Recognising the Infinite Cycle: A Way of Looking at the Halting Problem

Bohdan Hejna
Institute of Chemical Technology Prague,
Department of Mathematics,
Studentská 6, 166 28 Prague 6, Czech Republic
e-mail: hejnab@vscht.cz

Abstract
The formulations of the *undecidability* of the Halting Problem assume that the computing process being *observed*, the description of which is given on the input of the *'observing'* Turing Machine, is the exact copy of the computing process running in the observing Turing Machine itself (Cantor's diagonal argument). By this way an analogue of stationary state in thermodynamic sense or an *infinite cycle* in computing sense is created, *shielding* now what is to be possibly discovered - the infinite cycle in the observed computing process for a 'normal' input. This shield is the real result of Cantor's diagonal argument which has been used for solving the Halting Problem. We believe that it is possible to recognize the infinite cycle, but with a *time delay* or *staging* in evaluating the *trace* of the observed computing process. Furthermore, the control unit of any Turing Machine is a *finite automaton.* Both these facts enable that the *Pumping Lemma* in the observing Turing Machine is usable and the *general configuration types* are constructed for the observed Turing Machine. This enables (in finite time) us (the observing Turing Machine) to recognize that the computing process in the observed Turing machine has entered into an infinite cycle. These ideas differ from Cantor's diagonal argument.

Keywords: Heat and Information Entropy, Observation, Carnot Cycle, Information Channel, Turing Machine, Infinite Cycle.

1. Introduction

The formulations of the *undecidability* of the Halting Problem assume that the computing process being observed, the description of which is given on the input of the 'observing' Turing Machine, is the exact copy of the computing process running in the observing machine itself (Cantor's diagonal argument in Minski's proof [9]). By this way the Auto-Reference or an *infinite cycle* in computing sense or the Self-Observation in information sense or an analogue of stationary (equilibrium)

Supported by the grant of Ministry of Education of Czech Republic MSM 6046137307.

state in physical (thermodynamic) sense is created, *shielding* now what is to be possibly discovered - the infinite cycle in the observed computing process for a 'normal' input. This shield is the real result of Cantor's diagonal argument which has been used for solving the Halting Problem [9]. However, this shield is, also, a certain image of the sought-after possible infinite cycle. This shield could be, in the thermodynamic point of view, ceased or ended when the performance of the Perpetuum Mobile functionality was possible[1], or by the external activity, by a 'step-aside'. This situation is recognizable and as such *decidable.*

Thus, we show that it is possible to recognize the infinite cycle, but with a *time delay* or *staging* in evaluating the *trace*[2] of the observed computing process.[3] The trace is a message both about the input data and about the structure of the computing process being observed. In this phase the *observing* Turing Machine (we ourselves) is giving the question: "Is there an infinite cycle?" Following the trace the observing machine gains the answer. In our case, the trace is a *recorded sequence of configurations* of the *observed* Turing Machine. These configurations can be simplified to their *general configuration types*, creating now a word of a *regular language.* Furthermore, the control unit of any Turing Machine is a *finite automaton.* Both these facts enable the *Pumping Lemma* in the observing Turing Machine to be usable. In accordance with the Pumping Lemma, we know (the observing Turing Machine knows) that certain general configuration types must be *periodically repeated* in the case of the *infinite length* of their regular language. This fact enables us (the observing Turing Machine) to decide that the observed computing process has entered into an infinite cycle. This event is performed in a finite time and is, by this way, recognizable in finite time too. The method of staging of the observed process will be used.

When the described method is used, to any given computing process, it becomes an *instance of observation.* By application to 'itself' it becomes a *higher instance* of observation, now observing the trace of its previous instances. This sequence of ideas differs from Cantor's diagonal argument.

2. Notion of Turing Computing

Turing Machine (TM) is driven by a *program* which is interpreted by its *Control Unit (CU_{TM}).* The Control Unit CU_{TM} is a *finite automaton* (Mealy's or Moore's *sequential machine*). *The program* for the TM consists of the finite sequence $\overrightarrow{\eta}$ of instructions $\eta_{[\cdot]}$,

$$\overrightarrow{\eta} = (\eta_q)_{q=1}^{q\in\mathbb{N}} = [(s_i,\ x_k,\ s_j,\ y_l,\ D)_q]_{q=1}^{q\in\mathbb{N}},\quad |\overrightarrow{\eta}| \in \mathrm{N} \tag{1}$$

Each of these instructions describes an overwriting rule of a *regular grammar*,

$$s_i \longrightarrow (x_k,\ y_l,\ D)s_j \tag{2}$$

[1]When, e.g., the equation $x = x + 1$ would be solvable.

[2]The *listing*, the *cross-references* and the *memory dump* in the language of programmers.

[3]Instead of the time delay the *staging* is usable, running a longer time in each its repetition.

performed in the given step (time, moment) p, $p \in \mathbb{N}$, of the TM's activity;

- s_i is the i-th *non-terminal symbol* of the regular grammar, or, respectively, it is a *status* of the CU_{TM} within the actual step $p \in \mathbb{N}$ of the TM's activity,
- x_k is an *input terminal symbol* being read from the *input-output tape* of the TM within the actual step p of the TM's activity,
- y_l is an *output terminal symbol* by which the CU_{TM} overwrites the symbol x_k which has been read (in the actual step p of the TM's activity),
- s_j is the *successive* status of the CU_{TM}, given by the instruction for the following step $p+1$ of the TM's activity.

Within the actual step p of the TM's activity the CU_{TM} is changing its status to s_j [this change is based on the status s_i and the symbol x_k has been read ($s_i \xrightarrow{x_k} s_j$)], and is performing the transformation

$$x_k \longrightarrow y_l \tag{3}$$

on the scanned (actual) *position* of the input-output tape,

- D determines the *moving direction* of the *read-write head* of the CU_{TM} after the symbol y_l has been recorded [in the status s_{j_p} (s_{j_p} denotes s_j for the step p) used further as the following one, $s_{j_p} \overset{\text{Def}}{=} s_{i_{p+1}}$], $D \in \{L,\ R\}$.

The value L or R of the symbol D determines the *left slip* or the *right slip* from the actual position on the input-output tape to its (left or right) neighbor after the transformation x_k to y_l has been performed.

The *oriented edge* of the *transition graph* of the CU_{TM} (the finite automaton) is described by the symbol $s_i \xrightarrow{(x_k,\ y_l,\ D)} s_j$. The TM's activity generates a sequence of the instructions *having been performed* in steps p, $\left[(s_{i_p},\ x_{k_p},\ s_{j_p},\ y_{l_p},\ D_p)\right]_{p=1}^{p=p_{\text{last}}}$,

$$s_{i_p} \xrightarrow{(x_{k_p},\ y_{l_p},\ D_p)} s_{j_p},\ \text{further}\ \ s_{j_p} = s_{i_{p+1}} \tag{4}$$

(the edge of the oriented transition graph of the CU_{TM} in the step p), by which the *computing process* ($\overrightarrow{\kappa}$) has gone through (from the first step $p=1$ till, *for this while*, the last step $p = p_{\text{last}}$ of the TM's activity). They are also the overwriting rules of the *regular grammar*, being performed within each step p, $p \geq 1$,

$$s_{i_p} \longrightarrow (x_{k_p},\ y_{l_p},\ D_p)s_{j_p},\ \ s_{j_p}{=}s_{i_{p+1}} \tag{5}$$

By this way a *regular language* of the *words* $(x_{k_p},\ y_{l_p},\ D_p)$ or, respectively, a regular language of the instructions $(s_{i_p},\ x_{k_p},\ s_{j_p},\ y_{l_p},\ D_p)$ having been performed is defined. This second regular language is decribable by the rules (of a regular grammar)

$$S_{i_p} \longrightarrow (s_{i_p},\ x_{k_p},\ s_{j_p},\ y_{l_p},\ D_p)S_{j_p},\ \ S_{j_p} \overset{\text{Def}}{=} S_{i_{p+1}} \tag{6}$$

being applicated in each step $p \geq 1$ of the TM's activity. Thus, this language is to be acceptable by a certain finite automaton with n states $S_{[\cdot]}$.

When this language is <u>*infinite*</u>[4] *(the infinite chain of instruction of finite length),*

[4]Better said, having the *arbitrary* (but finite) *length.*

such its word

$$\left[(s_{i_1},\ x_{k_1},\ s_{j_1},\ y_{l_1},\ D_1),\ \ldots,\ (s_{i_p},\ x_{k_p},\ s_{j_p},\ y_{l_p},\ D_p)\right]_{p=L} \tag{7}$$

of the length L exists that for that finite automaton [with n states $S_{[\cdot]}$ *and the transition rules (4) or (6)] the Pumping Lemma [10, 11] is valid*

$$n \leq L < 2n \tag{8}$$

3. Notion of Auto-Reference

3.1 Auto-Reference in Information Transfer, Self-Observation

In any *information transfer channel* $\mathcal{K}$ the *channel equation*

$$H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{9}$$

is valid [13]. This equation describes the mutual relations among *information entropies [(average) information amounts]* in the channel $\mathcal{K}$.
The quantities $H(X)$, $H(Y)$, $H(X|Y)$ and $H(Y|X)$ are the *input*, the *output*, the *loss* and the *noise (diturbant)* entropy.
The difference $H(X)-H(X|Y)$ or the difference $H(Y)-H(Y|X)$ defines the *transinformation* $T(X;Y)$ or the transinformation $T(Y;X)$ respectively,

$$H(X) - H(X|Y) \triangleq T(X;Y) \quad = \quad T(Y;X) \triangleq H(Y) - H(Y|X) \tag{10}$$

When the channel $\mathcal{K}$ transfers the information (entropy) $H(X)$, but now just at the value of the entropy $H(X|Y)$, $H(X) = H(X|Y)$, then, necessarily, must be valid

$$T(X;Y) = 0 \quad [= H(Y) - H(Y|X)] \tag{11}$$

- For $H(Y|X) = 0$, we have $T(X;Y) = H(Y) = 0$.
- For $H(Y|X) \neq 0$ we have $H(Y) = H(Y|X) \neq 0$

In both these two cases the channel $\mathcal{K}$ operates as the *interrupted (with the absolute noise)* and the output $H(Y)$ is without any relation to the input $H(X)$ and, also, it doesn't relate to the structure of $\mathcal{K}$. This structure is expressed by the value of the quantity $H(X|Y)$. We assume, for the simplicity, that $H(Y|X) = 0$.
From the (9)-(11) follows that the **channel $\mathcal{K}$ can't** transfer (within the same step p of its *transfer process*) such an information which describes its inner structure and, thus, it can't **transfer - observe** (copy, measure) **itself**. It is valid both for the concrete information value and for the average information value as well.

Any channel $\mathcal{K}$ can't transfer its own states considered as the input messages (within the same steps p). Such an attempt is the information analogy for the ***Auto-Reference*** *known from Logics and Computing Theory. Thus a certain 'step-aside' leading to a not zero tranfer output,* $H(Y) = H(X) - H(X|Y) > 0$, *is needed.*

3.2 Auto-Reference in Thermodynamics - Stationarity

The transfer process running in an information transfer channel $\mathcal{K}$ is possible to be comprehended (modeled or, even, constructed) as the *direct* Carnot Cycle $\mathcal{O}$ [2, 3]. The relation $\mathcal{O} \cong \mathcal{K}$ is postulated. Further, we can imagine its observing method, equivalent to its 'mirror' $\mathcal{O}' \cong \mathcal{K}'$. This *mirror* $\mathcal{O}'$ is, at this case, the direct Carnot Cycle $\mathcal{O}$ as for its structure, but functionning in the *indirect, reverse* mode [2, 3]. Let us connect them together to a *combined heat cycle* $\mathcal{O}\mathcal{O}'$ *in such a way that the mirror (the reverse cycle* $\mathcal{O}'$) is gaining the message about the structure of the direct cycle $\mathcal{O}$. This message is (carrying) the information $H(X|Y)$ about the structure of the transformation (transfer) process ($\mathcal{O} \cong \mathcal{K}$) being 'observed'. *The mirror* $\mathcal{O}' \cong \mathcal{K}'$ *is gaining this information* $H(X|Y)$ *on its disturbant 'input'* $H(Y'|X')$, *[while* $H(X') = H(Y)$ *is its input].*

The quantities ΔQ_W, ΔA and ΔQ_0 or the quantities $\Delta Q'_W$, $\Delta A'$ and $\Delta Q'_0$ respectively, define the information entropies of the information transfer realized (thermodynamically) by the *direct* Carnot Cycle or by the *reverse* Carnot Cycle (the mirror) respectively (the *combined* cycle $\mathcal{O}\mathcal{O}'$ is created),

$$H(X) = \frac{\Delta Q_W}{\mathrm{k}T_W}, \quad \text{resp.} \quad H(Y') = \frac{\Delta Q'_W}{\mathrm{k}T'_W} \tag{12}$$
$$H(Y) = \frac{\Delta A}{\mathrm{k}T_W}, \quad \text{resp.} \quad H(X') = \frac{\Delta A'}{\mathrm{k}T'_W}$$
$$H(X|Y) = \frac{\Delta Q_0}{\mathrm{k}T_W}, \quad \text{resp.} \quad H(Y'|X') = \frac{\Delta Q'_0}{\mathrm{k}T'_W}$$

Our aim is to gain the *not zero* output mechanical work ΔA^* of the combined heat cycle $\mathcal{O}\mathcal{O}'$, $\Delta A^* > 0$. We want gain not zero information $H^*(Y^*) = \dfrac{\Delta A^*}{\mathrm{k}T_W} > 0$.
To achieve this aim, for the efficiencies η_{max} and η'_{max} of the both connected cycles $\mathcal{O}$ and $\mathcal{O}'$ (with the working temperatures $T_W = T'_W$ and $T_0 = T'_0,\ T_W \geq T_0 > 0$) it must be valid that $\eta_{max} > \eta'_{max}$; we want the validity of the relation[5]

$$\Delta^* A = \Delta A - \Delta' A > 0 \quad [\Delta A' = \Delta Q'_W - \Delta Q'_0] \tag{13}$$

When $\Delta Q_0 = \Delta Q'_0$ $[\Delta Q'_0 = (1 - \eta'_{max}) \cdot \Delta Q'_W,\ \ \Delta Q_0 = (1 - \eta_{max}) \cdot \Delta Q_W]$ should be valid, then $\Delta Q'_W < \Delta Q_W$ $[\Leftarrow (\eta_{max} > \eta'_{max})]$ and thus it should be valid that

$$\Delta A^* = \Delta Q_W \cdot \eta_{max} - \Delta Q'_W \eta'_{max} = \Delta Q_W - \Delta Q'_W > 0 \tag{14}$$

Thus the output work $\Delta A^* > 0$ should be genarated by the direct change of this heat $\Delta Q_W - \Delta Q'_W$ but within a cycle $\mathcal{O}\mathcal{O}'$. For $\eta_{max} < \eta'_{max}$ the same heat $\Delta Q_W - \Delta Q'_W$ should be pumped from T_0 to T_W without any compensation by a mechanical work. Our combined machine $\mathcal{O}\mathcal{O}'$ should be the *II. Perpetuum Mobile* in both two cases.

[5] We follow the proof of physical and thus logical impossibility of the construction and functionality of the *Perpetuum Mobile* of the *II.* and, equavalently [3], of the *I.* type.

Thus $\eta_{max} = \eta'_{max}$ must be valid (the heater and the cooler are common).

The whole amount of the information entropy, within that environment in which our combined machine $\mathcal{O}\mathcal{O}'$ ($\eta_{max} = \eta'_{max}$) is running, changes by the zero value

$$H^*(Y^*) = \frac{\Delta A^*}{\mathrm{k}T_W} = H(X) \cdot \eta_{max} - H(Y') \cdot \eta'_{max} = H(X) \cdot (\eta_{max} - \eta_{max}) = 0 \qquad (15)$$

Thus, the observation of the observed process $\mathcal{O}$ by the observing reverse process $\mathcal{O}'$ with the same structure (by itself), or the Self-Observation, is impossible in a physical sense, and, consequntly, in a logical sense too (the Auto-Reference in computing.)

Nevertheless, the construction of the Auto-Reference is describable and, as such, is recognizible, *decidible* just as a construction *sui generis.* It leads, necessarily, to the requirement of the *II.* Perpetuum Mobile functionality when the requirement (13) and (14) sustains.

In the Auto-Reference case, *the whole combined machine $\mathcal{O}\mathcal{O}'$ is a system in the equilibrium status. For this status we can introduce the term (quasi)**stationary status** in which the (infinitesimal) **part of heat is circulating**. Any round of this circulation is lasting the time interval Δt; infinite, $\Delta t \rightarrow \infty$, for <u>not ideal model</u>, or, finite, $\Delta t < \infty$, when the <u>ideal model</u> is used; then the part of heat is not to be the infinitesimal . With the exception of the II. Perpetuum Mobile functionality of this combined machine, which is not possible, see (13) and (14), only the opening the system and an external activity, a **certain 'step-aside'** between the cycles $\mathcal{O}$ and $\mathcal{O}'$, moves it away (prevent it) from this status.*

Nevertheless, we want gain the information (about) $H(X|Y)$ about the structure of the observed $\mathcal{O}$ (the transfer channel $\mathcal{K}$), we want the not zero value ΔA^*, the not zero information $H^*(Y^*) = \dfrac{\Delta A^*}{\mathrm{k}T_W} > 0$.

Then, necessarily, the mirror, the reverse Carnot Cycle $\mathcal{O}'$ (the transfer channel $\mathcal{K}'$) is to be constructed with that 'step-aside' (excluding that stationarity) from the observed $\mathcal{O} \cong \mathcal{K}$. The 'step-aside' of the *observing* process $\mathcal{O}$ from the *observed* process $\mathcal{O}'$ now we mean to be realized by the difference $T_W - T'_W > 0$. Now, within this thermodynamic point of view, it is valid that $\Delta A' < \Delta A$ for $T_0 = T'_0$,

$$\Delta A' \;=\; \Delta Q'_W \cdot \left(1 - \frac{T_0}{T'_W}\right) = \Delta Q_W \cdot \frac{T'_W}{T_W} \cdot \left(1 - \frac{T_0}{T'_W}\right), \quad T'_W \triangleq T_{0^*} \qquad (16)$$

Then, for the whole information amount $\dfrac{\Delta A^*}{\mathrm{k}T_W}$ of our combined cycle it is valid that

$$\frac{\Delta A^*}{\mathrm{k}T_W} = H(Y') - H(Y') \cdot \beta^* = H(Y') \cdot \left(1 - \frac{T_{0^*}}{T_W}\right) = H(Y') \cdot \left[1 - \frac{H(X|Y)}{H(X'|Y')}\right] \qquad (17)$$

The structure $H(X|Y)$ of the observed transfer (channel, process) $\mathcal{O} \cong \mathcal{K}$ is measurable with the 'step-aside' only, created now by different temperatures ($T_W > T'_W$). The result is[6]

$$\frac{\Delta A^*}{kT_W} > 0, \quad \left[\frac{\Delta A^*}{kT_W} \cong f\,[H(X|Y)] > 0\right] \tag{18}$$

Following (13), (14) and (15) the Auto-Reference arises just when

$$T_W = T'_W \quad [\Rightarrow \; H(Y) = 0] \tag{19}$$

4. Auto-Reference in Turing Computing

Although any instruction of the Turing Machine TM describes one step of the *computing process* in this TM, it is considerable as a *description (of one step) of an information transfer process* running in a certain transfer channel $\mathcal{K}$; we postulate the relation $TM \cong \mathcal{K}$. The computing process in the TM is, also, a transfer process in a channel $\mathcal{K}$. For $\mathcal{K} \cong \mathcal{O}$ it is valid that $TM \cong \mathcal{O}$.

In each step $p>1$ of its activity, the $TM \cong \mathcal{K}$ is accepting *its own configuration from the previous step* $p-1$ *as its input*, includes its contemporary status ($s_{i_p} = s_{j_{p-1}}$) and generates its status $[s_{i_{(p+1)}}]$ and the configuration for the next step $p+1$, etc.[7] Similar is valid for the configurations (denoted now by X_p and Y_p), see further.

For each $p \geq 1$ we consider the actual instances of the stochastic quantities[8] X, Y,

$$\begin{aligned} &X \stackrel{\Delta}{=} X_p,\; Y \stackrel{\Delta}{=} Y_p;\;\; X|Y \stackrel{\Delta}{=} X_p|Y_p,\; Y|X \stackrel{\Delta}{=} Y_p|X_p;\;\; Y_p = X_{p+1} \\ &X_p|X_{p+1} \stackrel{\Delta}{=} X_p \sqsubseteq X_pX_{p+1},\;\; X_{p+1}|X_p \stackrel{\Delta}{=} X_{p+1} \sqsubseteq (X_pX_{p+1})^{-1} \end{aligned} \tag{20}$$

In any step p of the TM's activity its own configurations $(\overrightarrow{\sigma_p},\, s_p,\, \overrightarrow{\varrho_p})$ - *members of the sequence* - of the **computing process** $\overrightarrow{\kappa} \stackrel{\text{Def}}{=} [(\overrightarrow{\sigma_p},\, s_p,\, \overrightarrow{\varrho_p})]_{p=1}^{p=\cdots}$, **can be considered as follows;**

- let now the stochastic quantity X_p be realized by the chain

$$(\overrightarrow{\sigma_p},\, s_p,\, \overrightarrow{\varrho_p}) \in \mathbf{T}^* \times \mathbf{S} \times \mathbf{T}^*; \;\; p=1, \;\; \overrightarrow{\sigma_1} = \overrightarrow{\varepsilon} \;\text{ a }\; \overrightarrow{\varrho_1} = \overrightarrow{\xi}, \;\; s_p = s_0 \tag{21}$$

[6]We use, also, the output $H(Y)$ of the measured process as its reaction to the input $H(X)$. The whole change of the information entropy, within the environment in which our combined cycle is running, is at the value

$$H(X)\cdot(1-\beta) - H(X)\cdot(1-\beta') = H(X)\cdot\frac{T_0}{T'_W}\cdot\left(1-\frac{T'_W}{T_W}\right) > 0$$

[7]$(s_{i_p},\, x_{k_p},\, s_{j_p},\, y_{l_p},\, D_p)(\overrightarrow{\sigma_p},\, s_p,\, \overrightarrow{\varrho_p}) \rightarrow (\overrightarrow{\sigma_{p+1}},\, s_{i_{p+1}},\, \overrightarrow{\varrho_{p+1}})$, see further.

[8]'$\sqsubseteq$' now denotes the *substring* from the begin of the string.

- let now the stochastic quantity Y_p be realized by the chain

$$\left[\overrightarrow{\sigma_{p+1}},\ s_{p+}),\ \overrightarrow{\varrho_{p+1}}\right] \in \mathbf{T}^{*}\times\mathbf{S}\times\mathbf{T}^{*} \tag{22}$$

Then, the computing process in the $TM \cong \mathcal{K}$ is describable informationally,[9] [10]

$$\begin{aligned}
H(X) &\triangleq H(X_p)=H(\overrightarrow{\sigma_p},\ s_p,\ \overrightarrow{\varrho_p}) \qquad (23)\\
H(Y) &\triangleq H(Y_p)=H[X_{p+1}]=H\left[\overrightarrow{\sigma_{p+1}},\ s_{p+1},\ \overrightarrow{\varrho_{p+1}}\right])\\
H(X|Y) &\triangleq H(X_p|Y_p)=H[X_p|X_{p+1}],\ \ H(Y|X)\triangleq H(Y_p|X_p)=H[X_{p+1}|X_p]\\
T(X;Y) &\triangleq T(X_p;Y_p)=H(X_p)-H[X_p|X_{p+1}]\\
&= H(\overrightarrow{\sigma_p},\ s_p,\ \overrightarrow{\varrho_p})-H\left[(\overrightarrow{\sigma_p},\ s_p,\ \overrightarrow{\varrho_p})\,|\left[\overrightarrow{\sigma_{p+1}},\ s_{p+1},\ \overrightarrow{\varrho_{p+1}}\right]\right]\\
T(Y;X) &\triangleq T(Y_p;X_p)=H(X_{p+1})-H[X_{p+1}|X_p]\\
&= H\left[\overrightarrow{\sigma_{p+1}},\ s_{p+1},\ \overrightarrow{\varrho_{p+1}}\right]-H[[\overrightarrow{\sigma_{p+1}},\ s_{p+1},\ \overrightarrow{\varrho_{p+1}}]|\,(\overrightarrow{\sigma_p},\ s_p,\ \overrightarrow{\varrho_p})]
\end{aligned}$$

The Auto-Reference arises with the following description of the computing (transfer, observation) process when, e.g., for a certain $p \geq p^{*} \geq 1$,

$$\begin{aligned}
H(X_p)-H[X_p|X_{p+1}] &= H(X_{p+1})-H[X_{p+1}|X_p],\ \ p\geq 1 \qquad (24)\\
H(X_p) &= H(X_p|X_{p+1});\ \ X_p=X_p\sqsubseteq X_{p+1},\ \ X_{p+1}=\varepsilon\\
H(X_{p+1}) &= H(X_{p+1}|X_p),\ \ [H\,[X_{p+1}|X_p]=0]
\end{aligned}$$

This way 'constructs' the TM 's infinite cycle from the point of a programmer view, Self-Observation in an information point of view and a stationary status from the thermodynamics point of view.

In any case a 'step-aside' to gain something else than the zero output is required.

By the 'step-aside' of the *observing* computing process from the *observed* computing process we mean a *time delay* between those two processes or, better said, a *staging of the trace* of the observed process.

4.1 Halting Problem as Auto-Reference

Now we are considering a certain TM (the observed machine) being driven by a program $\overrightarrow{\eta}$ and working with a certain input word $\overrightarrow{\xi}$. Let this activity is described by the word $[\overrightarrow{d(TM)}]$.

Let us consider that the TM with the input word $\overrightarrow{\xi}$

- **halts**, $\boldsymbol{HALT}_{TM}$, whether the word $\overrightarrow{\xi}$ *is accepted* or *rejected*,

$$\boldsymbol{HALT}_{TM}=\{\boldsymbol{HALT}_{TM}^{\text{Accept}}\cup\boldsymbol{HALT}_{TM}^{\text{Reject}}\} \tag{25}$$

[9]The transitions are given by the η_{q_p} called by X_p; $X_p|X_{p+1}\rightarrow\eta_{q_p}$, $[\eta_{q_p}^{-1}(X_{p+1})=X_p]$.
[10]$\eta_{q_p}^{-1}(X_{p+1})=X_p$ - comparison the structures of the X_p and the X_{p+1} (in $X_p|X_{p+1}$).

- **does not halt**, $\boldsymbol{LOOP^{\infty}_{TM}}$ (the TM's infinite cycle)

Let us construct the three new Turing Machines M_1, M_2 and M_3 as follows [9][11]

- M_1 works with the input word $[\overrightarrow{d(TM)} * \overrightarrow{\xi}\,]$ in that way that

- **halts**, $\boldsymbol{HALT^{\text{Accept}}_{M_1}}$,

- **stops**, $\boldsymbol{HALT^{\text{Reject}}_{M_1}}$,

$$\boldsymbol{HALT^{\text{Accept}}_{M_1}} \Leftarrow \boldsymbol{HALT_{TM}}, \quad \boldsymbol{HALT^{\text{Reject}}_{M_1}} \Leftarrow \boldsymbol{LOOP^{\infty}_{TM}} \tag{26}$$

- M_2 **modifies the activity of the** M_1 in that way, that the input word which is being worked with is $[\overrightarrow{d(TM)} * \overrightarrow{\xi}\,]$ and

- **halts**, $\boldsymbol{HALT_{M_2}}$,

- **does not halt**, $\mathbf{LOOP}^{\infty}_{M_2}$,

$$\begin{aligned} \boldsymbol{HALT_{M_2}} &\Leftarrow \boldsymbol{LOOP^{\infty}_{TM}} \quad [\Rightarrow \boldsymbol{HALT^{\text{Reject}}_{M_1}}] \\ \mathbf{LOOP}^{\infty}_{M_2} &\Leftarrow \boldsymbol{HALT_{TM}} \quad [\Rightarrow \boldsymbol{HALT^{\text{Accept}}_{M_1}}] \end{aligned} \tag{27}$$

- M_3 **is an 'extension' of the** M_2: it doubles its own input word $[\overrightarrow{d(TM)}]$ into $[\overrightarrow{d(TM)} * \overrightarrow{d(TM)}]$ and gives it to the input (of its sub-machine) M_2 and

- **halts**, $\boldsymbol{HALT_{M_3}}$,

- **does not halt**, $\mathbf{LOOP}^{\infty}_{M_3}$,

$$\begin{aligned} \boldsymbol{HALT_{M_3}} &\equiv \boldsymbol{HALT_{M_2}} \Leftarrow \boldsymbol{LOOP^{\infty}_{TM}} \quad [\Rightarrow \boldsymbol{HALT^{\text{Rject}}_{M_1}}] \\ \mathbf{LOOP}^{\infty}_{M_3} &\equiv \mathbf{LOOP}^{\infty}_{M_2} \Leftarrow \boldsymbol{HALT_{TM}} \quad [\Rightarrow \boldsymbol{HALT^{\text{Accept}}_{M_1}}] \end{aligned} \tag{28}$$

- But, when the machine $M_3 \equiv TM$ accepts the description $\overrightarrow{d(M_3)}$, thus it is valid that $\overrightarrow{d(M_3)} \equiv \overrightarrow{d(TM)}$, then

$$\begin{gathered} [\boldsymbol{HALT_{TM}} \Leftarrow \boldsymbol{LOOP^{\infty}_{TM}}] \;\wedge\; [\boldsymbol{LOOP^{\infty}_{TM}} \Leftarrow \boldsymbol{HALT_{TM}}] \\ \equiv \\ \boldsymbol{HALT_{TM}} \Leftrightarrow \boldsymbol{LOOP^{\infty}_{TM}} \end{gathered} \tag{29}$$

This result (29) is the **contradiction**. It is the consequence of the Cantor's diagonal argument has been used carrying the Auto-Reference to the sequence of the machines $(TM,\ M_1,\ M_2,\ M_3)$, or, respectively, to the sequence

$$(\underline{TM},\ M_3 \equiv \underline{TM}) \tag{30}$$

and is **leading us to that opinion that the preposition about the decidability of the Halting Problem** (recognizing the $\boldsymbol{LOOP^{\infty}_{TM}}$ state) **is not right.**
In any given step $p \geq 1$ the machine TM is deciding about itself (it is working,

[11] Minsky's proof for the undecidability of the Halting Problem (Entscheidungsproblem type).

with the description of its own actual status), it is the channel 'transferring' its own structure, it is the Self-Observer. Thus it is in a stationary status in the thermodynamic point of view.

Within this point of view, we can envisage two identical, but reversed mutually, ideal Carnot Cycles connected together. In this sense, these two machine $\mathcal{O}$ and $\mathcal{O}'$ create the equilibrium system $\mathcal{O}\mathcal{O}'$, in which we introduce the term ***stationary status****.*

Within such a system the (infinitesimal) ***part of heat is circulating*** *through the whole combined machine $\mathcal{O}\mathcal{O}'$. This fact let be now the* ***thermodynamic model of the infinite cycle*** *being started by the Self-Observation, by the Auto-reference action (29), (30); one run is like an uninterruptible operation.*

The *recursive call* of the function $\overrightarrow{d(TM)}$ (of the machine TM) by the same function $\overrightarrow{d(TM)}$ (by the machine TM) with the same argument $\left[\overrightarrow{d(TM)} * \overrightarrow{d(TM)}\right]$ is now given. The Auto-Reference (29), (30) is then the *generative function* for the infinite sequences, nevertheless thought only, as the *consequence of the stationarity concept*,

$$
\begin{aligned}
(TM,\ TM,\ ...,\ TM,\ ...\ ) &\triangleq (TM_{\Delta t_p})_{p=1}^{\infty} \qquad (31)\\
[d(TM),\ d(TM),\ ...,\ d(TM),\ ...\ ] &\triangleq \left[[d(TM)]_{\Delta t_p}\right]_{p=1}^{\infty}\\
[\boldsymbol{HALT}^{\infty}_{\boldsymbol{TM}},\ \boldsymbol{HALT}^{\infty}_{\boldsymbol{TM}},\ ...,\ \boldsymbol{HALT}^{\infty}_{\boldsymbol{TM}},\ ...\ ] &\triangleq [(\boldsymbol{HALT}^{\infty}_{\boldsymbol{TM}})_{\Delta t_p}]_{p=1}^{\infty}\\
[\boldsymbol{LOOP}^{\infty}_{\boldsymbol{TM}},\ \boldsymbol{LOOP}^{\infty}_{\boldsymbol{TM}},\ ...,\ \boldsymbol{LOOP}^{\infty}_{\boldsymbol{TM}},\ ...\ ] &\triangleq [(\boldsymbol{LOOP}^{\infty}_{\boldsymbol{TM}})_{\Delta t_p}]_{p=1}^{\infty}
\end{aligned}
$$

Within this *time-expansion* of the (29) or (30) [which possibility follows from the (quasi)stationarity concept] the envisage of the infinite cycle in the observed machine TM arises, based on its Self-Description $\left[\overrightarrow{\xi}\right] \equiv \left[\overrightarrow{d(TM)}\right]$. But, following the Auto-Refernce construction, it 'runs' in the double-machine $(TM, M_3 \equiv TM) \cong \mathcal{O}\mathcal{O}'$.[12]

The Auto-Reference step that is to solve the Halting Problem prooves, only, its own disusability for this aim; it **creates just a certain image of what is to be possibly discovered - the infinite cycle** in the form of the infinite constant time sequences [when the time expansion (31) for $p \geq 1$ is considered].

As it is valid for any ***stationary status*** *also this one can be ceased or* ***can be excluded*** *by an external action, by the 'step aside',* ***by the staging*** *as follows.*

5. Concept for Halting Problem

We suppose that in the case of a computing process running in a TM its status $\boldsymbol{LOOP}^{\infty}_{\boldsymbol{TM}}$ (the infinite cycle) is decidable within the *Observing* Turing Machine (OTM) by using the TM's trace. By this trace, the machine OTM generates and controls the 'combined observing process' for the process in the TM.

[12]Generally, $\mathcal{O}\mathcal{O}'$ can be reversible only.

We will show and use the fact, that certain *regular sequences are generated. If they are infinite, they are, inevitably, periodical*; as such, they are *decidable languages for their infinity* [1].

We will use the alphabet of terminal symbols $\mathbf{T} = \{I,\ B\}$ and these structures:

- $(s_i,\ x_k,\ s_j,\ y_l,\ D)$ is the *instruction*
- $(\overrightarrow{\sigma},\ s_{[\cdot]},\ \overrightarrow{\rho})$ is the *configuration*
- $(\varepsilon\ [\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}]\ \varepsilon)$ is the *configuration type*

Further, we introduce the *general configuration type* $\boldsymbol{X}$;

- $\boldsymbol{X} = (\overrightarrow{\boldsymbol{\sigma}},\ s_{[\cdot]},\ \overrightarrow{\boldsymbol{\rho}}) \triangleq (\mathbf{B}\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}\mathbf{B})$.

By this general configuration type the chains, e.g. $\overrightarrow{\mathbf{BIB}}s_{[\cdot]}\overrightarrow{\mathbf{BIB}}$ are ment,[13]

$$\begin{aligned}\mathbf{B}\overrightarrow{\mathbf{IB}}s_{[\cdot]}\overrightarrow{\mathbf{BI}}\mathbf{B} &= BBBBIIIIBBBBBIIIIIBBBs_{[\cdot]}BBIIBBBIIIBBBB\\ &\in\ (\overrightarrow{\boldsymbol{\sigma}},\ s_{[\cdot]},\ \overrightarrow{\boldsymbol{\rho}})\end{aligned}$$

The computing process in the observed TM generates the grammar of a regular language of instructions and, also, of general types of configurations especially, infinite possibly, and thus cyclical. As such, they are decidable languages for their infinity. These grammars are given by the initial input $\overrightarrow{\xi}$, or, respectively, by the initial configuration $(\varepsilon,\ s_0,\ \overrightarrow{\xi})$ and, also, by the instructions η_{q_p} of the program $\overrightarrow{\eta}$, being

---

[13]We consider the basic types of chains of the terminal symbols on the input-output TM's tape,

$$\begin{aligned}\mathbf{I} &\triangleq \overrightarrow{I}; &&\overrightarrow{I} = I,\ \overrightarrow{I} = II,\ \overrightarrow{I} = III...\\ \mathbf{B} &\triangleq \overrightarrow{B}; &&\overrightarrow{B} = B,\ \overrightarrow{B} = BB,\ \overrightarrow{B} = BBB,\ ...,\ \underline{\mathbf{B} \triangleq \varepsilon}\end{aligned}$$

Further types are

$$\begin{aligned}\mathbf{IB} &\triangleq \overrightarrow{I}\overrightarrow{B}; && \overrightarrow{I}\overrightarrow{B} = IB,\ \overrightarrow{I}\overrightarrow{B} = IIB,\ \overrightarrow{I}\overrightarrow{B} = III...B\\ & && \overrightarrow{I}\overrightarrow{B} = IBB,\ \overrightarrow{I}\overrightarrow{B} = IBBB...\\ & && \overrightarrow{I}\overrightarrow{B} = III...BBB...\\ \mathbf{BI} &\triangleq \overrightarrow{B}\overrightarrow{I}; && \overrightarrow{B}\overrightarrow{I} = BI,\ \overrightarrow{B}\overrightarrow{I} = BBI,\ \overrightarrow{B}\overrightarrow{I} = BBB...I\\ & && \overrightarrow{B}\overrightarrow{I} = BII,\ \overrightarrow{B}\overrightarrow{I} = BIII...\\ & && \overrightarrow{B}\overrightarrow{I} = BBB...III...\\ \mathbf{IBI} &\triangleq \overrightarrow{I}\overrightarrow{B}\overrightarrow{I} = III...BBB...III...I\\ \mathbf{BIB} &\triangleq \overrightarrow{B}\overrightarrow{I}\overrightarrow{B} = BBB...III...BBB...B\\ \mathbf{IB}B &\triangleq \mathbf{IB}\\ \mathbf{I}B\mathbf{B} &\triangleq \mathbf{IB},\\ \mathbf{I}I\mathbf{B} &\triangleq \mathbf{IB}\\ I\mathbf{IB} &\triangleq \mathbf{IB}\\ \overrightarrow{\mathbf{IB}} &\triangleq \mathbf{IB}, && \overrightarrow{\mathbf{IB}} \triangleq \mathbf{IBIB...IBIB}\\ \overrightarrow{\mathbf{IB}}B &\triangleq \mathbf{IB}B, && \overrightarrow{\mathbf{IB}}B \triangleq \mathbf{IBIB...IB}B = \mathbf{IBIB...IBIB} = \overrightarrow{\mathbf{IB}}\end{aligned}$$

generated by the sequence of the steps p of the TM's activiy in which the TM instructions are interpreted. This sequence itself is pressed out by the configurations having been generated.
The Auto-Reference arises when, e.g., for a certain $p \geq p* \geq p^0 \geq 1$,

$$\begin{aligned} H\left(\overrightarrow{\boldsymbol{X}_p}\right) - H\left(\overrightarrow{\boldsymbol{X}_p}|\overrightarrow{\boldsymbol{X}_{p+1}}\right) &= H\left(\overrightarrow{\boldsymbol{X}_{p+1}}\right) - H\left(\overrightarrow{\boldsymbol{X}_{p+1}}|\overrightarrow{\boldsymbol{X}_p}\right) = 0 \\ \overrightarrow{\boldsymbol{X}_p} &\triangleq (\boldsymbol{X}_{p*},\ \boldsymbol{X}_{p*+1},\ \dots\ \boldsymbol{X}_p) \\ \overrightarrow{\boldsymbol{X}_{p+1}} &\triangleq (\overrightarrow{\boldsymbol{X}_p},\ \boldsymbol{X}_{p+1},\ \dots,\ \boldsymbol{X}_{2p+2-p*}) \equiv (\overrightarrow{\boldsymbol{X}_p}\overrightarrow{\boldsymbol{X}_p}) \end{aligned} \tag{32}$$

Also we can write [the similar is writable for (22), (23); also see the remark 8].

$$\begin{aligned} \overrightarrow{\boldsymbol{X}_p} - \left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}\right) &= \overrightarrow{\boldsymbol{X}_{p+1}} - \left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}^{-1}\right) = \varepsilon \\ \overrightarrow{\boldsymbol{X}_p} = \left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}\right) &= \left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}^{-1}\right) \\ H\left(\overrightarrow{\boldsymbol{X}_p}\right) = H\left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}\right) &= H\left(\overrightarrow{\boldsymbol{X}_p} \sqsubseteq \overrightarrow{\boldsymbol{X}_{p+1}}^{-1}\right) = H\left(\overrightarrow{\boldsymbol{X}_{p+1}}|\overrightarrow{\boldsymbol{X}_p}\right) = 0 \\ \left[H\left(\overrightarrow{\boldsymbol{X}_{p+1}}\right) = 0\right] &\Leftrightarrow \left[Pr\left(\overrightarrow{\boldsymbol{X}_p}\right) = Pr\left(\overrightarrow{\boldsymbol{X}_p}\overrightarrow{\boldsymbol{X}_p}\right) = Pr\left(\left[\overrightarrow{\boldsymbol{X}_p}\right]^+\right) = 1\right] \end{aligned} \tag{33}$$

where $Pr(\cdot)$ denotes *probability*.

The following equivalences of chains of the terminal symbols are considerable,

$$\begin{aligned} \overrightarrow{\mathbf{IB}}\mathbf{B} &\triangleq \mathbf{IBIB...IBIBB} = \mathbf{IBIB...IBIB} = \overrightarrow{\mathbf{IB}} \\ \overrightarrow{\mathbf{IBI}} &\triangleq \mathbf{IBIIBI...IBIIBI} = \mathbf{IBIB...IBI} \\ \overrightarrow{\mathbf{IBI}}\mathbf{I} &\triangleq \overrightarrow{\mathbf{IBI}} \\ \overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}\mathbf{B}} &\triangleq \overrightarrow{\mathbf{IB}} \\ \overrightarrow{\overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}}\mathbf{B}} &\triangleq \overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}}\mathbf{B}, \quad \overrightarrow{\overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}}\mathbf{B}} \triangleq \overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}...\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\mathbf{BI}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\mathbf{BI}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\mathbf{B}...\overrightarrow{\mathbf{IB}\overrightarrow{\mathbf{I}}}\overrightarrow{\mathbf{B}\overrightarrow{\mathbf{I}}}\mathbf{B} = \overrightarrow{\mathbf{IB}} \\ \mathbf{B}B\mathbf{I} &\triangleq \mathbf{BI} \\ \overrightarrow{\mathbf{BI}} &\triangleq \mathbf{BI} \\ \overrightarrow{\mathbf{BI}} &\triangleq \mathbf{BIBI...BIBI} \\ \overrightarrow{\mathbf{BI}}I &\triangleq \mathbf{BI}I, \quad \overrightarrow{\mathbf{BI}}I \triangleq \mathbf{BIBI...BI}I = \mathbf{BIBI...BIBI} = \overrightarrow{\mathbf{BI}} \\ \overrightarrow{\mathbf{BI}}\mathbf{B} &\triangleq \mathbf{BIBI...BIBIB} = \overrightarrow{\mathbf{BIB}} \\ \overrightarrow{\mathbf{BIB}} &\triangleq \mathbf{BIBBIB...BIBBIB} = \mathbf{BIBI...BIB} \\ \overrightarrow{\mathbf{BIBI}} &\triangleq \overrightarrow{\mathbf{BI}} \\ \overrightarrow{\mathbf{BIB}}\mathbf{B} &\triangleq \overrightarrow{\mathbf{BIB}} \\ \overrightarrow{\mathbf{B}\overrightarrow{\mathbf{IBI}}} &\triangleq \mathbf{B}\overrightarrow{\mathbf{IB}}\mathbf{IB}\overrightarrow{\mathbf{IBI}}, \quad \overrightarrow{\mathbf{B}\overrightarrow{\mathbf{IBI}}} \triangleq \mathbf{B}\overrightarrow{\mathbf{IB}}\overrightarrow{\mathbf{IB}}...\overrightarrow{\mathbf{IB}}\mathbf{IB}\overrightarrow{\mathbf{IB}}\mathbf{IB}\overrightarrow{\mathbf{IBI}}...\mathbf{B}\overrightarrow{\mathbf{IB}}\mathbf{IB}\overrightarrow{\mathbf{IBI}} = \overrightarrow{\mathbf{BI}} \end{aligned}$$

5.1 Method - the *OTM*

✠[1] Let us the Turing Machine, driven by the program $\overrightarrow{\eta}$, do a certain number $e{\cdot}P$ of instructions $\eta_{[\cdot]}$ (of the program $\overrightarrow{\eta}$) beginning from the initial configuration $(\varepsilon,\ s_0,\ \overrightarrow{\xi})$. Let, e.g., $P = 2l+1,\ \ l = |\overrightarrow{\xi}|,\ \ e \geq 1;\ e \in \mathrm{N}$ is the number of the *stage* (for each stage we write, in a programmer style, $e := e+1$)

- The step ✠[1] generates the table of nine-partite structures. Its length of $e{\cdot}P$ rows[14]

$$[p \parallel q \parallel s_i\ x_k\ s_j\ y_l\ D \parallel [\overrightarrow{\sigma}\ s_{[\cdot]}\ \overrightarrow{\rho}] \parallel C \parallel \tag{34}$$
$$\parallel (\varepsilon\, [\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}]\, \varepsilon) \parallel g \parallel \overrightarrow{\boldsymbol{\sigma}}\ s_{[\cdot]}\ \overrightarrow{\boldsymbol{\rho}} \parallel G]_{p=1}^{p=e.P}$$

where the denotation used is

- C is the number of the *configuration* $(\overrightarrow{\sigma},\ s_{[\cdot]},\ \overrightarrow{\rho})$
- g is the number of the *configuration type* $(\varepsilon\, [\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}]\, \varepsilon)$
- G is the number of the *general configuration type* G, $(\overrightarrow{\boldsymbol{\sigma}},\ s_{[\cdot]},\ \overrightarrow{\boldsymbol{\rho}})$

✠[2] In the table (✠[1]) we are seeking two successive blocks of rows limited by those rows having the identical values in the columns (the identical six-partite structures)

$$[q \parallel s_i\ x_k\ s_j\ y_l\ D \parallel (\varepsilon\, [\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}]\, \varepsilon) \parallel g \parallel \overrightarrow{\boldsymbol{\sigma}}\ s_{[\cdot]}\ \overrightarrow{\boldsymbol{\rho}} \parallel G] \tag{35}$$

Thus, we are seeking for (the sequence of) the *three rows* being identical in those columns while the last row of the first block is the first row of the second block [this second ends by the third row (identical in the six columns considered)]. The numbers of these *separating* rows, the first, the second and the third row are the numbers of steps $p_{[\cdot]}$, $p_{[\cdot\cdot]}$ and $p_{[\cdot\cdot\cdot]}$ of the observed computing process. These rows are separated by numbers

$$\Delta p_{[\cdot]} \stackrel{\triangle}{=} p_{[\cdot\cdot]} - p_{[\cdot]} - 1 \geq 0, \qquad \Delta p_{[\cdot\cdot]} \stackrel{\triangle}{=} p_{[\cdot\cdot\cdot]} - p_{[\cdot\cdot]} - 1 \geq 0 \tag{36}$$

of rows laying between them. (They can follow immediatelly, $\Delta p_{[\cdot]} = 0,\ \Delta p_{[\cdot\cdot]} = 0$.)

✠[3] If the three separating rows are not found within the given stage e (✠[2]), we start the computing process driven by the program $\overrightarrow{\eta}$ and its tracing from the beginning [$(\varepsilon,\ s_0,\ \overrightarrow{\xi})$, (✠[1])] and let it run $e{\cdot}P$ steps , where $e := e+1$.

✠[4] If those three separating rows are found within the given stage e (✠[2]) (those two blocks covering the rows $p_{[\cdot]}$, ..., $p_{[\cdot\cdot\cdot]}$ where $p_{[\cdot\cdot\cdot]} = p_{[\cdot\cdot]} + \Delta p_{[\cdot\cdot]} + 1$ and $p_{[\cdot\cdot]} = p_{[\cdot]} + \Delta p_{[\cdot]} + 1$) we are checking both the two blocks, each of them from its beginning ($p_{[\cdot]}$, or $p_{[\cdot\cdot]}$ respectively) till its end ($p_{[\cdot\cdot]}$, or $p_{[\cdot\cdot\cdot]}$ respectively), seeking the rows with the identical values within their six columns (35),[15]

$$[q \parallel s_i\ x_k\ s_j\ y_l\ D \parallel (\varepsilon\, [\boldsymbol{\sigma},\ s_{[\cdot]},\ \boldsymbol{\rho}]\, \varepsilon) \parallel g \parallel \overrightarrow{\boldsymbol{\sigma}}\ s_{[\cdot]}\ \overrightarrow{\boldsymbol{\rho}} \parallel m]$$

[14] The symbols '[' and ']' in the tables denote the range of the input (its limits) and, also, the 'operating space' for the CU_{TM} in each step.

[15] If these bolcks are identical the infinite cycle is discovered, but we continue uniformly with ✠[5].

✠[5] Two or more such **identical six-partite structures** on the successive row positions (denoted by symbols z and m) are **considered as the only one row**

$$\begin{array}{lll} 1) & z_{[\cdot]},\ ...,\ z_{[\cdot\cdot]},\ p_{[\cdot]} = z_{[\cdot]} \leq z_{[\cdot\cdot]},\ m = 1 \overset{\text{Def}}{=} m_{[\cdot]} & (37) \\ 2) & z^{\cdot}_{[\cdot]},\ ...,\ z^{\cdot}_{[\cdot\cdot]},\ z^{\cdot}_{[\cdot]} > z_{[\cdot\cdot]},\ z^{\cdot}_{[\cdot\cdot]} \geq z^{\cdot}_{[\cdot]},\ m = 2 & \\ & z^{\cdot\cdot}_{[\cdot]},\ ...,\ z^{\cdot\cdot}_{[\cdot\cdot]},\ z^{\cdot\cdot}_{[\cdot]} > z^{\cdot}_{[\cdot\cdot]},\ z^{\cdot\cdot}_{[\cdot\cdot]} \geq z^{\cdot\cdot}_{[\cdot]},\ m = 3 & \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ \cdot \ldots \cdot) & z^{\cdot\cdot\cdot\ \cdot}_{[\cdot]},\ ...,\ z^{\cdot\cdot\cdot\ \cdot}_{[\cdot\cdot]},\ p_{[\cdot\cdot]} = z^{\cdot\cdot\cdot\ \cdot}_{[\cdot\cdot]} \geq z^{\cdot\cdot\cdot\ \cdot}_{[\cdot]},\ m = \cdot \ldots \cdot \overset{\text{Def}}{=} m_{[\cdot\cdot]} & \end{array}$$

(the first of them is considered only).[16]

✠[6] We check whether, by this way, two new **identical successive blocks** of unique six-partite structures are created [the first block is between the (newly numbered) rows $m_{[\cdot]} \div m_{[\cdot\cdot]}$ and the second is (newly) between the rows $m_{[\cdot\cdot]} \div m_{[\cdot\cdot\cdot]}$].
Their lengths are $\Delta m_{[\cdot]}$ and $\Delta m_{[\cdot\cdot]}$ [for $\Delta m_{[\cdot\cdot]}$: $m_{[\cdot\cdot]} := (m_{[\cdot\cdot]})_{\text{mod}(m_{[\cdot\cdot]}-1)}$],

$$\Delta m_{[\cdot]} = m_{[\cdot\cdot]} - m_{[\cdot]} + 1 \leq \Delta p_{[\cdot]},\quad \Delta m_{[\cdot\cdot]} = m_{[\cdot\cdot\cdot]} - m_{[\cdot\cdot]} + 1 \leq \Delta p_{[\cdot\cdot]}. \tag{38}$$

If NOT - we continue with ✠[2], $p* = p_{[\cdot\cdot\cdot]}$; ✠[1] when $e{\cdot}P$ is exhausted, $e := e + 1$;
If YES **- the distances ($\Delta m_{[\cdot]}$ [−2] and $\Delta m_{[\cdot\cdot]}$ [−2]) between the marginal rows of the new blocks (✠[5]) are constant, $\Delta m_{[\cdot]} = \Delta m_{[\cdot\cdot]}$,**
[the distances are counted in the number m of the unique six-partite structures (✠[5]), the last one of the first block is the first one of the second block], $\Delta m_{[\cdot]} = \Delta m_{[\cdot\cdot]}$.

✠[7] **If the distance $\Delta p_{[\cdot]}$ of the first and the last row of the first block (✠[2]) is less or equal to the distance $\Delta p_{[\cdot\cdot]}$ of the first and last row of the second blok $\Delta p_{[\cdot]} \leq \Delta p_{[\cdot\cdot]}$, we have discovered the infinite cycle driven by the program $\overrightarrow{\eta}$** [now the distaces are counted in the number of steps p].
We continue further but within the first block and with the $\Delta m_{[\cdot]}$ only.
From each unique six-partite structures (✠[5], ✠[6]) the three columns

$$[q \parallel G \parallel m] \tag{39}$$

are now taken only, being interpreted as the rules of a regular grammar

$$S_{q*} \longrightarrow G_m S_{q_m} \tag{40}$$

$$q* \in \{0\} \dot\cup \{1,\ ...,\ |\overrightarrow{\eta}|\},\qquad q_m \in \{1,\ ...,\ |\overrightarrow{\eta}|\},\quad m = m_{[\cdot]},\ ...,\ m_{[\cdot\cdot]} - 1$$

with the set $\mathbf{S}$ of non-terminal symbols and the set $\mathbf{T}'$ of terminal symbols,

$$\mathbf{S} \overset{\text{Def}}{=} \{S_0\} \dot\cup \{S_{q_m}\}_{m=m_{[\cdot]}}^{m_{[\cdot\cdot]}-1},\quad \text{card}\ \mathbf{S} = n,\quad \mathbf{T}' \overset{\text{Def}}{=} \{G_{q_m}\}_{m=m_{[\cdot]}}^{m_{[\cdot\cdot]}-1} \tag{41}$$

having the starting nonterminal symbol $S_0 \in \mathbf{S}$.[17]

[16]The situation in the first block is described only by (37).

[17]✠[7a] If the distance $\Delta p_{[\cdot]}$ of the first and the last row of the first block (✠[5]) is greater than the distance $\Delta p_{[\cdot\cdot]}$ of the first and last row of the second blok, $\Delta p_{[\cdot]} > \Delta p_{[\cdot\cdot]}$, we have discovered the finite cycle driven by the program $\overrightarrow{\eta}$.

✠[8] For the first block of the unique six–partite structers (✠[5], ✠[6], ✠[7]) the sequence of rules of a regular grammar is being generated,

$$\begin{array}{ccl} S_0 & \longrightarrow & G_{m_{[\cdot]}} S_{S_{q m_{[\cdot]}}} \\ S_{q m_{[\cdot]}} & \longrightarrow & G_{m_{[\cdot]}+1} S_{q m_{[\cdot]}+1} \\ \ldots & \ldots & \ldots \\ S_{q m_{[\cdot\cdot]}-2} & \longrightarrow & G_{m_{[\cdot\cdot]}-1} S_{q m_{[\cdot\cdot]}-1} \\ S_{q m_{[\cdot\cdot]}-1} & \longrightarrow & \vec{\varepsilon}\; S_0, \qquad [S_{q m_{[\cdot\cdot]}-1} \longrightarrow \varepsilon\; S_{HALT},\;\; S_{HALT} \notin \mathbf{S}] \end{array} \tag{42}$$

We have described the activity of a <u>*finite automaton*</u> *which accepts the infinite regular language of the general configurations types* (of the configurations of the observed machine TM). They are the words of the infinite length,

$$\left[(\vec{\boldsymbol{\sigma}}_m,\; s_m,\; \vec{\boldsymbol{\rho}}_m)_{m=m_{[\cdot]}}^{m_{[\cdot\cdot]}-1} \right]^+ = \left[\vec{\boldsymbol{X}} \right]^+ \quad [\triangleq [\mathbf{d}(TM^*)]^+] \tag{43}$$

Yet after the <u>second round</u> of the observed TM through the infinite cycle has been finished the Pumping Lemma is usable and valid for the length L of the relevant word of this infinite language, [card $\mathbf{S} \leq L < 2 \cdot$ card $\mathbf{S}$].
We can generate the status (the signal) S_{HALT} *to halt the whole machine OTM and the TM consequently.*

We can say, shortly, that: If such the three identical bi-partite structures $[\eta,\; (\vec{\boldsymbol{\sigma}},\; s,\; \vec{\boldsymbol{\rho}})]$, following each other, exist that for their distances $\Delta p_{[\cdot]}$ (measured by the number of steps of the observed process) betwen the first and the second bi-partite structure and $\Delta p_{[\cdot\cdot]}$ between the second and the third bi-partite structure it is valid that $\Delta p_{[\cdot]} \leq \Delta p_{[\cdot\cdot]}$, the observed TM is going through the infinite cycle.

The expression (43) means that we have discovered, within the dynamical system $\mathcal{O}\mathcal{O}'$, the dynamical subsystem $\mathcal{O}^*\mathcal{O}^{*\prime}$ ($\equiv TM^*$) which is in a *limit cycle*. It means the thermodynamic equilibrium within the double cycle $\mathcal{O}^*\mathcal{O}^{*\prime}$, thus for its temperatures is valid that $T^*{}_W = T^{*\prime}{}_W$, $T^*{}_0 = T^{*\prime}{}_0$; for sequences of the general configuration types ($G_{[\cdot]}$), $\overrightarrow{\boldsymbol{X}_p}$ and $\overrightarrow{\boldsymbol{X}_{p+1}}$ from (32), (33), is valid, for certain $p \geq p* \geq p^0 \geq 1$, that

$$\begin{array}{l} H\left(\overrightarrow{\boldsymbol{X}_p}\right) - H\left(\overrightarrow{\boldsymbol{X}_p} | \overrightarrow{\boldsymbol{X}_{p+1}}\right) = 0 \;\; \text{where} \\ \overrightarrow{\boldsymbol{X}_p} = (\boldsymbol{X}_{p+1},\; \ldots,\; \boldsymbol{X}_{2p+2-p*}) = (\boldsymbol{X}_{p*},\; \ldots,\; \boldsymbol{X}_p) \end{array} \tag{44}$$

which represents the *zero change of the information* and *thermodynamic <u>entropy</u>* within the working medium of a <u>reversible</u> Carnot cycle; for the sequences $\overrightarrow{X_p}$ and $\overrightarrow{X_{p+1}}$ of the observed TM's configurations $X_{[\cdot]}$ from (20) ($TM \equiv \mathcal{O}$) is valid that

$$H(\overrightarrow{X_p}) - H(\overrightarrow{X_p} | \overrightarrow{X_{p+1}}) = H(\overrightarrow{Y_p}) > 0, \;\; \overrightarrow{X_{[\cdot]}} \triangleq X_{p_*},\; \ldots,\; X_{[\cdot]},\;\; p^* \geq 1 \tag{45}$$

which represents the *not zero output and, also, the growth of the thermodynamic and information entropy* within the whole isolated system in which that reversible Carnot Cycle is running, see [2, 3]. Generally, any Carnot Cycle is, under its construction draft, the infinte cycle, and, thus, both the relations (44) and (45) represent the *information thermodynamic criterion for the infinite cycle* existence.

Example I

$$\overrightarrow{\eta} = (\eta_1,\ \eta_2,\ \eta_3,\ \eta_4); \tag{46}$$
$$\eta_1 = (s_0,\ I,\ s_0,\ I,\ R),\qquad \eta_2 = (s_0,\ B,\ s_1,\ I,\ L)$$
$$\eta_3 = (s_1,\ I,\ s_1,\ I,\ L),\qquad \eta_4 = (s_1,\ B,\ s_0,\ B,\ P)$$

This program generates the expanding sequence

$$B[IIIII]B \text{ resp. } B[IIIIII]B \text{ resp. } B[IIIIIII...I...]B \text{ resp. } \mathbf{B}\overrightarrow{\mathbf{I}}\mathbf{B} \tag{47}$$

Let $\overrightarrow{\xi} = IIIII$. Then the regular grammar of the language of the general configuration types is

$$\begin{array}{rcl} S_0 & \longrightarrow & \overrightarrow{\boldsymbol{B}}\ s_0\ \overrightarrow{\boldsymbol{IB}}\ S_1 \\ S_1 & \longrightarrow & \overrightarrow{\boldsymbol{BI}}\ s_0\ \overrightarrow{\boldsymbol{IB}}\ S_{2\ 3\ 4\ 5} \\ S_{2\ 3\ 4\ 5} & \longrightarrow & \overrightarrow{\boldsymbol{BI}}\ s_0\ \overrightarrow{\boldsymbol{B}}\ S_6 \\ S_6 & \longrightarrow & \overrightarrow{\boldsymbol{BI}}\ s_1\ \overrightarrow{\boldsymbol{IB}}\ S_{7\ 8\ 9\ 10} \\ S_{7\ 8\ 9\ 10} & \longrightarrow & \overrightarrow{\boldsymbol{B}}\ s_1\ \overrightarrow{\boldsymbol{IB}}\ S_{11} \\ S_{11} & \longrightarrow & \overrightarrow{\boldsymbol{B}}\ s_1\ \overrightarrow{\boldsymbol{BIB}}\ S_{12} \\ S_{12} & \longrightarrow & \overrightarrow{\varepsilon}\ S_0, \qquad [S_{12} \longrightarrow \varepsilon\ S_{HALT},\quad S_{HALT} \notin \mathbf{S}] \\ \mathbf{S} & = & \{S_0\ S_1\ S_{2\ 3\ 4\ 5}\ S_6\ S_{7\ 8\ 9\ 10}\ S_{11}\ S_{12}\},\ \text{card}\ \mathbf{S} = 7 \end{array} \tag{48}$$

and (the computing process described by it) generates the infinite word $\left[\overrightarrow{\boldsymbol{X}}\right]^+$,

$$\left[\overrightarrow{\boldsymbol{B}}\ s_0\ \overrightarrow{\boldsymbol{IB}},\ \overrightarrow{\boldsymbol{BI}}\ s_0\ \overrightarrow{\boldsymbol{IB}},\ \overrightarrow{\boldsymbol{BI}}\ s_0\ \overrightarrow{\boldsymbol{B}},\ \overrightarrow{\boldsymbol{BI}}\ s_1\ \overrightarrow{\boldsymbol{IB}},\ \overrightarrow{\boldsymbol{B}}\ s_1\ \overrightarrow{\boldsymbol{IB}},\ \overrightarrow{\boldsymbol{B}}\ s_1\ \overrightarrow{\boldsymbol{BIB}}\right]^+$$
$$= \left[\overrightarrow{\boldsymbol{X}}\right]^+ \triangleq [\mathbf{d}(TM^*)]^+ \tag{49}$$

After the second round through the indicated infinite cycle the word of the length $L = 12$ is generated out and, thus, the Pumping Lemma is valid,

$$\text{card}\ \mathbf{S} \leq L < 2\cdot(\text{card}\ \mathbf{S}),\quad \text{card}\ \mathbf{S} = 7$$

The following table shows the trace of the comouting process given by this example. In this process we are indicating the values

$$p_{[\cdot]} = 1,\ \ p_{[\cdot\cdot]} = 13,\ \ p_{[\cdots]} = 27,\ \ m_{[\cdot]} = 1,\ \ m_{[\cdot\cdot]} = 7 \text{ and thus}$$
$$\Delta p_{[\cdot]} = 11,\ \ \Delta p_{[\cdot\cdot]} = 13,\ \ \Delta p_{[\cdots]} = 15,\ \ \Delta m_{[\cdot]} = \Delta m_{[\cdot\cdot]} = \Delta m_{[\cdots]} = \ \dots\ = 7$$

Tab. 5.1. Tracing and staging for Example I

p ####	q ####	Instruction	Configuration	C ####	Config. Type	g ####	Gerneral Config. Type G	m ####
1	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon B[s_0\ IIIII]B\varepsilon$	1	$\varepsilon B[s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	1
2	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon B[I\ s_0\ IIII]B\varepsilon$	2	$\varepsilon B[I\ s_0\ I]B\varepsilon$	2	$\overrightarrow{BI}\ s_0\ \overrightarrow{IB}$	2
3	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon B[II\ s_0\ III]B\varepsilon$	3	$\varepsilon B[I\ s_0\ I]B\varepsilon$	2	.....	2
4	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon B[III\ s_0\ II]B\varepsilon$	4	$\varepsilon B[I\ s_0\ I]B\varepsilon$	2	.....	2
5	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon B[IIII\ s_0\ I]B\varepsilon$	5	$\varepsilon B[I\ s_0\ I]B\varepsilon$	2	.....	2
6	2	$s_0\ B\ s_1\ I\ L$	$\varepsilon B[IIIII\ s_0\ B]\varepsilon$	6	$\varepsilon B[I\ s_0\ B]\varepsilon$	3	$\overrightarrow{BI}\ s_0\ \vec{B}$	3
7	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon B[IIII\ s_1\ II]\varepsilon$	7	$\varepsilon B[I\ s_1\ I]\varepsilon$	4	$\overrightarrow{BI}\ s_1\ \overrightarrow{IB}$	4
8	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon B[III\ s_1\ III]\varepsilon$	8	$\varepsilon B[I\ s_1\ I]\varepsilon$	4	.....	4
9	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon B[II\ s_1\ IIII]\varepsilon$	9	$\varepsilon B[I\ s_1\ I]\varepsilon$	4	.....	4
10	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon B[I\ s_1\ IIIII]\varepsilon$	10	$\varepsilon B[I\ s_1\ I]\varepsilon$	4	.....	4
11	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon B[s_1\ IIIIII]\varepsilon$	11	$\varepsilon B[s_1\ I]\varepsilon$	5	$\vec{B}\ s_1\ \overrightarrow{IB}$	5
12	3	$s_1\ B\ s_0\ B\ R$	$\varepsilon[s_1\ BIIIIII]\varepsilon$	12	$\varepsilon[s_1\ BI]\varepsilon$	6	$\vec{B}\ s_1\ \overrightarrow{BIB}$	6
13	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[B\ s_0\ IIIIII]B\varepsilon$	$1'$	$\varepsilon[B\ s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	7, 1
14	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BI\ s_0\ IIIII]B\varepsilon$	$2'$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	$\overrightarrow{BI}\ s_0\ \overrightarrow{IB}$	2
15	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BII\ s_0\ IIII]B\varepsilon$	$3'$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
16	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIII\ s_0\ III]B\varepsilon$	$4'$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
17	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIIII\ s_0\ II]B\varepsilon$	$5'$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
18	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIIIII\ s_0\ I]B\varepsilon$	$5'$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
19	2	$s_0\ B\ s_1\ I\ L$	$\varepsilon[BIIIIII\ s_0\ B]\varepsilon$	$6'$	$\varepsilon[BI\ s_0\ B]\varepsilon$	3	$\overrightarrow{BI}\ s_0\ \vec{B}$	3
20	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIIIII\ s_1\ II]\varepsilon$	$7'$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	$\overrightarrow{BI}\ s_1\ \overrightarrow{IB}$	4
21	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIIII\ s_1\ III]\varepsilon$	$8'$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
22	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIII\ s_1\ IIII]\varepsilon$	$9'$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
23	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BII\ s_1\ IIIII]\varepsilon$	$10'$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
24	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BI\ s_1\ IIIIII]\varepsilon$	$10'$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
25	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[B\ s_1\ IIIIIII]\varepsilon$	$11'$	$\varepsilon[B\ s_1\ I]\varepsilon$	5	$\vec{B}\ s_1\ \overrightarrow{IB}$	5
26	3	$s_1\ B\ s_0\ B\ R$	$\varepsilon[s_1\ BIIIIIII]\varepsilon$	$12'$	$\varepsilon[s_1\ BI]\varepsilon$	6	$\vec{B}\ s_1\ \overrightarrow{BIB}$	6
27	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[B\ s_0\ IIIIIII]B\varepsilon$	$1''$	$\varepsilon[B\ s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	7, 1
28	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BI\ s_0\ IIIIII]B\varepsilon$	$2''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	$\overrightarrow{BI}\ s_0\ \overrightarrow{IB}$	2
29	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BII\ s_0\ IIIII]B\varepsilon$	$3''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
30	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIII\ s_0\ IIII]B\varepsilon$	$4''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
31	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIIII\ s_0\ III]B\varepsilon$	$5''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
32	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIIIII\ s_0\ II]B\varepsilon$	$5''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
33	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[BIIIIII\ s_0\ I]B\varepsilon$	$5''$	$\varepsilon[BI\ s_0\ I]B\varepsilon$	2	.....	2
34	2	$s_0\ B\ s_1\ I\ L$	$\varepsilon[BIIIIIII\ s_0\ B]\varepsilon$	$6''$	$\varepsilon[BI\ s_0\ B]\varepsilon$	3	$\overrightarrow{BI}\ s_0\ \vec{B}$	3
35	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIIIIII\ s_1\ II]\varepsilon$	$7''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	$\overrightarrow{BI}\ s_1\ \overrightarrow{IB}$	4
36	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIIIII\ s_1\ III]\varepsilon$	$8''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
37	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIIII\ s_1\ IIII]\varepsilon$	$9''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
38	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BIII\ s_1\ IIIII]\varepsilon$	$10''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
39	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BII\ s_1\ IIIIII]\varepsilon$	$10''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
40	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[BI\ s_1\ IIIIIII]\varepsilon$	$10''$	$\varepsilon[BI\ s_1\ I]\varepsilon$	4	.....	4
41	3	$s_1\ I\ s_1\ I\ L$	$\varepsilon[B\ s_1\ IIIIIIII]\varepsilon$	$11''$	$\varepsilon[B\ s_1\ I]\varepsilon$	5	$\vec{B}\ s_1\ \overrightarrow{IB}$	5
42	3	$s_1\ B\ s_0\ B\ R$	$\varepsilon[s_1\ BIIIIIIII]\varepsilon$	$12''$	$\varepsilon[s_1\ BI]\varepsilon$	6	$\vec{B}\ s_1\ \overrightarrow{BIB}$	6
43	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[B\ s_0\ IIIIIIII]B\varepsilon$	$1'''$	$\varepsilon[B\ s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	7, 1
44	..	.....	....	$2'''$	....	2	.....	2
..	..	.....	....	..	....	..	.....	..
60	..	.....	....	$12'''$	....	6	.....	6
61	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[B\ s_0\ IIIIIIIII]B\varepsilon$	$1''''$	$\varepsilon[B\ s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	7, 1
62	..	.....	....	$2''''$	....	2	.....	2
..	..	.....	....	..	....	..	.....	..
80	..	.....	....	$12''''$	....	6	.....	6
81	1	$s_0\ I\ s_0\ I\ R$	$\varepsilon[B\ s_0\ IIIIIIIIII]B\varepsilon$	$1'''''$	$\varepsilon[B\ s_0\ I]B\varepsilon$	1	$\vec{B}\ s_0\ \overrightarrow{IB}$	7, 1

Example II

$$
\begin{aligned}
\overrightarrow{\eta} &= (\eta_1,\ \eta_2,\ \eta_3,\ \eta_4,\ \eta_5,\ \eta_6,\ \eta_7,\ \eta_8,\ \eta_9,\ \eta_{10},\ \eta_{11},\ \eta_{12},\ \eta_{13}) \qquad (50)\\
\eta_1 &= (s_0,\ I,\ s_1,\ B,\ R)\\
\eta_2 &= (s_1,\ I,\ s_1,\ I,\ R)\\
\eta_3 &= (s_1,\ B,\ s_2,\ B,\ R)\\
\eta_4 &= (s_2,\ I,\ s_2,\ I,\ R)\\
\eta_5 &= (s_2,\ B,\ s_3,\ B,\ L)\\
\eta_6 &= (s_3,\ I,\ s_4,\ B,\ L)\\
\eta_7 &= (s_4,\ B,\ s_H,\ B,\ R)\\
\eta_8 &= (s_4,\ I,\ s_5,\ I,\ L)\\
\eta_9 &= (s_5,\ I,\ s_5,\ I,\ L)\\
\eta_{10} &= (s_5,\ B,\ s_6,\ B,\ L)\\
\eta_{11} &= (s_6,\ I,\ s_6,\ I,\ L)\\
\eta_{12} &= (s_6,\ B,\ s_0,\ B,\ R)\\
\eta_{13} &= (s_0,\ B,\ s_0,\ B,\ L)
\end{aligned}
$$

This program is figuring the differnce $4-3$ and the input is $\overrightarrow{\xi} = IIIIBII$.
From the following table followes that during our staging the whole double-machine halts itself. In the numbers of the general configuration types the following word **G** is generated,

$$
\begin{aligned}
\mathbf{G} &= [\mathbf{G}_1\mathbf{G}_2] \qquad (51)\\
\mathbf{G}_1 &= [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12]\\
\mathbf{G}_2 &= [14\ 15\ 16\ 17\ 18\ 19\ 20\ 21]
\end{aligned}
$$

We can write the regular grammer of the (regular) language of the general configuration types being generated by the process driven by $\overrightarrow{\eta}$,

$$
\begin{aligned}
S_0 &\longrightarrow \mathbf{G}_1\ S_{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13} \qquad (52)\\
S_{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13} &\longrightarrow \mathbf{G}_2\ S_{14\ 15\ 16\ 17\ 18\ 19\ 20\ 21}
\end{aligned}
$$

This grammar has the set **S** of the non-terminal symbols,

$$
\begin{aligned}
\mathbf{S} &= \{S_0\ \ S_{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13}\ \ S_{14\ 15\ 16\ 17\ 18\ 19\ 20\ 21} \equiv S_{HALT}\}\\
\text{card}\ \mathbf{S} &= 3
\end{aligned}
$$

Tab. 5.2. Tracing and staging for Example II

p ####	q ####	**Instruction**	**Configuration**	C ####	**Config. Type**	g ####	**Gerneral Config. Type** G	m ####
1	**1**	$s_0\ I\ s_1\ B\ R$	$\varepsilon[\ s_0\ IIIBII]B\varepsilon$	1	$\varepsilon[\ s_0\ \mathbf{IBI}]\varepsilon$	1	$\overrightarrow{B}\ s_0\ \overrightarrow{IB}$	1
2	2	$s_1\ I\ s_1\ I\ R$	$\varepsilon[B\ s_1\ IIBII]B\varepsilon$	2	$\varepsilon[\mathbf{B}\ s_1\ \mathbf{IBI}]\varepsilon$	2	$\overrightarrow{B}\ s_1\ \overrightarrow{IB}$	2
3	2	$s_1\ I\ s_1\ I\ R$	$\varepsilon[BI\ s_1\ IBII]B\varepsilon$	3	$\varepsilon[\mathbf{BI}\ s_2\ \mathbf{IBI}]\varepsilon$	3	$\overrightarrow{BI}\ s_2\ \overrightarrow{IB}$	3
4	3	$s_1\ B\ s_2\ B\ R$	$\varepsilon[BII\ s_1\ BII]B\varepsilon$	4	$\varepsilon[\mathbf{BI}\ s_2\ \mathbf{BI}]\varepsilon$	4	$\overrightarrow{BI}\ s_2\ \overrightarrow{BIB}$	4
5	4	$s_2\ I\ s_2\ I\ R$	$\varepsilon[BIIB\ s_2\ II]B\varepsilon$	5	$\varepsilon[\mathbf{BIB}\ s_2\ \mathbf{II}]\varepsilon$	5	$\overrightarrow{BIB}\ s_2\ \overrightarrow{IB}$	5
6	4	$s_2\ I\ s_2\ I\ R$	$\varepsilon[BIIBI\ s_2\ I]B\varepsilon$	6	$\varepsilon[\mathbf{BIBI}\ s_3\ \mathbf{I}]\varepsilon$	6	$\overrightarrow{BI}\ s_3\ \overrightarrow{IB}$	6
7	5	$s_2\ B\ s_3\ B\ L$	$\varepsilon[BIIBII\ s_2\ B]\varepsilon$	7	$\varepsilon[\mathbf{BIBI}\ s_3\ \mathbf{B}]\varepsilon$	7	$\overrightarrow{BI}\ s_3\ \overrightarrow{B}$	7
8	6	$s_3\ I\ s_4\ B\ L$	$\varepsilon[BIIBI\ s_3\ IB]\varepsilon$	8	$\varepsilon[\mathbf{BI}\ s_3\ \mathbf{IB}]\varepsilon$	8	$\overrightarrow{BI}\ s_3\ \overrightarrow{IB}$	8
9	8	$s_4\ I\ s_5\ I\ L$	$\varepsilon[BIIB\ s_4\ IBB]\varepsilon$	9	$\varepsilon[\mathbf{BIB}\ s_4\ \mathbf{IB}]\varepsilon$	9	$\overrightarrow{BIB}\ s_4\ \overrightarrow{IB}$	9
10	10	$s_5\ B\ s_6\ B\ L$	$\varepsilon[BII\ s_5\ BIBB]\varepsilon$	10	$\varepsilon[\mathbf{BI}\ s_5\ \mathbf{BIB}]\varepsilon$	10	$\overrightarrow{BI}\ s_5\ \overrightarrow{BIB}$	10
11	11	$s_6\ I\ s_6\ I\ L$	$\varepsilon[BI\ s_6\ IBIBB]\varepsilon$	11	$\varepsilon[\mathbf{BI}\ s_6\ \mathbf{IBIB}]\varepsilon$	11	$\overrightarrow{BI}\ s_6\ \overrightarrow{IB}$	11
12	11	$s_6\ I\ s_6\ I\ L$	$\varepsilon[B\ s_6\ IIBIBB]\varepsilon$	12	$\varepsilon[\mathbf{B}\ s_6\ \mathbf{BBIB}]\varepsilon$	12	$\overrightarrow{B}\ s_6\ \overrightarrow{BIB}$	12
13	12	$s_6\ B\ s_0\ B\ R$	$\varepsilon[B\ s_6\ BIIBIBB]\varepsilon$	13	$\varepsilon[\mathbf{B}\ s_6\ \mathbf{BIBIB}]\varepsilon$	13	$\overrightarrow{B}\ s_6\ \overrightarrow{BIB}$	12
14	**1**	$s_0\ I\ s_1\ B\ R$	$\varepsilon[BB\ s_0\ IIBIB]\varepsilon$	**1′**	$\varepsilon[\mathbf{B}\ s_0\ \mathbf{IBIB}]\varepsilon$	**1**	$\overrightarrow{B}\ s_0\ \overrightarrow{IB}$	**1**
15	2	$s_1\ I\ s_1\ I\ R$	$\varepsilon[BBB\ s_1\ IBIBB]\varepsilon$	2′	$\varepsilon[\mathbf{B}\ s_1\ \mathbf{IBIB}]\varepsilon$	2′	$\overrightarrow{B}\ s_1\ \overrightarrow{IB}$	2
16	3	$s_1\ B\ s_2\ B\ R$	$\varepsilon[BBBI\ s_1\ BIBB]\varepsilon$	4′	$\varepsilon[\mathbf{BI}\ s_1\ \mathbf{BIB}]\varepsilon$	14	$\overrightarrow{BI}\ s_1\ \overrightarrow{BIB}$	3
17	4	$s_2\ I\ s_2\ I\ R$	$\varepsilon[BBBIB\ s_2\ IBB]\varepsilon$	5′	$\varepsilon[\mathbf{BIB}\ s_2\ \mathbf{IB}]\varepsilon$	5′	$\overrightarrow{BIB}\ s_2\ \overrightarrow{IB}$	5
18	5	$s_2\ B\ s_3\ B\ L$	$\varepsilon[BBBIBI\ s_2\ BB]\varepsilon$	7′	$\varepsilon[\mathbf{BIBI}\ s_2\ \mathbf{B}]\varepsilon$	15	$\overrightarrow{BI}\ s_2\ \overrightarrow{B}$	13
19	6	$s_3\ I\ s_4\ B\ L$	$\varepsilon[BBBIB\ s_3\ IBB]\varepsilon$	8′	$\varepsilon[\mathbf{BIB}\ s_3\ \mathbf{IB}]\varepsilon$	16	$\overrightarrow{BIB}\ s_3\ \overrightarrow{IB}$	14
20	7	$s_4\ B\ s_H\ B\ L$	$\varepsilon[BBBI\ s_4\ BBBB]\varepsilon$	14	$\varepsilon[\mathbf{BI}\ s_4\ \mathbf{B}]\varepsilon$	17	$\overrightarrow{BI}\ s_4\ \overrightarrow{B}$	15
21 *xx*	x *xx*	$(s_H,\ I) \notin D_{\overrightarrow{\eta}}$ *xxxxx*	$\varepsilon[BBB\ s_H\ IBBBB]\varepsilon$ *xxxxx*	**15** *xx*	$\varepsilon[\mathbf{B}\ s_H\ \mathbf{IB}]\varepsilon$ *xxxxx*	**18** *xx*	$\overrightarrow{B}\ s_H\ \overrightarrow{IB}$ *xxxxx*	**16** *xx*

Remark.
As for the Example I, after the *observed (sub)machine* has entered into the infinite cycle, which is in a finite time, and has gone through this cycle twice *it is halted from the observing machine.*
As for the Example II, the whole *staging is ended by the natural end* of the process in the *observed machine.* For the finite number of steps and for each is lasting the finitely long time, both halt states of our double-Turing Machine, which is the Turing Machine too, occurre *in a finite time.*

6. Conclusion

The *unsolvable decision problems* are of two types. At first, the problem is solvable but not with the objects and decision-counting methods we have at hand. The example is the unsolvability of the binomic equations in the real axis. But with the Complex Numbers Theory they are solvable describing the physical reality. The help is that the *imaginary axis* (the new dimension) has been introduced. The second type of the unsolvable problems is of the *principle character* where *no relevant physical reality can exist.*[18] The all *undecidable* decision problems are given mistakenly by *having an Auto-Reference embedded*, they are the *paradoxes*, which invokes the infinite cycles.[19] Nevertheless we can want have the *infinite cycle for*

[18] Which doesn't mean that a counting under their description is not performed physically.

[19] And just for this they are reducible to the Halting Problem; they are requirements for Perpetuum Mobile performance.

technology purposes, e.g. the *push-pull circuit.* Here the infinite cycle's functionality is *created intentionally* and, as such, the push-pull circuit is the example of the *recursive counting.* Here the Auto-Reference is introduced intentionally, as the figuring method, creating a sequence of wanted values. Generally, such a sequence could be *divergent* or *convergent* and the divergent case is felt as the *real example of the infinite cycle in the very sense of this term.*[20] When in the recursive counting the number of figuring steps is not given explicitly, then, the results from the successive steps must be compared. When it is set badly or it is not set, the infinite cycle is here and, by the algorithm's definition requiring resultativness, it is errorneous.
The aim of this paper was to detect the infinite cycle from its own characteristics. The envisage is that although "the mathematics is an ocean of structures and only a few of them are of any physical meaning", the *counting itself is of the physical character* and, as such, is *subjected to the physical laws*, especially, to the *II.* Principle of Thermodynamics. The infinite cycle is viewed as a certain type of an equilibrium state.[21] To await the finite-time end of such states is the paradoxical and, as such, unachievable wish. But, all cycles are representable by the Carnot Cycle used as the thermodynamic model of a cyclic information transfer [2, 3, 5].[22] From this point of view the aim to recognize any infinite cycle, to decide the Halting Problem, is solvable. The information-thermodynamic considerations were expressed in terms of the Automaton Theory, the general configuration types of the observed Turing Machine were generated and the Pumping Lemma was used. The author plans studying the *incursivness* and connecting it with his concept together.
The author believes that he has shown that problems given paradoxically, errorneously as for resultativness, can have the Auto-Reference embedded both *in* the sense of the *objective* of the problem and also *in* the sense of *the solving* the problem - the *Auto-Reference can be in the solving method while the very objective of the problem can be solvable.* The author's wish is that the following claim would be considered as the theorem for recognizing, deciding, any infinite cycle:
Due to the fact that any infinite cycle starts at a finite time and for the Control Unit of any Turing Machine is an finite-state automaton **and due to the fact that the Pumping Lemma is valid for the regular infinite and thus periodical language of the general configuration types of the observed Turing Machine, the Halting Problem is decidable.** *Q.E.D.*

[20] We can say, rather joking, that the convergent counting halts, even if it was in the infinity, and that the divergent counting doesn't halt even in the infinity, including now the constant sequence too - the model is the interrupted information transfer channel.

[21] The interesting is that the stability of an equilibrium state and of an atomic structure are similar. Without the natural radioactivity the end of atoms seems to 'be in the infinity' too.

[22] We see the *growth* of thermodynamic entropy within the *whole isolated system* in which the cycle, or information transfer, is running and we see the *constant* or *decreasing* thermodynamic entropy within its *working medium*, or within the transfer channel in the information-thermodynamic representation. (The interesting is that, the Carnot Cycle, conceptually, is the infinite cycle too.)

References

1. Ajzerman, M. A. a kol. *Logika automaty a algoritmy*; Academia: Praha, 1971.
2. Hejna, B. Thermodynamic model of Noise Information Transfer. In *AIP Conference Proceedings*, Computing Anticipatory Systems: CASYS'07 – Eighth International Conference; Dubois, D., Ed.; American Institute of Physics: Melville, New York, 2008; pp 67–75.
3. Hejna, B. Gibbs Paradox as Property of Observation, Proof of *II*. Principle of Thermodynamics. In *Computing Anticipatory Systems: CASYS'09 - Nineth International Conference*; Dubois, D. M., Ed.; 2009. Proceedings of CASYS'09 131-140, ISBN 978-0-7354-0858-6, ISSN 0094-243X, American Institute of Physics, Melville, New York, 2010.
4. Hejna, B. (2010): *Informační Termodynamika I., Rovnovážná termodynamika přenosu informace*. ISBN 978-80-7080-747-7, VŠCHT Praha, Praha.
5. Hejna, B. (2011). Information Thermodynamics, Thermodynamics - Physical Chemistry of Aqueous Systems, Juan Carlos Moreno-Pirajan (Ed.), ISBN: 978-953-307-979-0, InTech, DOI: 10.5772/21309. Available from: http://www.intechopen.com/books/thermodynamics-physical-chemistry-of-aqueous-systems/information-thermodynamics.
6. Hopcroft, J. E.; Ullman J. D. *Formálne jazyky a automaty*; Alfa: Bratislava, 1978.
7. Horák, Z.; Krupka, F. *Technická fyzika*; SNTL/ALFA: Praha, 1976.
8. Kalčík, J.; Sýkora, K. *Technická termomechanika*; Academia: Praha, 1973.
9. Manna, Z. *Matematická teorie programů*; SNTL: Praha, 1981.
10. Mareš, J. *Technická kybernetika*; ČVUT: Praha, 1992, 1997.
11. Melichar, Z. *Gramatiky a jazyky*; ČVUT: Praha, 1979, 1982.
12. Moisil, Br. C. *Algebraická teorie automatů*; ČSAV: Praha, 1964.
13. Prchal, J. *Signály a soustavy*; SNTL/ALFA: Praha, 1987.
14. Kobrinskij N. E.; Trachtěnbrot B. A. *Úvod do teorie konečných automatů*; SNTL: Praha, 1967.