall y \Big(\big(S_1(x) \wedge S_1(y)\big) \vee \big(S_2(x) \wedge S_2(y)\big) \vee \big(S_3(x) \wedge S_3(y)\big)\Big) \Rightarrow \neg G(x, y)\Big)\end{aligned}$$

The rest of this section is devoted to the proof of the **NP**-completeness of SAT, i.e., to an alternative proof of the seminal Cook's theorem ([2]). In fact, we will prove that any instance of a problem Π in **NP** (expressed as described previously) can be transformed in polynomial time into a CNF (i.e., an instance of SAT) in such a way the latter is satisfiable iff the former admits a model.

Let Π be a problem defined by $\exists \mathcal{S} \Psi(\mathcal{P}, \mathcal{S})$. Without loss of generality, we can rewrite $\Psi(\mathcal{P}, \mathcal{S})$ in prenex form and redefine Π as $\exists \mathcal{S} Q_1(x_1) \dots Q_r(x_r) \Phi(x_1, \dots, x_r, \mathcal{P}, \mathcal{S})$, where Q_i, $i = 1, \dots, r$, are quantifiers and Φ quantifier-free.

In the first part of the proof, we are going to build in polynomial time a formula φ (depending on Π and on its instance represented by $\mathcal{P} = P_1, P_2, \dots, P_\ell$) such that φ is satisfiable iff there exists $\mathcal{S}$ satisfying formula $\Psi(\mathcal{P}, \mathcal{S})$ (recall that $\mathcal{S}$ is a p-tuple of predicates $S_1, S_2, \dots, S_p$). Then, we will show how one can modify construction above in order to get a CNF φ_S (instance of SAT) satisfiable iff φ do so.

We first build φ. For this, denote by r_i the arity of predicate S_i in the second-order formula describing Π, and by r the number of its quantifiers. Note that neither r_i's nor r depend on the instance of Π (the dependence of Φ on this instance is realized via predicates $P_i(x_{i_1}, \dots, x_{i_{k_i}})$).

Consider an instance of Π, and denote by $v_1, v_2, \dots, v_n$ the variables of set U. We will build a formula φ on $\sum_{j=1}^{p} n^{r_j}$ variables $y^j_{i_1, i_2, \dots, i_{r_j}}$, where $j \in \{1, \dots, p\}$ and $(i_1, i_2, \dots, i_{r_j}) \in \{1, 2, \dots, n\}^{r_j}$. In this way we will be able to specify a bijection f between the set of p-tuples of predicates $S_1, S_2, \dots, S_p$ of arities $r_1, r_2, \dots, r_p$, respectively, on $\{v_1, v_2, \dots, v_n\}$ and the set of the truth assignments for φ. If $\mathcal{S} = (S_1, S_2, \dots, S_p)$

is such a p-tuple of predicates, we define $f(\mathcal{S})$ as the following truth-value: *variable* $y^j_{i_1,i_2,\ldots,i_{r_j}}$ ***is true** iff* $(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}}) \in S_j$. Once this bijection f defined, we will inductively construct φ so that the following property is preserved:

$$\mathcal{S} \models Q_1\,(x_1)\,Q_2\,(x_2)\ldots Q_r\,(x_r)\,\Phi\,(x_1, x_2, \ldots, x_r, \mathcal{P}, \mathcal{S}) \Longleftrightarrow f\,(\mathcal{S}) \models \varphi \qquad (1)$$

We start by eliminating quantifiers. For this, remark that, for any formula φ :

- $(\forall x \varphi(x, \mathcal{P}, \mathcal{S})) \Longleftrightarrow \varphi(x = v_1, \mathcal{P}, \mathcal{S}) \wedge \varphi(x = v_2, \mathcal{P}, \mathcal{S}) \wedge \ldots \wedge \varphi(x = v_n, \mathcal{P}, \mathcal{S})$;
- $(\exists x \varphi(x, \mathcal{P}, \mathcal{S})) \Longleftrightarrow \varphi(x = v_1, \mathcal{P}, \mathcal{S}) \vee \varphi(x = v_2, \mathcal{P}, \mathcal{S}) \vee \ldots \vee \varphi(x = v_n, \mathcal{P}, \mathcal{S})$.

In this way, we can, in r steps, transform formula $Q_1(x_1)\ldots Q_r(x_r)\Phi(x_1, x_2, \ldots, x_r, \mathcal{P}, \mathcal{S})$ into one consisting of n^r conjunctions or disjunctions of formulæ $\Phi(x_1 = v_{i_1}, x_2 = v_{i_2}, \ldots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S})$. Formally, this new formula $\Psi'(\mathcal{P}, \mathcal{S})$ can be written as follows:

$$\bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \ldots \bigodot_{i_r=1}^{n} \Phi\,(x_1 = v_{i_1}, x_2 = v_{i_2}, \ldots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S})$$

where the ith $\bigodot$ stands for $\vee$ if $Q_i = \exists$ and for $\wedge$ if $Q_i = \forall$.

Now, $\varphi = t(\Psi')$ is built by induction. If Ψ' is an elementary formula, then:

1. if $\Psi' = S_j(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}})$, $\varphi = y^j_{i_1,i_2,\ldots,i_{r_j}}$;
2. if $\Psi' = P_j(v_{i_1}, v_{i_2}, \ldots, v_{i_{k_j}})$, $\varphi = \mathbf{true}$ if the instance is such that $(v_{i_1}, v_{i_2}, \ldots, v_{i_{k_j}}) \in P_j$ and **false** otherwise;
3. if Ψ' is formula $v_i = v_j$, then $\varphi = \mathbf{true}$ if $i = j$ and **false** otherwise.

Construction just described guarantees (1): in case 1, $\mathcal{S}$ verifies $\Psi'(\mathcal{P}, \mathcal{S})$ iff $(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}}) \in S_j$, i.e., iff $y^j_{i_1,i_2,\ldots,i_{r_j}}\mathbf{true}$, therefore, iff $f(S)$ satisfies φ; in cases 2 and 3, either any $\mathcal{S}$ verifies Ψ', i.e., φ is a tautology, or no $\mathcal{S}$ verifies Ψ', i.e., φ is not satisfiable.

Assume now that Ψ' is non-elementary (i.e., composed by elementary formulæ); then,

- if $\Psi' = \neg\Psi''$, then $\varphi = t(\Psi') = \neg t(\Psi'')$;
- if $\Psi' = \Psi_1 \wedge \Psi_2$, then $\varphi = t(\Psi') = t(\Psi_1) \wedge t(\Psi_2)$;
- if $\Psi' = \Psi_1 \vee \Psi_2$, then $\varphi = t(\Psi') = t(\Psi_1) \vee t(\Psi_2)$.

Dealing with the first of items above:

$$\mathcal{S} \models \Psi' \Longleftrightarrow \mathcal{S} \not\models \Psi'' \Longleftrightarrow f(\mathcal{S}) \not\models t(\Psi'') \Longleftrightarrow f(\mathcal{S}) \models \neg t(\Psi'')$$

For the second one (the third item is similar to the second one up to the replacement of "$\wedge$" by "$\vee$") we have:

$$\begin{aligned}\mathcal{S} \models \Psi' \Longleftrightarrow \mathcal{S} \models \Psi_1 \wedge \mathcal{S} \models \Psi_2 &\Longleftrightarrow f(\mathcal{S}) \models t(\Psi_1) \wedge f(\mathcal{S}) \models t(\Psi_2) \\ &\Longleftrightarrow f(\mathcal{S}) \models t(\Psi_1) \wedge t(\Psi_2)\end{aligned}$$

We finally obtain a formula φ on $\sum_j n^{r_j}$ variables of size $n^r|\Phi|$. Furthermore, given (1), φ is obviously satisfiable iff $\exists \mathcal{S} \Psi(\mathcal{P}, \mathcal{S})$.

In general, φ is not CNF. We will build in polynomial time a CNF φ_S satisfiable iff φ does so. From so on, we assume that, when we define Π by $\exists \mathcal{S} Q_1(x_1) \dots Q_r(x_r) \Phi(x_1, \dots, x_r, \mathcal{P}, \mathcal{S})$, Φ is CNF.

Denote by $\varphi_b(i_1, i_2, \dots, i_r)$ the image with respect to t of $\Phi(x_1 = v_{i_1}, x_2 = v_{i_2}, \dots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S})$. All these formulæ φ_b are, by construction, CNF and

$$\varphi = \bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \dots \bigodot_{i_r=1}^{n} \varphi_b(i_1, i_2, \dots, i_r)$$

where the $\bigodot$ are as previously. Starting from φ we will construct, in a *bottom-up* way, formula φ_S in r steps (removing one quantifier per step). Note that if no quantifier does exist, then φ is CNF.

Suppose that q quantifiers remain to be removed. In other words, φ is satisfiable iff the following formula is satisfiable:

$$\bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \dots \bigodot_{i_q=1}^{n} \left(C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q} \right)$$

where $C_i^{i_q}$ are disjunctions of literals.

If qth $\bigodot$ is $\wedge$, i.e., if qth quantifier is $\forall$, then$\wedge_{i_q=1}^{n}(C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q})$ is a conjunction of nm clauses, and consequently, we pass to $(q-1)$th quantifier.

If qth $\bigodot$ is $\vee$, things are somewhat more complicated. In this case, we define n new variables z^{i_q}, $i_q = 1, \dots, n$, and consider the following formula:

$$\varphi_q = \left(\bigvee_{i_q=1}^{n} z^{i_q} \right) \bigwedge \left(\bigwedge_{i_q=1}^{n} \left(z^{i_q} \Rightarrow \left(\bigwedge_{j=1}^{m} C_j^{i_q} \right) \right) \right)$$

Here, formula $\vee_{i_q=1}^{n} (C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q})$ is satisfiable iff formula φ_q does so. In fact,

- if a truth assignment satisfies the former, then for at least one q_0 conjunction of $C_j^{i_{q_0}}$ is true; then, we can extend this assignment by $z_{i_{q_0}} = \mathbf{true}$ and $z_{i_q} = \mathbf{false}$ if $q \neq q_0$;
- if a truth assignment satisfies φ_q, clause $(\vee_{i_q=1}^{n} z^{i_q})$ indicates that at least one $z_{i_{q_0}}$ is true; implication corresponding to this fact shows that conjunction of $C_j^{i_{q_0}}$ is true, and it suffices to restrict this truth assignment in order to satisfy formula $\vee_{i_q=1}^{n}$ $(C_1^{i_q} \wedge C_2^{i_q} \wedge \ldots \wedge C_m^{i_q})$.

Let us finally write φ_q in CNF. Note that:

$$z^{i_q} \Rightarrow \left(\bigwedge_{j=1}^{m} C_j^{i_q}\right) \equiv \left(\neg z^{i_q}\right) \vee \left(\bigwedge_{j=1}^{m} C_j^{i_q}\right) \equiv \bigwedge_{j=1}^{m} \left(\neg z_{i_q} \vee C_j^{i_q}\right)$$

In other words, $\neg z_{i_q} \vee C_j^{i_q}$ is a disjunction of literals. So, $\vee_{i_q=1}^{n} (C_1^{i_q} \wedge C_2^{i_q} \wedge \ldots \wedge C_m^{i_q})$ is satisfiable iff the following CNF formula is satisfiable:

$$\left(\bigvee_{i_q=1}^{n} z^{i_q}\right) \bigwedge \left(\bigwedge_{i_q=1}^{n} \bigwedge_{j=1}^{m} \left(\neg z^{i_q} \vee C_j^{i_q})\right)\right)$$

In all, we have added n new variables and constructed $1 + nm$ clauses. Obviously, construction described is polynomial. After r steps, we get a CNF φ_S satisfiable iff φ is satisfiable and overall construction is polynomial since each of its steps is polynomial (r does not depend on instance parameters). The proof of Cook's theorem is now complete.

Let us note that an analogous proof has pointed out to us after having accomplished what it has just presented. It is given by Immerman in [4]. Immerman's proof is quite condensed, and based upon another version of Fagin's theorem. Furthermore, the type of reduction used, called *first-order reduction*, is, following the author, weaker than classical Karp's reduction. This is not the case of our proof which, to our opinion, is a Karp's reduction.

2 Constructing CNFs for MAX INDEPENDENT SET and 3-COLORING

2.1 MAX INDEPENDENT SET

An instance of MAX INDEPENDENT SET consists of a graph $G(V, E)$, with $|V| = n$ and $|E| = m$, and an integer K. The question is if there exists a set $V' \subseteq V$, with

$|V'| \geqslant K$ such that no two vertices in V' are linked by an edge. The most natural way of writing this problem as a logical formula is the following:

$$\begin{aligned}\exists S \qquad & \forall x \forall y \, (S(x) \wedge S(y)) \Rightarrow \neg G(x,y) \\ & \wedge \exists y_1 \exists y_2 \neq y_1 \ldots \exists y_K \neq y_1 \ldots y_{K-1} S\,(y_1) \wedge S\,(y_2) \ldots \wedge S\,(y_K)\end{aligned}$$

However, in this form the number of quantifiers depends on K, therefore on problem's instance and transformation of Section 1 is no more polynomial. In order to preserve polynomiality of transformation, we are going to express MAX INDEPENDENT SET a problem of determining a permutation P on the vertices of G such that the the K first vertices of P form an independent set. Consider a predicate S of arity 2 such that $S(v_i, v_j)$ iff $v_j = P[v_i]$. MAX INDEPENDENT SET can be formulated as follows (in this formulation, $v_i \leqslant v_j$ means $i \leqslant j$) :

$$\begin{aligned}\exists S \qquad & \Big(\forall x \exists y S(x,y)\Big) \wedge \Big(\forall x \forall y \forall z \big(S(x,y) \wedge S(x,z)\big) \Rightarrow y = z\Big) \\ & \wedge \Big(\forall x \forall y \forall z \big(S(x,z) \wedge S(y,z)\big) \Rightarrow x = y\Big) \\ & \wedge \Big(\forall x \forall y \forall z \forall t \big(x \neq y \wedge S(x,z) \wedge S(y,t) \wedge z \leqslant v_K \wedge t \leqslant v_K\big) \Rightarrow \neg G(x,y)\Big)\end{aligned}$$

Here, first line expresses the fact that predicate S represents a function of the vertex-set in itself, the second one that this function is injective (consequently, bijective also) ; finally, third line indicates that the K first vertices $(v_1, \ldots, v_K)$ of $P[V]$ form an independent set. Formula above is rewritten in prenex form as follows:

$$\begin{aligned}\exists S \; \forall x \forall y \forall z \forall t \exists u \qquad & S(x,u) \wedge \Big(\neg S(x,y) \vee \neg S(x,z) \vee y = z\Big) \\ & \wedge \Big(\neg S(x,z) \vee \neg S(y,z) \vee x = y\Big) \\ & \wedge \Big(x = y \vee \neg S(x,z) \vee \neg S(y,t) \vee z > v_K \vee t > v_K \vee \neg G(x,y)\Big)\end{aligned}$$

We construct a CNF on $n^2 + n$ variables: n^2 variables $y_{i,j}$ representing the fact that $(v_i, v_j) \in S$, and n variables z^i because the last quantifier is existential. We so get the following clauses:

- clause $z^1 \vee z^2 \ldots \vee z^n$ coming from removal of the existential quantifier;
- n^2 clauses: $\bar{z}^j \vee y_{i,j}$ ($i = 1, \ldots, n$, $j = 1, \ldots, n$);
- $\forall (i,j,k,l) \in \{1, \ldots, n\}^4$ where $k \neq l$, clause $\bar{z}^j \vee \bar{y}_{i,k} \vee \bar{y}_{i,l}$;
- $\forall (i,j,k,l) \in \{1, \ldots, n\}^4$ where $i \neq k$, clause $\bar{z}^j \vee \bar{y}_{i,l} \vee \bar{y}_{k,l}$;
- $\forall (i,j,k,l,m) \in \{1, \ldots, n\}^5$ where $i \neq k$, $l \leqslant K$ and $m \leqslant K$ is such that edge $(v_i, v_k) \in E$, clause $\bar{z}^j \vee \bar{y}_{i,l} \vee \bar{y}_{k,m}$.

We so obtain a formula on n^2+n variables with at most $mnK^2+2(n^4-n^3)+n^2+1 \leqslant O(n^5)$ clauses.

2.2 3-COLORING

A graph G of order n is 3-colorable if there exists S_1, S_2, et S_3 such that:

$$\begin{aligned}\forall x \forall y \quad & \Big(S_1(x) \vee S_2(x) \vee S_3(x)\Big) \;\wedge\; \Big(\neg S_1(x) \vee \neg S_2(x)\Big) \;\wedge\; \Big(\neg S_1(x) \vee \neg S_3(x)\Big) \\ \wedge\; & \Big(\neg S_2(x) \vee \neg S_3(x)\Big) \;\wedge\; \Big(\neg G(x,y) \vee \neg S_1(x) \vee \neg S_1(y)\Big) \\ \wedge\; & \Big(\neg G(x,y) \vee \neg S_2(x) \vee \neg S_2(y)\Big) \;\wedge\; \Big(\neg G(x,y) \vee \neg S_3(x) \vee \neg S_3(y)\Big)\end{aligned}$$

Remark that the formula above is the CNF equivalent of the 3-COLORING formula seen in Section 1. Formula φ_S is then defined on:

- $3n$ variables y_i^j, $j = 1, \ldots, 3$ and $i = 1, \ldots, n$; $y_i^j = \mathbf{true}$ if v_i receives color j;
- n series of clauses $(y_i^1 \vee y_i^2 \vee y_i^3) \wedge (\bar{y}_i^1 \vee \bar{y}_i^2) \wedge (\bar{y}_i^1 \vee \bar{y}_i^3) \wedge (\bar{y}_i^2 \vee \bar{y}_i^3)$ (where i goes from 1 to n); series corresponding to index i represents the fact that vertex v_i receives one and only one color;
- clauses representing constraints on adjacent vertices, i.e., for any edge (v_i, v_j) of G, v_i and v_j are colored with different colors: $(\bar{y}_i^1 \vee \bar{y}_j^1) \wedge (\bar{y}_i^2 \vee \bar{y}_j^2) \wedge (\bar{y}_i^3 \vee \bar{y}_j^3)$;

We so get a CNF on $3n$ variables with $4n + 3m$ clauses, any clause containing at most 3 literals.

3 The Min NPO-completeness of MIN WSAT

In MIN WSAT, we are given a CNF φ on n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. Any variable x_i has a non-negative weight w_i, $i = 1, \ldots, n$. We assume that the assignment $x_i = 1$, $i = 1, \ldots, n$ is a feasible solution, and we denote it by $\mathrm{triv}(\varphi)$. The objective of MIN WSAT is to determine an assignment $T = (t_1, \ldots, t_n)$, $t_i \in \{0, 1\}$, on the variables of φ in such a way that (i) T is a model for φ and (ii) quantity $\sum_{i=1}^{n} t_i w_i$ is minimized.

Always based upon Fagin's characterization of **NP**, we show in this section the **Min NPO**-completeness of MIN WSAT under a kind of approximation preserving reduction, originally defined in [5], called *strict reduction*. The class **Min NPO** is the class of minimization **NPO** problems. An optimization problem is in **NPO** if its decision version is in **NP** (see [1] for more details about definition of **NPO**). More formally, an **NPO** problem Π is defined as a four-tuple $(\mathcal{I}, \mathrm{sol}, m, \mathrm{opt})$ such that: $\mathcal{I}$ is the set of instances of Π and it can be recognized in polynomial time; given $x \in \mathcal{I}$, $\mathrm{sol}(x)$ denotes the set of feasible solutions of x; for every $y \in \mathrm{sol}(x)$, $|y|$ is polynomial in $|x|$; given any x and any y polynomial in $|x|$, one can decide in polynomial time if $y \in \mathrm{sol}(x)$; given $x \in \mathcal{I}$ and $y \in \mathrm{sol}(x)$, $m(x, y)$ denotes the value of y for x; m is polynomially computable and is commonly called feasible value; finally, $\mathrm{opt} \in \{\max, \min\}$. We assume that any instance x of any **NPO** problem admits at least one feasible solution, denoted by $\mathrm{triv}(x)$, computable in polynomial time.

Given an instance x of Π, we denote by $\mathrm{opt}(x)$ the value of an optimal solution of x. For an approximation algorithm A computing a feasible solution y for x with value $m_{\mathtt{A}}(x, y)$, its approximation ratio is defined as $r_{\Pi}^{\mathtt{A}}(x, y) = m_{\mathtt{A}}(x, y)/\mathrm{opt}(x)$.

Consider two **NPO** problems $\Pi = (\mathcal{I}, \mathrm{sol}, m, \mathrm{opt})$ and $\Pi' = (\mathcal{I}', \mathrm{sol}', m', \mathrm{opt})$. A *strict reduction* is a pair (f, g) of polynomially computable functions, $f : \mathcal{I} \to \mathcal{I}'$ and $g : \mathcal{I} \times \mathrm{sol}' \to \mathrm{sol}$ such that:

- $\forall x \in \mathcal{I}$, $x \mapsto f(x) \in \mathcal{I}'$;
- $\forall y \in \mathrm{sol}'(f(x))$, $y \mapsto g(x, y) \in \mathrm{sol}(x)$;
- if r is an approximation measure, then $r_{\Pi}(x, g(x, y))$ is as good as $r_{\Pi}(f(x), y)$.

Completeness of MIN WSAT has been originally proved in [5], based upon an extension of Cook's proof ([2]) of SAT **NP**-completeness to optimization problems. As we have already mentioned just above, we give an alternative proof of this result, based upon Fagin's characterization of **NP**.

3.1 Construction of f

Consider a problem $\Pi = (\mathcal{I}, \mathrm{sol}, m, \min)$ and denote by $m(x, y)$ the value of solution y for instance $x \in \mathcal{I}$, set $n = |x|$ and assume two polynomials p and q such that, $\forall x \in \mathcal{I}$, $\forall y \in \mathrm{sol}(x)$, $0 \leqslant |y| \leqslant q(n)$ and $0 \leqslant m(x, y) \leqslant 2^{p(n)}$. As in the proof of [5], we define the following Turing-machine M:

Turing machine M
on input x :
if $x \notin \mathcal{I}$, then reject;
generate a string y such that $\|y\| \leqslant q(n)$;
if $y \notin \mathrm{sol}(x)$, then reject;
write y;
write $m(x, y)$;
accept.

By the proof of Fagin's theorem ([3]), one can construct a second-order formula $\exists S \Phi(S)$ satisfiable iff M accepts x. Revisit this proof for a while; it consists of writing, for an instance x, table $\mathcal{M}$ of $M_x(i, j)$, where $M_x(i, j)$ represents the symbol written at instant i in th jth entry of M (when running on x). If M runs in time n^k, then i and j range from 0 to $n^k - 1$. Second-order formula is then built in such a way that it describes the fact that, for an instance x, there exists such a table $\mathcal{M}$ corresponding to both the way M functions and to the fact that M arrives to acceptance in time $n^k - 1$. Consider that machine's alphabet is $\{0, 1, b\}$, where b is the blank symbol and suppose that when M arrives in acceptance state there is no further changes; this implies that when M attains acceptance state, one can read results of computation on line of $\mathcal{M}$ corresponding to instant $n^k - 1$.

What is of interest for us in Fagin's proof is predicates $S_0(t, s)$ and $S_1(t, s)$ representing the fact that 0, or 1, are written at instant i (encoded by t) on tape-entry j (encoded by s); t and s are two k-tuples $t_1, t_2, \ldots, t_k$ and $s_1, s_2, \ldots, s_k$ of values in $\{0, n - 1\}$. An integer $i \in \{0, n^k - 1\}$ written to the base n can be represented by a k-tuple $t_1, t_2, \ldots, t_k$ in such a way that $i = \sum_{l=1}^{k} t_i n^{i-1}$. In what follows $b(t)$ will denote the value whose $t = (t_1, \ldots, t_k)$ is the representation to the base n ($b(t) = \sum_{l=1}^{k} t_i n^{i-1}$). Predicates S_0 and S_1 allow recovering of value computed by M since this value is written on line corresponding to instant $n^k - 1 = b(t_{\max})$, with $t_{\max} = (n - 1, \ldots, n - 1)$.

By the way M is defined, in case of accepting computation, on the last line of the corresponding table $\mathcal{M}$, solution y and its value $m(x, y)$ are written. Denote by $c_0, c_1, \ldots, c_{p(n)}$ the entries of M where $m(x, y)$ is written (in binary). This value is:

$$m(x, y) = \sum_{j : c_j = 1} 2^j = \sum_{j : \begin{cases} c_j = b(s) \\ S_1(t_{\max}, s) \end{cases}} 2^j$$

We now transform second-order formula in Fagin's theorem into an instance of SAT as described in Section 1. Among other ones, this formula contains variables $y_{t,s}^1$ "representing" predicate S_1 of arity $2k$ (with $t = (t_1, \ldots, t_k)$, $s = (s_1, \ldots, s_k)$, where t_i and s_i, $i = 1, \ldots, k$, range from 0 to $n - 1$). Denote by φ the instance of SAT so-obtained and assume the following weights on variables of φ:

$$\begin{cases} w\left(y^1_{t,s}\right) = 2^j & \text{if } t = t_{\max} \text{ and } c_j = b(s) \\ w(y) = 0 & \text{otherwise} \end{cases}$$

In other words, we consider weight 2^j for variable representing the fact that entry c_j contains an 1.

We so obtain an instance of MIN WSAT and the specification of component f of strict reduction (f, g) transforming an instance of any **NPO** problem Π into an instance φ of MIN WSAT is complete.

3.2 Construction of g

Consider now an instance x de Π and a feasible solution z of $\varphi = f(x)$. Define component g of the reduction as:

$$g(x,z) = \begin{cases} \text{triv}(x) & \text{if } z = \text{triv}(f(x)) \\ \text{the solution accepted by } M & \text{otherwise} \end{cases}$$

Solution accepted by M and its value can both be recovered, as we have discussed, using predicates S_0 and S_1 (recall that truth values of these predicates are immediately deduced from z by the relation "$S_i(t, s)$ iff $y^i_{t,s} = \mathbf{true}$", $i \in \{0, 1\}$). Specification of g is now complete.

3.3 Reduction (f, g) is strict

The pair (f, g) specified above constitute a reduction of Π to MAX SAT. It remains to show that this reduction is strict. We distinguish the following two cases:

- if $z = \text{triv}(f(x))$, then $y = g(x, z) = \text{triv}(x)$; in this case:

  $$m(x,z) \leqslant \sum_{j=0}^{p(n)} 2^j = w(z)$$

  where by $w(z)$ we denote the total weight of solution z;
- otherwise, $y = g(x, z)$ and, by construction:

  $$m(x,y) = \sum_{j:c(j)=1} 2^j = \sum_{j:\left\{\substack{c_j=b(s) \\ S_1(t_{\max},s)}\right.} 2^j = \sum_{j:\left\{\substack{c_j=b(s) \\ y^1_{t_{\max},s}=\mathbf{true}}\right.} 2^j = w(z)$$

Since optimal solution-values of instances x and $f(x)$ are also equal, so do approximation ratios. Therefore reduction specified above is strict.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer, Berlin, 1999.

[2] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. STOC'71*, pages 151–158, 1971.

[3] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of computations*, pages 43–73. American Mathematics Society, 1974.

[4] N. Immerman. *Descriptive complexity*. Springer-Verlag, 1998.

[5] P. Orponen and H. Mannila. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.