Checking Dynamic Consistency of Conditional Hyper Temporal Networks via Mean Payoff Games

Hardness and (pseudo) Singly-Exponential Time Algorithm

Carlo Comin[a,b], Romeo Rizzi[c]

[a]*Department of Mathematics, University of Trento, Trento, Italy*
[b]*LIGM, Université Paris-Est Marne-la-Vallée, Champs-sur-Marne, Paris, France*
[c]*Department of Computer Science, University of Verona, Verona, Italy*

Abstract

Conditional Simple Temporal Network (CSTN) is a constraint-based graph-formalism for conditional temporal planning. It offers a more flexible formalism than the equivalent CSTP model of Tsamardinos et al. (2003), from which it was derived mainly as a sound formalization. Three notions of consistency arise for CSTNs: weak, strong, and dynamic. Dynamic consistency is the most interesting notion, but it is also the most challenging and it was conjectured to be hard to assess. Tsamardinos et al. (2003) gave a doubly-exponential time algorithm for checking dynamic consistency in CSTNs and to produce an exponentially sized dynamic execution strategy whenever the input CSTN is dynamically-consistent. CSTNs may be viewed as an extension of Simple Temporal Networks (STNs) (Dechter et al., 1991), directed weighted graphs where nodes represent events to be scheduled in time and arcs represent temporal distance constraints between pairs of events. Recently, STNs have been generalized into *Hyper Temporal Networks* (HyTNs), by considering weighted directed hypergraphs where each hyperarc models a *disjunctive* temporal constraint named *hyperconstraint*; being directed, the hyperarcs can be either *multi-head* or *multi-tail*. The computational equivalence between checking consistency in HyTNs and determining winning regions in Mean Payoff Games (MPGs) was also pointed out; MPGs are a family of 2-player infinite pebble games played on finite graphs, which is well known for having applications in model-checking and formal verification. In this work we introduce the *Conditional Hyper Temporal Network (CHyTN)* model, a natural extension and generalization of both the CSTN and the HyTN model which is obtained by blending them together. We show that deciding whether a given CSTN or CHyTN is dynamically-consistent is coNP-hard; and that deciding whether a given CHyTN is dynamically-consistent is PSPACE-hard, provided that the input instances are allowed to include both multi-head and multi-tail hyperarcs. In light of this, we continue our study by focusing on CHyTNs that allow only multi-head hyperarcs, and we offer the first deterministic (pseudo) singly-exponential time algorithm for the problem of checking the dynamic consistency of such CHyTNs, also producing a dynamic execution strategy whenever the input CHyTN is dynamically-consistent. Since CSTNs are a special case of CHyTNs, as a byproduct this provides the first sound-and-complete (pseudo) singly-exponential time algorithm for checking dynamic consistency in CSTNs. The proposed algorithm is based on a novel connection between CHyTNs and MPGs; due to the existence of efficient pseudo-polynomial time algorithms for MPGs, it is quite promising to be competitive in practice. The presentation of such connection is mediated by the HyTN model. In order to analyze the time complexity of the algorithm, we introduce a refined notion of dynamic consistency, named ϵ-dynamic consistency, and present a sharp lower bounding analysis on the critical value of the reaction time $\hat{\varepsilon}$ where a CHyTN transits from being, to not being, dynamically-consistent. The proof technique introduced in this analysis of $\hat{\varepsilon}$ is applicable more generally when dealing with linear difference constraints which include strict inequalities.

Keywords: Conditional Temporal Networks, Dynamic Consistency, Mean Payoff Games, Simple Temporal Networks, Hyper Temporal Networks, Singly-Exponential Time, Reaction Time.

Email addresses: `carlo.comin@unitn.it` (Carlo Comin), `romeo.rizzi@univr.it` (Romeo Rizzi)

1. Introduction and Motivation

In many areas of Artificial Intelligence (AI), including temporal planning and scheduling, the representation and management of quantitative temporal aspects is of crucial importance (see e.g., Pani and Bhattacharjee (2001); Smith et al. (2000); Eder et al. (1999); Bettini et al. (2002); Combi et al. (2012, 2014)). Examples of possible quantitative temporal aspects include constraints on the earliest start time and latest end time of activities and constraints over the minimum and maximum temporal distance between activities. In many cases these constraints can be represented by *Simple Temporal Networks* (STNs) (Dechter et al., 1991), i.e., directed weighted graphs where nodes represent events to be scheduled in time and arcs represent temporal distance constraints between pairs of events. Recently, STNs have been generalized into *Hyper Temporal Networks* (HyTNs) (Comin et al., 2014, 2016), a strict generalization of STNs introduced to overcome the limitation of considering only conjunctions of constraints, but maintaining a practical efficiency in the consistency checking of the instances. In a HyTN a single temporal hyperarc constraint is defined as a set of two or more maximum delay constraints which is satisfied when at least one of these delay constraints is satisfied. HyTNs are meant as a light generalization of STNs offering an interesting compromise. On one side, there exist practical pseudo-polynomial time algorithms for checking the consistency of HyTNs and computing feasible schedules for them. On the other side, HyTNs offer a more powerful model accommodating natural disjunctive constraints that cannot be expressed by STNs. In particular, HyTNs are weighted directed hypergraphs where each hyperarc models a disjunctive temporal constraint called *hyperconstraint*. The computational equivalence between checking consistency in HyTNs and determining winning regions in *Mean Payoff Games* (MPGs) (Ehrenfeucht and Mycielski, 1979; Zwick and Paterson, 1996; Brim et al., 2011) was also pointed out in Comin et al. (2014, 2016), where the approach was shown to be robust thanks to experimental evaluations (also see Brim and Chaloupka (2012)). MPGs are a family of 2-player infinite pebble games played on finite graphs which is well known for having theoretical interest in computational complexity, being one of the few natural problems lying in NP ∩ coNP, as well as various applications in model checking and formal verification (Grädel et al., 2002).

However, in the representation of quantitative temporal aspects of systems, *conditional* temporal constraints pose a serious challenge for conditional temporal planning, where a planning agent has to determine whether a candidate plan will satisfy the specified conditional temporal constraints. This can be difficult, because the temporal assignments that satisfy the constraints associated with one conditional branch may fail to satisfy the constraints along a different branch (see, e.g., Tsamardinos et al. (2003)). The present work unveils that HyTNs and MPGs are a natural underlying combinatorial model for checking the consistency of certain conditional temporal problems that are known in the literature and that are useful in some practical applications of temporal planning, especially, for managing the temporal aspects of Workflow Management Systems (WfMSs) (Bettini et al., 2002; Combi et al., 2012) and for modeling Healthcare's Clinical Pathways (Combi et al., 2014). Thus we focus on *Conditional Simple Temporal Networks (CSTNs)* (Tsamardinos et al., 2003; Hunsberger et al., 2012), a constraint-based model for conditional temporal planning. The CSTN formalism extends STNs in that: (1) some of the nodes are *observation events*, to each of them is associated a boolean variable whose value is disclosed only at execution time; (2) *labels* (i.e. conjunctions over the literals) are attached to all nodes *and* constraints, to indicate the situations in which each of them is required. The planning agent (or *Planner*) must schedule all the required nodes, meanwhile respecting all the required temporal constraints among them. This extended framework allows for the off-line construction of conditional plans that are guaranteed to satisfy complex networks of temporal constraints. Importantly, this can be achieved even while allowing for the decisions about the precise timing of actions to be postponed until execution time, in a least-commitment manner, thereby adding flexibility and making it possible to adapt the plan dynamically, in response to the observations that are made during execution. See Tsamardinos et al. (2003) for further details and examples.

Three notions of consistency arise for CSTNs: weak, strong, and *dynamic*. Dynamic consistency (DC) is the most interesting one; it requires the existence of conditional plans where decisions about the precise timing of actions are postponed until execution time, but it nonetheless guarantees that all the relevant constraints will be ultimately satisfied. Still, it is the most challenging and it was conjectured to be hard to assess by Tsamardinos et al. (2003). Indeed, to the best of our knowledge, the tightest currently known upper bound on the time complexity of deciding whether a given CSTN is dynamically-consistent is doubly-exponential time (Tsamardinos et al., 2003). It first builds an equivalent Disjunctive Temporal Problem (DTP) of size exponential in the input CSTN, and then applies to it an exponential-time DTP solver to check its consistency. However, this approach turns out to be quite limited in practice: experimental studies have already shown that the resolution procedures, as well as the currently known heuristics,

for solving general DTPs become quite burdensome with 30 to 35 DTP variables (see e.g., Tsamardinos and Pollack (2003); Moffitt and Pollack (2005); Oddi (2014)), thus dampening the practical applicability of the approach.

1.1. Contribution

In this work we introduce and study the *Conditional Hyper Temporal Network (CHyTN)* model, a natural extension and generalization of both the CSTN and the HyTN model which is obtained by blending them together. One motivation for studying it is to transpose benefits and opportunities for application, that have arisen from the introduction of HyTNs (see Comin et al. (2014, 2016)), to the context of *conditional* temporal planning. In so doing, the main and perhaps most important contribution is that to offer the first sound-and-complete deterministic (pseudo) singly-exponential time algorithm for checking the dynamic consistency of CSTNs. After having formally introduced the CHyTN model, we start by showing that deciding whether a given CSTN or CHyTN is dynamically-consistent is coNP-hard. Then, we offer a proof that deciding whether a given CHyTN is dynamically-consistent is PSPACE-hard, provided that the input CHyTN instances are allowed to include both multi-head and multi-tail hyperarcs. In light of this, we focus on CHyTNs that allow only multi-head hyperarcs. Concerning multi-head CHyTNs, perhaps most importantly, we unveil a connection between the problem of checking their dynamic consistency and that of determining winning regions in MPGs (of a singly-exponential size in the number of propositional variables of the input CHyTN), thus providing the first sound-and-complete (pseudo) singly-exponential time algorithm for this same task of deciding the dynamic consistency and yielding a dynamic execution strategy for multi-head CHyTNs. The resulting worst-case time complexity of the DC-Checking procedure is actually $O\big(2^{3|P|}|V||\mathcal{A}|m_{\mathcal{A}} + 2^{4|P|}|V|^2|\mathcal{A}||P| + 2^{4|P|}|V|^2 m_{\mathcal{A}} + 2^{5|P|}|V|^3|P|\big)W$, where $|P|$ is the number of propositional variables, $|V|$ is the number of event nodes, $|\mathcal{A}|$ is the number of hyperarcs, $m_{\mathcal{A}}$ is the size (i.e., roughly, the encoding length of $\mathcal{A}$), and W is the maximum absolute integer value of the weights of the input CHyTN. The algorithm is still based on representing a given CHyTN instance on an exponentially sized network, as first suggested in Tsamardinos et al. (2003). The difference, however, is that we propose to map CSTNs and CHyTNs on (exponentially sized) HyTNs/MPGs rather than on DTPs. This makes an important difference, because the consistency check for HyTNs can be reduced to determining winning regions in MPGs, as shown in Comin et al. (2014, 2016), which admits practical and effective pseudo-polynomial time algorithms (in some cases the algorithms for determining winning regions in MPGs exhibit even a strongly polynomial time behaviour, see e.g., Comin et al. (2016); Brim and Chaloupka (2012); Allamigeon et al. (2014); Chatterjee et al. (2014)). To summarize, we obtain an improved upper bound on the theoretical time complexity of the DC-checking of CSTNs (i.e., from 2-EXP to pseudo-$E \cap \text{NE} \cap \text{coNE}$) together with a faster DC-checking procedure, which can be used on CHyTNs with a larger number of propositional variables and event nodes than before. At the heart of the algorithm a suitable reduction to MPGs is mediated by the HyTN model, i.e., the algorithm decides whether a CHyTNs is dynamically-consistent by solving a carefully constructed MPG. In order to analyze the algorithm, we introduce a novel and refined notion of dynamic consistency, named ϵ-dynamic consistency (where $\epsilon \in \mathbf{R}_+$), and present a sharp lower bounding analysis on the critical value of the *reaction time* $\hat{\varepsilon}$ where a CHyTNs transits from being, to not being, dynamically-consistent. We believe that this contributes to clarifying (w.r.t. some previous literature, e.g., Tsamardinos et al. (2003); Hunsberger et al. (2012)) the role played by the reaction time $\hat{\varepsilon}$ in checking the dynamic consistency of CSTNs. Moreover, the proof technique introduced in this analysis of $\hat{\varepsilon}$ is applicable more generally when dealing with linear difference constraints which include strict inequalities; thus it may be useful in the analysis of other models of temporal constraints.

A preliminary version of this article appeared in the proceedings of the TIME symposium (Comin and Rizzi, 2015). Here, the presentation is extended as follows: (1) the definition of CSTN has been extended and generalized to that of CHyTN in order to allow the presence of hyperarcs as labeled temporal constraints already in the input instances; (2) some further facts and pertinent properties about CSTNs and CHyTNs have been established; (3) for instance, the following hardness result: deciding whether a given CHyTN is dynamically-consistent is PSPACE-hard (the reduction goes from 3-CNF-TQBF), provided that the input instances are allowed to include both multi-head and multi-tail hyperarcs; (4) the proposed (pseudo) singly-exponential time algorithm is presented here in its full generality, i.e., w.r.t. the CHyTN model; (5) several proofs have been polished, expanded and clarified (e.g., those concerning the reaction time analysis of $\hat{\epsilon}$).

The rest of the article is organized as follows. Section 2 recalls the basic formalism, terminology and known results on STNs and HyTNs. Particularly, Subsection 2.1 deals with STNs; Subsection 2.2 deals with HyTNs, its computational equivalence with MPGs and the related algorithmic results. Section 3 surveys CSTNs and, then, it introduces CHyTNs, also presenting some basic properties of the model. Section 4 tackles on the algorithmics of dynamic consistency: firstly, we provide a coNP-hardness lower bound, then we offer a PSPACE-hardness lower bound. Next, it is described the connection with HyTNs/MPGs and it is devised a (pseudo) singly-exponential time DC-checking algorithm. Section 5 offers a sharp lower bounding analysis on the critical value of the *reaction time* $\hat{\varepsilon}$ where the CSTN transits from being, to not being, dynamically-consistent. In Section 6, related works are discussed. The article concludes in Section 7.

2. Background

2.1. Simple Temporal Networks

Some definitions, notation and well know results about graphs and conservative graphs are introduced below; also, we recall the relation between the consistency property of STNs and the conservative property of weighted graphs. Our graphs are directed and weighted on the arcs. Thus, if $G = (V, A)$ is a graph, then every arc $a \in A$ is a triplet (t_a, h_a, w_a), where: $t_a \in V$ is the *tail* of a, $h_a \in V$ is the *head* of a, and $w_a \in \mathbf{R}$ is the *weight* of a. Moreover, since we use graphs to represent distance constraints, they do not need to have either loops (unary constraints are meaningless) or parallel arcs (two parallel constraints represent two different distance constraints between the same pair of nodes: only the most restrictive one is meaningful). We also use the notations $h(a)$ for h_a, $t(a)$ for t_a, and $w(a)$ or $w(t_a, h_a)$ for w_a, when it helps.

The *order* and *size* of a graph $G = (V, A)$ are denoted by $n \triangleq |V|$ and $m \triangleq |A|$, respectively; the size is actually a measure for the encoding length of G. Let $\mathbf{N}_+$ and $\mathbf{R}_+$ be the set of positive natural and positive real numbers, respectively. Let $[n] \triangleq \{1, 2, \ldots, n\}$, for every $n \in \mathbf{N}_+$. A *cycle* of G is a set of arcs $C \subseteq A$ cyclically sequenced as $a_0, \ldots, a_{\ell-1}$ so that $h(a_i) = t(a_j)$ if and only if $j = (i + 1) \mod \ell$; this is called a *negative cycle* if $w(C) < 0$, where $w(C) \triangleq \sum_{e \in C} w_e$. A graph is called *conservative* when it contains no negative cycle. A *potential* is a map $p : V \mapsto \mathbf{R}$. The *reduced weight* of an arc $a = (u, v, w_a)$ w.r.t. a potential p is defined as $w_a^p \triangleq w_a - p_v + p_u$. A potential p of $G = (V, A)$ is called *feasible* if $w_a^p \geq 0$ for every $a \in A$. Notice that, for any cycle C, $w^p(C) = w(C)$. Therefore, the existence of a feasible potential implies that the graph is conservative as $w(C) = w^p(C) \geq 0$ for every cycle C. The Bellman-Ford algorithm (Cormen et al., 2001) can be used to produce in $O(mn)$ time:

    – either a proof that G is conservative in the form of a feasible potential function;

    – or a proof that G is not conservative in the form of a negative cycle C in G.

When the graph is conservative, the *shortest path* between the nodes is well defined, and for a fixed root node r in G the potentials returned by the Bellman-Ford algorithm are, for each node v, the shortest path from r to v. Moreover, if all the arc weights are integers, then these potentials are integers as well. Therefore, the Bellman-Ford algorithm provides a proof of the following theorem.

Theorem 1 (Bellman (1958); Ford and Fulkerson (1962); Cormen et al. (2001)). *A graph admits a feasible potential if and only if it is conservative. When all the arc weights are integer valued, i.e., $w_a \in \mathbf{Z}$ for every $a \in A$, then the feasible potential is integer valued as well.*

An STN can be viewed as a weighted directed simple graph whose nodes are events that must be placed on the real line and whose arcs express mutual constraints on the allocations of their end-points. An STN $G = (V, A)$ is called *consistent* if it admits a *feasible schedule*, i.e., a schedule $s : V \mapsto \mathbf{R}$ such that:

$$s(v) \leq s(u) + w(u, v), \quad \forall \text{ arc } (u, v) \text{ of } G.$$

Corollary 1 (Bellman (1958); Dechter et al. (1991); Cormen et al. (2001)). *An STN G is consistent if and only if G is conservative.*

In this paper, we also deal with directed weighted *hypergraphs*.

Definition 1 (General Hypergraph). *A general hypergraph $\mathcal{H}$ is a pair $(V, \mathcal{A})$, where V is the set of nodes, and $\mathcal{A}$ is the set of* hyperarcs. *Each hyperarc $A \in \mathcal{A}$ is either a* multi-head *or a* multi-tail *hyperarc.*

– A multi-head hyperarc $A = (t_A, H_A, w_A)$ has a distinguished node t_A, called the tail *of A, and a non-empty set $H_A \subseteq V \setminus \{t_A\}$ containing the* heads *of A; to each head $v \in H_A$ is associated a* weight *$w_A(v) \in \mathbf{R}$, which is a real number (unless otherwise specified). Fig. 1a depicts a possible representation of a multi-head hyperarc: the tail is connected to each head by a dashed arc labeled by the name of the hyperarc and the weight associated to the considered head.*

– A multi-tail hyperarc $A = (T_A, h_A, w_A)$ has a distinguished node h_A, called the head *of A, and a non-empty set $T_A \subseteq V \setminus \{h_A\}$ containing the* tails *of A; to each tail $v \in T_A$ is associated a* weight *$w_A(v) \in \mathbf{R}$, which is a real number (unless otherwise specified). Fig. 1b depicts a possible representation of a multi-tail hyperarc: the head is connected to each tail by a dotted arc labeled by the name of the hyperarc and the weight associated to the tail.*

(a) Multi-Head Hyperarc $A = (t_A, H_A, w_A)$.

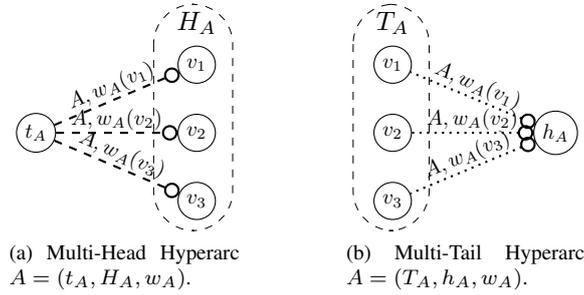(b) Multi-Tail Hyperarc $A = (T_A, h_A, w_A)$.

Figure 1: Multi-Head and Multi-Tail Hyperarcs.

The *cardinality* of a hyperarc $A \in \mathcal{A}$ is given by $|A| \triangleq |H_A \cup \{t_A\}|$ if A is multi-head, and $|A| \triangleq |T_A \cup \{h_A\}|$ if A is multi-tail; if $|A| = 2$, then $A = (u, v, w)$ is a standard arc. The *order* and *size* of a general hypergraph $(V, \mathcal{A})$ are denoted by $n \triangleq |V|$ and $m_{\mathcal{A}} \triangleq \sum_{A \in \mathcal{A}} |A|$, respectively.

2.2. Hyper Temporal Networks

This subsection surveys the *Hyper Temporal Network* (HyTN) model, which is a strict generalization of STNs, introduced to partially overcome the limitation of allowing only conjunctions of constraints. HyTNs have been introduced in Comin et al. (2014, 2016), the reader is referred there for an in-depth treatment of the subject. Compared to STN distance graphs, which they naturally extend, HyTNs allow for a greater flexibility in the definition of the temporal constraints.

A general HyTN is a directed weighted general hypergraph $\mathcal{H} = (V, \mathcal{A})$ where a node represents a time-point variable (or event node), and where a multi-head/multi-tail hyperarc stands for a set of temporal distance constraints between the tail/heads and the head/tails (respectively). Also, we shall consider two special cases of the general HyTN model, one in which all hyperarcs are only multi-head, and one where they're only multi-tail. In general, we say that a hyperarc is *satisfied* when at least one of its distance constraints is satisfied. Then, we say that a HyTN is *consistent* when it is possible to assign a value to each time-point variable so that all of its hyperarcs are satisfied.

More formally, in the HyTN framework the consistency problem is defined as the following decision problem.

Definition 2 (GENERAL-HYTN-CONSISTENCY). *Given a general HyTN $\mathcal{H} = (V, \mathcal{A})$, decide whether there exists a schedule $s : V \to \mathbf{R}$ such that, for every hyperarc $A \in \mathcal{A}$, the following hold:*

- *if $A = (t, h, w)$ is a standard arc, then:*

$$s(h) - s(t) \leq w;$$

- *if $A = (t_A, H_A, w_A)$ is a multi-head hyperarc, then:*

$$s(t_A) \geq \min_{v \in H_A} \{s(v) - w_A(v)\};$$

5

- *if $A = (T_A, h_A, w_A)$ is a multi-tail hyperarc, then:*

$$s(h_A) \leq \max_{v \in T_A} \{s(v) + w_A(v)\}.$$

Any such schedule $s : V \to \mathbf{R}$ is called *feasible*. A HyTN that admits at least one feasible schedule is called *consistent*.

Comparing the consistency of HyTNs with the consistency of STNs, the most important aspect of novelty is that, while in a distance graph of a STN each arc represents a distance constraint and all such constraints have to be satisfied by any feasible schedule, in a HyTN each hyperarc represents a disjunction of one or more distance constraints and a feasible schedule has to satisfy at least one of such distance constraints for each hyperarc.

Let us survey some interesting properties about the consistency problem for HyTNs. The first one is that any integer weighted HyTN admits an integer feasible schedule when it is consistent, as stated in the following lemma.

Lemma 1 (Comin et al. (2016)). *Let $\mathcal{H} = (V, \mathcal{A})$ be an integer weighted and consistent general HyTN. Then $\mathcal{H}$ admits an integer feasible schedule $s : V \to \{-T, -T+1, \ldots, T-1, T\}$, where $T = \sum_{A \in \mathcal{A}, v \in V} |w_A(v)|$.*

The following theorem states that GENERAL-HYTN-CONSISTENCY is NP-complete, in a strong sense.

Theorem 2 (Comin et al. (2016)). GENERAL-HYTN-CONSISTENCY *is an* NP-*complete problem even if input instances $\mathcal{H} = (V, \mathcal{A})$ are restricted to satisfy $w_A(\cdot) \in \{-1, 0, 1\}$ and $|H_A|, |T_A| \leq 2$ for every $A \in \mathcal{A}$.*

Theorem 2 motivates the study of consistency problems on HyTNs having either only multi-head or only multi-tail hyperarcs. In the former case, the consistency problem is called HEAD-HYTN-CONSISTENCY, while in the latter it is called TAIL-HYTN-CONSISTENCY; as stated in Theorem 5, the complexity of checking these two problems turns out to be lower than that for DTPs, i.e., NP ∩ coNP instead of NP-complete.

In the following theorem we observe that the two problems are inter-reducible, i.e., we can check consistency for any one of the two models in $f(m, n, W)$-time whenever we have a $f(m, n, W)$-time procedure for checking consistency for the other one.

Theorem 3 (Comin et al. (2016)). HEAD-HYTN-CONSISTENCY *and* TAIL-HYTN-CONSISTENCY *are inter-reducible by means of* log-*space, linear-time, local-replacement reductions.*

In the rest of this work we shall adopt the multi-head hypergraph as our reference model; but we will consider general hypergraphs again in the forthcoming sections, when proving PSPACE-hardness. Let's say that, when considering hypergraphs and HyTNs, we will be implicitly referring to the multi-head variant unless it is explicitly specified otherwise. So, let us consider the following specialized notion of consistency for HyTNs.

Definition 3 (HEAD-HYTN-CONSISTENCY). *Given a (multi-head) HyTN $\mathcal{H} = (V, \mathcal{A})$, decide whether there exists a schedule $s : V \to \mathbf{R}$ such that:*

$$s(t_A) \geq \min_{v \in H_A} \{s(v) - w_A(v)\}, \quad \forall A \in \mathcal{A}.$$

Remark 1. *Notice that this notion of consistency for HyTNs is a strict generalization of consistency for STNs. Generally, the feasible schedules of an STN are the solutions of a linear system and, therefore, they form a convex polytope. Since an STN may be viewed as a HyTN, the space of feasible schedules of an STN can always be described as the space of feasible schedules of a HyTN. The converse is not true because feasible schedules for a HyTN need not form a convex polytope. Let us consider, for example, a HyTN of just three nodes x_1, x_2, x_3 and a single hyperarc with heads $\{x_1, x_2\}$ and tail x_3 expressing the constraint $x_3 \geq \min\{x_1, x_2\}$. Observe that $(0, 2, 2)$ and $(-2, 0, 2)$ are both admissible schedules, but $(1, 1, 0) = \frac{1}{2}(0, 2, 2) - \frac{1}{2}(-2, 0, 2)$ is not an admissible schedule. In conclusion, the STN model is a special case of the Linear Programming paradigm, whereas the HyTN model is not.*

Next, we extend the characterization of STN consistency (recalled in Subsection 2.1) to HyTNs.

Definition 4 (Reduced Slack Value $w_A^p(v)$). *With reference to a potential $p : V \to \mathbf{R}$, we define, for every arc $A \in \mathcal{A}$ and every $v \in H_A$, the reduced slack value $w_A^p(v)$ as $w_A(v) + p(t_A) - p(v)$ and the reduced slack w_A^p as follows:*

$$ w_A^p \triangleq \max\{w_A^p(v) \mid v \in H_A\}. $$

A potential p is said to be feasible *if and only if $w_A^p \geq 0$ for every $A \in \mathcal{A}$.*

Notice that w_A^p has been defined with $\max$ (instead of $\min$) because if every multi-head hyperarc A has at least one arc (t_A, v) with positive $w_A^p(v)$ value, then the corresponding multi-head HyTN is consistent; also notice the similarity w.r.t. the potentials that are computed by the Bellman-Ford algorithm on STNs.

Again, as it was the case for STNs, a mapping $\phi : V \to \mathbf{R}$ is a feasible potential if and only if it is a feasible schedule. In order to better characterize feasible schedules, a notion of *negative cycle* is introduced next.

Definition 5. *Given a multi-head HyTN $\mathcal{H} = (V, \mathcal{A})$, a cycle is a pair $(S, \mathcal{C})$ with $S \subseteq V$ and $\mathcal{C} \subseteq \mathcal{A}$ such that:*

1. *$S = \bigcup_{A \in \mathcal{C}} \left(H_A \cup \{t_A\} \right)$ and $S \neq \emptyset$;*
2. *$\forall v \in S$ there exists an unique $A \in \mathcal{C}$ such that $t_A = v$.*

Moreover, we let $a(v)$ denote the unique arc $A \in \mathcal{C}$ with $t_A = v$, as required in item 2 above. Every infinite path in a cycle $(S, \mathcal{C})$ contains, at least, one finite cyclic sequence $v_i, v_{i+1}, \ldots, v_{i+p}$, where $v_{i+p} = v_i$ is the only repeated node in the sequence. A cycle $(S, \mathcal{C})$ is negative if and only if the following holds:

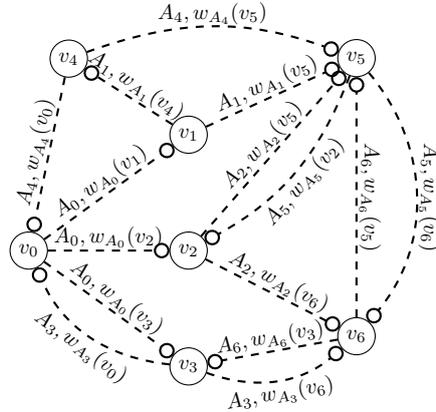$$ \sum_{t=1}^{p-1} w_{a(v_t)}(v_{t+1}) < 0, \ \text{for any finite cyclic sequence } v_1, v_2, \ldots, v_p. $$

Figure 2: A Cycle $(S, \mathcal{C})$, where $S = \{v_0, \ldots, v_6\}$ and $\mathcal{C} = \{A_0, \ldots, A_6\}$.

Example 1. *An example of a cycle $(S, \mathcal{C})$ is shown in Fig. 2; here, $S = \{v_0, \ldots, v_6\}$ and $\mathcal{C} = \{A_0, \ldots, A_6\}$, where $t_{A_i} = v_i$ for every $i \in \{0, \ldots, 6\}$; moreover, $H_{A_0} = \{v_1, v_2, v_3\}$, $H_{A_1} = \{v_4, v_5\}$, $H_{A_2} = \{v_5, v_6\}$, $H_{A_3} = \{v_0, v_6\}$, $H_{A_4} = \{v_0, v_5\}$, $H_{A_5} = \{v_2, v_6\}$, $H_{A_6} = \{v_3, v_5\}$.*

Lemma 2 (Comin et al. (2016)). *A HyTN with a negative cycle $(S, \mathcal{C})$ admits no feasible schedule.*

At first sight, it may appear that checking whether $(S, \mathcal{C})$ is a negative cycle might take exponential time since one should check a possibly exponential number of cyclic sequences. The next lemma asserts instead that it is possible to check the presence of a negative cycle in polynomial time.

Lemma 3 (Comin et al. (2016)). *Let $(S, \mathcal{C})$ be a cycle in a HyTN. Then, checking whether $(S, \mathcal{C})$ is a negative cycle can be done in polynomial time.*

A hypergraph $\mathcal{H}$ is called *conservative* when it contains no negative cycle $(S, \mathcal{C})$.

In the next paragraphs we will recall the existence of pseudo-polynomial time algorithms that always return either a feasible schedule or a negative cycle certificate, thus extending the validity of the classical good-characterization of STN consistency to general HyTN consistency. Here, we anticipate the statement of the main result in order to complete this brief introduction to HyTNs.

Theorem 4 (Comin et al. (2016)). *A HyTN $\mathcal{H}$ is consistent if and only if it is conservative. Moreover, when all weights are integers, then $\mathcal{H}$ admits an integer schedule if and only if it is conservative.*

From now on we shall focus on integer weighted multi-head hypergraphs and HyTNs.

Definition 6 (Hypergraph). *A hypergraph $\mathcal{H}$ is a pair $\langle V, \mathcal{A} \rangle$, where V is the set of nodes, and $\mathcal{A}$ is the set of hyperarcs. Each hyperarc $A = \langle t_A, H_A, w_A \rangle \in \mathcal{A}$ has a distinguished node t_A, called the* tail *of A, and a non-empty set $H_A \subseteq V \setminus \{t_A\}$ containing the* heads *of A; to each head $v \in H_A$ is associated a* weight $w_A(v) \in \mathbf{Z}$.

Again, provided that $|A| \triangleq |H_A \cup \{t_A\}|$, the *size* of a hypergraph $\mathcal{H} = \langle V, \mathcal{A} \rangle$ is defined as $m_{\mathcal{A}} \triangleq \sum_{A \in \mathcal{A}} |A|$, and it is used as a measure for the encoding length of $\mathcal{H}$; if $|A| = 2$, then $A = \langle u, v, w \rangle$ is a *standard arc*. In this way, hypergraphs generalize graphs.

At this point, a (multi-head) HyTN is thus a weighted hypergraph $\mathcal{H} = \langle V, \mathcal{A} \rangle$ where a node represents an *event* to be scheduled in time, and a hyperarc represents a set of temporal distance *constraints* between the *tail* and the *heads*.

The computational equivalence between checking the consistency of (integer weighted multi-head) HyTNs and determining winning sets in MPGs was pointed out in Comin et al. (2014, 2016). The tightest currently known worst-case time complexity upper bound for solving HEAD-HYTN-CONSISTENCY is expressed in the following theorem, which was proved in Comin et al. (2016) by relying on the Value-Iteration Algorithm for MPGs (Brim et al., 2011); the approach was shown to be robust thanks to experimental evaluations (also see Brim and Chaloupka (2012)).

Theorem 5. *(Comin et al., 2014, 2016) The following propositions hold on (integer weighted multi-head) HyTNs.*

1. *There exists an $O\big((|V|+|\mathcal{A}|)m_{\mathcal{A}}W\big)$ pseudo-polynomial time algorithm for checking* HEAD-HYTN-CONSISTENCY*;*
2. *There exists an $O\big((|V|+|\mathcal{A}|)m_{\mathcal{A}}W\big)$ pseudo-polynomial time algorithm such that, given as input any consistent HyTN $\mathcal{H} = (V, \mathcal{A})$, it returns as output a feasible schedule $\phi : V \rightarrow \mathbf{R}$ of $\mathcal{H}$;*
3. *There exists an $O\big((|V| + |\mathcal{A}|)m_{\mathcal{A}}W\big)$ pseudo-polynomial time algorithm such that, given as input any inconsistent HyTN $\mathcal{H} = (V, \mathcal{A})$, it returns as output a negative cycle $(S, \mathcal{C})$ of $\mathcal{H}$;*

Here, $W \triangleq \max_{A \in \mathcal{A}, v \in H_A} |w_A(v)|$.

In the forthcoming section we shall turn our attention to *conditional* temporal planning, where we generalize Conditional Simple Temporal Networks (CSTNs) by introducing Conditional Hyper Temporal Networks (CHyTNs).

3. Conditional Simple / Hyper Temporal Networks

In order to provide a formal support to the present work, this section recalls the basic formalism, terminology and known results on CSTPs and CSTNs. Since the forthcoming definitions concerning CSTNs are mostly inherited from the literature, the reader is referred to Tsamardinos et al. (2003) and Hunsberger et al. (2012) for an intuitive semantic discussion and for some clarifying examples of the very same CSTN model. Tsamardinos et al. (2003) introduced the *Conditional Simple Temporal Problem (CSTP)* as an extension of standard temporal constraint-satisfaction models used in non-conditional temporal planning. CSTPs augment STNs by including *observation* events, each one having a *boolean variable* (or *proposition*) associated with it. When an observation event is executed, the truth-value of its associated proposition becomes known. In addition, each event node and each constraint has a *label* that restricts the scenarios in which it plays a role. Although not included in the formal definition, Tsamardinos et al. (2003) discussed some supplementary assumptions that any well-defined CSTP must satisfy. Subsequently, those conditions have been further analyzed and formalized by (Hunsberger et al., 2012), leading to the definition of *Conditional Simple Temporal Network* (CSTN), which is now recalled.

Let P be a set of boolean variables, a *label* is any (possibly empty) conjunction of variables, or negations of variables, drawn from P. The *empty label* is denoted by λ. The *label universe* P^* is the set of all (possibly empty)

labels whose (positive or negative) literals are drawn from P. Two labels, ℓ_1 and ℓ_2, are called *consistent*, denoted[1] by $Con(\ell_1, \ell_2)$, when $\ell_1 \wedge \ell_2$ is satisfiable. A label ℓ_1 *subsumes* a label ℓ_2, denoted[1] by $Sub(\ell_1, \ell_2)$, when the implication $\ell_1 \Rightarrow \ell_2$ holds. Let us recall the formal definition of CSTNs from (Tsamardinos et al., 2003; Hunsberger et al., 2012).

Definition 7 (CSTNs). *A* Conditional Simple Temporal Network (CSTN) *is a tuple* $\langle V, A, L, \mathcal{O}, \mathcal{O}V, P \rangle$ *where:*

- *V is a finite set of* events*; $P = \{p_1, \ldots, p_q\}$ (some $q \in \mathbf{N}$) is a finite set of* boolean variables *(or* propositions*);*

- *A is a set of* labeled temporal constraints (LTCs) *each having the form $\langle v - u \leq w(u, v), \ell \rangle$, where $u, v \in V$, $w(u, v) \in \mathbf{R}$, and $\ell \in P^*$;*

- *$L : V \rightarrow P^*$ is a map that assigns a label to each event node in V; $\mathcal{O}V \subseteq V$ is a finite set of* observation *events; $\mathcal{O} : P \rightarrow \mathcal{O}V$ is a bijection mapping a unique observation event $\mathcal{O}(p) = \mathcal{O}_p$ to each $p \in P$;*

- *The following* well definedness assumptions *must hold:*

    (WD1) for any labeled constraint $\langle v - u \leq w, \ell \rangle \in A$ the label ℓ is satisfiable and subsumes both $L(u)$ and $L(v)$; i.e., whenever a constraint $v - u \leq w$ is required to be satisfied, both of its endpoints u and v must be scheduled (sooner or later) by the Planner;

    (WD2) for each $p \in P$ and each $u \in V$ such that either p or $\neg p$ appears in $L(u)$, we require: $\mathrm{Sub}(L(u), L(\mathcal{O}_p))$, and $\langle \mathcal{O}_p - u \leq -\epsilon, L(u) \rangle \in A$ for some (small) real $\epsilon > 0$; i.e., whenever a label $L(u)$ of an event node u contains a proposition p, and u gets eventually scheduled, the observation event $\mathcal{O}_p$ must have been scheduled strictly before u by the Planner.

    (WD3) for each labeled constraint $\langle v - u \leq w, \ell \rangle$ and $p \in P$, for which either p or $\neg p$ appears in ℓ, it holds that $\mathrm{Sub}(\ell, L(\mathcal{O}_p))$; i.e., assuming a required constraint contains proposition p, the observation event $\mathcal{O}_p$ must be scheduled (sooner or later) by the Planner.

We are now in the position to introduce the *Conditional Hyper Temporal Network (CHyTN)*, a natural extension and generalization of both the CSTN and the HyTN model obtained by blending them together. Even though the original STN and CSTN models allow for real weights, hereafter we shall restrict ourselves to the integers in order to rely on Theorem 5. All of our CSTNs and CHyTNs will be integer weighted from now on.

Definition 8 (CHyTNs). *A general* Conditional Hyper Temporal Network (CHyTN) *is a tuple* $\langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$*, where $V, P, L, \mathcal{O}$ and $\mathcal{O}V$ are defined as in CSTNs (see Definition 7), and where $\mathcal{A}$ is a set of* labeled temporal hyper constraints (LTHCs)*, each having one of the following forms:*

- *$A = (t, h, w, \ell)$, where (t, h, w) is a standard arc and $\ell \in P^*$; in this case, A is called a* standard *LTHC.*

- *$A = (t_A, H_A, w_A, L_{H_A})$, where (t_A, H_A, w_A) is a* multi-head *hyperarc and $L_{H_A} : H_A \rightarrow P^*$ is a map sending each head $h \in H_A$ to a label ℓ_h in P^*; in this case, A is called a* multi-head *LTHC.*

- *$A = (T_A, h_A, w_A, L_{T_A})$, where $A = (T_A, h_A, w_A)$ is a* multi-tail *hyperarc and $L_{T_A} : T_A \rightarrow P^*$ is a map sending each tail $t \in T_A$ to a label ℓ_t in P^*; in this case, A is called a* multi-tail *LTHC.*

- *The following* well definedness assumptions *must hold:*

    (WD1') for any labeled constraint A:

    - *if $A = (t, h, w, \ell)$ is a standard LTHC, the label ℓ is satisfiable and subsumes both $L(t)$ and $L(h)$;*

    - *if $A = (t_A, H_A, w_A, L_{H_A})$ is a multi-head LTHC, for each $h \in H_A$ the label $L_{H_A}(h)$ is satisfiable and subsumes both $L(t_A)$ and $L(h)$;*

    - *if $A = (T_A, h_A, w_A, L_{T_A})$ is a multi-tail LTHC, for each $t \in T_A$ the label $L_{T_A}(t)$ is satisfiable and subsumes both $L(h_A)$ and $L(t)$;*

---

[1] The notation $Con(\cdot, \cdot)$ and $Sub(\cdot, \cdot)$ is inherited from (Tsamardinos et al., 2003; Hunsberger et al., 2012).

(WD2) for each $p \in P$ and each $u \in V$ such that either p or $\neg p$ appears in $L(u)$, we require: $\mathrm{Sub}(L(u), L(\mathcal{O}_p))$, and $\langle \mathcal{O}_p - u \leq -\epsilon, L(u) \rangle \in \mathcal{A}$ for some (small) real $\epsilon > 0$; this is the same WD2 as defined for CSTNs.

(WD3') for each labeled constraint $A \in \mathcal{A}$ and boolean variable $p \in P$:

- *if $A = (t, h, w, \ell)$ is a standard LTHC and p or $\neg p$ appears in ℓ, then $\mathrm{Sub}(\ell, L(\mathcal{O}_p))$;*
- *if $A = (t_A, H_A, w_A, L_{H_A})$ is a multi-head LTHC and either p or $\neg p$ appears in $L_{H_A}(h)$ for some $h \in H_A$, then $\mathrm{Sub}(L_{H_A}(h), L(\mathcal{O}_p))$;*
- *if $A = (T_A, h_A, w_A, L_{T_A})$ is a multi-tail LTHC and either p or $\neg p$ appears in $L_{T_A}(t)$ for some $t \in T_A$, then $\mathrm{Sub}(L_{T_A}(t), L(\mathcal{O}_p))$;*

Of course every CSTN is a CHyTN (i.e., one having only standard LTHCs). We shall adopt the notation $x \xrightarrow{[a,b],\ell} y$, where $x, y \in V$, $a, b \in \mathbf{N}$, $a < b$ and $\ell \in P^*$, to compactly represent the pair $\langle y - x \leq b, \ell \rangle, \langle x - y \leq -a, \ell \rangle \in A$; also, whenever $\ell = \lambda$, we shall omit ℓ from the graphics, see e.g., Fig. 3a and Fig. 3b here below.

Example 2. *Fig. 3a depicts an example CSTN $\Gamma_0 = \langle V, A, L, \mathcal{O}, \mathcal{O}V, P \rangle$ having three event nodes A, B and C as well as two observation events $\mathcal{O}_p$ and $\mathcal{O}_q$. Formally, $V = \{A, B, C, \mathcal{O}_p, \mathcal{O}_q\}$, $P = \{p, q\}$, $\mathcal{O}V = \{\mathcal{O}_p, \mathcal{O}_q\}$, $L(v) = \lambda$ for every $v \in V \setminus \{\mathcal{O}_q\}$ and $L(\mathcal{O}_q) = p$, $\mathcal{O}(p) = \mathcal{O}_p$, $\mathcal{O}(q) = \mathcal{O}_q$. Next, the set of LTCs is: $A = \{\langle C - A \leq 10, \lambda \rangle, \langle A - C \leq -10, \lambda \rangle, \langle B - A \leq 3, p \wedge \neg q \rangle, \langle A - B \leq 0, \lambda \rangle, \langle \mathcal{O}_p - A \leq 5, \lambda \rangle, \langle A - \mathcal{O}_p \leq 0, \lambda \rangle, \langle \mathcal{O}_q - A \leq 9, p \rangle, \langle A - \mathcal{O}_q \leq 0, p \rangle, \langle C - B \leq 2, q \rangle, \langle C - \mathcal{O}_p \leq 10, \lambda \rangle$.*

Fig. 3b depicts an example of a multi-head CHyTN $\Gamma_1 = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$. Notice that $V, L, \mathcal{O}, \mathcal{O}V$ and P are the same as in the CSTN Γ_0, whereas $\mathcal{A}$ is defined as follows: $\mathcal{A} = A \cup \{\alpha \triangleq (B, \{C, \mathcal{O}_q\}, \langle w_\alpha(C), w_\alpha(\mathcal{O}_q) \rangle = \langle 2, -1 \rangle, \langle L_\alpha(C), L_\alpha(\mathcal{O}_q) \rangle = \langle \lambda, p \rangle)\}$, where A is the set of LTCs of the CSTN Γ_0 and the additional constraint α is a multi-head LTHC with tail $t_\alpha = B$ and heads $H_\alpha = \{C, \mathcal{O}_q\}$.

Sometimes we will show the scheduling time of a node with a label in boldface on the sidelines of the node itself, as for A in Fig. 3.

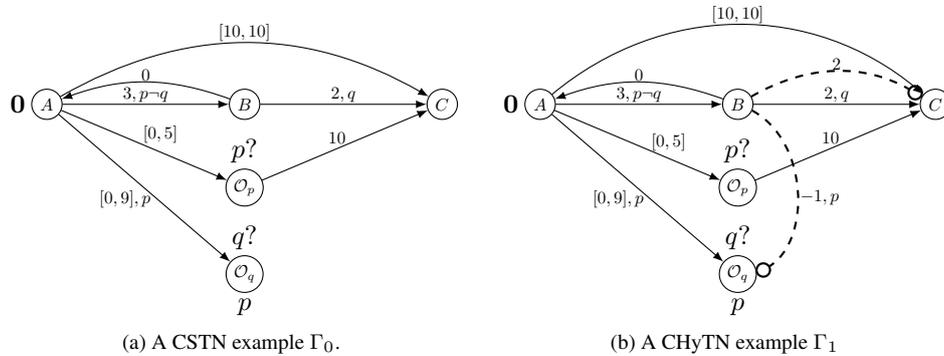(a) A CSTN example Γ_0.　　　　　　(b) A CHyTN example Γ_1

Figure 3: An example CSTN (a), and an example CHyTN (b).

In the following definitions we will implicitly refer to some CHyTN which is denoted by $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$.

Definition 9 (Scenario). *A scenario over a set P of boolean variables is a truth assignment $s : P \to \{\top, \bot\}$, i.e., s is a map that assigns a truth value to each proposition $p \in P$. The set of all scenarios over P is denoted by Σ_P.*

If $s \in \Sigma_P$ and $\ell \in P^*$, then $s(\ell) \in \{\top, \bot\}$ denotes the truth value of ℓ induced by s in the natural way.

Notice that any scenario $s \in \Sigma_P$ can be described by means of the label $\ell_s \triangleq l_1 \wedge \cdots \wedge l_{|P|}$ such that, for every $1 \leq i \leq |P|$, the literal $l_i \in \{p_i, \neg p_i\}$ satisfies $s(l_i) = \top$.

Example 3. *Let $P = \{p, q\}$. The scenario $s : P \to \{\top, \bot\}$ defined as $s(p) = \top$ and $s(q) = \bot$ can be compactly described by the label $\ell_s = p \wedge \neg q$.*

Definition 10 (Schedule)**.** *A* schedule *for a subset of events* $U \subseteq V$ *is a map* $\phi : U \to \mathbf{R}$ *that assigns a real number to each event node in* U. *The set of all schedules over* U *is denoted by* Φ_U.

Definition 11 (Scenario Restriction)**.** *Let* $s \in \Sigma_P$ *be a scenario. The* restriction *of* V *and* $\mathcal{A}$ *w.r.t.* s *are defined as:*

$$V_s^+ \triangleq \Big\{ v \in V \mid s(L(v)) = \top \Big\};$$

$$\mathcal{A}_s^+ \triangleq \Big\{ (u, v, w) \mid \exists (\ell \in P^*) \text{ s.t. } (u, v, w, \ell) \in \mathcal{A} \text{ and } s(\ell) = \top \Big\} \cup$$
$$\cup \Big\{ (t, H_A', w_A') \mid \exists (H_A \supseteq H_A'; L_{H_A} : H_A \to P^*; w_A : H_A \to \mathbf{Z}) \text{ s.t. } (t, H_A, w_A, L_{H_A}) \in \mathcal{A},$$
$$w_A' = w_{A|_{H_A'}}, \forall (h \in H_A)\, s(L_{H_A}(h)) = \top \iff h \in H_A' \Big\} \cup$$
$$\cup \Big\{ (T_A', h, w_A') \mid \exists (T_A \supseteq T_A'; L_{T_A} : T_A \to P^*; w_A : T_A \to \mathbf{Z}) \text{ s.t. } (T_A, h, w_A, L_{T_A}) \in \mathcal{A},$$
$$w_A' = w_{A|_{T_A'}}, \forall (t \in T_A)\, s(L_{T_A}(t)) = \top \iff t \in T_A' \Big\}.$$

The restriction of Γ *w.r.t.* s *is defined as* $\Gamma_s^+ \triangleq \langle V_s^+, \mathcal{A}_s^+ \rangle$.
Finally, it is worthwhile to introduce the notation $V_{s_1, s_2}^+ \triangleq V_{s_1}^+ \cap V_{s_2}^+$.

Note that if Γ is a CHyTN, then Γ_s^+ is a HyTN; and if Γ is a CSTN, then Γ_s^+ is an STN.

Example 4. *Fig. 4 depicts the restriction STN* $\Gamma_{0\,s}^+$ *of the CSTN* Γ_0, *and the restriction HyTN* $\Gamma_{1\,s}^+$ *of the CHyTN* Γ_1 *(see Example 2 and Fig. 3), w.r.t. the scenario* $s(p) = s(q) = \bot$.

(a) The restriction STN $\Gamma_{0\,s}^+$ of the CSTN Γ_0 w.r.t. $s(p) = s(q) = \bot$

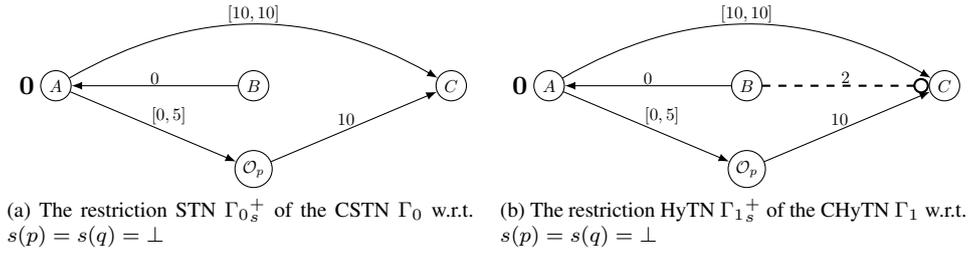(b) The restriction HyTN $\Gamma_{1\,s}^+$ of the CHyTN Γ_1 w.r.t. $s(p) = s(q) = \bot$

Figure 4: The restriction $\Gamma_{0\,s}^+$ (a), and the restriction $\Gamma_{1\,s}^+$ (b), w.r.t. the scenario $s(p) = s(q) = \bot$

Definition 12 (Execution Strategy (Hunsberger et al., 2012))**.** *An* execution strategy *for* Γ *is a mapping* $\sigma : \Sigma_P \to \Phi_V$ *such that, for any scenario* $s \in \Sigma_P$, *the domain of the schedule* $\sigma(s)$ *is* V_s^+. *The set of execution strategies of* Γ *is denoted by* $\mathcal{S}_\Gamma$. *The* execution time *of an event node* $v \in V_s^+$ *in the schedule* $\sigma(s) \in \Phi_{V_s^+}$ *is denoted by* $[\sigma(s)]_v$.

Definition 13 (Scenario History (Hunsberger et al., 2012))**.** *Let* $\sigma \in \mathcal{S}_\Gamma$ *be an execution strategy, let* $s \in \Sigma_P$ *be a scenario and let* $v \in V_s^+$ *be an event node. The* scenario history *scHst(v, s, σ) of* v *in the scenario* s *for the strategy* σ *is defined as follows:*

$$scHst(v, s, \sigma) \triangleq \Big\{ \big(p, s(p)\big) \mid p \in P, \mathcal{O}_p \in V_s^+ \cap \mathcal{OV}, [\sigma(s)]_{\mathcal{O}_p} < [\sigma(s)]_v \Big\}.$$

The scenario history can be compactly expressed by the conjunction of the literals corresponding to the observations comprising it. Thus, we may treat a scenario history as though it were a label.

Definition 14 (Viable Execution Strategy (Hunsberger et al., 2012))**.** *We say that* $\sigma \in \mathcal{S}_\Gamma$ *is a* viable *execution strategy whenever, for each scenario* $s \in \Sigma_P$, *the schedule* $\sigma(s) \in \Phi_V$ *is feasible for the restriction HyTN (or STN)* Γ_s^+.

Definition 15 (Dynamic Consistency (Hunsberger et al., 2012))**.** *An execution strategy* $\sigma \in \mathcal{S}_\Gamma$ *is called* dynamic *if, for any* $s_1, s_2 \in \Sigma_P$ *and any event node* $v \in V_{s_1, s_2}^+$, *the following implication holds:*

$$\text{Con}(scHst(v, s_1, \sigma), s_2) \Rightarrow [\sigma(s_1)]_v = [\sigma(s_2)]_v. \tag{DC}$$

We say that Γ *is* dynamically-consistent *if it admits* $\sigma \in \mathcal{S}_\Gamma$ *which is both viable and dynamic.*

Definition 16 (DC-Checking (Hunsberger et al., 2012)). *The problem of checking whether a given CHyTN (which allows both multi-head and multi-tail LTHCs) is dynamically-consistent is named* General-CHyTN-DC.

That of checking whether a given CHyTN, allowing only multi-head or only multi-tail LTHCs, is dynamically-consistent is named CHyTN-DC. *Checking whether a given CSTN is dynamically-consistent is named* CSTN-DC.

Example 5. *Consider the CHyTN Γ_1 of Fig. 3b, and let the scenarios s_1, s_2, s_3, s_4 be defined as: $s_1(p) = \top$, $s_1(q) = \top$; $s_2(p) = \top$, $s_2(q) = \bot$; $s_3(p) = \bot$, $s_3(q) = \top$; $s_4(p) = \bot$, $s_4(q) = \bot$. The following defines an execution strategy $\sigma \in \mathcal{S}_\Gamma$: $[\sigma(s_i)]_A = 0$ for every $i \in \{1,2,3,4\}$; $[\sigma(s_i)]_B = 8$ for every $i \in \{1,3,4\}$ and $[\sigma(s_2)]_B = 3$; $[\sigma(s_i)]_C = 10$ for every $i \in \{1,2,3,4\}$; $[\sigma(s_i)]_{\mathcal{O}_p} = 1$ for every $i \in \{1,2,3,4\}$. The reader can check that σ is viable and dynamic. Indeed, σ admits the tree-like representation depicted in Fig 5.*
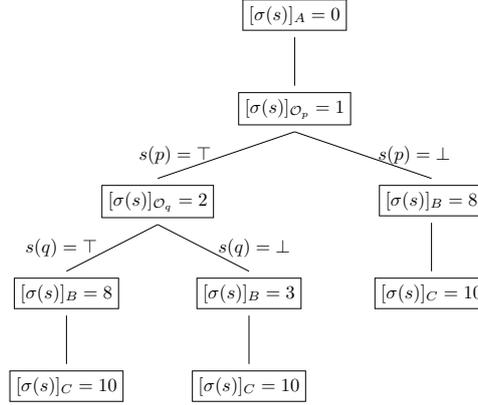

Figure 5: A tree-like representation of a dynamic execution strategy σ for the CHyTN Γ_1 of Fig. 3b, where s denotes scenarios and $[\sigma(s)]_X$ is the corresponding schedule.

Next, we recall a crucial notion for studying the dynamic consistency of CHyTNs: the *difference set* $\Delta(s_1; s_2)$.

Definition 17 (Difference Set (Tsamardinos et al., 2003)). *Let $s_1, s_2 \in \Sigma_P$ be two scenarios. The set of observation events in $V_{s_1}^+ \cap \mathcal{OV}$ at which s_1 and s_2 differ is denoted by $\Delta(s_1; s_2)$. Formally,*

$$\Delta(s_1; s_2) \triangleq \{\mathcal{O}_p \in V_{s_1}^+ \cap \mathcal{OV} \mid s_1(p) \neq s_2(p)\}.$$

Notice that commutativity may not hold (i.e., generally it may be the case that $\Delta(s_1; s_2) \neq \Delta(s_2; s_1)$).

Example 6. *Consider the CSTN Γ_0 of Fig. 3a and the scenarios s_1, s_2 defined as follows: $s_1 \triangleq p \wedge q$; $s_2 \triangleq \neg p \wedge \neg q$. Then, $\Delta(s_1; s_2) = \{\mathcal{O}_p, \mathcal{O}_q\}$ and $\Delta(s_2; s_1) = \{\mathcal{O}_p\}$.*

The next lemma will be useful later on in Section 4.

Lemma 4 (Tsamardinos et al. (2003)). *Let $s_1, s_2 \in \Sigma_P$ and $v \in V_{s_1,s_2}^+$. Let $\sigma \in \mathcal{S}_\Gamma$ be an execution strategy.*
Then, σ is dynamic if and only if the following implication holds for every $s_1, s_2 \in \Sigma_P$ and for every $u \in V_{s_1,s_2}^+$:

$$\left(\bigwedge_{v \in \Delta(s_1; s_2)} [\sigma(s_1)]_u \leq [\sigma(s_1)]_v \right) \Rightarrow [\sigma(s_1)]_u = [\sigma(s_2)]_u \tag{L4}$$

Proof. Notice that, by definition of $Con(\cdot, \cdot)$ and $scHst(\cdot, \cdot, \cdot)$, $Con(scHst(u, s_1, \sigma), s_2)$ holds if and only if there is no observation event $v \in \Delta(s_1; s_2)$ which is scheduled by $\sigma(s_1)$ strictly before u. Therefore, $Con(scHst(u, s_1, \sigma), s_2)$ holds if and only if $\bigwedge_{v \in \Delta(s_1; s_2)} [\sigma(s_1)]_u \leq [\sigma(s_1)]_v$. At this point, substituting the $Con(scHst(u, s_1, \sigma), s_2)$ expression with the equivalent formula $\bigwedge_{v \in \Delta(s_1; s_2)} [\sigma(s_1)]_u \leq [\sigma(s_1)]_v$ inside the definition of dynamic execution strategy (see Definition 15), the thesis follows. $\square$

4. Algorithmics of Dynamic Consistency

Firstly, let us offer the following coNP-hardness result for CSTN-DC; notice that, since any CSTN is also a CHyTN, the same hardness result holds for CHyTNs.

Theorem 6. *CSTN-DC is* coNP-*hard even if the input instances* $\Gamma = \langle V, A, L, \mathcal{O}, \mathcal{O}V, P \rangle$ *are restricted to satisfy* $w_A(\cdot) \in \{-1, 0\}$ *and* $\ell \in \{p, \neg p \mid p \in P\} \cup \{\lambda\}$ *for every* $(u, v, w, \ell) \in A$.

Proof. We reduce 3-SAT to the complement of CSTN-DC. Let φ be a boolean formula in 3CNF. Let X be the set of variables and let $\mathcal{C} = \{C_0, \ldots, C_{m-1}\}$ be the set of clauses comprising $\varphi = \bigwedge_{j=0}^{m-1} C_j$.

(1) Let N^φ be the CSTN $\langle V^\varphi, A^\varphi, L^\varphi, \mathcal{O}^\varphi, \mathcal{O}V^\varphi, P^\varphi \rangle$, where: $V^\varphi \triangleq X \cup \mathcal{C}$, and all the nodes are given an empty label, i.e., $L^\varphi(v) = \lambda$ for every $v \in V^\varphi$; each variable in X becomes an observation event and each clause in $\mathcal{C}$ a non-observation, i.e., $P^\varphi \triangleq \mathcal{O}V^\varphi \triangleq X$, so, $\mathcal{O}^\varphi$ is the identity map; moreover, all observation events will be forced to be executed simultaneously before any of the non-observation events, thus for every $u, v \in \mathcal{O}V^\varphi$ we have $\langle u - v \le 0, \lambda \rangle \in A^\varphi$, and for every $x \in X$ and $C \in \mathcal{C}$ we have $\langle x - C \le -1, \lambda \rangle \in A^\varphi$; finally, there is a negative loop among all the $C \in \mathcal{C}$ which plays an important role in the rest of the proof, particularly, for each $j = 0, \ldots, m-1$ and for each literal $\ell \in C_j$, we have $\langle C_j - C_{(j+1) \bmod m} \le -1, \ell \rangle \in \mathcal{A}_\varphi$. Notice that $|V^\varphi| = n + m$ and $|A^\varphi| = n^2 + nm + 3m$.

(2) We show that, if φ is satisfiable, there must be an unavoidable negative circuit among all the $C_j \in \mathcal{C}$. Assume that φ is satisfiable. Let ν be a satisfying truth-assignment of φ. In order to prove that N^φ is not dynamically-consistent, observe that the restriction of N^φ w.r.t. the scenario ν is an inconsistent STN. Indeed, if for every $j = 0, \ldots, m-1$ we pick a standard arc $\langle C_j - C_{(j+1) \bmod m} \le -1, \ell_j \rangle$ with ℓ_j being a literal in C_j such that $\nu(\ell_j) = \top$, then we obtain a negative circuit.

(3) We show that, if φ is unsatisfiable, there can't be a negative circuit among the $C_j \in \mathcal{C}$ because for each scenario, there will be at least one j such that all three labels, α_j, β_j and γ_j will be false. Assume that φ is unsatisfiable. In order to prove that N^φ is dynamically-consistent, we exhibit a viable and dynamic execution strategy σ for N^φ. Firstly, schedule every $x \in X$ at $\sigma(x) \triangleq 0$. Therefore, by time 1, the strategy has full knowledge of the observed scenario ν. Since φ is unsatisfiable, there exists an index j_ν such that $\nu(C_{j_\nu}) = \bot$. At this point, set $\sigma(C_{(j_\nu + k) \bmod m}) \triangleq k$ for each $k = 1, \ldots, m$. The reader can verify that σ is viable and dynamic for N^φ. $\qquad\square$

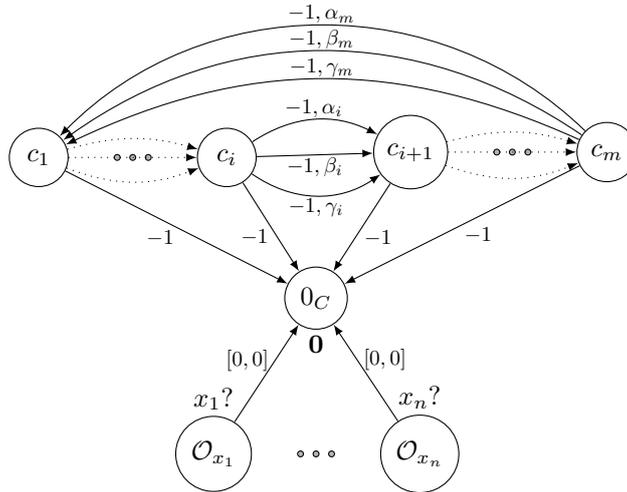Figure 6: The CSTN N^φ where $\varphi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} c_i$ for $c_i = (\alpha_i \vee \beta_i \vee \gamma_i)$.

An illustration of the CSTN N^φ, which was constructed in the proof of Theorem 6, is shown in Fig. 6; to ease the representation we have introduced an additional non-observation event 0_C in Fig. 6, which is executed at time $t = 0$, together with all of the observation events in X.

13

Next, we show that when the input CHyTN instances are allowed to have *both* multi-heads *and* multi-tail LTHCs then the DC-Checking problem becomes PSPACE-hard.

Theorem 7. *General-CHyTN-DC is* PSPACE-*hard, even if the input instances* $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$ *are restricted to satisfy the following two constraints:*

$- w_a(\cdot) \in [-n-1, n+1] \cap \mathbf{Z}$ *and* $\ell_a \in \{p, \neg p \mid p \in P\} \cup \{\lambda\}$ *for every weight* w_a *and label* ℓ_a *appearing in any* standard *LTHC* $a \in \mathcal{A}$;

$- w_A(\cdot) \in \{-1, 0, 1\}$, $\ell_A = \lambda$ *and* $|A| \leq 2$ *for every weight* w_A *and label* ℓ_A *appearing in any* multi-tail/head *LTHC* $A \in \mathcal{A}$.

Proof. To show that General-CHyTN-DC is PSPACE-hard, we describe a reduction from the problem 3-CNF-TQBF (True Quantified Boolean Formula in 3-CNF).

Let us consider a 3-CNF quantified boolean formula with $n \geq 1$ variables and $m \geq 1$ clauses:

$$\varphi(x_1, \ldots, x_n) = Q_1 x_1 \ldots Q_n x_n \bigwedge_{i=1}^{m} (\alpha_i \vee \beta_i \vee \gamma_i),$$

where for every $j \in [n]$ the symbol Q_j is either $\exists$ or $\forall$, and where $\mathcal{C}_i = (\alpha_i \vee \beta_i \vee \gamma_i)$ is the i-th clause of φ and each $\alpha_i, \beta_i, \gamma_i \in \{x_j, \neg x_j \mid 1 \leq j \leq n\}$ is a positive or negative literal. We also say that $Q_1 x_1 \ldots Q_n x_n$ is the *prefix* of φ.
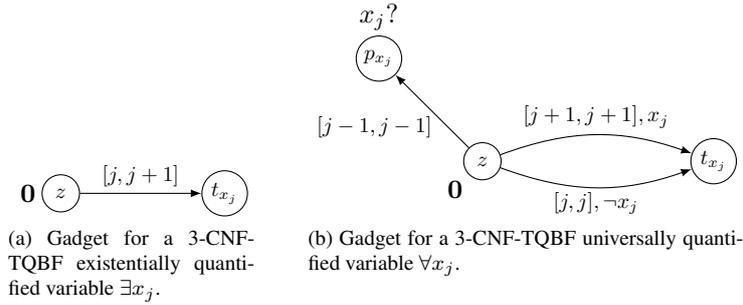

(a) Gadget for a 3-CNF-TQBF existentially quantified variable $\exists x_j$.

(b) Gadget for a 3-CNF-TQBF universally quantified variable $\forall x_j$.

Figure 7: Gadgets for quantified variables used in the reduction from 3-CNF-TQBF to General-CHyTN-DC.

<u>Construction.</u> We associate to φ a CHyTN $\Gamma_\varphi = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$. In so doing, our first goal is to simulate the interaction between two players: Player-$\exists$ (corresponding to the Planner in CHyTNs) and Player-$\forall$ (corresponding to the Nature in CHyTNs), which corresponds directly to the chain of alternating quantifiers in the prefix of φ. Naturally, the Planner is going to control those variables that are quantified existentially in φ, whereas the Nature is going to control (by means of some observation events in $\mathcal{OV}$) those variables that are quantified universally in φ. Briefly, P contains one boolean variable for each universally quantified variable of φ, and V contains the following: two special events z and z' to be executed at time 0 and $n+1$, respectively; an observation event p_{x_j} for each universally quantified variable $\forall x_j$; a non-observation event t_{x_j} for each quantified variable x_j; two non-observation events l_{x_j} and $l_{\overline{x}_j}$ for each quantified variable x_j, these will play (respectively) the role of positive and negative literals of φ (i.e., the α, β and γ in each clause $\mathcal{C}_i$); finally, a non-observation event $\mathcal{C}_i$ for each clause.

Let us describe the low-level details of Γ_φ. We let $P \triangleq \{x_j \mid ``\forall x_j"$ appears in the prefix of $\varphi\}$. Moreover, V contains a node z (i.e., the *zero* node that has to be executed at time $t = 0$).

Next, for each existential quantification $\exists x_j$ in the prefix of φ, V contains a node named t_{x_j} and $\mathcal{A}$ contains the following two standard LTHCs: $(z, t_{x_j}, j+1, \lambda)$ and $(t_{x_j}, z, -j, \lambda)$; the underlying intuition being that, during execution, it will be the responsibility of the Planner to schedule t_{x_j} either at time j (and this means that the Planner chooses to set x_j to false in φ) or at time $j+1$ (and this means that he chooses to set x_j to true in φ). See Fig. 7a for an illustration of the $\exists x_j$ gadget.

Moreover, for each universal quantification $\forall x_j$ in the prefix of φ (i.e., for each $x_j \in P$), V contains two nodes named p_{x_j} and t_{x_j}. Particularly, p_{x_j} is an observation event (i.e., $p_{x_j} \in \mathcal{OV}$) such that $\mathcal{O}(x_j) = p_{x_j}$; hence, $\mathcal{OV} \triangleq \{p_{x_j} \mid x_j \in P\}$. Also, for each $\forall x_j$ in φ's prefix (i.e., for each $x_j \in P$), $\mathcal{A}$ contains the following six

standard LTHCs: $(z, p_{x_j}, j - 1, \lambda)$, $(p_{x_j}, z, -j + 1, \lambda)$, $(z, t_{x_j}, j + 1, x_j)$, $(t_{x_j}, z, -j - 1, x_j)$, $(z, t_{x_j}, j, \neg x_j)$ and $(t_{x_j}, z, -j, \neg x_j)$; the underlying intuition being that the Nature must choose whether to schedule t_{x_j} at time j (setting x_j to false in φ by controlling the observation event p_{x_j}) or at time $j + 1$ (setting x_j to true in φ again, by controlling the observation event p_{x_j}). Fig. 7b illustrates the gadget for universally quantified variables $\forall x_j$.

In both cases (existentially and universally quantified variables), the weights of the involved standard temporal constraints depend on j in such a way that their scheduling times and their corresponding propositional choices must occur one after the other in time. More precisely, for every $j \in [n]$, t_{x_j} is going to be scheduled either at time j (if x_j is true in φ) or at time $j + 1$ (when instead x_j is false in φ). In addition to this, when x_j is quantified universally in φ (i.e., when $x_j \in P$), the observation event that determines its propositional value (i.e., p_{x_j}) is always scheduled at time $j - 1$ (and this leaves enough space for the reaction time; actually, an entire unit of time between time $j - 1$ and time j).

This concludes the description of our gadgets for simulating the chain of alternating quantifiers in the prefix of φ.

At this point, we have an additional node in V, named z', which is always scheduled at time $n + 1$; for this, $\mathcal{A}$ contains the following two standard LTHCs: $(z, z', n + 1, \lambda)$ and $(z', z, -n - 1, \lambda)$. Next, we shall describe two additional gadgets (that make use of z') for simulating the 3-CNF formula $\bigwedge_{i=1}^{m}(\alpha_i \vee \beta_i \vee \gamma_i)$, one for the literals, and one for the clauses. We have a gadget for the positive (i.e., x_j) and the negative (i.e., $\neg x_j$) literals. It goes as follows: for each $j \in [n]$, V contains two nodes named l_{x_j} (i.e., positive literal) and $l_{\overline{x}_j}$ (i.e., negative literal). Moreover, $\mathcal{A}$ contains the following four standard LTHCs, $(z', l_{x_j}, 1, \lambda), (l_{x_j}, z', 0, \lambda)$ and $(z', l_{\overline{x}_j}, 1, \lambda), (l_{\overline{x}_j}, z', 0, \lambda)$, plus the following *multi-head* LTHC,

$$A^h(l_{x_j}, l_{\overline{x}_j}) \triangleq \left(z', \{l_{x_j}, l_{\overline{x}_j}\}, \langle w(l_{x_j}), w(l_{\overline{x}_j}) \rangle = \langle 0, 0 \rangle, \langle L(l_{x_j}), L(l_{\overline{x}_j}) \rangle = \langle \lambda, \lambda \rangle \right),$$

and the following *multi-tail* LTHC,

$$A^t(l_{x_j}, l_{\overline{x}_j}) \triangleq \left(\{l_{x_j}, l_{\overline{x}_j}\}, z', \langle w(l_{x_j}), w(l_{\overline{x}_j}) \rangle = \langle -1, -1 \rangle, \langle L(l_{x_j}), L(l_{\overline{x}_j}) \rangle = \langle \lambda, \lambda \rangle \right).$$

The idea here is that the standard LTHCs are going to force the scheduling times of both l_{x_j} and $l_{\overline{x}_j}$ to fall within the real interval $[n + 1, n + 2]$ (i.e., not before z' and at most 1 time unit after z'). Meanwhile, the multi-head constraint $A^h(l_{x_j}, l_{\overline{x}_j})$ forces that at least one between l_{x_j} and $l_{\overline{x}_j}$ happen not later than time $n + 1$ (i.e., not later than the scheduling time of z'); similarly, the multi-tail constraint $A^t(l_{x_j}, l_{\overline{x}_j})$ is going to force that at least one between l_{x_j} and $l_{\overline{x}_j}$ happen not before time $n + 2$ (i.e., not before the scheduling time of z' plus 1). Therefore, exactly one between l_{x_j} and $l_{\overline{x}_j}$ will be forced to happen at time $n + 1$, and the other one at time $n + 2$.

Up to this point, the key idea is that, for every $j \in [n]$, we can force the scheduling time of each node l_{x_j} and $l_{\overline{x}_j}$ to be uniquely determined, according to a suitable translation of the scheduling time of t_{x_j}. Particularly, we want to schedule at time $n + 1$ (i.e., at the same scheduling time of z') the one node between l_{x_j} and $l_{\overline{x}_j}$ whose corresponding literal was chosen to be false in φ (that is l_{x_j} if t_{x_j} was scheduled at time j, and $l_{\overline{x}_j}$ if t_{x_j} was scheduled at time $j + 1$); similarly, we want to schedule at time $n + 2$ (i.e., at the same scheduling time of z' *plus* 1 time unit) the one node between l_{x_j} and $l_{\overline{x}_j}$ whose corresponding literal was chosen to be true (that is l_{x_j} if t_{x_j} was scheduled at time $j + 1$, and $l_{\overline{x}_j}$ if t_{x_j} was scheduled at time j). In order to achieve this, for each $j \in [n]$, $\mathcal{A}$ contains the following two standard LTHCs: $(t_{x_j}, l_{x_j}, n + 1 - j, \lambda)$ and $(l_{x_j}, t_{x_j}, -n - 1 + j, \lambda)$ (in Fig. 8a they are depicted with a unique arc $t_{x_j} \overset{[k,k],\lambda}{\longrightarrow} l_{x_j}$ where $k = n + 1 - j$); in this way, l_{x_j} is forced to happen at the same time of t_{x_j} *plus* $n + 1 - j$ units of time. Therefore, if t_{x_j} was scheduled at time j (i.e., x_j is false in φ), then node l_{x_j} is scheduled at time $j + n + 1 - j = n + 1$; otherwise, if t_{x_j} was scheduled at time $j + 1$ (i.e., x_j is true in φ), then node l_{x_j} is scheduled at time $j + 1 + n + 1 - j = n + 2$. At this point, the scheduling time of the node $l_{\overline{x}_j}$ is determined uniquely thanks to the hyperarcs $A^h(l_{x_j}, l_{\overline{x}_j}), A^t(l_{x_j}, l_{\overline{x}_j})$ and the standard constraints $(z', l_{\overline{x}_j}, 1, \lambda), (l_{\overline{x}_j}, z', 0, \lambda)$: if the node l_{x_j} is scheduled at time $n + 1$ (i.e., if x_j is false in φ), then $l_{\overline{x}_j}$ must be scheduled at time $n + 1 + 1 = n + 2$ (i.e., if $\overline{x}_j$ is true in φ) so that to satisfy $A^t(l_{x_j}, l_{\overline{x}_j})$ and $(z', l_{\overline{x}_j}, 1, \lambda)$; otherwise, if l_{x_j} is scheduled at time $n + 2$ (i.e., if x_j is true in φ), then $l_{\overline{x}_j}$ must be scheduled at time $n + 1 + 0 = n + 1$ (i.e., if $\overline{x}_j$ is false in φ) so that to satisfy $A^h(l_{x_j}, l_{\overline{x}_j})$ and $(l_{\overline{x}_j}, z', 0, \lambda)$. Notice that the literals $\alpha_i, \beta_i, \gamma_i$ of φ are thus instances of the nodes l_{x_i} or $l_{\overline{x}_i}$ described in Fig. 8a.

Finally, we describe the gadget for the clauses: for each $i \in [m]$, the CHyTN Γ_φ contains a node $\mathcal{C}_i$ for each clause $\mathcal{C}_i = (\alpha_i \vee \beta_i \vee \gamma_i)$ of φ; also, each node $\mathcal{C}_i$ is connected by:

(a) Gadget for 3-CNF-TQBF positive x_j and negative $\neg x_j$ literal.

(b) Gadget for 3-CNF-TQBF clause $\mathcal{C}_i = (\alpha_i \vee \beta_i \vee \gamma_i)$ where each $\alpha_i, \beta_i, \gamma_i$ is a positive or negative literal.
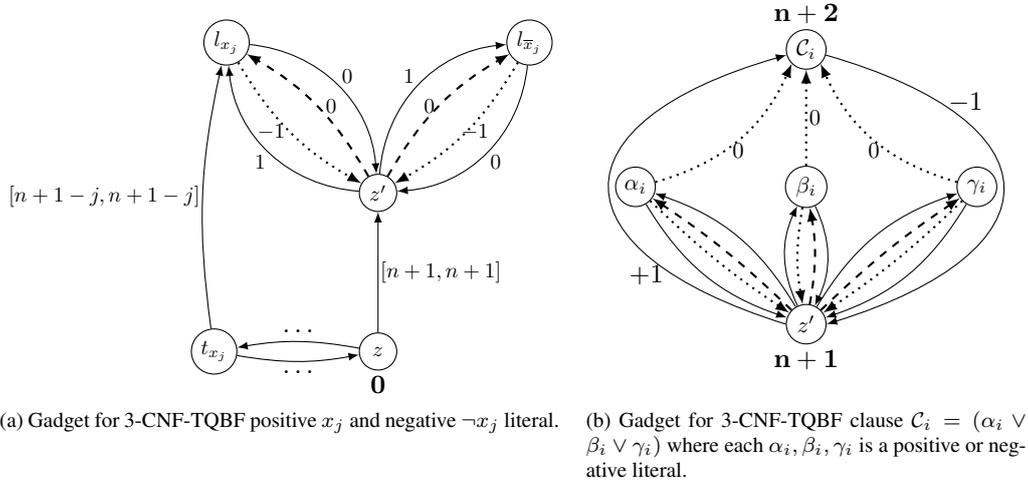
Figure 8: Gadgets used for variables and clauses in the reduction from 3-CNF-TQBF to General-CHyTN-DC.

– a multi-tail hyperarc with head in $\mathcal{C}_i$ and tails over the literals $\alpha_i, \beta_i, \gamma_i$ occurring in $\mathcal{C}_i$ and having weight 0 and label λ, i.e., by a multi-tail LTHC:

$$A^c(\alpha_i, \beta_i, \gamma_i) \triangleq \Big(\{\alpha_i, \beta_i, \gamma_i\}, \mathcal{C}_i, \langle w(\alpha_i), w(\beta_i), w(\gamma)_i \rangle = \langle 0, 0, 0 \rangle, \langle L(\alpha_i), L(\beta_i), L(\gamma)_i \rangle = \langle \lambda, \lambda, \lambda \rangle \Big),$$

for some literals $\alpha_i, \beta_i, \gamma_i \in \{l_{x_j}, l_{\overline{x}_j} \mid 1 \leq j \leq n\}$.

– two standard and opposite LTHCs, $(z', \mathcal{C}_i, 1, \lambda)$ and $(\mathcal{C}_i, z', -1, \lambda)$, with node z'.

See Fig. 8b for an illustration of the clauses' gadget; the dashed arrows form the multi-head LTHCs and the dotted arrows form the multi-tail LTHCs. Every node of Γ_φ has an empty label, i.e., $L(v) = \lambda$ for every $v \in V$. The rationale of the clauses' gadget is that, for each i, at least one of the $\alpha_i, \beta_i, \gamma_i$ must occur at the same time instant of $\mathcal{C}_i$ (i.e., at least one must occur at time $n + 2$, because one of the literals must be true)

This concludes our description of Γ_φ.

More formally and succinctly, the CHyTN $\Gamma_\varphi = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$ is defined as follows:

- $P \triangleq \{x_j \mid \text{``}\forall x_j\text{''}$ appears in the prefix of $\varphi\}$;

- – $V \triangleq \{z, z'\} \cup \{t_{x_j} \mid 1 \leq j \leq n\} \cup \{p_{x_j} \mid x_j \in P\} \cup$
    $\cup \{l_{x_j} \mid 1 \leq j \leq n\} \cup \{l_{\overline{x}_j} \mid 1 \leq j \leq n\} \cup \{\mathcal{C}_i \mid 1 \leq i \leq m\}$;

  – $\mathcal{OV} \triangleq \{p_{x_j} \mid x_j \in P\}$ and $\mathcal{O}(x_j) = p_{x_j}$ for every $x_j \in P$;

  – $L(v) = \lambda$ for every $v \in V$;

- $\mathcal{A} \triangleq \bigcup_{j: \text{``}\exists x_j\text{''} \in \varphi} \exists\text{-Qnt}_j \cup \bigcup_{j: \text{``}\forall x_j\text{''} \in \varphi} \forall\text{-Qnt}_j \cup \bigcup_{j=1}^{n} \text{Var}_j \cup \bigcup_{i=1}^{m} \text{Cla}_i \cup \big\{ (z, z', n+1, \lambda), (z', z, -n-1, \lambda) \big\}$,

  where:

  – $\exists\text{-Qnt}_j \triangleq \Big\{ (z, t_{x_j}, j+1, \lambda), (t_{x_j}, z, -j, \lambda) \Big\}$;
    This defines the existential quantifier gadget as depicted in Fig. 7a;

  – $\forall\text{-Qnt}_j \triangleq \Big\{ (z, p_{x_j}, j-1, \lambda), (p_{x_j}, z, -j+1, \lambda),$
    $(z, t_{x_j}, j+1, x_j), (t_{x_j}, z, -j-1, x_j), (z, t_{x_j}, j, \neg x_j), (t_{x_j}, z, -j, \neg x_j) \Big\}$;
    This defines the universal quantifier gadget as depicted in Fig. 7b;

16

- $\text{Var}_j = \Big\{ (z', l_{x_j}, 1, \lambda), (l_{x_j}, z', 0, \lambda), (z', l_{\overline{x}_j}, 1, \lambda), (l_{\overline{x}_j}, z', 0, \lambda),$

  $A_j^t \triangleq \Big(\{l_{x_j}, l_{\overline{x}_j}\}, z', \langle w_{A_j^t}(l_{x_j}), w_{A_j^t}(l_{\overline{x}_j}) \rangle = \langle -1, -1 \rangle, \langle L_{A_j^t}(l_{x_j}), L_{A_j^t}(l_{\overline{x}_j}) \rangle = \langle \lambda, \lambda \rangle \Big),$

  $A_j^h \triangleq \Big(z', \{l_{x_j}, l_{\overline{x}_j}\}, \langle w_{A_j^h}(l_{x_j}), w_{A_j^h}(l_{\overline{x}_j}) \rangle = \langle 0, 0 \rangle, \langle L_{A_j^h}(l_{x_j}), L_{A_j^h}(l_{\overline{x}_j}) \rangle = \langle \lambda, \lambda \rangle \Big),$

  $(t_{x_j}, l_{x_j}, n+1-j, \lambda), (l_{x_j}, t_{x_j}, -n-1+j, \lambda) \Big\}.$

  This defines the variable gadget for x_j as depicted in Fig. 8a;

- $\text{Cla}_i = \Big\{ (z', \mathcal{C}_j, 1), (\mathcal{C}_j, z', -1),$

  $A_i^c \triangleq \Big(\{\alpha_j, \beta_j, \gamma_j\}, \mathcal{C}_j, \langle w_{A_i^c}(\alpha_j), w_{A_i^c}(\beta_j), w_{A_i^c}(\gamma_j) \rangle = \langle 0, 0, 0 \rangle,$

  $\langle L_{A_i^c}(\alpha_j), L_{A_i^c}(\beta_j), L_{A_i^c}(\gamma_j) \rangle = \langle \lambda, \lambda, \lambda \rangle \Big) \Big\}.$

  This defines the clause gadget for clause $\mathcal{C}_j = (\alpha_i \vee \beta_i \vee \gamma_i)$ as depicted in Fig. 8b.

Notice that $|V| \leq 1 + 4n + m = O(m+n)$ and $m_\mathcal{A} \leq 16n + 5m = O(m+n)$; the transformation is thus linear.

<u>Correctness.</u> Let us show that φ is true if and only if Γ_φ is dynamically-consistent.

($\Rightarrow$) Assume φ is true, so Player-$\exists$ has a strategy to satisfy $\bigwedge_{i=1}^m (\alpha_i \vee \beta_i \vee \gamma_i)$ no matter how Player-$\forall$ decides to assign the universally quantified variables of φ. Suppose that Player-$\exists$ and Player-$\forall$ alternate their choices by assigning a truth value to the variables of φ; we can construct a dynamic and viable execution strategy $\sigma \in \mathcal{S}_{\Gamma_\varphi}$ for Γ_φ by reflecting these choices, as follows. The nodes z and z' are scheduled at time 0 and $n+1$ (respectively) under all possible scenarios. For each $j = 1, \ldots, n$, the node t_{x_j} is scheduled at time j if x_j is set to true in φ, either by Player-$\exists$ or Player-$\forall$, otherwise at time $j+1$; and, when x_j is quantified universally in φ, the node p_{x_j} is scheduled at time $j-1$ under all possible scenarios; also, the node l_{x_j} is scheduled at time $n+2$ if x_j is set to true in φ, either by Player-$\exists$ or Player-$\forall$, otherwise at time $n+1$; symmetrically, $l_{\overline{x}_j}$ is scheduled at time $n+1$ if x_j is true in φ, otherwise at time $n+2$. Finally, for each $i = 1, \ldots, m$, the node $\mathcal{C}_i$ is scheduled at time $n+2$ under all possible scenarios. It is easy to check that all LTHCs of Γ_φ are satisfied by σ under all possible scenarios, so σ is viable for Γ_φ; moreover, since σ reflects the alternating choices of Player-$\exists$ and Player-$\forall$, then σ is also dynamic. Therefore, Γ_φ is dynamically-consistent.

($\Leftarrow$) Vice versa, assume that Γ_φ is dynamically-consistent. Let $\sigma \in \mathcal{S}_{\Gamma_\varphi}$ be a viable and dynamic execution strategy for Γ_φ. Firstly, we argue that σ is integer valued, i.e., that $[\sigma(s)]_v \in \mathbf{Z}$ for every $v \in V$ and $s \in \Sigma_{\Gamma_\varphi}$. Indeed, since σ is viable, it is easy to check that the scheduling time of z, z', $\mathcal{C}_i$ (for every $i = 1, \ldots, m$) and p_{x_j} (for every universally quantified variable x_j in φ) is forced to be $0, n+1, n+2$ and $j-1$ (respectively); also, for each universally quantified variable x_j in φ, the scheduling time of p_{x_j} is forced to be $j-1$, and that of t_{x_j} is forced to be either j or $j+1$ according to whether x_j is true or false in the current scenario. Still, for each existentially quantified variable x_j in φ, the two standard LTHCs $(z, t_{x_j}, j+1, \lambda)$ and $(t_{x_j}, z, -j, \lambda)$ allow t_{x_j} to be scheduled anywhere within $[j, j+1]$, i.e., even at non-integer values. However, on one side, the scheduling time of l_{x_j} is forced to be that of t_{x_j} plus $n+1-j$, on the other side, l_{x_j} must be scheduled either at time $n+1$ or $n+2$ because of the multi-head $A^h(l_{x_j}, l_{\overline{x}_j})$ and multi-tail $A^t(l_{x_j}, l_{\overline{x}_j})$ LTHCs (respectively). Thus, for σ to be viable, t_{x_j} must be scheduled either at time j or $j+1$. Therefore, σ is integer valued. Now, suppose to execute σ step-by-step over the integer line; we can construct a strategy for Player-$\exists$ by reflecting the integer choices that the Planner makes to schedule the nodes of Γ_φ, as follows. For each existentially quantified variable x_j in φ, Player-$\exists$ sets x_j to true if the Planner schedules t_{x_j} at time $j+1$ (i.e., if l_{x_j} is scheduled at time $n+2$, and $l_{\overline{x}_j}$ at time $n+1$), and to false otherwise (i.e., if t_{x_j} is at time j, l_{x_j} at time $n+1$ and $l_{\overline{x}_j}$ at time $n+2$). Then, since σ is viable, for each clause $\mathcal{C}_i$ of φ, at least one of the literals $\alpha_i, \beta_i, \gamma_i$ must be true, thanks to the multi-tail LTHC $A^c(\alpha_i, \beta_i, \gamma_i)$; and since σ is also dynamic, then Player-$\exists$ wins, so φ is true.

To conclude, notice that any LTHC $A \in \mathcal{A}$ of Γ_φ has weights $w_A(\cdot) \in \{-1, 0, 1\}$ and size $|A| \leq 3$. Since any hyperarc with three heads (tails) can be replaced by two hyperarcs each having at most two heads (tails), then General-CHyTN-DC remains PSPACE-hard even if $w_A(\cdot) \in \{-1, 0, 1\}$ and $|A| \leq 2$ for every multi-tail/head LTHC $A \in \mathcal{A}$. Also notice that $w_a(\cdot) \in [-n-1, n+1] \cap \mathbf{Z}$ and $\ell_a \in \{p, \neg p \mid p \in P\} \cup \{\lambda\}$ holds for every weight w_a and label ℓ_a appearing in any *standard* LTHC $a \in \mathcal{A}$. This concludes the proof. $\qquad\square$

Theorem 7 motivates the study of consistency problems on CHyTNs having either only multi-head or only multi-

tail hyperarcs. Since we are interested in dynamic consistency, where time moves only forward of course, and the execution strategy depends only on past observations, from now on we shall consider only multi-head CHyTNs.

4.1. ϵ-dynamic consistency

In CHyTNs, decisions about the precise timing of actions are postponed until execution time, when information gathered from the execution of the observation events can be taken into account. However, the Planner is allowed to factor in an observation, and modify its strategy in response to it, only strictly after the observation has been made (whence the strict inequality in Definition 13). Notice that this definition does not take into account the actual reaction time, which, in most applications, is non-negligible. In order to deliver algorithms that can also deal with the *reaction time* ϵ of the Planner we now introduce ϵ-dynamic consistency, a refined notion of dynamic consistency. The intuition underlying Definition 18 is that to model a specific kind of disjunctive constraint: given a small real number $\epsilon > 0$, for any two scenarios $s_1, s_2 \in \Sigma_P$ and any event $u \in V^+_{s_1,s_2}$, the scheduling time of u under s_1 must be greater or equal to either that of u under s_2 or that of v under s_2 plus ϵ for at least one $v \in \Delta(s_1; s_2)$. Let us remind the fact that, from now on, our CHyTNs admit only multi-head hyperarcs. The definition of ϵ-dynamic consistency follows below.

Definition 18 (ϵ-dynamic consistency)**.** *Given any CHyTN $\langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$ and any real number $\epsilon \in (0, \infty)$, an execution strategy $\sigma \in \mathcal{S}_\Gamma$ is ϵ-dynamic if it satisfies all the H_ϵ-constraints, namely, for any two scenarios $s_1, s_2 \in \Sigma_P$ and any event $u \in V^+_{s_1,s_2}$, the execution strategy σ satisfies the following constraint $H_\epsilon(s_1; s_2; u)$:*

$$[\sigma(s_1)]_u \geq \min \Big(\big\{ [\sigma(s_2)]_u \big\} \cup \big\{ [\sigma(s_1)]_v + \epsilon \mid v \in \Delta(s_1; s_2) \big\} \Big)$$

We say that a CHyTN Γ is ϵ-dynamically-consistent if it admits $\sigma \in \mathcal{S}_\Gamma$ which is both viable and ϵ-dynamic.
   The problem of checking whether a given CHyTN is ϵ-dynamically-consistent is named CHyTN-ϵ-DC.

It follows directly from Definition 18 that, whenever $\sigma \in S_\Gamma$ satisfies some $H_\epsilon(s_1; s_2; u)$, then σ satisfies $H_{\epsilon'}(s_1; s_2; u)$ for every $\epsilon' \in (0, \epsilon]$ as well. This proves the following lemma.

Lemma 5. *Let Γ be a CHyTN. If Γ is ϵ-dynamically-consistent for some real $\epsilon > 0$, then Γ is ϵ'-dynamically-consistent for every $\epsilon' \in (0, \epsilon]$.*

Given any dynamically-consistent CHyTN, we may ask for the maximum reaction time ϵ of the Planner beyond which the network is no longer dynamically-consistent.

Definition 19 (Reaction time $\hat{\epsilon}$)**.** *Let Γ be a CHyTN. Let $\hat{\epsilon} \triangleq \hat{\epsilon}(\Gamma)$ be the least upper bound of the set of all real numbers $\epsilon > 0$ such that Γ is ϵ-dynamically-consistent, i.e.,*

$$\hat{\epsilon} \triangleq \hat{\epsilon}(\Gamma) \triangleq \sup\{\epsilon > 0 \mid \Gamma \text{ is } \epsilon\text{-dynamically-consistent}\}.$$

Let us consider the (affinely) extended real numbers $\overline{\mathbf{R}} \triangleq \mathbf{R} \cup \{-\infty, \infty\}$, where *every* subset S of $\overline{\mathbf{R}}$ has an infimum and a supremum. Particularly, recall that $\sup \emptyset = -\infty$ and, if S is unbounded above, then $\sup S = \infty$.

If Γ is dynamically-consistent, then $\hat{\epsilon}(\Gamma)$ exists and $\hat{\epsilon}(\Gamma) \neq -\infty$ (i.e., the set on which we have taken the supremum in Definition 19 is non-empty), as it is now proved in Lemma 6.

Lemma 6. *Let σ be a dynamic execution strategy for the CHyTN Γ. Then, there exists a sufficiently small real number $\epsilon \in (0, \infty)$ such that σ is ϵ-dynamic.*

Proof. Let $s_1, s_2 \in \Sigma_P$ be two scenarios and let us consider any event $u \in V^+_{s_1,s_2}$. Since σ is dynamic, then by Lemma 4 the following implication necessarily holds:

$$\Big(\bigwedge_{v \in \Delta(s_1;s_2)} [\sigma(s_1)]_u \leq [\sigma(s_1)]_v \Big) \Rightarrow [\sigma(s_1)]_u \geq [\sigma(s_2)]_u \tag{*}$$

Notice that, w.r.t. Lemma 4, we have relaxed the equality $[\sigma(s_1)]_u = [\sigma(s_2)]_u$ in the implicand of (L4) by introducing the inequality $[\sigma(s_1)]_u \geq [\sigma(s_2)]_u$. At this point, we convert (*) from implicative to disjunctive form, first by applying

18

the rule of material implication[2], and then De Morgan's law[3], resulting in the following equivalent expression:

$$\Big([\sigma(s_1)]_u \geq [\sigma(s_2)]_u\Big) \vee \Big(\bigvee_{v \in \Delta(s_1;s_2)} [\sigma(s_1)]_u > [\sigma(s_1)]_v \Big) \qquad (**)$$

Then, we argue that there exists a real number $\epsilon \in (0, \infty)$ such that the following disjunction holds as well:

$$\Big([\sigma(s_1)]_u \geq [\sigma(s_2)]_u\Big) \vee \Big(\bigvee_{v \in \Delta(s_1;s_2)} [\sigma(s_1)]_u \geq [\sigma(s_1)]_v + \epsilon \Big).$$

In fact, since the disjunction $(**)$ necessarily holds, then one can pick the following real number $\epsilon > 0$:

$$\epsilon \triangleq \min_{\langle s_1, s_2, u \rangle \in \Sigma_P \times \Sigma_P \times V^+_{s_1,s_2}} \epsilon(s_1; s_2; u),$$

where the values $\epsilon(s_1; s_2; u) \in (0, \infty)$ are defined as follows, for every $\langle s_1, s_2, u \rangle \in \Sigma_P \times \Sigma_P \times V^+_{s_1,s_2}$:

$$\epsilon(s_1; s_2; u) \triangleq \left\{ \begin{array}{l} 1, \text{ if } [\sigma(s_1)]_u \geq [\sigma(s_2)]_u; \\ \min\Big\{ [\sigma(s_1)]_u - [\sigma(s_1)]_v \mid v \in \Delta(s_1; s_2), \; [\sigma(s_1)]_u > [\sigma(s_1)]_v \Big\}, \text{ otherwise.} \end{array} \right.$$

This implies that σ satisfies every H_ϵ-constraint of Γ, thus σ is ϵ-dynamic. $\qquad \square$

Next, we prove a converse formulation of Lemma 6.

Lemma 7. *Let σ be an ϵ-dynamic execution strategy for a CHyTN Γ, for some real number $\epsilon \in (0, \infty)$. Then, σ is dynamic.*

Proof. For the sake of contradiction, let us suppose that σ is not dynamic. Let F be the set of all the triplets $\langle u, s_1, s_2 \rangle \in V^+_{s_1,s_2} \times \Sigma_P \times \Sigma_P$, for which the implication (L4) given in Lemma 4 does not hold. Notice, $F \neq \emptyset$; indeed, since σ is not dynamic, by Lemma 4 there exists at least one $\langle u, s_1, s_2 \rangle$ for which (L4) doesn't hold. So, it holds that $\langle u, s_1, s_2 \rangle \in F$ if and only if the following two properties hold:

1. $[\sigma(s_1)]_u \leq [\sigma(s_1)]_v$, for every $v \in \Delta(s_1; s_2)$;
2. $[\sigma(s_1)]_u \neq [\sigma(s_2)]_u$.

Let $\langle \hat{u}, \hat{s_1} \rangle$ be an event whose scheduling time $[\sigma(\hat{s_1})]_{\hat{u}}$ is minimum and for which (1) and (2) hold, namely, let:

$$\langle \hat{u}, \hat{s_1} \rangle \triangleq \arg\min \Big\{ [\sigma(s_1)]_u \mid \exists s_2 \; \langle u, s_1, s_2 \rangle \in F \Big\}.$$

Since $\langle \hat{u}, \hat{s_1} \rangle$ is minimum in $[\sigma(\hat{s_1})]_{\hat{u}}$, then $[\sigma(\hat{s_1})]_{\hat{u}} \leq [\sigma(s_2)]_{\hat{u}}$ for every $s_2 \in \Sigma_P$ such that $\langle \hat{u}, \hat{s_1}, s_2 \rangle \in F$; moreover, since $\langle \hat{u}, \hat{s_1}, s_2 \rangle \in F$, then $[\sigma(\hat{s_1})]_{\hat{u}} \neq [\sigma(s_2)]_{\hat{u}}$ holds by (2), so that $[\sigma(\hat{s_1})]_{\hat{u}} < [\sigma(s_2)]_{\hat{u}}$. At this point, recall that σ is ϵ-dynamic by hypothesis, hence $[\sigma(\hat{s_1})]_{\hat{u}} < [\sigma(s_2)]_{\hat{u}}$ implies that there exists $v \in \Delta(\hat{s_1}; s_2)$ such that:

$$[\sigma(\hat{s_1})]_{\hat{u}} \geq [\sigma(\hat{s_1})]_v + \epsilon > [\sigma(\hat{s_1})]_v,$$

but this inequality contradicts item (1) above. Indeed, $F = \emptyset$ and σ is thus dynamic. $\qquad \square$

In Section 5, the following theorem is proved.

Theorem 8. *For any dynamically-consistent CHyTN Γ, where V is the set of events and Σ_P is the set of scenarios, it holds that $\hat{\epsilon}(\Gamma) \geq |\Sigma_P|^{-1}|V|^{-1}$.*

Notice that one really needs to consider rational values for $\hat{\epsilon}$, as it is shown in the following example.

---

[2]The rule of material implication: $\models p \Rightarrow q \iff \neg p \vee q$.
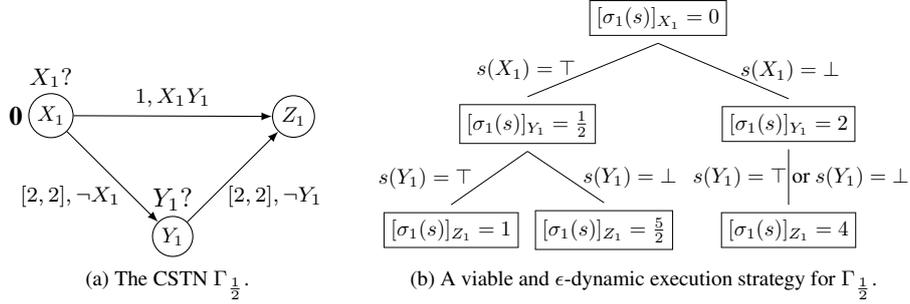[3]De Morgan's law: $\models \neg(p \wedge q) \iff \neg p \vee \neg q$.

(a) The CSTN $\Gamma_{\frac{1}{2}}$.

(b) A viable and ϵ-dynamic execution strategy for $\Gamma_{\frac{1}{2}}$.

Figure 9: A dynamically-consistent CSTN whose viable and dynamic execution strategies are fractional.

Example 7. *Consider the CSTN $\Gamma_{\frac{1}{2}}$ shown in Fig. 9a. The Planner needs to schedule and to observe X_1 at time 0 under all possible scenarios. But it is not viable to schedule Y_1 or Z_1 at time 0, because X_1 and Y_1 may turn out to be $\perp$; so Y_1 and Z_1 both need to be scheduled strictly after 0. Next, assume that X_1 turns out to be $\top$ at time 0. Then, it is not viable to schedule Y_1 at time 1, because Z_1 needs to be scheduled within time 1 if Y_1 is $\top$ and strictly after otherwise, and the Planner can't react instantaneously to the observation made at Y_1. Thus, if X_1 is $\top$ at time 0, then Y_1 needs to be scheduled at time $t \in (0, 1)$, e.g., $t = \frac{1}{2}$. The corresponding execution strategy is shown in Fig. 9b.*

Also notice that, in Definition 15, dynamic consistency was defined by strict-inequality and equality constraints. However, by Theorem 9, dynamic consistency can also be defined in terms of H_ϵ-constraints only (i.e., no strict-inequalities are required).

Theorem 9. *Let Γ be a CHyTN. Let $\epsilon \triangleq |\Sigma_P|^{-1}|V|^{-1}$. Then, Γ is dynamically-consistent if and only if Γ is ϵ-dynamically-consistent.*

By Theorem 9, any algorithm for checking ϵ-dynamic consistency can be used to check dynamic consistency.

4.2. A (pseudo) Singly-Exponential Time Algorithm for CSTN-DC and CHyTN-DC

In this section, we present a (pseudo) singly-exponential time algorithm for solving CSTN-DC and CHyTN-DC, also producing a dynamic execution strategy whenever the input CHyTN is dynamically-consistent.

The main result of this paper is summarized in the following theorem, which is proven in the next Section 5.

Theorem 10. *The following two algorithmic results hold for CHyTNs.*

1. *There exists an $O\big(|\Sigma_P|^2|\mathcal{A}|m_{\mathcal{A}} + |\Sigma_P|^3|V||\mathcal{A}||P| + |\Sigma_P|^3|V|m_{\mathcal{A}} + |\Sigma_P|^4|V|^2|P|\big)WD$ time deterministic algorithm for deciding CHyTN-ϵ-DC on input $\langle \Gamma, \epsilon \rangle$, for any CHyTN $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$ and any rational number $\epsilon = N/D$ where $N, D \in \mathbf{N}_+$. Particularly, given any ϵ-dynamically-consistent CHyTN Γ, the algorithm returns as output a viable and ϵ-dynamic execution strategy $\sigma \in \mathcal{S}_\Gamma$.*

2. *There exists an $O\big(|\Sigma_P|^3|V||\mathcal{A}|m_{\mathcal{A}} + |\Sigma_P|^4|V|^2|\mathcal{A}||P| + |\Sigma_P|^4|V|^2m_{\mathcal{A}} + |\Sigma_P|^5|V|^3|P|\big)W$ time deterministic algorithm for checking CHyTN-DC on any input $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$. Particularly, given any dynamically-consistent CHyTN Γ, it returns as output a viable and dynamic execution strategy $\sigma \in \mathcal{S}_\Gamma$.*

Here, $W \triangleq \max_{a \in A} |w_a|$.

Since every CSTN is also a CHyTN, Theorem 10 holds for CSTNs as well.

We now present the reduction from CHyTN-DC to HEAD-HYTN-CONSISTENCY. Again, since any CSTN is a CHyTN, the same argument reduces CSTN-DC to HEAD-HYTN-CONSISTENCY. Firstly, we argue that any CHyTN can be viewed as a succinct representation which can be expanded into an exponentially sized HyTN.

The *Expansion* of CSTNs is introduced below.

Definition 20 (Expansion $\langle V_\Gamma^{\text{Ex}}, \Lambda_\Gamma^{\text{Ex}} \rangle$). *Let* $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$ *be a CHyTN. Consider the family of distinct and disjoint HyTNs* $\langle V_s, \mathcal{A}_s \rangle$, *one for each scenario* $s \in \Sigma_P$, *which is defined as follows (where* $v_s \triangleq (v, s)$ *for every* $v \in V$ *and* $s \in \Sigma_P$):

$$V_s \triangleq \{v_s \mid v \in V_s^+\},$$

$$\mathcal{A}_s \triangleq \Big\{ \big(t_s, \underbrace{\{h_s^{(1)}, \ldots, h_s^{(k)}\}}_{\text{heads labeled with } s}, \underbrace{\langle w(h_s^{(1)}), \ldots, w(h_s^{(k)}) \rangle}_{\text{corresponding weights}} \big) \Big| $$

$$ \big(\underbrace{t}_{\text{tail}}, \underbrace{\{h^{(1)}, \ldots, h^{(k)}\}}_{\text{heads}}, \underbrace{\langle w(h^{(1)}), \ldots, w(h^{(k)}) \rangle}_{\text{corresponding weights}} \big) \in \mathcal{A}_s^+ \Big\}.$$

(Of course, in the above notation, $k = 1$ when Γ is a CSTN, whereas $k \in \mathbf{N}_+$ when Γ is a CHyTN.)

Next, we define the expansion $\langle V_\Gamma^{Ex}, \Lambda_\Gamma^{Ex} \rangle$ *of* Γ *as follows:*

$$\langle V_\Gamma^{Ex}, \Lambda_\Gamma^{Ex} \rangle \triangleq \Big(\bigcup_{s \in \Sigma_P} V_s, \bigcup_{s \in \Sigma_P} \mathcal{A}_s \Big).$$

Notice that $V_{s_1} \cap V_{s_2} = \emptyset$ whenever $s_1 \neq s_2$ and that $\langle V_\Gamma^{\text{Ex}}, \Lambda_\Gamma^{\text{Ex}} \rangle$ is an STN/HyTN with at most $|V_\Gamma^{\text{Ex}}| \leq |\Sigma_P| \cdot |V|$ nodes and size at most $|\Lambda_\Gamma^{\text{Ex}}| \leq |\Sigma_P| \cdot |\mathcal{A}|$.

We now show that the expansion of a CHyTN can be enriched with some (extra) multi-head hyperarcs in order to model ϵ-dynamic consistency, by means of a particular HyTN which is denoted by $\mathcal{H}_\epsilon(\Gamma)$.

Definition 21 (HyTN $\mathcal{H}_\epsilon(\Gamma)$). *Let* $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$ *be a CHyTN. Given any real number* $\epsilon \in (0, \infty)$, *the HyTN* $\mathcal{H}_\epsilon(\Gamma)$ *is defined as follows:*

- *For every two scenarios* $s_1, s_2 \in \Sigma_P$ *and for every event node* $u \in V_{s_1, s_2}^+$, *define a hyperarc* $\alpha \triangleq \alpha_\epsilon(s_1; s_2; u)$ *as follows (with the intention to model* $H_\epsilon(s_1; s_2; u)$ *from Def. 18):*

$$\alpha_\epsilon(s_1; s_2; u) \triangleq (t_\alpha, H_\alpha, w_\alpha), \quad \forall s_1, s_2 \in \Sigma_P \text{ and } u \in V_{s_1, s_2}^+.$$

  where:

  – $t_\alpha \triangleq u_{s_1}$ *is the tail of the (multi-head) hyperarc* $\alpha_\epsilon(s_1; s_2; u)$;
  – $H_\alpha \triangleq \{u_{s_2}\} \cup \Delta(s_1; s_2)$ *is the set of the heads of* $\alpha_\epsilon(s_1; s_2; u)$;
  – $w_\alpha(u_{s_2}) \triangleq 0$, *and* $w_\alpha(v) \triangleq -\epsilon$ *for each* $v \in \Delta(s_1; s_2)$.

- *Consider the expansion* $\langle V_\Gamma^{Ex}, \Lambda_\Gamma^{Ex} \rangle$ *of* Γ. *Then,* $\mathcal{H}_\epsilon(\Gamma)$ *is defined as* $\mathcal{H}_\epsilon(\Gamma) \triangleq (V_\Gamma^{Ex}, \mathcal{A}_{H_\epsilon})$, *where,*

$$\mathcal{A}_{H_\epsilon} \triangleq \Lambda_\Gamma^{Ex} \cup \bigcup_{\substack{s_1, s_2 \in \Sigma_P \\ u \in V_{s_1, s_2}^+}} \alpha_\epsilon(s_1; s_2; u).$$

Notice that each $\alpha_\epsilon(s_1; s_2; u)$ has size $|\alpha_\epsilon(s_1; s_2; u)| = 1 + \Delta(s_1; s_2) \leq 1 + |P|$.

Here below, Algorithm 1 provides a pseudocode for constructing $\mathcal{H}_\epsilon(\Gamma)$.

Example 8. *An excerpt of the HyTN* $\mathcal{H}_\epsilon(\Gamma_0)$ *corresponding to the CSTN* Γ_0 *of Fig. 3a is depicted in Fig. 10; here, two scenarios* $s_1 \triangleq p \wedge q$ *and* $s_4 \triangleq \neg p \wedge \neg q$ *are considered, on top we have* $\Gamma_{0 s_1}^+$, *whereas* $\Gamma_{0 s_4}^+$ *is below, finally, the corresponding hyperconstraints* $H_\epsilon(s_1; s_4; u)$ *and* $H_\epsilon(s_4; s_1; u)$ *are depicted as dashed hyperarcs.*

The following establishes the connection between dynamic consistency of CHyTNs and consistency of HyTNs.

Theorem 11. *Given any CHyTN* $\Gamma = \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{O}V, P \rangle$, *there exists a sufficiently small real number* $\epsilon \in (0, \infty)$ *such that the CHyTN* Γ *is dynamically-consistent if and only if the HyTN* $\mathcal{H}_\epsilon(\Gamma)$ *is consistent.*

Moreover, the HyTN $\mathcal{H}_\epsilon(\Gamma)$ *has at most* $|V_{\mathcal{H}_\epsilon}| \leq |\Sigma_P| \cdot |V|$ *nodes,* $|\mathcal{A}_{\mathcal{H}_\epsilon}| = O(|\Sigma_P||\mathcal{A}| + |\Sigma_P|^2|V|)$ *hyperarcs, and it has size* $m_{\mathcal{A}_{\mathcal{H}_\epsilon}} = O(|\Sigma_P|m_\mathcal{A} + |\Sigma_P|^2|V||P|)$.

Algorithm 1: construct_$\mathcal{H}(\Gamma, \epsilon)$

Input: a CHyTN $\Gamma \triangleq \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$, a rational number $\epsilon > 0$

1 **foreach** *($s \in \Sigma_P$)* **do**

2 $\quad\big|\quad V_s \leftarrow \{v_s \mid v \in V_s^+\}$; // see Def. 20

3 $\quad\big\lfloor\quad A_s \leftarrow \{a_s \mid a \in A_s^+\}$; // see Def. 20

4 $V_\Gamma^{\text{Ex}} \leftarrow \cup_{s \in \Sigma_P} V_s$;

5 $\Lambda_\Gamma^{\text{Ex}} \leftarrow \cup_{s \in \Sigma_P} A_s$;

6 **foreach** *($s_1, s_2 \in \Sigma_P$ and $u \in V_{s_1,s_2}^+$)* **do**

7 $\quad\big|\quad t_\alpha \leftarrow u_{s_1}$;

8 $\quad\big|\quad H_\alpha \leftarrow \{u_{s_2}\} \cup \Delta(s_1; s_2)$;

9 $\quad\big|\quad w_\alpha(u_{s_2}) \leftarrow 0$;

10 $\quad\big|\quad$ **foreach** $v \in \Delta(s_1; s_2)$ **do**

11 $\quad\big|\quad\big\lfloor\quad w_\alpha(v_{s_1}) \leftarrow -\epsilon$;

12 $\quad\big\lfloor\quad \alpha_\epsilon(s_1; s_2; u) \leftarrow (t_\alpha, H_\alpha, w_\alpha)$;

13 $\mathcal{A}_{\mathcal{H}_\epsilon} \leftarrow \Lambda_\Gamma^{\text{Ex}} \cup \bigcup_{\substack{s_1, s_2 \in \Sigma_P \\ u \in V_{s_1, s_2}^+}} \alpha_\epsilon(s_1; s_2; u)$;

14 $\mathcal{H}_\epsilon(\Gamma) \leftarrow (V_\Gamma^{\text{Ex}}, \mathcal{A}_{H_\epsilon})$;

15 **return** $H_\epsilon(\Gamma)$;

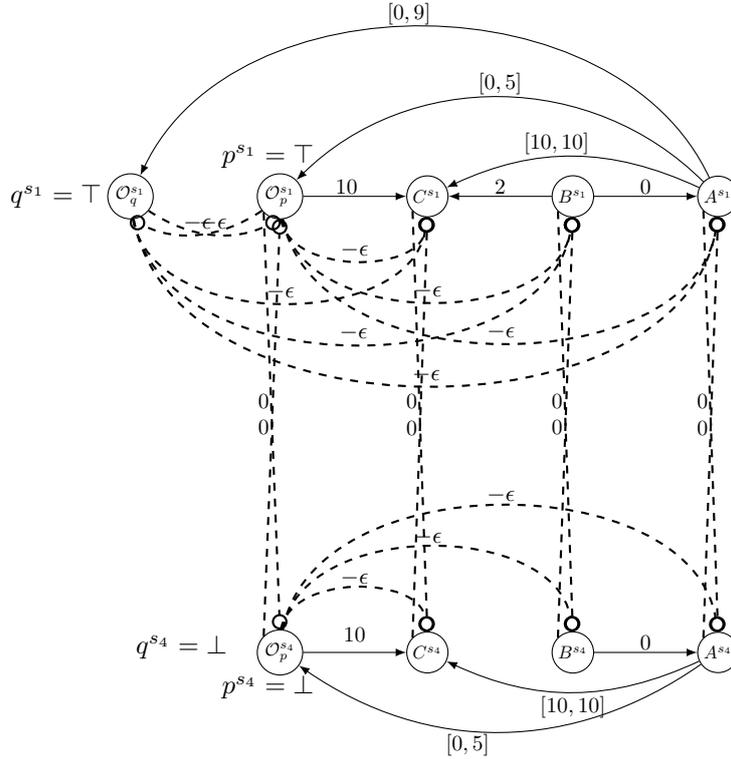Algorithm 1: Constructing $\mathcal{H}_\epsilon(\Gamma)$.

Figure 10: An excerpt of the HyTN $\mathcal{H}_\epsilon(\Gamma_0)$ corresponding to the CSTN Γ_0 of Fig. 3a, in which two scenarios, s_1 and s_4, are considered.

Proof. For any real number $\epsilon \in (0, \infty)$, let $\mathcal{H}_\epsilon(\Gamma) = \langle V_\Gamma^{\text{Ex}}, \mathcal{A}_{H_\epsilon} \rangle$ be the HyTN of Definition 21.

(1) By Definitions 20 and 21, $|V_{\mathcal{H}_\epsilon}| = |V_\Gamma^{\text{Ex}}| \leq |\Sigma_P| \cdot |V|$; also, $|\mathcal{A}_{\mathcal{H}_\epsilon}| = |\Lambda_\Gamma^{\text{Ex}}| + \left| \bigcup_{s_1, s_2 \in \Sigma_P; u \in V_{s_1, s_2}^+} \alpha_\epsilon(s_1; s_2; u) \right| = O(|\Sigma_P||\mathcal{A}| + |\Sigma_P|^2|V|)$, and since $\alpha_\epsilon(s_1; s_2; u)$ has at most P heads, then $m_{\mathcal{A}_{\mathcal{H}_\epsilon}} = O(|\Sigma_P|m_\mathcal{A} + |\Sigma_P|^2|V||P|)$.

(2) We claim that, for any $\epsilon > 0$, $\mathcal{H}_\epsilon(\Gamma)$ is consistent if and only if Γ is ϵ-dynamically-consistent.

($\Rightarrow$) Given any feasible schedule $\phi : V_\Gamma^{\text{Ex}} \to \mathbf{R}$ for the HyTN $\mathcal{H}_\epsilon(\Gamma)$, let $\sigma_\phi(s) \in \mathcal{S}_\Gamma$ be the execution strategy defined as follows:

$$[\sigma_\phi(s)]_v \triangleq \phi(v_s), \text{ for every } v_s \in V_\Gamma^{\text{E}}, \text{ where } v \in V \text{ and } s \in \Sigma_P.$$

Notice that each hyperarc $\alpha_\epsilon(s_1; s_2; u)$ is satisfied by ϕ if and only if the corresponding H_ϵ-constraint $H_\epsilon(s_1; s_2; u)$ is satisfied by σ_ϕ; moreover, recall that $\Lambda_\Gamma^{\text{Ex}} \subseteq \mathcal{A}_{H_\epsilon}$, and that $\Lambda_\Gamma^{\text{Ex}}$ contains all the *original* standard/hyper difference constraints of Γ (i.e., those induced by $\mathcal{A}$, by means of Def. 20). At this point, since ϕ is feasible for the HyTN $\mathcal{H}_\epsilon(\Gamma)$, then σ_ϕ must be viable and ϵ-dynamic for Γ (because it satisfies all the required constraints).

Therefore, Γ is ϵ-dynamically-consistent.

($\Leftarrow$) Given any viable and ϵ-dynamic execution strategy $\sigma \in \mathcal{S}_\Gamma$, for some real number $\epsilon \in (0, \infty)$, let $\phi_\sigma : V_\Gamma^{\text{Ex}} \to \mathbf{R}$ be the schedule of the HyTN $\mathcal{H}_\epsilon(\Gamma)$ defined as follows:

$$\phi_\sigma(v_s) \triangleq [\sigma(s)]_v \text{ for every } v_s \in V_\Gamma^{\text{Ex}}, \text{ where } v \in V \text{ and } s \in \Sigma_P.$$

Also in this case, we have that $\Lambda_\Gamma^{\text{Ex}} \subseteq \mathcal{A}_{H_\epsilon}$, and a moment's reflection reveals that each hyperarc $\alpha_\epsilon(s_1; s_2; u)$ is satisfied by ϕ_σ if and only if $H_\epsilon(s_1; s_2; u)$ is satisfied by σ. At this point, since σ is viable and ϵ-dynamic for the CHyTN Γ, then ϕ_σ must be feasible for $\mathcal{H}_\epsilon(\Gamma)$. Therefore, $\mathcal{H}_\epsilon(\Gamma)$ is consistent.

This proves that, for any $\epsilon \in (0, \infty)$, $\mathcal{H}_\epsilon(\Gamma)$ is consistent if and only if Γ is ϵ-dynamically-consistent.

(3) At this point, by composition with (1), Lemma 6 implies that there exists a sufficiently small real number $\epsilon \in (0, \infty)$ such that Γ is dynamically-consistent if and only if $\mathcal{H}_\epsilon(\Gamma)$ is consistent. $\square$

At this point, we are in the position to show the pseudocode for checking CHyTN-ϵ-DC, it is given in Algorithm 2:

Algorithm 2: `check_CHyTN-ε-DC`(Γ, ϵ)

Input: a CHyTN $\Gamma \triangleq \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$, a rational number $\epsilon \triangleq N/D$, for $N, D \in \mathbf{N}_+$
1 $\mathcal{H}_\epsilon(\Gamma) \leftarrow$ `construct_H`(Γ, ϵ); // ref. Algorithm 1
2 foreach $(A = \langle t_A, H_A, w_A \rangle \in \mathcal{A}_{\mathcal{H}_\epsilon(\Gamma)}$ **and** $h \in H_A)$ **do**
3 $w_A(h) \leftarrow w_A(h) \cdot D$; // scale all weights of $\mathcal{H}_\epsilon(\Gamma)$, from $\mathbf{Q}$ to $\mathbf{Z}$

4 $\phi \leftarrow$ `check_HEAD-HyTN-CONSISTENCY`$(\mathcal{H}_\epsilon(\Gamma))$; // ref. Thm 5
5 if (ϕ *is a feasible schedule of* $\mathcal{H}_\epsilon(\Gamma)$) **then**
6   **foreach** (*event node* $v \in V_{\mathcal{H}_\epsilon(\Gamma)}$) **do**
7 $\phi(v) \leftarrow \phi(v)/D$; // re-scale back to size the scheduling time, from $\mathbf{Z}$ to $\mathbf{Q}$, w.r.t. ϵ
8   **return** $\langle YES, \phi \rangle$;

9 else return NO ;

Algorithm 2: Checking CHyTN-ϵ-DC on input (Γ, ϵ).

whereas, the pseudocode for checking CHyTN-DC is provided in Algorithm 3, here below:

Algorithm 3: `check_CSTN-DC/CHyTN-DC`(Γ)

Input: a CHyTN $\Gamma \triangleq \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$
1 $\hat{\epsilon} \leftarrow |\Sigma_P|^{-1}|V|^{-1}$; // ref. Thm. 9
2 return `check_CHyTN-ε-DC`$(\Gamma, \hat{\epsilon})$;

Algorithm 3: Checking CHyTN-DC on input Γ.

Notice that the latter (Algorithm 3) invokes the former (Algorithm 2); more details follow.

Description of Algorithm 3. Firstly, Algorithm 3 computes a sufficiently small *rational* number $\epsilon \in (0, \infty) \cap \mathbf{Q}$, by relying on Theorem 9, i.e., it is set $\hat{\epsilon} \triangleq |\Sigma_P|^{-1}|V|^{-1}$ (line 1). Secondly, Algorithm 2 is invoked on input $(\Gamma, \hat{\epsilon})$. At this point, Algorithm 2 firstly constructs $\mathcal{H}_{\hat{\epsilon}}(\Gamma)$ (line 1 of Algorithm 2) by invoking Algorithm 1, and then it scales *every* hyperarc's weight, appearing in $\mathcal{H}_{\hat{\epsilon}}(\Gamma)$, from $\mathbf{Q}$ to $\mathbf{Z}$ (at lines 2-3). This is done by multiplying each weight by a factor D (line 3), where $D \in \mathbf{N}_+$ is the denominator of $\hat{\epsilon}$ (i.e., $D = |\Sigma_P| \cdot |V|$). Thirdly, $\mathcal{H}_{\hat{\epsilon}}(\Gamma)$ is solved with the HEAD-HYTN-CONSISTENCY-Checking algorithm underlying Theorem 5 (at line 4), i.e., within the underlying algorithmic engine, an instance of the HEAD-HYTN-CONSISTENCY problem is solved by reducing it to the problem of determining winning regions in a carefully constructed MPG (see Comin et al. (2014, 2016) for the details of such a reduction). At this point, if the HEAD-HYTN-CONSISTENCY algorithm outputs YES, together with a feasible schedule ϕ of $\mathcal{H}_{\hat{\epsilon}}(\Gamma)$, then the time values of ϕ are scaled back to size w.r.t. $\hat{\epsilon}$, and then $\langle \text{YES}, \phi \rangle$ is returned as output (lines 5-8); otherwise, the output is simply NO (at line 9). Still, notice that, thanks to Item 3 of Theorem 5, we could also return a negative certificate, because negative instances are well characterized in terms of generalized negative cycles (see Definition 5).

Remark 2. *The same algorithm, with essentially the same upper bound on its running time and space, works also in case we allow for arbitrary boolean formulae as labels, rather than just conjunctions.*

Remark 3. *We remark that the HyTN/MPG algorithm that is at the heart of our approach requires integer weights (i.e., it requires that $w(u, v) \in \mathbf{Z}$ for every $(u, v) \in A$); somehow, we could not play it differently (see Comin et al. (2014, 2016) for a discussion). Moreover, the algorithm always computes an integer solution to HyTNs/MPGs and, therefore, it always computes* rational *feasible schedules for the CHyTNs given as input. As such, it seems to us that this "requirement" actually turns out to be a plus in practice. It is actually the integer assumption that allows us to analyze the algorithm quantitatively, also presenting a sharp lower bounding analysis on the critical value of the reaction time $\hat{\epsilon}$, where the CHyTN transits from being, to not being, dynamically-consistent. We believe that these issues deserve much attention, and going into them required a "discrete" approach to the notion of numbers.*

The correctness and the time complexity of Algorithms 2 and 3 is analyzed in Section 5.

5. Bounding Analysis on the Reaction Time $\hat{\epsilon}$

In this section we present an asymptotically sharp lower bound for $\hat{\epsilon}(\Gamma)$, that is the critical value of reaction time where the CHyTN transits from being, to not being, dynamically-consistent. The proof technique introduced in this analysis is applicable more generally, when dealing with linear difference constraints which include strict inequalities. This bound implies that Algorithm 3 is a (pseudo) singly-exponential time algorithm for solving CHyTN-DC.

To begin, we are going to provide a proof of Theorem 8; for this, let us firstly introduce some further notation.

Let $\Gamma \triangleq \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$ be a dynamically-consistent CHyTN. By Theorem 11, there exists $\epsilon > 0$ such that the HyTN $\mathcal{H}_\epsilon(\Gamma)$ is consistent. Then, let $\phi : V_\Gamma^{\text{Ex}} \to \mathbf{R}$ be a feasible schedule for $\mathcal{H}_\epsilon(\Gamma)$. For any hyperarc $A = \langle t_A, H_A, w_A \rangle \in \mathcal{A}_{\mathcal{H}_\epsilon}$, define a standard arc a_A as follows:

$$a_A \triangleq \langle t_A, \hat{h}, w_A(\hat{h}) \rangle, \text{ where } \hat{h} \triangleq \arg \min_{h \in H_A} \big(\phi(h) - w_A(h) \big).$$

Then, notice that the network $T_\epsilon^\phi(\Gamma) \triangleq \langle V_\Gamma^{\text{Ex}}, \bigcup_{A \in \mathcal{A}_{\mathcal{H}_\epsilon}} a_A \rangle$ is always an STN. Moreover, a moment's reflection reveals that, by definition of $\hat{h}$ as above, then ϕ is a feasible schedule for the STN $T_\epsilon^\phi(\Gamma)$.

At this point, assuming $v \in V_\Gamma^{\text{Ex}}$, let us consider the *fractional part* r_v of ϕ_v, i.e.,

$$r_v \triangleq \phi_v - \lfloor \phi_v \rfloor.$$

Then, let $R \triangleq \{r_v\}_{v \in V_\Gamma^{\text{Ex}}}$ be the set of all the fractional parts induced by V_Γ^{Ex}. Sort R by the common ordering on $\mathbf{R}$ and assume that $S \triangleq \{r_1, \ldots, r_k\}$ is the resulting ordered set (without repetitions), i.e., $|S| = k$, $S = R$, $r_1 < \ldots < r_k$. Now, let $\text{pos}(v) \in [k]$ be the (unique) index position such that:

$$r_{\text{pos}(v)} = r_v.$$

Then, we define a new fractional part r'_v as follows:

$$r'_v \triangleq \frac{\mathrm{pos}(v) - 1}{|\Sigma_P| \cdot |V|}, \tag{NFP}$$

and a new schedule function as follows:

$$\phi'_v \triangleq \lfloor \phi_v \rfloor + r'_v. \tag{NSF}$$

Then the following holds.

Remark 4. *Notice that (NFP) doesn't alter the ordering relation among the fractional parts, i.e.,*

$$r'_u < r'_v \iff r_u < r_v, \text{ for any } u, v \in V_\Gamma^{Ex}.$$

Moreover, since $\big(\mathrm{pos}(v) - 1\big) < |\Sigma_P| \cdot |V|$, observe that (NSF) doesn't change the value of any integer part, i.e.,

$$\lfloor \phi'_u \rfloor = \lfloor \phi_u \rfloor, \text{ for any } u \in V_\Gamma^{Ex}.$$

We are now in the position to prove Theorem 8.

Proof of Theorem 8. Let $\Gamma \triangleq \langle V, \mathcal{A}, L, \mathcal{O}, \mathcal{OV}, P \rangle$ be dynamically-consistent. By Theorem 11 there exists $\epsilon' > 0$ such that $\mathcal{H}_{\epsilon'}(\Gamma)$ is consistent and it admits some *feasible* schedule $\phi : V_\Gamma^{Ex} \to \mathbf{R}$. As mentioned, ϕ is feasible for the STN $T_{\epsilon'}^\phi(\Gamma)$. Now, let $\hat{\epsilon} \triangleq |\Sigma_P|^{-1}|V|^{-1}$. Moreover, let $T_{\hat{\epsilon}}^\phi(\Gamma)$ be the STN obtained from $T_{\epsilon'}^\phi(\Gamma)$ simply by replacing, in the weights of the arcs, each weight $-\epsilon'$ with $-\hat{\epsilon}$. We argue that ϕ' (as defined in (NSF) w.r.t. ϕ, V, Σ_P), is a feasible schedule for the STN $T_{\hat{\epsilon}}^\phi(\Gamma)$. Indeed, every constraint of $T_{\hat{\epsilon}}^\phi(\Gamma)$ has form $\phi_v - \phi_u \leq w$, for some $w \in \mathbf{Z}$ or $w = -\hat{\epsilon}$.

- Consider the case $w \in \mathbf{Z}$. Notice that $\phi_v - \phi_u \leq w$ holds because ϕ is feasible for the STN $T_{\epsilon'}^\phi(\Gamma)$. Then, it is not difficult to see that $\phi'_v - \phi'_u \leq w$ holds as well, because of Remark 4.

- Consider the case $w = -\hat{\epsilon}$. Notice that $\phi_v - \phi_u \leq -\epsilon'$ holds because ϕ is feasible for the STN $T_{\epsilon'}^\phi(\Gamma)$.

  Then, notice that the following implication always holds,

  $$\phi_v - \phi_u \leq -\epsilon' \implies \phi_v \neq \phi_u.$$

  Hence, again by Remark 4, we can conclude that $\phi'_v \neq \phi'_u$. At this point, we observe that the temporal distance between ϕ'_u and ϕ'_v is, therefore, at least $\hat{\epsilon}$ by definition of (NSF) and (NFP), i.e.,

  $$\phi'_u - \phi'_v \geq |\Sigma_P|^{-1}|V|^{-1} = \hat{\epsilon}.$$

  That is to say, $\phi'_v - \phi'_u \leq -\hat{\epsilon}$.

This proves that ϕ' is a feasible schedule also for the STN $T_{\hat{\epsilon}}^\phi(\Gamma)$. Since $T_{\hat{\epsilon}}^\phi(\Gamma)$ is thus consistent, then, a moment's reflection reveals that $\mathcal{H}_{\hat{\epsilon}}(\Gamma)$ is consistent as well thanks to the same schedule ϕ'.

Therefore, by Theorem 11, the CHyTN Γ is $\hat{\epsilon}$-dynamically-consistent, provided that $\hat{\epsilon} \triangleq |\Sigma_P|^{-1}|V|^{-1}$. $\qquad\square$

The correctness proof and the time complexity of Algorithm 3 is given next.

Proof of Theorem 10. To begin, notice that some of the temporal constraints introduced during the reduction step depend on a sufficiently small parameter $\hat{\epsilon} \in (0, \infty) \cap \mathbf{Q}$, whose magnitude turns out to depend on the size of the input CHyTN. It is proved below that the time complexity of the algorithm depends multiplicatively on D, where $\hat{\epsilon} = N/D$ for some $N, D \in \mathbf{N}_+$. By Theorem 8, $\hat{\epsilon}(\Gamma) \geq |\Sigma_P|^{-1}|V|^{-1}$; so line 1 of Algorithm 3 is correct. Therefore, as a corollary of Theorem 11, we obtain that Algorithm 3 correctly decides CSTN-DC.

Concerning its time complexity, the most time-expensive step of the algorithm is clearly line 4 of Algorithm 2, which relies on Theorem 5 in order to solve an instance of Head-HyTN-Consistency on input $\mathcal{H}_\epsilon(\Gamma)$. From Theorem 11 we have an upper bound on the size of $\mathcal{H}_\epsilon(\Gamma)$, while Theorem 5 gives us a pseudo-polynomial upper bound for the computation time. Also, recall that we scale weights by a factor D at lines 2-3 of Algorithm 2, where

$\hat{\epsilon} = N/D$ for some $N, D \in \mathbf{N}_+$. Thus, by composition, Algorithm 3 decides CHyTN-DC in a time T_Γ^{Algo3} which is bounded as follows, where $W \triangleq \max_{a \in A} |w_a|$ and $D \in \mathbf{N}_+$:

$$T_\Gamma^{\text{Algo3}} = O\Big(\big(|V_{\mathcal{H}_\epsilon(\Gamma)}| + |\mathcal{A}_{\mathcal{H}_\epsilon(\Gamma)}|\big)m_{\mathcal{A}_{\mathcal{H}_\epsilon}(\Gamma)}\Big)WD.$$

Whence, taking into account the upper bound on the size of $\mathcal{H}_\epsilon(\Gamma)$ give by Theorem 11, the following holds:

$$T_\Gamma^{\text{Algo3}} = O\Big(\big(|\Sigma_P||V| + |\Sigma_P||\mathcal{A}| + |\Sigma_P|^2|V|\big)\big(|\Sigma_P|m_{\mathcal{A}} + |\Sigma_P|^2|V||P|\big)\Big)WD$$

$$= O\Big(\underline{|\Sigma_P|^2|V|m_{\mathcal{A}}} + \underline{|\Sigma_P|^3|V|^2|P|} + |\Sigma_P|^2|\mathcal{A}|m_{\mathcal{A}} + |\Sigma_P|^3|\mathcal{A}||V||P| + |\Sigma_P|^3|V|m_{\mathcal{A}} + |\Sigma_P|^4|V|^2|P|\Big)WD$$

$$= O\Big(|\Sigma_P|^2|\mathcal{A}|m_{\mathcal{A}} + |\Sigma_P|^3|\mathcal{A}||V||P| + |\Sigma_P|^3|V|m_{\mathcal{A}} + |\Sigma_P|^4|V|^2|P|\Big)WD$$

By Theorem 8, it is sufficient to check ϵ-dynamic consistency for $\hat{\epsilon} = |\Sigma_P|^{-1}|V|^{-1}$.
Therefore, the following worst-case time bound holds on Algorithm 3:

$$T_\Gamma^{\text{Algo3}} = O\Big(|\Sigma_P|^3|V||\mathcal{A}|m_{\mathcal{A}} + |\Sigma_P|^4|\mathcal{A}||V|^2|P| + |\Sigma_P|^4|V|^2 m_{\mathcal{A}} + |\Sigma_P|^5|V|^3|P|\Big)W.$$

Since $|\Sigma_P| \leq 2^{|P|}$, the (pseudo) singly-exponential time bound follows. $\qquad\square$

At this point, a natural question is whether the lower bound given by Theorem 8 can be improved up to $\hat{\epsilon}(\Gamma) = \Omega(|V|^{-1})$. In turn, this would improve the time complexity of Algorithm 3 by a factor $|\Sigma_P|$. However, the following theorem shows that this is not the case, by exhibiting a CSTN for which $\hat{\epsilon}(\Gamma) = 2^{-\Omega(|P|)}$. This proves that the lower bound given by Theorem 8 is (almost) asymptotically sharp.

Theorem 12. *For each $n \in \mathbf{N}_+$ there exists a CSTN Γ^n such that:*

$$\hat{\epsilon}(\Gamma^n) < 2^{-n+1} = 2^{-|P^n|/3+1},$$

where P^n is the set of boolean variables of Γ^n.

Proof. For each $n \in \mathbf{N}_+$, we define a CSTN $\Gamma^n \triangleq \langle V^n, A^n, L^n, \mathcal{O}^n, \mathcal{O}V^n, P^n\rangle$ as follows.
See Fig. 11 for a clarifying illustration.

- $V^n \triangleq \{X_i, Y_i, Z_i \mid 1 \leq i \leq n\}$;

- $A^n \triangleq B \cup \bigcup_{i=1}^n C_i \cup \bigcup_{i=1}^{n-1} D_i$
  where:

  - $B \triangleq \{\langle X_1 - v \leq 0, \lambda\rangle \mid v \in V^n\} \cup \{\langle Z_1 - X_1 \leq 1, X_1 \wedge Y_1\rangle\}$;
  - $C_i \triangleq \{\langle Y_i - X_i \leq 2, \neg X_i\rangle, \langle X_i - Y_i \leq -2, \neg X_i\rangle, \langle Z_i - Y_i \leq 2, \neg Y_i\rangle, \langle Y_i - Z_i \leq -2, \neg Y_i\rangle\}$;
  - $D_i \triangleq \{\langle X_{i+1} - X_i \leq 5, Z_i\rangle, \langle X_i - X_{i+1} \leq -5, Z_i\rangle, \langle X_{i+1} - Y_i \leq 5, \neg Z_i\rangle, \langle Y_i - X_{i+1} \leq -5, \neg Z_i\rangle, \langle Z_{i+1} - Y_i \leq 5, Z_i \wedge X_{i+1} \wedge Y_{i+1}\rangle, \langle Y_i - Z_{i+1} \leq -5, Z_i \wedge X_{i+1} \wedge Y_{i+1}\rangle, \langle Z_{i+1} - Z_i \leq 5, \neg Z_i \wedge X_{i+1} \wedge Y_{i+1}\rangle, \langle Z_i - Z_{i+1} \leq -5, \neg Z_i \wedge X_{i+1} \wedge Y_{i+1}\rangle\}$;

- $L^n(v) \triangleq \lambda$ for every $v \in V^n$; $\mathcal{O}V^n \triangleq V^n$; $\mathcal{O}^n(v) \triangleq v$ for every $v \in \mathcal{O}V^n$; $P^n \triangleq V^n$.

We exhibit an execution strategy $\sigma_n : \Sigma_{P^n} \to \Phi_{V^n}$, which we will show is dynamic and viable for Γ^n.
Let $\{\delta_i\}_{i=1}^n$ and $\{\Delta_i\}_{i=1}^n$ be two real valued sequences such that:

$$(1)\ \Delta_1 \triangleq 1;\ (2)\ 0 < \delta_i < \Delta_i;\ (3)\ \Delta_i \triangleq \min(\delta_{i-1}, \Delta_{i-1} - \delta_{i-1}).$$

Then, the following also holds for every $1 \leq i \leq n$:

$$(4)\ 0 < \Delta_i \leq 2^{-i+1},$$

where the equality holds if and only if $\delta_i = \Delta_i/2$.
Hereafter, provided that $s \in \Sigma_P$ and $\ell \in P^*$, we will denote $\mathbb{1}_{s(\ell)} \triangleq 1$ if $s(\ell) = \top$ and $\mathbb{1}_{s(\ell)} \triangleq 0$ if $s(\ell) = \bot$.
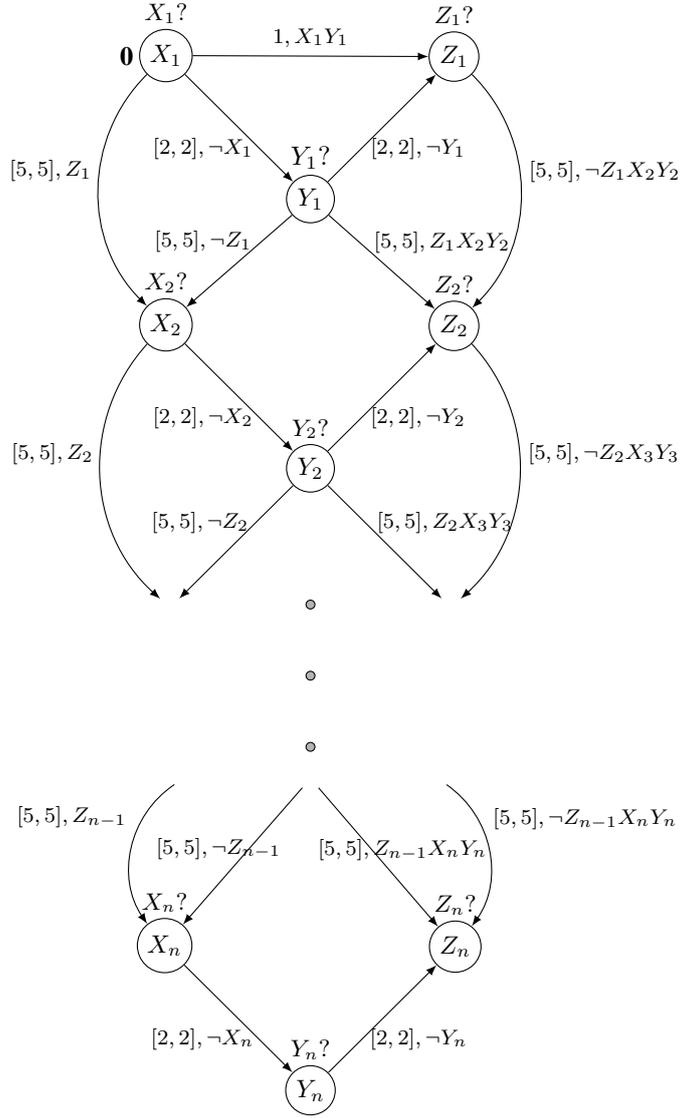We are in the position to define $\sigma_n(s)$ for any $s \in \Sigma_P$:

Figure 11: A CSTN Γ^n such that $\hat{\epsilon}(\Gamma^n) = 2^{-\Omega(|P^n|)}$.

- $[\sigma_n(s)]_{X_1} \triangleq 0;$

- $[\sigma_n(s)]_{Y_1} \triangleq \delta_1 \mathbb{1}_{s(X_1)} + 2\mathbb{1}_{s(\neg X_1)};$

- $[\sigma_n(s)]_{Z_1} \triangleq \mathbb{1}_{s(X_1 \wedge Y_1)} + (2 + [\sigma_n(s)]_{Y_1})\mathbb{1}_{s(\neg X_1 \vee \neg Y_1)};$

- $[\sigma_n(s)]_{X_i} \triangleq 5 + [\sigma_n(s)]_{X_{i-1}}\mathbb{1}_{s(Z_{i-1})} +$
  $+ [\sigma_n(s)]_{Y_{i-1}}\mathbb{1}_{s(\neg Z_{i-1})}$, for any $2 \le i \le n;$

- $[\sigma_n(s)]_{Y_i} \triangleq [\sigma_n(s)]_{X_i} + \delta_i \mathbb{1}_{s(X_i)} + 2\mathbb{1}_{s(\neg X_i)}$, for any $2 \le i \le n;$

- $[\sigma_n(s)]_{Z_i} \triangleq \big(5 + [\sigma_n(s)]_{Y_{i-1}}\mathbb{1}_{s(Z_{i-1})} +$
  $+ [\sigma_n(s)]_{Z_{i-1}}\mathbb{1}_{s(\neg Z_{i-1})}\big)\mathbb{1}_{s(X_i \wedge Y_i)} +$
  $+ (2 + [\sigma_n(s)]_{Y_i})\mathbb{1}_{s(\neg X_i \vee \neg Y_i)}$, for any $2 \le i \le n;$

27

Let us prove, by induction on $n \geq 1$, that σ_n is viable and dynamic for Γ^n.

- *Base case.* Let $n = 1$. Notice that Γ^1 almost coincides with the CSTN $\Gamma_{\frac{1}{2}}$ described in Example 7; so, it is really needed that $0 < \delta_1 < 1$. Then, by construction, σ_1 leads to the schedule depicted in Figure 12. This shows that σ_1 is viable and dynamic for Γ^1.
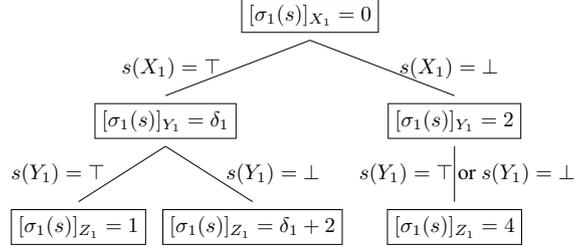

Figure 12: A viable and dynamic execution strategy for the base case $n = 1$.

- *Inductive step.* Let us assume that σ_{n-1} is viable and dynamic for Γ^{n-1}. Then, by construction, $[\sigma_n(s)]_v = [\sigma_{n-1}(s)]_v$ for every $s \in \Sigma_P$ and $v \in V_{n-1}$. Hence, by induction hypothesis, σ_n is viable and dynamic on V_{n-1}. Moreover, by construction, σ_n leads to the schedule depicted in Figure 13 and Figure 14. This shows that σ_n is viable and dynamic even on $V_n \setminus V_{n-1}$. Thus, σ_n is viable and dynamic for Γ^n, i.e., Γ^n is dynamically-consistent.
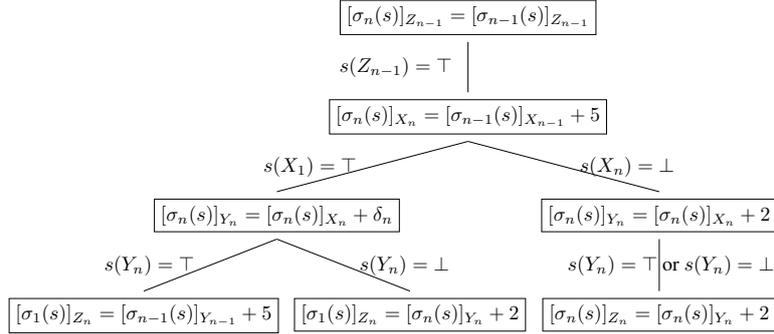

Figure 13: A viable and dynamic execution strategy for the inductive step $n - 1 \rightsquigarrow n$ when $s(Z_{n-1}) = \top$.

We claim that $\hat{\epsilon}(\Gamma^n) < 2^{-n+1} = 2^{-|P^n|/3+1}$ for every $n \geq 1$. Consider the following scenario $\hat{s}$ for $1 \leq i \leq n$:

$$\hat{s}(X_i) \triangleq \hat{s}(Y_i) \triangleq \top; \quad \hat{s}(Z_i) \triangleq \begin{cases} \top, & \text{if } \delta_i \leq \Delta_i/2 \\ \bot, & \text{if } \delta_i > \Delta_i/2 \end{cases}$$

We shall assume that σ is an execution strategy for Γ^n and study necessary conditions to ensure that σ is viable and dynamic, provided that the observations follow the scenario $\hat{s}$. First, σ must schedule X_1 at time $[\sigma(\hat{s})]_{X_1} = 0$. Then, since $\hat{s}(X_1) = \top$, we must have $0 < [\sigma(\hat{s})]_{Y_1} < 1$, because of the constraint $(Z_1 - X_1 \leq 1, X_1 \wedge Y_1)$. Stated otherwise, it is necessary that:

$$0 < [\sigma(\hat{s})]_{Y_1} - [\sigma(\hat{s})]_{X_1} < \Delta_1.$$

After that, since $\hat{s}(Y_1) = \top$, then σ must schedule Z_1 at time $[\sigma(\hat{s})]_{Z_1} = 1 = \Delta_1$. A moment's reflection reveals that almost identical necessary conditions now recur for X_2, Y_2, Z_2, with the crucial variation that it will be necessary to require: $0 < [\sigma(\hat{s})]_{Y_2} < \Delta_2$. Indeed, by proceeding inductively, it will be necessary that for every $1 \leq i \leq n$ and every $n \in \mathbf{N}_+$:

$$0 < [\sigma(\hat{s})]_{Y_i} - [\sigma(\hat{s})]_{X_i} < \Delta_i.$$
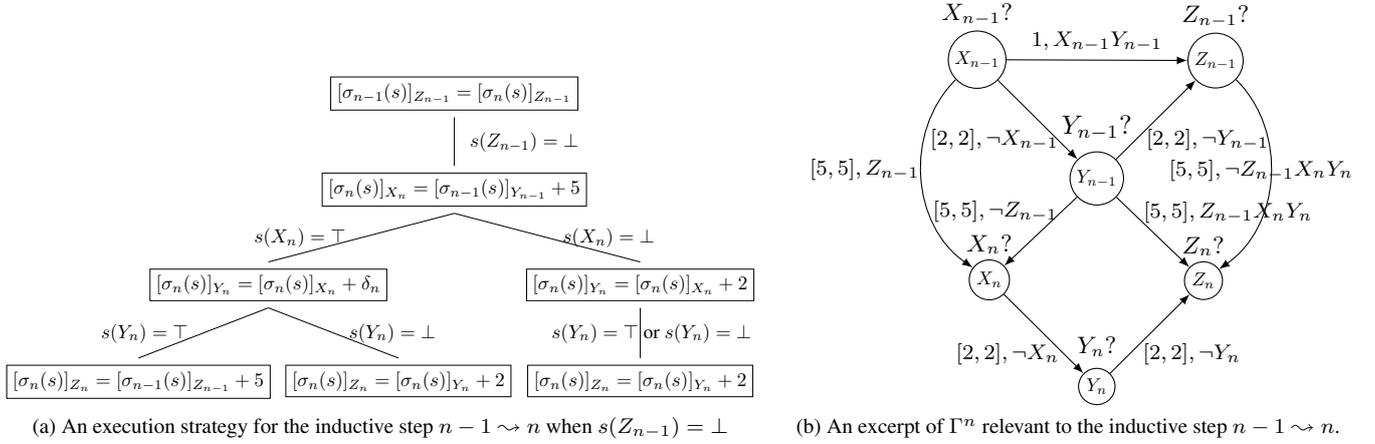
28

$$[\sigma_{n-1}(s)]_{Z_{n-1}} = [\sigma_n(s)]_{Z_{n-1}}$$
$$s(Z_{n-1}) = \bot$$
$$[\sigma_n(s)]_{X_n} = [\sigma_{n-1}(s)]_{Y_{n-1}} + 5$$
$$s(X_n) = \top \qquad s(X_n) = \bot$$
$$[\sigma_n(s)]_{Y_n} = [\sigma_n(s)]_{X_n} + \delta_n \qquad [\sigma_n(s)]_{Y_n} = [\sigma_n(s)]_{X_n} + 2$$
$$s(Y_n) = \top \qquad s(Y_n) = \bot \qquad s(Y_n) = \top \text{ or } s(Y_n) = \bot$$
$$[\sigma_n(s)]_{Z_n} = [\sigma_{n-1}(s)]_{Z_{n-1}} + 5 \qquad [\sigma_n(s)]_{Z_n} = [\sigma_n(s)]_{Y_n} + 2 \qquad [\sigma_n(s)]_{Z_n} = [\sigma_n(s)]_{Y_n} + 2$$

(a) An execution strategy for the inductive step $n - 1 \rightsquigarrow n$ when $s(Z_{n-1}) = \bot$ (b) An excerpt of Γ^n relevant to the inductive step $n - 1 \rightsquigarrow n$.

Figure 14: The inductive step $n - 1 \rightsquigarrow n$ when $s(Z_{n-1}) = \bot$.

As already observed in (4), $0 < \Delta_n \leq 2^{-n+1}$. Thus, any viable and dynamic execution strategy σ for Γ^n must satisfy:

$$0 < [\sigma(\hat{s})]_{Y_n} - [\sigma(\hat{s})]_{X_n} < \frac{1}{2^{n-1}} = \frac{1}{2^{|P^n|/3-1}}.$$

Therefore, once the Planner has observed the outcome $\hat{s}(X_n) = \top$ from the observation event X_n, then he must react by scheduling Y_n within time $2^{-n+1} = 2^{-|P^n|/3+1}$ in the future w.r.t. $[\sigma(\hat{s})]_{X_n}$.

Therefore, $\hat{\epsilon}(\Gamma^n) < 2^{-n+1} = 2^{-|P^n|/3+1}$ holds for every $n \geq 1$. $\qquad\qquad\square$

6. Related Works

This section discusses of some alternative approaches offered by the current literature. Recall that the article of Tsamardinos et al. (2003) has already been discussed in the introduction.

The work of Cimatti et al. (2014) provided the first sound-and-complete algorithm for checking the dynamic controllability of CSTNs with Uncertainty (CSTNU), and thus it can be employed for checking the dynamic consistency of CSTNs as a special case. The algorithm reduces the DC-Checking of CSTNUs to the problem of solving Timed Game Automata (TGA). Nevertheless, no worst-case upper bound on the time complexity of the procedure was provided in Cimatti et al. (2014). Still, one may observe that solving TGAs is a problem of much higher complexity than solving MPGs. Compare the following known facts: solving 1-player TGAs is PSPACE-complete and solving 2-player TGAs is EXP-complete; on the contrary, the problem of determining MPGs lies in NP∩coNP and it is currently an open problem to prove whether it is in P. Indeed, the algorithm offered in Cimatti et al. (2014) has not been proven to be singly-exponential time bounded, to the best of our knowledge it is still open whether singly-exponential time TGA-based algorithms for CSTN-DC do exist.

Next, a sound algorithm for checking the dynamic controllability of CSTNUs was given by Combi, Hunsberger, Posenato in (Combi et al., 2013). However, it was not shown to be complete. To the best of our knowledge, it is currently open whether or not it can be extended in order to prove completeness w.r.t. the CSTNU model.

Regarding the particular CSTN model, Hunsberger et al. (2015) presented, at the same conference in which the preliminary version of this work appeared, a sound-and-complete DC-checking algorithm for CSTNs. It is based on the propagation of temporal constraints labeled by propositions. However, to the best of our knowledge, the worst-case complexity of the algorithm is currently unsettled. Also notice that the algorithm in Hunsberger et al. (2015) works on CSTNs only, regardless of the CHyTN model. Indeed, we believe that our approach (based on tractable games plus reaction-time $\hat{\epsilon}$) and the approach of Hunsberger et al. (2015) (based on the propagation of labeled temporal constraints) can benefit from each other; for instance, recently Hunsberger and Posenato (2016) presented an alternative, equivalent semantics for ϵ-dynamic consistency, as well as a sound-and-complete ϵ-DC-checking algorithm based on the propagation of labeled constraints.

Finally, in Cairo et al. (2016), it is introduced and studied π-DC, a sound notion of dynamic consistency with an instantaneous reaction time, i.e., one in which the Planner is allowed to react to any observation at the same instant of time in which the observation is made. It turns out that π-DC is not equivalent to ϵ-DC with $\epsilon = 0$, and that the latter is actually inadequate for modeling an instantaneous reaction-time. Still, a simple reduction from π-DC-Checking to DC-Checking is identified; combined with Theorem 10, this provides a π-DC-Checking procedure whose time complexity remains (pseudo) singly-exponential in the number of propositional variables.

7. Conclusion

In this work we introduced the *Conditional Hyper Temporal Network (CHyTN)* model, a natural extension and generalization of both the CSTN and the HyTN model which is obtained by blending them together. We proved that deciding whether a given CSTN or CHyTN is dynamically-consistent is coNP-hard, and that deciding whether a given CHyTN is dynamically-consistent is PSPACE-hard, provided that the input instances are allowed to include both multi-head and multi-tail hyperarcs. In light of this, we focused on CHyTNs that allow only multi-head hyperarcs, and offered the first deterministic (pseudo) singly-exponential time algorithm for the problem of checking the dynamic consistency of multi-head CHyTNs, also producing a dynamic execution strategy whenever the input CHyTN is dynamically-consistent. As a byproduct, this provides the first sound-and-complete (pseudo) singly-exponential time algorithm for checking the dynamic consistency of CSTNs. The algorithm is based on a novel connection between CHyTNs and Mean Payoff Games. The presentation of such connection was mediated by the HyTN model. The algorithm actually manages a few more general variants of the problem; e.g., those where labels are not required to be conjunctions. To summarize, at the heart of the algorithm a reduction to MPGs is mediated by the HyTN model. The CHyTN is dynamically-consistent if and only if the corresponding MPG is everywhere won, and a dynamic execution strategy can be conveniently read out by an everywhere winning positional strategy. The size of this MPG is at most polynomial in the number of the possible scenarios; as such, the term at the corresponding time complexity exponent is linear, at worst, in the number of the observation events. The same holds for the running time of the resulting algorithm. In order to analyze the algorithm, we introduced a refined notion of dynamic consistency, named ϵ-dynamic consistency, also presenting a sharp lower bounding analysis on the critical value of the reaction time $\hat{\varepsilon}$ where a CHyTN transits from being, to not being, dynamically-consistent.

In future works we would like to settle the exact computational complexity of CSTN-DC/CHyTN-DC, as well as to extend our approach in order to check the dynamic controllability of CSTN with Uncertainty (Hunsberger et al., 2012). An extensive experimental evaluation taking good account of optimizations and heuristics is also planned.

References

X. Allamigeon, P. Benchimol, S. Gaubert, The Tropical Shadow-Vertex Algorithm Solves Mean Payoff Games in Polynomial Time on Average, in: ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, 89–100, doi:\bibinfo{doi}{10.1007/978-3-662-43948-7_8}, 2014.

R. Bellman, On a Routing Problem, Quarterly of Applied Mathematics 16 (1) (1958) 87–90.

C. Bettini, X. S. Wang, S. Jajodia, Temporal Reasoning in Workflow Systems, Dist. & Paral. Data. 11 (3) (2002) 269–306, doi:\bibinfo{doi}{10.1023/A:1014048800604}.

L. Brim, J. Chaloupka, Using Strategy Improvement to Stay Alive, Int. J. Found. Comput. Sci. 23 (3) (2012) 585–608, doi:\bibinfo{doi}{10.1142/S0129054112400291}.

L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, J. F. Raskin, Faster algorithms for mean-payoff games, Formal Methods in System Design 38 (2) (2011) 97–118, ISSN 0925-9856, doi:\bibinfo{doi}{10.1007/s10703-010-0105-x}.

M. Cairo, C. Comin, R. Rizzi, Instantaneous Reaction-Time in Dynamic-Consistency Checking of Conditional Simple Temporal Networks, in: 23rd International Symposium on Temporal Representation and Reasoning, TIME 2016, Kongens Lyngby, Denmark, October 17-19, 2016, 80–89, doi:\bibinfo{doi}{10.1109/TIME.2016.16}, 2016.

K. Chatterjee, M. Henzinger, S. Krinninger, D. Nanongkai, Polynomial-Time Algorithms for Energy Games with Special Weight Structures, Algorithmica 70 (3) (2014) 457–492, doi:\bibinfo{doi}{10.1007/s00453-013-9843-7}.

A. Cimatti, L. Hunsberger, A. Micheli, R. Posenato, M. Roveri, Sound and Complete Algorithms for Checking the Dynamic Controllability of Temporal Networks with Uncertainty, Disjunction and Observation, in: 21st Intern. Symp. on Temp. Repres. and Reasoning, TIME 2014, Verona, Italy, 27–36, doi:\bibinfo{doi}{10.1109/TIME.2014.21}, 2014.

C. Combi, M. Gambini, S. Migliorini, R. Posenato, Representing Business Processes Through a Temporal Data-Centric Workflow Modeling Language: An Application to the Management of Clinical Pathways, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 44 (9) (2014) 1182–1203, ISSN 2168-2216, doi:\bibinfo{doi}{10.1109/TSMC.2014.2300055}.

C. Combi, M. Gozzi, R. Posenato, G. Pozzi, Conceptual Modeling of Flexible Temporal Workflows, ACM Trans. Auton. Adapt. Syst. 7 (2) (2012) 19:1–19:29, ISSN 1556-4665, doi:\bibinfo{doi}{10.1145/2240166.2240169}.

C. Combi, L. Hunsberger, R. Posenato, An Algorithm for Checking the Dynamic Controllability of a Conditional Simple Temporal Network with Uncertainty, in: ICAART 2013 - Proc. of the 5th Intern. Conf. on Agents and Artif. Intell., Vol. 2, Spain, 2013, 144–156, doi:\bibinfo{doi}{10.1007/978-3-662-44440-5_19}, 2013.

C. Comin, R. Posenato, R. Rizzi, A Tractable Generalization of Simple Temporal Networks and its Relation to Mean Payoff Games, in: 21th Internat. Symp. on Temporal Representation and Reasoning (TIME 2014), Verona, Italy, doi:\bibinfo{doi}{10.1109/TIME.2014.19}, 2014.

C. Comin, R. Posenato, R. Rizzi, Hyper Temporal Networks, Constraints (2016) 1–39ISSN 1572-9354, doi:\bibinfo{doi}{10.1007/s10601-016-9243-0}.

C. Comin, R. Rizzi, Dynamic Consistency of Conditional Simple Temporal Networks via Mean Payoff Games: a Singly-Exponential Time DC-Checking, in: 22nd International Symposium on Temporal Representation and Reasoning (TIME 2015), IEEE CPS, 19–28, doi:\bibinfo{doi}{10.1109/TIME.2015.18}, 2015.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 2001.

R. Dechter, I. Meiri, J. Pearl, Temporal Constraint Networks, Artificial Intelligence 49 (1–3) (1991) 61–95, doi:\bibinfo{doi}{10.1016/0004-3702(91)90006-6}.

J. Eder, E. Panagos, M. Rabinovich, Time Constraints in Workflow Systems, in: M. Jarke, A. Oberweis (Eds.), Advanced Inform. Systems Eng., vol. 1626 of *LNCS*, Springer Berlin Heidelberg, ISBN 978-3-540-66157-3, 286–300, doi:\bibinfo{doi}{10.1007/3-540-48738-7_22}, 1999.

A. Ehrenfeucht, J. Mycielski, Positional strategies for mean payoff games, Int. Journal of Game Theory 8 (2) (1979) 109–113, doi:\bibinfo{doi}{10.1007/BF01768705}.

L. R. Ford, D. R. Fulkerson, Flows in networks, Princeton University Press Princeton, N.J, 1962.

E. Grädel, W. Thomas, T. Wilke (Eds.), Automata Logics, and Infinite Games: A Guide to Current Research, Springer-Verlag New York, Inc., New York, NY, USA, ISBN 3-540-00388-6, 2002.

L. Hunsberger, R. Posenato, Checking the Dynamic Consistency of Conditional Temporal Networks with Bounded Reaction Times, in: Proc. of the Twenty-Sixth Internat. Conf. on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016., 175–183, 2016.

L. Hunsberger, R. Posenato, C. Combi, The Dynamic Controllability of Conditional STNs with Uncertainty, in: Proc. of the Plan. and Plan Exec. for Real-World Syst.: Princip. and Pract. (PlanEx), ICAPS-2012, Atibaia, Sao Paulo, Brazil, 121–128, 2012.

L. Hunsberger, R. Posenato, C. Combi, A Sound-and-Complete Propagation-Based Algorithm for Checking the Dynamic Consistency of Conditional Simple Temporal Networks, in: 22nd International Symposium on Temporal Representation and Reasoning, TIME 2015, Kassel, Germany, September 23-25, 2015, 4–18, doi:\bibinfo{doi}{10.1109/TIME.2015.26}, 2015.

M. D. Moffitt, M. E. Pollack, Applying Local Search to Disjunctive Temporal Problems, in: IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, 242–247, 2005.

A. Oddi, Constraint-Based Strategies for the Disjunctive Temporal Problem: Some New Results, in: Proceedings of the Sixth European Conference on Planning, 2014.

A. Pani, G. Bhattacharjee, Temporal representation and reasoning in artificial intelligence: A review, Mathematical and Computer Modelling 34 (1–2) (2001) 55–80, ISSN 0895-7177, doi:\bibinfo{doi}{10.1016/S0895-7177(01)00049-8}.

D. Smith, J. Frank, A. Jónsson, Bridging the gap between planning and scheduling, Knowledge Engineering Review 15 (1) (2000) 47–83, doi:\bibinfo{doi}{10.1017/S0269888900001089}.

I. Tsamardinos, M. E. Pollack, Efficient solution techniques for disjunctive temporal reasoning problems, Artif. Intell. 151 (1-2) (2003) 43–89, doi:\bibinfo{doi}{10.1016/S0004-3702(03)00113-9}.

I. Tsamardinos, T. Vidal, M. Pollack, CTP: A New Constraint-Based Formalism for Conditional, Temporal Planning, Constraints 8 (4) (2003) 365–388, ISSN 1383-7133, doi:\bibinfo{doi}{10.1023/A:1025894003623}.

U. Zwick, M. Paterson, The Complexity of Mean Payoff Games on Graphs, Theoretical Computer Science 158 (1996) 343–359, doi:\bibinfo{doi}{10.1016/0304-3975(95)00188-3}.