On the decision problem for formulas in which all disjunctions are binary.

By

Stål O. Aanderaa

Institute of Mathematics
University of Oslo
1970 *)

*) The first draft of this paper was written when the author was guest-investigator at Rochefeller University, New York, March-May 1970.
Abstract. Let $Z_1$ be the class of closed formulas of the form $\exists a \forall y \text{Kay} \& \forall x \exists u \forall y \text{Mxuy}$ where Kay and Mxuy are conjunctions of binary disjunctions of signed atomic formulas of the form $Fa\beta$ or $\neg Fa\beta$ where $F$ is a binary predicate symbol, and $\alpha$ and $\beta$ are one of the variables $a, x, u$ and $y$. We prove in our paper that there is no recursive set which separates the non-satisfiable formulas in $Z_1$ from those satisfiable in a finite domain.

§ 1. Introduction. In order to state the result of this paper, it is convenient first to introduce some definitions.

Definition 1. For any class of formulas $X$ let $N(X)$, $I(X)$, and $F(X)$ be the subclasses of $X$ which contain all formulas in $X$ which have respectively, no model, only infinite models, finite models.

Note that $N(X)$ and $F(X)$ are r.e. (recursive enumerable) if $X$ is r.e.

Definition 2. A class of formulas $X$ is a Trachtenbrot class if $N(X)$ and $F(X)$ are recursively inseparable.

(Two disjoint sets $A_o$ and $A_1$ are recursively inseparable if there exists no recursive set $B$ such that $A_o \subseteq B$ and $A_1 \cap B = \emptyset$.)

Note that if $X$ is a Trachtenbrot class, then neither $N(X)$ nor $F(X)$ nor $I(X)$ are recursive.

Trachtenbrot (1953) proved that the class of all formulas in first order predicate calculus is a Trachtenbrot class.

We shall here deal with formulas which are in prenex normal form or which are a conjunction of formulas in prenex normal form.
Definition 3. Let $Q_1, Q_2, \ldots, Q_n$ (n = 1, 2, \ldots) be a sequence of strings of quantifiers. Then a formula $S$ is a $Q_1 & Q_2 & \ldots & Q_n$-formula iff $S$ is a closed formula in first order predicate calculus of the form $Q_1 M_1 & Q_2 M_2 & \ldots & Q_n M_n$ where $M_i$ (i = 1, 2, \ldots n) is quantifier-free and contains neither the equality sign nor function symbols. The $Q_1 & Q_2 & \ldots & Q_n$-class is the class of all $Q_1 & Q_2 & \ldots & Q_n$-formulas. If $X$ is a class of formulas then a $Q_1 & Q_2 & \ldots & Q_n \cap X$-formula is a formula which is both a $Q_1 & Q_2 & \ldots & Q_n$-formula and a formula in $X$. The $Q_1 & Q_2 & \ldots & Q_n \cap X$-class is the intersection of the classes $Q_1 & Q_2 & \ldots & Q_n$ and $X$.

Hao Wang (1962) has proved that both the $\forall \forall \forall$ class and the $\exists \exists \exists$ class are Trachtenbrot classes. But I($\exists \exists \forall \forall \forall \forall \forall$-$\forall$-class) and I($\exists \exists \exists \exists \forall \forall \forall \forall \forall$-$\exists$-class) are empty classes. Hence the classes $N(\exists \exists \forall \forall \forall \forall \forall$-$\forall$-class), $F(\exists \exists \forall \forall \forall \forall \forall$-$\forall$-class), $N(\exists \exists \exists \exists \forall \forall \forall \forall \forall$-$\exists$-class) and $F(\exists \exists \exists \exists \forall \forall \forall \forall \forall$-$\exists$-class) are all recursive. This shows that the $\exists \exists \forall \forall \forall \forall \forall$-$\forall$-classes and the $\exists \exists \exists \exists \forall \forall \forall \forall \forall$-$\exists$-classes are not Trachtenbrot classes. Hence the problem of deciding whether a prefix class is a Trachtenbrot class is solved. These problems are in fact also solved for such classes as $Q_1 & Q_2 & \ldots & Q_n$-classes.

If we also put some restrictions on the matrix, then new cases occur. Some of these cases have been solved. We may classify the formulas according to the atomic subformulas. See Dreben, Kahn, Wang 1962 and Wang 1962, and Aanderaa 1966.

Melvin R. Krom and S.Ju Maslov have studied formulas in which the matrices consist of conjunctions of binary disjunctions. See Krom 1962, 1967a, 1967b, 1968, 1970 and Maslov 1964. The aim of this paper is also to investigate such formulas. It is therefore convenient to introduce the following definition.
Definition 4. Let $A$ be a formula in first-order predicate calculus and let $A'$ be the result of deleting the quantifiers in $A$. Then $A$ is a Krom formula iff $A'$ consists of a conjunction

$$C_1 \land C_2 \land \ldots \land C_m$$

of binary disjunctions $C_i = D_{1i} \lor D_{2i}$ of signed atomic formulas $D_{1i}, D_{2i}$, $i = 1, 2, \ldots, m$. Each term $C_i$ in (2) is called a conjunct of $A'$. The class of Krom formulas is denoted by $\text{Kr}$. 

Note that to each Krom formula $A$, there corresponds a Krom formula $B$ in prenex normal form such that $\vdash A \equiv B$.

The main theorem of the first part of this paper is:

Theorem 1. The $\forall \forall \forall \land \forall \forall \forall \land \text{Kr}$-class is a Trachtenbrot class.

From theorem 1 follows immediate the following corollaries.

Corollary 1. The decision problem for the $\forall \forall \forall \land \forall \forall \forall \land \text{Kr}$-class is recursively unsolvable.

Corollary 2. The $\forall \forall \forall \land \text{Kr}$-class is a Trachtenbrot class.

Corollary 3. The $\forall \forall \forall \land \text{Kr}$-class is a Trachtenbrot class.

Corollary 4. The decision problem for the $\forall \forall \forall \land \text{Kr}$-class is recursively unsolvable.

Corollary 5. The decision problem for the $\forall \forall \forall \land \text{Kr}$-class is recursively unsolvable.

Krom 1970 has proved a weaker form of corollary 5. He proved that the decision problem for the class of Krom formulas in prenex normal form with a prefix of the form $\forall \forall \forall \ldots \forall \forall$ is recursively undecidable.
We shall also prove the following theorem in the first part of this paper:

**Theorem 2.** The classes \( \mathcal{M} \cup \mathcal{M} \cap \text{Kr} \), \( \mathcal{M} \cup \text{Kr} \) and \( \mathcal{M} \cap \text{Kr} \) are reduction classes.

It turns out, however, that the classes \( N(\mathcal{M} \cap \text{Kr-class}) \) and \( P(\mathcal{M} \cap \text{Kr-class}) \) are recursive. See § 4 in this paper.

Maslov 1964 has proved that the class \( N(\mathcal{M} \cdots \mathcal{M} \cap \text{Kr-class}) \) is recursive.

We shall give the proofs of theorems 1 and 2 in detail; and our intention is that the proofs should be elementary. We shall reduce an output problem for registermachines to the problem of deciding whether \( \mathcal{M} \cup \mathcal{M} \)-formulas are consistent or have a finite model.

We shall only define registermachines and state the result we need from the theory of registermachines. Two-registermachines are called 2-type non-writing machines in Minsky 1961. \( n \)-registermachines are called program machines in Minsky 1967, pp. 199-215, and two-register machines are studied on pp. 255-258. Registermachines are also some times called countermachines. By using an appropriate coding, 2-registermachines may be used to define recursive functions. See for instance Fischer 1966, Minsky 1962, Minsky 1967, Shepherdson 1965, or Shepherdson and Sturgis 1963.

We shall first establish a lemma about registermachines and recursively unseparable sets in § 2. We shall prove the theorems 1 and 2 in § 3. Finally, we shall state some further new results in § 4. But since these results seems to be of less importance we shall in § 4 only sketch the proofs.
§ 2. The n-register machine. An n-register machine \( R_n \) consists of \( n \) registers (or also called counters) \( T_1, T_2, \ldots, T_n \), capable of storing arbitrary large natural numbers, \( R_n \) is programmed by a numbered sequence \( I_1, I_2, \ldots, I_r \) of instructions. An instantaneous description (abbreviated ID) of \( R \) is denoted by
\[
(i, x_1, x_2, \ldots, x_n)
\]
(\( 1 \leq i \leq r \) and \( x_1, x_2, \ldots, x_n \geq 0 \)) and describes \( R_n \) ready to execute instruction \( I_i \), with registers \( T_1, T_2, \ldots, T_n \) containing \( x_1, x_2, \ldots, x_n \), respectively. The instructions are all chosen from the instruction repertoire,
\[
\{H_0, H_1, P(h), DJ(h,j) \mid h = 1, 2, \ldots, n \quad j = 1, 2, \ldots, r\}.
\]
Here

\( H_0 \) means: halt and output zero.

\( H_1 \) means: halt and output 1.

\( P(h) \) means: add 1 to the contents of register number \( h \).

Go on to next instruction.

\( DJ(h,j) \) means: If contents of register number \( h \) is not zero, decrease it by 1 and jump to instruction number \( j \). If contents of register number \( h \) is zero, go on to next instruction.

A register machine \( R \) is defined when its program
\[
(3) \quad I_1, I_2, I_3, \ldots, I_r
\]
is defined. In order to deal with computations we shall introduce relations \( \vdash_R \) and \( \vdash^*_R \) between ID's (see table 1). We shall often write \( \vdash \) and \( \vdash^* \) for \( \vdash_R \) and \( \vdash^*_R \) when no confusion results. From now on we shall deal mainly with 2-register machines.
We shall use $i, j$ with or without subscript to denote numbers of the set \{1, 2, \ldots, r\} and $\alpha$ and $\beta$ with or without subscript to denote non-negative integers. Then $\vdash (i, \alpha, \beta)$ means that $(i, \alpha, \beta)$ is the initial ID. In § 3 $\vdash (i, \alpha, \beta)$ iff $(i, \alpha, \beta) = (1, 0, 0)$.

$(i_1, \alpha_1, \beta_1) \vdash (i_2, \alpha_2, \beta_2)$ means that the ID $(i_2, \alpha_2, \beta_2)$ follows immediately from $(i_1, \alpha_1, \beta_1)$ according to the program (3). If $z = 0$ or $z = 1$ then $(i, \alpha, \beta) \vdash z$ means that $(i, \alpha, \beta)$ is a halting state with output $z$, 0 and 1 are called improper instantaneous descriptions, (proper instantaneous description are of the form $(i, \alpha, \beta)$ where $i, \alpha, \beta$ are non-negative integers and $1 \leq i \leq r$). Suppose that the relation $\vdash$ is defined, Then we define $\vdash^*$ as follows.

**Definition 5.** $b \vdash^*_R c$ means that either $b = c$ or there exist ID's $d_0, d_1, \ldots, d_n$ $(n \geq 0)$ where $d_0 = b$ and $d_n = c$ and $d_k \vdash_R d_{k+1}$ $(k = 0, 1, \ldots, n-1)$. Moreover, $b \vdash^*_R c$ iff there exists on a such that $a \vdash^*_R c$ and $\vdash_R a$. We shall say that $c$ is an immediate successor of $b$ iff $b \vdash_R c$. Moreover, $c$ is a successor of $b$ iff $b \vdash^*_R c$.

We shall in the next section give a precise definition of the relation $\vdash$ and at the same time associate a formula to each 2-register machine $R$.

§ 3. Reduction to 2-register machines.

To each 2-register machine $R$ with program (3), we shall associate a first order language $L_R$ and a formula $S_R$ in $L_R$. To each instruction $I_i$ we associate a binary predicate letter $F_i$. The intended interpretation of $F_i$ is an interpretation over the non-negative integers such that $F_i \alpha \beta$ is true iff
We shall now define the relation \( \vdash_R \) describing \( R \)'s operation on its ID's. At the same time we shall define the Krom formula of the form

\[
(4) \quad \forall a \forall y \text{Kay} \land \forall x \text{Mx} \forall y \text{Mxuy}
\]

which correspond to \( R \), by listing its binary disjunctions. Each binary disjunction \( C \) is a conjunct in Kay iff neither \( x \) nor \( u \) occur in \( C \) and \( C \) is a conjunct in Mxuy iff \( a \) does not occur in \( C \).

Then \( \vdash_R \) and (4) are defined according to the following table, which is constructed according to the numbered sequence of instruction (3).

<table>
<thead>
<tr>
<th>Case</th>
<th>If ( I_1 = )</th>
<th>then the relation ( \vdash ) is defined to satisfy</th>
<th>and the following binary disjunction is added</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( I_1 )</td>
<td>( \vdash (1,0,0) )</td>
<td>( F_1aa \lor F_2aa )</td>
</tr>
<tr>
<td>1</td>
<td>( P(1) )</td>
<td>( (i,\alpha,\beta) \vdash (i+1,\alpha+1,\beta) )</td>
<td>( \neg F_{i+1}xy \lor F_{i}uy )</td>
</tr>
<tr>
<td>2</td>
<td>( P(2) )</td>
<td>( (i,\alpha,\beta) \vdash (i+1,\alpha,\beta+1) )</td>
<td>( \neg F_{i}yx \lor F_{i+1}yu )</td>
</tr>
<tr>
<td>3</td>
<td>( D(1,j) )</td>
<td>( (i,\alpha+1,\beta) \vdash (j,\alpha,\beta) )</td>
<td>( \neg F_{i}uy \lor F_{j}xy )</td>
</tr>
<tr>
<td>4</td>
<td>( D(1,j) )</td>
<td>( (i,0,\beta) \vdash (i+1,0,\beta) )</td>
<td>( \neg F_{i}ay \lor F_{i+1}ya )</td>
</tr>
<tr>
<td>5</td>
<td>( D(2,j) )</td>
<td>( (i,\alpha,\beta+1) \vdash (j,\alpha,\beta) )</td>
<td>( \neg F_{i}yu \lor F_{j}yx )</td>
</tr>
<tr>
<td>6</td>
<td>( D(2,j) )</td>
<td>( (i,\alpha,0) \vdash (i+1,\alpha,0) )</td>
<td>( \neg F_{i}ya \lor F_{i+1}ya )</td>
</tr>
<tr>
<td>7</td>
<td>( H_0 )</td>
<td>( (i,\alpha,\beta) \vdash 0 )</td>
<td>( \neg F_{i}xy \lor \neg F_{i+1}xy )</td>
</tr>
<tr>
<td>8</td>
<td>( H_1 )</td>
<td>( (i,\alpha,\beta) \vdash 1 )</td>
<td>( \neg F_{i}xy \lor F_{i+1}xy )</td>
</tr>
</tbody>
</table>

Table 1.

Note that in case 8 in table 1, the binary disjunction is a tautology. Hence we may in this case add no binary disjunction as well.
Definition 6. \( b \vdash^* c \) means that ID's \( d_0, d_1, \ldots, d_n \) \((n \geq 0)\) exist where \( d_0 = b \), \( d_n = c \) and \( d_i \vdash d_{i+1} \) \((i = 0, \ldots, n-1)\). Moreover, \( b \vdash^* c \) iff \((1,0,0) \vdash^* c\). We shall say that \( c \) is an immediate successor of \( b \) iff \( b \vdash c \). Moreover, \( c \) is a successor of \( b \) iff \( b \vdash^* c \).

Definition 7. A finite computation (or a converging computation) is a finite sequence of ID's \( d_1, d_2, \ldots, d_m \) such that \( d_i \vdash d_{i+1} \) \((i = 1, 2, \ldots, m-1)\) and \( d_{m-1} \) is terminal and \( d_m \) is an improper ID.

An infinite computation (or a diverging computation) is an infinite sequence of ID's \( d_1, d_2, d_3, \ldots \) such that \( d_i \vdash d_{i+1} \) \((i = 1, 2, 3, \ldots)\).

We shall in § 4 consider computations where \( d_1 \) is \((1,0,0)\).

Example 1. Consider the following 2-register machine \( R_1 \) and its corresponding formulas. \( \exists a \forall y (K_1 a y \land \forall x \exists y (\exists a y \land \forall x u y m_1 x u y) \)

\[
\begin{array}{|c|c|c|}
\hline
I & \text{DJ(1,6)} & F_{ia} a a \lor F_{ia} a a, \neg F_{ia} u y \lor F_{ia} x y, \neg F_{ia} a y \lor F_{ia} a y \\
\hline
I_2 & P(1) & \neg F_{ia} x y \lor F_{ia} a y \\
\hline
I_3 & \text{DJ(2,1)} & \neg F_{ia} u y \lor F_{ia} x y, \neg F_{ia} a y \lor F_{ia} a y \\
\hline
I_4 & H_0 & \neg F_{ia} x y \lor \neg F_{ia} x y \\
\hline
I_5 & P(2) & \neg F_{ia} x y \lor F_{ia} a y \\
\hline
I_6 & H_1 & \neg F_{ia} x y \lor F_{ia} a y \\
\hline
\end{array}
\]

Hence

\( K_1 a y \) is

\((F_{ia} a a \lor F_{ia} a a) \land (\neg F_{ia} a y \lor F_{ia} a y) \land (\neg F_{ia} a y \lor F_{ia} a y)\)
and $M_{xyu}$ is

$$(\neg F_{1uy} \lor F_{6xy}) \land (\neg F_{2xy} \lor F_{5uy}) \land (\neg F_{3yu} \lor F_{1yx})$$

$$(\neg F_{4xy} \lor \neg F_{4xy}) \land (\neg F_{5yx} \lor F_{6yu}) \land (\neg F_{6xy} \lor F_{6xy})$$.

The following is the computation from empty registers in example 1: $(1,0,0), (2,0,0), (3,1,0), (4,1,0), 0$.

Consider the following examples.

<table>
<thead>
<tr>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>DJ(1,6)</td>
</tr>
<tr>
<td>$I_2$</td>
<td>P(1)</td>
</tr>
<tr>
<td>$I_3$</td>
<td>P(2)</td>
</tr>
<tr>
<td>$I_4$</td>
<td>DJ(2,1)</td>
</tr>
<tr>
<td>$I_5$</td>
<td>H_0</td>
</tr>
<tr>
<td>$I_6$</td>
<td>H_1</td>
</tr>
</tbody>
</table>

The computation according to example 2 from empty registers is:

$(1,0,0), (2,0,0), (3,1,0), (4,1,1),$

$(1,1,0), (6,0,0), 1$.

The computation from empty registers according to example 3 is infinite, and the first seven ID's are

$(1,0,0), (2,1,0), (3,2,0), (1,1,0), (2,2,0),$

$(3,3,0), (1,2,0), (2,3,0), ...$

Let $d_1, d_2, \ldots, d_m$ be a finite computation. The output of the computation from $d_1$ is then $d_m$. If a computation diverges, then the output is not defined.

Consider now computation where the initial ID is $(1,2^i,0)$ for some $i$. Then each register machine $R$ as defined above defines a partial recursive function $\psi$ such that range $\psi \subseteq \{0,1\}$.
and such that

\[(5) \quad (1, 2^i, 0) \models_R^* 0 \iff \psi(i) = 0\]

\[(6) \quad (1, 2^i, 1) \models_R^* 1 \iff \psi(i) = 1.\]

Moreover, given a partial recursive function \(\psi\) such that range \(\psi \subseteq \{0, 1\}\) there exists a register machine \(R\) such that (5) and (6) are satisfied. (See Minsky 1967 p.257 or Hopcroft and Ullman 1969, p.100 or Fischer 1966, p.377).

The following lemma is then easily proved by standard methods in recursion theory. (See Rogers 1961, p.94).

**Lemma 1.** Let \(N^+, I^+\) and \(F^+\) be the set of 2-register machines such that the output of the computation from empty registers are 0, not defined and 1, respectively. Then \(N^+\) and \(F^+\) are recursively inseparable.

**Proof.** Suppose that there exist a recursive set \(A\) such that \(F^+ \subseteq A\) and \(N^+ \cap A = \emptyset\). (We shall prove lemma 1 by obtaining a contradiction from this assumption.) Then there exists recursive function \(f\) such that range \(f = \{0, 1\}\) and such that

\[R_x \in A \implies f(x) = 0.\]

\[R_x \notin A \implies f(x) = 1.\]

Here \(R_x\) is the 2-register machine with Gödel number \(x\). Hence

\[R_x \in F^+ \implies f(x) = 0\]

\[R_x \in N^+ \implies f(x) = 1.\]

There exists a recursive function \(h\) such that

\[(1, 2^x, 0) \models_R^* 1 \iff (1, 0, 0) \models_R^* h(x) 1\]
and

\[(1,2^x,0) \vdash_R^* 0 \iff (1,0,0) \vdash_R^* 0\]

Let \( g(x) = f(h(x)) \). Then \( g \) is a recursive function and range \( g = \{0,1\} \). Then

\[(7) \quad (1,2^x,0) \vdash_R^* 1 \iff g(x) = 0\]
\[(8) \quad (1,2^x,1) \vdash_R^* 0 \iff g(x) = 1\]

Choose \( z \) such that

\[(9) \quad (1,2^x,0) \vdash_R^* 1 \iff g(x) = 1\]
\[(10) \quad (1,2^x,0) \vdash_R^* 0 \iff g(x) = 0\]

Substituting \( z \) for \( x \) in (7) and (8) we obtain a contradiction. This proves lemma 1.

**Lemma 2.** The effective mapping \( \Pi_1 \) of 2-register machines into the \( \forall \forall \land \forall \forall \cap kr \)-class of formulas defined by (4) and table 1, satisfies the following conditions.

\[(11) \quad R \in N^+ \implies \Pi_1(R) \in N(\forall \forall \land \forall \forall \cap kr)\]
\[(12) \quad R \in F^+ \implies \Pi_1(R) \in F(\forall \forall \land \forall \forall \cap kr)\]

(Here \( N^+ \) and \( F^+ \) are defined as in lemma 1.)

**Proof.** We shall first prove (11).

Suppose that \( R \in N^+ \), and suppose that \( \Pi(R) \) is consistent. We shall prove that this is a contradiction. Since \( R \in N^+ \) we have that \( \vdash_R^* 0 \) i.e. \( (1,0,0) \vdash_R^* 0 \).

The formula \( \Pi_1(R) \) is of the form

\[(4) \quad \forall a \forall y \forall x \forall u \forall y M_{xuy} \]
We can now use a either modeltheoretic argument or a syntactical argument. Since (3) has a model there exist elements in $a_0, a_1, a_2, \ldots, a_n$ such that
\[(13) \quad \forall y \, K_a y \& \forall y \, M_a a_1 y \& \forall y \, M_a a_2 y \& \ldots \& \forall y \, M_a a_n y \]
is true. Here we choose $n$ larger than the maximum of the content of the registers in the computation
\[(14) \quad d_0, d_1, \ldots, d_t, \ldots, d_{m-1}, d_m \]
where $d_0 = (1,0,0)$ and $d_m = 0$.

Hence we have that
\[(15.\beta) \quad K_{a \beta} a_0 \& M_{a \beta} a_1 a_\beta \& M_{a \beta} a_2 a_\beta \& \ldots \& M_{a \beta} a_n a_\beta \]
for $\beta = 0, 1, \ldots, n$.

We shall now prove that if $d_t = (j, \alpha, \beta)$ in (14) $(t = 0, 1, 2, \ldots, m-1)$, then $F_{j \alpha \beta} a_0$ is true. The proof is by induction on $t$. If $t = 0$, then $d_t = d_0 = (1,0,0)$, since $a_0 = a$ we have according to case 0 in table 1 that $F_{1 \alpha \beta} a_0$ is true. Suppose that $d_{t-1} = (i, \alpha, \beta)$ and $d_t = (j, \gamma, \delta)$ and that $F_{i \alpha \beta} a_\beta$ is true. We shall prove that $F_{j \alpha \beta} a_\beta$ is true. We have 6 cases according to table 1. Suppose that $I_i$ is the instruction $P(1)$ (case 1). Then $d_t = (i+1, \alpha+1, \beta)$. We shall prove that $F_{i+1 \alpha+1 \beta} a_\beta \& M_{a \alpha+1 \beta} a_\beta$ is true according to (15.\beta). According to case 1 $\neg F_{i \alpha \beta} a_\beta \vee F_{i+1 \alpha \beta} a_\beta$ is a binary disjunction (conjunct) $M_{a \alpha} a_\beta$. Hence $\neg F_{i \alpha \beta} a_\beta \vee F_{i+1 \alpha \beta} a_\beta$ is a conjunct in $M_{a \alpha} a_\beta$. Hence $\neg F_{i \alpha \beta} a_\beta \vee F_{i+1 \alpha \beta} a_\beta$ is true. Moreover, $F_{i \alpha \beta} a_\beta$ is true by induction hypothesis, since $d_t = (i, \alpha, \beta)$. Hence $F_{i+1 \alpha+1 \beta} a_\beta$ is true. This completes the induction proof in the case $d_{t-1} = (i, \alpha, \beta)$ and $I_i$ is $P(1)$. The case $I_i$ is $P(2)$ (case 2) is proved in the same way. If $I_i$ is $D(1,j)$ then we have to distinguish between the case 3 where $d_{t-1} = (i, \alpha, \beta)$ and $\alpha > 0$ and the case 4 where $\alpha = 0$. 

In case 3 we have that \( \forall \alpha \forall \beta (F_{i} a_{\alpha} a_{\beta} \lor F_{j} a_{\alpha-1} a_{\beta}) \) is a conjunct of \( M_{\alpha} a_{\alpha} a_{\beta} \). In case \( \alpha = 0 \) (case 4) we have that \( \forall \alpha \forall \beta (F_{i} a_{\alpha} a_{\beta} \lor F_{i+1} a_{\alpha} a_{\beta}) \) is a conjunct of \( K a_{0} a_{\beta} \) in (15.\( \beta \)). Otherwise the proof is as before. Hence we have in particular that if \( d_{m-1} = (j, \alpha, \beta) \) then \( F_{j} a_{\alpha} a_{\beta} \) is true, but \( I_{j} \) is \( H_{0} \). Hence we have that \( \forall \alpha \forall \beta (F_{j} a_{\alpha} a_{\beta} \lor F_{j+1} a_{\alpha} a_{\beta}) \) is true since \( \forall \alpha \forall \beta (F_{j} a_{\alpha} a_{\beta} \lor F_{j} a_{\alpha} a_{\beta}) \) is a conjunct in \( M_{\alpha} a_{\alpha+1} a_{\beta} \) which is true according to (15.\( \beta \)). Then \( F_{j} a_{\alpha} a_{\beta} \) is false, which is a contradiction. This completes the proof of (11).

Note that we can easily obtain a syntactical proof by proving that (4) implies the existential closure of (13) which in turn implies the existential closure of the conjunction

\[
(15.0) \land (15.1) \land \ldots \land (15.n).
\]

Then the argument is as before, except for replacing "true" by "provable".

To prove (12) let

\[
(16) \quad d_{0}, d_{1}, \ldots, d_{t}, \ldots, d_{m-1}, d_{m}
\]

be a computation where \( d_{0} = (1,0,0) \) and \( d_{m} = 1 \). Let \( n \) be larger than the maximal content of the registers in the computation (16). It is easy to verify that the formula \( \Pi_{1}(R) \) of the form (4) is satisfiable in an infinite domain \( \{a_{0}, a_{1}, a_{2}, \ldots\} \) where \( F_{i} a_{\alpha} a_{\beta} \) is true iff \((i, \alpha, \beta)\) occurs in (16).

To see this we choose \( a = a_{0} \) in \( \exists a \forall y K a y \) and if \( x \) in \( \forall x \exists u \forall y M x u y \) has the value \( a_{\alpha} \) then we pick the value \( a_{\alpha+1} \) for \( u \).

Since \( n \) was larger than the maximal content of the registers, we have that \( F_{i} a_{\alpha} a_{\beta} \) is false if \( \alpha \geq n+1 \) or \( \beta \geq n+1 \). Hence we have that \( F_{i} a_{\alpha} a_{\beta} = F_{i} a_{n} a_{\beta} \) and \( F_{\beta} a_{\alpha} = F_{\beta} a_{\beta n} \) if \( \alpha \geq n \). Hence (4) is satisfiable in the domain \( \{a_{0}, a_{1}, a_{2}, \ldots, a_{n}\} \), where \( F_{i} a_{\alpha} a_{\beta} \) is true iff \((i, \alpha, \beta)\) occurs in (16).
This proves (12) and the proof of lemma 2 is complete.

To prove theorem 1, suppose that theorem 1 is false. Then there exists a recursive set \( Y_1 \) which separates \( N(Z_1) \) and \( F(Z_1)^* \), i.e. \( F(Z_1) \subseteq Y_1 \) and \( N(Z_1) \cap Y_1 = \emptyset \). Let \( \Pi_1 \) be the mapping mentioned in lemma 2. Let \( S \) be the set \( \Pi_1^{-1}(Y_1) \) i.e.

\[
(17) \quad R \in S \iff \Pi_1(R) \in Y_1.
\]

Let \( N^+ \) and \( F^+ \) be as in lemma 1. Then \( N^+ \cap S = \emptyset \) since suppose that \( R \in N^+ \), then \( \Pi_1(R) \in N(Z_1) \) by (11). But \( \Pi_1(R) \in N(Z_1) \) and \( N(Z_1) \cap Y_1 = \emptyset \). Hence \( \Pi_1(R) \notin Y_1 \) and therefore \( R \notin S \) by (17). Hence \( N^+ \cap S = \emptyset \). Moreover \( F^+ \subseteq S \), since suppose \( R \in F^+ \). Then \( \Pi(R) \in F(Z_1) \) by (12). Hence \( \Pi(R) \in Y_1 \) since \( F(Z_1) \subseteq Y_1 \), hence \( R \in S \). This shows that \( F^+ \subseteq S \). But \( S \) is recursive since the set \( Y_1 \) is recursive and the mapping \( \Pi_1 \) is recursive. Hence \( S \) is a recursive set which separates \( N^+ \) and \( F^+ \), which is impossible according to lemma 1. This proves theorem 1.

To prove theorem 2, we use the following familiar fact in logic: (see H. Wang 1962)

**Lemma 3.** There is an effective partial procedure by which, given a formula in first order predicate calculus, we can test whether it has no model, a finite model, or only infinite models. The procedure terminates in the first two cases but does not terminate in the last case.

Hence, given a formula \( S \) in first order predicate calculus, we can effectively construct a 2-Register machine \( R(S) \), which gives output 0 if \( S \) has no model, and output 1 if \( S \) has a finite model, and diverges otherwise. Then by lemma 2, \( \Pi_1(R(S)) \) is consistent iff \( S \) is consistent. Hence \( \exists \forall \land \forall \forall \land Kr \) is a

\[^*\] \( Z_1 \) is the class of \( \exists \forall \land \forall \forall \land Kr \)-formulas
reduction class. This proves theorem 2.

§ 4. Further results.

We shall state some further results which may be proved by refinement of the technique used so far. Since the result seems to be of less importance than the earlier results we shall only sketch the proof. First we shall here consider other computations started on empty registers. The initial values of the registers are called input. In the construction of the formulas, input may be taken care of by adding new monadic predicate letters $G_i$ and $H_i$. Intuitively $G_i x$ means $x = i$ and $H_i x$ means $x = 2^i$. Binary disjunctions of the form $G_0 a \lor G_0 a$ and $\neg G_i x \lor G_{i+1} u$, $G_{i+1} u \lor \neg G_i x$, $\neg G_j x \lor H_i x$ and $G_j x \lor \neg H_i x$ where $j = 2^i$ will take care of the input.

As far as model theory is concerned, we shall partly follow Shoenfield with respect to notions and notations. See Shoenfield 1967 p. 14-23. Let $A$ be a structure for a first order language $L$. $|A|$ is the universe of $A$ and the elements of $|A|$ are called the individuals of $A$. Then $L(A)$ is the first order language obtained from $L$ by adding all the names of individuals of $A$. If $A$ is a closed formula in $L(A)$, let $A(A) = T$ mean that $A$ is true in $A$. This is also often expressed by $A \models A$ or $\models _A A$. Let $\Gamma^+$ be the set of atomic formulas in $L(A)$ and let $\Gamma^-$ be the set of negation of formulas in $\Gamma^+$. Following Robinson 1963 p.24, we define the **positive diagram** $D^+(A)$ of $A$ to be the set of formulas $A$ in $\Gamma^+$ such that $A(A) = T$, and the **negative diagram** $D^-(A)$ of $A$ is the set of formulas $A$ in $\Gamma^-$ such that $A(A) = T$. The **diagram** $D(A)$ of $A$ is the set $D^+(A) \cup D^-(A)$. If $\Gamma = \Gamma^+ \cup \Gamma^-$, note that $D^+(A), D^-(A)$
and $D(A)$ in the sense of Robinson 1963 p.24 correspond to $D_+=D(A), D_-(A)$ and $D(A)$, respectively, in the sense of Shoenfield 1967 p.74.

**Definition 8.**

Let $Z$ be a class of formulas where each formula in $Z$ is no more complex than

$$ (18) \quad \exists a \forall x \exists u \forall y \text{Maxuy} $$

or a conjunction of formulas of the form (18) or simpler. Here Maxuy is quantifier-free. A Büchi model for the class $Z$ is a model $\mathcal{A}$ such that $|\mathcal{A}| = \{0,1,2,\ldots\}$ and such that for each part of the form (18), we have that $\mathcal{A}(\text{Monn'm}) = T$ where $n' = n+1$, for each number $n$ and $m$.

The main theorem in this section is:

**Theorem 3.** Let $A_0$ and $A_1$ be two disjoint r.e. sets. Then there exist two sequences of Krom formulas $B_i(A_0,A_1)$ and $B'_i(A_0,A_1)$ of the forms

$$ (19) B_i = B_i(A_0,A_1) = \exists a_0 \forall x \exists u \forall y (N_1 a_0 y \& N_2 a_0 a_1 y \& N_2 a_1 x y \& \ldots \& N_2 a_j y \& F a_j ) \& \forall x (\exists \text{P}_i x \lor \text{F}_o x) $$

$$ (20) B'_i = B'_i(A_0,A_1) = \exists a_0 \exists a_1 \ldots \exists a_j \forall y (N_1 a_0 y \& N_2 a_0 a_1 y \& N_2 a_1 a_2 y \& \ldots \& N_2 a_j y \& F a_j ) \& \forall x (\exists \text{P}_i x \lor \text{F}_o x) $$

where $M_xu$ is an initial segment of an infinite conjunction $Mxu$. Moreover, $Mxu$ contains only monadic predicates. $N_1 ay$ and $N_2 xuy$ are also quantifier-free and contain only monadic and symbols dyadic predicate $\text{f}$. The sets $\{B_0,B_1,B_2\ldots\}$ and $\{B'_0,B'_1,B'_2\ldots\}$ are denoted by $Z(A_0,A_1)$ and $Z'(A_0,A_1)$ respectively. The relation between $A_0, A_1, B_i, \text{ and } B'_i$ are as follows:
i. $\vdash B_i \supset B'_i$

ii. Every model $A'$ of $B'_i$ can be extended to a model $A$ of $B_i$, such that $|A| = |A'|$.

iii. $i \in A_0 \iff B_i \in N(Z(A_0, A_1))$ (inconsistent)

iv. $i \in A_1 \iff B_i \in F(Z(A_0, A_1))$

v. The class $\{B_i | i \notin A_0\}$ is a consistent class.

vi. Let $A_4$ be a finite subset of $A_1$. Then the class of formulas $\{B_i | i \in A_4\}$ is satisfiable in a finite domain.

vii. Let $A_2$ be a r.e. set of natural numbers such that $A_2 \cap A_0 = \emptyset$. Then there exists a Büchi model $A_2$ for the class $\{B_i | i \in A_2\}$ whose diagram is r.e. Moreover, we also have that $i \in A_2 \iff A_2(B_i) = T$.

viii. Let $A_3$ be a recursive set of natural numbers such that $A_3 \cap A_0 = \emptyset$. Then there exists a Büchi model $A_3$ for the class $A_3$ such that the diagram of $A_3$ is recursive and such that $i \in A_3 \iff A_3(B_i) = T$.

ix. There exists a Krom formula $B'' = B''(A_0, A_1)$ of the form

\[
\exists x \forall y \exists u \forall v \exists w \forall xuy
\]

such that

\[
i \in A_1 \iff \vdash B'' \supset B'_i\]

and

\[
i \in A_0 \iff \vdash B'' \supset \neg B'_i .\]

Hence if $A_0$ and $A_1$ are recursively inseparable, then $B''$ is an essentially undecidable theory.

x. If $A_0$ and $A_1$ are recursively inseparable then $B''$ has no recursive Büchi model.
In theorem 3 we have written \( B_i = B_i(A_0, A_1) \), \( B'_i = B'_i(A_0, A_1) \) and \( B'' = B''(A_0, A_1) \) to emphasize that \( B_i, B'_i \) and \( B'' \) depend on \( A_0 \) and \( A_1 \). Note also that all \( B_i \) contain a fixed number of dyadic predicates, but the number of monadic predicates increases by the order of \( 2^i \) in \( B_i \). All \( B'_i \) contain a fixed number of predicate symbols.

Sketch of a proof of theorem 3.

In order to define input we use monadic predicates in \( M_{xu} \). \( F_0 \) is the only monadic predicate which occurs both in \( M_i \) and \( N_1 \). \( N_2 \) does not contain monadic predicates. \( F_0 x \) means \( x \) is an input. \( N_1 \) may contain the disjunction \( \forall F_0 y \lor F_1 y a \). In the formulas \( B'_i (i = 0, 1, 2 \ldots) \) input is defined by \( F_0 a_j \) where \( j = 2^i \). The reason why this will work is that the part \( \forall y (N_1 a_0 y \& N_2 a_0 a_1 y \& \ldots \& N_2 a_{j-1} a_j y \& F_0 a_j) \) of the formula \( B'_i \) forces \( a_0, a_1, a_2, \ldots, a_j \) to be an initial segment of a Büchi model for \( \forall a \forall y N_1 a y \& \forall x \exists u \forall y N_2 x u y \). Then theorem 3 i, ii, iii, v, vi and viii is easy to prove. In order to prove theorem 3 iv we have to modify the construction somewhat. It is easy to prove that

\[ iv'. \quad i \in A_1 \implies B_i \in F(Z(A_0, A_1)) \]

but

\[ iv''. \quad B_i \in F(Z(A_0, A_1)) \implies i \in A_1 \]

may not be true in general. Suppose that \( i \notin A_1 \). If \( i \in A_0 \) then \( B_i \notin F(Z(A_0, A_1)) \).

Hence suppose that \( i \notin A_0 \) also. If the registermachine \( R \) with input \( 2^i \) cycles (goes into a finite loop) we would have that \( B_i \in F(Z(A_0, A_1)) \). But it is easy to construct the registermachine in such a way that it never cycles. It is still difficult to prove that \( B_i \notin F(Z(A_0, A_1)) \). We can solve the problem by adding some new binary disjunctions to \( N_2 x a y \). Let \( G \) be a new binary
predicate. The new disjunctions are \( \neg Gxx \lor \neg Gxx \), \( \neg Gyx \lor Gyu \), \( \neg F_1xy \lor Gxu \), \( \neg F_1yx \lor Gxu \), where \( i = 1, 2, \ldots, r \). In order to prove theorem 3 iv, we use the fact that the formula

\[
\forall x \forall y ((\neg Gxx \lor \neg Gxx) \land (\neg Gyx \lor Gyu) \land Gxu)
\]

is consistent but that it has no finite models.

In order to prove theorem 3 ix and x we shall construct the formula (21) using the same binary predicate letters as in \( N_1ay \) and \( N_2xuy \) in (19) and (20).

Let \( A \) be a set of natural numbers, such that \( A \cap A_0 = \emptyset \). Then we may define a structure \( \mathcal{A} \) satisfying all formulas \( B_i \) and \( B'_i \) where \( i \in A \), and where \( F_j \alpha \beta \) is true

\[ \iff (\exists i)(i \in A \land (1, 2^i, 0) \models (j, \alpha, \beta)) \] .

The intended model \( \mathcal{C} \) for (21) is such that

\[ F_j \alpha \beta \text{ is true} \iff (\exists i)(\exists \alpha_1)(\exists \beta_1)((j, \alpha, \beta) \models (i, \alpha_1, \beta_1)) \]

and \( I_i \) is \( H_1 \).

Hence in case 1 when \( I_i \) is \( H_1 \) we add \( F_1 xy \lor F_1 xy \) and for each binary disjunction in table 1 of the form \( \forall P \lor Q \) where \( P \) and \( Q \) are atomic formulas, we also add \( P \lor \forall Q \). In this way we obtain \( P \equiv Q \), which are what we want, since if

\[ (j, \alpha_1, \beta_1) \models (j_2, \alpha_2, \beta_2) \] then

\[ (\exists i)(\exists \alpha_3)(\exists \beta_3)((j_1, \alpha_1, \beta_1) \models (i, \alpha_3, \beta_3) \text{ and } I_i \text{ is } H_1) \]

\[ (\exists i)(\exists \alpha_3)(\exists \beta_3)((j_2, \alpha_2, \beta_2) \models (i, \alpha_3, \beta_3) \text{ and } I_i \text{ is } H_1) \]

We also add \( F_{o} y \lor \forall F_{1} ya \) to \( N_1 ay \) to obtain \( N'_1 ay \). These are the main steps in the proof of theorem 3 ix. Suppose that \( A_0 \) and \( A_1 \) are recursively inseparable. Then the set \( \{ \alpha \mid F_0 \alpha \text{ is true} \} \) is not a recursive set in a \( B \)uchi model for \( B'' \). This proves theorem 3 x. This completes the outline of the proof of theorem 3.
From theorem 3 we obtain the following corollaries.

**Corollary 6.** The formula
\[ \exists x \forall y \forall z \exists u (H_{x} \land H_{y} \land H_{z} \land H_{u}) \]
where \( N'_{1} \) and \( N'_{2} \) are as in (21) and \( H \) is a new predicate letter, has no recursive model.

**Definition 9.** A class \( Z \) of formulas in first order predicate calculus is called a conservative reduction class iff there exists an effective procedure by which, when an arbitrary formula \( S \) in first order predicate logic is given, a corresponding formula \( S_{Z} \) of the class \( Z \) can be found such that

i. \( S \) is satisfiable \( \iff \) \( S_{Z} \) is satisfiable

ii. \( S \) is satisfiable in a finite domain \( \iff \) \( S_{Z} \) is satisfiable in a finite domain.

From lemma 3 and theorem 3, we obtain the following improvements of theorem 2.

**Corollary 7.** The classes \( N(V \land \forall x \exists y \forall z \exists u) \), \( N(V \land \forall x \exists y \forall z \exists u) \) and \( N(V \land \forall x \exists y \forall z \exists u) \) are conservative reduction classes.

**Theorem 4.** The classes \( N(V \land \forall x \exists y \forall z \exists u) \), \( N(V \land \forall x \exists y \forall z \exists u) \) and \( N(V \land \forall x \exists y \forall z \exists u) \) are all non-empty and recursive.

In order to prove theorem 4 we first prove that we reduce the case to consider formulas

(22) \[ \forall x \exists u \forall y \exists x \exists u \]
which contain atomic parts of the forms
(23) \( F_{xy}, F_{yx}, F_{uy}, F_{yu} \) and 

(24) \( F_{xx}, F_{yy}, F_{uu} \),

only, where \( F \) is a binary predicate symbol.

We shall here first consider the set \( N(VV\forall \lor K^r) \). If (22) is satisfiable, then (22) has models whose domain is the integers \( \{\ldots -1, 0, 1, 2, \ldots \} = Z \) and such that \( M_\alpha(a+1)\beta \) is true for every pair of integers \( \alpha, \beta \).

Let \( M'xay \) be the formula obtained by deleting all disjunctions which contain atomic parts of the form (24). We shall now consider sets of pairs of integers \( (\alpha_1, \beta_1) \) satisfying

(25) \( (\forall \alpha)(\forall \beta)M'(a+1)\beta \supset (G_1\alpha_0 \beta_0 \supset G_2(\alpha_0+\alpha_1)(\beta_0+\beta_1)) \)

where \( G_{xy} \) is \( F_{xy} \), or \( \neg F_{xy} \) or \( F_{yx} \) or \( \neg F_{yx} \) for some binary predicate \( F_{xy} \). As in the theory of bounded languages in Ginsburg 1966 we regard \( Z^n \) instead of \( N^n \) where \( N = \{0, 1, 2, \ldots \} \), to be a subset of the space \( \mathcal{P}^n \). Moreover, given subsets \( C \) and \( P \) of \( Z^n \), let \( L(C; P) \) denote the set of all elements in \( Z^n \) which can be represented by the form

\[ c_0 + x_1 + x_2 + \ldots + x_m \]

for some \( c_0 \) in \( C \) and some (possibly empty) sequence \( x_1, \ldots, x_m \) of elements of \( P \). \( C \) is called the set of constants and \( P \) the set of periods of \( L(C; P) \).

\( L \subseteq Z^n \) is said to be a linear set if \( C \) consists of exactly one element, say \( C = \{c\} \), and \( P \) is finite, say \( \{p_1, \ldots, p_n\} \). A subset of \( Z^n \) is said to be semilinear if it is a finite union of linear sets.

Note that we consider here subsets of \( Z^n \) instead of \( N^n \) as used in the theory for bounded languages.
We can now prove the following lemma.

**Lemma 4.** The set of pairs \((\alpha, \beta)\) satisfying (25) is a semilinear set. There exists an effective procedure by which a representation of the semilinear sets can be obtained from \(M'xuy\).

Next we prove the following lemma:

**Lemma 5.** The set of integers \(\alpha\) satisfying

(26) \((\forall \alpha)(\forall \beta)\) \(M\alpha(\alpha + 1)\beta \supset (H_1\alpha_0 \supset H_2(\alpha_0 + \alpha))\)

is a semilinear set. Here \(H_1x\) is \(Fxx\) or \(\neg Fxx\) for some binary predicate \(F\). There exists an effective procedure by which a representation of the semilinear sets can be obtained from \(Mxuy\).

The lemmas 4 and 5 are proved almost in the same way as Parikh's theorem which says that if \(L \subseteq a^*b^*\) is a contextfree language then the set \(\{(i,j) \mid a_i b_j \in L\}\) is semilinear. See Ginsburg 1966, pp.146-149.

The theorem 4 now follows easily.
References.


