ALGEBRAIC METHODS IN LOGIC AND IN COMPUTER SCIENCE
BANACH CENTER PUBLICATIONS, VOLUME 28
INSTITUTE OF MATHEMATICS
POLISH ACADEMY OF SCIENCES
WARSZAWA 1993

ON THE DETECTION OF SOME PERIODIC LOOPS DURING THE EXECUTION OF PROLOG PROGRAMS

DIMITER SKORDEV

Department of Mathematics, Sofia University
Blvd. J. Bourchier 5, 1126 Sofia, Bulgaria

1. Introduction. As is well known, the usual Prolog inference-strategy may easily lead to infinite loops in the execution of programs, and sometimes even in cases when a systematic search would give a result after a small number of steps. Therefore the problem how to detect such loops has a practical significance, and a number of authors have studied this problem (cf., for example, [3, 4, 5, 10, 12, 13, 16, 1, 2, 7, 11, 6]). Unfortunately, it is not possible to give a general effective solution of the problem, due to undecidability results of the Theory of Computability. So, any proposed solution can only be partial, and no best solution could be expected to exist in any absolute sense.

The present paper treats the problem of detection of periodic loops during the execution of programs. We use a variation of the approach from our abstract [15], and attention is now also paid to more details concerning the detection algorithm (in the case of propositional Prolog programs, the approach from [15], attacking the loop detection problem for deterministic recursive algorithms in the sense of [9], can be used without modification, but the exposition of part of the mentioned algorithmic details is not superfluous even in that case). Using this approach we proceed in a direction whose characteristic feature is looking for repetition of situations during the execution of the programs and detecting the loops on the base of the observed repetitions (previous work of other authors in this direction is presented, for example, in [3, 4, 5, 16, 1]).

1991 *Mathematics Subject Classification*: 68N17, 68N05.

Research partially supported by the Bulgarian Ministry of Science and Higher Education, Contracts Nos. 56/1990, 43/1991.

The paper is in final form and no version of it will be published elsewhere.

2. Formulation of the problem. A Prolog program is a finite sequence of *program clauses* and operates on *queries*. The process of execution of a Prolog program on a given query is called *evaluation of the query on the base of the program* and, roughly speaking, consists in consecutive generation of *goals* by application of *SLD-resolution*. We restrict ourselves to the case when the program clauses and the queries are built only from atomic formulas of some first order language ([1]) (thus we do not allow, for example, cut operators to occur in program clauses or queries). The reader is expected to be familiar with such Prolog programs and with the basic notions concerning their execution. We shall assume that during a depth-first search all applications of resolution are done upon the leftmost subgoal of the current goal and use the first program clause whose head is unifiable with this subgoal. Such an application of resolution will be called a *canonical resolution step*. It is a single-valued partial operation up to renaming variables.

At the beginning of a depth-first search, the program starts its job having some goal for evaluation. If a canonical resolution step is applicable to this goal then the given goal is replaced by the result of such a step. The same is done with the new goal, and so on as long as possible. A *loop* arises in the considered depth-first search if the described process goes forever. Since the detection of such loops is not effectively possible in the general case, we shall study the following problem: define a reasonable notion of periodic loop in such a way that each such loop could be effectively detected after generating finitely many members of the mentioned sequence of goals.

An unsatisfactory solution is to consider only the case when the sequence of the generated goals is ultimately periodic up to renaming variables in its members. Loops of this sort can be detected by an algorithm using information about only two of the generated goals (the last generated goal being one of them). One could, for example, use a method proposed (for arbitrary iterative programs) by R. P. Brent or its modification and generalization (again for that case) by the present author (cf. [8, Section 3.1, Exercise 7] and [14], respectively, or Section 6 of the present paper) ([2]). The so-called tortoise-and-hare technique (cf. [16]) can also be used, but for comparing the complete goals instead of comparing only their first subgoals as in [16] ([3]). The great disadvantage of a restriction only to this

([1]) We shall use notations which are usual for a certain dialect of Prolog: the constants, the functional symbols and the propositional and predicate symbols will be strings beginning with small letters from the Roman alphabet, and the variables will be strings beginning with capital letters from the same alphabet.

([2]) No previous work on loop detection is quoted in our paper [14], due to the lack of information in this respect at the moment of writing the paper. For a similar reason, no citation of Brent's method is given in the abstract [15] (only the first edition of the book [8] was at our disposal until recently).

([3]) A repetition of the first subgoal is not sufficient, in general, for being sure that the evaluation process is non-terminating, and therefore the method proposed in [16] is incorrect (cf. footnote to Example 7 of the present paper).

sort of loops is that such loops are not typical for the execution of Prolog programs and therefore it would be very insufficient to be able to detect only them.

We shall consider a larger class of loops with some periodic features, to be described further. An algorithm will be proposed which can be based, in essential, on some of the first two methods mentioned above and which will have almost the same complexity characteristics as in the previous restricted case.

3. Towards a more abstract treatment of the problem. As we noted in the previous section, a canonical resolution step is a single-valued operation up to renaming variables. In general, it is not single-valued in the ordinary sense. One usually does not care very much about this and does not distinguish between two goals which are equal up to renaming variables. There is, however, one unpleasant problem which requires more carefulness in this respect. The problem is that, when considering finite sequences, one is inclined to think in terms of concatenation, but concatenation, unfortunately, does not preserve equality up to renaming variables. Therefore, when we do the mentioned identification, we shall avoid using concatenation of sequences of formulas.

From now on, we shall consider objects called *abstract goals*. These will be the equivalence classes of goals with respect to equality up to renaming variables. The set of all abstract goals (in the given first order language) will be denoted by $\mathcal{G}$. In this set a correct definition is possible of the partial operation of deleting the last subgoal. Namely, let $G \in \mathcal{G}$, and let the ordinary goal $:- A_1, \ldots, A_m$ belong to G. If $m > 0$ then the result of applying the operation in question to G will be the equivalence class of the ordinary (possibly empty) goal $:- A_1, \ldots, A_{m-1}$; this equivalence class will be called the *predecessor* of G and will be denoted by $\pi(G)$. If $m = 0$ then application of the operation to G is not possible, and $\pi(G)$ is not defined. The correctness of this definition follows from the fact that if two ordinary goals are equal up to renaming variables, and one of them is non-empty, then so is the other, and the results of deleting the last subgoals of the given goals are again equal up to renaming variables.

LEMMA 1. *For each G in $\mathcal{G}$, there is some natural number k such that $G \notin \mathrm{dom}(\pi^k)$.*

Proof. If $G \in \mathcal{G}$ and $:- A_1, \ldots, A_m$ belongs to G then $G \notin \mathrm{dom}(\pi^{m+1})$. ∎

Now the main point is that a canonical resolution step can be represented by a single-valued partial unary operation ϱ in $\mathcal{G}$. To define $\varrho(G)$, where G is a given element of $\mathcal{G}$, we proceed as follows: we take an ordinary goal from the equivalence class G and try to do a canonical resolution step with it; if the step gives some result then we take $\varrho(G)$ to be its equivalence class, otherwise we consider $\varrho(G)$ not defined. To prove the correctness of this definition, one has to prove two facts: independence from the choice of a result of the canonical resolution step, and independence from the choice of an ordinary goal from G. The first of these facts is equivalent to the statement that the result of a canonical resolution step,

if any, is unique up to renaming variables. The second statement follows from the fact that if the canonical resolution step gives some result on an ordinary goal, then the same result can be obtained from any other goal equal to the first one up to renaming variables.

Adopting the convention that $\varphi\psi = \lambda G.\varphi(\psi(G))$ for any two partial mappings φ and ψ of $\mathcal{G}$ into $\mathcal{G}$ and using $\varphi \subseteq \psi$ to denote that φ is a restriction of ψ, we have two more lemmas.

LEMMA 2. *The inclusion* $\varrho\pi \subseteq \pi\varrho$ *holds.*

Proof. Let the application of $\varrho\pi$ to some element G of $\mathcal{G}$ give the result H. Then $G \in \operatorname{dom}\pi$ and $\varrho(\pi(G)) = H$. Consider an arbitrary element $:- A_1, \ldots, A_m$ of G. Then $m > 0$ and $:- A_1, \ldots, A_{m-1}$ is an element of $\pi(G)$. Thanks to the equality $\varrho(\pi(G)) = H$, a canonical resolution step can be done with the goal $:- A_1, \ldots, A_{m-1}$ (hence $m > 1$) and the result belongs to H. Let $A :- B_1, \ldots, B_n$ be the first clause in P such that canonical application of resolution is possible to it and to $:- A_1, \ldots, A_{m-1}$. Let $A' :- B'_1, \ldots, B'_n$ be a clause equal to $A :- B_1, \ldots, B_n$ up to renaming variables, but having no common variables with the goal $:- A_1, \ldots, A_{m-1}, A_m$. Then A' and A_1 are unifiable. Denote by Θ a most general unifier of A' and A_1. It is clear that $B'_1\Theta, \ldots, B'_n\Theta, A_2\Theta, \ldots, A_{m-1}\Theta$ will be a result of a canonical resolution step with $:- A_1, \ldots, A_{m-1}$, and it is easy to verify that $:- B'_1\Theta, \ldots, B'_n\Theta, A_2\Theta, \ldots, A_{m-1}\Theta, A_m\Theta$ will be a result of a canonical resolution step with $:- A_1, \ldots, A_{m-1}, A_m$. Since $:- B'_1\Theta, \ldots, B'_n\Theta, A_2\Theta, \ldots, A_{m-1}\Theta$ belongs to H, and $:- B'_1\Theta, \ldots, B'_n\Theta, A_2\Theta, \ldots, A_{m-1}\Theta, A_m\Theta$ belongs to $\varrho(G)$, we see that $H = \pi(\varrho(G))$, i.e. the application of $\pi\varrho$ to G also gives the result H. ∎

LEMMA 3. *The inclusions* $\operatorname{dom}\varrho \subseteq \operatorname{dom}\pi$, $\operatorname{dom}\varrho \cap \operatorname{dom}(\pi^2) \subseteq \operatorname{dom}(\varrho\pi)$ *hold.*

Proof. Let $G \in \operatorname{dom}\varrho$, and let $:- A_1, \ldots, A_m$ belong to G. Then a canonical resolution step can be performed with $:- A_1, \ldots, A_m$. Hence $m > 0$ and therefore $:- A_1, \ldots, A_m$ belongs to $\operatorname{dom}\pi$. Suppose now that G belongs also to $\operatorname{dom}(\pi^2)$. Then $\pi(G) \in \operatorname{dom}\pi$, and since $:- A_1, \ldots, A_{m-1}$ belongs to $\pi(G)$, we may be sure that $m > 1$. Thus $:- A_1, \ldots, A_{m-1}$ will have the first subgoal A_1 which is the same as the first subgoal of $:- A_1, \ldots, A_m$. Hence a canonical resolution step can also be performed with $:- A_1, \ldots, A_{m-1}$, and this shows that $\pi(G) \in \operatorname{dom}\varrho$, i.e. $G \in \operatorname{dom}(\varrho\pi)$. ∎

In the next sections, we shall use the properties stated in the above three lemmas to define a more convenient notion of periodic loop during depth-first search than the one described in Section 2 and to give a method for detection of such loops.

4. Goal spaces and their partial ordering. From this section on, we shall proceed in an axiomatic way. Namely, we shall consider an arbitrary set $\mathcal{G}$ together with two partial unary operations π and ϱ in it having the properties

from Lemmas 1–3. The triple $\langle \mathcal{G}, \pi, \varrho \rangle$ will then be called a *goal space*. The reader who is not interested in this level of generality may simply think that $\mathcal{G}$, π, ϱ have the same meaning as in the previous section (however, he or she must then skip the application to deterministic Mazurkiewicz algorithms at the end of Section 6). Accordingly, we shall use notations and terminology which are in consonance with the above interpretation of $\mathcal{G}$, π, ϱ (in particular, the elements of $\mathcal{G}$ will sometimes be called goals). For the other readers, we give two examples of goal spaces not of the above kind—an artificial example and a natural one.

EXAMPLE 1. Let $\mathcal{G}$ be the set of all positive integers, $\operatorname{dom}\pi$ be the set of the even ones, $\operatorname{dom}\varrho$ be the set of positive integers divisible by 10, and the actions of π and ϱ be multiplication by $\frac{3}{2}$ and $\frac{1}{5}$, respectively. Then $\langle \mathcal{G}, \pi, \varrho \rangle$ is a goal space.

EXAMPLE 2. This example is connected with a deterministic case of recursive algorithms in the sense of [9] (this case, mentioned above, is considered in [15] from the point of view of loop detection). Let E and M be some sets (the set of *labels* and the set of *memory states*). An *E-instruction over M* is, by definition, a triple (λ, f, w), where $\lambda \in E$, f is a partial mapping of M into M, and $w \in E^*$ (i.e. w is a string of labels). Suppose a set R of E-instructions over M is given such that $\operatorname{dom} f_1 \cap \operatorname{dom} f_2 = \emptyset$ whenever (λ, f_1, w_1) and (λ, f_2, w_2) are two distinct elements of R. Let ϱ be the least defined partial mapping of $E^* \times M$ into itself such that $\varrho(v\lambda, x) = (vw, f(x))$ whenever $v \in E^*$, $(\lambda, f, w) \in R$, and $x \in \operatorname{dom} f$ (this partial mapping is denoted by S in [15]). Let π be the least defined partial mapping of $E^* \times M$ into itself such that $\pi(\lambda v, x) = (v, x)$ whenever $\lambda \in E$, $v \in E^*$, $x \in M$. Then $\langle E^* \times M, \pi, \varrho \rangle$ is a goal space.

DEFINITION 1. Let $G', G \in \mathcal{G}$. We shall say that G' is a *beginning* of G iff $G' = \pi^p(G)$ for some non-negative integer p. We shall then write $G' \leq G$.

Remark 1. This terminology is appropriate for a "standard" example of goal space, as mentioned in the first paragraph of this section. In the case of a goal space of the kind considered in Example 2, the term "end" would be more appropriate than "beginning".

From the above definition, reflexivity and transitivity of $\leq$ follow immediately. The definition also implies that any two beginnings of an abstract goal are comparable in the sense that one of them is a beginning of the other. To prove that $\leq$ is a partial ordering in $\mathcal{G}$, it is sufficient to prove that $p = 0$ is the only case when $G \in \operatorname{dom}(\pi^p)$ and $\pi^p(G) = G$ (this obviously implies the anti-symmetry of $\leq$). The proof is straightforward: if $G \in \operatorname{dom}(\pi^p)$ and $\pi^p(G) = G$ then an easy induction shows that, for each natural number n, $G \in \operatorname{dom}(\pi^{np})$ and $\pi^{np}(G) = G$, and, due to the property from Lemma 1, this is possible only if $p = 0$. Note that the statement just proved also implies the following property of π: whenever $G \in \operatorname{dom}(\pi^k)$, $G \in \operatorname{dom}(\pi^l)$ and $\pi^k(G) = \pi^l(G)$, then $k = l$. So there is a one-to-one correspondence between the set of all beginnings of a given element G of $\mathcal{G}$ and the set of all natural numbers p satisfying $G \in \operatorname{dom}(\pi^p)$.

PROPOSITION 1. *Let k and l be non-negative integers, and let $G \in \mathrm{dom}(\pi^k)$, $G \in \mathrm{dom}(\pi^l)$. Then $\pi^k(G) \leq \pi^l(G)$ iff $k \geq l$.*

Proof. If $\pi^k(G) \leq \pi^l(G)$ then $\pi^k(G) = \pi^p(\pi^l(G)) = \pi^{p+l}(G)$ for some natural number p, hence $k = p + l$ and therefore $k \geq l$. Conversely, if $k \geq l$ then $\pi^k(G) = \pi^{k-l}(\pi^l(G)) \leq \pi^l(G)$. ∎

Of course, each element of $\mathcal{G}$ belongs to $\mathrm{dom}(\pi^0)$, and whenever $G \notin \mathrm{dom}(\pi^k)$, then $G \notin \mathrm{dom}(\pi^i)$ for all $i > k$. Hence, for each goal G, there is a greatest non-negative integer i with $G \in \mathrm{dom}(\pi^i)$.

DEFINITION 2. If $G \in \mathcal{G}$ then the greatest non-negative integer i satisfying $G \in \mathrm{dom}(\pi^i)$ will be called the *length* of $\mathcal{G}$ and denoted by $|G|$.

This definition immediately yields

PROPOSITION 2. *For all G in $\mathrm{dom}(\pi^p)$, $|\pi^p(G)| = |G| - p$.*

COROLLARY 1. *Let G' and G'' be beginnings of one and the same goal. Then $G'' \leq G'$ iff $|G''| \leq |G'|$.*

So, if a goal G has length l, then, assigning to each beginning of G its length, we get an order preserving one-to-one correspondence between the set of all beginnings of G and the set $\{0, 1, \ldots, l\}$. Therefore, we may use without ambiguity phrases like "the shortest beginning of G such that ..." in all cases when there is at least one beginning of G with the considered property.

Using the inclusion $\varrho\pi \subseteq \pi\varrho$, one easily proves (by induction) that $\varrho^r\pi^p \subseteq \pi^p\varrho^r$ for all natural numbers r and p. Hence we get the following proposition.

PROPOSITION 3. *Let r be a natural number, $G' \in \mathrm{dom}(\varrho^r)$ and $G' \leq G$. Then:* (i) *$G \in \mathrm{dom}(\varrho^r)$ and $\varrho^r(G') \leq \varrho^r(G)$;* (ii) *$|\varrho^r(G)| - |\varrho^r(G')| = |G| - |G'|$ and* (iii) *if $G \leq \varrho^r(G)$ then $G' \leq \varrho^r(G')$.*

Proof. Let p be a natural number such that $G' = \pi^p(G)$. Then $\varrho^r(G') = \varrho^r\pi^p(G)$. Therefore $G \in \mathrm{dom}(\pi^p\varrho^r)$, hence $G \in \mathrm{dom}(\varrho^r)$, and $\varrho^r(G') = \pi^p\varrho^r(G) = \pi^p(\varrho^r(G)) \leq \varrho^r(G)$. By Proposition 2, we get $|G'| = |G| - p$, $|\varrho^r(G')| = |\varrho^r(G)| - p$. From these equalities we see that $|\varrho^r(G)| - |\varrho^r(G')| = |G| - |G'| = p$. Suppose now that $G \leq \varrho^r(G)$. Then $G = \pi^q(\varrho^r(G))$ for some natural number q. Hence $G' = \pi^p(\pi^q(\varrho^r(G))) = \pi^q(\pi^p(\varrho^r(G))) = \pi^q(\varrho^r(G')) \leq \varrho^r(G')$. ∎

Remark 2. The statement (i) follows easily (by induction) from its special case when $r = 1$.

5. Cyclic elements and periodic loops in goal spaces. We again suppose that a goal space $\langle \mathcal{G}, \pi, \varrho \rangle$ is given. Now we shall try to define a convenient notion of a cyclic element of $\mathcal{G}$, having in mind an element which in some sense reproduces itself after application of some function ϱ^r with a positive r (the period of the considered cyclic element). The simplest solution would be to call an element G of $\mathcal{G}$ cyclic with period r iff $G \in \mathrm{dom}(\varrho^r)$ and $\varrho^r(G) = G$. This, however, would not

be a convenient definition, for the reasons mentioned in Section 2. Nevertheless, to have a term for this notion, we shall call any such element G *perfectly cyclic with period* r.

An essential property of perfectly cyclic elements with period r is that they all belong to $\operatorname{dom}\varrho$ and are transformed by ϱ again into such elements. Let us call a subset $\mathcal{E}$ of $\mathcal{G}$ *invariant with respect to* ϱ iff $\mathcal{E} \subseteq \operatorname{dom}\varrho$ and $\varrho(G) \in \mathcal{E}$ for each G from $\mathcal{E}$. Then the above property of perfectly cyclic elements can be expressed by the statement that, for any positive integer r, the set of perfectly cyclic elements of $\mathcal{G}$ with period r is invariant with respect to ϱ. It is desirable that a similar statement holds for the better notion of cyclic element which we want to introduce. Namely, it is clear that in this case obtaining a cyclic goal in the process of execution of a Prolog program will imply non-termination of the execution: whenever $\mathcal{E}$ is a subset of $\mathcal{G}$ invariant with respect to ϱ, then $\mathcal{E} \subseteq \operatorname{dom}(\varrho^i)$ for each natural number i.

A better notion of cyclic element is given by the following definition.

DEFINITION 3. Let r be a positive integer. An element G of $\mathcal{G}$ will be called *strongly cyclic with period* r iff $G \in \operatorname{dom}(\varrho^r)$ and $G \leq \varrho^r(G)$.

The state of affairs considered in the above definition can be visualized by Fig. 1.

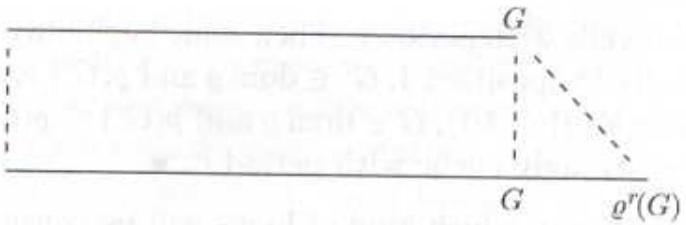

Fig. 1. A strongly cyclic element G with period r

Remark 3. The reflexivity of the relation $\leq$ ensures that all perfectly cyclic elements with period r are strongly cyclic with period r.

PROPOSITION 4. *For each positive integer* r, *the set of strongly cyclic elements of* $\mathcal{G}$ *with period* r *is invariant with respect to* ϱ.

Proof. Let r be a positive integer, and let G be an element of $\mathcal{G}$ strongly cyclic with period r. Clearly (since $G \in \operatorname{dom}(\varrho^r)$), $G \in \operatorname{dom}\varrho$. Using this fact, the inequality $G \leq \varrho^r(G)$ and Proposition 3(i), we conclude that $\varrho^r(G) \in \operatorname{dom}\varrho$ and $\varrho(G) \leq \varrho(\varrho^r(G)) = \varrho^r(\varrho(G))$. Hence $\varrho(G)$ is again a strongly cyclic element with period r. ∎

In the Prolog interpretation of what we are now doing, an abstract goal G is strongly cyclic with period r if it has the following property: starting from an ordinary goal belonging to G, we can do r consecutive canonical resolution steps, and after doing them we shall have an ordinary goal with a beginning

equal to the initial ordinary goal up to renaming variables. This situation is more frequently encountered than the complete repetition of the goal up to renaming variables. However, we can further enlarge the applicability of our considerations by introducing a further, more general notion of a cyclic element.

DEFINITION 4. Let r be a positive integer. An element G of $\mathcal{G}$ will be called *cyclic with period* r iff some beginning of G is strongly cyclic with period r.

The state of affairs considered in this definition can be visualized by Fig. 2.

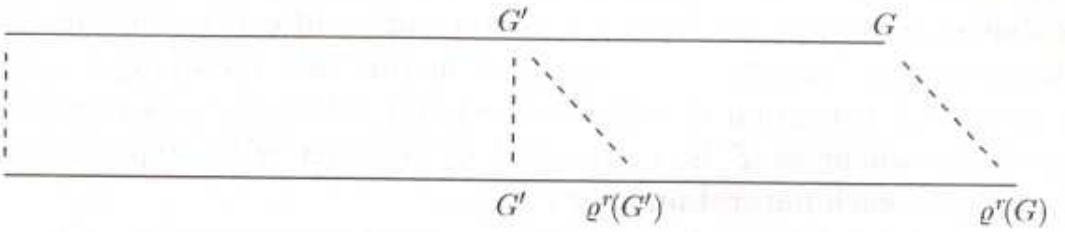

Fig. 2. A cyclic element G with period r and a strongly cyclic beginning G' of G

Remark 4. The reflexivity of the relation $\leq$ ensures that all strongly cyclic elements with period r are cyclic with period r.

PROPOSITION 5. *For each positive integer r, the set of cyclic elements of $\mathcal{G}$ with period r is invariant with respect to ϱ.*

Proof. Let G be cyclic with period r. Then some beginning G' of G is strongly cyclic with period r. By Proposition 4, $G' \in \operatorname{dom} \varrho$ and $\varrho(G')$ is also strongly cyclic with period r. By Proposition 3(i), $G \in \operatorname{dom} \varrho$ and $\varrho(G') \leq \varrho(G)$. Hence $\varrho(G)$ has a beginning which is strongly cyclic with period r. ∎

We are now ready to say which kind of loops will be considered.

DEFINITION 5. An element G of $\mathcal{G}$ is called *cyclic* iff there is a positive integer r such that G is cyclic with period r.

From this definition and Proposition 5 we get the following corollary.

COROLLARY 2. *If G is a cyclic element of $\mathcal{G}$ then $G \in \operatorname{dom}(\varrho^i)$ for each natural number i.*

DEFINITION 6. A *ϱ-path* in $\mathcal{G}$ is a finite or infinite sequence $\{G_t\}_{t \in I}$ of elements of $\mathcal{G}$, where I is some finite initial segment of the set $\mathcal{N} = \{0, 1, 2, \ldots\}$ or $I = \mathcal{N}$, and the following condition is satisfied: for each t in I, the number $t + 1$ belongs to I iff $G_t \in \operatorname{dom} \varrho$, in this case the equality $G_{t+1} = \varrho(G_t)$ being valid.

DEFINITION 7. Let $\{G_t\}_{t \in I}$ be a ϱ-path in $\mathcal{G}$. We say that *a periodic loop (with period r) is present* in this ϱ-path iff there is a cyclic element (with period r) among its members G_t.

PROPOSITION 6. *If a periodic loop with period r is present in the ϱ-path $\{G_t\}_{t \in I}$ then $I = \mathcal{N}$ and all G_t from some t on are cyclic with period r.*

Proof. By Proposition 5 and Definition 6, if $t \in I$ and G_t is a cyclic element with period r, then $t+1 \in I$ and G_{t+1} is a cyclic element with period r, too. ■

Here is a simple example of a periodic loop of the above kind, arising in the execution of a Prolog program.

EXAMPLE 3. Let the following propositional Prolog program be given:

```
p :- q.
q.
r :- p, r, q.
s :- p, r, s.
```

Assume that the query ?- q, s must be evaluated. Since no variables are present in the program clauses and in the query, we shall use ordinary goals instead of abstract ones as elements of the goal space $\mathcal{G}$ (the definitions of π and ϱ are, of course, easier in this case). The first six members of the ϱ-path generated in the evaluation process are:

G_0	:- q, s.	G_3	:- q, r, s.
G_1	:- s.	G_4	:- r, s.
G_2	:- p, r, s.	G_5	:- p, r, q, s.

They are sufficient to see that a periodic loop with period 3 is present in this ϱ-path: although G_2 is not a beginning of G_5, the beginning :- p, r of G_2 goes into the goal :- p, r, q after three canonical resolution steps (in some sense, the additional subgoal s plays no essential role and remains passive during the corresponding three canonical resolution steps starting with the whole G_2). Note that the beginning :- p of G_2 is "too short" for it being possible to do three canonical resolution steps starting with it.

Our next task will be to give a convenient means for checking whether an element of a goal space is a cyclic goal with a period not exceeding some given upper bound.

According to Definition 4, to check whether a given element G of $\mathcal{G}$ is cyclic with a given period r, we ought to check all beginnings of G whether they are strongly cyclic with this period. Fortunately, there is a much better strategy. Namely, Proposition 3 implies the following statements: (i) if an element of $\mathcal{G}$ is cyclic with period r then it belongs to $\mathrm{dom}(\varrho^r)$; (ii) whenever an element G' of $\mathcal{G}$ is strongly cyclic with period r, then each beginning of G' belonging to $\mathrm{dom}(\varrho^r)$ is also strongly cyclic with period r. Of course, if an element G of $\mathcal{G}$ belongs to $\mathrm{dom}(\varrho^r)$ then there is a shortest beginning of G belonging to $\mathrm{dom}(\varrho^r)$. This leads to the following conclusion:

PROPOSITION 7. *Let r be a positive integer. An element G of $\mathcal{G}$ is cyclic with period r iff $G \in \mathrm{dom}(\varrho^r)$ and the shortest beginning of G belonging to $\mathrm{dom}(\varrho^r)$ is strongly cyclic with period r.*

A further simplification of the checking procedure is possible.

PROPOSITION 8. *Let r be a positive integer, $G', G \in \operatorname{dom}(\varrho^r)$, $G' \leq G$, $G' \leq \varrho^r(G)$ and $|G| \leq |\varrho^r(G)|$. Then G' is strongly cyclic with period r, and hence G is cyclic with period r.*

Proof. By Proposition 3(iii),

$$|\varrho^r(G)| - |G| = |\varrho^r(G')| - |G'|. \tag{1}$$

Hence $|G'| \leq |\varrho^r(G')|$. But both G' and $\varrho^r(G')$ are beginnings of $\varrho^r(G)$. Therefore (by Corollary 1), $G' \leq \varrho^r(G')$, i.e. G' is strongly cyclic with period r. ∎

THEOREM 1. *Let G be an arbitrary element of $\mathcal{G}$. Then G is cyclic with period r iff $G \in \operatorname{dom}(\varrho^r)$, $|G| \leq |\varrho^r(G)|$ and the shortest beginning of G belonging to $\operatorname{dom}(\varrho^r)$ is a beginning of $\varrho^r(G)$.*

Proof. Suppose G is cyclic with period r. Then, by Proposition 6, $G \in \operatorname{dom}(\varrho^r)$ and, denoting by G' the shortest beginning of G belonging to $\operatorname{dom}(\varrho^r)$, we have $G' \leq \varrho^r(G')$. Since $\varrho^r(G') \leq \varrho^r(G)$, we get $G' \leq \varrho^r(G)$. By Proposition 3(iii), we also have (1). Since $|G'| \leq |\varrho^r(G')|$, we get $|G| \leq |\varrho^r(G)|$.

The converse follows from Proposition 8. ∎

Theorem 1 reduces the check whether a given element G of $\mathcal{G}$ is periodic with a given period r to checking whether $G \in \operatorname{dom}(\varrho^r)$ and, in case this condition is satisfied, to finding $\varrho^r(G)$ and the shortest beginning of G belonging to $\operatorname{dom}(\varrho^r)$ (in the Prolog interpretation of the present considerations, the other things which must be done cause no difficulties, and we shall consider them "easy" also in the general case). Checking whether an element belongs to $\operatorname{dom}(\varrho^r)$ and finding $\varrho^r(G)$ in such a case cause no additional troubles if proceeding without loop detecting activities requires generating consecutively $\varrho^i(G)$ for all i such that $G \in \operatorname{dom}(\varrho^i)$. So the problem is how to find efficiently the shortest beginning of G belonging to $\operatorname{dom}(\varrho^r)$ (in case $G \in \operatorname{dom}(\varrho^r)$). To make steps in this direction, we shall also use the property from Lemma 3.

PROPOSITION 9. *Let $G \in \operatorname{dom}\varrho$. Then the shortest beginning of G belonging to $\operatorname{dom}\varrho$ has length 1.*

Proof. Since $G \in \operatorname{dom}\varrho$, the first inclusion from Lemma 3 shows that $G \in \operatorname{dom}\pi$ and hence $|G| \geq 1$. If p is a non-negative integer such that $p \leq |G| - 2$ and $\pi^p(G) \in \operatorname{dom}\varrho$ then, using the second inclusion from Lemma 3 and the fact that $\pi^p(G) \in \operatorname{dom}(\pi^2)$, we get $\pi^p(G) \in \operatorname{dom}(\varrho\pi)$, which implies $\pi^{p+1}(G) \in \operatorname{dom}\varrho$. Since $\pi^0(G) = G \in \operatorname{dom}\varrho$ the above remark shows that $\pi^p(G) \in \operatorname{dom}\varrho$ for all non-negative integers p satisfying $p \leq |G| - 1$. In particular, this is true for $p = |G| - 1$. On the other hand, $\pi^{|G|}(G) \notin \operatorname{dom}\pi$ and hence $\pi^{|G|}(G) \notin \operatorname{dom}\varrho$. By Definition 1 and Proposition 1, $\pi^{|G|-1}(G)$ is the shortest beginning of G belonging to $\operatorname{dom}\varrho$, and, by Proposition 2, its length is 1. ∎

THEOREM 2. *Let $G \in \mathrm{dom}(\varrho^r)$. For $i = 0, 1 \ldots, r$, let l_i denote $|\varrho^i(G)|$ and p_i be the greatest non-negative integer p not exceeding $|G|$ such that $\pi^p(G) \in \mathrm{dom}(\varrho^i)$. Then $p_0 = |G|$ and $p_{i+1} = \min\{p_i, l_i - 1\}$ for $i = 0, 1, \ldots, r-1$.*

Proof. The equality $p_0 = |G|$ follows from the fact that $\pi^p(G) \in \mathrm{dom}(\varrho^0)$ for all non-negative integers $p \leq |G|$. Suppose now i is a non-negative integer less than r. Then $G \in \mathrm{dom}(\varrho^{i+1})$, i.e. $G \in \mathrm{dom}(\varrho^i)$ and $\varrho^i(G) \in \mathrm{dom}\,\varrho$. Let p be a non-negative integer not exceeding $|G|$, and let $G' = \pi^p(G)$. In order to have $G' \in \mathrm{dom}(\varrho^{i+1})$, it is necessary and sufficient to have $G' \in \mathrm{dom}(\varrho^i)$ and $\varrho^i(G') \in \mathrm{dom}\,\varrho$. The condition $G' \in \mathrm{dom}(\varrho^i)$ is equivalent to $p \leq p_i$. Since $\varrho^i(G') \leq \varrho^i(G) \in \mathrm{dom}\,\varrho$, application of Propositions 9 and 3 shows that $\varrho^i(G') \in \mathrm{dom}\,\varrho$ iff $|\varrho^i(G')| \geq 1$. If $G' \in \mathrm{dom}(\varrho^i)$ then, by Proposition 3(iii), $|\varrho^i(G')| = |\varrho^i(G)| - (|G| - |G'|) = l_i - p$. Hence the inequality $|\varrho^i(G')| \geq 1$ in the above necessary and sufficient condition can be replaced by $p \leq l_i - 1$. Therefore $G' \in \mathrm{dom}(\varrho^{i+1})$ iff $p \leq \min\{p_i, l_i - 1\}$. Since $p_i \leq |G|$ the equality $p_{i+1} = \min\{p_i, l_i - 1\}$ is thus established. ∎

Theorems 1 and 2 give the following method for checking whether a given element G of $\mathcal{G}$ is cyclic with period r, where r is a given positive integer. We generate consecutively the elements $\varrho^i(G)$ for $i = 1, \ldots, r$, and calculate the corresponding numbers p_i using the equalities from Theorem 2. If some of these $\varrho^i(G)$ turns out to be undefined, then surely G will not be cyclic. Suppose all of them, including $\varrho^r(G)$, are defined. Then the shortest beginning of G belonging to $\mathrm{dom}(\varrho^r)$ will be $G' = \pi^p(G)$ with $p = p_r$, and we check whether $|G| \leq |\varrho^r(G)|$ and whether $G' \leq \varrho^r(G)$. By Theorem 1, the element G will be cyclic with period r iff both these conditions are satisfied.

Only a small modification of the above method is needed to make it applicable for checking whether an element G of $\mathcal{G}$ is periodic with period not exceeding a positive integer r. The modification is the following: when some $\varrho^i(G)$ with $1 \leq i \leq r$ turns out to be defined, we check whether $|G| \leq |\varrho^i(G)|$ and whether $\pi^p(G) \leq \varrho^i(G)$ for $p = p_i$.

EXAMPLE 4. In the situation of Example 3, we shall use the above method for checking whether $G = G_2$ is cyclic with period not exceeding 3. To do this, we must generate $\varrho(G) = G_3$, $\varrho^2(G) = G_4$, $\varrho^3(G) = G_5$ and calculate the numbers p_1, p_2, p_3 ($p_1 = 2$, $p_2 = 2$, $p_3 = 1$). Since $\pi^2(G_2)$ is not a beginning of G_3, and $|G_2| > |G_4|$, it is clear that G is not cyclic with period 1 or 2. On the other hand, $|G_2| < |G_5|$, and $\pi^1(G_2)$ is a beginning of G_5. Hence G is cyclic with period 3.

6. Detection of periodic loops in goal spaces. We again assume that a goal space $\langle \mathcal{G}, \pi, \varrho \rangle$ is fixed. Now we also suppose that a ϱ-path $\{G_t\}_{t \in I}$ in $\mathcal{G}$ is given. The problem to be considered is how to detect a possible periodic loop in this ϱ-path.

By Proposition 6, if a periodic loop with period r is present in $\{G_t\}_{t \in I}$, then $I = \mathcal{N}$ and all G_t from some t on are cyclic elements of $\mathcal{G}$ with period r. This fact

allows one to detect loops of the considered kind by some methods proposed earlier for detection of loops with complete repetitions of the computational states. Convenient methods are, for example, R. P. Brent's method described in [8, Section 3.1, Exercise 7] and a similar method proposed in [14]. Each of them uses, so to say, temporary recording of the computational states at moments which are members of a fixed increasing sequence of natural numbers. For Brent's method, this is the sequence $0, 1, 3, 8, 15, 31, 63, \ldots$ of numbers of the form $2^n - 1$; for the other method the sequence is $0, 1, 3, 8, 21, 55, 144, \ldots$, the Fibonacci numbers with even indices. More generally, we have (compare with Lemma 2 in [14]).

THEOREM 3. *Let $\tau_0, \tau_1, \tau_2, \ldots$ be a strictly increasing infinite sequence of natural numbers such that the sequence $\{\tau_{n+1} - \tau_n\}_{n=0}^{\infty}$ is unbounded. Then the presence of a periodic loop in $\{G_t\}_{t \in I}$ is equivalent to the existence of a natural number n with the following properties*: (i) $\tau_n \in I$; (ii) *the element G_{τ_n} of $\mathcal{G}$ is cyclic with period not exceeding $\tau_{n+1} - \tau_n$.*

Proof. We only have to prove that the presence of a periodic loop in $\{G_t\}_{t \in I}$ implies the existence of a natural number n with the properties (i) and (ii). Suppose there is a periodic loop with period r in $\{G_t\}_{t \in I}$. Then $\tau_n \in I$ for all n, and all G_{τ_n} from some n on are cyclic with period r. On the other hand, there are infinitely many n such that $\tau_{n+1} - \tau_n \geq r$. Hence there is an n satisfying all three conditions just formulated: $\tau_n \in I$, G_{τ_n} is cyclic with period r, and $\tau_{n+1} - \tau_n \geq r$. ∎

The practical application of this theorem is as follows: in the process of generating the goals $G_0, G_1, G_2 \ldots$, whenever some G_{τ_n} is reached without a loop being detected, this goal is recorded temporarily, and the process of generating the goals G_t with $\tau_n < t \leq \tau_{n+1}$ is accompanied with checking whether G_{τ_n} is cyclic with period not exceeding $\tau_{n+1} - \tau_n$ (it was shown in the previous section how this checking can be done). Two examples follow, where $\tau_0 = 0$, $\tau_1 = 1$, $\tau_2 = 3$, $\tau_3 = 8$ (further τ_n are not used in these examples).

EXAMPLE 5. Let P be the Prolog program

```
r(f(X), Y) :- r (X, U), r(U, V), r(V, Y).
r(c, f(c)).
r(f(c), f(c)).
```

Let the query to be evaluated be ?- r(f(c), f(Z)) (writing this query and writing other goals further, we shall have in mind the corresponding abstract goals). Then an execution of the program leads to the following infinite sequence of goals:

```
G0   :- r(f(c), f(Z)).
G1   :- r(c, U), r(U, V), r(V, f(Z)).
G2   :- r(f(c), V), r(V, f(Z)).
```

G_3 :– r(c, U1), r(U1, V1), r(V1, V), r(V, f(Z)).
G_4 :– r(f(c), V1), r(V1, V), r(V, f(Z)).
G_5 :– r(c, U2), r(U2, V2), r(V2, V1), r(V1, V), r(V, f(Z)).
. . . . . .

To detect the loop which is present in the execution, the execution algorithm for Prolog programs has to be extended with additional activities aiming at the detection. In this example, we must first check whether G_0 is cyclic with period 1. For that purpose, when G_1 is generated, we calculate the corresponding number p_1 (the result is $p_1 = 0$), and check whether $|G_0| \leq |G_1|$ and $\pi^0(G_0) \leq G_1$. Since this is not the case, we go on. We check whether G_1 is cyclic with a period not exceeding 2. For that purpose, when G_2 is generated, we calculate the corresponding p_1 (the result is $p_1 = 2$), and check whether $|G_1| \leq |G_2|$ and $\pi^2(G_1) \leq G_2$. Again this is not the case, and we go further. When G_3 is generated, we calculate p_2 (the result is $p_2 = 1$), and check whether $|G_1| \leq |G_3|$ and $\pi^1(G_1) \leq G_3$. It turns out that this is indeed the case, and so the loop is detected by seeing that G_1 is cyclic with period 2.

EXAMPLE 6. The program will be the same as in Example 5, but the query will now be ?– r(f(f(f(c))), f(Z)). We shall now think from the very beginning about an execution combined with detection activities. To visualize things and to present them in a more compact form, we shall mark by $*$ the lines whose numbers are of the form τ_n. For the same purpose, the value of each number $p_i = p_{t-\tau_n}$, calculated after generating a G_t with $\tau_n < t \leq \tau_{n+1}$ will be represented by means of a broken line which divides the listing of each goal G_s with $\tau_n \leq s \leq t$ into two parts, the right one containing exactly p_i atomic formulas. And we shall immediately stop generating new goals when a loop is detected.

If we proceed as explained above, we get the following picture (note that the broken line representing the number p_1, calculated after generating G_2, is a part of the line representing p_2, calculated after generating G_3, due to the equality $p_1 = p_2$):

G_0 $*$:– r(f(f(f(c))), f(Z)).
G_1 $*$:– r(f(f(c)), U), r(U, V), r(V, f(Z)).
G_2 :– r(f(c), U1), r(U1, V1), r(V1, U), r(U, V), r(V, f(Z)).
G_3 $*$:– r(c, U2), r(U2, V2), r(V2, U1), r(U1, V1), r(V1, U), r(U, V), r(V,f(Z)).
G_4 :– r(f(c), V2), r(V2, U1), r(U1, V1), r(V1, U), r(U, V), r(V, f(Z)).
G_5 :– r(c, U3), r(U3, V3), r(V3, V2), r(V2, U1), r(U1, V1), r(V1, U), r(U, V), r(V,f(Z)).

Our algorithm shows that G_3 is cyclic with period 2 (since $|G_3| < |G_5|$, and the sequence r(c, U2), r(U2, V2) coincides, up to renaming variables, with r(c, U3), r(U3, V3)). So a periodic loop with period 2 has been detected.

Let us consider also an example of application of the detection method in the case where no loop is present during the depth-first search.

EXAMPLE 7. Let the query be ?– r(Y, c), and the program be

```
r(c, b).
r(a, X) :- r(X, b), r(c, X).
r(X, a) :- r(b, X), r(X, b).
r(X, X) :- r(b, X), r(X, a).
r(b, X) :- r(X, b).
```

Then the initial depth-first part of the execution process looks as follows, when accompanied with loop detection activities (the same conventions are adopted as in the previous example):

```
G0  *  :- r(Y, c).
G1  *  :- r(c, b), r(c, c).
G2     :- r(c, c).
G3  *  :- r(b, c), r(c, a).
G4     :- r(c, b), r(c, a).
G5     :- r(c, a).
G6     :- r(b, c), r(c, b).
G7     :- r(c, b), r(c, b).
G8  *  :- r(c, b).
G9     :-.
```

Here the check of the inequality $|G| \leq |\varrho^r(G)|$ eliminates all cases except comparing G_0 with G_1, G_1 with G_3, and G_3 with G_4, G_6, G_7. In these cases the inequalities $\pi^0(G_0) \leq G_1$, $\pi^0(G_1) \leq G_3$, $\pi^1(G_3) \leq G_4$, $\pi^0(G_3) \leq G_6$, $\pi^0(G_3) \leq G_7$ have to be checked, and none of them turns out to hold ([4]).

Remark 5. Since the presented loop detection method is applicable in the case of arbitrary goal spaces, Example 2 shows that periodic loops in the execution of deterministic recursive algorithms can also be detected by means of this method. An example follows.

EXAMPLE 8. In the notations of Example 2, let E consist of two labels, **A** and **B**, and M be the set of integers. Let R consist of the following four E-instructions

([4]) If we try to apply the loop detection method from [16] to the considered program execution, then we shall obtain the incorrect conclusion that a loop is present, since G_6 has the same first subgoal as G_3. When using our method, we have to check whether the whole G_3 is a beginning of G_6, and this is not the case.

over M:

$$(\mathbf{A}, 2x \mapsto 3x+1, \mathbf{BA}), \quad (\mathbf{A}, 2x+1 \mapsto x+2, \varepsilon),$$
$$(\mathbf{B}, 2x \mapsto x+1, \varepsilon), \quad (\mathbf{B}, 2x+1 \mapsto 3x+2, \mathbf{AB}),$$

where x ranges over M, and ε is the empty string. The ϱ-paths in the corresponding goal space from Example 2 can be regarded as representing computations based on the following recursive definition of partial unary functions A and B in M:

$$A(2x) := B(A(3x+1)), \quad A(2x+1) := x+2,$$
$$B(2x) := x+1, \qquad\qquad B(2x+1) := A(B(3x+2)).$$

For instance, the process of computation of $A(162)$ can be represented by means of a ϱ-path looking as follows: $(\mathbf{A}, 162)$, $(\mathbf{BA}, 244)$, $(\mathbf{BBA}, 367)$, $(\mathbf{BB}, 185)$, $(\mathbf{BAB}, 278)$, $(\mathbf{BA}, 140)$, $(\mathbf{BBA}, 211)$, $(\mathbf{BB}, 107)$, $(\mathbf{BAB}, 161)$, $(\mathbf{BAAB}, 242)$, $(\mathbf{BAA}, 122)$, $(\mathbf{BABA}, 184)$, $(\mathbf{BABBA}, 277)$, $(\mathbf{BABB}, 140)$, $(\mathbf{BAB}, 71)$, $(\mathbf{BAAB}, 107)$, $(\mathbf{BAAAB}, 161), \ldots$ Applying the method in question to this ϱ-path and using the sequence $0, 1, 3, 8, 21, \ldots$, we observe that $(\mathbf{BAB}, 161)$ is a cyclic goal and therefore the ϱ-path is infinite. Thus $A(162)$ is undefined.

Acknowledgments. The main part of the present paper was written, in essential, in June 1990 during the author's stay at the Institute of Practical Computer Science of the University in Duisburg, Federal Republic of Germany. The author is indebted to the participants of the Institute Seminar for useful comments concerning the loop detection problem. Special thanks are due to Professor Dr. H. Kleine Büning and to Dr. Th. Lettmann; the discussions with them helped the author to better feel some specific features of the above problem in the case of Prolog programs. The author is also very grateful to Professor Dr. E. Börger and to Professor Dr. W. Schönfeld who gave him much information about other investigations of the problem. The author greatly appreciates the invitation from the Banach Center in Warsaw, which offered him the possibility to present his results in a lecture and in this paper. Finally, he thanks an anonymous referee for some helpful suggestions.

References

[1] Ph. Besnard, *On infinite loops in logic programming*, Rapports de Recherche 1096, IRISA, Rennes 1989.

[2] R. N. Bol, K. R. Apt and J. W. Klop, *An analysis of loop checking mechanisms for logic programs*, technical report, Centre for Mathematics and Computer Science, Amsterdam 1989.

[3] D. R. Brough and A. Walker, *Some practical properties of logic programming interpreters*, in: Internat. Conf. on Fifth Generation Computer Systems, Institute for New Generation Computing, Tokyo 1984, 149–156.

[4] M. A. Covington, *Eliminating unwanted loops in logic programming*, SIGPLAN Notices 20 (1) (1985), 20–26.

[5] M. A. Covington, *A further note on looping in Prolog*, ibid. 20 (8) (1985), 28–31.
[6] D. de Schreye, M. Bruynooghe and K. Verschaetse, *On the existence of nonterminating queries for a restricted class of Prolog clauses*, Artificial Intelligence 41 (1989/90).
[7] H. Kleine Büning, U. Löwen and S. Schmitgen, *Loop detection in propositional Prolog programs*, in: CSL '88, 2nd Workshop on Computer Science Logic, Duisburg, October 3–7, 1988, E. Börger, H. Kleine Büning and M. M. Richter (eds.), Lecture Notes in Comput. Sci. 385, Springer, 1989, 148–165.
[8] D. Knuth, *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms*, second ed., Addison-Wesley, Reading, Mass., 1981.
[9] A. Mazurkiewicz, *Recursive algorithms and formal languages*, Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys. 20 (1972), 799–803.
[10] D. Nute, *A programming solution to certain problems with loops in Prolog*, SIGPLAN Notices 20 (8) (1985), 32–37.
[11] L. Plümer, *Termination proofs for logic programs*, dissertation, Univ. Dortmund, 1989.
[12] D. Poole and R. Goebel, *On eliminating loops in Prolog*, SIGPLAN Notices 20 (8) (1985), 38–40.
[13] A. Schmücker, *Analyse und Transformationen von Hornklausel-Programmen unter Verwendung von Templateketten*, dissertation, Univ. Kaiserslautern, 1986.
[14] D. Skordev, *An extremal problem concerning the detection of cyclic loops*, C. R. Acad. Bulgare Sci. 40 (10) (1987), 5–8.
[15] —, *On the detection of periodic loops in computational processes*, J. Symbolic Logic 57 (1992), 335–336.
[16] A. Van Gelder, *Efficient loop detection in Prolog using the tortoise-and-hare technique*, J. Logic Programming 4 (1987), 23–31.