Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Technical Memo
TM-94-02

Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations

Rainer Bleisinger, Berthold Kröll

March 1994

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum
für
Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❑ Intelligent Engineering Systems
- ❑ Intelligent User Interfaces
- ❑ Computer Linguistics
- ❑ Programming Systems
- ❑ Deduction and Multiagent Systems
- ❑ Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Friedrich J. Wendl
Director

Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations

Rainer Bleisinger, Berthold Kröll

DFKI-TM-94-02

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9003 0).

ISSN 0046-0071

Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations

Rainer Bleisinger, Berthold Kröll

German Research Center for Artificial Intelligence (DFKI)
P.O. Box 2080
D-67608 Kaiserslautern, Germany
e-mail: bleising@dfki.uni-kl.de

Abstract: For representing natural language expressions with temporal repetition the well known time interval calculus of Allen [Allen 83] is not adequat. The fundamental concept of this calculus is that of convex intervals which have no temporal gaps. However, natural language expressions like “every Summer“ or “on each Monday“ require the possibility of such temporal gaps.
Therefore, we have developed a new calculus based on non-convex intervals and have defined a set of corresponding non-convex relations. The non-convex intervals are sets of convex intervals and contain temporal gaps. The non-convex relations are tripels: a first part for specifying the intended manner of the whole relation, a second part for defining relations between subintervals, and a third part for declaring relations of whole, convexified non-convex intervals. In the non-convex calculus the convex intervals and relations of Allen are also integrated as a special case.
Additionally, we have elaborated and fully implemented a constraint propagation algorithm for the non-convex relations. In comparison with the convex case we get a more expressive calculus with same time complexity for propagation and only different by a constant factor.

Keywords: knowledge representation, qualitative reasoning, temporal reasoning

Contents:

1. Introduction

Real world events happen during time. All computer programs handle and model parts of the real world. Therefore, most of these systems, like office automation systems, project management systems or natural language systems, require a representation of time in which events happen and properties hold. In this context, qualitative (relational) as well as quantitative (numeric) temporal aspects must be considered for both temporal order and temporal duration [Faidt et al 89, Bleisinger 91]. The quantitative time representation is based on a calendaric chronological system which combines dates and clock time (see also [Ladkin 86b]).
In our approach the basic objects of time are time intervals, and the infinite time structure is linearly ordered. Also, no smallest time unit is defined, hence the time representation is close.
By modeling an event on a time interval the distinction between two kinds of intervals is important. The calculus proposed in [Allen 83] considers convex intervals and the thirteen convex relations[1] that can hold between those intervals, respectively. Convex intervals are intuitively those without temporal gaps. But what about those events, like swimming every Summer or meeting on each Monday morning, which happen in an interval containing gaps?
Principally, two different approaches are possible. On the one hand, the interval calculus of Allen is extended with special predicates for repetitive events [Becker 90], for example holds_periodic or holds_at_holiday.
On the other hand, the restriction to convex intervals is abolished. A new type of intervals containing gaps, so called non-convex intervals, is introduced. The first paper discussing this theme is [Ladkin 86a,b]. Based on calendar and clock time units an extensive set of “useful“ relations for specifying relationships between non-convex intervals is explained. Relations between non-convex intervals are called non-convex relations. Other ideas are presented in [Leban et al 86] and [Kortüm 91]. Aim of [Leban et al 86] is first of all the introduction of operators to allow an effective means for representing non-convex intervals, here called "collections" of intervals. But we think their representation of non-convex intervals to be unsuitable to further application, for example reasoning. Disadvantages of [Kortüm 91] are established in their low expression power. So all proposed techniques are not satisfactory.
The aim of this paper is the elaboration of an appropriate calculus of non-convex intervals because we need this type of more general time intervals for many purposes in AI. In Chapter 2 we present our non-convex interval calculus. After the informal introduction of non-convex intervals we define calendar based non-convex intervals. Main part of our calculus is the representation of the binary relations that can hold between two non-convex intervals. Within this part we provide examples of these relations applied to the description of tasks and events. The propagation algorithm for our non-convex relations is discussed in detail in Chapter 3. Because our non-convex relations are tripels, the propagation of each part of the tripel will be explained separately. A summary and short discussion of our non-convex interval calculus and the appropriate propagation method conclude this paper.

2. Calculus of non-convex intervals

2.1 Informal definition of non-convex intervals

The convex time intervals proposed in [Allen 83] are the starting point for the definition of non-convex time intervals. In analogy to the approach in [Ladkin 86b] non-convex intervals consist

[1]Convex relations means relations between convex intervals. In [Nökel 89] a different meaning is intended.

intuitively of some (maximal) convex subintervals with convex temporal gaps in between them. In this way non-convex time intervals are unions of convex time intervals which often occur in the real world. Generally, non-convex intervals are finite as well as infinite sets of convex intervals. Any recurring time period can be represented in this form. For example, we can regard the infinite non-convex period MONDAYS as being composed of any individual, convex Monday, or the finite non-convex interval of all WEEKENDS IN JANUARY 1992.
A graphical representation of a non-convex time interval looks like this:

This non-convex interval i has five parts, i.e. convex subintervals, which we call *consubints*. Each of those consubints describes the time of validity of a certain event.
Now we have to elaborate a formal specification of non-convex intervals motivated by natural language statements. Analogous to the treatment of quantitative and qualitative aspects to describe attributes and relationships of convex time intervals 8Bleisinger 919, similar information may be described for non-convex time intervals.
So the sentences “every two days in summer“ or “five times a week“ are quantitative descriptions of recurring time periods and thereby of non-convex intervals. To represent quantitative information we use calendar based expressions which are introduced in Section 2.2.
The statements “always reading the newspaper during the breakfast“ or “mostly taking a shower after jogging“ are examples of qualitative information concerning a non-convex interval. Information of this kind sets two intervals in a non-convex relation. As we will see temporal adverbs like “always“, “mostly“ etc. will have a special bearing by the development of non-convex relations that can hold between non-convex intervals. In this paper qualitative information of non-convex intervals is more important and will be discussed in the Section 2.3.

2.2 Calendar based non-convex time intervals

For reasoning about years, months, days, minutes etc. we have to develop a possibility to express quantitative aspects in a formal manner. Here we introduce a standard form for an interval which represents an instance of one of the mentioned calendaric chronological units. All the basic units will be convex intervals. Afterwards, we show how to develop non-convex intervals as unions of these standard convex intervals.

To represent standard convex intervals we use sequences of integers. For example, 81991, 12, 13, 79 denotes the hour starting at 7 am on December 13th, 1991. It is obvious how to extend the hierarchy to smaller standard intervals. Certain relationships are essential between these standard intervals which are formulated by a couple of axioms (see 8Ladkin 86b9).

Now we are able to define several standard non-convex intervals as unions of standard convex intervals. In the following a, b, g, denote standard convex intervals. The function `length` counts the number of positions contained in the sequence denoting a standard convex interval.

Between standard convex intervals essential relationships are formulated by a couple of axioms (see also 8Ladkin 86b9). For example, a standard convex interval a meets exactly one b and is met-by exactly one g of the same length (a = 1991, b = 1992, g = 1990).

Now we define several standard non-convex intervals as unions of standard convex intervals.

- MONTHS = {a | `length` (a) = 2},
- DAYS = {a | length (a) = 3} etc.

Additionally, we may also define arbitrary non-convex intervals only loosely coupled with a calendar. Therefore, we define the operator to iterate the meets relations:

Definition 1: (iterated meets-relations)
Let a, b and g be standard convex intervals of the same length and i an integer. The iterated meets-relations are then defined by:

- f_0 (a, b) ° (a *meets* b)
- f_{i+1} (a, b) ° (æ g) (f_i (a, g) å (g *meets* b)) for $0 \leq i$
- f* is the symmetric, transitive closure of f for any binary relation f

The f_i are the iterated meets-relations for standard intervals of a special length.

Note that, as we have defined them, a given a of a standard interval meets exactly one b and is met-by exactly one g of the same standard interval.

To define arbitrary non-convex intervals we make use of the iterated meets-relations. For the next example we suppose a ranging over the set DAYS.

• MONDAYS = {a | $(f_6)^*$ ([1992, 1, 13], a)} with: [1992, 1, 13] is a Monday.

Besides those quantitative aspects describing non-convex intervals we have to reflect upon the possibilities how to describe non-convex intervals by qualitative aspects. This will be done by non-convex relations.

2.3 Definition of non-convex relations

In the convex case qualitative statements are realized by the specification of the well known thirteen convex relations [Allen 83]. In parenthesis the same abbreviations as in [Allen 83] are listed for these relations. Therefore, the set of convex relations kR is defined as following:

Definition 2: (set of convex relations, **kR**)

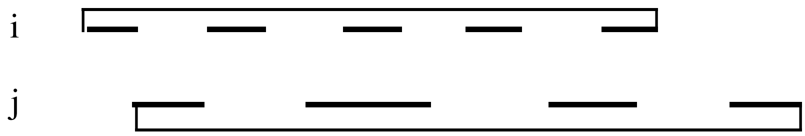
kR = {before (<), meets (m), overlaps (o), starts (s), during (d), finishes (f), equal (=), after (>), met-by (mi), overlapped-by (oi), started-by (si), contains (di), finished-by (fi)}.

Here we investigate binary relations that can hold between non-convex intervals. Those relations will be called *non-convex relations*. In 8Lad86a9 a theorem states that the number of relations between non-convex intervals is at least exponential in the number of consubints. That means an exhaustive enumeration of non-convex relations is infeasible. To avoid the combinatorial explosion implied by the theorem Ladkin chose some basic relations that don´t depend on the number of consubints. We didn´t fully accept those relations but refine the idea of generalized convex relations (see 8Lad86a9). At a first view our non-convex relations show certain similarities to the generalized relations arising from [Kortüm 91], but note that Kortüms relations in fact are only a subset of our relations.

In our calculus a non-convex relation is composed of three parts:

- a functor whose name acts as key for the intended manner of the whole relation
- a first argument which specifies the convex relation between consubints of the first interval and consubints of the second interval
- a second argument which specifies the convex relation between the convex conclusions of the participating intervals.

To illustrate the second argument consider the following non-convex intervals i and j. The convex conclusion of an interval is just the smallest convex interval that covers all consubints of the original non-convex interval. In case of the intervals i and j the list *(overlaps)* specifies the relation between the convex conclusions of i and j.



The convex conclusion will be realized by the interval operator `convexify` which takes the first and the last consubint of a non-convex interval as its argument and returns the convex conclusion.

In the following, we introduce the formal notation of a non-convex relation in form of a tripel, but first we give the definition of the set of all functors.

Definition 3: (set of functors, **func**)
The set of all functors is specified by:

func = {1:1, 01:1, 1:01, n:1, 1:n, n<m, n=m, n>m, 01:1-f, 1:01-s, n:1-s, 1:n-f, 1:1-b}.

Definition 4: (set of non-convex relations, **R**)
A non-convex relation is noted in form of a tripel <*functor list1 list2*>. The set of all non-convex relation is defined as:

R = ˙ {<*functor list1 list2*> | *functor* ™***func***; *list1, list2* ∑ **kR**}.

A disjunctive collection of tripels is realized by gathering all participating tripels in a list. In accordance with disjunctions of convex relations this disjunction should be understood as XOR.

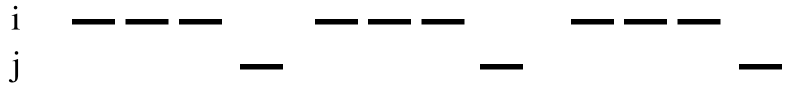
In the following we explain the different functors with the help of several examples of these relations applied to the description of tasks and events. Thereby, the set of functors is treated in three groups. At first, we discuss functors for two non-convex intervals.

1:1: i *<1:1 list1 list2>* j
The intervals i and j contain the same number of consubints and between each matched pair of consubints one of the realitions of *list1* is valid. The functor *1:1* is taken from the adverb *always*.
E.g. “always during the breakfast read the newspaper“ — i *<1:1 (contains) (contains)* > j

1:01: i *<1:01 list1 list2>* j
For every consubint of j exists one consubint of i so that one of the relations of *list1* holds. This allows the possibility that there are other consubints of i, but not of j. The functor *1:01* is taken from the adverb *mostly*.
E.g. “mostly after jogging take a shower“ — i *<1:01 (after) (overlaps)>* j

n:1: i *<n:1 list1 list2>* j

For every consubint of j, there is a group of consubints of i that is related to it in one of the elements of *list1*. Note, the conversion is not valid: you can´t gather arbitrary consubints of i and expect these groups to be in a well defined relation to consubints of j.
E.g. “after some hours of learning take an hour to relax“ — i *<n:1 (after)(overlaps)>* j



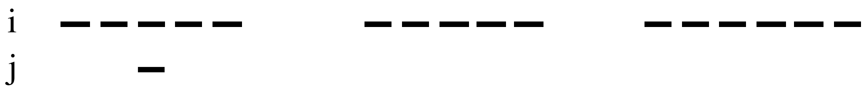
n<m, n=m, n>m: i *<n<m list1 list2>* j
For every chosen group of consubints of j, there is a group of consubints of i that is related to it in one of the elements of *list1*. The number of consubints of groups from interval i is always less than the number of consubints of the matched groups from interval j.
In analogy to this definition the functors *n=m* and *n>m* are defined; only the number relationship of consubints of corresponding groups is different.
E.g “he enjoys his holidays always just the last few days“ — i *<n>m (finishes) (finishes)>* j

Note that *n>m* and *n<m* are converse functors. The converse functors of *1:01* and *n:1* are ***01:1*** resp. ***1:n***. They are not enumerated here, but their semantics should be clear.

Besides the presented functors we have defined a group of functors for the specification of relations that can hold between a non-convex and a convex interval.

1:01-s: i *<1:01-s list1 list2>* j
One of the consubints of i is related to the convex interval j in one of the elements of *list1*. The postfix *-s* (“second“) designates the second interval - here interval j - to be the convex one.
E.g. “I became familiar with her the last day of our holiday“ — i*<1:01-s (equal) (finishes)>* j

n:1-s: i *<n:1-s list1 list2>* j
A chosen group of consubints of i is related to the convex interval j in one of the elements of *list1*.
E.g.“she failed once in a test during the first examination day“ — i*<n:1-s (during) (during)>*j



Additionally there are the functors ***01:1-f*** and ***1:n-f*** as converse functors. Here the postfix *-f* (“first“) designates the first interval to be the convex one.

The notation of non-convex relations in form of tripels allows the representation of pure convex relations in a suitable manner: if *list* is a disjunction of convex relations between two convex intervals, those relations can be transformed to the equivalent tripel notation *<1:1-b list list>*. The positions 2 and 3 of the tripel are both *list* and the new defined functor is called ***1:1-b***. The postfix *-b* (“both“ intervals are convex) distinguishes this functor from the pure non-convex functor *1:1*.

Why did we choose those functors and why did we determine the tripelform as presentation of non-convex relations? The advantages are evident:

- three positions of a tripel imply a high expressive power
- relations between arbitrary types of intervals (convex/non-convex) can be expressed by one form (our tripelform)
- the relations are disjunct
- they don´t depend on the number of consubints of their argument intervals
- the set of all possible relations is exclusive concerning the product of relations (see Section 3.1.4)
- our functors allow a natural presentation of relationships between intervals

Finally another remark concerning the presentation of non-convex intervals within our calculus: After that we have the possibility to define non-convex intervals by sets we may furthermore use ordinary setoperations to construct and modify non-convex intervals. Besides we can combine formal quantitative and qualitative descriptions to define new intervals as shown in the examples below.

- *beginning in march 1991 every three days* = {a | $(f_2)^*$ ([1991,3,1], a) å a ≥ [1991,3,1]}
- *three times everyday* = FIRST · SECOND · THIRD å
(FIRST <*1:1 (contained-in) (contained-in)*> DAYS) å
(SECOND <*1:1 (contained-in) (contained-in)*> DAYS) å
(THIRD <*1:1 (contained-in) (contained-in)*> DAYS) å
(FIRST <*1:1 (before) (overlaps)*> SECOND) å (SECOND <*1:1 (before) (overlaps)*> THIRD)

Now, an expressive formal calculus of non-convex intervals is at our disposal. Thereby, the presented non-convex relations may act as the basis of a relation algebra that is going to be developed within the next chapter.

3. Propagation of non-convex relations

In 8Allen 839 an algorithm to propagate convex relations within a time interval network is presented. Allen´s algorithm is motivated by the following question: if xRy and ySz, where x, y and z are convex intervals and R, S are convex relations, how is the relation p(R, S) to be determined? An essential part of his algorithm was the introduction of the two operations product and intersection on relation sets. The elaboration of our non-convex calculus as extension of Allen´s convex calculus avoids the developement of an entire different algorithm.
In this section we redesign Allen´s algorithm to an appropriate algorithm for non-convex relations. That means, we have to redefine the mentioned operations, product and intersection, on sets of non-convex relations.

3.1 Product of relations as tripel product

The axiomatization of the relation product in the convex case, p(R, S), R, S ™ **kR**, is given by a relation product table ([Allen 83]). Because of the enormous amout of different tripels a similar proceeding in the non-convex case will be complicated. Nevertheless, the representation of non-convex relations in tripelform offers the way of a separate product building about every position of two tripels. This can be done with regard to certain dependencies of the three positions. That is, we have to construct products p1, p2 and p3 over the first, the second resp. the third positions of each two tripels. In a final step we combine the results to receive the tripel product.

3.1.1 The product p1 of two functors

The product of p1 is independent of the other positions of the participating tripels. So we can axiomatisize p1 by a functor product table. Instead of introducing this table (for details see 8Kröll 919) we point out some fundamental principles that are pursuited in the table.

• Entries in the table always consist of subsets of **func**.

• Certain entries consist of the empty set which implies an inconsistency, e.g., p1(1:1, 01:1-f) = {}. *1:1* implies the first two intervals to be non-convex whereas *01:1-f* requires the second interval to be convex. This is a contradiction.

• All entries consist of a maximal set of valid functors. This technique guarantees that no inconsistencies will be inferred where no inconsistencies are. By it the similar semantics of certain functors leads to a grouping of functors. The groups are {1:1, n=m} , {1:01, n:1, n>m} , {01:1, 1:n, n<m} , {1:01-s, n:1-s} , {01:1-f, 1:n-f} , {1:1-b}. Pay attention to the similarity of *1:01* and *n:1* for example. The entries in the table (besides the empty set) consist always of a whole group or of unions of groups. Examples are p1(1:1, 1:01) = {1:01, n:1, n>m}, and p1(1:01, 01:1) = {1:1, 1:01, 01:1, n:1, 1:n, n<m, n=m, n>m}.

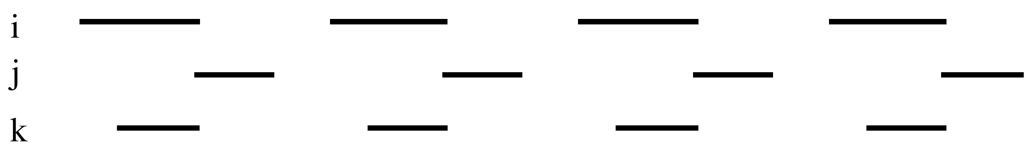
• The consideration of the types of the participating intervals (convex/non-convex) is essential to the construction of p1. So *01:1-f* relates a convex and a non-convex interval, *1:01-s* relates a non-convex and a convex interval, and the product p1(01:1-f, 1:01-s) inferes only the functor *1:1-b* which is defined between two convex intervals.

The disjunction of several functors will be realized later by the disjunction of whole tripels. But we still have to define the products p2 and p3 to get there.

3.1.2 The product p3 of the third positions of tripels

The third position of a tripel specifies the convex relations between two intervals after application of the `convexify`-operator. This operator transforms the set of consubints of a non-convex interval to a convex interval. This offers the application of Allen´s product table to axiomatisize the product p3. But, we have to regard dependencies from the result of the product p2 of the second positions. Consulting Allen´s product table with the example

i <1:1 (meets) (overlaps)> j and j <1:1 (met-by) (overlapped_by)> k



we receive the disjunction {o, oi, =, s, d, f, fi, di, si}. In fact, only a subset of this disjunction is valid. This subset is identical to the disjunction {f, fi, =}received as result of p2.

It is difficult to formulate rules to represent dependencies of this type. Besides these dependencies we have to regard dependencies of p2 and p3 in the other direction too. That is we have cyclical dependencies which are very difficult to handle. Our realization of p3 abandons those dependencies. For practical applications of the algorithm this proceeding is all right: the precise valid relationset is always just a subset of the inferred relation set. Though the result of p3 is still correct it is not guaranteed that the local consistency check detects all inconsistencies.

3.1.3 The product p2 of the second positions of tripels

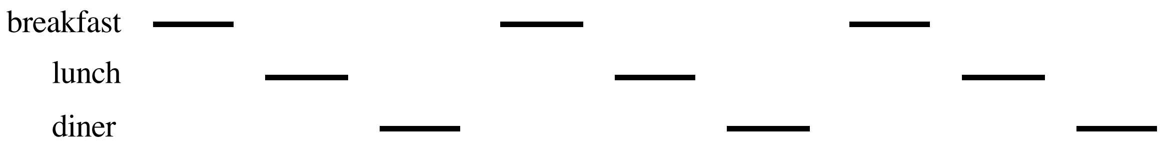
The second position of a tripel consists of a list of convex relations which relates according to the valid functor single consubints or groups of consubints of the participating intervals. This dependency is assigned to the product p2 which will be primary dependent of

- the semantics of the functors of the product building tripels

and • the semantics of the functors of the resulting tripels.

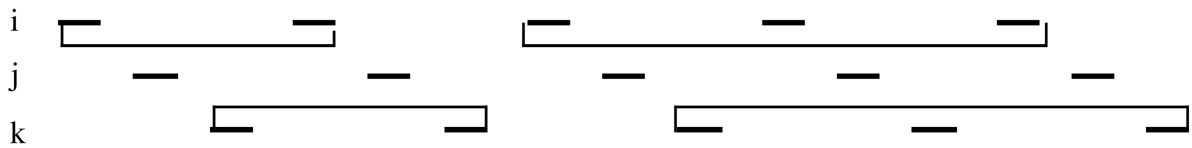
The trick how to take this primary dependency in consideration is the definition of several different tables to axiomatisize p2[2]. Because of the complexity of p2 (i.e. use of different product tables, when to consult a special table etc.) we will confine here the representing of essential results. We will make use of some examples to determine possible values of p2:

• Define the interval of the daily lunch through lunch <1:1 (after) (overlapped-by)> breakfast and lunch <1:1 (before) (overlaps)> diner. Further let diner <1:1 (after) (overlapped-by)> lunch and diner <1:1 (before) (overlaps)> breakfast.



Propagating those relations leads to the inconsistency that lunch is as well *before* as *after* breakfast. We took this case into acount by introducing a new "convex" relation named *no_rel*. The pseudo-relation *no_rel* avoids the statement of a not existing inconsistency of this kind. So if the second position of the resulting tripel consists of the disjunction *(before no_rel) no_rel* is always to be understood as additional possible relation. Therefore, *no_rel* will be listed in some entries of Allen´s product table besides the original entries.

• We will use the example above to show another resulting tripel. p1(1:1, 1:1) not only leads to the functor 1:1 but also to the functor n=m (see formation of groups of functors by p1). So if the resulting tripel owns the functor n=m, what about the second position of this tripel? In this special case the second positions of the product building tripels are irrelevant. The result of p2 is identical with the result of p3.

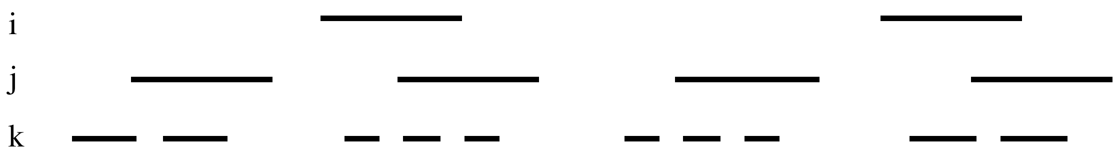


Summed up we get p(<1:1 (before) (overlaps)>, <1:1 (before) (overlaps)>) = (<1:1 (before no_rel) (before overlaps meets)> <n=m (before overlaps meets) (before overlaps meets)>) where p designates the tripel product.

• Sometimes it is impossible to infer a restricted set of relations in a resulting second position. E.g. i (*<01:1 (overlaps) (during)*>) j and j (*<1:n (overlapped-by) (overlapped-by)*>) k

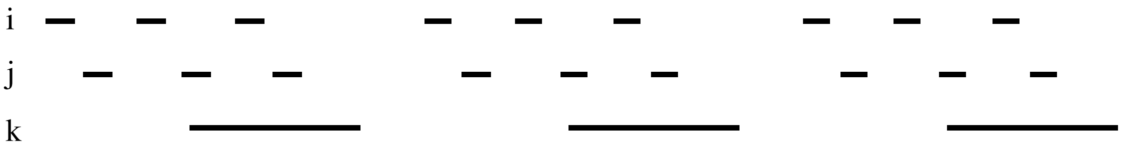
---

[2]The dependencies between p2 and p3 mentioned in the last passage will not be considered here again. In contrary to the primary dependencies whose discussion will be quite extensive they are called secundary dependencies.



The *01:1*-functor implies a lack of information about the pairing of structures of the intervals i and k. We know nothing about the site of consubints of i nor of k. In this case the second positions of resulting tripels consist of the constant kR+nr, kR+nr = **kR** ˙ {no_rel}.

• In some cases another modification of the product table for convex relations is consulted. Again an example shall show the necessity of this table.
E.g. i (<*1:1 (before) (overlaps)*>) j and j (<*n:1 (overlaps) (overlaps)*>) k



We are interested in the second position of the resulting n:1-tripel. Consulting the ordinary product table the entry of the second position would be the incomplete relation *(before)*. In fact, this position is filled with the relationset *(before meets overlaps no_rel)*. Just this entry is to be found in the new defined product table.

3.1.4 The tripel product p

The tripel product p is finally realized by a 13*13 table whose arguments are each two tripels and whose entries are disjunctions of resulting tripels. Those resulting tripels are composed of the three positions that are at their part constructed by p1, p2 and p3.
We will conclude this passage by some annotations. A major principle when defining the tripel product p was the regard on certain logical aspects. I.e. we tried to realize p in a way adequate to the human way of thinking. A typical feature of questions of AI is the problem of modeling cognitive processes and, by it, the impossibility to place an exact mathematical buildung at disposal. So especially p2 lacks an exact, or "strong", logic.

3.2 Intersection of relations

Let C and D be disjunctions of tripels relating two intervals i and j. Intersection of C and D reduces the number of valid tripels between i and j. If the result of the intersection is the empty set no relations between the participating intervals i and j are valid.
Intersection of disjunctions of tripels is more complicated as the intersection in the convex case. We have to define two additional intersection operators œ$_{TK}$ and œ$_{T}$. œ$_{TK}$ intersects two disjunctions of tripels by constructing the union of the intersection œ$_{T}$ of each two tripels whose functors are equal. œ$_{T}$ on his side intersects each two tripels with the same functor. The intersection returns an empty set, that means an inconsistency, if

- either œ$_{TK}$ returns an empty set

or
- the intersection œ of the relation lists of the second or the third position of the two intersecting tripels is empty.

4. Conclusion

In this paper a new non-convex time interval calculus is introduced. The non-convex intervals make it possible to represent repetitive natural language expressions. For the specification of relationships between non-convex intervals appropriate non-convex relations in form of tripels are defined. Each position of the relation focus on a special detail of the relationship between non-convex intervals.
In this way we reach a powerfull calculus which subsumes the convex time interval calculus proposed by Allen. Moreover, we have extended the constraint propagation algorithm introduced by Allen for handling with non-convex relations. Thereby, the tripel form is used for separate constraint propagation of the particular parts.
Another word concerning the complexity of our algorithm. In the O-notation it acts with a complexity of $O(n^3)$ and therewith in polynomial time, like its convex equivalent. In contrast, the constant factor in our algorithm is much higher.
The developed non-convex propagation algorithm is fully implemented on a SUN SparcStation. Today, only a few examples are tested. So, the useability in “real domains“ has to be shown in future.

References

[Allen 83] Allen J. F.: *Maintaining Knowledge about Temporal Intervals*, Comm. ACM., 26 (11), November 1983, pp 832-843.

[Becker 90] Becker, W.: *Time Reasoning Component in Prolog* (in German), Bachelor Thesis, University Stuttgart, March 1990.

8Bleisinger 919 Bleisinger R.: *TEMPO - an Integrative Approach for Modeling Qualitative and Quantitative Temporal Information* (in German), Proceedings der 15. Fachtagung für Künstliche Intelligenz, GWAI-91, Bonn, September 1991, pp 167-176.

[Faidt et al 89] Faidt K., Flohr S., Bleisinger R.: *Representation and Processing of Temporal Knowledge* (in German), Proceeding der 5. Österreichischen Artificial-Intelligence-Tagung, ÖGAI-89, Igls/Tirol, March 1989, pp 303-312.

[Kortüm 91] Kortüm G.: *Temporal Reasoning with Generalized Relations*, Proc. der 15. Fachtagung für Künstliche Intelligenz, GWAI-91, Bonn, September 1991, pp 177-181.

[Kröll 92] Kröll B.: *Treatment of Repetetive Temporal Information* (in German), Master Thesis, University Kaiserslautern, January 1992.

[Ladkin 86a] Ladkin P. B.: *Time Representation: A Taxonomy of Interval Relations*, Proceedings of the 5th AAAI, Philadelphia, 1986, pp 360-366.

[Ladkin 86b] Ladkin P. B.: *Primitives and Units for Time Specification*, Proceedings of the 5th AAAI, Philadelphia, 1986, pp 354-359.

[Leban et al 86] Leban, B., McDonald D.D., Forster D.R.: *A Representation for Collections of Temporal Intervals*, Proceedings of the 5th AAAI, Philadelphia, 1986, pp 367-371.

[Nökel 89] Nökel, K.: *Convex Relations between Time Intervals*, Proc. der 5. Österreichischen Artificial-Intelligence-Tagung, ÖGAI-89, Igls/Tirol, March 1989, pp 298-302.

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-93-10
Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11
Bernhard Nebel, Hans-Jürgen Bürckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12
Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13
Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14
Joachim Niehren, Andreas Podelski, Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-30
Stephen P. Spackman, Elizabeth A. Hinkelman:
Corporate Agents
14 pages

RR-93-31
Elizabeth A. Hinkelman, Stephen P. Spackman:
Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

RR-93-32
David R. Traum, Elizabeth A. Hinkelman:
Conversation Acts in Task-Oriented Spoken Dialogue
28 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.):
Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-43
M. Bauer, G. Paul: Logic-based Plan Recognition for Intelligent Help Systems
15 pages

RR-93-44
Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

RR-93-46
Philipp Hanschke: A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

RR-93-48
Franz Baader, Martin Buchheit, Bernhard Hollunder:
Cardinality Restrictions on Concepts
20 pages

RR-94-01
Elisabeth André, Thomas Rist:
Multimedia Presentations:
The Support of Passive and Active Viewing
15 pages

RR-94-02
Elisabeth André, Thomas Rist:
Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

RR-94-03
Gert Smolka:
A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
34 pages

RR-94-05
Franz Schmalhofer, J.Stuart Aitken, Lyle E. Bourne jr.:
Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
81 pages

RR-94-06
Dietmar Dengler:
An Adaptive Deductive Planning System
17 pages

RR-94-07
Harold Boley: Finite Domains and Exclusions as First-Class Citizens
25 pages

RR-94-08
Otto Kühn, Björn Höfling: Conserving Corporate Knowledge for Crankshaft Design
17 pages

RR-94-10
Knut Hinkelmann, Helge Hintze:
Computing Cost Estimates for Proof Strategies
22 pages

RR-94-11
Knut Hinkelmann: A Consequence Finding Approach for Feature Recognition in CAPP
18 pages

RR-94-12
Hubert Comon, Ralf Treinen:
Ordering Constraints on Trees
34 pages

RR-94-13
Jana Koehler: Planning from Second Principles —A Logic-based Approach
49 pages

RR-94-14
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Towards a Sharable Knowledge Base on Recyclable Plastics
14 pages

RR-94-15
Winfried H. Graf, Stefan Neurohr: Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
20 pages

RR-94-16
Gert Smolka: A Foundation for Higher-order Concurrent Constraint Programming
26 pages

DFKI Technical Memos

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

TM-93-04
Hans-Günther Hein:
Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

TM-93-05
Michael Sintek: Indexing PROLOG Procedures into DAGs by Heuristic Classification
64 pages

TM-94-01
Rainer Bleisinger, Klaus-Peter Gores:
Text Skimming as a Part in Paper Document Understanding
14 pages

TM-94-02
Rainer Bleisinger, Berthold Kröll:
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

DFKI Documents

D-93-10
Elizabeth Hinkelman, Markus Vonerden, Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11
Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12
Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-15
Robert Laux:
Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-16
Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt: Design & KI
74 Seiten

D-93-20
Bernhard Herbig:
Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21
Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten

D-93-22
Andreas Abecker:
Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

D-93-24
Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25
Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-26
Frank Peters: Unterstützung des Experten bei der Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

D-93-27
Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman, Hans Uszkoreit (Eds.):
Report of theEAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-94-01
Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
75 pages

D-94-02
Markus Steffens: Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

D-94-03
Franz Schmalhofer: Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

D-94-04
Franz Schmalhofer, Ludger van Elst:
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution
22 pages

D-94-06
Ulrich Buhrmann:
Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 pages

D-94-08
Harald Feibel: IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

D-94-07
Claudia Wenzel, Rainer Hoch:
Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten