Journal talk - Annelyse Thevenin

Sorting by transposition is Difficult

Laurent Bulteau Guillaume Fertin Irena Rusu

LINA, University of Nantes, France

2010

Introduction

In comparative genomics, there exist several distances such that the **transposition distance**. A **transposition** consists in swapping two consecutive sequences.

$$G_1: \; 0 \; \underline{4\;2} \; \underline{1\;3} \; 5$$

$$0 \; 1 \; \underline{3\;4} \; \underline{2} \; 5$$

$$G_2: \; 0\;1\;2\;3\;4\;5 \; (\mathrm{Id})$$

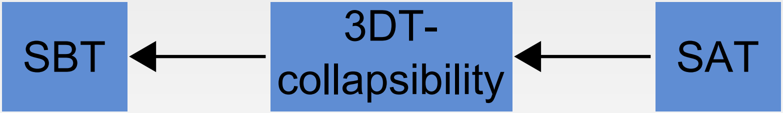
SORTING BY TRANSPOSITION (SBT) [Bafna and Pevzner, 1995]

Find the minimum number of transposition needed to transform a genome into an another.

Aim

Prove that SBT is NP-hard

Tool: Polynomial reductions



I - Three problems

1. Sorting by Transpositions

SBT - Context

- **Introduce** by Bafna and Pevzner - 1995
- There exist lot of **approximation** and **heuristics**
- The best known fixed-ratio algorithm being a **1.375-approximation** [Elias and Hartman - 2006]
- **Variants** of this problem: prefix transposition or distance between strings, etc.

Sorting a permutation by block-interchanges (i.e. exchanges of non-necessarily consecutive sequences) is a polynomial problem [Christie, 1996].

SBT - Definition

Given three integers $0 < i < j < k \leq n$, the **transposition** $\tau_{i,j,k}$ over $[\![0, n]\!]$ is the following permutation:

$$\mathcal{T}_{i,j,k} = \begin{pmatrix} 0 \dots i\text{-}1 & i \;\; i+1 \dots \dots j\text{-}1 & j \;\; j+1 \dots \dots k\text{-}1 & k \;\; k+1 \dots n \\ 0 \dots i\text{-}1 & j \;\; j+1 \dots \dots k\text{-}1 & i \;\; i+1 \dots \dots j\text{-}1 & k \;\; k+1 \dots n \end{pmatrix}$$

Let π be a **permutation** of $[0; n]$. The **transposition distance** $\mathbf{d_t}(\pi)$ from π to Id_n is the minimum value k for which there exist k transpositions $\tau_1, \tau_2, \dots, \tau_k$ such that $\pi \circ \tau_k \circ \dots \circ \tau_2 \circ \tau_1 = Id_n$.

Sorting by Transpositions problem

INPUT: A permutation π, an integer k

QUESTION: Is $d_t(\pi) \leq k$?

I - Three problems

2. 3DT-collapsibility

3DT-instance - definition

Let Σ an alphabet of at most n elements.

A **3DT-instance** I of span n is composed of :

- A word composed by $\bullet$ and distinct letters from Σ, and
- a set of *ordered triples* of elements of Σ, *partitioning* Σ: $T_I = \{(a_i, b_i, c_i) \mid 1 \leq i \leq |T_I|\}$.

Two examples with n=6:

$$I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c2)\}$$
$$I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c2)\}$$

Positions

The function Ψ: $\Sigma \to [1; n]$ is an *injection*. $\Psi(\sigma)$ is the position of σ in the word of I.

- $\sigma_1 \prec \sigma_2$ if $\Psi(\sigma_1) < \Psi(\sigma_2)$
- $\sigma_1 \lhd \sigma_2$ if $\sigma_1 \prec \sigma_2$ and $\nexists x \in \Sigma$, $\sigma_1 \prec x \prec \sigma_2$.
- The function $succ_I$: for all $(a, b, c) \in T_I$, $\Psi(a) \mapsto \Psi(b)$, $\Psi(b) \mapsto \Psi(c)$, and $\Psi(c) \mapsto \Psi(a)$.

$$I = a_1\ c_2\ b_1\ b_2\ c_1\ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c2)\}$$
$$I' = \bullet\ b_2\ \bullet\ c_2\ \bullet\ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c2)\}$$

Triplet well-ordered

Let I be a 3DT-instance, and (a, b, c) be a triple of T_I. Write $i = min\{\Psi(a), \Psi(b), \Psi(c)\}$, $j = succ_I(i)$, and $k = succ_I(j)$.

The triplet $(a, b, c) \in T_I$ is **well-ordered** if we have $i < j < k$. In such case, we write $\tau[a, b, c, \Psi]$ the **transposition** $\tau_{i,j,k}$.

$$I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c2)\}$$
$$I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c2)\}$$

3DT-step

Definition: 3DT-step

Let I be a 3DT-instance with $(a, b, c) \in T_I$ a well-ordered triple. The **3DT-step** of parameter (a, b, c) is the operation written $\xrightarrow{(a,b,c)}$, transforming I into the 3DT-instance I' such that

$$T_{I'} = T_I - (a, b, c) \text{ and}$$

$$\Psi(\sigma) = \tau^{-1}(\Psi(\sigma)).$$

$$I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c2)\}$$

$$I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c2)\}$$

3DT-collapsibility

Definition: 3DT-collapsible

A 3DT-instance I is **3DT-collapsible** if there exists a sequence of 3DT-instances $I_k, I_{k-1}, \ldots, I_0$ such that $I_k = I$, $I_0 = \epsilon$, and $\forall i \in [1; k], \exists (a, b, c) \in T_I, I_i \xrightarrow{(a,b,c)} I_{i-1}$.

I and I' are 3DT-collaspible, since we have

$$I \xrightarrow{(a_1,b_1,c_1)} I' \xrightarrow{(a_2,b_2,c_2)} \epsilon.$$

$$I = a_1\ c_2\ b_1\ b_2\ c_1\ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c2)\}$$
$$I' = \bullet\ b_2\ \bullet\ c_2\ \bullet\ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c2)\}$$

3DT-collapsibility

Definition: 3DT-collapsible

A 3DT-instance I is **3DT-collapsible** if there exists a sequence of 3DT-instances $I_k, I_{k-1}, \ldots, I_0$ such that $I_k = I$, $I_0 = \epsilon$, and $\forall i \in [1;k], \exists (a,b,c) \in T_I, I_i \xrightarrow{(a,b,c)} I_{i-1}$.

3DT-COLLASPIBLITY problem

INPUT: A 3DT-instance I

QUESTION: Is I 3DT-collaspible?

I - Three problems

3. SAT

Definition - SAT

SAT problem

INPUT: Formula in conjunctive normal form ϕ
QUESTION: Is ϕ satisfiable?

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2} \vee x_4)$$

ϕ has two **clauses** C_1 and C_2 (denoted by parentheses), four **boolean variables** (x_1, x_2, x_3, x_4), and three **literals** per clause.

SAT was the first known example of a NP-complet problem.

II - 3DT-collapsibility is NP-hard to decide

Aim

1. **Define** for any boolean formula ϕ, a corresponding 3DT-instance I_ϕ.
2. **Prove** that I_ϕ is 3DT-collapsible iff ϕ is satisfiable.

II - 3DT-collapsibility is NP-hard to decide

1. Definitions

Definition: l-block decomposition

l-block-decomposition $\mathcal{B}$ of a 3DT-instance I of span n is an l-tuple $(s_1, \ldots, s_l)$ such that $s_1 = 0$, for all $h \in [\![1; l-1]\!]$, $s_h < s_{h+1}$ and $s_l < n$.

Example of a 3-block-decomposition of I:

$$\mid a_1 \;\; c_2 \mid b_1 \;\; b_2 \;\; c_1 \mid a_2, \qquad\qquad s_1 = a_1, s_2 = b_1, s_3 = a_2.$$

Definition: Variable - 1/2

A **variable** A of a 3DT-instance I is a pair of triples $A = [(a, b, c), (x, y, z)]$ of T_I.
It is **valid** in an I-block-decomposition $\mathcal{B}$ if:

(i) $\exists h_0 \in [\![1\,;\,l]\!]$ *such that* $block_{I,\mathcal{B}}(b) = block_{I,\mathcal{B}}(x) = block_{I,\mathcal{B}}(y) = h_0$

(ii) $\exists h_1 \in [\![1\,;\,l]\!]$, $h_1 \neq h_0$, *such that* $block_{I,\mathcal{B}}(a) = block_{I,\mathcal{B}}(c) = block_{I,\mathcal{B}}(z) = h_1$

(iii) *if* $x \prec y$, *then we have* $x \lhd b \lhd y$

(iv) $a \prec z \prec c$

Definition: Variable - 2/2

... | .. xyb .. | ... | .. a .. z .. c .. | ...

... | .. b .. y .. x .. | ... | .. a .. z .. c .. | ...

... | .. y .. b .. x .. | ... | .. a .. z .. c .. | ...

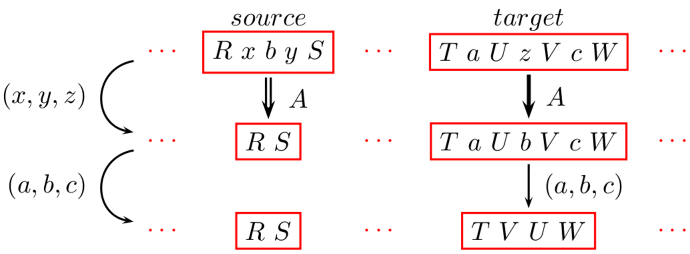
... | .. y .. x .. b .. | ... | .. a .. z .. c .. | ...

output

input

The 3DT-step $I \xrightarrow{(x,y,z)} I'$ is called the **activation** of A (it requires that (x, y, z) is well-ordered).

Definition: Variable - 2/2



The 3DT-step $I \xrightarrow{(x,y,z)} I'$ is called the **activation** of A (it requires that (x, y, z) is well-ordered).

Definition: Basic block

They define 4 **basic blocks**:

- The basic block **var**:

  $[A_1, A_2] = \text{var}(A) \quad = \quad d_1\ y_1\ a\ d_2\ y_2\ e_1\ a'\ e_2\ x_1\ b_1\ f_1\ c'\ z\ b'\ c\ x_2\ b_2\ f_2$

- The basic block **copy**:

  $[A_1, A_2] = \text{copy}(A) \quad = \quad a\ y_1\ e\ z\ d\ y_2\ x_1\ b_1\ c\ x_2\ b_2\ f$

- The basic block **or**:

  $A = \text{or}(A_1, A_2) \quad = \quad a_1\ b'\ z_1\ a_2\ d\ y\ a'\ x\ b\ f\ z_2\ c_1\ e\ c'\ c_2$

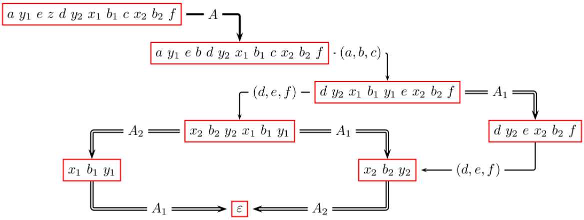
- The basic block **and**:

  $A = \text{and}(A_1, A_2) \quad = \quad a_1\ e\ z_1\ a_2\ c_1\ z_2\ d\ y\ c_2\ x\ b\ f$

The basic block **copy**

The **input** variable: $A = [(a, b, c), (x, y, z)]$

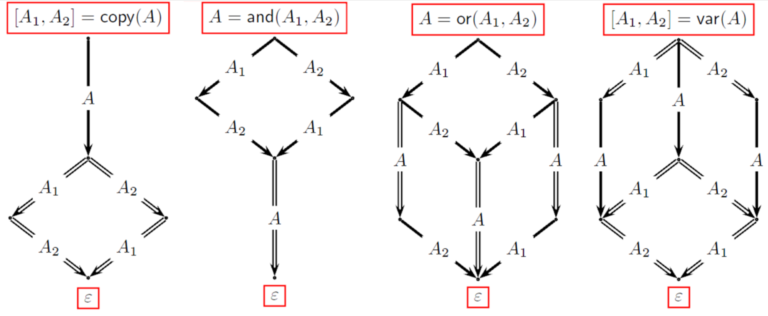
The **output** variables: $A_1 = [(a_1, b_1, c_1)], (x_1, y_1, z_1)]$ and $A_2 = [(a_2, b_2, c_2)], (x_2, y_2, z_2)]$



Behavior graph of the block $[A_1, A_2] = \mathsf{copy}(A)$.

Any of the two output variables can only be activated after the input variable has been activated.

Behavior graph of four basic blocks



II - 3DT-collapsibility is NP-hard to decide

2. Construction of a 3DT-instance

Construction - step 1

Let ϕ be a boolean formula, over the boolean variables $x_1, \ldots, x_m$, given in conjunctive normal from: $\phi = C_1 \wedge C_2 \ldots \wedge C_\gamma$.
The 3DT-instance I_ϕ is defined as an assembling of basic blocks.

1. Create a set of variables

- The variables $X_i, X_i^j, \bar{X}_i$ and $\bar{X}_i^j$ representing all occurrences of x_i and of $\bar{x}_i$
- The variable Γ_C representing the clause C_C
- The variables A_ϕ and A_ϕ^i, representing the formula ϕ.
- The intermediate variables U, $\bar{U}$, V, W and Y.

Construction - step 2

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$$

2. Start with an empty 3DT-instance ϵ and add blocks successively:

(*) Blocks **var** and **copy** defining the variables $X_i, X_i^j, \bar{X}_i$ and $\bar{X}_i^j$

$$[X_i, \bar{X}_i] = \text{var}(A_\phi^i)$$

$$\begin{aligned} [X_i^1, U_i^2] &= \text{copy}(X_i) \\ [X_i^2, U_i^3] &= \text{copy}(U_i^2) \\ &\vdots \\ [X_i^{q_i-2}, U_i^{q_i-1}] &= \text{copy}(U_i^{q_i-2}) \\ [X_i^{q_i-1}, X_i^{q_i}] &= \text{copy}(U_i^{q_i-1}) \end{aligned}$$

$$\begin{aligned} [\bar{X}_i^1, \bar{U}_i^2] &= \text{copy}(\bar{X}_i) \\ [\bar{X}_i^2, \bar{U}_i^3] &= \text{copy}(\bar{U}_i^2) \\ &\vdots \\ &\vdots \\ [\bar{X}_i^{\bar{q}_i-2}, \bar{U}_i^{\bar{q}_i-1}] &= \text{copy}(\bar{U}_i^{\bar{q}_i-2}) \\ [\bar{X}_i^{\bar{q}_i-1}, \bar{X}_i^{\bar{q}_i}] &= \text{copy}(\bar{U}_i^{\bar{q}_i-1}) \end{aligned}$$

Construction - step 2

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2} \vee x_4)$$

2. Start with an empty 3DT-instance ϵ and add blocks successively:

(**) Blocks **or** defining Γ_C

$$\begin{aligned} V_c^2 &= \text{or}(L_1, L_2) \\ V_c^3 &= \text{or}(V_c^2, L_3) \\ &\vdots \\ V_c^{k-1} &= \text{or}(V_c^{k-2}, L_{k-1}) \\ \Gamma_c &= \text{or}(V_c^{k-1}, L_k) \end{aligned}$$

Construction - step 2

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2} \vee x_4)$$

2. Start with an empty 3DT-instance ϵ and add blocks successively:

(***) Blocks **and** defining A_ϕ

$$
\begin{aligned}
W_2 &= \text{and}(\Gamma_1, \Gamma_2) \\
W_3 &= \text{and}(W_2, \Gamma_3) \\
&\vdots \\
W_{\gamma-1} &= \text{and}(W_{\gamma-2}, \Gamma_{\gamma-1}) \\
A_\phi &= \text{and}(W_{\gamma-1}, \Gamma_l)
\end{aligned}
$$

Construction - step 2

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$$

2. Start with an empty 3DT-instance ϵ and add blocks successively:

(****) Blocks **copy** defining A_ϕ^i and Y.

$$[A_\phi^1, Y_2] = \text{copy}(A_\phi)$$
$$[A_\phi^2, Y_3] = \text{copy}(Y_2)$$
$$\vdots$$
$$[A_\phi^{m-2}, Y_{m-1}] = \text{copy}(Y_{m-2})$$
$$[A_\phi^{m-1}, A_\phi^m] = \text{copy}(Y_{m-1})$$

II - 3DT-collapsibility is NP-hard to decide

2. Proof: Let ϕ be satisfiable

Proof: Let ϕ be satisfiable - 1

Let ϕ be **satisfiable**. Let P be the **set of indices** i such that x_i is assigned to true.

Starting from I_ϕ, we can follow a path **3DT-steps** that activates all the variables of I_ϕ in the specific order.

We need six steps to activate all the variables of I_ϕ.

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

1. If $i \in P$, activate X_i in block **var** in (*). Then, we can activate some blocks **copy** in (*).
Otherwise, activate $\bar{X}_i$ in block **var** in (*). Then, we can activate some blocks **copy** in (*).

$$[X_i, \bar{X}_i] = \text{var}(A^i_\phi)$$

$$[X_i^1, U_i^2] = \text{copy}(X_i) \qquad [\bar{X}_i^1, \bar{U}_i^2] = \text{copy}(\bar{X}_i)$$

$$[X_i^2, U_i^3] = \text{copy}(U_i^2) \qquad [\bar{X}_i^2, \bar{U}_i^3] = \text{copy}(\bar{U}_i^2)$$

$$\vdots \qquad \vdots$$

$$[X_i^{q_i-2}, U_i^{q_i-1}] = \text{copy}(U_i^{q_i-2}) \qquad \vdots$$

$$[X_i^{q_i-1}, X_i^{q_i}] = \text{copy}(U_i^{q_i-1}) \qquad [\bar{X}_i^{\bar{q}_i-2}, \bar{U}_i^{\bar{q}_i-1}] = \text{copy}(\bar{U}_i^{\bar{q}_i-2})$$

$$[\bar{X}_i^{\bar{q}_i-1}, \bar{X}_i^{\bar{q}_i}] = \text{copy}(\bar{U}_i^{\bar{q}_i-1})$$

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

2. For each c, since C_C is true, at least one literal λ_{p_0} is true. Using the block **or** in (**), we activate V_c^p and finally Γ_C ($L_{p_0} = X_j^i$ or $L_{p_0} = \bar{X}_j^i$).

$$\begin{aligned} V_c^2 &= \text{or}(L_1, L_2) \\ V_c^3 &= \text{or}(V_c^2, L_3) \\ &\vdots \\ V_c^{k-1} &= \text{or}(V_c^{k-2}, L_{k-1}) \\ \Gamma_c &= \text{or}(V_c^{k-1}, L_k) \end{aligned}$$

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

3. Since all variables Γ_C have been activated, we can activate W_C and A_ϕ using block **and** in (***).

$$
\begin{aligned}
W_2 &= \mathsf{and}(\Gamma_1, \Gamma_2) \\
W_3 &= \mathsf{and}(W_2, \Gamma_3) \\
&\vdots \\
W_{\gamma-1} &= \mathsf{and}(W_{\gamma-2}, \Gamma_{\gamma-1}) \\
A_\phi &= \mathsf{and}(W_{\gamma-1}, \Gamma_l)
\end{aligned}
$$

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

4. Using blocks **copy** in (****), we activate Y_i and $A_\phi^1, \ldots, A_\phi^m$.

$$
\begin{aligned}
[A_\phi^1, Y_2] &= \text{copy}(A_\phi) \\
[A_\phi^2, Y_3] &= \text{copy}(Y_2) \\
&\vdots \\
[A_\phi^{m-2}, Y_{m-1}] &= \text{copy}(Y_{m-2}) \\
[A_\phi^{m-1}, A_\phi^m] &= \text{copy}(Y_{m-1})
\end{aligned}
$$

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

5. Since the variables A^i_ϕ has been activated, we activate the remaining variable X_i or $\bar{X}_i$ and U^j_i or $\bar{U}^j_i$ in the block **var** in (*).

$$[X_i, \bar{X}_i] = \mathsf{var}(A^i_\phi)$$

$$\begin{aligned}
[X^1_i, U^2_i] &= \mathsf{copy}(X_i) \\
[X^2_i, U^3_i] &= \mathsf{copy}(U^2_i) \\
&\vdots \\
[X^{q_i-2}_i, U^{q_i-1}_i] &= \mathsf{copy}(U^{q_i-2}_i) \\
[X^{q_i-1}_i, X^{q_i}_i] &= \mathsf{copy}(U^{q_i-1}_i)
\end{aligned}$$

$$\begin{aligned}
[\bar{X}^1_i, \bar{U}^2_i] &= \mathsf{copy}(\bar{X}_i) \\
[\bar{X}^2_i, \bar{U}^3_i] &= \mathsf{copy}(\bar{U}^2_i) \\
&\vdots \\
&\vdots \\
[\bar{X}^{\bar{q}_i-2}_i, \bar{U}^{\bar{q}_i-1}_i] &= \mathsf{copy}(\bar{U}^{\bar{q}_i-2}_i) \\
[\bar{X}^{\bar{q}_i-1}_i, \bar{X}^{\bar{q}_i}_i] &= \mathsf{copy}(\bar{U}^{\bar{q}_i-1}_i)
\end{aligned}$$

Proof: Let ϕ be satisfiable - 2

6 steps to activate all the variables of I_ϕ

6. In (**), since all variables L_p have been activated, we activate the remaining intermediate variables V_C^P.

$$\begin{aligned} V_c^2 &= \text{or}(L_1, L_2) \\ V_c^3 &= \text{or}(V_c^2, L_3) \\ &\vdots \\ V_c^{k-1} &= \text{or}(V_c^{k-2}, L_{k-1}) \\ \Gamma_c &= \text{or}(V_c^{k-1}, L_k) \end{aligned}$$

Proof: Let ϕ be satisfiable - 3

Every variable has been activated
$\Rightarrow$ the resulting instance is 3DT-collapsible.

If ϕ is satisfiable then I_ϕ is 3DT-collapsible.

II - 3DT-collapsibility is NP-hard to decide

3. Proof: Let I_ϕ is 3DT-collapsible

Let I_ϕ be **3DT-collapsible**. Let $\mathcal{Q}$ be the set of **variables activated before $\mathbf{A}_\phi$** and P the **set of indices** i such that $X_i \in Q$.

3 steps to show that the true assignment defined by $(x_i = true \Leftrightarrow i \in P)$ satisfies the formula ϕ.

1. A_C^i cannot belong to $\mathcal{Q}$ (**copy** (****)). Hence

$$\bar{X}_i \in \mathcal{Q} \Rightarrow X_i \notin \mathcal{Q} \text{ (\textbf{var} in (*))}$$

$$X_i^j \in \mathcal{Q} \Rightarrow X_i \in \mathcal{Q} \text{ (\textbf{copy} in (*))}$$

$$\bar{X}_i^j \in \mathcal{Q} \Rightarrow \bar{X}_i \in \mathcal{Q} \text{ (\textbf{copy} in (*))}$$

Let I_ϕ be **3DT-collapsible**. Let $\mathcal{Q}$ be the set of **variables activated before $\mathbf{A}_\phi$** and P the **set of indices** i such that $X_i \in Q$.

3 steps to show that the true assignment defined by ($x_i = true \Leftrightarrow i \in P$) satisfies the formula ϕ.

2. Since A_ϕ is defined in a block $A_\phi = and(W_{\lambda-1}, \Gamma_\lambda)$ in (***), we necessarily have: $W_{\lambda-1} \in \mathcal{Q}$ and $\Gamma_\lambda \in \mathcal{Q}$.

Since $W_{\lambda-1}$ is defined by $W_{\lambda-1} = and(W_{\lambda-2}, \Gamma_{\lambda-1})$, we also have $W_{\lambda-2}\ in \mathcal{Q}$ and $\Gamma_{\lambda-1} \in \mathcal{Q}$.

Recursively: $\Gamma_C \in \mathcal{Q}$ for each $c \in [\![1; \lambda]\!]$.

Let I_ϕ be **3DT-collapsible**. Let $\mathcal{Q}$ be the set of **variables activated before $\mathbf{A}_\phi$** and P the **set of indices** i such that $X_i \in Q$.

3 steps to show that the true assignment defined by $(x_i = true \Leftrightarrow i \in P)$ satisfies the formula ϕ.

3. For each clause C_c, there exists some p_0 such that the variable L_{p_0} is activated before Γ_c: hence $l_{P_0} \in \mathcal{Q}$.

If the corresponding literal $\lambda_{\underline{p_0}}$ is the j-th occurrence of x_i (resp. $\neg x_i$), then $L_{p_0} = X_i^j$ (resp. $\bar{X}_i^j$) thus $X_i \in \mathcal{Q}$ (resp $\bar{X}_i \in \mathcal{Q}$) and $i \in P$ (resp. $i \notin P$).

The literal λ_{p_0} is true in the truth assignment defined by $(x_i = true \Leftrightarrow i \in P)$.

Theorem

So, if I_ϕ is 3DT-collapsible, they have found a truth assignment such that at least one literal is true in each clause of the formula ϕ, and thus ϕ is satisfiable.

Theorem

So, if I_ϕ is 3DT-collapsible, they have found a truth assignment such that at least one literal is true in each clause of the formula ϕ, and thus ϕ is satisfiable.

Theorem

3DT-collapsibility problem is NP-hard.

Proof: Let ϕ be a boolean formula, and I_ϕ the 3DT-instance defined previously. The construction of I_ϕ is polynomial in the size of ϕ, and ϕ is satisfiable iff I_ϕ is 3DT-collapsible.$\square$

III - SBT is NP-hard to decide

1. Construction of a permutation π_I

Build π_I from I

Aim: Build in polynomial time a permutation π_I such that $I \sim \pi_I$.

Theorem

Let I be a 3DT-instance of span n with $\mathcal{B}$ an I-block-decomposition such that $(I, \mathcal{B})$ is an assembling of basic blocks.

Then there exists a permutation π_I, computable in polynomial time in n, such that $I \sim \pi_I$.

The permutation π_I defined by this theorem is in fact a 3-permutation, i.e. a permutation whose cycle graph contains only 3-cycles.

III - SBT is NP-hard to decide

2. Proof

Proof

1. Given any instance ϕ of **SAT**, create a **3DT-instance** I_ϕ, being an assembling of basic blocks, which is 3DT-collapsible iff ϕ is satisfiable.
2. Then create a **3-permutation** π_{I_ϕ} equivalent to I_ϕ (previous theorem).

The above two steps can be done in **polynomial** time.

Proof

Finally, set $k = \frac{d_b(\pi_{I_\phi})}{3} = \frac{n}{3}$. We then have:

$$\begin{aligned} \phi \text{ is satisfiable} &\Leftrightarrow I_\phi \text{ is 3DT-collapsible} \\ &\Leftrightarrow d_t(\pi_{I_\phi}) = k \text{ (because } \pi_{I_\phi} \sim I_\phi) \\ &\Leftrightarrow d_t(\pi_{I\phi}) \leq k \text{ (because } d_t(\pi) \geq \tfrac{d_b(\pi)}{3}) \end{aligned}$$

Theorem

SORTING BY TRANSPOSITIONS problem is NP-hard.

Conclusions

Conclusion

Main theorem

SORTING BY TRANSPOSITIONS problem is NP-hard.

Corollary

The following two decision problems are NP-hard:

- Given a permutation π of $[\![0; n]\!]$, is the equality $d_t(\pi) = \frac{d_b(\pi)}{3}$ satisfied?
- Given a 3-permutation π of $[\![0; n]\!]$, is the equality $d_t(\pi) = \frac{n}{3}$ satisfied?

Prospect

- A polynomial-time **approximation** scheme?
- Relevant **parameters** for which problem is fixed parameter tractable?