Dynamic Conflict Resolution Using Justification Based Reasoning[*]

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

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 manoeuvres. 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

[*] 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).

G. Caltais and J. Krivine (eds.): CREST 2019
EPTCS 308, 2019, pp. 47–65, doi:10.4204/EPTCS.308.4

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. Finally we report on a small case study using a prototype implementation (employing the Yices SMT solver [14]) of the conflict resolution algorithm. 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 [10] and strategy synthesis for levels of cooperation like [11, 7].

Outline In Sect. 2 we introduce the types of conflict on a running example and develop a formal notion of conflict between two agents. 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 and outline in Sect. 4.2 an algorithm for analysing conflict situations as requested by our resolution protocol and for deriving explanation of the conflict for the resolution. 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" [18] 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. 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:
(C_1) Shared situational awareness
(C_2) Sharing strategies
(C_3) Sharing goals
(C_4) Agreeing on which goals to sacrifice and which strategy to follow
Corresponding to (C_1) to (C_4), 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.

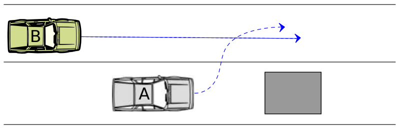

Figure 1: Car A wants to circumvent the obstacle (grey box). Car B is approaching from behind.

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 observation 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* of 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 believed past. 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 (C_1) and (C_2), 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* (C_3) *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} for which a combined winning strategy for A and B exists. 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. Such a function will have to reflect, e.g., moral, ethics and jurisdiction.

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 $f_1 : X \to Y_1, \ldots, f_n : X \to Y_n$, 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. A believes at a state $s \in S$ that a subset V of $\mathcal{V}$ is true and $\mathcal{V} \setminus V$ is 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 $\langle act_A, act_B, act_{Env} \rangle$ where $act_A \in \mathcal{A}_A$ is an action of A, $act_B \in \mathcal{A}_B$ an action of 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 $T \subseteq S \times S$, $\lambda : T \to \mathcal{A}_A \times \mathcal{A}_B \times \mathcal{A}_{Env}$, $\pi : S \to 2^{\mathcal{V}}$ and $s_*, s_c \in S$ representing the initial state and the current state respectively. We use M_A to encode the current believes on the present, past and about the possible futures. If an agent, let us say A wlog, follows a strategy, it decides for an action based on its believed past, i.e., a strategy for A is a function $\delta_A : (2^{\mathcal{V}})^* \to \mathcal{A}_A$. Given strategies $\delta_A : (2^{\mathcal{V}})^* \to \mathcal{A}_A$ for A, $\delta_B : (2^{\mathcal{V}})^* \to \mathcal{A}_B$ for B, $\delta = (\delta_A, \delta_B) : (2^{\mathcal{V}})^* \to \mathcal{A}_A \times \mathcal{A}_B$ is a common strategy of A and B, which chooses the actions of A according to δ_A and the actions of B according to δ_B.

A (believed) *run* $r = (s_0, s_1, \ldots)$ is an infinite sequence of states starting with the initial state $s_0 = s_*$ and $(s_i, s_{i+1}) \in T$ for all $i \in \mathbb{N}$. A run $r = (s_0, s_1, \ldots)$ *results from a strategy* δ_A in A's world M_A, denoted as $r \in r(\delta_A, M_A)$, if and only if $\lambda(s_i, s_{i+1}) = \delta_A(\pi(s_0), \pi(s_1), \ldots \pi(s_i))$ for all $i \in \mathbb{N}$. Given a set of possible worlds $\mathcal{M}(A)$ for A, we use $r(\delta_A, \mathcal{M}(A))$ to denote the set of runs that result from δ_A in a $M_A \in \mathcal{M}(A)$, $r(\delta_A, \mathcal{M}(A)) = \bigcup_{M_A \in \mathcal{M}(A)} r(\delta_A, M_A)$.

We use linear-time temporal logic (LTL) to specify goals. For a run r and a goal (or a conjunction of goals) φ we write $r \models \varphi$, if the valuation of propositions along r satisfies φ[1]. We say δ is a (believed) winning strategy for φ in M_A, if for all $r \in r(\delta, M_A)$ it holds that $r \models \varphi$. 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 $r \models \bigwedge_{\varphi \in \Phi} \varphi$. We say subgoal $\Phi' \subseteq \Phi$ is maximal for r if $r \models \Phi'$ and $w(\Phi') \geq w(\Phi'')$ for all $\Phi'' \subseteq \Phi$ with $r \models \Phi''$. `true` is the empty subgoal. We say $\Phi' \subseteq \Phi$ is maximal for a set R of runs if for all $r \in R$ $r \models \Phi'$ and for all $\Phi'' \subseteq \Phi$ with $r \models \Phi''$ for all $r \in R$, $w(\Phi') \geq w(\Phi'')$.

There is one "special" world model that represents the ground truth, i.e., it reflects how the reality evolves. We refer the interested reader to [12] for a more elaborate presentation of our concept of reality and associated beliefs. Agent A considers several worlds possible at a time. At each state s of the real world A has a set of possible worlds $\mathcal{M}_A(s)$ and for each world $M_A \in \mathcal{M}(A)$ a believed current state and beliefs on the goals of B, $\Phi_B(M_A)$ and the goal weight assignment function of B, w_B. A possible world is labeled with the set of evidences that justifies that the world is regarded as possible. The real world changes states according to the actions of A, B and E. The set of possible worlds $\mathcal{M}_A(s)$ changes to $\mathcal{M}_A(s')$ due to the believed 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.

We say that A has a (believed) winning strategy δ_A for Φ_A at the real world state s_c if δ_A is a winning strategy for Φ_A in all possible worlds $M_A \in \mathcal{M}_A(s_c)$.

Definition 1 (Believed Possible Conflict)**.** *Let Φ_A^{max} be the set of maximal subgoals of A at state s for which a believed winning strategy $(\delta_A, \delta_B') : (2^{\mathcal{V}_A})^* \to \mathcal{A}_A \times \mathcal{A}_B$ in $\mathcal{M}_A$ exists.*

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

[1] cf. Def. 9

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

In Def. 1 Φ_A^{max} is the set of maximal subgoals that A can achieve in all possible worlds with the help of B. A believes that B might decide for a strategy to accomplish some of its maximal subgoals and B takes this decision wrt A's possible worlds. Note that A assuming the goals of B being `true` means that A has to deal with arbitrary behaviour of B. Also note that B always has a winning strategy in every M since Φ_B is maximal wrt M. If A cannot find one winning strategy that fits all possible choices of B then A believes that it is in a conflict with B. Note that A analyses the conflict within its possible worlds $\mathcal{M}_A(s)$ and in particular it beliefs that B believes that in one of its possible wolds. That A has got beliefs about (deviations of) the beliefs of B is an interesting future extension.

3 Epistemic Logic, Justifications and Justification Graph

Conflict analysis demands to know who believes to be in conflict with whom and what pieces of information made him belief that he is in conflict. To this end we introduce the *logic of justification graphs* that allows to keep track of external information and extends purely propositional formulae by so called *belief atoms* (cf. p. 54), which are used to label the sources of information. In Sect. 2 we already used such formulae, e.g., "`radar:car` *B* `is fast`". Our logic provides several atomic accessibility relations representing justified beliefs of various sources, as required for our examples of Sect. 2. It provides justification graphs as a mean to identify *belief entities* which compose different justifications to consistent information even when the information base contains contradicting information of different sources, as required for analysing conflict situations.

First, this section provides a short overview on epistemic modal logics and multi-modal extensions thereof. Such logics use modal operators to expressing knowledge and belief stemming from different sources. Often we will refer to this knowledge and belief as *information*, especially when focusing on the sources or of the information. Thereupon the basic principles of justifications logics are shortly reviewed. Justification logics are widely seen as interesting variants to epistemic logics as they allow to trace back intra-logical and external justifications of derived information. In the following discussion it turns out that tracing back external justifications follows the same principles as the distribution of information over different sources.

Consequently, the concept of information source and external justification are then unified in our variant of an epistemic modal logic. This logic of justification graphs extends the modal logic by a justification graph. The nodes of a justification graph are called belief entities and represent groups of consistent information. The leaf nodes of a justification graphs are called belief atoms, which are information source and external justifications at the same time, as they are the least constituents of external information. We provide a complete axiomatisation with respect to the semantics of the logic of justification graphs.

3.1 Justification Graphs

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 notions 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 and, hence, pushes any derivable logical truth into the range of the modal operator $\Box$. This principle is also known as *logical awareness*. Various modalities like belief or knowledge can be described by adding additional axioms encoding the characteristic properties of the respective modal operator. 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.

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. E.g., the formula $e_i{:}\phi$ expresses that the piece of information ϕ belongs to the modality e_i. Modal operators can also be used to represent modalities referring to time. E.g., in the formula $\mathrm{X}\phi$ the temporal modality X expresses that ϕ will hold in the next time step. 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, …, 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 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 [5] are variants 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 from justification constants 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. In our framework we were not able to derive any meaningful example using the sum axiom of justification logic. Therefore this axiom is omitted in the following discussion.

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: for any derivation $\vdash \phi$ there exists a justification term t such that $\vdash t{:}\phi$ holds. Since ϕ was derived using axioms and rules only, also the justification term t is exclusively built from justification constants dedicated to the involved axioms. Beyond that, t is hardly informative as it does not help to reveal *external* 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.[2] These insights allows us to argue merely about justification groups instead of justification terms. It turns out that a proper reformulation of **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. To each e we assign the modal operator $e{:}$. Hence, for any formula ϕ also $e{:}\phi$ is a formula saying "e has information ϕ". Belief atoms play different roles in our setting. A belief atom may represent a sensor collecting information about the state of the world, or 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 has to be accepted or rejected as a whole. Due to its external and indivisible nature, e is the only source of evidence for its information. The only justification for information of e is e itself. Consequently, $e{:}\phi$ can also be read as "e is the justification for ϕ". 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 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

[2] For any instance $\vdash s{:}(\phi \to \psi) \to (s{:}\phi \to [s \cdot s]{:}\psi)$ of **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.

distinguished by a justification graph. In contrast to belief groups, belief entities and belief atoms are not allowed to have inconsistent information. Hence a justification graph allows us to restrict the awareness of extra-logical evidences – so we can distinctively integrate logical resources that have to be consistent.

Justification Graphs Let $\mathcal{V}$ be a set of propositional variables 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 **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 [24].

(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 [17]).*

Definition 5 (Proof)**.** *Let G be a justification graph. A proof (derivation) of ϕ in G is a sequence of formulae $\phi_1, \ldots, \phi_n$ 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 formulae)**.** *Let G be a justification graph and Σ be a set of formulae. 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 formulae 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 formulae 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. In Sect. 2 we introduced world models and runs on world models. There a world model captured the evolution of a states in time.

In this section we focus an the epistemic notions of knowledge and belief and therefore our main concern is the accessibility relation of information. We hence presume that the set of runs of a possible world of Sect. 2 is given, that then defines the evolution in time. Formally 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 $\mathcal{R}$.

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

(i) S is a state space,

(ii) $\mathcal{R}$ 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 $M = (S, \mathcal{R}, \pi, (\mapsto_e)_{e\in\mathcal{E}})$ 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 $(M, r(t)) \models_G \phi$ as follows:

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

When $(M, r(t)) \models_G \phi$ holds, we call $(M, r(t))$ 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., [19]), 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 [17].

Extracting Justifications Let $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ be a finite set of formulae logically describing the situation which is object of our investigation. Each formula $\sigma_i \in \Sigma$ 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 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. If there is a proof $\Sigma \vdash_G e{:}\phi$, then ϕ is included in e's information. To extract a justification for $e{:}\phi$ we use that $\Sigma \cup \{\neg e{:}\phi\}$ is G-inconsistent and accordingly unsatisfiable for G. If we succeed in extracting a minimal unsatisfiable core $\Sigma' \subseteq \Sigma \cup \{\neg e{:}\phi\}$ 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 $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ be a set of formulae 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. Then $G = \{e \mapsto_G e_i | e_i \text{ occurs in } \Sigma\}$ is a justification graph for Σ if and only if $\Phi = \{\phi_1, \dots \phi_n\}$ is satisfiable over the non-epistemic fragment of the logic of justification graphs.*

In order to proof the proposition one shows that any model of Φ in the non-epistemic fragment can be extended to a model of Σ for the given G by adding trivial accessibility relations. On the other hand, for any model M and run r with $(M,r) \models_G \Sigma$ there exists a run r' which is accessible from $\mapsto_e$ such that $(M,r') \models_G \Phi$. Since Φ does not contain any epistemic modal operators, dropping the accessibility relations from M still yields a model of Φ. A more detailed version of this proof can be found in [12].

4 Identifying and Analysing Conflicts

In this section we first present an abstract algorithm for the conflict resolution of Sect. 2 that starts at level (C_1) and proceeds resolution stepwise up to level (C_4). We then sketch our small case study where we applied an implementation of the abstract algorithm.

4.1 Analysing Conflicts

For the analysis of conflicts we employ SMT solvers. Prop. 2 reduces the satisfiability of a justification graph to a SAT problem. To employ SMT solving for conflict analysis, we encode the (real and possible) worlds of Sect. 2 via logic formulae as introduced in Sect. 3. Each state s_i is represented as a conjunction of literals, $s_i \equiv \bigwedge v \wedge \bigwedge \neg v'$. Introducing a dedicated propositional variable v_t for each $v \in \mathcal{V}$ and time step

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

```
function FindStrategy($\Sigma, \Phi_A^{max}, \Phi_B^{max}, \mathcal{A}_A, \mathcal{A}_B$)
    $\Sigma_{\mathcal{M}} \leftarrow$ PossibleWorlds($\Sigma, \mathcal{A}_A, \mathcal{A}_B$)    ▷ construct set of possible worlds
    $\Delta_A \leftarrow$ StratA($\mathcal{A}_A, \mathcal{A}_B, \Sigma_{\mathcal{M}}, \Phi_A^{max}$)    ▷ construct $\{(\delta_A, \delta'_B) \mid r((\delta_A, \delta'_B), \Sigma_{\mathcal{M}}) \models \Phi_A$ with $\Phi_A \in \Phi_A^{max}\}$
    $\mathcal{C} \leftarrow \emptyset$    ▷ set of conflict causes
    for all $(\delta_A, \delta'_B) \in \Delta_A$ with $r((\delta_A, \delta'_B), \Sigma_{\mathcal{M}}) \models \Phi_A \in \Phi_A^{max}$ do
        $E \leftarrow$ TestIfNotWinning($(\delta_A, \delta'_B), \Sigma_{\mathcal{M}}, \Phi_A, \Phi_B^{max}, \mathcal{A}_A, \mathcal{A}_B$)    ▷ cf. Alg. 2
        if $E \neq \emptyset$ then    ▷ $(\delta_A, \delta'_B)$ is not winning for all $\Sigma_M \in \Sigma_{\mathcal{M}}$, i.e. $r((\delta_A, \delta'_B), \Sigma_{\mathcal{M}}) \not\models \Phi_A$
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{E\}$    ▷ memorize justifications E
            $\Delta_A = \Delta_A \setminus \{(\delta_A, \delta'_B)\}$
    if $\Delta_A = \emptyset$ then    ▷ A is in conflict with B
        for $i \in [1, 2, 3, 4]$ do    ▷ traverse resolution levels
            $\Sigma', \Phi_A^{max\prime}, \Phi_B^{max\prime} \leftarrow$ FixConflict($\mathcal{C}, (C_i), \Sigma, \Phi_A^{max}, \Phi_B^{max}$)    ▷ cf. Alg. 3
            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
                $\Delta_A \leftarrow$ FindStrategy($\Sigma', \Phi_A^{max\prime}, \Phi_B^{max\prime}, \mathcal{A}_A, \mathcal{A}_B$)    ▷ new attempt with new information
                if $\Delta_A \neq \emptyset$ then    ▷ new attempt was successful, stop and return
                    break
    return $\Delta_A$    ▷ select $(\delta_A, \delta'_B) \in \Delta_A$ to reach some goal in $\Phi_A^{max}$
```

t allows us to obtain a formula describing a finite run on M. A predicate of the form $\bigwedge_{(s,s')\in T}(s_t \rightarrow s'_{t+1})$ encodes the transition relation T. The effect of performing an action a_t at state s is captured by a formula of the form $a_t \rightarrow (s_t \rightarrow s'_{t+1})$. Using this we can encode a strategy δ in a formula ψ_δ such that its valuations represent runs of M according to δ. All runs according to δ achieve goals Φ if and only if $\psi_\delta \wedge \neg\Phi$ is unsatisfiable. These logical encodings are the main ingredients for using a SAT solver for our conflict analysis. Since there are only finitely many possible strategies, we examine for each strategy which goals can be (maximally) achieved in a world M or in a set of worlds $\mathcal{M}$. Likewise we check whether A has a winning strategy that is compatible with the strategies A believes B might choose.

Since we iterate over all possible worlds for our conflict analysis, we are interested in summarising possible worlds. We are usually not interested in all v_t – e.g. the speed of B may at times t be irrelevant. We are hence free to ignore differences in v_t in different possible worlds and are even free to consider all valuations of v_t, even if A does not consider them possible. This insight leads us to a symbolic representation of the possible worlds, collecting the relevant constraints. Now the justification graph groups the constraints that are relevant, with other words, $e{:}\phi$ and $e'{:}\neg\phi$ will not be components of the same justification graph if the valuation of ϕ is relevant. In the following we hence consider the maximal consistent set of possible worlds, meaning encodings of possible worlds that are uncontradictory wrt. the relevant propositions, which are specified via the justification graph.

4.2 Algorithmic approach

In this section, we sketch an abstract algorithm for the conflict resolution at levels (C_1) to (C_4) as in Sect. 2. Note that we do not aim with Alg. 1 for efficiency or optimal solutions but aim to illustrate how satisfiability checks can be employed to analyse our conflicts.

The following algorithms describe how we deal with logic formulae encoding sets of possible worlds, sets of runs on them, etc. to analyse conflicts (cf. Def. 1, p. 51) via SMT solving. We use Σ_M to refer to a formula that encodes a maximal consistent set of possible worlds (cf. Sect. 4.1), i.e., that corresponds to a

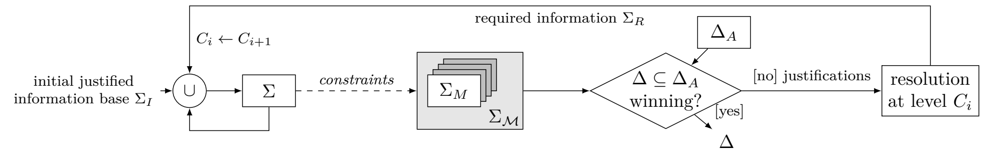

Figure 2: Abstract resolution process with information base, possible worlds and strategies.

justification graph. We use $\Sigma_{\mathcal{M}}$ to refer to a set of formulas $\Sigma_M \in \Sigma_{\mathcal{M}}$ that encode the set of possible worlds $\mathcal{M}$ structured into sets of possible worlds via justification graphs. We use $\mathcal{M}$ and $\Sigma_{\mathcal{M}}$ synonymously. Also we often do not distinguish between Σ_M and M – neglecting that Σ_M represents a set of worlds that are like M wrt to the relevant constraints.

Fig. 2 provides an overview of the relation between the initial information base Σ_I of agent A, its set $\Sigma_{\mathcal{M}}$ of possible worlds Σ_M, winning strategies, resolution, and stepwise update of the information Σ_R during our conflict resolution process. The initial information base defines the set of possible worlds $\mathcal{M}$, which is organised in sets of maximal consistent worlds Σ_M. Based on $\Sigma_{\mathcal{M}}$, A's set of strategies Δ_A is checked whether it comprises a 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. At each level (C_i) the resolution procedure tries to determine information Σ_R of level (C_i) to resolve the conflict. If the possible worlds are enriched by this information, the considered conflict vanishes. The 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.

How to find a believed winning strategy Alg. 1 finds a winning strategy of agent A for a goal Φ_A in $\Sigma_{\mathcal{M}}$ tolerating that B follows an arbitrary winning strategy in $\Sigma_M \in \Sigma_{\mathcal{M}}$ for its goals, i.e. it finds a strategy that satisfies Φ_A in all possible worlds $\Sigma_{\mathcal{M}}$ where Φ_A is maximal for $\Sigma_{\mathcal{M}}$ and in each possible world $\Sigma_M \in \Sigma_{\mathcal{M}}$ agent B may also follow a winning strategy for one of its maximal goals Φ_B. If such a strategy cannot be found, A believes to be in conflict with B (cf. Def. 1).

Input for the algorithm is (i) a set Σ of formulae describing the current belief of A, e.g. its current observations and its history of beliefs –we call it the *information base* in the sequel–, (ii) a set of goals Φ_A^{max} of A that is maximal in $\mathcal{M}$, (iii) a set of believed goals Φ_B^{max} of B that is maximal for a $\Sigma_M \in \Sigma_{\mathcal{M}}$, (iv) a set of possible actions $\mathcal{A}_A$ for A and (v) a set of believed possible actions $\mathcal{A}_B$ for B.

First (L. 2 of Alg. 1) is to construct sets of maximal consistent sets of possible worlds that together represent $\mathcal{M}$. In L. 3 the set Δ_A is determined, which is the set of strategies accomplishing a maximal goal combination for A assuming B agrees to help, i.e., all winning strategies (δ_A, δ'_B) that satisfy $\Phi_A \in \Phi_A^{max}$ in all possible worlds $\Sigma_M \in \Sigma_{\mathcal{M}}$, where Φ_A is maximal for $\mathcal{M}$.

In lines 5 ff. we examine whether one of A's strategies (where B is willing to help) works even when B follows its strategy to achieve one of its maximal goals $\Phi_B \in \Phi_B^{max}$ in Σ_M.

To this end TESTIFNOTWINNING is called for all of A's winning strategies $(\delta_A, \delta'_B) \in \Delta_A$ (L. 6). The function TESTIFNOTWINNING performs this test iteratively for one maximal consistent set of worlds Σ_M (Alg. 2 L. 2). Let Δ_B be the set of joint strategies achieving a goal $\Phi_B \in \Phi_B^{max}$ that is maximal in Σ_M. We check the compatibility of A's strategy (δ_A, δ'_B) to every $(\delta'_A, \delta_B) \in \Delta_B$ (Alg. 2 L. 3). A strategy of A (δ_A, δ'_B) is compatible to all of B's strategies (δ'_A, δ_B) if all joint strategies (δ_A, δ_B) achieve the maximal goals for A and B (Alg. 2 L. 6).[3] If the joint strategy (δ_A, δ_B) is not a winning strategy for the joint goal

[3] 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.

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

```
1: function $\textsc{TestIfNotWinning}((\delta_A,\delta'_B),\Sigma_{\mathcal{M}},\Phi_A,\Phi_B^{max},\mathcal{A}_A,\mathcal{A}_B)$
2:     for all $\Sigma_M \in \Sigma_{\mathcal{M}}$ do
3:         $\Delta_B \leftarrow \textsc{StratB}(\mathcal{A}_A,\mathcal{A}_B,\Sigma_M,\Phi_B^{max})$    ▷ construct $\{(\delta'_A,\delta_B) \mid r((\delta'_A,\delta_B),\Sigma_M) \models \Phi_B$ 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 $r((\delta'_A,\delta_B),\Sigma_M) \models \Phi_B$ and $\Phi_B$ is maximal in $\Sigma_M$ do
6:                 if $r((\delta_A,\delta_B),\Sigma_M) \not\models \Phi_A \cup \Phi_B$ then    ▷ $(\delta_A,\delta'_B)$ is not winning for all M and all $(\delta'_A,\delta_B)$
7:                     return $\textsc{GetJustifications}((\delta_A,\delta_B) \not\models \Phi_A \cup \Phi_B)$
8:                 else
9:                     return $\emptyset$
```

Algorithm 3 Try to fix a conflict by resolving contradictions.

```
1: function $\textsc{FixConflict}(\mathcal{C},(C_i),\Sigma,\Phi_A^{max},\Phi_B^{max})$
2:     for $E \in \mathcal{C}$ do
3:         $\mathcal{C} \leftarrow \mathcal{C} \setminus \{E\}$
4:         $\Sigma,\Phi_A^{max},\Phi_B^{max} \leftarrow \textsc{Resolve}(\Sigma,E,\Phi_A^{max},\Phi_B^{max},(C_i))$    ▷ try resolution according to level $(C_i)$
5:     return $\Sigma,\Phi_A^{max},\Phi_B^{max}$
```

$\Phi_A \cup \Phi_B$ (Alg. 2 L. 6), the function GetJustifications extracts the set of justifications for this conflict situation (Alg. 2 L. 7). The set of justifications is added to the set of conflict causes $\mathcal{C}$ (Alg. 1 L. 8.). Since strategy (δ_A,δ'_B) is not compatible to all of B's strategies, it is hence not further considered as a possible conflict-free strategy for A (Alg. 1 L. 9).

A strategy that remains in Δ_A at Alg. 1 L. 10 is a winning strategy for one of A's goals in all possible worlds Σ_M regardless of what maximal goals B tries to achieve in Σ_M. However, if Δ_A is empty at Alg. 1 L. 10 , A is in a (believed) conflict with B (Def. 1). In this case, conflict resolution is attempted (cf. lines 10 ff. in Alg. 1). Function FixConflict from Alg. 3 is called with the set of conflict causes, the current conflict resolution level, and the current information base and goals. For each conflict cause, an attempt of resolution is made by function Resolve. The conflict is analysed to identify whether adding/updating information of the current resolution level helps to resolve the conflict. If there are several ways to resolve a conflict, justifications can be used to decide which resolution should be chosen. Note that conflict resolution hence means updating of the information base Σ or goal sets Φ_A^{max} and Φ_B^{max}.

Line 13 of Alg. 1 checks if some new information was obtained from the resolution procedure. If not, resolution will be restarted at the next resolution level. 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 at the next resolution level. If Δ_A is empty at level (C_4), the conflict cannot be resolved and the algorithm terminates.

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 $\Sigma_{\mathcal{M}}$ terminates since there is a finite number of possible consistent combinations of formulae and the time horizon for the unrolling of a possible world Σ_M 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 StratA and StratA, terminates 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. If we assume that that sharing information leads only to dismissing possible worlds rather then considering more worlds possible, then this a monotonic process never removing any information.[4] 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 sacrificed 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.

So in summary, the algorithm terminates under certain artificial assumptions but cannot determine a resolution in case without an outside arbiter. In practice such a conflict resolution process has to be equipped with time bounds and monitors. We consider these aspects as future work.

4.3 Case study

We implemented the algorithm sketched above employing Yices [14] to determine contradictions and analysed variations of a toy example to evaluate and illustrate our approach. More details on the case study can be found in [12]. The implementation is available under `https://uol.de/en/hs/downloads/`.

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. 4.2. For the steps (C_1) to (C_4) 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.

[4] Otherwise the set of already examined worlds can be used to define a fixed point.

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 a winning strategy for the model.

5 Related work

Studying Traffic Conflicts According to Tiwari in his 1998 paper [26] studying traffic conflicts in India, one of the earliest studies concerned with *traffic conflicts* is the 1963 paper [21] 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 [26] 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" [1]. 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 [26, 21].

Tiwari also states in [26] that it is necessary to develop a better understanding of conflicts and conjectures that *illusion of control* [23] and *optimism bias theories* like in [13] 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 [15] 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 [25] in order to determine trajectories for vehicles at an intersection. Both works [15, 25] 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 [10] 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 [10], 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 [11] 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 [11] 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 [7] remark, a fully adversarial environment (including B) is usually a bold abstraction. By assuming in (ii) that B maximises 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-guarantee [9] and assume-admissible [7] synthesis. Basically we consider the type of strategy (winning/admissible/dominant) as exchangeable, the key aspect of our definition is that goals are not achievable but can be achieved with the help of the other.

Logics Justification logic was introduced in [5, 6] 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 [4] 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. $t{:}\phi \to K_i\phi$. 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 [2]. While justification logic and related approaches [16, 3], 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 [8]. 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 [22].

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] F. H. Amundsen & C. Hyden (1977): *Proc. of the first Workshop on Traffic Conflicts, Oslo, Norway*. 1st Workshop on Traffic Conflicts, LTH Lund.

[2] E. Antonakos (2007): *Justified and common knowledge: Limited conservativity*. In: International Symposium on Logical Foundations of Computer Science, Springer, pp. 1–11, doi:10.1007/978-3-540-72734-7_1.

[3] S. Artemov & R. Kuznets (2009): *Logical omniscience as a computational complexity problem*. In: Proc. of the 12th Conference on Theoretical Aspects of Rationality and Knowledge, ACM, pp. 14–23, doi:10.1145/1562814.1562821.

[4] S. Artemov & E. Nogina (2005): *Introducing justification into epistemic logic*. Journal of Logic and Computation 15(6), pp. 1059–1073, doi:10.1093/logcom/exi053.

[5] S. N. Artemov (2006): *Justified common knowledge*. Theor. Comput. Sci. 357(1-3), pp. 4–22, doi:10.1016/j.tcs.2006.03.009.

[6] S. N. Artemov (2008): *The Logic of Justification*. Rew. Symb. Logic 1(4), pp. 477–513, doi:10.1017/S1755020308090060.

[7] R Brenguier, J.-F. Raskin & O. Sankur (2017): *Assume-admissible synthesis*. Acta Informatica 54(1), pp. 41–83, doi:10.1007/s00236-016-0273-2.

[8] S. Bucheli, M. Ghari & T. Studer (2017): *Temporal Justification Logic*. In: Proc. of the 9th Workshop on Methods for Modalities, January 2017, EPTCS 243, Open Publishing Association, pp. 59–74, doi:10.4204/EPTCS.243.5.

[9] K. Chatterjee & T. A. Henzinger (2007): *Assume-Guarantee Synthesis*. In: Tools and Algorithms for the Construction and Analysis of Systems, Springer, pp. 261–275, doi:10.1007/978-3-540-71209-1_21.

[10] W. Damm & B. Finkbeiner (2011): *Does It Pay to Extend the Perimeter of a World Model?* In: FM 2011: Formal Methods, Springer, pp. 12–26, doi:10.1007/978-3-642-21437-0_4.

[11] W. Damm, B. Finkbeiner & A. Rakow (2016): *What You Really Need To Know About Your Neighbor*. In: Proc. Fifth Workshop on Synthesis, SYNT@CAV 2016, EPTCS 229, pp. 21–34, doi:10.4204/EPTCS.229.4.

[12] W. Damm, M Fränzle, W. Hagemann, P. Kröger & A. Rakow (2019): *Justification Based Reasoning for Dynamic Conflict Resolution*. arXiv e-prints:arxiv:1905.11764. Available at http://arxiv.org/abs/1905.11764.

[13] D. M. DeJoy (1989): *The optimism bias and traffic accident risk perception*. Accident Analysis & Prevention 21(4), pp. 333 – 340, doi:10.1016/0001-4575(89)90024-9.

[14] B. Dutertre (2014): *Yices 2.2*. In: Computer Aided Verification, Springer, pp. 737–744, doi:10.1007/978-3-319-08867-9_49.

[15] S. El hadouaj, A. Drogoul & S. Espié (2001): *How to Combine Reactivity and Anticipation: The Case of Conflicts Resolution in a Simulated Road Traffic*. In: Multi-Agent-Based Simulation, Springer, pp. 82–96, doi:10.1007/3-540-44561-7_6.

[16] R. Fagin & J. Y. Halpern (1987): *Belief, awareness, and limited reasoning*. Artificial intelligence 34(1), pp. 39–76, doi:10.1016/0004-3702(87)90003-8.

[17] R. Fagin, J. Y. Halpern, Y. Moses & Moshe Y. Vardi (2003): *Reasoning About Knowledge*. MIT Press.

[18] J. Galtung (1969): *Violence, Peace, and Peace Research*. Journal of Peace Research 6(3), pp. 167–191, doi:10.1177/002234336900600301.

[19] J. Gerbrandy (1998): *Distributed knowledge*. In: Twendial 1998: Formal Semantics and Pragmatics of Dialogue, 98, pp. 111–124.

[20] E. L. Gettier (1963): *Is justified true belief knowledge?* Analysis 23(6), pp. 121–123, doi:10.1093/analys/23.6.121.

[21] J. I. Harris & S. R. Perkins (1967): *Traffic conflict characteristics: accident potential at intersections*. Highway Research Board 225, pp. 35–43.

[22] I. Kokkinis, Z. Ognjanovic & T. Studer (2016): *Probabilistic Justification Logic*. In: Logical Foundations of Computer Science - International Symposium, LFCS 2016. Proc., LNCS 9537, Springer, pp. 174–186, doi:10.1007/978-3-319-27683-0_13.

[23] J. E. Langer (1975): *The Illusion of Control*. Journal of Personality and Social Psychology 32, pp. 311–328, doi:10.1037/0022-3514.32.2.311.

[24] O. Lichtenstein, A. Pnueli & L. Zuck (1985): *The glory of the past*. In: Workshop on Logic of Programs, Springer, pp. 196–218, doi:10.1007/3-540-15648-8_16.

[25] N. Murgovski, G. R. de Campos & J. Sjöberg (2015): *Convex modeling of conflict resolution at traffic intersections*. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 4708–4713, doi:10.1109/CDC.2015.7402953.

[26] G. Tiwari, D. Mohan & J. Fazio (1998): *Conflict analysis for prediction of fatal crash locations in mixed traffic streams*. Accident Analysis & Prevention 30(2), pp. 207 – 215, doi:10.1016/S0001-4575(97)00082-1.