JUSTIFICATION BASED REASONING IN DYNAMIC CONFLICT RESOLUTION*

A PREPRINT

Werner Damm **Martin Fränzle** **Willem Hagemann** **Paul Kröger** **Astrid Rakow**
Department of Computing Science
University of Oldenburg, Germany
{werner.damm, martin.fraenzle, willem.hagemann, paul.kroeger, a.rakow}@uol.de

May 24, 2019

ABSTRACT

We study conflict situations that dynamically arise in traffic scenarios, where different agents try to achieve their set of goals and have to decide on what to do based on their local perception. We distinguish several types of conflicts for this setting. In order to enable modelling of conflict situations and the reasons for conflicts, we present a logical framework that adopts concepts from epistemic and modal logic, justification and temporal logic. Using this framework, we illustrate how conflicts can be identified and how we derive a chain of justifications leading to this conflict. We discuss how conflict resolution can be done when a vehicle has local, incomplete information, vehicle to vehicle communication (V2V) and partially ordered goals.

1 Introduction

As humans are replaced by autonomous systems, such systems must be able to interact with each other and resolve dynamically arising conflicts. Examples of such conflicts arise when a car wants to enter the highway in dense traffic or simply when a car wants to drive faster than the preceding. Such "conflicts" are pervasive in road traffic and although traffic rules define a jurisdictional frame, the decision, e.g., to give way, is not uniquely determined but influenced by a list of prioritised goals of each system and the personal preferences of its user. If it is impossible to achieve all goals simultaneously, autonomous driving systems (ADSs) have to decide "who" will "sacrifice" what goal in order to decide on their maneuvers. Matters get even more complicated when we take into account that the ADS has only partial information. It perceives the world via sensors of limited reach and precision. Moreover, measurements can be contradicting. An ADS might use V2V to retrieve more information about the world, but it inevitably has a confined insight to other traffic participants and its environment. Nevertheless, for the acceptance of ADSs, it is imperative to implement conflict resolution mechanisms that take into account the high dimensionality of decision making. These decisions have to be explained and in case of an incident, the system's decisions have to be accountable.

In this paper we study conflict situations as dynamically occurring in road traffic and develop a formal notion of conflict between two agents. We distinguish several types of conflicts and propose a conflict resolution process where the different kinds of conflicts are resolved in an incremental fashion. This process successively increases the required cooperation and decreases the privacy of the agents, finally negotiating which goals of the two agents have to be sacrificed. We present a logical framework enabling the analysis of conflicts. This framework borrows from epistemic and modal logic in order to accommodate the bookkeeping of evidences used during a decision process. The framework in particular provides a mean to summarise consistent evidences and keep them apart from inconsistent evidences. We hence can, e.g., fuse compatible perceptions into a belief b about the world and fuse another set of compatible perceptions to a belief b' and model decisions that take into account that b might contradict b'. Using the framework we illustrate how conflicts can be explained and algorithmically analysed as required for our conflict resolution process.

*This work is partly supported by the German Research Council (DFG) as part of the PIRE SD-SSCPS project (Science of Design of Societal Scale CPS, grant no. DA 206/11-1, FR 2715/4-1) and the Research Training Group SCARE (System Correctness under Adverse Conditions, grant no. DFG GRK 1765).

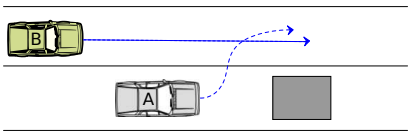

Figure 1: Car A wants to circumvent the obstacle (grey box). Car B is approaching from behind.

Finally we report on a small case study using a prototype implementation (employing the Yices SMT solver [1]) of the conflict resolution algorithm.

Outline. In Sect. 2 we introduce the types of conflict on a running example and develop a formal notion of conflict between two agents. In Sect. 4.1 we outline an algorithm for analysing conflict situations as requested by our resolution protocol and for deriving explanation of the conflict for the resolution. We elaborate on the logical foundations for modelling and analysing conflicts and the logical framework itself in Sect. 3. We sketch our case study on conflict analysis in Sect. 4.. Before drawing the conclusions in Sect. 6, we discuss related work in Sect. 5.

2 Conflict

Already in 1969 in the paper "Violence, Peace and Peace Research" [2] J. Galtung presents his theory of the *Conflict Triangle*, a framework used in the study of peace and conflict. Following this theory a conflict comprises three aspects: opposing *actions*, incompatible *goals*, inconsistent *beliefs* (regarding the reasons of the conflict, knowledge of the conflict parties,...).

We focus on conflicts that arise dynamically between two agents in road traffic. We develop a characterisation of *conflict* as a situation where one agent can accomplish its goals with the help of the other, but both agents cannot accomplish all their goals simultaneously and the agents have to decide what to do based on their local beliefs. In Sect. 2.1 we formalise our notion of conflict. For two agents with complete information, we may characterise a conflict as: Agents A and B are in conflict, if 1. A would accomplish its set of goals Φ_A, if B will do what A requests, while 2. B would accomplish its set of goals Φ_B, if A will do what B requests, and 3. it is impossible to accomplish the set of goals $\Phi_A \cup \Phi_B$. A situation where A and B both compete to consume the same resource is thus an example of a conflict situation. Since we study conflicts from the view-point of an agent's beliefs, we also consider believed conflicts, which can be resolved by sharing information regarding the others observations, strategies or goals. We consider a conflict as *resolved* when a decision of what to do is derived. To resolve a conflict we propose a sequence of steps that require an increasing level of cooperation and decreasing level of privacy – the steps require to reveal information or to constrain acting options. Our resolution process defines the following steps:

C1 Shared situational awareness
C2 Sharing strategies
C3 Sharing goals
C4 Agreeing on which goals to sacrifice and which strategy to follow

Corresponding to C1 to C4, we introduce different kinds of conflicts on a running example – a two lane highway, where one car, A, is heading towards an obstacle at its lane and at the lane to its left a fast car, B, is approaching from behind (cf. Fig. 1). An agent has a prioritised list of goals (like 1. "collision-freedom", 2."changing lane" and 3. "driving fast"). We assume that an agent's goals are achievable.

We discuss related work in Sect. 5. In particular we discuss work regarding the notion of traffic conflict and relate our works with work on the perimeter in game theory [3] and strategy synthesis for levels of cooperation like [4, 5].

An agent A has a set of *actions* act_A and exists within a world. At a time the world has a certain state. The world "evolves" (changes state) as determined by the chosen actions of the agents within the world and events determined by the environment within the world. The agent perceives the world only via a set of *observation predicates*, that are predicates whose valuation is determined by an oberservation of the agent. Without an observation the agent has no (direct) evidence for the valuation of the respective observation predicate.

Example 1. *Let car A want to change lane. It perceives that it is on a two lane highway, the way ahead is free for the next 500 m and B is approaching. Let A perceive B's speed via radar. That is A makes the observation* `car` B `is fast` *justified by the evidence* `radar`. *We annotate this briefly as* `radar:car` B `is fast`. *Further let A derive from lidar data that B is slow* – `lidar:car` B `is slow`.

In this situation we say agent A has *contradicting evidences*. Certain evidences can be combined without contradiction and others not. We assume that an agent organises its evidences in maximal consistent sets (i.e., *justification graphs* introduced in Sect. 3), where each represents a set of possible worlds:

Example 2. *There are possible worlds of A where it is on a two lane highway, the way ahead is free for the next 500 m and B is slowly approaching. Analogously A considers possible worlds where B is fast. The state of the world outside of its sensors' reach is unconstrained.*

Observing the world (for some time), an agent A assesses what it can do to achieve its goals in all possible worlds. That is, A tries to find a *strategy* that guarantees to achieve its goals in all its possible worlds. A strategy determines at each state the action of the agent – the agent decides for an action based on its beliefs formed in the past regarding its possible worlds. If there is one such strategy for A to accomplish its goals Φ_A, then A has a (believed) winning strategy for Φ_A. This strategy might not be winning in the "real" world though, e.g., due to misperceptions.

Example 3. *Let A want to drive slowly and comfortably. A wants to avoid collisions and it assumes that also B wants to avoid collisions. Although A has contradicting evidences on the speed of B and hence believes that it is possible that "B is fast" and also that "B is slow", it can follow the strategy to stay at its lane and wait until B has passed. This strategy is winning in all of A's possible worlds.*

Even when A has no believed winning strategy, it can have a winning strategy for a subset of possible worlds. Additional information on the state of world might resolve the conflict by eliminating possible worlds. We call such conflicts *observation-resolvable* conflicts.

Example 4. *Let A want to change lane to circumvent the obstacle. It is happy to change directly after B but only if B is fast. If B is slow, it prefers to change before B passed. Further let A have contradicting evidences on the speed of B. A considers a conflict with B possible in some world and hence has no believed winning strategy. Now it has to resolve its inconsistent beliefs. Let B tell A, it is fast, and A trust B more than its own sensors, then A might update its beliefs by dismissing all worlds where B is slow. Then "changing after B passed" becomes a believed winning strategy.*

In case of inconsistent evidences, as above, A has to decide how to update its beliefs. The decision how to update its beliefs will be based on the analysis of justifications (cf. Sect. 3) of (contradicting) evidences. The `lidar` contradicts the `radar` and `B` reports on its speed. Facing the contradiction of evidences justified by `lidar` and `radar` A trusts the evidence justified by B.

Let the agents already have exchanged observations and A still have no believed winning strategy. A conflict might be resolved by communicating part of the other agent's (future) strategy:

Example 5. *Let A want to change lane. It prefers to change directly after B, if B passes A fast. Otherwise, A wants to change in front of B. Let B so far away that B might decelerate, in which case it might slow down so heavily that A would like to change in front of B even if B currently is fast.*

Let A believe "B is fast". Now A has no believed winning strategy, as B might decelerate. According to (C2), information about parts of the agent's strategies are now communicated. A asks B whether it plans to decelerate. Let B be cooperative and tell A that it will not decelerate. Then A can dismiss all worlds where B slows down and "changing after B passed" becomes a believed winning strategy for A.

Let the two agents have performed steps (C1) and (C2), i.e., they exchanged missing observations and strategy parts, and still A has no winning strategy for all possible worlds.

Example 6. *Let now, in contrast to Ex. 5, B not tell A whether it will decelerate. Then step (C3) is performed. So A asks B to respect A's goals. Since A prefers B to be fast and B agrees to adopt A's goal as its own, A can again dismiss all worlds where B slows down.*

Here the conflict is resolved by communicating goals and the agreement to adopt the other's goals. So an agent's strategy might change in order to support the other agent. We call this kind of conflicts *goal-disclosure-resolvable* conflicts.

The above considered conflicts can be resolved by some kind of information exchange between the two agents, so that the sets of an agent's possible worlds is adapted and in the end all goals Φ_A of A and Φ_B of B are achievable in all remaining possible worlds. The price to pay for conflict resolution is that the agents will have to reveal information. Still there are cases where simply not all goals are (believed to be) achievable. In this case A and B have to negotiate which goals $\Phi_{AB} \subseteq \Phi_A \cup \Phi_B$ shall be accomplished. While some goals may be compatible, other goals are conflicting. We hence consider goal subsets Φ_{AB} of $\Phi_A \cup \Phi_B$ for which a combined winning strategy for A and B exists to achieve G_{AB}. We assume that there is a weight assignment function w that assigns a value to a given goal combination $2^{\Phi_A \cup \Phi_B} \to \mathbb{N}$ based on which decision for a certain goal combination is taken. This weighting of goals reflects the relative value of goals for the individual agents. To obtain such a function is a difficult task, since it is situation-dependent and influenced by moral and jurisdiction, plus the personal preferences of the agents. It is not in the scope of this paper to demonstrate how to define w.

Example 7. *Let A's and B's highest priority goal be collision-freedom, reflected in goals $\varphi_{A,col}$ and $\varphi_{B,col}$. Further let A want to go fast $\varphi_{A,fast}$ and change lane immediately $\varphi_{A,lc}$. Let also B want to go fast $\varphi_{B,fast}$, so that A cannot change immediately. Now in step (C4) A and B negotiate what goals shall be accomplished. In our scenario collision-freedom is valued most, and B's goals get priority over A's, since B is on the fast lane. Hence our resolution is to agree on a strategy accomplishing $\{\varphi_{A,col}, \varphi_{B,col}, \varphi_{B,fast}\}$, which is the set of goals having the highest value among all those for which a combined winning strategy exists.*

Note that additional agents are captured as part of the environment here. At each step an agent can also decide to negotiate with some other agent than *B* in order to resolve its conflict.

2.1 Formal Notions

In the following we introduce basic notions to define a conflict. Conflicts, as introduced above, arise in a wide variety of system models, but we consider in this paper only a propositional setting.

Let $f_1 : X \to Y_1$,..., $f_n : X \to Y_n$, and $f : X \to Y_1 \times \ldots \times Y_n$ be functions. We will write $f = (f_1, \ldots, f_n)$ if and only if $f(x) = (f_1(x), \ldots, f_n(x))$ for all $x \in X$. Note that for any given f as above the decomposition into its components f_i is uniquely determined by the projections of f onto the corresponding codomain.

Each agent *A* has a set of actions $\mathcal{A}_A$. The sets of actions of two agents are disjoint. To formally define a (possible) world model of an agent *A*, let S be a set of states and $\mathcal{V}$ be a set of propositional variables. $\mathcal{V}$ represents the set of belief propositions. A state $s \in S$ of a (possible) world is labelled with a subset $V \subset \mathcal{V}$ that is (assumed to be) true. $\mathcal{V} \setminus V$ is (assumed to be) false.

A (possible) world model M for an agent *A* is a transition system over S with designated initial state and current state, all states are labelled with the belief propositions that hold at that state and transitions labeled with actions $\langle act_A, act_B, act_{Env}\rangle$ with $act_A \in \mathcal{A}_A$ an action of agent *A*, $act_B \in \mathcal{A}_B$ an action of agent *B* and $act_{Env} \in \mathcal{A}_{Env}$ an action of the environment.

The set of actions of an agent includes send and receive actions via which information can be exchanged, the environment guarantees to transmit a send message to the respective receiver. Formally a possible world is $M_A = (S, T, \lambda, \pi, s_*, s_c)$ with

- $T \subseteq S \times S$,
- $\lambda : T \to \mathcal{A}_A \times \mathcal{A}_B \times \mathcal{A}_{Env}$,
- $\pi : S \to 2^{\mathcal{V}}$,
- $s_* \in S$, $s_c \in S$ s.t.
  - $\forall s_a, s_b, s_1, s_2 \in S$: $(s_a, s_1) \in T \wedge (s_b, s_2) \in T \wedge \lambda(s_a, s_1) = \lambda(s_b, s_2) \wedge \pi(s_a) = \pi(s_b) \Rightarrow \pi(s_1) = \pi(s_2)$ (an action has a unique effect on the state propositions)
  - $\forall s \in S$: $(s, s_*) \notin T$ (s_0 is the initial state)
  - $\exists s_1, \ldots, s_{n+1} \in S$:$\forall 1 \leq i \leq n\ (s_i, s_{i+1}) \in T, \wedge\ s_1 = s_* \wedge s_{n+1} = s_c$ (the current state s_c is reachable from the initial state)
    $\wedge\ ((s_i, s') \in T \Rightarrow s' = s_{i+1})$ (M is linear between s_* and s_c)

The part of M between s_* and s_c represents the *history* of the current state. A finite *run* in M is a sequence of states $r = s_1 s_2 \ldots s_{n+1}$ with $\forall 1 \leq i \leq n : (s_i, s_{i+1}) \in T$.

There is one "special" world model that represents the ground truth, i.e., it reflects how the reality evolves. An agent *A* considers several worlds possible at a time. This is, at each state s of the real world, *A* has a set of possible worlds $\mathcal{M}_A(s)$. The real world changes states according to the actions of *A*, *B* and *Env*. The set of possible worlds $\mathcal{M}_A(s)$ changes to $\mathcal{M}_A(s')$ due to the passing of time and due to belief updates triggered by e.g. observations. For the scope of this paper though, we do not consider the actual passing of time, but study the conflict analysis at a single state of the real world from the point of view of an agent. If an agent follows a strategy, it decides for an action based on the history.Since at each state an agent *A* may consider several worlds possible, it may also consider several histories possible. A strategy is hence a function $\delta_A : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A$, that determines an action for *A* based on the set of possible histories. $\mathbb{H} \in 2^{(2^{\mathcal{V}})^*}$ represents a set of histories, where a history $h \in \mathbb{H}$ is given via the sequence of valuations of $\mathcal{V}$ along the path from s_* to s_c. The set of possible histories at state s is the union of histories of possible worlds $M_A \in \mathcal{M}_A(s)$, denoted as $\mathbb{H}(\mathcal{M}_A(s))$. Given two strategies $\delta_1, \delta_2 : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$, we denote as $\delta_1 \oplus \delta_2$ the strategy $\delta : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$ that chooses the actions of *A* according to δ_1 and the actions of *B* according to δ_2. Let $r = s_0 s_1 \ldots s_n$ be a run in M_A and $\gamma = \upsilon_1 \upsilon_2 \ldots \upsilon_n \in (\mathcal{A}_B \times \mathcal{A}_{Env})^n$ be a sequence of actions of agent *B* and *Env*

along r. r follows strategy δ, $r = r(\delta, \gamma)$, if $\lambda(s_{i-1}, s_i) = \delta(\mathbb{H}(\mathcal{M}_A(s_{i-1}))v_i, \forall 0 \leq i \leq n$. We also write $r(\delta, M_A)$ to denote the set of runs of M_A that follow δ.

We use linear-time temporal logic (LTL) to specify goals (cf. Def. 9). For a run *r* and a goal (or a conjunction of goals) φ, we write $r \models \varphi$ if *r* satisfies φ.[2] We say δ is a (believed) winning strategy for φ in M_A, if all runs r of M_A that follow δ also satisfy φ, $\forall r \in r(M_A, \delta) : r \models \varphi$. We say that δ is a *(believed) winning strategy* of *A* for φ at the real world state s if δ is a winning strategy for φ in all possible worlds $M_A \in \mathcal{M}_A(s)$.

An agent *A* has a set of goals Φ and a weight assignment function $w_A : 2^\Phi \to \mathbb{N}$ that assigns values to a given goal combination. We write $r \models \Phi$ as shorthand for $r \models \bigwedge_{\varphi \in \Phi} \varphi$. $r \models \Phi'$ and $w(\Phi') \geq w(\Phi'')$ for all $\Phi'' \subseteq \Phi$ with $r \models \Phi''$. We say $\Phi' \subset \Phi$ is a *believed achievable goal* at real world s if there is a strategy δ, that is winning for the conjunction of all goals $\varphi \in \Phi'$ in all possible worlds $M_A \in \mathcal{M}_A(s)$. We say $\Phi' \subset \Phi$ is a *believed maximal goal* at real world state s if its is a believed achievable goal and for all believed achievable goals $\Phi'' \subset \Phi$ it holds that $w(\Phi') \geq w(\Phi'')$. The empty subgoal is defined to be true ($\top$).

For each world possible $M_A \in \mathcal{M}_A(s)$ agent *A* also has

1. beliefs on the goals of *B*, $\Phi_B(M_A)$, and
2. beliefs on the importance of subgoals of $\Phi_B(M_A)$ to *B*, $w_B(M_A)$, and
3. a set $\mathbb{J}(M_A)$ of justifications for M_A, $\Phi_B(M_A)$ and $w_B(M_A)$.

So at state s of the real world an agent *A* has belief $\mathcal{B}(A, s) = \bigcup_{M_A \in \mathcal{M}_A(s)} (M_A, \Phi_B(M_A), w_B(M_A), \mathbb{J}(M_A))$. The justifications support decision making by keeping track of (source or more generally meta) information. They hence can influence decisions on how to update an agent's knowledge, how to negotiate and what resolutions are acceptable.

Our notion of conflict captures the following concept: Let Φ^{max} is the set of maximal goals that *A* beliefs it can achieve with the help of *B*. But since *B* might choose a strategy to accomplish some of its maximal goals, *A* believes that it is in a conflict with *B*, if it cannot find one winning strategy that fits all possible strategy choices of *B*.

Definition 1 (Believed Possible Conflict)**.** *Let Φ^{max} be the set of believed maximal goals of A at state s for which a strategy $\delta_{A(B)} : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$ exists.*

Agent A believes at state s, it is in a possible conflict with B, if for each winning strategy $(\delta_A, \delta'_B) : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$ for a maximal goal $\Phi_A \in \Phi_A^{max}$,

- *there is a world $M_A \in \mathcal{M}_A(s)$ that A considers possible and a strategy $(\delta'_A, \delta_B) : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$ such that (δ'_A, δ_B) achieves Φ_B that is a believed maximal subgoal of the believed goals of B,*
- *but (δ_A, δ_B) is not winning strategy for $\Phi_A \cup \Phi_B$ in M_A.*

The above notion of conflict captures that *A* analyses the situation within its possible worlds $\mathcal{M}_A(s)$. It assumes that *B* will follow some winning strategy to accomplish its own goals, while *Env* is assumed to behave fully adversarial. Following this line of thought, we think it is an interesting future extension to also allow *A* having beliefs about the beliefs of *B*. This is well supported by the logical framework introduce in Sect. 3.

For example, consider a situation where *A* drives on a highway side by side of *B* and *A* just wants to stay collision-free, *A* does not believe to be in a conflict situation when it believes that *B* also prioritizes collision-freedom, since *B* will not suddenly choose to crash into *A* which would violate its own goal. But in case *B* has no strategy to accomplish collision-freedom (assume a broken car in front of *B*) within M_A, then *A* assume that *B* behaves arbitrarily (achieving its remaining goal $\top$) and *A* believes to be in conflict with *B*.

2.2 Applying the Formal Notion

In this subsection we consider the examples given at the start of this section and relate them to the formal notions intoduced in the previous subsection.

Propositional Characterisation of the World For the sake of a small example, let us consider the following propositional characterisation of a world: For each agent $X \in \{A, B\}$ there is a pair of variables (l_X, p_X) storing its position in the road. Further each agent drives a certain speed s_X abstracted to three different levels, $s_X \in \{0, 1, 2\}$ encoding slow, medium and fast speed levels. We consider only time bounded properties. The evolution along the observed time window is captured via copies of (l_X, p_X, s_X), $(l_{X,t}, p_{X,t}, s_{X,t})$ where $0 \leq t \leq$ *max_obs_time* encodes the observed time points. Each agent X can change lane, encoded by increasing or decreasing l_X, and choose between three

[2] We assume that runs are infinite here. In case of finite runs, we make them infinite by repeating the last state infinitely often.

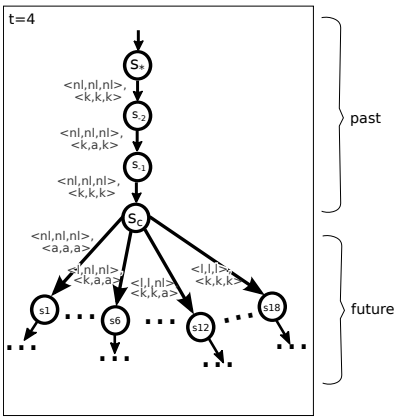

Figure 2: The transition labelling λ is sketched within the figure itself. The state labelling is omitted there. Let us assume that the initial state $\pi(s_*)$ is labelled with $\{l_A = 1,\ l_B = 2,\ l_o = 1,\ p_A = 3,\ p_B = 1,\ p_o = 7,\ s_A = medium,\ s_B = medium,\ s_o = slow\}$ describing the situation of Fig. 1. Currently we are at the time $t = 4$. *A*, *B* and the environment (determining the moves of the obstacle) have done three moves: (1) all three stayed at their respective lane and kept their speed, (2) the same but *B* accelerates and (3) same as (1). The state labelling reflects the changes induced by the chosen moves. So the propositions that are true at, e.g., s_{-2} difer from the one of s_* only in terms of the respective positions: $\{l_A = 1, l_B = 2, l_o = 1, p_A = 4, p_B = 4, p_o = 7, s_A = medium, s_B = medium, s_o = slow\}$.

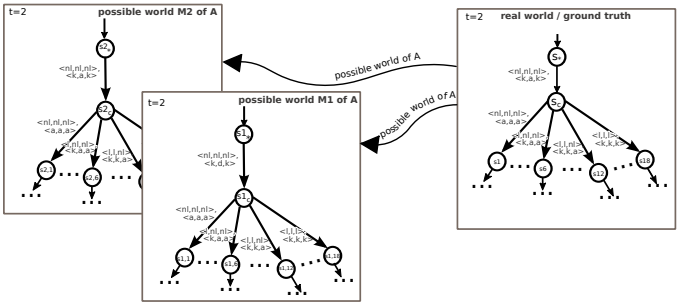

Figure 3: The real world to the right is associated with beliefs of agent *A*. *A* considers at time $t = 2$ two worlds as possible, one, M1, is bisimilar to the real world and a second one, M2, where *B* accelerates as its first move.

different speeds, that is, (i) decelerate inducing a change from fast to medium, or, medium to slow, respectively, or (ii) accelerates from slow to medium, or, from medium to fast, respectively.

A Real World Model In this setting each state of the real world model is labelled with propositions $\{l_X, p_X, s_X \mid X \in \{A, B, o\}\}$ and there are transitions from a state s to a state s' labelled λ(s,s')=($lc_1, lc_2, lc_3, sc_1, sc_2, sc_3$), where $lc_i \in \{\texttt{lane\_change}, \neg\texttt{lane\_change}\}$ and $sc_i \in \{a, d, k\}$. lc_i encodes whether agent X_i chooses to perform a lane change and sc_i encodes how X_i chooses to change its speed, i.e., to accelerate, decelerate or to keep its speed. The target state is labeled according to effect of the chosen action.

The initial state encodes the start situation (of the tour) and the subgraph from the intial state to the current state captures the observed past. The (real) world model has a branching structure from the current state towards the future into the possible different options of lane changing and choices of speed change. Such a world model describes the past, the current state of the world and possible future evolutions. For each point in time t there is hence such a world model. See Fig. 2 for a sketch of an example.

Possible Worlds Additionally to labelling of states and transitions, the real world is also labeled with beliefs of the agents at that time. The gist is

R the real world model captures the past, presence and the possible futures at a time t.

B At time t an agent within world model M considers a set of worlds possible. This belief is justified by e.g. evidences from its sensors.

An example is sketched in Fig. 3, where only *A*'s beliefs are sketched. Note that the state labelling, i.e. the set of true propositions, is not specified in Fig. 3 in order to declutter the figure. Some state labelling is given in Fig. 4. Let for Fig. 3 the initial states be labelled with the same set of propositions. That means the agent only cosiders the real world past as possible.

Let us now consider Ex. 1. Agent *A* has evidence for *B being fast* and it also has evidence for *B being slow*. Fig. 4 illustrates that agent *A* coniders the two (sets of) worlds possible that differ in the valuation of the respective state propositions. Agent *A* believes that a world is possible where B is fast – this is justified by its radar data–, and *A* considers a world possible where B is slow – justified by its lidar.

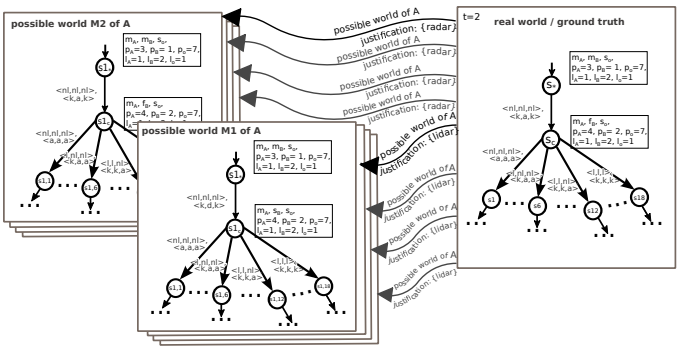

Figure 4: *A* considers currently two sets of worlds possible, one set contains all possible worlds where B is slow in accordance to the lidar and all worlds in the other set satisfy that B is fast in accordance to the the radar.

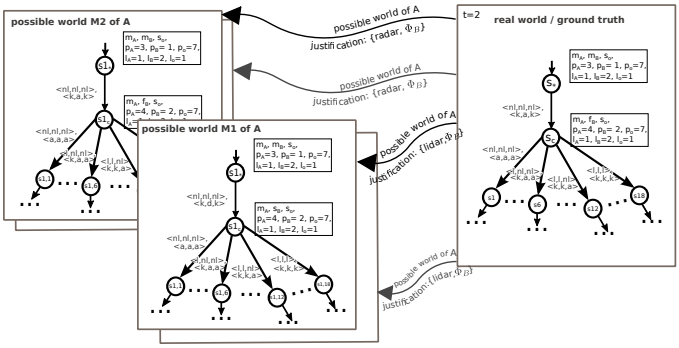

Figure 5: *A* derived additional constraints from *B* goals that constrain its set of possible worlds.

We assume that an agent considers any world M possible that can be justified by some set of consistent evidences. So a possible world M has to satisfy a set of constraints it derived from its observations, i.e., the sensory evidences, and it also has to be compatible to the agent's laws/rules about the world, e.g., physical laws.

The evidences provided by radar and lidar in Ex. 1 imply constraints $s_A = fast$ and $s_B = slow$. These constraints are contradictory and hence there is no possible world that satisfies both constraints. So there cannot be an arrow in Fig. 4 from the real world to a possible world that is labelled with a justification set containing both justifications, $\mathbb{J}$ = $\{$*radar, lidar*$\}$. Nevertheless, the radar and lidar evidences justify that agent *A* believes in alternative worlds (e.g., B is fast, so it is possible that (a) B was driving at medium speed and accelerated or (b) B was fast and kept its speed.)

Strategy and Possible Worlds Let us formalise Ex. 3. *A* considers worlds possible where it has the evidence `radar:car` *B* `is fast` and hence considers worlds possible where *B* is fast, and also worlds where B is not fast, due to its evidence `lidar:car` *B* `is slow`. We already sketched the possible worlds of *A* above.

In order to specify the goals of *A* and the goals *A* believes *B* has, we use the usual LTL operators[3] and in addition $\mathrm{F}_{\leq t}\varphi$ to express that within the next t steps φ has to be true and likewise $\mathrm{G}_{leqt}\varphi$ to specify that at all times up to t φ has to hold.

A wants to drive slowly and comfortably, $\varphi_{cf} = \mathrm{G}_{\leq 3}(s_A = medium \vee s_A = slow)$ and avoid collisions $\varphi_{cl} = \mathrm{G}_{\leq 10}(p_A \neq p_B \wedge p_A \neq p_o)$. *A* also assumes that *B* wants to avoid collisions, $\varphi_{B,cl} = \mathrm{G}_{\leq 10}(p_B \neq p_A \wedge p_B \neq p_o)$. The weight assigment to subsets of goals for *A* is $w_A(\{\varphi_{cl}, \varphi_{cf}\}) = 2$, $w_A(\{\varphi_{cl}\}) = 1$, $w_A(\Phi) = 0$ forall all other subsets, which expresses that collsion-freedom is indispensible. Further *A* believes collision-freedom is also indispensibale for *B*. Additionally, *A* derives from Φ_B a constraint that expresses that *B* will not jeodardize collision-freedom and hence it will not drive irrationally into *A*. This constraint further restricts the set of worlds that *A* considers possible (cf. Fig. 5).

In this situation, *A* decides on its next move. It is not aware of the state of real world and decides only based on its current beliefs regarding the possible worlds and associated goals of *B* and goal weights. *A* determines that staying on its current lane and not changning its speed now is a good move since it can stop and wait in any case, i.e., this move is the prefix of a winning strategy in M1 and all other possible worlds, in which *B* is slow, and also in M2 and all other possible worlds that satisfy that *B* is fast (cf. Fig. 4).

Conflicts

[3]"F" denotes the finally modal operator, "G" denotes globally, "U" denotes until.

Observation Resolvable Conflict In Ex. 4 *A* has the goals • avoid collisions $\varphi_{cl} = \mathrm{G}_{\leq 10}(p_A \neq p_B \wedge p_A \neq p_o)$ and • change lane $\varphi_{lc} = \mathrm{F}_{\leq 5}$ *change_lane* and • chane lane before *B* has passed, if *B* is slow, $\varphi_{flc} = \neg(\mathrm{G}_{\leq 3} s_B = fast) \Rightarrow p_A \geq p_B \,\mathrm{U}\, \textit{change\_lane} \wedge \mathrm{F}_{\leq 3}\textit{change\_lane}$ and • do not change before *B* has passed, if *B* is fast, $\varphi_{slc} = (\mathrm{G}_{\leq 3} s_B = fast) \Rightarrow \mathrm{G}_{\leq 3}\neg\textit{change\_lane}$. We assume here that *A* has only short term goals and global goals are determined at a higher level.[4] *A* also assumes that *B* wants to avoid collisions. The weight assigment to subsets of goals for *A* is specified as follows $6 = w_A(\{\varphi_{cl}, \varphi_{lc}, \varphi_{flc}, \varphi_{slc}\}) > w_A(\{\varphi_{cl}, \varphi_{lc}, \varphi_{slc}\}) = w_A(\{\varphi_{cl}, \varphi_{lc}, \varphi_{flc}\}) > w_A(\{\varphi_{cl}, \varphi_{lc}\}) > w_A(\{\varphi_{cl}, \varphi_{flc}\}) = w_A(\{\varphi_{cl}, \varphi_{slc}\})$[5] $> w_A(\{\varphi_{cl}, \}) = 1$, $w_A(\Phi) = 0$ forall all other subsets.

Obviously *A* has a winning strategy $\delta_{A(B)} : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$, i.e., if it could determine also *B*'s future moves. In this case it can achieve $\{\varphi_{cl}, \varphi_{lc}, \varphi_{flc}, \varphi_{slc}\}$. If *A* assumes that *B* follows a strategy achieving *B*'s own goals under the assumption that *A* will cooperate (i.e. *B* can rule out that *A* changes lane, forcing *B* to decelerate), then *B* can e.g. make up a winning strategy $\delta_{(A)B} : 2^{(2^{\mathcal{V}})^*} \to \mathcal{A}_A \times \mathcal{A}_B$, where *A* stays at lane 1, *B* at lane 2 and *B* chooses its speed arbitrarily without endangering collision-freedom. *A* does not have a winning strategy for all these strategies of *B*, since *A* cannot follow the same strategy if (i) *B* is fast in its next three steps and if (ii) *B* is not fast in at least one of the next three steps.

If *B* tells *A* how fast it will go in its next three steps. The additional information provided by *B*, lets *A* dismiss all possible worlds that do not satisfy the evidence on *B*'s future behaviour. *A* can determine a appropriate strategy for all (remaining) possible worlds and the conflict situation is hence resolved.

3 Epistemic Logic, Justifications and Justification Graph

In this section we first discuss our variant of a logical foundation of justified beliefs. We provide a complete axiomatisation with respect to the semantics for the logic of justification graphs, which extends the single agent semantics as presented in the former section by adding several atomic accessibility relations representing justified beliefs of various sources. For related work see Sect. 5.

Modal Logic and Epistemic Logic. Modal logic extends the classical logic by modal operators expressing necessity and possibility. The formula $\Box\phi$ is read as "ϕ is necessary" and $\Diamond\phi$ is read as "ϕ is possible". The notion of possibility and necessity are dual to each other, $\Diamond\phi$ can be defined as $\neg\Box\neg\phi$. The weakest modal logic K extends propositional logic by the axiom $\mathbf{K}_\Box$ and the necessitation rule $\mathbf{Nec}_\Box$ as follows

$$\vdash \Box(\phi \to \psi) \to (\Box\phi \to \Box\psi), \qquad (\mathbf{K}_\Box) \qquad\qquad \text{from } \vdash \phi \text{ conclude } \vdash \Box\phi. \qquad (\mathbf{Nec}_\Box)$$

The axiom $\mathbf{K}_\Box$ ensures that whenever $\phi \to \psi$ and ϕ necessarily hold, then also ψ necessarily has to hold. The necessitation rule $\mathbf{Nec}_\Box$ allows to infer the necessity of ϕ from any proof of ϕ. It allows us to push any derivable logical truth into the range of the modal operator $\Box$. This principle is also known as *logical awareness*. The following two axioms are useful to model knowledge and belief:

$$\vdash \Box\phi \to \phi, \qquad (\mathbf{T}_\Box) \qquad\qquad \vdash \Box\phi \to \Diamond\phi. \qquad (\mathbf{D}_\Box)$$

The axiom $\mathbf{T}_\Box$ and $\mathbf{D}_\Box$ relate necessity with the factual world. While the truth axiom $\mathbf{T}_\Box$ characterises *knowledge* as it postulates that everything which is necessary is also factual, $\mathbf{D}_\Box$ characterises *belief* as it postulates the weaker property that everything which is necessary is also possible. Under both axioms $\vdash \Box\bot \to \bot$ holds, i.e. a necessary contradiction yields also a factual contradiction. Often we will use the notion of information when it is not important whether we refer to knowledge or belief.

Multi-modal logics are easily obtained by adding several modal operators with possibly different properties and can be used to express the information of more than one agent. Modal operators can also be used to represent modalities referring to time. An important representative of a temporal extension is linear temporal logic (LTL).

In multi-agent logics the notions of common information and distributed information play an important role. While common knowledge captures the information which is known to every agent e_i, we are mainly interested in information that is distributed within a group of agents $E = \{e_1, \ldots, e_n\}$.The distributed information within a group E contains any piece of information that at least one of the agent $e_1, \ldots, e_n$ has. Consequently, we introduce a set-like notion for groups, where an agent e is identified with the singleton group $\{e\}$ and the expression $\{e_1, \ldots, e_n\}{:}\phi$ is used to

[4] Note that also collision-freedom might be sacrificed in so-called dilemma situations.

[5] Note, that φ_{flc} does not imply φ_{lc}.

denote that ϕ is distributed information within the group E. The distribution of information is axiomatised by

$$\vdash E{:}\phi \to F{:}\phi, \text{ where } E \text{ is a subgroup of } F. \qquad (\mathbf{Dist}_{E,F})$$

Note that groups may not be empty. The *modal logic for distributed information* contains for every group E at least the axiom $\mathbf{K}_E$, the necessitation rule $\mathbf{Nec}_E$, and the axiom $\mathbf{Dist}_{E,F}$ for any group F with $E \subseteq F$.

Justification Logics. Justification logics [6] are an extension of epistemic modal logics where the modal operators of knowledge and belief are unfolded into justification terms. Hence, justification logics allow a complete realisation of Plato's characterisation of knowledge as justified true belief. A typical formula of justification logic has the form $s{:}\phi$, where s is a justification term built from justification constants, and it is read as "ϕ is justified by s". The basic justification logic J_0 results from extending propositional logic by the application axiom and the sum axioms

$$\vdash s{:}(\phi \to \psi) \to (t{:}\phi \to [s \cdot t]{:}\psi), \qquad (\mathbf{Appl}) \qquad\qquad \vdash s{:}\phi \to [s+t]{:}\phi, \quad \vdash s{:}\phi \to [t+s]{:}\phi, \qquad (\mathbf{Sum})$$

where s, t, $[s \cdot t]$, $[s+t]$, and $[t+s]$ are justification terms which are assembled using the operators $+$ and $\cdot$ according to the axioms. Justification logics tie the epistemic tradition together with proof theory. Justification terms are reasonable abstractions for constructions of proofs. If s is a proof of $\phi \to \psi$ and t is a proof of ϕ then the application axiom postulates that there is a common proof, namely $s \cdot t$, for ψ. Moreover, if we have a proof s for ϕ and some proof t then the concatenations of both proofs, $s + t$ and $t + s$, are still proofs for ϕ. Since, in our framework, we were not able to derive any meaningful example using the sum axiom of justification logic, this axiom is omitted in the following discussion.

All instances of classical logical tautologies, like $A \vee \neg A$ and $s{:}A \vee \neg s{:}A$, are provable in justification logics. But in contrast to modal logics, justification logics do not have a necessitation rule. The lack of the necessitation rule allows justification logics to break the principle of logical awareness, as $s{:}(A \vee \neg A)$ is not necessarily provable for an arbitrary justification term s. Certainly, restricting the principle of logical awareness is attractive to provide a realistic model of restricted logical resources. Since we are mainly interested in revealing and resolving conflicts, the principle of logical awareness is indispensable in our approach.

Nevertheless, justification logic can simulate unrestricted logical awareness by adding proper axiom internalisation rules $\vdash e{:}\phi$ for all axioms ϕ and justification constants e. In such systems a weak variant of the necessitation rule of modal logic holds: from $\vdash \phi$ conclude $\vdash t{:}\phi$ for some justification term t. Suppose $\vdash \phi$ holds. Since ϕ is derivable from the axioms only, it has to be a tautology. Hence, the justification term t is exclusively built from those justification constants which were dedicated to the axioms needed to derive the tautology ϕ. Beyond that, t is hardly informative as it does not help to reveal extra-logical causes of a conflict. Hence, we omit the axiom internalisation rule and add the modal axiom $\mathbf{K}_t$ and the modal necessitation rule $\mathbf{Nec}_t$ for any justification term t to obtain a justification logic where each justification term is closed under unrestricted logical awareness.

An important consequence of the proposed system is that $\cdot$ becomes virtually idempotent and commutative.[6] These insights allows us to argue merely about justification groups instead of justification terms. It turns out that a proper reformulation of $\mathbf{Appl}$ with regard to justification groups is equivalent to $\mathbf{Dist}_{E,F}$, finally yielding the same axiomatisation for distributed information and compound justifications.

Belief Atoms, Belief Groups, and Belief Entities. So far, we argued that assembling distributed information and compound justifications follow the same principle. In the following we even provide a unified concept for the building blocks of both notions. A *belief atom* e is the least constituent of external information in our logic. Any piece of information of e can be written as a formula $e{:}\phi$, where ϕ is a logical formula representing the information ϕ, making e a modal operator. A belief atom is not necessarily reliable, i.e. even if $e{:}\phi$ holds this does not mean that ϕ is true. So, we call ϕ the belief of e.

Belief atoms play different roles in our setting. E.g., a belief atom may represent a sensor yielding information about the state of the world, and it may represent certain operational rules as well as a certain goal of the system. The characteristic property of a belief atom is that the information of a belief atom can be accepted or rejected as a whole. Due to its external and indivisible nature, e is the only source of evidence for its information. Hence, the only justification for the beliefs of e is e itself. This is what belief atoms and justifications have in common: either we trust a justification or not.

The information of a system is distributed among its belief atoms. The modal logic for distributed information allows us to consider the information which is distributed over a *belief group*. While belief groups can be built arbitrarily from

[6] For any instance $\vdash s{:}(\phi \to \psi) \to (s{:}\phi \to [s \cdot s]{:}\psi)$ of $\mathbf{Appl}$ there is an instance $\vdash s{:}(\phi \to \psi) \to (s{:}\phi \to s{:}\psi)$ of $\mathbf{K}_s$ in the proposed system. Moreover, it is an easy exercise to show that any instance of $\vdash s{:}(\phi \to \psi) \to (t{:}\phi \to [t \cdot s]{:}\psi)$ is derivable in the proposed system.

belief atoms, we also introduce the concept of *belief entities*. A belief entity is either a belief atom, or a distinguished group of belief entities. Belief entities are dynamically distinguished by a justification graph. In contrast to belief groups, belief entities and belief atoms are not allowed to have inconsistent information. Hence, instead of restricting any logical resources, a justification graph allows us to restrict the awareness of extra-logical evidences.

Justification Graphs. Let $\mathcal{V}$ be a set of propositional variables including action labels and let $\mathcal{E}$ be the set of belief entities. The designated subset $\mathcal{E}_A$ of $\mathcal{E}$ denotes the set of belief atoms.

Definition 2 (Language of Justification Graphs)**.** *A formula ϕ is in the language of justification graphs if and only if ϕ is built according to the following BNF, where $A \in \mathcal{V}$ and $\emptyset \neq E \subseteq \mathcal{E}$:*

$$\phi ::= \bot \mid A \mid (\phi \to \phi) \mid E{:}(\phi) \mid \mathrm{X}(\phi) \mid \mathrm{P}(\phi) \mid (\phi)\mathrm{U}(\phi) \mid (\phi)\mathrm{S}(\phi).$$

Using the descending sequence of operator precedences (:, $\neg$, $\vee$, $\wedge$, $\to$, $\leftrightarrow$), we can define the well-known logical connectives $\neg$, $\vee$, $\wedge$ and $\leftrightarrow$ from $\to$ and $\bot$. Often, we omit brackets if the formula is still uniquely readable. We define $\to$ to be right associative. For singleton sets $\{e\} \subseteq \mathcal{E}$ we also write $e{:}\phi$ instead of $\{e\}{:}\phi$. The language allows the usage of temporal operators for next time (X), previous time (P), until (U), and since (S). Operators like always in the future (G) or always in the past (H) can be defined from the given ones.

Definition 3 (Justification Graph)**.** *A* justification graph *is a directed acyclic graph G whose nodes are belief entities of $\mathcal{E}$. An edge $e \mapsto_G f$ denotes that the belief entity e has the* component *f. The set of all direct components of an entity e is defined as $G(e) := \{f \mid e \mapsto_G f\}$.*

The leaf nodes of a justification graph are populated by belief atoms, i.e. for any belief entity e it holds $e \in \mathcal{E}_A$ if and only if $G(e) = \emptyset$.

Definition 4 (Axioms of a Justification Graph)**.** *Let G be a justification graph. The logic of a justification graph has the following axioms and rules.*

(i) As an extension of propositional logic the rule of modus ponens $\mathbf{MP}$ *has to hold: from $\vdash \phi$ and $\vdash \phi \to \psi$ conclude $\vdash \psi$. Any substitution instance of a propositional tautology ϕ is an axiom.*

(ii) Belief groups are closed under logical consequence and follow the principle of logical awareness. Information is freely distributed along the subgroup-relation. For any belief group E the axiom $\mathbf{K}_E$ and the necessitation rule $\mathbf{Nec}_E$ hold. For groups E and F with $E \subseteq F$ the axiom $\mathbf{Dist}_{E,F}$ holds.

(iii) Belief entities are not allowed to have inconsistent information. Non-atomic belief entities inherit all information of their components. For any belief entity e the axiom $\mathbf{D}_e$ holds. If E is a subgroup of the components of e, then the axiom $\mathbf{Dist}_{E,e}$ holds.

(iv) In order to express temporal relation the logic for the justification graph includes the axioms of Past-LTL (LTL with past operator). A comprehensive list of axioms can be found in [7].

(v) Information of a belief entity $e \in \mathcal{E}$ and time are related. The axiom $(\mathbf{PR}_E)$: $\vdash e{:}\mathrm{P}\phi \leftrightarrow \mathrm{P}e{:}\phi$ ensures that every belief entity e correctly remembers its prior beliefs and establishes a principle which is also known as perfect recall *(e.g. see [8]).*

Definition 5 (Proof)**.** *Let G be a justification graph. A proof (derivation) of ϕ in G is a sequence of formulas $\phi_1, \ldots, \phi_n$ with $\phi_n = \phi$ such that each ϕ_i is either an axiom of the justification graph or ϕ_i is obtained by applying a rule to previous members $\phi_{j_1}, \ldots, \phi_{j_k}$ with $j_1, \ldots, j_k < i$. We will write $\vdash_G \phi$ if and only if such a sequence exists.*

Definition 6 (Proof from a set of formulas)**.** *Let G be a justification graph and Σ be a set of formulas. The relation $\Sigma \vdash_G \phi$ holds if and only if $\vdash_G (\sigma_1 \wedge \cdots \wedge \sigma_k) \to \phi$ for some finite subset $\{\sigma_1, \ldots, \sigma_k\} \subseteq \Sigma$ with $k \geq 0$.*

Definition 7 (Consistency with respect to a justification graph)**.** *Let G be a justification graph.*

(i) A set Σ of formulas is G-inconsistent if and only if $\Sigma \vdash_G \bot$. Otherwise, Σ is G-consistent. A formula ϕ is G-inconsistent if and only if $\{\phi\}$ is G-inconsistent. Otherwise, ϕ is G-consistent.

(ii) A set Σ of formulas is maximally G-consistent if and only if Σ is G-consistent and for all $\phi \notin \Sigma$ the set $\Sigma \cup \{\phi\}$ is G-inconsistent.

Semantics. Let S be the *state space*, that is the set of all possible states of the world. An interpretation π over S is a mapping that maps each state s to a truth assignment over s, i.e. $\pi(s) \subseteq \mathcal{V}$ is the subset of all propositional variables which are true in the state s. A *run* over S is a function r from the natural numbers (the time domain) to S. The set of all runs is denoted by Π.

Definition 8. *Let G be a justification graph. A Kripke structure M for G is a tuple $M = (S, \Pi, \pi, (\mapsto_e)_{e \in \mathcal{E}})$ where*

(i) S is a state space,

(ii) Π *is the set of all runs over* S*,*
(iii) π *is an interpretation over* S*,*
(iv) each $\mapsto_e$ *in* $(\mapsto_e)_{e\in\mathcal{E}}$ *is an individual accessibility relation* $\mapsto_e \subseteq S \times S$ *for a belief entity* e *in* $\mathcal{E}$*.*

Definition 9 (Model for a Justification Graph)**.** *Let* $M = (S, \Pi, \pi, (\mapsto_e)_{e\in\mathcal{E}})$ *be a Kripke structure for the justification graph* G*, where*

(i) $\mapsto_e$ *is a serial relation for any belief entity* $e \in \mathcal{E}$*,*
(ii) $\mapsto_E$ *is defined as* $\mapsto_E = \bigcap_{e\in E} \mapsto_e$ *for any belief group* $E \subseteq \mathcal{E}$*,*
(iii) $\mapsto_e \subseteq \mapsto_E$ *holds for all non-atomic belief entities* $e \in \mathcal{E} \setminus \mathcal{E}_A$ *and any subgroup* $E \subseteq G(e)$*.*

We recursively define the model relation $(M, r(m)) \models_G \phi$ *as follows:*

$$
\begin{array}{lll}
(M, r(m)) \not\models_G \bot & & \\
(M, r(m)) \models_G P & :\Longleftrightarrow & P \in \mathcal{V} \text{ and } P \in \pi(r(m)). \\
(M, r(m)) \models_G \phi \to \psi & :\Longleftrightarrow & (M, r(m)) \models_G \phi \text{ implies } (M, r(m)) \models_G \psi. \\
(M, r(m)) \models_G E{:}\phi & :\Longleftrightarrow & (M, r'(m) \models_G \phi \text{ for all } r' \text{ with } r(m) \mapsto_E r'(m). \\
(M, r(m)) \models_G \mathrm{X}\phi & :\Longleftrightarrow & (M, r(m+1)) \models_G \phi \\
(M, r(m)) \models_G \mathrm{P}\phi & :\Longleftrightarrow & (M, r(m')) \models_G \phi \text{ for some } m' \text{ with } m'+1 = m. \\
(M, r(m)) \models_G \phi \mathrm{U} \psi & :\Longleftrightarrow & (M, r(m')) \models_G \psi \text{ for some } m' \geq m \text{ and} \\
 & & (M, r(m'')) \models_G \phi \text{ for all } m'' \text{ with } m \leq m'' < m'. \\
(M, r(m)) \models_G \phi \mathrm{S} \psi & :\Longleftrightarrow & (M, r(m')) \models_G \psi \text{ for some } 0 \leq m' \leq m \text{ and} \\
 & & (M, r(m'')) \models_G \phi \text{ for all } m'' \text{ with } m' < m'' \leq m.
\end{array}
$$

When $(M, r(m)) \models_G \phi$ *holds, we call* $(M, r(m))$ *a* pointed model *of* ϕ *for* G*. If* $(M, r(0))$ *is a pointed model of* ϕ *for* G*, then we write* $(M, r) \models_G \phi$ *and say that the run* r *satisfies* ϕ*. Finally, we say that* ϕ *is satisfiable for* G*, denoted by* $\models_G \phi$ *if and only if there exists a model* M *and a run* r *such that* $(M, r) \models_G \phi$ *holds.*

Proposition 1 (Soundness and Completeness)**.** *The logic of a justification graph* G *is a sound and complete axiomatisation with respect to the model relation* $\models_G$*. That is, a formula* ϕ *is* G*-consistent if and only if* ϕ *is satisfiable for* G*.*

While the soundness proof is straightforward, a self-contained completeness proof involve lengthy sequences of various model constructions and is far beyond the page limit. However, it is well-known, (e.g. [9]), that K_n^D, the n-agent extension of K with distributive information is a sound and complete axiomatisation with respect to the class of Kripke structures having n arbitrary accessibility relations, where the additional accessibility relations for groups are given as the intersection of the participating agents, analogously to Def. 9.(ii). Also the additional extension KD_n^D with $\mathbf{D}_E$ for any belief group E is sound and complete with respect to Kripke structures having serial accessibility relations, analogously to Def. 9.(i). The axioms of justification graph are between these two systems. Def. 9.(iii) explicitly allows belief entities to have more information than its components. Various completeness proofs for combining LTL and epistemic logics are given e.g. in [8].

Extracting Justifications. Let $\Sigma_A = \{\sigma_1, \ldots, \sigma_n\}$ be a finite set of formulas logically describing the situation which is object of our investigation. Each formula $\sigma_i \in \Sigma_A$ encodes information of belief atoms ($\sigma_i \equiv e_i{:}\phi_i$ with $e_i \in \mathcal{E}_A$), facts ($\sigma_i \equiv \phi_i$ where ϕ_i does not contain any epistemic modal operator), or is an arbitrary Boolean combinations thereof. Further, let G be a justification graph such that Σ_A is G-consistent and e be a non-atomic belief entity of G. For any formula ϕ we may now ask whether ϕ is part of the information of e. Clearly, if there is a proof $\Sigma_A \vdash_G e{:}\phi$, then ϕ is necessarily included in e's information. To extract a justification for $e{:}\phi$ we use that $\Sigma_A \cup \{\neg e{:}\phi\}$ is G-inconsistent and accordingly unsatisfiable for G. If we succeed in extracting a minimal unsatisfiable core $\Sigma' \subseteq \Sigma_A \cup \{\neg e{:}\phi\}$ a minimal inconsistency proof can be recovered, from which finally the used justifications are extracted.

The following proposition allows to use SAT/SMT-solvers for a restricted setting and has been used in our case study.

Proposition 2 (SAT Reduction)**.** *Let* $\Sigma_A = \{\sigma_1, \ldots, \sigma_n\}$ *be a set of formulas such that each element* σ_i *is of the form* $e_i{:}\phi_i$ *with* $e_i \in \mathcal{E}_A$ *and* ϕ_i *does not contain any epistemic modal operators. Further, let* e *be an arbitrary belief entity that does not occur in* Σ_A*. Then* $G = \{e \mapsto_G e_i | e_i \text{ occurs in } \Sigma_A\}$ *is a justification graph for* Σ_A *if and only if* $\Phi = \{\phi_1, \ldots \phi_n\}$ *is satisfiable over the non-epistemic fragment of the logic of justification graphs.*

Proof. The satisfiability relation for the non-epistemic fragment is independent of the accessibility relations $\mapsto_e$, $e \in \mathcal{E}$ and, consequently, also independent of G. In particular, Φ is satisfiable if and only if there exists a model $M' = (S, \Pi, \pi)$ and a run r such that $(M', r) \models \Phi$.

Let $G = \{e \mapsto_G e_i | e_i \text{ occurs in } \Sigma_A\}$ be a graph.

Algorithm 1 Determining winning strategy based on observations, goals, and possible actions.

1: **function** FINDSTRATEGY($\Sigma, \Phi_A^{max}, \Phi_B^{max}, \mathcal{A}_A, \mathcal{A}_B$)
2: $\quad\mathcal{M}_A \leftarrow$ MAXCONSISTENTWORLDS($\Sigma, \mathcal{A}_A, \mathcal{A}_B$) ▷ *construct set of possible worlds*
3: $\quad\Delta_A \leftarrow$ STRATA($\mathcal{A}_A, \mathcal{A}_B, \mathcal{M}_A, \Phi_A^{max}$) ▷ *construct* $\{(\delta_A, \delta'_B) \mid r((\delta_A, \delta'_B), \mathcal{M}_A) \models \Phi_A$ *with* $\Phi_A \in \Phi_A^{max}\}$
4: $\quad\mathcal{C} \leftarrow \emptyset$ ▷ *set of conflict causes*
5: $\quad$**for all** $(\delta_A, \delta'_B) \in \Delta_A$ with $r((\delta_A, \delta'_B), \mathcal{M}_A) \models \Phi_A \in \Phi_A^{max}$ **do**
6: $\quad\quad E \leftarrow$ TESTIFNOTWINNING$((\delta_A, \delta'_B), \mathcal{M}_A, \Phi_A, \Phi_B^{max}, \mathcal{A}_A, \mathcal{A}_B)$ ▷ *cf. Alg. 2*
7: $\quad\quad$**if** $E \neq \emptyset$ **then** ▷ (δ_A, δ'_B) *is not winning for all* $M_A \in \mathcal{M}_A$, *i.e.* $r((\delta_A, \delta'_B), \mathcal{M}_A) \not\models \Phi_A$
8: $\quad\quad\quad\mathcal{C} \leftarrow \mathcal{C} \cup \{E\}$ ▷ *memoise justifications* E
9: $\quad\quad\quad\Delta_A = \Delta_A \setminus \{(\delta_A, \delta'_B)\}$
10: $\quad$**if** $\Delta_A = \emptyset$ **then** ▷ A *is in conflict with* B
11: $\quad\quad$**for** (C) = (C1) $\cdots$ (C4) **do** ▷ *iterate over ordered levels of privacy*
12: $\quad\quad\quad\Sigma', \Phi_A^{max\prime}, \Phi_B^{max\prime} \leftarrow$ FIXCONFLICT$(\mathcal{C}, (\mathrm{C}), \Sigma, \Phi_A^{max}, \Phi_B^{max})$ ▷ *cf. Alg. 3*
13: $\quad\quad\quad$**if** $(\Sigma' \neq \Sigma) \vee (\Phi_A^{max\prime} \neq \Phi_A^{max}) \vee (\Phi_B^{max\prime} \neq \Phi_B^{max})$ **then** ▷ *new information generated*
14: $\quad\quad\quad\quad\Delta_A \leftarrow$ FINDSTRATEGY$(\Sigma', \Phi_A^{max\prime}, \Phi_B^{max\prime}, \mathcal{A}_A, \mathcal{A}_B)$ ▷ *new attempt with new information*
15: $\quad\quad\quad\quad$**if** $\Delta_A \neq \emptyset$ **then** ▷ *new attempt was successful, stop and return*
16: $\quad\quad\quad\quad\quad$**break**
17: $\quad$**return** Δ_A ▷ *select* $(\delta_A, \delta'_B) \in \Delta_A$ *to reach some goal in* Φ_A^{max}

Let us first assume that G is a justification graph for Σ_A. Then according to Def. 9 there exists a Kripke structure $M = (S, \Pi, \pi, (\mapsto_e)_{e\in\mathcal{E}})$ and a run r such that $(M, r) \models_G e_i{:}\phi_i$ for all $e_i{:}\phi_i \in \Sigma_A$. Hence, we have $(M, r') \models_G \phi_i$ for all r' with $r \mapsto_{e_i} r'$. Furthermore, from item (i) and (iii) of Def. 9 we observe that $\mapsto_e$ is not empty and $\mapsto_e \subseteq \mapsto_{e_1} \cap \cdots \cap \mapsto_{e_n}$. Hence, there exists at least one run r' that satisfies all formulas in Φ. Since Φ does not contain epistemic operators, we found a model $M' = (S, \Pi, \pi)$ and a run r' such that $(M', r') \models \Phi$.

For the other direction, let us assume that there exists some model $M' = (S, \Pi, \pi)$ and a run r such that $(M', r) \models \Phi$. We extend M' to a Kripke structure $M = (S, \Pi, \pi, (\mapsto_{e'})_{e'\in\mathcal{E}})$ by setting $r \mapsto_{e'} r'$ for all $e' \in \mathcal{E}$ if and only if $r = r'$ for all $r, r' \in \Pi$. Then $(M, r) \models_G e_i{:}\phi_i$ for all $e_i{:}\phi_i \in \Sigma_A$ since $(M', r') \models \phi_i$ holds for all $r \mapsto_{e_i} r'$ by construction of the accessibility relations. Moreover, since all accessibility relations are equal and reflexive, $\mapsto_e$ is serial. □

4 Case Study: Identifying and Analysing Conflicts

4.1 Algorithmic approach

In the previous sections, we introduced formal notions of conflicts and conflict resolution based on justified beliefs. In this section, we sketch an abstract algorithm for determining winning strategies derived from those foundations. Hereby, the focus is on the identification and resolution of conflicts at levels (C1) to (C4). Note that we do not aim with Alg. 1 for efficiency or optimal solutions but aim to illustrate our concepts.

How to find a believed winning strategy. Alg. 1 is supposed to find a winning strategy for agent A, i.e. a strategy that satisfies a maximal goal Φ_A in all possible worlds $M_A \in \mathcal{M}_A$, in particular in those worlds where agent B also tries to achieve one of its maximal goals Φ_B. The input for the algorithm comprises a set Σ of formulae describing the current belief of A, e.g. its current observations and the history that led to the current state. Throughout this section, we will call Σ the *information base*. In addition, Alg. 1 requires a set of maximal goals Φ_A^{max} of A, a believed set of maximal goals Φ_B^{max} of which B, a set of possible actions $\mathcal{A}_A$ for A, and a set of believed possible actions $\mathcal{A}_B$ for B.

The first step is to construct the set of maximal consistent worlds $\mathcal{M}_A$ agent A believes to be possible based on the current information base and the (believed) possible actions for A and B. This is done by the function MAXCONSISTENTWORLDS as intended in **??**. In a second step, the set Δ_A of all winning strategies (δ_A, δ'_B) that satisfy a goal $\Phi_A \in \Phi_A^{max}$ for all possible worlds $M_A \in \mathcal{M}_A$ is constructed by function STRATA.

The set Δ_A might comprise strategies where agent B does not achieve a goal although this is possible. For each $M_A \in \mathcal{M}_A$, we assume that B tries to achieve a goal $\Phi_B \in \Phi_B^{max}$ if this is possible in M_A. We also assume that agent A has no information about which of the assumed goals B tries to achieve. As already indicated in Sect. 2, we thus have to restrict Δ_A to those joint strategies where A achieves a goal in all possible worlds while for all possible worlds M_A, B can achieve all of its (believed) goals that are achievable in M_A. This is done in lines 5 ff. in Alg. 1. For all winning strategies in Δ_A, function TESTIFNOTWINNING from Alg. 2 is called which iterates over the set of

Algorithm 2 Test if a strategy is winning in all possible worlds.

1: **function** TESTIFNOTWINNING($(\delta_A, \delta'_B), \mathcal{M}_A, \Phi_A, \Phi_B^{max}, \mathcal{A}_A, \mathcal{A}_B$)
2: **for all** $M_A \in \mathcal{M}_A$ **do**
3: $\Delta_B \leftarrow$ STRATB($\mathcal{A}_A, \mathcal{A}_B, M_A, \Phi_B^{max}$) ▷ *construct* $\{(\delta'_A, \delta_B) \mid r((\delta'_A, \delta_B), M_A) \models \Phi_B$ *with* $\Phi_B \in \Phi_B^{max}\}$
4: **for all** $(\delta'_A, \delta_B) \in \Delta_B$ **do**
5: **for all** $\Phi_B \in \Phi_B^{max}$ with $r((\delta'_A, \delta_B), M_A) \models \Phi_B$ **do**
6: **if** $(\delta_A, \delta_B) \not\models \Phi_A \cup \Phi_B$ **then** ▷ (δ_A, δ'_B) *is not winning for all* M *and all* (δ'_A, δ_B)
7: **return** GETJUSTIFICATIONS($(\delta_A, \delta_B) \not\models \Phi_A \cup \Phi_B$)
8: **else**
9: **return** $\emptyset$

Algorithm 3 Try to fix a conflict by resolving contradictions.

1: **function** FIXCONFLICT($\mathcal{C}$, (C), $\Sigma, \Phi_A^{max}, \Phi_B^{max}$)
2: **for** $E \in \mathcal{C}$ **do**
3: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{E\}$
4: $\Sigma, \Phi_A^{max}, \Phi_B^{max} \leftarrow$ RESOLVE($\Sigma, E, \Phi_A^{max}, \Phi_B^{max}$, (C)) ▷ *try resolution according to privacy level (C)*
5: **return** $\Sigma, \Phi_A^{max}, \Phi_B^{max}$

possible worlds and constructs the set Δ_B of winning strategies (δ'_A, δ_B) that achieve a goal $\Phi_B \in \Phi_B^{max}$ in M_A.[7] If then there is a goal Φ_B that is achieved by a strategy (δ'_A, δ_B), but the joint strategy (δ_A, δ_B) is not a winning strategy for the joint goal $\Phi_A \cup \Phi_B$, the function GETJUSTIFICATIONS extracts the set of justifications for this situation which is added to the set of conflict causes $\mathcal{C}$ in Alg. 1. The corresponding strategy (δ_A, δ'_B) is removed from the set of winning strategies Δ_A for A.

If this is done for all strategies in the initial set Δ_A, all strategies that remain in Δ_A are winning strategies for one of A's goals in all possible worlds M_A regardless of the goal that B tries to achieve in M_A. However, if Δ_A became empty, A is in a (believed) conflict with B (cf. **??**). In this case, an attempt is made to fix the conflict in lines 10 ff. in Alg. 1. Function FIXCONFLICT from Alg. 3 is called with the set of conflict causes, the current level of privacy, and the current information base and goals. For each conflict cause, an attempt of resolution is mady by function RESOLVE according to the type of the conflict cause encoded in the set of justifications and the level of privacy. Each such attempt possibly updates the information base Σ and goal sets Φ_A^{max} and Φ_B^{max}.

Line 13 checks if some new information was obtained from the resolution procedure. If not, resolution will be restarted with decreased level of privacy. If new information was obtained, FINDSTRATEGY is called with the updated information. If the result is a non-empty set of strategies, the algorithm terminates by returning them as (believed) winning strategies for A. However, if the result is the empty set, resolution is restarted with decreased level of privacy. If the ordered list of levels of privacy has been traversed completely with Δ_A still being empty, the conflict cannot be resolved and the algorithm terminates by returning the empty set.

Fig. 6 provides an abstract scheme of the relation between the initial information base Σ_I, the set $\mathcal{M}_A$ of possible worlds M_A, winning strategies, resolution, and the information Σ_R required to resolve a conflict. The initial information base forms constraints on the set of possible worlds. Based on $\mathcal{M}_A$, the set of strategies is checked to comprise winning strategies in presence of an agent B that tries to achieve its own goals. If no such winning strategy exists, A believes to be in conflict with B. The resolution procedure is initialised which tries to determine the required information Σ_R. This new information is added to the existing information base and the over-all process is re-started again until either winning strategies are found or Σ_R is empty.

[7] Note that according to Sect. 2.1, we have $\Phi_B = \texttt{true}$ if B cannot achieve any goal. This reflects that A cannot make any assumption about B's behaviour in such a situation. Δ_B will be the set of all possible strategies in M_A.

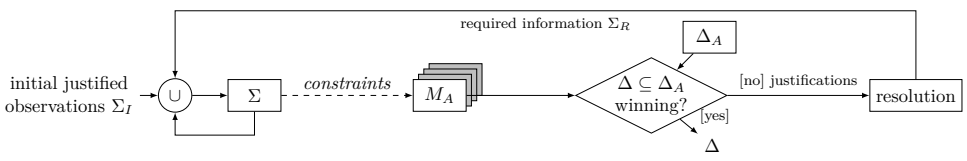

Figure 6: Scheme of relations between information base, possible worlds, strategies and resolution.

Termination. Alg. 1 eventually terminates under the following assumptions. The first assumption is that the set of variables $\mathcal{V}$ and hence the information base Σ is finite. In this case, the construction of maximal consistent possible worlds $\mathcal{M}_A$ terminates since there is a finite number of possible consistent combinations of formulae and the time horizon for the unrolling of a possible world M_A is bounded.

Together with finite sets Φ_A^{max}, Φ_B^{max}, $\mathcal{A}_A$, and $\mathcal{A}_B$, the construction of strategies, i.e. functions StrategyA and StrategyA, terminate since there are only finite numbers of combinations of input histories and output action and there is only a finite number of goals to satisfy. All loops in algorithms 1, 2, and 3 hence iterate over finite sets.

The extraction of justifications terminates since runs are finite and consequently the number of actions involved in the run, too. Furthermore, for each state in a run, there are only a finite number of propositions that apply. Together with a finite number of propositions representing the goal, GetJustifications can simply return the (not necessarily minimal) set of justifications from all these finite many formulae as a naive approach.

Alg. 1 terminates if a non-empty set Δ_A is derived by testing and/or resolution, or if a fixed point regarding Σ, Φ_A^{max}, and Φ_B^{max} is reached. Since all other loops and functions terminate, the only open aspect is the fixed point whose achievement depends on Resolve. We assume that Alg. 1 is executed at a fixed time instance s.t. A's perception of the environment does not change during execution. Thus, Σ contains only a finite number of pieces of information to share while sharing information is a monotonic process never removing any information. Furthermore, we assume that the partial order of goals leads, if necessary, to a monotonic process of goal negotiation which itself can repeated finite many times until no further goals can be sacrified or adopted from B. Thus, if Δ_A remains to be the empty set in line 15, the fixed point will eventually be reached.

Furthermore, we do not consider any kind of race conditions occurring from concurrency, e.g. deadlock situations where A can't serve B's request because it does not know what its strategy will be since A wait's for B's respond, and vice versa.

4.2 Case study

We implemented the algorithm sketched above in a Java program employing Yices [1] to determine contradictions and analyzed variations of a toy example to evaluate and illustrate our approach. More details can be found in [10].

We modelled a system of two agents on a two lane highway. Each agent is represented by its position and its lane. Each agent has a set of actions: it can change lane and drive forward with different speeds. We captured this via a discrete transition relation where agents hop from position to position. The progress of time is encoded via unrolling, that is we have for each point in time a corresponding copy of a variable to hold the value of the respective attribute at that time. Accordingly the transition relation then refers to these copies.

Since we analyse believed conflicts of an agent, we consider several worlds. In other words, we consider several variations of a Yices model. Each variation represents a justification graph summarising the maximal consistent set of evidences and thereby representing a set of worlds which is justified by this set of evidences.

We modify the Yices file by adding additional constraints according to the algorithm Sect. 4.1. For the steps (C1) to (C4) we add constraint predicates, e.g., that encode that information about certain observations have been communicated by say B to A, constraints that specify that B tells A it will decelerate at step 4 and constraints that encode goal combinations.

We employed Yices to determine whether there is conflict. The key observation is: If Yices determines that it holds that $\neg\varphi$ is satisfiable in our system model, then there is the possibility that the goal is not achieved – otherwise each evolution satisfies φ and there is winning strategy for the model.

5 Related work

Studying Traffic Conflicts. According to Tiwari in his 1998 paper [11] studying traffic conflicts in India, one of the earliest studies concerned with *traffic conflicts* is the 1963 paper [12] of Perkins and Harris. It aims to predict crashes in road traffic and to obtain a better insight to causal factors. The term *traffic conflict* is commonly used according to [11] as "an observable situation in which two or more road users approach each other in space and time to such an extent that a collision is imminent if their movements remain unchanged" [13]. In this paper we are interested in a more general and formal notion of conflict. We are not only interested in collisions-avoidance but more generally in situations where traffic participants have to cooperate with each other in order to achieve their goals – which might be collision-freedom. Moreover, we aim to provide a formal framework that allows to explain real world observations as provided by, e.g., the studies of [11, 12].

Tiwari also states in [11] that it is necessary to develop a better understanding of conflicts and conjectures that *illusion of control* [14] and *optimism bias theories* like in [15] might explain fatal crashes. In this paper we develop a formal framework that allows us to analyse conflicts based on beliefs of the involved agents, –although supported by our framework–we here do not compare the real world evolution with the evolution that an agent considers possible. Instead we analyse believed conflicts, that are conflicts which an agent expects to occur based on its beliefs. Such conflicts will have to be identified and analysed by prediction components of the autonomous vehicles architecture, especially in settings where misperception and, hence, wrong beliefs are possible.

In [16] Sameh et al. present their approach to modelling conflict resolution as done by humans in order to generate realistic traffic simulations. The trade-off between anticipation and reactivity for conflict resolution is analysed in [17] in order to determine trajectories for vehicles at an intersection. Both works [16, 17] focus on conflicts leading to accidents. Regarding the suggested resolution approaches, our resolution process suggests cooperation steps with increasing level cooperation. This resolution process is tailored for autonomous vehicles that remain autonomous during the negation process.

Strategies and Games. For strategy synthesis Finkbeiner and Damm [3] determined the right perimeter of a world model. The approach aims to determine the right level of granularity of a world model allowing to find a remorse-free dominant strategy. In order to find a winning (or remorse-free dominant) strategy, the information of some aspects of the world is necessary to make a decision. We accommodated this as an early step in our resolution protocol. Moreover in contrast to [3], we determine information that agent A then want requests from agent B in order to resolve a conflict with B – there may still be no winning (or remorse-free dominant) strategy for all goals of A. In [4] Finkbeiner et. al.presented an approach to synthesise a cooperative strategy among several processes, where the lower prioritised process sacrifices its goals when a process of higher priority achieves its goals. In contrast to [4] we do not enforce a priority of agents but leave it open how a conflict is resolved in case not all their goals are achievable. Our resolution process aims to identify the different kinds of conflict as introduced in Sect. 2 that arise when local information and beliefs are taken into account and which not necessarily imply that actually goals have to be sacrificed.

We characterize our conflict notion in a game theoretic setting by considering the environment of agents A and B as adversarial and compare two scenarios where (i) the agent B is cooperative (angelic) with the scenario where (ii) B is not cooperative and also not antagonistic but reasonable in following a strategy to achieve its own goals. As Brenguier et al. in [5] remark, a fully adversarial environment (including B) is usually a bold abstraction. By assuming in (ii) that B maximses its own goals – we assume that B follows a winning strategy for its maximal accomplishable goals. So we are in a similar mind set than at assume-guaratee [18] and assume-admissible [5] synthesis. Basically we consider the type of strategy (winning/admissable/dominant) as exchangeable, the key aspect of our defnition is that goals are not achievable but can be achieved with the help of the other.

Logics. Justification logic was introduced in [6, 19] as an epistemic logic incorporating knowledge and belief modalities into justification terms and extends classical modal logic by Plato's characterisation of knowledge as justified true belief. However, even this extension might be epistemologically insufficient as Gettier already pointed out in 1963 [20]. In [21] a combination of justification logics and epistemic logic is considered with respect to common knowledge. The knowledge modality K_i of any agent i inherits all information that are justified by some justification term t, i.e. $t{:}\phi \rightarrow K_i\phi$. In such a setting any justified information is part of common knowledge. Moreover, justified common knowledge is obtained by collapsing all justification terms into one modality J and can be regarded as a special constructive sort of common knowledge. While our approach neglects the notion of common information, we use a similar inheritance principle where a belief entity inherits information of its components, cf. Def. 4.(iii). A comparison of the strength of this approach with different notions of common knowledge can be found in [22]. While justification logic and related approaches [23, 24], aim to restrict the principle of logical awareness and the related notion of logical omniscience, we argue in Sec. 3 that the principle of logical awareness as provided by modal logic is indispensable in our approach. A temporal (LTL-based) extension of justification logic has been sketched in [25]. This preliminary work differs from our approach wrt. the axiom systems used for the temporal logic part and the justification / modal logic part, cf. the logic of justification graphs axiomatised in Section 3. Our logic and its axiomatisation incorporates a partial order on the set of beliefs that underlies their prioritization during conflict resolution, which contrasts with the probabilistic extension of justification logic outlined in [26].

6 Conclusion

Considering local and incomplete information, we presented a new notion of conflict that captures situations where an agent believes it has to cooperate with another agent. We proposed steps for conflict resolution with increasing level of cooperation. Key for conflict resolution is the analysis of a conflict, tracing and identifying contradictory

evidences. To this end we presented a formal logical framework unifying justifications with modal logic. Alas, to the authors' best knowledge there are no efficient satisfiability solvers addressing distributed information so far. However, we exemplified the applicability of our framework in a restricted but non-trivial setting. On the one hand, we plan to extend this framework by efficient implementations of adapted satisfiability solvers, on the other hand by integrating richer logics addressing decidable fragments of first order logic, like linear arithmetic, and probabilistic reasoning.

References

[1] B. Dutertre. Yices 2.2. In *Computer Aided Verification*, pages 737–744. Springer, 2014.

[2] J. Galtung. Violence, Peace, and Peace Research. *Journal of Peace Research*, 6(3):167–191, September 1969.

[3] W. Damm and B. Finkbeiner. Does it pay to extend the perimeter of a world model? In *FM 2011: Formal Methods*, pages 12–26. Springer, 2011.

[4] W. Damm, B. Finkbeiner, and A. Rakow. What you really need to know about your neighbor. In *Proc. Fifth Workshop on Synthesis, SYNT@CAV 2016*, volume 229 of *EPTCS*, pages 21–34, 2016.

[5] R Brenguier, J.-F. Raskin, and O. Sankur. Assume-admissible synthesis. *Acta Informatica*, 54(1):41–83, Feb 2017.

[6] S. N. Artemov. Justified common knowledge. *Theor. Comput. Sci.*, 357(1-3):4–22, 2006.

[7] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Workshop on Logic of Programs*, pages 196–218. Springer, 1985.

[8] R. Fagin, J. Y. Halpern, Y. Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 2003.

[9] J. Gerbrandy. Distributed knowledge. In *Twendial 1998: Formal Semantics and Pragmatics of Dialogue*, volume 98, pages 111–124, 1998.

[10] W. Damm, M Fränzle, W. Hagemann, P. Kröger, and A. Rakow. Justification based reasoning for dynamic conflict resolution – extended version. Technical report, SFB/TR 14 AVACS, 2019. to appear in March.

[11] G. Tiwari, D. Mohan, and J. Fazio. Conflict analysis for prediction of fatal crash locations in mixed traffic streams. *Accident Analysis & Prevention*, 30(2):207 – 215, 1998.

[12] J. I. Harris and S. R. Perkins. Traffic conflict characteristics: accident potential at intersections. *Highway Research Board*, 225:35–43, 1967.

[13] F. H. Amundsen and C. Hyden. Proc. of the first workshop on traffic conflicts, oslo, norway, 1977. 1st Workshop on Traffic Conflicts, LTH Lund.

[14] J. E. Langer. The illusion of control. *Journal of Personality and Social Psychology*, 32:311–328, 08 1975.

[15] D. M. DeJoy. The optimism bias and traffic accident risk perception. *Accident Analysis & Prevention*, 21(4):333 – 340, 1989.

[16] S. El hadouaj, A. Drogoul, and S. Espié. How to combine reactivity and anticipation: The case of conflicts resolution in a simulated road traffic. In *Multi-Agent-Based Simulation*, pages 82–96. Springer, 2001.

[17] N. Murgovski, G. R. de Campos, and J. Sjöberg. Convex modeling of conflict resolution at traffic intersections. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4708–4713, Dec 2015.

[18] K. Chatterjee and T. A. Henzinger. Assume-guarantee synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 261–275. Springer, 2007.

[19] S. N. Artemov. The logic of justification. *Rew. Symb. Logic*, 1(4):477–513, 2008.

[20] E. L. Gettier. Is justified true belief knowledge? *Analysis*, 23(6):121–123, 1963.

[21] S. Artemov and E. Nogina. Introducing justification into epistemic logic. *Journal of Logic and Computation*, 15(6):1059–1073, 2005.

[22] E. Antonakos. Justified and common knowledge: Limited conservativity. In *International Symposium on Logical Foundations of Computer Science*, pages 1–11. Springer, 2007.

[23] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial intelligence*, 34(1):39–76, 1987.

[24] S. Artemov and R. Kuznets. Logical omniscience as a computational complexity problem. In *Proc. of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 14–23. ACM, 2009.

[25] S. Bucheli, M. Ghari, and T. Studer. Temporal justification logic. In Proc. of the 9th Workshop on *Methods for Modalities,* January 2017, volume 243 of *EPTCS*, pages 59–74. Open Publishing Association, 2017.

[26] I. Kokkinis, Z. Ognjanovic, and T. Studer. Probabilistic justification logic. In *Logical Foundations of Computer Science - International Symposium, LFCS 2016. Proc.*, volume 9537 of *LNCS*, pages 174–186. Springer, 2016.