From Binary Temporal Relations to Non-Binary Ones and Back ⋆

Steffen Staab

Institute for Applied Informatics and Formal Description Methods (AIFB), Karlsruhe University, 76227 Karlsruhe, Germany

`http://www.aifb.uni-karlsruhe.de/~sst`

`staab@aifb.uni-karlsruhe.de`

Abstract

In this paper a new approach towards temporal reasoning is presented that scales up from the temporal relations commonly used in Allen's qualitative interval calculus and in quantitative temporal constraint satisfaction problems to include interval relations with distances, temporal rules and other non-binary relations into the reasoning scheme. For this purpose, we generalize well-known methods for constraint propagation, determination of consistency and computation of the minimal network from simpler schemes that only allow for binary relations. Thereby, we find that levels of granularity play a major role for applying these techniques in our more expressive framework. Indeed, the technical preliminaries we provide are especially apt to investigate the switching between different granularities of representation, hence illucitating and exploiting the tradeoff between expressiveness and efficiency of temporal reasoning schemes on the one side and between expressiveness and understandability on the other side.

1 INTRODUCTION

Expressive temporal reasoning is much sought after in a multitude of applications, such as natural language understanding, planning or temporal databases. Common approaches to qualitative reasoning [2,45] and quantitative reasoning [9] as well as integrations of them [23,27,3] have fulfilled the requirements for temporal reasoning at different levels of granularity, still major problems remain to be tackled when one wants to offer a toolbox of comprehensive, flexible and tractable temporal reasoning mechanisms for such broad

⋆ Submitted 9/1999; Revised 12/2000; Parts of this paper have been published in [39] and in [42].

ranges of applications. For such purposes, we here investigate a framework that is more expressive than either of the mechanisms cited above, since it allows the inclusion of non-binary temporal relations. [1]

As principal scheme we propose a network of relations where relations consist of disjunctions of conjoined constraints and constraints come in the form $p = (a, q, b)$ with a, b being time points, q being an interval and the constraint denoting $b - a \in q$. In contrast to previous approaches, this scheme allows for interval relations augmented by distances like "*interval A is clearly disjoint from interval B*", for temporal rules like "*if point a before point b then point c before point d*", or for non-binary relations such as "*interval A is between intervals B and C*" (cf. Section 3 for an informal and Section 4 for a formal description).

The generalization has a large overlap with existing temporal constraint networks. Nevertheless, the common algebraic operators and relators for intersection, composition, and subsumption that are needed for determining consistency do not straightforwardly carry over to generalizations. In fact, we investigate how three major factors underlying our generalization, *viz. interval structures*, *relation topology*, and *network topology*, affect implications that typically hold in simpler schemes and, thus, influence the reasoning process. In particular, we find that the notion of *path consistency* does not carry over to our model, but a slightly weaker version, which we call *weakly generalized path consistency* (WGPC), is reached by constraint propagation and may be used as a basis for *determining consistency*. Though we must face the fundamental trade-off between expressiveness and efficiency, our approach achieves a smooth scale-up from previous mechanisms. The reason is that on those problems that could be handled by previous models the constraint propagation mechanism has the same order of computational complexity. If constructs are added which are only possible in our extension the complexity of the constraint propagation increases only smoothly (Section 5).

Besides of the determination of consistency the computation of the *minimal network* is the second major task one must face in temporal reasoning schemes. We find that "easier" frameworks build on basic assumptions concerning the definition of minimality that do not transfer to non-binary relations. Hence, we propose a generalized notion for minimal networks that makes the *level of granularity* explicit for which minimality is computed (Section 6).

Concerning the flexibility and efficiency requirements mentioned above, our framework exhibits another advantage besides its expressiveness. Our reason-

[1] Meiri in [27], p. 377: "Future research should enrich the representation language to facilitate modeling of more involved reasoning tasks. In particular, non-binary constraints (for example, 'If John leaves home before 7:15 a.m. he arrives at work before Fred') should be incorporated in my model."

ing scheme is especially well suited to switch smoothly between different levels of granularity. We research how this switching affects reasoning and representation. Whereas the principal tradeoff between expressiveness and efficiency is well-known, we may add here to its deeper understanding, when we touch on the expressive side of temporal reasoning (Section 7). Furthermore, we look at the balance between expressiveness and understandability — hardly ever considered for temporal reasoning mechanisms so far —, even though this dimension may become a decisive factor for whether expressive temporal reasoning technology finds its way outside of AI laboratories (cf. McGuiness et al. [26]; cf. Section 8).

Before we start with a summary of the fundamental reasoning mechanisms we build on (Section 2), we just want to mention that the proposal we make here has been part of a larger effort toward understanding and reasoning with natural language degree expressions [40].

2 TCSP NETWORKS & ALLEN'S CALCULUS

To facilitate understanding and to distinguish the features gained through this proposal, we introduce the basic concepts from which the GTN model is generalized. In particular, we give a short survey of Simple Temporal Problems (STPs; cf. Dechter et al. [9]), their generalization in form of TCSPs [9], and the integration (cf. Meiri [27]) of TCSPs with Allen's Calculus (cf. Allen [2]).

The data structures underlying all of these approaches are graphs the vertices of which are time point or time interval variables and the edges of which are annotated with relations. [2] In general, the goal is to determine *consistency* of the network and to compute the *minimal network* equivalent to the given one. Consistency is usually computed by enforcing path consistency on networks with convex relations (cf. Montanari [28]), e.g., singleton labellings, which can be enumerated with backtracking. Path consistency is enforced by repeatedly intersecting ("$\cap$" for intersection) known relations with restrictions computed from the pairwise composition of relations ("$\circ$" for composition). Often it turns out that the computation of the minimal network can be stated in terms of computing consistency.

An STP network is given by a set of time point variables V and a single interval constraint $q_{i,j}$ between each pair of these variables [9]. "$\circ$" and "$\cap$" are given by the addition and intersection of intervals on the real line, respectively. For instance, one may denote that time point t_1 is between 10 and 20 units

[2] Throughout this paper we will assume that constraints are simple interval constraints between time points, while relations may group several constraints.

earlier than time point t_2, which itself is between 20 and 30 units earlier than t_3. By computing path consistency one can determine consistency, and, in this example, one may conclude that t_1 is between 30 and 50 units earlier than t_3.

A TCSP network has a similar structure, but allows for disjunctions of interval constraints between points [9]. For instance (cf. PP relations in Fig. 1), if t_1 is between 10 to 20 units *or* between 110 to 120 units earlier than t_2, and t_2 is between 20 to 30 units earlier than t_3, then one may conclude that either t_1 is between 30 to 50 units earlier than t_3 *or* t_1 is between 130 to 150 units earlier than t_3. "$\circ$" is given by the pairwise application of interval addition and the union of the results. "$\cap$" is the set intersection.

Allen's calculus considers disjunctions of 13 primitive and mutually exclusive relations between intervals (cf. the II relations in Fig. 1). For instance, from *"interval A before or overlaps interval B"* and *"B overlaps interval C"* follows *"A before or meets or overlaps C"*. For primitive relations "$\circ$" is given by a composition table, for disjunctions the union is taken over all the results of the pairwise composition of primitive relations. "$\cap$" is defined by the intersection of sets of primitive interval relations.

The integration of TCSPs and Allen's calculus (cf. Fig. 1) requires the full mechanism for TCSP reasoning as well as that for Allen's calculus (cf. [23,27]). Furthermore, Meiri [27] provides an intermediate layer between these two subnetworks for point-interval and interval-point relations (PI in Fig. 1) which communicates between the PP and the II levels. Depending on the types of relations that are composed (PP – PP, PP – PI, PI – II, II – II) the corresponding composition and intersection operator is chosen. For instance, if time point t_2 is 20 to 30 units earlier than time point t_3 and t_3 is before ("<") or starts ("s") interval A (cf. Fig. 1), then t_2 is before A.

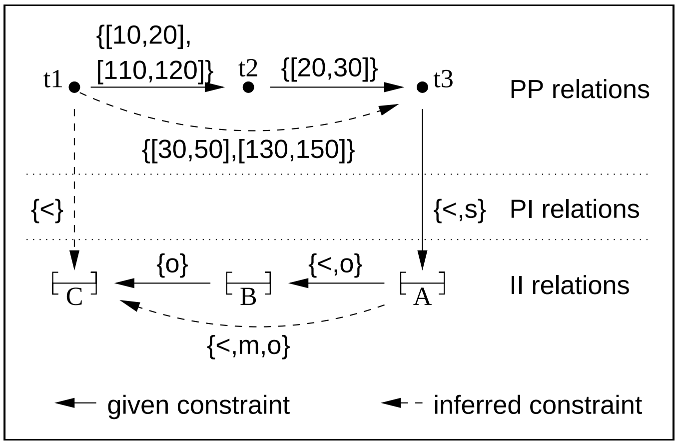


Fig. 1. Integrating TCSPs with Allen's Calculus

3 NON–BINARY RELATIONS

Notwithstanding its benefits, the integration given by Meiri [27] does not scale up to more complex temporal problems, e.g., temporal rules like *"if a before b then c before d"*, non-binary constraints like *"interval A between intervals B and C"*, or the integration of numbers into interval relations like in *"disjoint by more than n units"*. An example we want to cover and that cannot be captured by these mechanisms is:

(1) James is a shuttle driver for a major hotel in New York. His duties include coaching guests from the airports or the train stations to the hotel. Today's schedule posts Mr. Roget and Mr. Meyer from Paris, Mrs. Meyer from Philadelphia, and Mr. George from Sidney for transportation. The hotel's clerk told him that Mr. Roget and Mr. Meyer have tickets for different flights from Paris to NY. Mr. Roget is scheduled to arrive in NY at 3:00pm local time, and Mr. Meyer should arrive in NY two hours later. However, they currently try to arrange for sharing a flight which would arrive in NY at 6:00pm local time. When Mr. Meyer arrives in NY he will immediately call his wife, Mrs. Meyer, who will get the next train to NY. Hence, she will be in NY less than 4 hours after her husband has arrived. Furthermore, Mr. George's flight leaves Sidney at 12:00pm NY time, and he has got a very long flight.
Problem: In which order must James service the guests?

Let us reconsider the relations exemplified in Fig. 1. They are expressed in the integration model of TCSPs and Allen's calculus, but can also be denoted in terms of constraints on time points only. PP relations are disjunctions of single constraints between time points. Assume that $(t_1, [10, 20], t_2)$ denotes a constraint between the time points t_1 and t_2 such that $t_2 - t_1 \in [10, 20]$ then one may write the PP relation between t_1 and t_2 as follows: [3]

$$(t1, [10, 20], t_2) \vee (t1, [110, 120], t_2).$$

PI relations affect three time points. The PI relation given in Fig. 1 can be denoted by one basic assumption (cf. the illustration in Fig. 2 that "zooms" into Fig. 1), namely that the beginning time point of the interval is before the ending one (expressed by $(A_b, (0, +\infty), A_e)$), and by a disjunction of two conjoined underlying constraints (cf. "relation1" in Fig. 2). The disjunction denotes

$$((t_3, (0, +\infty), A_b) \wedge (t_3, (0, +\infty), A_e)) \vee ((t_3, [0, 0], A_b) \wedge (t_3, (0, +\infty), A_e))$$

and can be reduced to

$$(t_3, [0, +\infty), A_b) \wedge (t_3, (0, +\infty), A_e).$$

Moreover, PI relations only come with ordinal constraints, namely $(-\infty, 0)$, $[0, 0]$, or $(0, +\infty)$. Thus, the difference to PP relations is the number of edges be-

[3] "$[d, e]$" denote closed, "$[d, e)$","$(d, e]$" semi-open, and "(d, e)" open intervals.

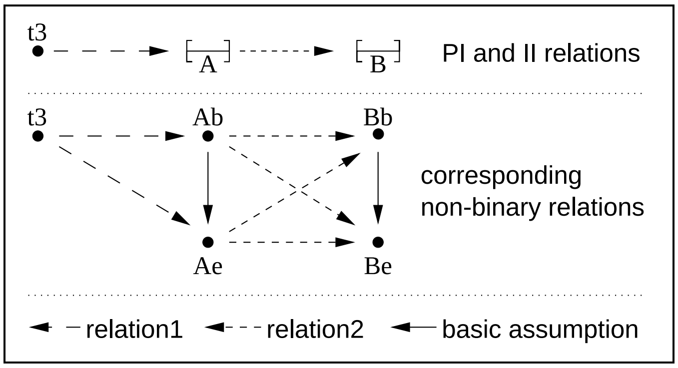


Fig. 2. From PI and II to non-binary relations

tween time point variables that must be considered simultaneously and the type of intervals that are to be allowed.

Finally, II relations affect four time points. Two basic assumptions guarantee that the endings of the two intervals are after their beginnings, while each remaining edge is annotated by an ordinal constraint, analogously to PI relations (cf. "relation2" in Fig. 2). The non-binary relation corresponding to *"A is before or overlaps B"* from Fig. 1 is given by $(A_e, (0, +\infty), B_b) \vee ((A_b, (0, +\infty), B_b) \wedge (A_e, (-\infty, 0), B_b) \wedge (A_e, (0, +\infty), B_e))$.

Figure 3 which has been adapted from Freksa [13] illustrates how Allen's primitive relations and the proposed non-binary representation interact. It also shows that conversion from Allen's primitive relations to non-binary relations and vice versa may proceed by a somewhat tedious, but otherwise straightforward algorithm. A similar proposition can be made for PI relations.

Exchanging the old notation for our new one reveals two dimensions of expressiveness. The first one accounts for the number of constraints that are conjoined in a relation, and the second relates to the algebraic structure underlying the constraints, their composition and intersection. Loosening up on the structural requirements implicit in Meiri's integration model and its underlying schemes, one comes up with a rather free choice for disjunctions of conjoined constraints. The new scheme allows the modeling of problem (1) as follows:

(2) a. 12:00pm: t_0
 b. End of Mr. Roget's flight: t_1
 c. End of Mr. Meyer's flight: t_2
 d. Arrival of Mrs. Meyer in NY: t_3
 e. Beginning of Mr. George's flight: t_4
 f. End of Mr. George's flight: t_5
 g. If Mr. Roget arrives at 3:00pm, then Mr. Meyer arrives two hours later; otherwise, they arrive together at 6:00pm:

$$((t_0, [3, 3], t_1) \wedge (t_1, [2, 2], t_2)) \vee ((t_0, [6, 6], t_1) \wedge (t_1, [0, 0], t_2))$$

h. Mrs. Meyer arrives less than 4 hours after her husband:
$$(t_2, (0,4), t_3)$$
i. Mr. George has a very long flight:
$$(t_4, [\,\text{"very long"}, +\infty), t_5)$$
j. Mr. George's flight starts at 12:00pm:
$$(t_0, [0,0], t_4)$$

In order to answer questions like the one stated in example (1), one must find solutions to the problems arising from this generalized model:

- How is propagation defined on these new relations?
- How can consistency of a network be decided?
- How can a minimal network be computed?
- How can information be dealt with at different levels of granularity (*abstraction*)?
- How can a high level interface be provided (*generalization*)?

The rest of this paper is dedicated to these questions. We begin with a formal description of the approach to non-binary relations.

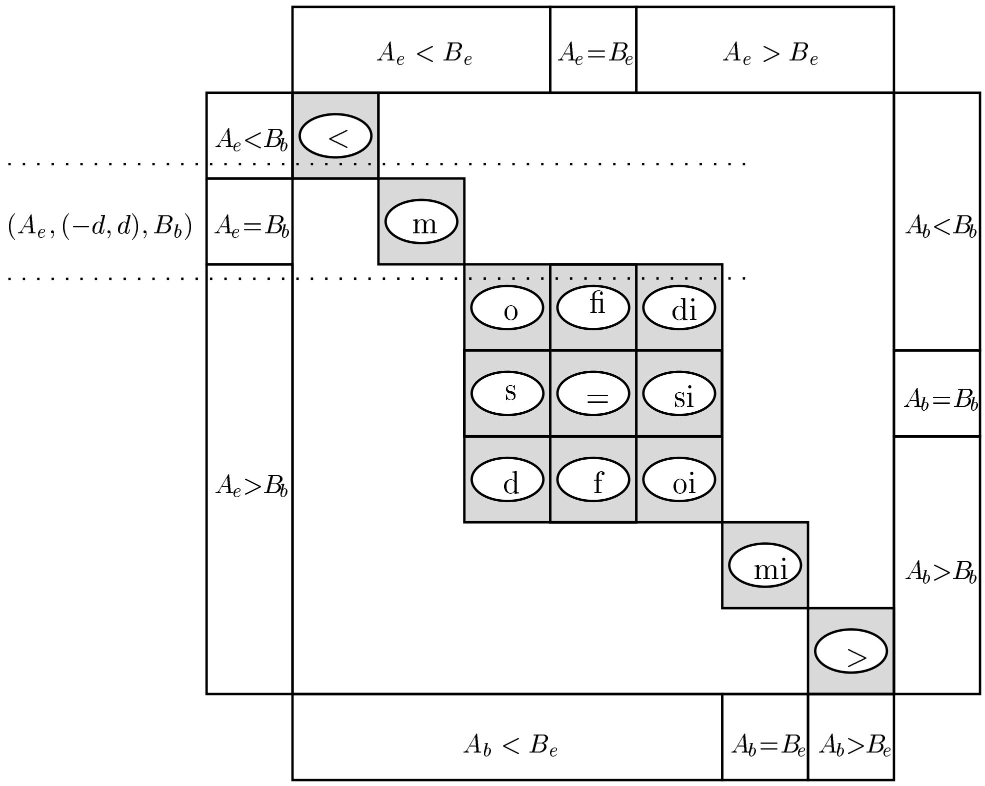


Fig. 3. Changing between Representations

4 GENERALIZED TEMPORAL NETWORKS

We now turn towards a formalization of the data structures for models like (2). As has been illustrated, these data structures, which we refer to as Generalized Temporal Networks (GTNs), build on interval constraints between time points. The intervals may come from different structures such that they can be adapted to various needs:

Definition 1 (Interval Structure) *An interval structure $\mathcal{I}$ is a quadruple $(I, D, \circ, \cap)$. I is a set of (semi-)intervals on a domain D. I is closed under the composition and intersection operators, $\circ$ and $\cap$, respectively.*

The two interval structures used in this paper are rational and ordinal ones: First, the rational interval structure is defined by $\mathcal{I}_{\mathbb{Q}} := (I_{\mathbb{Q}}, \mathbb{Q}, \circ, \cap)$, where $I_{\mathbb{Q}}$ are the intervals on the line of rational numbers $\mathbb{Q}$, $(d_1, d_2) \circ (d_3, d_4)$ is defined as $(d_1 + d_3, d_2 + d_4)$, and intersection is defined as set intersection. For our computational purposes here, we use rational numbers to approximate reals. Second, the ordinal interval structure is given by $\mathcal{I}_O := (I_O, \mathbb{Q}, \circ, \cap)$, which restricts I_O to the intervals $(-\infty, 0), [0, 0], (0, +\infty)$, and their convex unions. [4]

GTN data structures and operators are defined such that the representation of the network as a whole can be described by "$(\bigwedge(\bigvee(\bigwedge p_1 \ldots p_n) \ldots) \ldots)$", with p_i being binary interval constraints between time points:

Definition 2 (GTN) *A generalized temporal network (GTN) $\mathcal{N}$ is a triple $(\mathcal{V}, \mathcal{R}, \{\mathcal{I}_1, \ldots\})$, of vertices $\mathcal{V}$ and relations $\mathcal{R}$ with constraints from a family of interval structures $\{\mathcal{I}_1, \ldots\}$, where:*

- $\mathcal{V} = \{v_i | i = 1 \ldots N\}$ *is a set of time point variables on* $\mathbb{Q}$.
- $\mathcal{R} = \{R_k | R_k = \{P_{k,l} | l = 1 \ldots L_k\}, k = 1 \ldots M\}$ *is a set of relations consisting of disjunctions of conjoined constraints (cf. below for* $P_{k,l}$*); for each* R_k *there exists exactly one* E_k*, the topology of* R_k*, such that* $\mathcal{E} = \{E_k | k = 1 \ldots M\}$ *is a covering of* $\{(v_i, v_j) | i < j \wedge v_i, v_j \in \mathcal{V}\}$. $\mathcal{E}$ *is called the network topology.*
- $\mathcal{P} = \{p_{i,j,k,l} | i, j = 1 \ldots N, i < j, k = 1 \ldots M, l = 1 \ldots L_k\}$ *is a set of primitive constraints* $p_{i,j,k,l} := (v_i, q_{i,j,k,l}, v_j)$, $q_{i,j,k,l} \in I_1 \cup I_2 \cup \ldots$ *and* I_g *is the first component of interval structure* $\mathcal{I}_g$*. Note that for shorthand we sometimes write* $y \in p_{i,j,k,l}$ *instead of* $y \in q_{i,j,k,l}$.
- $P_{k,l} = \{p_{i,j,k,l} | (v_i, v_j) \in E_k\}, k = 1 \ldots M, l = 1 \ldots L_k$, *are conjunctions of*

[4] Some propositions we make here, *e.g.* concerning soundness, also carry over to qualitative schemes, like ones proposed by Clementini et al. [7]. However, other statements, like those concerning efficiency, rely on particular structural properties and cannot be transfered. Their treatment would weaken the statements we want to make here and, hence, are mostly neglected in the rest of this paper (cf. [40]).

primitive constraints. When all vertices are connected, they form an STP network.

- $V : \mathcal{R} \mapsto \mathcal{V}$, $V(R_k) := \{v_i | \exists v_j : (v_i, v_j) \in E_k \vee (v_j, v_i) \in E_k\}$.

For instance, the example model (2) comes with 5 time point variables, *i.e.* $\mathcal{V} = \{t_1, \ldots, t_5\}$. Four relations of the network, given in (2g) to (2j), partially determine the network topology. In particular, the relation in (2g) exhibits the topology $E_1 = \{(t_0, t_1), (t_1, t_2)\}$, with $P_{1,1} = \{(t_0, [3,3], t_1), (t_1, [2,2], t_2)\}$ and $P_{1,2} = \{(t_0, [6,6], t_1), (t_1, [0,0], t_2)\}$ being the elements of R_1. $V(R_1)$ results in $\{t_1, t_2, t_3\}$.

An interesting special case is an unambiguous GTN:

Definition 3 (UGTN) *An unambiguous generalized temporal network (UGTN) $\mathcal{N}$ is a GTN, where all relations consist only of a single clause, i.e., $\forall k : R_k = \{P_{k,1}\}$.*

Starting from the GTN model, we define projection and interval mappings in order to approach the definition of composition.

Definition 4 (Projection) *The projection $\pi : 2^{\mathcal{R}} \times \mathcal{E} \mapsto \mathcal{R}$ is a binary function ($\pi_x(y) := \pi(y, x)$). $\pi_{E_g}(\{R_1, \ldots, R_n\})$ selects all the constraints in $\{R_1, \ldots, R_n\}$ which constrain the edges in E_g. It is defined on the three levels of simple conjoined constraints, of disjunctions of conjoined constraints and of conjoined relations. Its input is described referring to sets (of tuples) K_1, K_2, and K_3, respectively:*

- $\pi_{E_g}(\bigwedge_{(i,j,k,l)\in K_1} p_{i,j,k,l}) := \bigwedge_{(i,j,k,l)\in K_1, (v_i,v_j)\in E_g} p_{i,j,k,l}$,
- $\pi_{E_g}(\bigvee_{(k,l)\in K_2} P_{k,l}) := \bigvee_{(k,l)\in K_2} \pi_{E_g}(P_{k,l})$, *and*
- $\pi_{E_g}(\bigwedge_{k\in K_3} R_k) := \pi_{E_g}(\bigwedge_{k\in K_3}(\bigvee_{l\in[1\ldots L_k]} P_{k,l})) =$
  $\pi_{E_g}(\bigvee_{(x_1,\ldots,x_{|K_3|}), x_i\in[1\ldots L_i]}(\bigwedge_{k\in K_3, q=x_k} P_{k,q}))$.

An example for π is given in (3) which incorporates the information given in (2g):

(3) $\pi_{\{(t_1,t_2)\}}(\ ((t_0, [3,3], t_1) \wedge (t_1, [2,2], t_2)) \vee ((t_0, [6,6], t_1) \wedge (t_1, [0,0], t_2))\ ) =$
$(t_1, [2,2], t_2) \vee (t_1, [0,0], t_2)$

As is illustrated in this example, the application of projection only eliminates restrictions:

Lemma 5 *For all $E \in \mathcal{E}$ and $\{R_1, \ldots, R_m\} \in 2^{\mathcal{R}}$: the constraints given by $\{R_1, \ldots, R_m\}$ entail the constraints given by $\pi_E(\{R_1, \ldots, R_m\})$.*

PROOF. Consider the three levels at which projection is defined:

- For conjunctions of simple propositions, projection is equivalent to conjunction elimination. Hence, Lemma 5 holds at level 1.
- For disjunctions (of A_i) of conjoined propositions $a_{i,j}$ holds by definition: $\pi_E(\bigvee_i A_i) = \pi_E(\bigvee_i(\bigwedge_j a_{i,j})) = \bigvee_i \pi_E(\bigwedge_j a_{i,j}) = \bigvee_i \pi_E(A_i)$. At level 1 for all i: $B_i := \pi_E(A_i)$ is entailed by A_i. Hence, by induction over the length of the disjunction, $\bigvee_i B_i = \bigvee_i \pi_E(A_i)$ is also entailed by $\bigvee_i A_i$, and Lemma 5 holds at level 2.
- The definition of projection at level 3 reduces level 3 to level 2 by applying distributivity of $\wedge$ over $\vee$. Hence, Lemma 5 also holds at level 3. ■

In order to compose constraints from different interval structures, *interval mappings* are established that communicate restrictions.

Definition 6 (Interval Mapping) *Interval mappings are functions* $\mu_{\mathcal{I}_r,\mathcal{I}_s} : I_r \mapsto I_s$ *from one interval structure,* $\mathcal{I}_r = (I_r, D_r, \circ_r, \cap_r)$*, to another one,* $\mathcal{I}_s = (I_s, D_s, \circ_s, \cap_s)$*, such that the following properties are fulfilled:*
$\forall i,j,k,l : (v_i, q_{i,j,k,l}, v_j) \Rightarrow (v_i, \mu_{\mathcal{I}_r,\mathcal{I}_s}(\mu_{\mathcal{I}_s,\mathcal{I}_r}(q_{i,j,k,l})), v_j)$*. If* $D_r = D_s$*, it is also required that* $\forall i,j,k,l : (v_i, q_{i,j,k,l}, v_j) \Rightarrow (v_i, \mu_{\mathcal{I}_r,\mathcal{I}_s}(q_{i,j,k,l}), v_j)$*.*

For instance, the resulting quantitative constraints in example (3) are mapped onto a common ordinal one by $\mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}$:

(4) $(t_1, \mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}([2,2]), t_2) \vee (t_1, \mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}([0,0]), t_2) =$
$(t_1, (0,+\infty), t_2) \vee (t_1, [0,0], t_2) =$
$(t_1, [0,+\infty), t_2)$

Definition 7 (Composition) *The composition of two relations* $R_3 := R_1 \circ R_2$ *is defined by* $R_3 := \bigwedge_{E_k \in \mathcal{E}} \pi_{E_k}(PC(R_1 \cap R_2))$*.*

Thereby, $R_1 \cap R_2 := \bigvee_{P_{1,l} \in R_1, P_{2,l'} \in R_2}(P_{1,l} \wedge P_{2,l'})$*, and* $PC(P_{k,l})$ *computes all the consequences entailed by the STP network corresponding to* $P_{k,l}$ *and returns this network.* $PC(R_k)$ *is defined by* $\bigvee_{P_{k,l} \in R_k} PC(P_{k,l})$*.*

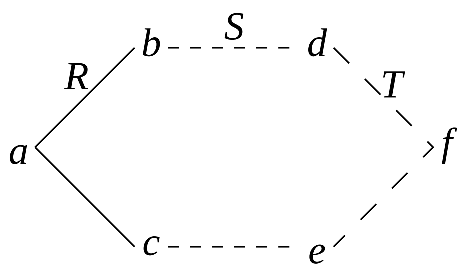
$PC(P_{k,l})$ amounts to the path-consistent version of $P_{k,l}$. In Definition 7, it is implicitly assumed that interval mappings, e.g., $\mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}$, establish a common ground for conjoining constraints from different interval structures. It is always assumed here that such a common ground exists.

5 DETERMINING CONSISTENCY

Given a particular problem (e.g., (1)), a solution is found by computing consequences and, in particular, by determining consistency. One way to approach consistency is by propagating relations. Though, in general, propagation is insufficient to determine consistency, at least it solves simple constraint problems and achieves *path consistency* as an approximation of consistency in more difficult reasoning problems (cf. [28,27]).

5.1 Weakly Generalized Path Consistency

In contrast to simpler approaches, repeated applications of composition need not lead to a path consistent version of GTNs. Figure 4 shows an example that indicates why this is the case. All relations in this example only cover one edge except for the three relations R, S, and T which cover two edges. The problem is that instantiating the "loose ends" a and f with any pair of numbers does not allow for a path consistent assignment of values to b, c, d, and e, since the path (R, S, T) by itself is inconsistent. Due to the *network topology* [5] repeated composition cannot detect this inconsistency and, thus, repeated composition does not enforce path consistency. The reader may note that this problem is not incurred by the particular way that composition is defined for GTNs, but it always prevails when the result of composing two relations is only propagated to restrict relations that already exist.



$$R := ((a, [1,1], b) \wedge (a, [2,2], c)) \vee ((a, [2,2], b) \wedge (a, [1,1], c))$$

$$S := (b, [0,0], d) \wedge (c, [0,0], e)$$

$$T := ((d, [1,1], f) \wedge (e, [1,1], f)) \vee ((d, [2,2], f) \wedge (e, [2,2], f))$$

Fig. 4. PC not computable by repeated composition

A slightly weaker, but very valuable, criterion than path consistency is *weakly generalized path consistency*, which can be enforced independently from network topology by repeated composition.

Definition 8 (WGPC) *A relation R_k is weakly generalized path consistent (WGPC) with regard to a vertex path $(v_0 \ldots v_n)$ and a relation path $(R_1 \ldots R_n)$ iff $(v_0, v_n) \in E_k \wedge \forall i \in [0 \ldots n-1] : (v_i, v_{i+1}) \in E_{i+1} \wedge \forall x_0, x_n \in \mathbb{Q} : (x_n - x_0) \in \pi_{\{(v_0,v_n)\}}(R_k)$ implies $\exists x_1, \ldots, x_{n-1} \in \mathbb{Q}\ \forall i \in [0 \ldots n-1] : (x_{i+1} - x_i) \in \pi_{\{(v_i,v_{i+1})\}}(R_{i+1})$. A GTN is WGPC iff all its relations are WGPC with regard to all vertex paths and relation paths.*

[5] If there was a relation R_h with $E_h = \{(a,d),(a,e)\}$, then this inconsistency could be detected.

The intuition behind WGPC is that when one projects all non-binary relations onto binary ones between time points, the resulting network of binary relations is path consistent. For instance, propagating relations in the example GTN N given in Figure 4 yields a GTN N′ by adding some constraints on relations other than R, S and T. The corresponding "projection network" N″ with only binary relations $R_1 := (a, [1, 1], b) \vee (a, [2, 2], b)$, $R_2 := (a, [2, 2], c) \vee (a, [1, 1], c)$, $S_1 := (b, [0, 0], d)$, $S_2 := (c, [0, 0], e)$, $T_1 := (d, [1, 1], f) \vee (d, [2, 2], f)$, $T_2 := (e, [1, 1], f) \vee (e, [2, 2], f)$ and further binary relations projected from N′, e.g. between a and f, is path consistent. Therefore N′ is WGPC.

Lemma 9 *A GTN is WGPC iff all relations R_k are WGPC with regard to all relation paths of length 2.*

PROOF. "$\Rightarrow$": Trivial.

"$\Leftarrow$": Induction over the length of relation paths.

Induction start is given by the premise of the theorem.

Assumption: All relations R_k are WGPC with regard to all relation paths of length n.

Induction step: Consider a relation R_k, a vertex path $(v_0 \dots v_{n+1})$ and a relation path $(R_1 \dots R_{n+1})$ such that the premise of the implication in Def. 8 holds. According to the structure of GTNs and the induction assumption there is a relation R'_k that is WGPC with regard to $(v_0 \dots v_n)$ and $(R_1 \dots R_n)$. Also by the induction assumption R_k is WGPC with regard to (v_0, v_n, v_{n+1}) and (R'_k, R_{n+1}). Therefore, for all values x_0 and x_{n+1} allowed by R_k for v_0 and v_{n+1}, respectively, we find a value x_n for v_n such that $x_n - x_0 \in \pi_{\{(v_0,v_n)\}}(R'_k)$ and $x_{n+1} - x_n \in \pi_{\{(v_n,v_{n+1})\}}(R_{k+1})$. Due to the induction assumption we also find values $x_1 \dots x_{n-1}$ for $v_1 \dots v_{n-1}$. ■

In order to apply this lemma, one must abstract from the models underlying a relation by a syntactic criterion. For Allen's or for TCSP relations, this abstraction, i.e. *subsumption*, can be easily computed by comparing the constraint sets, e.g. $\{<, m, o\}$ subsumes $\{<, m\}$. For GTNs subsumption may be hard to compute, however a syntactic criterion that only implies semantic subsumption, but that itself need not be implied by semantic subsumption may be given by by assigning models in Euclidean space to all relations and comparing these models, *viz. θ-subsumption*:

Definition 10 (θ-Subsumption) *A relation* $R_1 = \{P_{1,l} | l = 1 \dots L_1\}$ *θ-subsumes a relation* $R_2 = \{P_{2,l} | l = 1 \dots L_2\}$ $(R_1 \trianglerighteq R_2, R_2 \trianglelefteq R_1)$ *if* $\bigcup_{l=1\dots L_2} P'_{2,l} \setminus \bigcup_{l=1\dots L_1} P'_{1,l} = \emptyset$, *where*

$$P'_{k,l} = \bigtimes_{i,j=1\dots N, i<j} \begin{pmatrix} q_{i,j,k,l}, \text{ iff } (v_i, v_j) \in R_k \\ D, \quad \text{otherwise} \end{pmatrix}.$$

I.e., $P'_{k,l}$ are given interpretations as hyper-quadrics in $D^{|\mathcal{V}|(|\mathcal{V}|-1)/2}$ partially in-/excluding their boundaries and "$\setminus$" denotes set difference. The following lemma associates the notion of θ-subsumption with the models possible for a relation.

Lemma 11 *If $R_1 \trianglerighteq R_2$ then every model for the relation R_2 that assigns values to time point variables in $\mathcal{V}$ is also a model for the relation R_1.*

PROOF. Assume an interpretation which assigns values $\bar{x} = \{x_i | i = 1 \dots N\}$ to all time point variables in $\mathcal{V}$ and which is a model for R_2. *I.e.*, $\exists P_{2,l} \in R_2\ \forall (v_i, v_j) \in E_2 : x_j - x_i \in p_{i,j,2,l}$. By construction this implies that $\bar{x} \in \bigcup_{l=1\dots L_2} P'_{2,l}$. By the definition of θ-subsumption also $\bar{x} \in \bigcup_{l=1\dots L_1} P'_{1,l}$. Hence, $\exists l' : \bar{x} \in P'_{1,l'}$. Thus, $\bar{x}$ fulfills all restrictions of R_1. ■

For instance, the result in (3) θ-subsumes the input given to $\pi_{\{(t_1,t_2)\}}$, the result in (4) θ-subsumes the result of (3), and due to the transitivity of θ-subsumption the relation in (4) θ-subsumes the one in (2g).

Now, we can give a syntactic check for WGPC.

Theorem 12 *A GTN is WGPC if* $\forall R_g, R_h, R_k \in \mathcal{R} : \pi_{E_k}(R_g \circ R_h) \trianglerighteq R_k$.

PROOF. Due to Lemma 9 we only have to show that if the premise of Theorem 12 holds all relation paths of length 2 are WGPC. This is true by Definition 7 (composition) and by Lemma 11. ■

5.2 Constraint Propagation

With Theorem 12, composition, and θ-subsumption, all the necessary means for computing WGPC are supplied. However, the way composition is defined still prevents efficient computations in all but the most benign cases. Given any pair of relations R_1, R_2 with L_1 and L_2-many disjunctions, $R_1 \circ R_2$ yields $L_1 \cdot L_2$-many disjunctions. After n iterations the representation of relations would most often involve a number of disjunctions exponential in n. In general, this explosion cannot be avoided, since even simple TCSP problems may incur such *fragmentation* which renders the number of disjunctions in one relation exponential to the numbers of relations in the network [6] (cf. [37]). However, very often relations overlap, contain each other or there are only a finite number of them — such as in networks based on $\mathcal{I}_O$. Thus, having computed composition, we optimize the resulting representation before we proceed with further iteration.

Optimization may be done in a naive way, *e.g.* by simply enumerating disjunctions at the finest granularity and removing "atoms" that are redundant (henceforth called *naive optimization*). However, this simplistic method is not really an optimization, since it usually involves an unnecessary abundance of disjunctions, *e.g.* assuming a granularity of one units, 150 disjunctions for the simple relation $(t_1, [0, 10], t_2) \wedge (t_2, [0, 15], t_3)$ are needed. Hence, we conceived optimizations that minimize the number of disjunctions.

Lemma 13 *A locally optimal representation $Opt(R_k)$ for disjunctions $R_k = P_{k,1} \vee \ldots \vee P_{k,L_k}$, where $p_{i,j,k,l} \in I_{\mathbb{Q}}$, can be found in $\mathcal{O}(2^{|E_k|} L_k^{3|E_k|+2})$.*

Proof Sketch [7] **:** An optimizing hyper-quadric θ-subsumes two or more disjunctions while it is θ-subsumed by the set of all disjuncts. Thus, proceed as follows: For a candidate θ-subsuming hyper-quadric there are L_k possibilities to choose the upper and lower boundary in each of the $|E_k|$ dimensions. Testing whether it actually θ-subsumes more than one disjunct can be done in $\mathcal{O}(L_k)$. Testing whether it is itself θ-subsumed by the complete disjunction $P_{k,1} \vee \ldots \vee P_{k,L_k}$ takes $\mathcal{O}(2^{|E_k|} L_k^{|E_k|+1})$ primitive algebraic operations (cf. [40]). Since at most $L_k - 1$ optimizations may be executed, the whole optimization process needs time $\mathcal{O}(L_k^{2|E_k|}) \cdot \mathcal{O}(2^{|E_k|} L_k^{|E_k|+1} + L_k) \cdot \mathcal{O}(L_k) = \mathcal{O}(2^{|E_k|} L_k^{3|E_k|+2})$.

The result is locally optimal, because, by way of construction, no further pair of $P_{k,l}, P_{k,l'}$ can be substituted by a single conjunction of constraints, but it is not clear whether repeated shrinking and growing of quadrics could produce a globally better result (i.e., fewer disjunctions). ■

Efficient parallel implementations can readily build upon this optimization procedure. In our current implementation we use a less powerful, but more efficient optimization which discards one $P_{k,l}$ when it is already θ-subsumed

[6] E.g., consider $((a_1, [1, 1], a_2) \vee (a_1, [2, 2], a_2)) \wedge \ldots \wedge ((a_{n-1}, [2^{2n-4}, 2^{2n-4}], a_n) \vee (a_{n-1}, [2^{2n-3}, 2^{2n-3}], a_n))$ then there are 2^{n-1} disjunctions for the relation on (a_1, a_n).

[7] Cf. [40] for an elaborate exposition.

Algorithm 1. Computing WGPC
Input: $(\mathcal{V}, \mathcal{R}, \{\mathcal{I}_1, \ldots\})$
begin
$Q := \{\{R_g, R_h\} | V(R_g) \cap V(R_h) \neq \emptyset\}$;
while $Q \neq \emptyset$ **do**
   select and delete a set $\{R_g, R_h\}$ from Q;
   $R' := R_g \circ R_h$ with $E' := E_g \cup E_h$;
   forall R_k such that
         $E_k \cap (V(R_g) \cup V(R_h)) \times (V(R_g) \cup V(R_h)) \neq \emptyset$
         do $R'_k := R_k \leftharpoonup \pi_{E_k}(R')$;
            if $R'_k = \emptyset$ **then** **exit**(inconsistent); **fi**;
            if $\mathit{Improved}(R'_k, R_k)$ **then** **do**
               $R_k := R'_k$;
               $Q := Q \cup \{\{R_k, R_f\} | V(R_k) \cap V(R_f) \neq \emptyset\}$;
               od **fi**;
         od;
od;

by the rest. As mentioned above, this test can be performed for one $P_{k,l}$ of one R_k in $\mathcal{O}(2^{|E_k|} L_k^{|E_k|+1})$ algebraic operations.

Using Theorem 12, WGPC is now computed as follows in Algorithm 1: One composes all relations that have at least one node in common and intersects the result with all relations that may be tightened by this composed relation. Thus, one computes consistency for each triplet of time point variables and achieves a scale-up from TCSPs. For now, we assume that the intersection operator "$\leftharpoonup$" is defined as $R_1 \leftharpoonup R_2 := \mathit{Opt}(PC(R_1 \cap R_2))$, and the Boolean function $\mathit{Improved}(R'_k, R_k)$ returns true iff $\neg(R'_k \trianglerighteq R_k)$.

One can prove that:

Theorem 14 *Algorithm 1 is sound, yet incomplete. If the GTN is based solely on $\{\mathcal{I}_{\mathbb{Q}}\}$ or only on $\{\mathcal{I}_O\}$, upon termination of the algorithm the resulting network is WGPC.*

PROOF. Soundness: The only actual operation on the network is $R_k := \mathit{Opt}(PC(R_k \cap \pi_{E_k}(R_g \circ R_h)))$. "$Y := \mathit{Opt}(X)$" optimizes the representation X in a way such that $X \trianglerighteq Y$ and $Y \trianglerighteq X$ and therefore, according to Lemma 11 all models for X are models for Y and vice versa. "PC" computes consequences which are sound. "$\cap$" conjoins the restrictions from R_k with those from $\pi_{E_k}(R_g \circ R_h)$, hence this operation is sound when the restrictions in $\pi_{E_k}(R_g \circ R_h)$ are sound. Projection is sound by Lemma 5. $R_g \circ R_h$ is defined by $\bigwedge_{E_k \in \mathcal{E}} \pi_{E_k}(PC(R_g \cap R_h))$, which is sound by similar considerations as have just been outlined for projection, "PC", and "$\cap$". Since, the only actual operation is composed by sound operations, it is sound too.

WGPC: The WGPC condition for networks based on complete substructures of $\mathcal{I}_{\mathbb{Q}}$ is that for all triples $R_g, R_h, R_k \in \mathcal{R} : R_k \trianglelefteq \pi_{E_k}(R_g \circ R_h)$ (cf. Theorem 12). The initialization of the queue Q with all pairs $\{R_g, R_h\}$ ensures that this criterion is checked for all triples. If this criterion has just been established for a triple R_g, R_h, R_k, its validity is checked for all triples that may be affected by the revision of R_k. Only when the θ-subsumption condition is fulfilled for all triples the queue Q becomes empty and the algorithm stops.

Note that networks with constraints from $\mathcal{I}_{\mathbb{Q}}$ and $\mathcal{I}_O$ freely interspersed may fail to enforce the condition $\forall R_g, R_h, R_k \in \mathcal{R} : \pi_{E_k}(R_g \circ R_h) \trianglerighteq R_k$ if the operator $\mu_{\mathcal{I}_{\mathbb{Q}},\mathcal{I}_O}$ is used to map the results of composition back onto coarser constraints. Thus, such mixed networks may not become WGPC.

Incompleteness: Networks which can represent Allen's interval relations can model the network for which Allen's path propagation is incomplete (cf. [2]). That network is inconsistent, but path consistent. Since path consistency in Allen's model entails WGPC in the corresponding GTN model and since the achievement of WGPC terminates Algorithm 1, the inconsistency cannot be detected. Hence, it is incomplete for networks that model Allen's relations.

■

With constraint propagation one may approximate the determination of consistency. But for UGTNs one fares better:

Lemma 15 *A weakly generalized path consistent UGTN that has admissible values for all relations is consistent.*

PROOF. A weakly generalized path consistent UGTN is equivalent to a path consistent STP. For STPs path consistency with admissible values for all relations is equivalent to consistency (cf. [9]). ■

5.3 Efficiency

Applying Algorithm 1, which enforces WGPC, one may now search with backtracking in the space of UGTNs underlying a GTN to determine consistency. As an alternative, one may directly use Algorithm 1 as an approximation algorithm. Either way the performance crucially depends on its computational complexity.

Theorem 16 *If $\mathcal{E}$ is a partitioning, Algorithm 1 terminates in $\mathcal{O}(N^3T^{3u+u^2})$, where $N = |\mathcal{V}|$, $T = \max_{p_{i,j,k,l}\in\mathcal{P}}(\max_{x\in q_{i,j,k,l}} x - \min_{x\in q_{i,j,k,l}} x)$ is the maximal range of single constraints*[8]*, and $u = \max_{E_k\in\mathcal{E}} |E_k|$ is the maximum number of edges one relation has. If $\mathcal{E}$ is a partitioning and only naive optimization is performed, Algorithm 1 terminates in $\mathcal{O}(N^3T^{3u})$.*

PROOF. $R_g \circ R_h$ involves $L_g \cdot L_h$ times determining path consistency (for $P_{g,l} \wedge P_{h,l'}$). A single relation has less than T^u disjunctions which means that path consistency must be computed at most T^{2u} times. Enforcing path consistency for STPs takes $\mathcal{O}(n^3)$ with n being the number of vertices in the network (cf. [9]). Hence, each enforcement of path consistency in the $P_{g,l} \wedge P_{h,l'}$ STP network takes time $\mathcal{O}(|V(R_g) \cup V(R_h)|^3) = \mathcal{O}((2t)^3)$, where $t = \max_{R_k\in\mathcal{R}}(|V(R_k)|)$. Thus, a single composition needs $\mathcal{O}(8t^3T^{2u})$.

Computing $\leftharpoondown$ is done at most $t(t-1)/2$ times and each time it may result in at most T^{2u} many disjunctions, which need to be considered. Enforcing path consistency on each requires $\mathcal{O}(t^3)$ steps. Naive optimization may be seamlessly integrated into the computation of the T^{2u} many disjunctions and results in at most T^u disjunctions. When we apply our current opimization strategy we invest another $\mathcal{O}(T^u 2^u (T^u)^{u+1}) = \mathcal{O}(2^u T^{u^2+2u})$ steps into the computation. Hence, computing the operations associated with $\leftharpoondown$ takes time $\mathcal{O}(t^2(T^{2u}+t^3T^u+2^uT^{u^2+2u}))$, which amounts to $\mathcal{O}(T^{2u})$ and $\mathcal{O}(T^{2u+u^2})$ without and with our optimization strategy, respectively, when we neglect t and u as rather small constants. Thus, the worst case effort for each relation set in the queue is bound by $\mathcal{O}(T^{2u}) + \mathcal{O}(T^{2u+u^2}) = \mathcal{O}(T^{2u+u^2})$ with our optimization scheme and $\mathcal{O}(T^{2u})$ with only naive optimization.

At most M relations may be updated at most T^u times and, thereby, at most tN new relation sets may be put into the queue. Hence, Algorithm 1 terminates with our and with naive optimization only in $\mathcal{O}(MNT^{3u+u^2})$ and $\mathcal{O}(MNT^{3u})$, respectively. There are N^2 many edges. When $\mathcal{E}$ is a partitioning there are between N^2 and N^2/u many relations with u and t fixed. Hence, M is of $\mathcal{O}(N^2)$ and Algorithm 1 terminates in $\mathcal{O}(N^3T^{3u+u^2})$ — $\mathcal{O}(N^3T^{3u})$ with naive optimization only. ∎

Though, at first sight, the result that good optimizations of representations incur higher costs than their naive counterpart is somewhat counterintuitive, the reason for this result is quite straightforward. Whereas our optimization process does not produce any benefits in the worst case, it always requires an expensive computation process (cf. footnote 6). Nevertheless, in preliminary

[8] We assume integer ranges here. Rational constraints can be transformed into equivalent integer constraints.

practical experiences our optimization scheme seemed to considerably improve performance.

Algorithm 1 shows a reasonable computational behavior, because its performance decreases only smoothly in comparison to constraint propagation algorithms for less expressive mechanisms. In particular, one may recognize that the larger part of its computational complexity stems from numeric fragmentation as it already occurs in TCSPs. The generalization to non-binary relations does not incur an increase of computational complexity for qualitative relations, because in such a generalization T and u are small constants and the overall complexity is in the order of $\mathcal{O}(N^3)$ — the same as for Allen's propagation of interval relations. Concerning quantitative interval structures the difference between the TCSP scheme ($\mathcal{O}(N^3T^3)$ steps) and our approach stems from the parameter u which mirrors the increased expressiveness in terms of more complicated relations. Assuming $u = 1$ and disregarding optimization, which is trivial for $u = 1$, our constraint propagation algorithm shows the same behavior as the one for TCSPs.

Still, the range factor in the computational complexity of the propagation algorithm may prove too hard to live with for very many applications. Since we will delve more deeply into issues of trading off between expressiveness and efficiency in Section 7, let us also postpone the discussion of strategies that confront this matter to that section.

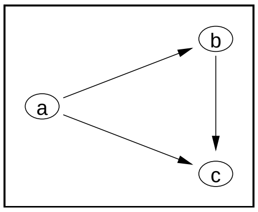
6 COMPUTING THE MINIMAL NETWORK

The two major propositions commonly sought from a temporal constraint network concern its consistency and its minimal equivalent network. Moving between various temporal reasoning mechanisms, the meaning of consistency remained by and large unaffected, though we had to rethink its preliminaries, *viz.* (WG)PC and constraint propagation. For the problem of computing minimality, switches between levels of granularity turn out to be even more pervasive. To illuminate the difficulties, let us consider the common definition of "minimal network" first:

Definition 17 *The minimal network of a given network $\mathcal{N}$ is the tightest equivalent network $\mathcal{N}'$. A network $\mathcal{N}'$ is at least as tight as another one $\mathcal{N}$ if all constraints in $\mathcal{N}'$ are subsumed by the corresponding constraints in $\mathcal{N}$.*

There are two underlying assumptions in this common definition that appear difficult for non-binary temporal relations. First, it is assumed that the *comparison of tightness* of relations may be easily computed. Second, it is assumed that there *exists* a tightest relation for a set of equivalent relations.

Figure 5 shows a simple example network which illustrates some part of the problem implied by these assumptions. There is a single relation R in this network which covers the three available edges. Though we have a singleton labelling and WGPC is established, the relation may be considered non-minimal, e.g., $R' := ((b,[1,1],c) \wedge (a,[0,0],b) \wedge (a,[1,1],c)) \vee ((b,[1,1],c) \wedge (a,(-\infty,0),b) \wedge (a,(-\infty,1),c)) \vee ((b,[1,1],c) \wedge (a,(0,\infty),b) \wedge (a,(1,\infty),c))$ has tighter constraints, because it is θ-subsumed by R and it does itself not θ-subsume R, but it has the same models as the depicted relation. Indeed, since $(b,[1,1],c)$ creates a linear dependency between the restrictions on (a,b) and (b,c), namely $b-a=c-a-1$, there is no GTN relation that does not θ-subsume a semantically equivalent GTN relation.



$$
\begin{aligned}
R := \quad & (a,(-\infty,+\infty),b)\wedge \\
& (a,(-\infty,+\infty),c)\wedge \\
& (b,[1,1],c)
\end{aligned}
$$

Fig. 5. A Non-Minimal, WGPC, and Singleton Labelling

At a first glimpse, one might be tempted to trace this difficulty only to the definition of θ-subsumption given here. However, one may recognize that *in general* the comparison of tightness of non-binary relations, which may be composed by many constraints in an intricate way, may be a computationally hard task. Therefore, true subsumption between non-binary relations may very often be hard to decide and, hence, θ-subsumption — or a similar syntax-based operator — may be the only decision criterion available.

Furthermore, for practical purposes there is a frequent need for explicitly specifying the atomic level of relations. Minimal networks are often, e.g. for Allen's calculus, computed in the following way: The network is split into disjunctions, which depend on the atomic level of relations, such that for all disjuncts the enforcement of path consistency entails their minimality. The unions of these single results then form the minimal relations. Therefore, the efficiency and the actual result of computing the minimal network depends on the granularity of relations considered atomic.

For instance, for a relation in a given network, e.g. a GTN corresponding to Meiri's integration model, one may decide that Allen's {"*precedes*"} is atomic. {"*precedes*"} may then be a perfect minimal labelling for the relation even when other constraints could in princple enforce a tighter relation. Alternatively, an explicit specification of the atomic level could allow for the break down of "*precedes*" into "*precedes by more than 10 units*" and "*precedes by at most 10 units*". Then the labelling {"*precedes*"} might turn out not to be minimal, because {"*precedes*"} = {"*precedes by more than 10 units*","*precedes by at most 10 units*"} and "{*precedes by more than 10 units*}" may be a tighter labelling enforced by the network.

In order to make underlying assumptions about the granularity of atomic relations transparent in our generalized framework, we here introduce the notion of *minimality at a certain level of granularity.* For this purpose, however, we must first formalize the notion of *granularity level.*

Definition 18 (Granularity Level for a Topology) *A set of GTN relations $S := \{R_{k_i} | k = 1 \dots M, i = 1 \dots J_k\}$ describes a* granularity level G *for a corresponding topology $\mathcal{E} := \{E_k | k = 1 \dots M\}$ iff,* (i), *$\forall R_{k_i} \in S : R_{k_i}$ has a valid instantiation and $L_{k_i} = 1 \wedge \forall R_{k_j} \neg(R_{k_i} \trianglerighteq R_{k_j}) \wedge R_{k_i} = \pi_{E_k}(PC(R_{k_i}))$ and,* (ii), *$\forall E_k : \bigvee_{i=1 \dots J_k} R_{k_i} \trianglerighteq R_k^0$, where $R_k^0 := \bigwedge_{(v_i,v_j) \in E_k} (v_i, D_{i,j}, v_j)$ are the non-constraining relations for topology $\mathcal{E}$ and $D_{i,j}$ are the domains relevant for $q_{i,j,k,l}$.*

Condition *(i)* in Definition 18 describes a criterion for atomicity of a set of instantiable relations and is, thus, appropriate for describing a level of granularity. Atomicity of a relation is dependent on one's view on the system. Allen's relations (e.g., "before") may be considered atomic from one point of view, but divisible from another one (e.g., "before" may be split into "before, but at most 1 unit" and "more then 1 unit before"). The subcondition $R_{k_i} = \pi_{E_k}(PC(R_{k_i}))$, which enforces $R_{k,i}$ to be path-consistent, in combination with $L_{k_i} = 1$ ensures that the "$\trianglerighteq$"-operator allows the comparison of all the relations in S according to their instantiations — hence it allows to reverse the proposition of Lemma 11 given its additional premises:

Lemma 19 *Given two relations $R_k := \{P_{k,1}\}, R_{k'} := \{P_{k',1}\}$. If (a) $R_k = \pi_{E_k}(PC(R_k))$ and if (b) all proper instantiations of R_k are also proper instantiations of $R_{k'}$, then (c) $R_{k'} \trianglerighteq R_k$.*

PROOF. (a) ensures that $\{P_{k,1}\}$ is (a subset of) a minimal STP network. $\{P_{k',1}\}$ is (a subset of) an arbitrary STP network. Due to (b) every instantiation of $\{P_{k,1}\}$ is also a proper instantiation of $\{P_{k',1}\}$, hence all the constraints in $\{P_{k,1}\}$ are tighter than in $\{P_{k',1}\}$, *i.e.* $\forall q_{i,j,k,1} \in P_{k,1} : q_{i,j,k,1} \subseteq q_{i,j,k',1} \vee (v_i, v_j) \notin E_{k'}$. Therefore the Euclidean model of R_k is contained in the one of $R_{k'}$, *i.e.* $R_{k'} \trianglerighteq R_k$.

Condition *(ii)* in Definition 18 guarantees that S is complete, i.e., it allows for all instantiations of all time points that are possible *a priori.*

Definition 20 (Minimality at Granularity Level) *A network $\mathcal{R}'$ with topology $\mathcal{E}$ is minimal at granularity level G, defined by S with the corresponding topology $\mathcal{E}$, if for all relations R'_k and for each split of R'_k into $R'_k = \bigvee R_{k_s}$,* *where $R_{k_s} \in S$, there is a value for all R_{k_s} such that a consistent instantiation can be chosen for the rest of the network.*

We may then claim:

Corollary 21 *If all relations R_k of a UGTN with topology $\mathcal{E}$ are from S and the UGTN is WGPC, then the UGTN is minimal with regard to the chosen granularity G (described by S and $\mathcal{E}$).*

PROOF. Follows directly from Lemmata 15 and 19. ■

Thus, minimality can be computed by splitting GTNs into UGTNs, by splitting UTGNs into relations from granularity G, by computing consistency for each resulting UGTN of granularity G, and by taking the union over the single results.

In particular, this is an interesting result for GTNs building on $\mathcal{I}_O$. The classification of the finite number of ordinal relations [9] (for a bounded $\max_{E_k \in \mathcal{E}} |\{v_i| \exists v_j : (v_i, v_j) \in E_k \vee (v_j, v_i) \in E_k\}|$) allows the establishment of minimality at this level of granularity — which is equivalent to the original notion of minimality for ordinal (non-)binary relations, like Allen's networks.

7 ABSTRACTION AT THE REASONING LEVEL — EXPRESSIVENESS VS. EFFICIENCY

Venturing from binary to non-binary temporal relations was what we described so far. However, for purposes of flexibility, efficiency and understandability we also need to consider switching back from complex, expressive temporal theories into sparser ones that are more accessible for computations — and possibly for humans (described in Section 8). The GTN model has been devised in order to provide an apt foundation for switching between different levels of reasoning. GTNs based on intervals from the rationals, $\mathcal{I}_{\mathbb{Q}}$, bring about a very fine-grained level of temporal reasoning as a general frame of reference. Switching to coarser models is possible relative to at least two dimensions: First, the dimension of interval structures of different granularities permits such changes, e.g., abstractions from quantitative constraints to qualitative, e.g., ordinal ones, like $\mathcal{I}_O$, or granularity changes between days, weeks and months, as described, *e.g.*, by Bettini *et al.* [5] or by Chandra *et al.* [6]. Second, one may consider disjunctions of conjoined constraints as already too

[9] For instance, there exist 13 primitive qualitative relations on three time points and 59 ones on four time points.

sophisticated a level of representation. From such a level, it is, nevertheless, possible to move into a sparser theory, by using abstraction on the propositional level (cf. Giunchiglia & Walsh [15]).

Figure 6 is an excerpt of a network heterarchy of temporal reasoning schemes (with arrows pointing from less towards more expressive formalisms). $GTN(\mathcal{I}_{\mathbb{Q}})$ and $GTN(\mathcal{I}_O)$ denote GTNs based only on the interval structures $\mathcal{I}_{\mathbb{Q}}$ and $\mathcal{I}_O$, respectively. STP and TCSP stand for non-disjunctive and disjunctive quantitative constraint systems, respectively, as described by Dechter *et al.* [9]. [10] The term *integration* stands for the integration of TCSPs with Allen's model [27]. TCSP-LPC (cf. Schwalb & Dechter [37]) is not really a representation schema on its own. Viewed from a representational perspective, it is equivalent to TCSPs, but it propagates only a limited number of disjunctions in each step such that propagation, as a whole, remains polynomial in the number of relations.

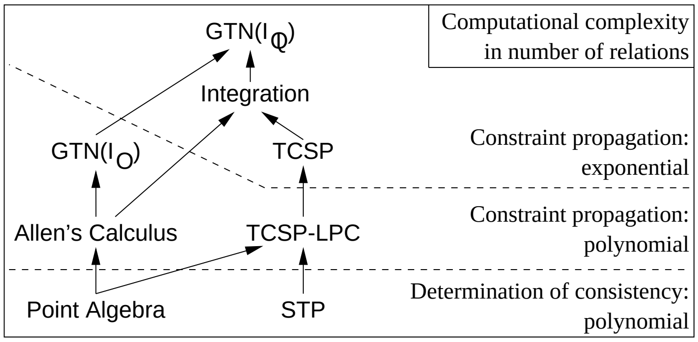


Fig. 6. Expressiveness of Reasoning Schemes

This heterarchy mirrors the well-known trade-off between expressiveness and efficiency. Determining consistency is NP-hard in all formalisms, except for the point algebra and for STP networks (cf. [45,9,14,22]). However, even approximating constraint propagation algorithms can be very expensive when large ranges are embodied in the network.

We attempt to deal with this complexity bottleneck by providing smooth shifts among different levels of expressiveness. Following Hobbs's strategy that "idealization allows simplifications into tractable local theories", our proposal approximates given information by "simpler" one. These shifts are performed by two families of operators already introduced before: The first one, π_{E_g}, takes interdependent constraints as input and disregards their relationships, e.g., as with the disjunction in (3). The second one, $\mu_{\mathcal{I}_r,\mathcal{I}_s}$, allows switching between different interval granularities, as, e.g., illustrated by collapsing information in (4).

[10] In the formal framework of GTNs, for TCSPs we require $\forall k : |E_k| = 1$, and for STPs we assume $\forall k : |E_k| = 1 \wedge L_k = 1$.

As can be read off from the diagram, both idealizations abstract from networks composed of detailed representations, with expensive constraint processing, to coarser representations, which allow for more efficient reasoning. Hence, expressiveness is traded off against efficiency. Disregarding structural interdependencies, e.g., allows the projection of $GTN(\mathcal{I}_O)$ information into an efficiently solvable point algebra. A coarser level of quantities, and thus a small overall range, is directly reflected by a tighter worst-case bound for constraint propagation (cf. Theorem 16).

Thus, one may control the extent to which constraints are propagated in GTNs in order to approximate, e.g. determination of consistency. Instead of having only some crude heuristics for control, GTNs as an encompassing framework allow explicit control along the heterarchy shown above. Hence, one may decide to do only reasoning as for a point algebra within a full-fledged GTN and, thereby, exploit the beneficial computational properties of PA — of course incurring incompleteness.

Thereby, the soundness of both abstraction operators is ensured by Definition 6 for operators μ and by Lemma 5 for operators π.

Let us now illustrate the use of these abstraction mechanisms by considering the temporal reasoning problem given in (2). In order to retrieve qualitative ordering information, such as determining arrival orderings, it is often desirable to move down the heterarchy from GTN($\mathcal{I}_\mathbb{Q}$) to a point algebra. This is done for two relevant pieces of knowledge. For (2g) this happens by the composition of operations in (3) and (4),

(2g) If Mr. Roget arrives at 3:00pm, then Mr. Meyer arrives two hours later; otherwise, they arrive together at 6:00pm:
$((t_0, [3,3], t_1) \wedge (t_1, [2,2], t_2)) \vee ((t_0, [6,6], t_1) \wedge (t_1, [0,0], t_2))$

(3) $\pi_{\{(t_1,t_2)\}}(\ ((t_0, [3,3], t_1) \wedge (t_1, [2,2], t_2)) \vee ((t_0, [6,6], t_1) \wedge (t_1, [0,0], t_2))\ ) =$
$(t_1, [2,2], t_2) \vee (t_1, [0,0], t_2)$

(4) $(t_1, \mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}([2,2]), t_2) \vee (t_1, \mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}([0,0]), t_2) =$
$(t_1, (0,+\infty), t_2) \vee (t_1, [0,0], t_2) =$
$(t_1, [0,+\infty), t_2)$

And for (2h) this move is drawn in (5):

(2h) Mrs. Meyer arrives less than 4 hours after her husband: $(t_2, (0,4), t_3)$

(5) $\mu_{\mathcal{I}_\mathbb{Q},\mathcal{I}_O}((t_2, (0,4), t_3)) = (t_2, (0,+\infty), t_3) \Leftrightarrow t_2 < t_3$

From (4) and (5) we may easily read off that Mr. Roget, Mr. Meyer and

Mrs. Meyer arrive just in this order, the men may even arrive simultaneously.

Given that we have neglected knowledge about durations, we do not know how Mr. George's arrival is ordered with respect to the other ones. What is needed is reasoning at the level of TCSPs — on the one hand:

(6) $\pi_{\{(t_0,t_1)\}}(\ ((t_0,[3,3],t_1) \wedge (t_1,[2,2],t_2)) \vee ((t_0,[6,6],t_1) \wedge (t_1,[0,0],t_2))\ ) = (t_0,[3,3],t_1) \vee (t_0,[6,6],t_1)$

From (3) and (6) we derive:

(7) $(t_0,[5,5],t_2) \vee (t_0,[3,3],t_2) \vee (t_0,[8,8],t_2) \vee (t_0,[6,6],t_2)$

From (7) and (2h) we conclude:

(8) $(t_0,(5,9),t_3) \vee (t_0,(3,7),t_3) \vee (t_0,(8,12),t_3) \vee (t_0,(6,10),t_3) = (t_0,(3,12),t_3)$

On the other hand, one needs to account for background knowledge about the duration of flights. Assuming an interval structure (like the ones proposed by Clementini *et al.* [7]) referring to flights of "short", "medium", "long", and "very long" time extension, a common grounding between "very long" and hour units may be that "very long flights" take at least 15 hours (the link between "very long" and its context "flight durations" may be computed as proposed by Staab & Hahn [41]). With this information and with (8), one may conclude, finally, that Mr. George will arrive last.

Though for most temporal reasoning mechanisms the two families of abstraction operators, π and μ, play the major role, one may think of alternative operators, too. For instance, Schwalb & Dechter [37] encountered the TCSP fragmentation problem, which is also reflected in the highly range-dependent worst case bound of Theorem 16, by restricting propagation to (almost) convex constraints. An operator τ that abstracts from general non-convex relations into a limited number of convex disjunctions may render constraint propagation similarly efficient in our account. However, the disadvantage remains that the resulting network does not have a similarly relevant status as, say, an interval algebra, for which path consistency has been determined.

A second strategy for tackling the fragmentation problem is derived from the flexibility of our account that allows for switching back and forth between coarse and fine-grained levels of temporal reasoning. Adapting from a coarse level of reasoning to a finer grain size is very often given by the identity operation [11], while the operators π and μ mostly lift reasoning onto a coarser level.

[11] One notable exception arises when granularity levels are not directly comparable,

The combination of both renders a powerful computational strategy. Given a problem that requires representations and computations at an expressive level, one may perform computations at a coarse and cheap level of reasoning first (*e.g.*, consistency in point algebra), and hence find all the easy results early and easily. Taking full advantage of the easy computations, one may map the results back to the fine-grained level, proceeding with reasoning at the expressive level in order to determine the hard results, too. Thus, trading in this way between expressiveness and efficiency allows to solve the easy tasks easily, while rendering the hard tasks not impossible.

8 ABSTRACTION AT THE INTERFACE LEVEL — EXPRESSIVENESS VS. UNDERSTANDABILITY

Increased expressiveness and the application of powerful abstraction mechanisms that mediate between different precision levels of reasoning may actually aggravate the application of a temporal reasoning system. While thirteen primitive interval relations in Allen's calculus or disjunctions of interval constraints in TCSPs may already pose non-trivial problems for a human to deal with, GTN relations have an even more complicated structure. Thus, GTN relations are often too unwieldy to be used in a temporal query language or by a module of a larger system, though an application may actually require their use. For instance, a text understanding and generation system dealing with the scheduling problem as given in (1) may need to account for complex propositions such as (2g). This means that high-level conceptual representation structures, e.g., *"a very long flight"* or *"X arrives after Y"*, that are typically employed by such a system must be translated to low-level GTN expressions when in-depth temporal reasoning is required.

To bridge the conceptual distance, we here introduce an interface level that abstracts from unnecessary details and, hence, generalizes to the *relevant* distinctions that need to be made. In doing so, we provide definitions of abstracting relations that are used to move from the interface level down to the reasoning level – and in the reverse direction. Switching from the interface to the reasoning level, e.g., when posing a query to a temporal reasoning system, one simply has to expand the definition of the abstracting relation. Table 1 shows some examples of such "macro" definitions.

Switching back, i.e., outputting an abstracted relation to the interface level, e.g., as an answer to a query posed by a "naive" user, requires reasonable criteria for the selection of those interface relations that are best suited to abstract from a given low-level relation. We here define two notions of "best

e.g., month *vs.* week (cf. [5]).

Table 1
A Sample of Abstracting Relations

A [12]	Interval A **meets** interval B with **tolerance** d
	$(A_e, (-d, d), B_b) \wedge (A_b, (0, \infty), B_b) \wedge (A_e, (0, \infty), B_e)$
B	Interval A **is between** interval B and interval C
	$((A_b, (-\infty, 0), B_e) \wedge (A_e, (0, +\infty), C_b)) \vee$
	$((A_b, (-\infty, 0), C_e) \wedge (A_e, (0, +\infty), B_b))$
C	Interval A **is at least** n **units disjoint** from B
	$(A_e, [n, +\infty), B_b) \vee (A_b, (-\infty, -n], B_e)$
D	**If** time point a **before** time point b **then** time point c **before** time point d
	$((a, (0, +\infty), b) \wedge (c, (0, +\infty), d)) \vee (a, (-\infty, 0], b)$
E	Time points a, b, c appear **in this order**.
	$(a, (0, +\infty), b) \wedge (b, (0, +\infty), c)$
F	Time point a **is between** time point b and time point c
	$((a, (-\infty, 0), b) \wedge (a, (0, +\infty), c)) \vee ((a, (0, +\infty), b) \wedge (a, (-\infty, 0), c))$
G	Time point a **being at least** d **after** time point b
	correlates with time point a **being at least** d **after** c
	$(b, [d, +\infty), a) \wedge (c, [d, +\infty), a)$

approximations":

Definition 22 *Let a set of abstracting relations be given by $R_1^a, \ldots, R_n^a$.*

A relation R_i^a is a smallest upper approximation of a relation R with regard to $R_1^a, \ldots, R_n^a$, iff $R_i^a \trianglerighteq R$ and there is no $R_j^a, i \neq j$ such that $R_i^a \trianglerighteq R_j^a \trianglerighteq R$.

A relation R_i^a is a greatest lower approximation of a relation R with regard to $R_1^a, \ldots, R_n^a$, iff $R_i^a \trianglelefteq R$ and there is no $R_j^a, i \neq j$ such that $R_i^a \trianglelefteq R_j^a \trianglelefteq R$.

This definition may yield several smallest upper and greatest lower approximations. A unique upper approximation is given by the conjunction of the best upper bounds, while a unique lower approximation is given by the disjunction of the best lower bounds.

We do not present an algorithm here for computing the approximating rela-

[12] For illustration of some of the scope of these definitions, macro A is also graphically indicated by the intersection of its three primitive constraints, $(A_e, (-d, d), B_b), (A_b, (0, \infty), B_b), (A_e, (0, \infty), B_e)$, in Fig. 3.

tions, since its appropriateness depends heavily on the abstracting relations being given and the temporal reasoning system being used. Three obvious problems may illustrate these interdependencies: First, for abstracting relations with quantitative parameters the proper instantiation of free parameters with actual values in the corresponding relation allows for redundant variation. Symmetric relations like *"time point t_1 is at most 1 unit away from t_2"* require particular care, since the equivalent *"time point t_2 is at most 1 unit away from t_1"* does not yield any new information. Second, additional constraints are needed to control proper instantiation of an abstracting relation. For instance, the definition A in Table 1 should be supplemented by the ontological restriction that A_b and A_e really form an interval. Though, in principle, all pairs of time points may determine an interval that one could talk about, in practice, this generality should be avoided. Third, another additional constraint considered plausible for all abstracting relations is the unique name assumption which prevents, e.g., the unification of the three variables a, b, c in the abstracting relation G from Table 1.

Let us now illustrate our notion of generalization with two examples. Assume we want to mine the GTN resulting from (2) for interesting complex rules. For our first example, we are interested in temporal rules on how the arrival time of Mr. Roget influences the schedule of Mrs. Meyer appearing after him. Then, we add an unconstrained relation R_z to the GTN with $E_z := \{(t_0, t_1), (t_1, t_3)\}$. Composing the relation given in (2g) with the one from (2h) and projecting the result onto R_z yields:

(9) $((t_0, [3, 3], t_1) \wedge (t_1, (2, 6), t_3)) \vee ((t_0, [6, 6], t_1) \wedge (t_1, (0, 4), t_3))$

Generalizing this relation, obviously, only the abstracting relations E, F and G may apply (cf. Table 1), since the other ones require intervals instead of time points (e.g., A, B and C) or a different number of time points (*viz.* four as in D). Approximating "from above", abstracting relation G does not generalize (9) at all, while *"Time points t_0, t_1, t_3 appear in this order"* is the best generalization, since it is more specific than the corresponding instantiation of F. An approximation "from below" fails, because none of the abstracting relations is more specific than the relation in example (9).

Correspondingly, we may ask how Mr. George's arrival correlates with those of Mr. Roget and Mr. Meyer. Given that we have only qualitative information about the length of Mr. George's flight, it seems most appropriate to reason entirely on a qualitative interval structure. For the sake of brevity, we may here ignore many of the intricating presuppositions involved in algebraic operations on qualitative durations (cf. [7]) and simply present the result derived from the corresponding inference process:

(10) $((t_1, [\text{"medium"}, +\infty), t_5) \wedge (t_2, [\text{"medium"}, +\infty), t_5))$

This result is generalized ("from above *and* below") by *"Time point t_5 being at least a medium time after t_1 correlates with t_5 being at least a medium time after time point t_2"*.

Conceptualizations at the interface level are of particular value for combining single evidence and generalizing it. In our text understanding application, e.g., we represent graded information like *"hard disk A is faster than hard disk B"* by GTN relations (cf. Hahn *et al.* [16,41]). Most of these relations can be handled by a comparatively inexpensive representation formalism. However, we also have to deal with much more complex utterances like *"up from a block size of 32 KB the data throughput decreases from 800 KB/s to less than 600 KB/s"*, which require more expressive representations, and, hence, costly reasoning. By flexibly assigning reasoning tasks to the least expensive representation level the entire understanding process might still be executed within feasible bounds. When just few of the represented GTN relations are complex, which is the case most of the time, reasoning at the finer levels remains feasible. Only if complicated GTN relations abound, one must resort to reasoning at coarser levels as an approximation — and eventually to an abstracting interface level that makes generalizations accessible to the user instead of a myriad of tiny bits of detail.

9 RELATED WORK

Levels of granularity of temporal reasoning, as *static* notions, pervade the heterarchy of calculi discussed in Section 7. This derives from the fact that these constraint systems stand for different levels of expressiveness. As the arrows in Fig. 6 indicate there are rather limited calculi (e.g., point algebra [45]), ones with increased expressiveness (e.g., Allen's calculus [2] or TCSPs [9]) and fairly general ones (such as integration models for Allen's calculus with metrical reasoning [23,27,3]). As a framework for our research, we have introduced a very general model, *viz.* Generalized Temporal Networks (GTNs). Its expressiveness exceeds that of all previously mentioned calculi, since it allows for the description of non-binary relations. It scales up smoothly from binary relations, including the propagation of their quantitative and qualitative constraints in the network as well as the computation of the minimal network according to different granularities. Thus, our temporal reasoning scheme lays down the foundations for formalizing temporal constraints at different levels of granularity. Weaker constraint systems may be an appropriate choice for applications which require less specific constraints and offer on their bonus side the tractability for certain reasoning algorithms.

Using this trade-off between expressiveness and computational complexity in a strategic manner leads to the idea to navigate this graph of different levels of expressiveness on demand — depending on the needs of the particular application. The idea to offer a new expressive temporal reasoning scheme, *viz.* GTNs, that allows for *dynamic* shifts between less expressive and computationally cheaper systems and more expressive though computationally more expensive ones during run-time is the starting point of our work, and has been on the research agenda for quite a long time (cf. Hobbs [17], Sathi *et al.* [36], Nakhimovsky [30], Meiri [27]). This flexible manouvering between granularities as a principle method rather than as an impeding side condition constitutes the main difference between our approach and common reasoning systems that implement several metric systems.

For instance, Bettini *et al.* [5] have extended STP networks in order to represent interval structures from a large range of granularity levels. Thereby, they have even included non-contiguous structures (*e.g.*, business days). As an approximating reasoning algorithm they propagate constraints in parallel networks of single granularities. Operators that map constraints between granularities communicate between the different networks. However, propositional abstraction, such as defined by our operator π is neglected in their approach as well as in other temporal reasoning systems (along similar lines also cf. earlier work by Chandra *et al.* [6] and Dean [8]).

This negligence may even be a drawback with regard to performance issues. Approaches for efficient temporal reasoning use, e.g., approximating propagation mechanisms (cf. Schwalb & Dechter [37]) or heuristics that optimize the search process (cf. Stergiou & Koubarakis [43]). Though our proposal still lacks comparable empirical evidence, we can guarantee the determination of criteria important for the inferencing task (e.g., consistency for point algebra, path consistency for qualitative relations) in polynomial time, when granularities are switched to compute coarser results first, and refinements at more precise levels are postponed to subsequent rounds. Optimized schemes like those in [37,43] may still not terminate and, if they are terminated from outside due to exhausted time budgets (as set up by anytime devices, cf. Russell & Zilberstein [34]), the network cannot be asserted to be in a similarly well-defined state as a cascade of GTNs at different granularities.

A complementary proposal has been made by Euzenat [11], who permits to represent seemingly contradictory information at different levels of granularity, e.g., at some given level one may perceive that two intervals meet, while at a finer level one may recognize that the first is just a tiny bit before the second. His abstraction operators reflect how perception may change by switching between different levels.

Several other temporal reasoning proposals do not consider granularity issues at all, but are interesting due to their expressive reasoning facilities. For instance, Navarrete & Marin [31], Wetprasit & Sattar [46] and Pujari et al. [33,32] extend the time point algebra (cf. [45]) by comparisons on distances, which our approach does not allow for. However, they are complementary to GTNs, because they cannot express non-binary relations. Jonsson & Bäckström [20,21] and Koubarakis [24] use networks where each relation is *Horn*, meaning that at most one positive literal must exist per conjunction. This way, a scale-up is achieved from subclasses of Allen's calculus to interval relations with quantities where consistency can be determined in polynomial time. Its disadvantage is that disjunctions of two-sided restrictions, e.g., $(a_1 \leq b \wedge b \leq c_1) \vee (a_2 \leq b \wedge b \leq c_2)$, cannot be formulated. One may speculate whether these classes could be used to backtrack efficiently in GTNs.

As for more general frameworks, there has been a surge of interest in non-binary constraint problems, recently. For instance, general constraint problems are handled by Faltings & Gelle [12] and Bessiere & Regin [4]. They compute *arc consistency* for non-convex higher-arity constraint networks. However, global consistency is hard to tackle at this point and not achieved for these general problems. Sam-Haroud & Faltings [35] treat ternary constraint problems (noting that relations of general arity can be transformed into ternary ones) and define *(3,2)-relational consistency* as a generalization of *binary path consistency*. For temporal reasoning applications its main drawback is its restriction to (almost) convex relations which prohibits expressions like *"disjoint by more than n units"* or *"if a before b then c before d"*. Naturally, this line of research neglects the actual algebraic operations on higher-arity *temporal* relations, like composition and intersection, and their implications which are given in our proposal (along the same lines cf. [10]).

To sum up, none of these approaches [31,33,20,21,24,12,4,35,10] uses abstraction – neither for efficiency nor for understandability purposes — such as we do. There exist few approaches to temporal abstraction, e.g., cf. Shahar & Cheng [38]. However, their abstraction does not move along the heterarchy of temporal reasoning mechanisms, such as our proposal does. Rather they conceive of summaries that mostly depend on ontological knowledge other than temporal data, e.g., two occurrences of anemia may be summarized into one medical description. Thus, our proposal nicely complements their approach.

10 CONCLUSION

This work has been motivated by two goals: On the one hand, there has been the urgent need to integrate non-binary relations into temporal reasoning in order to extend its range to common reasoning problems. On the other hand, we wanted to make some progress toward a "*Great Unified Theory of Temporal Reasoning*". Several popular versions of temporal reasoning turned out to be derivable from just a few parameter settings in our more general framework.

We identified three dimensions which are crucial with regard to the trade-offs between expressiveness and efficiency as well as between expressiveness and understandability, namely *interval structures for constraints*, *relation topology*, and *network topology*. We investigated how major tasks in temporal networks, *viz.* constraint propagation, determination of consistency, and computation of the minimal equivalent network, are affected by these parameters. Operators, *viz.* $\pi, \mu, \trianglerighteq$, provided for switching smoothly along the three dimensions between different granularities as far as the reasoning proper and communication via interfaces are concerned. Abstraction at the interface level has been achieved by approximating temporal relations with macro definitions, a research issue that to the best of our knowledge has not been dealt with so far.

Obviously, complementary dimensions exist which give rise to further possibilities and difficulties, e.g., non-binary constraints not representable in our scheme (e.g., $a-b=c-d+1$; cf. [20,24]), comparisons between distances (cf. Navarrete & Marin [31]) or infinitly repeated structures (cf. Morris *et al.* [29]). Beginning with the simplest model, which we assume to be a point algebra, and pursuing an extension into further dimensions, one reaches regions of NP-hardness for determining consistency of a network very fast. Thus, a unifying theory should not subscribe to a "one method fits all needs" policy, but it should rather provide a family of methods the interdependencies of which are well understood and accessible for switching between them, such that their advantages may be combined — similarly as in our approach.

The major open issue is then *when* to bring *what* level of abstraction into play. In our opinion, there is no general solution to this problem. In the research environment we work in, a natural language text understanding system, the appropriate choice of adequate abstraction levels often comes with the author's choice of specific linguistic expressions occurring in the text, their corresponding semantic interpretation and the progression of the text (cf. Matsushita et al. [25]). Having fixed such a starting point, we proceed from the cheapest level possible and turn to more expressive and expensive levels only when this is needed for proper text understanding.

Acknowledgements

The research reported in this paper has been funded by a DFG scholarship in the graduate program on "Human and Artificial Intelligence at Freiburg University and by a BMBF grant (number 01IN802; project "GETESS") at Karlsruhe University. Special thanks go to my former advisor, Udo Hahn, and my colleagues, Waltraud Hiltl, Katja Markert, Susanne Schacht, Holger Schauer, Klemens Schnattinger, and Stefan Schulz for valuable comments during different stages of this paper, to Oliver Karch for inspiring discussions regarding the optimization issues, and to two anonymous reviewers for giving hints to improve the presentation.

References

[1] *AAAI '98/IAAI '98: Proceedings of the 16th National Conference on Artificial Intelligence & Innovative Applications of Artificial Intelligence Conference. Madison, Wisconsin, July 26-30, 1998*, Menlo Park, CA; Cambridge, MA, 1998. AAAI Press; MIT Press.

[2] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[3] S. Badaloni and M. Berati. Hybrid temporal reasoning for planning and scheduling. In *Time-96: Proc. of the 3rd International Workshop on Temporal Representation and Reasoning*, Los Alamitos, CA, 1996. IEEE Computer Society.

[4] C. Bessiere and J.C. Régin. Arc consistency for general constraint networks: preliminary results. In IJCAI [18], pages 398–404.

[5] C. Bettini, X.S. Wang, and S.S. Jajodia. Satisfiability of quantitative temporal constraints with multiple granularities. In G. Smolka, editor, *Principles and Practice of Constraint Programming — CP '97*, LNCS 1330, pages 435–445. Springer, 1997.

[6] R. Chandra, A. Segev, and M. Stonebreaker. Implementing calendars and temporal rules in next generation databases. In *Proc. of the International Conference on Data Engineering (ICDE)*, pages 264–273, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[7] E. Clementini, P. Di Felice, and D. Hernández. Qualitative representation of positional information. *Artificial Intelligence*, 95(2):317–356, 1997.

[8] T. Dean. Artificial intelligence: Using temporal hierarchies to efficiently maintain large temporal databases. *Journal of the ACM*, 36(4):687–718, 1989.

[9] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95, 1991.

[10] Regin *et al.*, editor. *Proc. of the Workshop on Non-Binary Constraints at IJCAI-99*. IJCAI Org., 1999.

[11] J. Euzenat. An algebraic approach for granularity in qualitative space and time representation. In *IJCAI-95: Proc. of the 14th International Joint Conference on Artificial Intelligence. Montreal, CA. August 20-25, 1995*, pages 849–900, San Francisco, CA, 1995. Morgan Kaufmann.

[12] B. Faltings and E. Gelle. Local consistency for ternary numeric constraints. In IJCAI [18], pages 392–397.

[13] C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1-2):199–227, 1992.

[14] A. Gerevini and L. Schubert. On point-based temporal disjointness. *Artificial Intelligence*, 70(1-2), 1994.

[15] F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 56(2-3):323–390, 1992.

[16] U. Hahn, M. Romacker, and S. Schulz. How knowledge drives understanding: matching medical ontologies with the needs of medical language processing. *Artificial Intelligence in Medicine*, 15(1):25–51, 1999.

[17] J.R. Hobbs. Granularity. In *IJCAI-85: Proc. of the 9th International Joint Conference on Artificial Intelligence*, pages 432–435. Morgan Kaufmann, 1985.

[18] *IJCAI-97: Proc. of the 15th International Joint Conference on Artificial Intelligence. Nagoya, Japan. August 23-29, 1997*, San Francisco, CA, 1997. Morgan Kaufmann.

[19] *IJCAI-99: Proc. of the 16th International Joint Conference on Artificial Intelligence. Stockholm, Sweden. August 1-7, 1999*, San Francisco, CA, 1999. Morgan Kaufmann.

[20] P. Jonsson and C. Bäckström. A linear programming approach to temporal reasoning. In *AAAI '96/IAAI '96: Proceedings of the 14th National Conference on Artificial Intelligence & Innovative Applications of Artificial Intelligence Conference*, pages 1235–1240, Menlo Park, CA; Cambridge, MA, 1996. AAAI Press; MIT Press.

[21] P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102:143–155, 1998.

[22] P. Jonsson, T. Drakengren, and C. Bäckström. Computational complexity of relating time points with intervals. *Artificial Intelligence*, 109(1-2):273–295, 1999.

[23] H.A. Kautz and P.B. Ladkin. Integrating metric and qualitative temporal reasoning. In *AAAI '91: Proceedings of the 9th National Conference on Artificial Intelligence*, pages 241–246, Menlo Park, CA; Cambridge, MA, 1991. AAAI Press; MIT Press.

[24] M. Koubarakis. Tractable disjunctions of linear constraints. In E. C. Freuder, editor, *Principles and Practice of Constraint Programming — CP '96*, LNCS 1118, pages 297–307. Springer, 1996.

[25] M. Matsushita, M. Ohta, and T. Iida. A visualization method of time expressions using starting/ending point plane. In Time-98 [44], pages 162–168.

[26] D. McGuiness and P. Patel-Schneider. Usability issues in knowledge representation systems. In AAAI [1], pages 608–614.

[27] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence*, 87(1-2):343–385, 1996.

[28] U. Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

[29] R.A. Morris, W.D. Shoaff, and L. Khatib. Path consistency in a network of non-convex intervals. In *IJCAI-93: Proc. of the 13th International Joint Conference on Artificial Intelligence. Chambery, France, August 28 - September 3, 1993*, pages 655–660, San Mateo, CA, 1993. Morgan Kaufmann.

[30] A. Nakhimovsky. Aspect, aspectual class, and the temporal structure of narrative. *Computational Linguistics*, 14(2):29–43, 1988.

[31] I. Navarrete and R. Marin. Qualitative temporal reasoning with points and durations. In IJCAI [18], pages 1454–1459.

[32] A. Pujari, G.V. Kumari, and A. Sattar. INDU: An interval and duration network. In *Australian Joint Conference on Artificial Intelligence*, pages 291–303, 1999.

[33] A. Pujari and A. Sattar. A new framework for reasoning about points, intervals and durations. In IJCAI [19], pages 1259–1264.

[34] S.J. Russell and S. Zilberstein. Composing real-time systems. In *IJCAI-91: Proc. of the 12th International Joint Conference on Artificial Intelligence. Sidney, Australia, August 24-30, 1991*, pages 212–217, San Mateo, CA, 1991. Morgan Kaufmann.

[35] D. Sam-Haroud and B.V. Faltings. Solving non-binary convex CSPs in continuous domains. In E. C. Freuder, editor, *Principles and Practice of Constraint Programming — CP '96*, LNCS 1118, pages 410–424. Springer, 1996.

[36] A. Sathi, M.S. Fox, and M. Greenberg. Representation of activity knowledge for project management. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7(5), 1985.

[37] E. Schwalb and R. Dechter. Processing disjunctions in temporal constraint networks. *Artificial Intelligence*, 93(1-2):29–61, 1997.

[38] Y. Shahar and C. Cheng. Model-based visualization of temporal abstractions. In Time-98 [44], pages 11–20.

[39] S. Staab. On non-binary temporal relations. In *ECAI-98: Proc. of the 13th European Conference on Artificial Intelligence. August 23-28, Brighton, UK*, pages 567–571, Chichester, 1998. Wiley.

[40] S. Staab. *Grading Knowledge — Extracting Degree Information from Texts.* LNAI 1744. Springer, Heidelberg, 1999.

[41] S. Staab and U. Hahn. “Tall”, “good”, “high” — Compared to What? In IJCAI [18], pages 996–1001.

[42] S. Staab and U. Hahn. Scalable temporal reasoning. In IJCAI [19], pages 1247–1252.

[43] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. In AAAI [1].

[44] *Time-98: Proc. of the 5^{th} International Workshop on Temporal Representation and Reasoning*, Los Alamitos, CA, 1998. IEEE Computer Society.

[45] M. Vilain, H. Kautz, and P. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In D.S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physics*, pages 373–381. Morgan Kaufmann, 1989.

[46] R. Wetprasit and A. Sattar. Temporal reasoning with qualitative and quantitative information about points and durations. In AAAI [1], pages 656–663.