GRAMMAR LOGICS

Luis FARIÑAS DEL CERRO & Martti PENTTONEN

1. *Introduction*

In this paper we present a simple method to define modal logics for formal grammars. Given a formal grammar, we associate with each rule an axiom of a modal logic. By this construction, testing whether a word is generated by a formal grammar is equivalent with proving a theorem in the logic. First constructions produce multimodal logics possessing several agents [Ko], but the technique can be applied to construct also more classical logics with only one modal operator.

Our approach is suggested by a method used in logic programming, where the analysis or generation of a sentence is transformed to theorem proving [Co, PW]. Other related work is done by Wolper [Wo], who defines an extension of the linear temporal logic of programs to include regular expressions.

2. *Minimal grammar logics*

We begin with logics that are rather artificial but are closely related with formal grammars.

The expressions of the minimal grammar logic are constructed of the following symboles:

VAR	set of *propositional variables*,
MOD	*modal alphabet*, interpreted as a set of agents,
[]	*modal operator constructors*, and
$\rightarrow$	*conditional* connective,

where VAR and MOD are nonempty, disjoint sets. The set FOR of the *formulas* of the minimal grammar logic is defined as follows:

VAR $\subseteq$ FOR,
if A, B $\in$ FOR then A $\rightarrow$ B $\in$ FOR,
if A $\in$ FOR and a $\in$ MOD then [a]A $\in$ FOR.

The constructions [a], where $a \in MOD$, are called *modal operators*, and can be considered as agents. Furthermore, for arbitrary $n \geq 0$, $a_i \in MOD$, $[a_1 a_2 ... a_n]$ is shorthand for $[a_1][a_2]...[a_n]$, and we call it a *modal word*.

A *type 0 grammar* is a quadruple G=(V,T,P,S), where V and T are disjoint finite alphabets, $S \in V$, and P is a finite set of pairs (u,v), $u \in (V \cup T)^* V(V \cup T)^*$, $v \in (V \cup T)^*$. A word x over $V \cup T$ *derives directly* to another word y, $x \Rightarrow y$, if x and y can be decomposed to $x = x'uc''$, $y = x'vx''$, where $(u,v) \in P$. The relation *derives*, $x \Rightarrow^* y$, is the reflexive, transitive closure of $\Rightarrow$. The non-reflexive, transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^+$. The *language generated* by G is the set $L(G) = \{w \mid S \Rightarrow^* w, w \in T^*\}$.

It is well known (see [Sa]) that the *membership problem* of type 0 grammars, i.e. the question whether a given word is generated by a given grammar, is undecidable. We use this result to construct undecidable propositional modal logics.

Given a type 0 grammar G=(V,T,P,S), we associate to it a *minimal grammar logic* L_G that has the *axiom schemes*

A1. $[u]A \rightarrow [v]A$ for all $(u,v) \in P$, where A does not contain $\rightarrow$, and the *inference rules*

R1. transitivity $\dfrac{A \rightarrow B \quad B \rightarrow C}{A \rightarrow C}$

R2. conditional necessitation $\dfrac{A \rightarrow B}{[a]A \rightarrow [a]B}$

In axioms and rules, A, B, and C range over formulas, and "a" ranges over the modal alphabet.

Under the epistemic interpretation, the axiom A1 defines an interdependence between the knowledges of agents. For example $[a][b]A \rightarrow [c]A$ can be read as follows: "If a knows that b knows A, then c knows a".

The notions of a proof and a theorem are defined as usual. A *proof* of a formula A from a set S of formulas is a finite sequence of formulas, each of which is either an axiom or a formula in S or is obtained by a rule of inference from earlier elements of the sequence. This is denoted by $S \vdash_G A$. A formula is a *theorem*, $\vdash_G S$, if it has a proof from the empty set.

Theorem 2.1. *It is undecidable whether a formula of a minimal grammar logic is a theorem.*

Proof. Consider a type 0 grammar G=(V,T,P,S). Because the membership problem of type 0 grammars is undecidable, it is sufficient to prove that $\vdash_G [u]q \rightarrow [v]q$ iff $u \Rightarrow^+_G v$, where q is a propositional variable.

We shall show first that for any theorems of the form $[u]q \rightarrow [v]q$ there is a derivation $u \Rightarrow^+ v$. The axioms of L_G correspond to derivations of length 1. Assume that $[u]q \rightarrow [w]q$ is proved from $[u]q \rightarrow [v]q$ and $[v]q \rightarrow [w]q$ by R1. By induction hypothesis, $u \Rightarrow^+ v$ and $v \Rightarrow^+ w$, and consequently $u \Rightarrow^+ w$. If $[a][u]q \rightarrow [a][v]q$ is proved by R2 from the theorem $[u]q \rightarrow [v]q$. The assertion $au \Rightarrow^+ av$ follows directly from the induction hypothesis $u \Rightarrow^+ v$.

For the inverse, consider a derivation $u \Rightarrow^+ vxw \Rightarrow vyw$, where $(x,y) \in P$. By induction hypothesis, $[u]q \rightarrow [v][x][w]q$ is a theorem. As $[x][w]q \rightarrow [y][w]q$ is an axiom, by repeated application of R2 we see that $[v][x][w]q \rightarrow [v][x][w]q$ is a theorem. Now $[u]q \rightarrow [v][y][w]q$ is proved by an application of R1.

3. *Grammar logics and Thue logics*

We shall now prove a similar theorem for more standard propositional modal logics, which include the classical proposional logic as a sublogic. We call these logics grammar logics. We study also a special case, where the rules of the grammar are symmetric. These logics are called Thue logics.

The *alphabet* of a grammar logic contains the propositional constant *false* $\perp$ in addition to the symbols of section 2. $\perp$ and propositional variables are *formulas*. If A and B are formulas and $a \in$ MOD, then $A \rightarrow B$, and $[a]A$ are formulas. The connectives $\neg$ (*negation*), $\vee$ (*disjunction*), $\wedge$ (*conjunction*), and $\leftrightarrow$ (*biconditional*) are introduced as usual: $\neg A = A \rightarrow \perp$, $A \vee B = \neg A \vee B$, $A \wedge B = \neg(A \rightarrow \neg B)$, and $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$. The modal operator $<a>$ is defined by $<a>A = \neg [a] \neg A$.

Consider a type zero grammar G=(V,T,P,S), and denote MOD=V ∪ T. The *grammar logic* L_G is axiomatized as follows. It has the *axiom schemes*

A1. $A \to (B \to A)$, $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$,
$((A \to \bot) \to \bot) \to A$,
A2. $[a](A \to B) \to ([a]A \to [a]B)$ for all $a \in MOD$,
A3. $[u][w]q \to [v][w]q$ for $(u,v) \in P$, $w \in MOD^*$, and $q \in VAR$,
and the *inference rules*

R1. Modus Ponens $\dfrac{A,\ A \to B}{B}$

R2. Necessitation $\dfrac{A}{[a]A}$

The proof and the theorem are defined as in section 2. We also have the undecidability theorem, but now the proof is less trivial.

Theorem 3.1. *It is undecidable whether a formula of a grammar logic is a theorem.*

Proof. Let $G=(V,T,P,S)$ be a type 0 grammar and consider the grammar logic L_G associated with it. We shall show that $S \Rightarrow^*_G w$ iff $\vdash_G [S]q \to [w]q$, where q is a propositional variable.

Assume first that $S \Rightarrow^* w$. If the length of the derivation is 0, then $w=S$ and $\vdash [S]A \to [w]A$ follows from A1 by R1. Otherwise assume that $S \Rightarrow^* w'uw'' \Rightarrow w'vw''=w$. By induction hypothesis we have $\vdash [S]q \to [w'][u][w'']q$. Because $[u][w'']q \to [v][w'']q$ is an axiom, by repeated application of R2, A2, and R1, we see that $\vdash [w'][u][w'']q \to [w'][v][w'']q$. The assertion $\vdash [S]q \to [w]q$ now follows by A1 and R1.

We shall now prove that if $\vdash_G [S]q \to [w]q$ then $S \Rightarrow^*_G w$. We do this by defining the concept of G-truth – if a formula is G-true then it corresponds to a derivation. We shall show that all theorems are G-true, because the axioms are G-true and the inference rules preserve G-truth.

By the axiomatization of L_G, we can assume that the formulas do not contain other connective than $\to$. The *context* c(p,A) of an occurrence p of a variable in a formula A is defined as follows:

$c(p,p)=$ empty,
$c(p,A)=c(p,B)$ (or $c(p,C)$) if $A=B \to C$ and p occurs in B (resp. in C),
$c(p,A)=ac(p,B)$ if $A=[a]B$.

Intuitively, c(p,A) is the sequence of modalities governing p in A. A subformula $Mp \rightarrow Nq$, where M and N are modal words and p and q are variables, is called a *pair*, and occurrences p and q are *paired*. If an occurrence of a variable is not paired, it is *single*. A pair $Mp \rightarrow Nq$ is *strict* if $p = q$ and there is no single occurrence r of the same variable such that $c(r,A) = c(q,A)$ or $c(r,A) = c(q,A)$. The G-*truth* of a subformula in a formula A is defined recursively:

$\perp$ is G-false,
variables not occurring in a strict pair are G-false,
a pair $Mp \rightarrow Nq$ is G-true iff it is strict and $c(p,A) \Rightarrow^{*}_{G} c(q,A)$,
a non-pair subformula $B \rightarrow C$ is G-true iff B is G-false or C is G-true,
[a]B, containing a connective, is G-true iff B is G-true.

We say that a strict pair gets its G-value by *derivation*, other subformulas get their values by *definition* or by *computation* (two last lines). A formula A is G-*sure* iff for all modal words u, [u]A is G-true in [u]A.

We shall now prove by induction on the length of the proof that all theorems of L_G are G-sure.

A1. $A \rightarrow (B \rightarrow A)$, $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$, $((A \rightarrow \perp) \rightarrow \perp) \rightarrow A$. All of these are proved in the same way, we choose the second as an example. The first A is not within a pair because the right hand side is not of the form Mp with a modal word M and a variable p. Hence $A \rightarrow B$ and $A \rightarrow C$ are not strict pairs. If $(B \rightarrow C)$ is not a strict pair, then all of A, B, and C get their values uniformly and the whole axiom gets the value G-true by a propositional computation. If $(B \rightarrow C)$ is a strict pair, the latter B and C are G-false and $(A \rightarrow B) \rightarrow (A \rightarrow C)$ becomes G-true, and hence the axiom is G-true. In order to prove the G-sureness, we have to establish that the addition of a modal word in front of the axiom does not violate the G-truth. Indeed, a modal prefix can change the G-truth of some strict pairs, but as it was seen above, the G-truth of the whole axiom does not depend on the G-truth of the strict pairs.

A2. $[a](A \rightarrow B) \rightarrow ([a]A \rightarrow [a]B)$. If A and B do not form a strict pair, the axiom is G-true by a classical propositional calculation. If $A \rightarrow B$ is a strict pair, also $[a]A \rightarrow [a]B$ is, and their G-truth depends on the same derivation. Hence A2 is G-true. G-sureness is seen as above.

A3. $[u][w]q \rightarrow [v][w]q$. The axiom is a strict pair which corresponds to a derivation of one step, and consequently is G-true. G-sureness is obvious, because arbitrary context can be added.

Modus Ponens. Assume that A and $A \rightarrow B$ are G-sure. As A is G-sure, it gets its value by a computation and therefore it is G-sure in $A \to B$, too. Therefore also B must be G-sure.

Necessitation preserves G-sureness, because the definition of the G-sureness requires G-truth in all contexts.

We have proved that all theorems are G-sure. Hence $\vdash_G [S]q \to [w]q$ implies the G-sureness of $[S]q \to [w]q$, i.e. the existence of a derivation $S \Rightarrow_G^* w$, which was to be proved.

The special form of the axiom A3 does not look nice in comparison with the other axioms. It remains open whether A3 can be replaced by the scheme $[u]A \to [v]A$. However, if the rules of the grammar are symmetric, i.e. $(u.v) \in P$ implies $(v,u) \in P$, then we can prove the undecidability with such a scheme. Indeed, if the rules of the grammar are symmetric, we can have axioms of the form $[u]A \leftrightarrow [v]A$. We have come to what we call Thue logics.

A *Thue system* is a pair $T=(V,P)$, where V is a finite alphabet and P is a finite set of pairs of words over V, i.e. $P \subseteq V^* \times V^*$. Define the *congruence* $\equiv_T \subseteq V^* \times V^*$ as follows: $x \equiv_T y$ iff

(i) $x=y$,
(ii) $x=x'ux''$, $y=x'vx''$, where $(u,v) \in P$ or $(v,u) \in P$,
(ii) there is a word z such that $x \equiv z$ and $z \equiv y$.

The theorem 3.1 holds for Thue systems with the modification A3′. $[u]A \leftrightarrow [v]A$ for all $(u,v) \in P$.
The concepts of T-truth and T-sureness are defined as G-truth and G-sureness, but now a strict pairs $Mp \to Nq$ and $Mp \leftrightarrow Nq$ are T-true in a formula A iff $c(p,A) \equiv_T c(q,A)$. It is sufficient to check that A3′ is T-valid in the sense of section 3. This is true because for any paired variables p and q in A, $uc(p,A)m(p,A) \equiv_T uc(q,A)m(q,A)$ if and only if $vc(p,A)m(p,A) \equiv_T vc(q,A)m(q,A)$. However, we prefer to state a somewhat stronger theorem.

Theorems 2.1 and 3.1 do not speak of a particular undecidable propositional modal logic, but tell about the impossibility of a decision algorithm common to all grammar logics. Indeed, there is an undecidable logic. We use an undecidability result of algebra.

It was proved by Cejtin [Ce] that the word problem "$x \equiv y$?" is undecidable for the Thue system $T=(\{a,b,c,d,e\},P)$, where P consists of the pairs

$$ac = ca,\ ad = da,\ bc = cb,\ bd = db,\ abac = abace,\ eca = ae,\ edb = be.$$

Using this result, we get

Theorem 3.2 *There is an undecidable propositional modal logic.*

Proof. The construction is analogous to the one of the previous theorem, now with Cejtin's system T. Consider the logic with the following axioms and rules of inference:

A1. $A \to (B \to A)$, $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$,
$((A \to \bot) \to \bot) \to A$,
A2. $[a](A \to B) \to ([a]A \to [a]B)$
A3. $[u]A \leftrightarrow [v]A$ for all $(u,v) \in P$,
R1. Modus Ponens,
R2. Necessitation.

The undecidability now follows from the assertion $u \equiv_T v$ if $\vdash_T [u]p \leftrightarrow [v]p$ for an arbitrary propositional variable p. We omit the proof because it is analogous to the proof of Theorem 3.1 and we have already sketched the change in the proof of A3.

4. *Soundness and completeness of Thue logics*

Until now we have concentrated in the undecidability of some propositional modal logics. We shall now prove that the Thue logics are "well-behaving" logics in the sense that they are complete in terms of Kripke style semantics.

In the following, we define the semantics of the Thue logics. To define the meaning of formulas, we shall fix the set of states or worlds, and the relations between the words corresponding to the modal operators.

Formally, a *model* is a triple

$$M = (W, \{R_a \mid a \in MOD\}, m),$$

where W is a nonempty set of *states*, for each a in MOD, R_a is a relation on $W \times W$ (to be characterized later), and m is the *meaning function* mapping each variable a to a subset of W, consisting of those states where this variable is true.

Given a model $M=(W, \{R_a \mid a \in MOD\}, m)$, we say that a formula A is *(un)satisfiable* in a state w, denote M,w sat A (resp. M,w unsat A), iff one of the following conditions holds:

(i) M,w unsat $\perp$,
(ii) M,w sat p ($p \in VAR$) if $w \in m(A)$ else M,w unsat p,
(iii) M,w sat $A \rightarrow B$ iff M,w unsat A or M,w sat B,
(iv) M,w sat [a]A iff for all w′ such that wR_aw', M,w′ sat B.

The set $ext_M(A)=\{w \mid w \in M, M,w \text{ sat } A\}$ is called the *extension* of A in M. A formula A is *true* in a model M, if $ext_M(A)=W$. A formula A is *valid*, $\models A$, iff A is true in all models. A formula A is a *logical consequence of a set S of formulas*, $S \models A$, iff for every model M and for every w in M, if M,w sat B for all $B \in S$ then M,w sat A. A formula A is *satisfiable* iff for some w in W and for some model M of A, M,w sat A. A set S of formulas is satisfiable in a model M and a state w, M,w sat S, iff M,w sat A for all A in S. Finally, a set S of formulas is satisfiable iff M,w sat S for some M and w.

We shall show that if the relations in the models are restricted in a suitable way, the axiomatization of Thue logics is sound and complete.

Consider any model $M=(W, \{R_a \mid a \in MOD\}, m)$. For the empty word e, let R_e be the identity relation on W, and for any $a \in MOD$ and $w \in MOD$, let $R_{aw}=R_a \cdot R_w$, where $\cdot$ refers to the composition of relations.

Let T=(MOD,P) be a Thue system. We say that a model $M=(W, \{R_a \mid a \in MOD\}, m)$ is a *T-model*, iff for all $(u,v) \in P$, $R_v=R_u$. We also speak of *T-satisfiability*, *T-validity* etc. and use the notation $\models_T$ for them.

Theorem 4.1. (Soundness) *If* $\vdash_T A$ *then* $\models_T A$.

Proof. One can easily check that axioms A1 to A3 are T-valid, and R1 and R2 preserve the T-validity. For example, the T-validity $[u]A \leftrightarrow [v]A$ is seen as follows. If M,w does not satisfy [v]A, there is a state w′ such that $(w,w') \in R_v$ and M,w′ does not satisfy A. Now by $R_v \subseteq R_u$, $(w,w') \in R_u$ and hence M,w does not satisfy [u]A either. By symmetry, M,w satisfies $[u]A \leftrightarrow [v]A$.

The converse result, the completeness, states that all valid formulas are theorems. We prove it by the method of canonical models, following the guidelines of [HC].

In order to simplify our reasoning, we use the classical connectives $\neg$, $\vee$, etc. such as they were defined in section 3.

A set S of formulas is T-*consistent*, if there is no finite subset $\{B_1,...,B_n\}$ of S such that $\vdash_T (B_1 \wedge ... \wedge B_n) \rightarrow (\neg A)$. A set S of formulas is *maximal* T-*consistent*, if it is T-consistent and for any formula A, either $A \in S$ or $\neg A \in S$. The following lemmas are quite obvious:

Lemma 4.1. [HC] *Let* S *be a maximal* T-*consistent set of formulas. Then for any formulas* A *and* B,
(*a*) *exactly one of* A *and* $\neg$ A *is in* S,
(*b*) $A \vee B \in S$ *iff* $A \in S$ *or* $B \in S$,
(*c*) *if* $\vdash_T A$ *then* $A \in S$,
(*d*) *if* $\vdash_T A \rightarrow B$ *and* $A \in S$ *then* $B \in S$.

Lemma 4.2. (HC] *Any* T-*consistent set* S *can be extended to a maximal* T-*consistent set.*

For any set S of formulas and $u \in MOD^*$, denote

$$S^u = \{A \mid [u]A \in S\}.$$

The *canonical* T-*model* (W, $\{R_a \mid a \in MOD\}$, m) is defined as follows:

(1) W is the set of all maximal T-consistent sets of formulas,
(2) for any $w,w' \in W$, $(w,w') \in R_a$ iff $w^a \subseteq w'$,
(3) for any variable A, $m(A) = \{w \mid A \in w\}$.

We shall now prove that the canonical model, indeed, is a model:

Lemma 4.3. *The canonical* T-*model* $M = (W, \{R_a \mid a \in MOD\}, m)$ *is a* T-*model.*

Proof. We must show that if $(u,v) \in P$, then $R_u = R_v$. By symmetry, it is sufficient to prove that $R_v \subseteq R_u$. Assume $(w,w') \in R_v$, i.e. $w \subseteq w'$. In order to show the inclusion $R_v \subseteq R_u$, we must show that $(w,w') \in R_u$. Therefore, for any $[u]A \in w$, we should prove $A \in w'$. If $[u]A \in w$, then by $\vdash_T [u]A \rightarrow [v]A$ and Lemma 4.1.(d), $[v]A \in w$. $A \in w'$ now follows from $w^v \subseteq w'$.

The completeness theorem is largely based on the following property of the canonical model:

Lemma 4.4. *Let* $M=(W, \{R_a \mid a \in MOD\}, m)$ *be the cononical model of* T. *Then for any formula* A *and any world* $w \in W$, M,w *sat* A *iff* $A \in w$.

Proof. The Lemma holds for propositional variables, by the definition of M. We shall prove that if it holds for formulas A and B, then it will hold for $\neg A$, $A \vee B$, and [a]A, too.

Consider the formula $\neg A$. By induction, M,w sat A iff $A \in w$. Hence, M,w sat $\neg A$ iff not M,w sat A iff $A \notin w$. But now, by Lemma 4.1.(a), $A \notin w$ iff $\neg A \in w$.

By the definition of sat, M,w sat $A \vee B$ iff M,w sat A or M,w sat B. By induction hypothesis, M,w sat A or M,w sat B iff $A \in w$ or $B \in w$. By Lemma 4.1.(b), this equivalent to $A \vee B \in w$.

Consider now [a]A. "if". If $[a]A \in w$ then by the definition of R_a, $A \in w'$ for all w′ satisfying $(w,w') \in R_a$. By induction hypothesis, for all these w′, L,w′ sat A, which implies M,w sat [a]A.

"only if". If $[a]A \notin w$, then by Lemma 4.1.(a), $\neg [a]A \in w$. First we shall show that $w^a \cup \{\neg A\}$ is T-consistent. Otherwise there would exist formulas $B_1,...,B_n \in w^a$ such that $\vdash_T (B_1 \wedge ... \wedge B_n) \rightarrow A$. By the rule of necessity and by A1, also $\vdash_T [a](B_1 \wedge ... \wedge B_n) \rightarrow [a]A$. By further applications of A1 and propositional calculus, $\vdash_T [a]B_1 \wedge ... \wedge [a]B_n \rightarrow [a]A$. But now, because $[a]B_i \in w$, by Lemma 4.1.(d) also $[a]A \in w$ would hold, contradicting $\neg [a]A \in w$ and the T-consistency of w. Hence, $w^a \cup \{\neg A\}$ is T-consistent. Let w′ be a maximal T-consistent set containing it. By induction hypothesis, M,w′ satisfies $\neg A$ and does not satisfy A. By the definition of R_a, $(w,w') \in R_a$. Hence, M,w does not satisfy [a]A.

We are now ready to prove the completeness theorem:

Theorem 4.2. (Completeness) *If* $\models_T A$ *then* $\vdash_T A$.

Proof. If a formula A is T-valid, it is true in all T-models, by Lemma 4.3 especially in the canonical T-model. Assume that A were not a theorem. Hence, $\neg A$ is T-consistent. Let w be a maximal T-consistent set containing $\neg A$. By Lemma 4.4, M,w satisfies $\neg A$, and consequently cannot satisfy A, contradicting our assumption.

Remark 4.1. The logics we have introduced are multimodal. The above proofs remain valid if we replace every a_i in $MOD=\{a_1,...,a_n\}$ by an affirmative modality $\Diamond^i\Box$ and use the Lemmon correspondence, which provides every affirmative modality $\emptyset$ with an accessibility relation $R^{\emptyset}$ characterizing $\emptyset$ (see Lemmon [Le], pp. 62-63).

5. *Conclusion*

In this note we have presented a method to define modal logics that simulate the behaviour of grammars. As a corollary we get a simple proof for the undecidability of a propositional modal logic. Of course, if more restricted classes of grammars (eg. regular grammars) are considered, these logics may be decidable. This methodology is related with the logic grammars (see [Co] and [PW]) that have become popular in logic programming. However, grammars based on classical logic need predicate calculus, while in our case propositional calculus was sufficient to get full computational power. Still for the easiness of expression and practical efficiency the introduction of predicates is useful.

We have not considered here the mechanical proof procedures that are very important in programming. It will be the matter of future research.

Acknowledgements. We are grateful to David Makinson for valuable comments, both on general structure of the representation and on details such as Remark 4.1, and to Andrés Raggio, who revealed an error in the undecidability proof of an earlier version of this paper. This work was funded by the ALPES project of ESPRIT, and by the Academy of Finland.

Université Paul Sabatier &
University of Joensuu

L. FARIÑAS DEL CERRO &
MARTTI PENTTONEN

REFERENCES

[Ce] G.S. Cejtin: An associative calculus with an unsolvable problem of equivalence (In Russian). *Doklady Akademii Nauk SSSR* 107, 370-371 (1956).

[Co] A. Colmerauer: Metamorphosis grammars. In L. Bloc (ed): *Natural language Communication with Computer*, 133-189. Springer Verlag 1979.

[HC] G.E. Hughes, M.J. Creswell: *A Companion to Modal Logic*. Methuen 1984.

[Ko] K. Konolige: A deduction model of belief and its logic. SRI International, Technical Note 326, 1984, Menlo Park, California, USA.

[Le] E.J. Lemmon: An Introduction to modal logic. *American Philosophical Quaterly*, Oxford 1977.

[PW] F.L.M. Pereira, D.H.D. Warren: Definite clause grammars for language analysis. *Artificial Intelligence* 13, 231-278 (1980).

[Ra] M.O. Rabin: Decidability of second order theories and automata on infinite trees. *Transactions of American Mathematical society* 1969, 1-35.

[Sa] A Salomaa: *Formal Languages*. Academic Press 1973.

[Wo] P. Wolper: Temporal logic could be more expressive. *Information and Control* 56, 72-99 (1983).