ICS 03.220.20
CCS R04

group standard

T/ITS 0293.4-2025

Autonomous Transportation System Traffic Semantic Representation Language Part 4: The Reasoning Framework

Released on 2025-11-26 Implementation on 2025-11-26

China Intelligent Transportation Industry Alliance released

table of contents

preface

This document is drafted in accordance with GB/T 1.1—2020 "Guidelines for Standardization Work Part 1: Structure and Drafting Rules of Standardization Documents".

Please note that some content in this document may involve patents. The publisher of this document is not responsible for identifying patents.

This document is proposed and overseen by the China Intelligent Transportation Industry Alliance (C-ITS).

The primary drafting institutions for this document include: Beijing Jiaotong University, Beijing University of Posts and Telecommunications, the Ministry of Transport's Highway Research Institute, Hualu Yiyun Technology Co., Ltd., China Unicom Smart Network Technology Co., Ltd., Beihang University, Zhuzhou CRRC Times Electric Co., Ltd., and the Ministry of Transport's Water Transport Research Institute.

The primary drafters of this document are: Dong Honghui, Hong Yipeng, Ou Fan, Wu Hao, Yuan Quan, Li Jinglin, Ren Yilong, Yu Haiyang, Li Zhenhua, She Hongyan, Xin Liang, Yu Chaoyang, Wang Quandong, Pan Xiaoxi, Zhou Yucheng, Chen Yi, Wang Zhipeng, Niu Yajie, Wang Jiajia, Li Wei, Zhou Weijie, Lan Zhengxing, Gu Huinan, Ma Panke, Lin Jun, Liu Junlan, and Jiang Peiyuan.

Autonomous Transportation System Traffic Semantic Representation Language Part 4: Reasoning Framework

1 Scope

This paper defines a semantic logic reasoning framework for autonomous vehicle systems, which includes traffic semantic knowledge representation, traffic semantic reasoning rules and traffic semantic reasoning engine.

This document applies to semantic consistency modeling and reasoning in multi-domain transportation systems, including road, rail, and water transport, supporting interoperability and semantic collaboration among transportation entities. It aims to ensure semantic and logical consistency of information between entities, thereby enhancing the efficiency of semantic collaboration and the accuracy of behavioral decision-making across transportation entities.

2 Normative reference documents

This document incorporates the provisions of the following referenced documents into this specification. For references without specified dates, the latest version (including all amendments) shall apply to this specification.

ISO/IEC 24707 Common Logic (CL)

T/ITS 0292-2025 Model of Interoperability Mechanism for Autonomous Transport Systems

T/ITS 0293.1-2025 Autonomous Traffic Systems Traffic Semantics Representation Language Part 1: General Definitions

T/ITS 0293.2-2025 Autonomous Traffic System Traffic Semantic Representation Language Part 2: Grammar Specification

W3C Web Ontology Language 2 (OWL 2)

W3C Semantic Web Rule Language (SWRL)

3GPP TS 24.379 Mission Critical Push To Talk (MCPTT)

3 Terms and Definitions

The following terms and definitions defined in T/ITS 0292-2025, T/ITS 0293.1-2025 and below apply to this document.

3.1

autonomous transportation system

It is a highly intelligent and highly autonomous traffic system characterized by autonomous perception, autonomous decision-making and autonomous execution.

[Source: T/ITS 0292-2025]

3.2

Traffic Semantic Representation Language

Traffic semantic representation language is a formal language to describe traffic content accurately, which has the abilities of semantic representation, semantic understanding, semantic interaction, logical reasoning and interoperability.

[Source: T/ITS 0293.1—2025]

3.3

Autonomous traffic agent

Autonomous traffic agent is a traffic intelligent agent which can realize the closed loop of perception, cognition, decision-making and control in the complex traffic environment and achieve the predetermined traffic task.

[Source: T/ITS 0293.1—2025]

3.4

Traffic Semantic Knowledge Base

The information set of traffic domain knowledge is represented and stored by traffic semantic representation language.

3.5

Traffic Semantic Rule Base

The rules set of traffic semantics representation language describes the traffic behavior logic, operation constraints and semantic reasoning mechanism.

3.6

semantic inference

The process of deriving implicit knowledge from the facts and rules of the existing semantic representation.

3.7

Ontology language

A semantic web language designed to represent rich and complex knowledge about things, groups of things, and relationships between things.

[Source: W3C OWL 2 Web Ontology Language — 2012.12.11]

3.8

Traffic semantic reasoning rules

The formal expression of logical relation and behavior constraint between traffic entities is described, and the semantic-driven decision-making is realized by the inference engine.

3.9

unification unification

The upgraded inference rules require the process of finding the substitution that makes different logical representations become the same.

3.10

forward chaining

A data-driven inference algorithm is proposed, which starts from the known facts in the knowledge base. If all premises of the implication are known, the conclusion is added to the known facts set.

3.11

backward chaining

A form of goal-guided reasoning, which starts from the goal and deduces the link rules backward to find the known facts that support the proof.

3.12

incremental inference

The pattern of local updating and local reasoning of dynamic traffic data.

3.13

Traffic Semantic Binding Table

A variable binding table that records the constrained state of variables and enables backtracking in reasoning.

3.14

Traffic Semantic Dynamic Table

It is a data structure for recording the status of traffic semantic entity, context information and reasoning intermediate results in real time.

3.15

Traffic Semantic Execution Stack

A stack structure used to save the reasoning process, call level and temporary variables.

4 Abbreviations

ABox: Assertional Box

TBox: Terminological Box

CTCS: China Train Control System

CBTC: Communication Based Train Control

AIS: Automatic Identification System

SIP: Session Initiation Protocol

MSRP: Message Session Relay Protocol

MQTT: Message Queuing Telemetry Transport

CoAP: Constrained Application Protocol

RLRL: Traffic Semantic Representation Language

5. Traffic Semantic Reasoning Structure

The traffic semantic reasoning framework comprises four key components: semantic knowledge representation, semantic inference rules, an inference engine, and practical applications, as illustrated in Figure 1. Built upon multi-source traffic data, this framework employs semantic processing, logical analysis, and rule-based operations to establish a closed-loop semantic reasoning system that spans from data perception to intelligent decision-making. The framework's core components are as follows:

a) Traffic Semantic Knowledge Representation (TSRL) is responsible for extracting, cleaning, and semantically describing multi-source traffic data from sensors, communication systems, and business systems. By formally representing traffic entities, attributes, relationships, and events through TSRL, it establishes the foundational content of the traffic semantic knowledge base, providing a unified data semantic support for subsequent reasoning.

b) Traffic Semantic Reasoning Rules: These rules define the logical frameworks and constraints in traffic systems, covering causal relationships between entities, behavioral restrictions, and event-triggered logic. Organized in TSRL format, they support both forward and backward reasoning modes to drive the traffic semantic inference engine in executing logical judgments and generating results.

c) Traffic Semantic Reasoning Engine: As the core component of the semantic reasoning framework, this engine dynamically updates the traffic semantic knowledge base through mechanisms including rule matching, fact unification, spatiotemporal reasoning, and conflict resolution. It enables semantic condition triggering, event classification, state prediction, and conclusion generation. The engine also supports traceability and interpretability analysis of the reasoning process, ensuring logical consistency and stability in complex scenarios.

d) Traffic semantic reasoning applications: These applications enable the business-oriented utilization of reasoning results and facilitate interactions with external systems. Through protocol adaptation and s emantic-to-message mapping mechanisms, the reasoning conclusions are translated into control commands, alert messages, or decision recommendations.

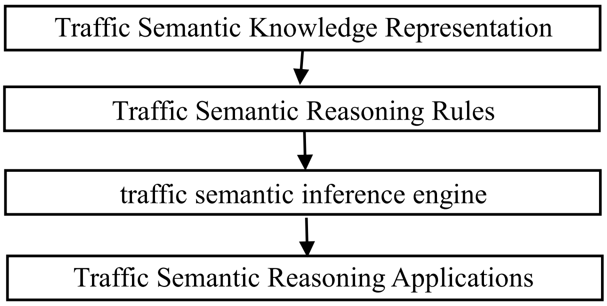

Figure 1 Traffic semantic reasoning structure

Through the synergistic effect of the four levels, the semantic reasoning structure of traffic realizes the semantic data, the logic rules, the intelligent reasoning and the closed loop of application, which provides the system support for the semantic consistency modeling and intelligent collaborative decision-making of multi-traffic modes.

6 Semantic Knowledge Representation of Traffic

6.1 Traffic Semantic Knowledge Representation Process

Traffic semantic knowledge representation is a process of formalizing the entities, relationships and behaviors in the field of traffic based on the ontology language. According to the TSRL terminology library defined by T/ITS 0293.1-2025, the semantic concepts of traffic subjects and their attribute relationships are defined to establish a unified semantic structure model. The formation process of traffic semantic knowledge is shown in Figure 2.

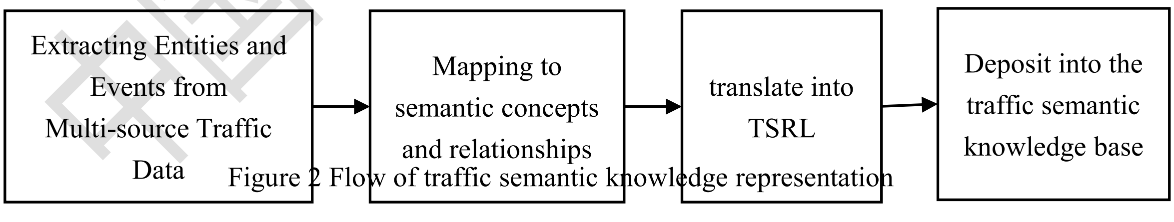

Figure 2 Flow of traffic semantic knowledge representation

6.2 Construction of Traffic Semantic Knowledge Base

The Traffic Knowledge Base (TKB) serves as the core infrastructure for semantic reasoning, consisting of two components: TBox and ABox.

—TBox (Terminology Box): Defines traffic entities, attributes, relationships, and hierarchical structures to form the ontology of the transportation domain.

—ABox (asserting facts): Stores real-time traffic data, including vehicle status, signal status, and segment occupancy.

The construction process of the Traffic Knowledge Base (TKB) is as follows:

—Import the traffic ontology model and define classes and relationship constraints;

— Analyze multi-source data and convert it into TSRL;

— Perform consistency checks and remove redundant or conflicting data;

—Establishing a terminology database according to different transportation modes.

The specific form of the traffic semantic knowledge base is as follows:

```
module TrafficLib{
        category RoadSys {
                class Road {
                        HasRoadType(x,type);
                        HasSpeedLimit(x,limitspeed);
            IsTrafficSignal(x);
            HasTrafficSignalPhase(x,phase);
            ......
        }
            class Vehicle {
                    IsCar(x);
                        HasCarPosition(x,position);
                        HasCarSpeed(x,speed);
            HasCarAcceleration(x,acceleration);
            ......
        }
    ......
        category RailWaySys {
    ......
    }
        category WaterWaySys {
```

......

}

}

7 Traffic Semantic Reasoning Rules

7.1 Structure Pattern of Traffic Semantic Reasoning Rules

The structure elements of traffic semantic reasoning rules are shown in Table 1.

Table 1 Elements of Traffic Semantic Reasoning Rules

Element name	description	ask
RuleID	Traffic Semantic Reasoning Rule Unique Identifier	global unique
RuleName	Traffic Semantic Reasoning Rules Brief Name	Application of traffic semantic reasoning rules
Condition	Traffic Semantic Logic Expression, described with TSRL	Reference to the predicate and parameter types defined in the traffic semantic knowledge base
Conclusion	Traffic facts or actions generated by traffic semantic reasoning	Consistent with the concept of traffic semantic knowledge base
Priority	Priority of traffic semantic reasoning rules	Determine the trigger sequence of traffic semantic inference rules

7.2 Representation Format of Traffic Semantic Reasoning Rules

The semantic inference rules of traffic system are defined by hierarchical relationship, constraints and consistency check of traffic entity, and express complex temporal, spatial and behavioral logic. The semantic inference rules of traffic system are written in the form of extended predicate logic expressions, using the IF-THEN model, and supporting standard logical operators and built-in functions.

General expression format:

RuleID <RuleName>

<Conclusion predicate expression>: - <predicate expression1> [logical connector] <predicate e xpression2>...

[Priority = n]

Predicate expression composition:

— Predicate: A term denoting entities, attributes, or relationships;

—Variable: A universal symbol in the rule;

— Logical connectors (∧, ∨, ¬): used to connect multiple conditions;

—Built-in: Used for comparison, calculation, or conversion.

7.3 Requirements for writing traffic semantic reasoning rules

a) Basic technical requirements:

1) Uniqueness: Each traffic semantic inference rule must have a unique RuleID.

2) Interpretability: The names of traffic semantic inference rules and their metadata should clearly refl ect their purposes.

3) Consistency: the concept and relation used in the condition and result should be consistent with th e definition of traffic ontology;

4) Modularity: Each traffic semantic inference rule should only describe a single logical relationship, which is convenient for maintenance and combination.

5) Machine-executable: Traffic semantic inference rules must follow the specified syntax and be direct ly parseable by the inference engine.

b) Requirements for safety design of the rules:

1) Conflict Resolution Mechanism: The combination of traffic semantic reasoning rules with self-contr adictory conclusions must not exist. The priority of rules must be defined explicitly, and conflict resolution strategies must be set between priority rules.

2) Security constraints: the traffic semantic inference rules should declare the potential dangerous con clusions which are forbidden to be deduced;

3) Physical implementability: The results of traffic semantic reasoning rules must be associated with a ctionable control instructions or warnings.

4) Full-chain traceability: When traffic semantic inference rules are triggered, all input data sources m ust be recorded to support security auditing and accountability.

7.4 Classification of Traffic Semantic Reasoning Rules

The classification of traffic semantic inference rules is presented in Table 2. Based on the requirements for compiling traffic semantic inference rules and the construction of traffic semantic knowledge base, the traffic semantic rule database is developed, as detailed in Appendix A.

Table 2 Classification of Traffic Semantic Reasoning Rules

order number	traffic reasoning rule	content
1	Traffic status recognition rules	Traffic state recognition rules identify the current status of a traffic system from raw perception data, such as the position, speed, and signal status of traffic entities.
2	traffic behavior trigger rule	Traffic behavior trigger rules are used to activate corresponding traffic behaviors or system responses, such as emergency braking, based on current traffic conditions.
3	traffic priority reasoning rule	The traffic priority inference rules are used to derive the traffic order in real time in multi-participant intersection scenarios, such as yielding, avoiding, etc.
4	Traffic Semantic Relation Extension Rules	Used to expand or complete the semantic relationship between traffic concepts, such as "emergency vehicle" implies "can cross red light".
5	Traffic timing reasoning rules	It is used to deal with the traffic behavior logic with time sequence relation, such as first come first served, time-limited passage, etc.
6	Traffic Space Logic Rules	It is used to judge the traffic spatial position relationship between traffic participants, such as lane merging, following distance, blind spot, etc.
7	Traffic Collaborative Reasoning Rules	It is used for the logic of multi-agent cognitive and action coordination, such as lane change negotiation and intersection avoidance.
8	Traffic abnormality detection rules	To identify traffic violations or behaviors that defy common sense, such as driving against traffic flow or occupying emergency lanes.

8 Traffic Semantic Reasoning Engine

8.1 Structure of Traffic Semantic Reasoning Engine

The traffic semantic inference engine comprises four core components: a traffic semantic knowledge base, a traffic semantic rule base, a traffic semantic inference module, and a traffic semantic conflict detector. The inference module retrieves relevant semantic knowledge and logical rules from the knowledge base and rule base based on input reasoning content. After performing semantic inference and conflict resolution, it outputs final traffic decisions while updating both the knowledge base and rule base. The knowledge base manager ensures dynamic updates and consistency maintenance, guaranteeing the system's timeliness and logical coherence. Through these mechanisms, the traffic semantic inference engine achieves intelligent semantic reasoning and closed-loop control. Figure 3 illustrates the engine's architecture.

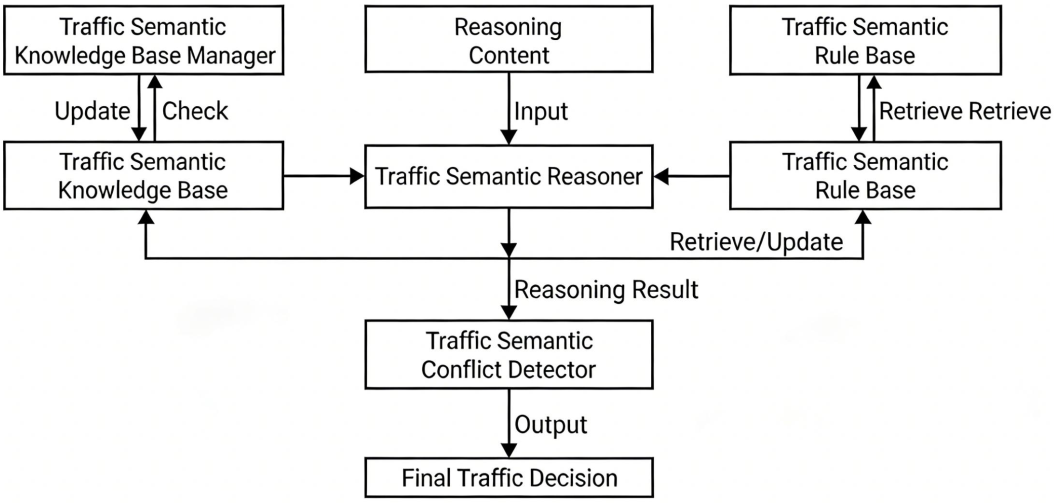

Figure 3 Structure of traffic semantic inference engine

8.2 Traffic Semantic Knowledge Base Manager

The traffic semantic knowledge base manager includes the following:

a) Check for consistency:

1) The current traffic semantic knowledge base is scanned to verify whether each traffic individual meets the definition constraints of the class.

2) If a conflict is found, generate a conflict report and mark the inconsistent entities.

b) Traffic type inference process:

1) Visit all new or updated traffic entities;

2) Based on the class hierarchy and attribute constraints, it determines whether an entity belongs to a more specific subclass and automatically updates the entity type in the traffic semantic knowledge base.

c) Traffic attribute inheritance and relation derivation:

Extend the logic for traffic entity attributes:

1) If the traffic attribute is transitive (e.g., "on the same route"), it automatically deduces the indirect relationship.

2) If the traffic attribute is symmetry (e.g., "adjacent"), the relationship is completed in the opposite

direction.

3) If a traffic attribute chain (e.g., "vehicle-road-section") exists, it leads to a higher-level relationship (e.g., "vehicle belongs to section").

8.3 Traffic Semantic Reasoner

8.3.1 Traffic Semantic Reasoning Mechanism

The traffic semantic inference engine employs the following reasoning methods:

a) Forward link:

1) When the sensor, event or external system inputs new traffic facts, the system adds them to the traffic semantic knowledge base.

2) Trigger the prerequisite conditions of traffic rules based on the traffic facts, and check whether they are met one by one.

3) For the traffic rules that meet the conditions, the conclusion part is implemented, and the newly derived facts are added to the traffic semantic knowledge base.

4) Submitting new traffic facts to the dispatcher may continue to trigger chain reasoning.

b) backward chaining ：

1) When the user or system proposes a target query, the traffic inference engine treats the target as a conclusion to be proven.

2) The rules matching the target in the conclusion part of the traffic semantic rule base are found;

3) The premise of traffic rules is transformed into a new sub-objective, and the traffic semantic knowledge base is checked whether it meets the premise. If not, the other traffic rules that can prove the premise are searched recursively.

4) If all premises are proven, the target conclusion is confirmed. If any premise cannot be proven, the result is failure or partially confirmed.

c) Temporal logic reasoning: This approach centers on temporal constraints, logically representing and deducing the sequence, duration, and concurrency of events. It encompasses the following aspects:

1) The temporal relationship of traffic events is expressed by sequential logic;

2) Supports temporal predicates such as "Before", "After", "During", and "Until" to describe sequential logic in scenarios like traffic light changes, train entry/exit from sections, and vehicle movement at intersections.

3) Through temporal constraint detection and violation judgment, it enables early identification of traffic anomalies.

d) Spatial logic reasoning: This method evaluates spatial relationships and topological positioning among traffic entities, including the following aspects:

1) The geospatial relationship model describes the spatial relationships among traffic entities, such as "adjacent", "inclusive", "crossing", "separation", etc.

2) The spatial relationship predicate is used to express the spatial constraints among vehicles, rail trains, ships and other objects;

3) It supports dynamic spatial analysis by matching GIS coordinates, track section numbers, or route topologies.

e) Spatiotemporal Correlation Reasoning: This approach integrates temporal and spatial dimensions to perform joint reasoning and prediction of complex behaviors, including the following aspects:

1) By establishing the spatio-temporal event model, the event flow on the timeline is associated with the spatial topological state;

2) executing compound rules in the reasoning process;

3) It supports event chain prediction and conflict trend judgment, providing logical support for scheduling decisions and security control.

8.3.2 Traffic Semantic Reasoning Process

The traffic inference system consists of five core components: traffic semantic knowledge base, traffic semantic rule base, traffic semantic binding table, traffic semantic dynamic table, and traffic semantic execution stack. The inference engine workflow is shown in Figure 4, with the following traffic inference steps:

a) Query retrieval and initialization: Users submit queries via the ASK interface, pushing the initial target to the traffic semantic execution stack while simultaneously initializing the dynamic table entries and traffic semantic binding table for this inference session.

b) Candidate rule retrieval: The traffic semantic execution stack sends a request to the traffic semantic rule base for rule matching. The rule base returns a set of candidate rules, which can be sorted by priority or heuristic.

c) Process rules sequentially: For each candidate rule, the traffic semantics stack saves the current traffic semantics binding or dynamic table snapshot for backtracking, then reads the rule's premise.

d) Read candidate facts: For each premise, the traffic semantics stack requests potentially matching facts from the traffic semantics knowledge base, which then returns a set of matching facts.

e) Premise matching and variable unification: For each candidate fact, perform matching and unification with premises. During the matching process, read and update the traffic semantic binding table, then write successful variable constraints into it. If the premise itself is a sub-goal, recursively push it into the traffic semantic execution stack.

f) Record temporary states and intermediate results: Each matching attempt writes to the traffic sema ntics dynamic table to preserve intermediate evidence and constraints. When needed, the traffic semantics dynamic table can provide intermediate results to the traffic semantics execution stack or external modules.

g) Backtracking strategy: If a premise fails to match, use the snapshot recovery variable in the traffic semantic dynamic table or binding table to restore the binding state, then select the next candidate fact or roll back to the previous layer to select the next candidate rule and continue the attempt.

h) Rule validation and conclusion return: When all premises of a rule are successfully matched, the rule is validated. The traffic semantic execution stack generates or returns the matching traffic facts, and then returns the results layer by layer to the initial target.

i) Failure judgment: If all candidate rules and facts have been tried without resolution, the target is deemed failed, and the traffic semantics execution stack returns failure information.

j) After the inference process, the system uniformly releases and clears the temporary variable bindings in the traffic semantic binding table and the temporary states in the traffic semantic dynamic table to maintain the consistency of the traffic semantic knowledge base. Finally, the final conclusion and necessary intermediate evidence are returned to the ASK interface.

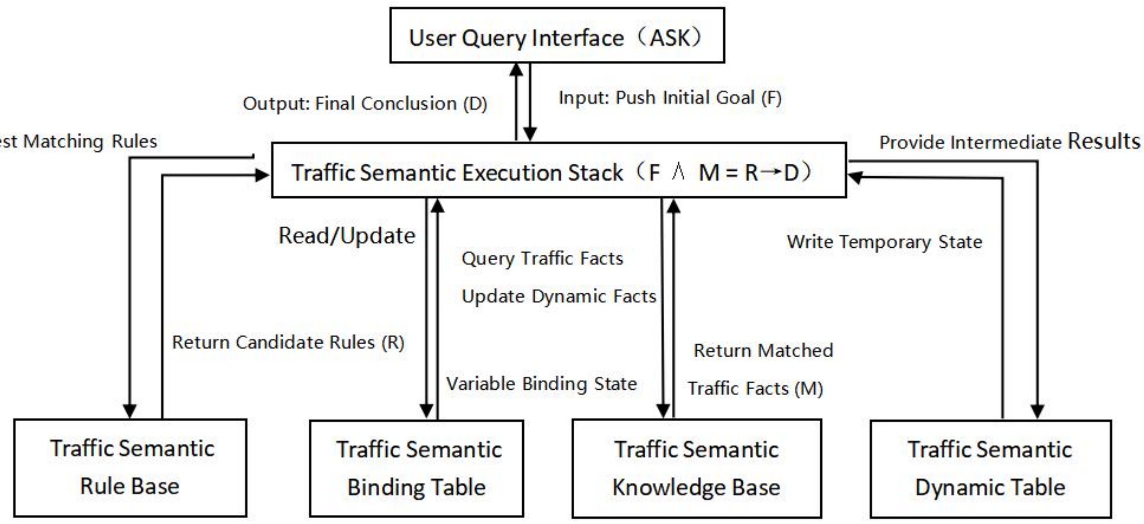

Figure 4: Reasoning Process

8.4 Traffic Semantic Conflict Detector

In complex traffic systems, different semantic inference rule sets or data sources may lead to conflicting or inconsistent prioritization of inference results. The traffic semantic detection and resolution mechanism is designed to identify, classify, and correct semantic conflicts during the inference process, ensuring the reliability and consistency of system conclusions. The specific implementation includes:

a) Traffic semantic conflict detection mechanism: The system performs semantic consistency checks and logical validation algorithms to identify potential conflicts between results derived from different traffic semantic inference rules, including the following aspects:

1) The ontology consistency check is used to verify the compliance of classes, attributes and relationship constraints.

2) The typical logical contradiction is found by conflict pattern recognition, such as "the vehicle is in two places at the same time" or "the signal is red and green at the same time";

3) The traffic semantic inference rules are used to track and propagate the constraint mechanism, and the specific inference path and data source of the conflict are identified.

b) Traffic semantic conflict resolution mechanism: The abrupt resolution mechanism determines the final credible inference results through strategies such as traffic semantic rule priority, time weight, and exception coverage, including the following:

1) The priority principle of traffic semantic inference rules: the explicit priority hierarchy is configured for the set of traffic semantic inference rules, and the high priority rules override the low priority rules in the conflict scenario;

2) Time Last Principle: When the same fact is deduced multiple times, the latest timestamp-based inference takes precedence.

3) Exception coverage principle: When both general traffic semantic rules and exception rules are applicable, the exception rule that fits the specific conditions should be adopted.

c) Traceability and Interpretation Mechanism: To ensure the interpretability and credibility of system decisions, a reasoning traceability mechanism is introduced, which includes the following components:

1) The rule trajectory is generated automatically in each reasoning process, recording the triggered rule sequence and reasoning path.

2) Build a chain of evidence by listing the original data items, sources, and timestamps supporting each rule trigger.

3) Users can view the inference path and evidence chain through the visual interface, enabling transparent result verification and accountability tracing.

8.5 Update of traffic semantic knowledge rule base

The knowledge base of traffic semantics adopts incremental reasoning mechanism to realize the dynamic updating and consistency maintenance of knowledge and rules. The system ensures the efficiency and real-time response even when the multi-source data change frequently by the strategies of dependency tracking, local re-reasoning and result replacement. The main process includes:

-Constructing a semantic reasoning dependency map of traffic, recording each semantic reasoning result and its dependent data, rules and context information;

—When new traffic data is added or existing data is modified or deleted, the system automatically identifies the affected rules and inference chains.

—Update the affected parts through local inference instead of rerunning the entire inference library to improve computational efficiency.

—Replace the old conclusion with the updated inference result in real time to maintain the semantic consistency and timeliness of the traffic semantic knowledge base;

—When the load is high, the incremental inference module can execute necessary subset inference tasks by priority to ensure real-time system response capability.

Appendix A

(Data Appendix)

Traffic Semantic Rules Library Example

A.1 Traffic Semantic Reasoning Rules

a) Example of road rules

Rule1 <Status Recognition Rule>

OverspeedVehicle(x,t) :- IsCar(x,v) ∧ HasSpeed(x,v) ∧ HasRoadType(r,Urban) ∧ On(v,r) ∧ GreaterThan(v,30).

The vehicles that speed over 30 km/h on the city road are identified as speeding.

Rule2 <Behavior Trigger Rule>

MustStop(v,t):- Approaching(v,SignalizedIntersection,t) ∧ SignalColor(t,red) ;

The vehicle must stop when approaching the intersection with a red light.

Rule3 <Priority Reasoning Rule>

HasRightOfWay(v1,v2):- IsEmergencyVehicle(x1) ∧ ¬IsEmergencyVehicle(x2) ∧ ConflictAtJunction(x1,x2,t);

Emergency vehicles have priority to non-emergency vehicles when they collide at the intersection.

Rule4 <semantic relation extension rule>

SchoolBus(x):- CanStopInMovingTraffic(x) ∧ LetAllStop(x);

The bus semantics implied that it could stop on the road and trigger other vehicles to stop completely.

Rule5 <Temporal Reasoning Rule>

HasPriority(x1,x2):- ArrivalAtMerge(x1,t1) ∧ ArrivalAtMerge(x2,t2) ∧ LessThan(t1,t2);

The vehicle that arrives first at the confluence point has the right of way.

Rule6 <space> logical rule

TailgatingWarning(x2,t):- IsBehind(x2,x1) ∧ DistanceBetween(x1,x2,d,t) ∧ LessThan(d,SafeDistance(x1,v,t));

The rear car and the front car distance is less than the safe following distance, which constitutes the rear-end warning.

Rule7 <Collaborative Reasoning Rules>

ShouldCreateGap(x1,x2,t):- RequestLaneChange(x1,Left,t) ∧ HasSpace(x2,t) ∧ FlashLight(x1,t);

When a vehicle requests to change lanes to the left with its lights flashing, the neighboring vehicle should actively yield space.

Rule8 <Abnormal Detection Rule>

IllegalShoulderUseWarning(x,t):- Occupied(x,HardShoulder,t) ∧ ¬IsEmergency(x) ∧ ¬IsBreakdown(x);

The non-emergency state occupies the emergency lane and triggers a violation warning.

b) Example of track rules

Rule1 <Status Recognition Rule>

CriticalFaultStatus(y,t):- IsSegmentA(y) ∧ SignalFailure(y,t) ∧ TrackOccupied(y,t);

The orbit section signal fault and occupied is identified as serious fault state.

Rule2 <Behavior Trigger Rule>

ActivateBarrier(y,t):- IsTrain(x) ∧ IsCrossing(y) ∧ Approaching(x,y,d,t) ∧ LessThan(d,500);

The barrier is triggered when the train is 500 meters away from the crossing.

Rule3 <Priority Reasoning Rule>

YieldPriority(x2,x1):- IsTrain(x1) ∧ IsTrain(x2) ∧ HighSpeedRail(x1) ∧ FreightRail(x2) ∧ SharedTrack(x1,x2,t);

In the mixed passenger and freight lines, freight trains must give way to high-speed trains.

Rule4 <semantic relation extension rule>

MaintenanceActivity(x):- IsTrain(x) ∧ ImposesSpeedRestriction(x,40)∧RequiresSafetyZone(x,50);

The track section related to the maintenance activity triggers the speed limit of 40 km/h and the safety zone of 50 meters.

Rule5 <Temporal Reasoning Rule>

DispatchNext(x1,x2,t):- IsTrain(x1) ∧ IsTrain(x2) ∧ IsStation(y) ∧ ArrivalAtStation(x1,y,t1) ∧ ArrivalAtStation(x2,y,t2) ∧ LessThan(t1,t2).

The first arriving train is given priority in dispatching departure.

Rule6 <space> logical rule

MaintainHeadway(x1,x2,t):- IsTrain(x1) ∧ IsTrain(x2) ∧ SameLine(x1,x2) ∧ DistanceBetween(x1,x2,d,t) ∧ GreaterThan(d,MinSafeHeadway).

Two non-connected trains in the same block area constitute collision risk.

Rule7 <Collaborative Reasoning Rules>

ApproveRerouting(x,Platform2,t):- IsTrain(x) ∧ IsPlatform(y) ∧ RequestPlatformChange(x,y,t) ∧ AvailableCapacity(y,t);

When the train requests to change the platform and the capacity is sufficient, the path adjustment should be approved.

Rule8 <Abnormal Detection Rule>

OverspeedWarning(x,t)]:- IsTrain(x) ∧ IsSection(y) ∧ DetectedSpeed(x,v,t) ∧ MaxAllowed(y,Vmax) ∧ GreaterThan(v,Vmax+10%);

An overspeed warning is triggered when the detected speed exceeds the section speed limit by 10%.

c) Example of water transport regulations

Rule1 <Status Recognition Rule>

OverdraftVessel(x,t):- IsVessel(x) ∧ IsWaterWay(w) ∧ HasVesselSpeed(x,v) ∧ GreaterThan(v,LimitSpeed(w)) ∧ On(v,w);

The ship is identified as overspeeding when the speed exceeds the speed limit in a specific channel.

Rule2 <Behavior Trigger Rule>

ActivateWarningSignal(x,t):- IsVessel(x) ∧ IsPortEntry(y) ∧ Near(x,y,500m,t) ∧ HasWeatherWarning(y,Storm);

When the ship approaches the port entrance and there is a storm warning, the warning signal should be triggered.

Rule3 <Priority Reasoning Rule>

HasNavigationalPriority(x1,x2):- IsVessel(x1) ∧ IsVessel(x2) ∧ IsLargeVessel(x1)∧ IsSmallVessel(x1) ∧ ConflictAtWaterway(x1,x2,t);

Large tonnage vessels have priority to pass small vessels in the channel conflict.

Rule4 <semantic relation extension rule>

IsVLCC(x):- IsVessel(x) ∧ RequiresTugEscort(x) ∧ ImposesTrafficSeparation(x);

If a vessel satisfies both the 'tug escort required' and 'may cause channel separation' criteria, it can be classified as an IsVLCC (Super Vessel Large Crude Carrier).

Rule5 <Temporal Reasoning Rule>

MustExecuteHoldingPattern(x, t):- IsVessel(x) ∧ IsPort(y) ∧ ETA(x, y, etatime) ∧ TidalWindow(y, starttime, endtime) ∧ NotWithin(etatime, starttime, endtime);

If the expected arrival time is outside the tidal window, the aircraft must circle and stand by.

Rule6 <space> logical rule

SinglePassageRequired(y, t):- IsVessel(x) ∧ IsWaterWay(y) ∧ InNarrowChannel(x, y, t) ∧ HasVesselWidth(x, width1) ∧ ChannelWidth(y, width2) ∧ LessThan(width2 - width1, SafeMargin).

If a vessel is navigating in a narrow channel with a channel width less than the safety margin, the channel shall be required to be one-way at that time.

Rule7 <Collaborative Reasoning Rules>

ExecuteBerthingOperation(x,t):- IsVessel(x) ∧ IsPort(y) ∧ ArrivalConfirmed(x,y,t1) ∧ TugAssigned(x,tug,t1) ∧ PilotOnboard(x,y,t1) ∧ WeatherCondition(y,Safe,t1);

The berthing operation shall be carried out when the ship has arrived at the port, the tugboat and pilot are in place, and the weather conditions are safe.

Rule8 <Abnormal Detection Rule>

ActivateRescue(x,t):- IsVessel(x) ∧ Detected(DistressSignal,x,t) ∧ Near(x,IncidentArea,2km);

The ship automatically enters the emergency rescue state when the distress signal is detected in the lane and the ship is close to the accident area.

Appendix B

(Data Appendix)

Traffic Semantic Reasoning Example

B.1 Traffic semantic reasoning scenarios

Taking the railway section station freight train dispatching and passenger train priority scene as an example, the traffic semantic logic reasoning is carried out.

B.2 traffic semantic reasoning

a) initial state detection

The equipment on the side of the track will upload the current line occupancy and the train operation plan to the dispatching center in real time.

Send by FreightTrain1&FreightTrain2&PassengerTrain1: IsPassengerTrain(PassengerTrain1), IsFreightTrain(FreightTrain1), IsFreightTrain(FreightTrain2), OnTrack(FreightTrain1, TrackA), OnTrack(FreightTrain2, TrackB), PlannedRoute(PassengerTrain1, TrackA) Piority(PassengerTrain1, 1);

Received by ControlCentreTrafficDispaicher: IsPassengerTrain(PassengerTrain1), IsFreightTrain(FreightTrain1), IsFreightTrain(FreightTrain2), OnTrack(FreightTrain1, TrackA), OnTrack(FreightTrain2, TrackB), PlannedRoute(PassengerTrain1, TrackA) Piority(PassengerTrain1, 1);

Each train reports to the dispatch center: FreightTrain1 is on main track TrackA, FreightTrain2 is on side track TrackB, PassengerTrain1's planned route is TrackA, and PassengerTrain1 has the highest priority (priority=1).

b) Collision Detection and Scheduling Decision

Inferred by ControlcentreTraficDispatcher: Conflict(PassengerTrain1, FreightTrain1, TrackA), Control(FreightTrain1, TransferTo(TrackC));

Send by ControlcentreTraficDispatcher: Control(FreightTrain1, TransferTo(TrackC));

Received by TracksideEauipment: Control(FreightTrain1, TransferTo(TrackC));

The dispatch center identified through logical analysis that PassengerTrain1 and FreightTrain1 were in conflict on TrackA. Following the priority rule (PassengerTrain1 takes precedence), the center issued a decision to the trackside equipment: directing FreightTrain1 to switch to the idle TrackC to resolve the conflict.

c) Path availability feedback

Send by TracksideEquipment: TrackAvailable(TrackC), NoConflict(TrackC), RightSwitchPosition(TrackC);

Received by ControlCentreTrafficDispaicher: TrackAvailable(TrackC), NoConflict(TrackC), RightSwitchPosition(TrackC);

Description: Trackside equipment continuously provides real-time feedback to the dispatch center regarding TrackC status: TrackC is available and unoccupied; no other trains are occupied or in conflict; relevant switches are in the correct positioning state, allowing entry. This ensures the feasibility of the transfer plan for the dispatch center.

d) Transfer Line Authorization

Inferred by ControlCentreTrafficDispatcher: LetGo(FreightTrain1, TrackC);

Send by ControlCentreTrafficDispatcher: LetGo(FreightTrain1, TrackC);

Received by FreightTrain1:LetGo(FreightTrain1, TrackC);

The dispatch center issues a control command to FreightTrain1: authorizing it to enter TrackC to complete the transfer operation, thereby releasing TrackA. This constitutes the critical transition from 'dispatch decision' to 'actual execution'.

e) main orbit state update

Send by TracksideEquipment: TrainPosition(PassengerTrain1), TrackAvailable(TrackA), NoConflict(TrackA), RightSwitchPosition(TrackA);

Received by ControlCentreTrafficDispaicher: TrainPosition(PassengerTrain1), TrackAvailable(TrackA), NoConflict(TrackA), RightSwitchPosition(TrackA);

Function: Trackside equipment continuously transmits real-time data to the dispatch center, including PassengerTrain1's position and track status. The straight track confirms PassengerTrain1 has cleared the section, TrackA is free with no conflicts, and the turnout is properly positioned, authorizing FreightTrain1 to proceed as planned. This sets the stage for subsequent clearance authorization.

f) Approach clearance issued

Inferred by ControlCentreTrafficDispatcher: TrackAPassable(FreightTrain1, TrackA);

Send by ControlCentreTrafficDispatcher: TrackAPassable(FreightTrain1, TrackA);

Received by FreightTrain1: TrackAPassable(FreightTrain1, TrackA);

Function: The dispatch center issues a 'route clearance' to FreightTrain1 based on track status and priority rules, authorizing it to enter and pass TrackA. PassengerTrain1 has departed the section, while FreightTrain2 remains in a pending avoidance state, ensuring the entire route is safe.

Appendix C

(Data Appendix)

Application of traffic semantic reasoning

order number	mode of transportation	Traffic Semantic Reasoning Applications
1	road	Passing Priority Judgment Based on Vehicle State and Signal State
2		Congestion Detection and Dynamic Signal Timing Based on Road Occupancy Information
3		Accident Detection and Early Warning Based on Event Logic
4		Pedestrian Avoidance and Dangerous Behavior Recognition Based on Semantic Rules
5	track	Train Section Occupation and Route Opening Condition Judgment
6		Semantic Interoperability and Standard Boundary Coordination of Train Control System
7		Priority Reasoning and Conflict Resolution in Column Sets
8		Automatic Verification and Logical Consistency Check of Dispatching Instructions
9	water carriage	Ship Track Prediction and Route Conflict Detection
10		The Order of Berthing and Priority Reasoning in Port Area
11		Dynamic Risk Identification Based on AIS Semantic
12		Ship Grouping and Collision Avoidance Strategy Generation

T/ITS 0293.4-2025

China Intelligent Transportation Industry Alliance
Standard
Self-organizing traffic system Traffic semantics representation language Part 4: The Reasoning Framework
T/ITS 0293.4-2025

8 Xitucheng Road, Haidian District, Beijing (100088)
China Intelligent Transportation Industry Alliance Printing
Website: http://www.c-its.org.cn

2025 11 First edition 2025 11 First printing