

团体标准

T/ITS 0293.2-XXXX

自主式交通系统 交通语义表示语言 第2部分：语法规范

Autonomous transportation system

Traffic semantic representation language part 2: Syntax Specification

(征求意见稿)

(本稿完成时间：2025年9月)

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

20XX-XX-XX 发布

20XX-XX-XX 实施

中国智能交通产业联盟 发布

目 次

前 言 1

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 交通语义表示语言词法规范 2

5 交通语义表示语言句法规范 6

6 交通语义表示语言使用规范 9

中国智能交通产业联盟

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国智能交通产业联盟（C-ITS）提出并归口。

本文件主要起草单位：待定。

本文件主要起草人：待定。

自主式交通系统 交通语义表示语言 第2部分：语法规范

1 范围

本文件规定了自主式交通系统中的交通知识语义表示相关语法规范，并支撑多交通主体之间的语义交互。

本文件适用于自主式交通系统中的交通知识表达，包括但不限于交通规则、交通标志、交通信号、车辆行为等。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 1.1-2020 标准化工作导则 第1部分：标准化文件的结构和起草规则

ISO/IEC 13211-1:1995 信息技术 编程语言 Prolog 第1部分：通用核心

ISO/IEC 14882:2020 信息技术 编程语言 C++

Python Language Reference 3.10 Python语言参考手册3.10

Oracle. Java Language Specification Java语法规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

自主式交通系统 autonomous transportation system:

自主式交通系统是以自感知、自适应、自学习、自组织为特征的高度自治的交通系统。

3.2

交通语义表示语言 traffic semantic representation language:

交通语义表示语言是用于描述和定义自主式交通系统中交通行为、规则、状态及交互的规范化语言。

3.3

一阶逻辑表示first order logical representation:

指使用逻辑符号和结构来表达和描述概念、关系、条件等信息，以实现清晰的知识传递和分析。

3.4

事实fact:

描述交通实体属性的基本语句，由“fact”关键字定义。

3.5

关系relation:

描述交通实体间关联的语句，由“relation”关键字定义。

3.6

规则rule:

描述交通逻辑约束的条件语句，由“rule”关键字定义。

4 交通语义表示语言词法规范

本章规定了交通语义表示语言的词法规范。

4.1 字符集和编码

在本交通语义表示语言中采用Unicode字符集，支持中英文交通术语，源文件编码格式为UTF-8。

4.2 标识符

——TSRL中标识符用于命名模块、类别、类、属性、谓词等，由字母、数字、下划线组成。

——模块名、类别名、类名，首字母大写，如TrafficLib、Vehicle。

——属性名、谓词名、函数名采用大写字母开头，如HasSpeed、Distance。

——常量为特定实体命名，由大写字母开头，可以包含数字和下划线，如Vehicle_1、Lane5。

——变量由小写字母或下划线开头，若仅有下划线符号，则为匿名变量，如_、speed(Car1)。

4.3 关键字

关键字是交通语义表示语言中用于构建核心语法结构的基础符号，通过特定标识界定术语组织形式、实体属性、关联关系及操作逻辑，是构建规范化交通语义描述体系的核心元素。具体关键字及说明如下表所示：

表 1 关键字

关键字	说明
module	定义术语库模块
category	定义交通系统类别
class	定义交通实体类
fact	定义实体属性事实
relation	定义实体间关系
action	定义实体动作
rule	定义逻辑规则
let	定义交通实体操作
asserta	定义知识库前插入操作
assertz	定义知识库后插入操作
retract	定义知识库删除操作
ASK	信息查询/询问
TELL	传递/告知信息
Print	定于打印语句
fun	定义自定义函数
False	表示逻辑假值
True	表示逻辑真值
Nil	表示控制/无实体
and	逻辑与运算符
or	逻辑或运算符
not	逻辑非运算符

4. 4 基本数据类型

4. 4. 1 数值常量

- 整型：int型整数为32位有符号数，如：1、20、333。
- 浮点型：float为小数数值对象的表示方法，32位单精度浮点数。

4. 4. 2 布尔值

实际值为布尔值的状态判断符类型，True为真值、False为假值。

在TSRL语言中，使用Nil表示空值。

4.4.4 字符串

由双引号包裹的文本（如"十字路口"、"green"），支持转义字符n（换行）、\"（双引号）。

4.5 复合数据类型

4.5.1 表

表是 TSRL 语言中用于组织交通实体集合的可变容器型实体类型，支持动态管理多类型交通语义元素，适配交通场景中动态变化的实体组需求（如车道车辆队列、事件序列）。表实体化的一般形式为：

$$l = [x_1, x_2, \dots x_n];$$

- 符号说明：[]为表的界定符； $x_i(i=1,2,\cdots,n)$ 为表中的项，代表交通实体或语义元素。
- 特殊形式：不含任何元素的表称为空表，记为[]，可用于初始化待动态填充的交通实体集合。

4.5.2 字典

字典是 TSRL 语言中用于建立交通语义键值映射的可变容器型实体类型，通过唯一键关联交通实体或属性值，适配交通场景中“标识-语义”映射需求（如信号灯相位-时长映射、车辆 ID-状态映射）。字典实体化的一般形式为：

$$d = \{key_1: value_1, key_2: value_2, \dots key_n: value_n\};$$

- 符号说明：{}为字典的界定符； $key_n: value_n$ 为键值对，“:”是键与值的分隔符，键值对之间用“,”分隔。
- 键的约束：键用于唯一标识交通语义项，值为关联的实体或属性。

4.6 运算符与分隔符

表 2 比较运算符

符号	用途
>	判断左侧数值是否大于右侧数值
<	判断左侧数值是否小于右侧数值
>=	判断左侧数值是否大于等于右侧数值
<=	判断左侧数值是否小于等于右侧数值
==	判断两侧值是否相等（支持数值、字符串、布尔值）

!=	判断两侧值是否不相等
----	------------

表 3 算术运算符

符号	用途
+	数值加法运算/字符串拼接
-	数值减法运算/表示负数值
*	数值乘法运算
/	浮点除法运算，结果保留小数
**	幂运算
//	整数除法运算，结果取整数部分
%	取余运算

表 4 逻辑运算符

运算符	名称	优先级	示例
¬	否定	1	not P(x)
∧	合取	2	P(x) and Q(x) P(x) & Q(x)
∨	析取	3	P(x) or Q(x) P(x) Q(x)
⇒	蕴含	4	Q(x) ⇒ P(x)
↔	等价	5	P(x) ↔ Q(x)

表 5 量词运算符

运算符	名称	格式	示例
∀	全称量词	∀ x P(x)	Forall xP(x)
∃	存在量词	∃ x P(x)	Exists xP(x)

表 6 分隔符

符号	用途
()	包裹函数及谓词的参数列表/改变表达式运算优先级

[]	包裹表的元素列表/表的索引访问
{ }	包裹模块、类别、类的代码块/包裹字典的键值对
;	语句结束符

续表 6 分隔符

符号	用途
,	分隔参数/分隔元素/分隔键值对
:	分隔字典中的键与值/类中属性的类型定义

4.7 注释

TSRL语言中的注释由“#”开头，至物理行尾结束，形式如下：

#定义车辆类属性

5 交通语义表示语言句法规范

本章规定了交通语义表示语言句法规范，定义了组成交通语义表示语言的句法结构和类。

5.1 TSRL 句法结构

- (1) 事实的句法结构，用于描述交通系统中的具体信息和状态，规定采用谓词和常量等来表示这些事实，结构如下：

<Fact>

<谓词名> (<项表>);

<Fact>

- 谓词分为实体、类别、属性、关系。
- 实体：Vehicle(ID)。例：Vehicle(x)表示ID为x的车辆实体。
- 类别：Is谓词(参数1, 参数2, ..., 参数n)。例：IsVehicle(x)，表示x是车辆。
- 属性：Has谓词(参数1, 参数2, ..., 参数n)。例：HasSpeed(Vehicle, Speed)表示车辆Vehicle的速度为Speed。
- 关系：谓词(参数1, 参数2, ..., 参数n)。例：On(Vehicle, Road)表示车辆在某条道路上。

- (2) 规则的句法结构，用于描述交通系统中的约束条件和行为规范，包含条件和结论，使用逻辑运算符和量词来表示，形式如下：

<Rule>

<谓词名>(<项表>) :- <谓词名>(<项表>){,<谓词名>(<项表>)};

<Rule>

—— “:-” 号表示 “if”（也可以直接写为if），其左部的谓词是规则的结论（亦称为头），右部的谓词是规则的前提（亦称为体）。

—— “{}” 表示零次或多次重复，逗号表示and（逻辑与）。

- (3) 提问的句法结构，用于获取系统中的信息，规定通过谓词和变量来表示查询条件和结果。其基本在谓词语句前面加一个特殊的符号 “?-”，形式如下：

?-<谓词名>(<项表>){,<谓词名>(<项表>)};

- (4) 操作的句法结构，用于描述交通系统中实体的具体操作或行为，规定将其分为控制、知识库操作、赋值操作，形式如下：

- 1) 控制：表示对某个对象执行的具体动作指令，通过 Let 关键字，可以触发某个行为，形式如下：

Let(<谓词名>(<项表>){,<谓词名>(<项表>)});

- 2) 知识库操作：就知识库即内存中的动态数据结构进行操作，由事实组成，规定 3 个动态数据库操作谓词，形式如下：

KBSelect(<谓词名>);

asserta(<fact>);

assertz(<fact>);

retract(<fact>);

——<fact>表示对应事实的某条谓词表达式。

——KBSelect(<谓词名>)查询语句，程序返回动态知识库中包含对应谓词名的谓词表达式。

——asserta(<fact>)把 fact 插入当前动态数据库中的同名谓词的事实之前。

——assertz(<fact>)把 fact 插入当前动态数据库中的同名谓词的事实之后。

——retract(<fact>)把 fact 从当前动态数据库中删除。

- 3) 赋值操作：将某个表达式的值赋予另一个对象，同时返回被赋予的值。形式又以下两种：

<项> = <项>;

<项> is <项>;

——使用 “=” 进行赋值时，如果实体属于基本实体类型，会新建一个实体进行引用。

——使用赋值号 is，则会使得两个实体指向同一地址。

- 4) 打印操作：Print 语句用于打印表达式的值。

Print Expression;

- 如果表达式 Expression 为计算表达式且可计算，直接打印计算结果。
- 如果表达式 Expression 为谓词表达式，直接打印出该谓词表达式，而不会验证其真值。

5.2 类

(1) TSRL 中类的概念及特性

- 类是对物理世界某一集体概念的抽象。
- 类具有某些属性，这些属性实体是某个类型的实例化。
- 类存在某些关系，表现为谓词/函词表达式。创建类时会自带一些谓词/函词表达式，用于实体类型检验、属性取值、属性检验等。用户也可自行构建类的谓词/函词表达式。
- 类包含某些规则，这些规则使用规则语句表示。
- 类的继承，继承关系满足谓词/函词表达式 $\text{Subset}(\text{subclass}, \text{superclass})$ ，同时派生类也有用于实体类型检验、属性取值、属性检验的谓词/函词表达式。

(2) TSRL 中类的构造

构造类的基本表示形式为：

```
class classname(superclass) {
    ClasstypeA propertyA;
    PredicateNameA(entityp,1, entityp,2, ...);
    FunctionNameA(entityf,1, entityf,2, ...) = entityf,x;
    RuleNameA(entityr,1, entityr,2, ...) :- PredicateName1(entityrp,1, entityrp,2, ...), ...
};
```

(3) TSIL 类的实例化/实体化

实例化语句：

classname(entityname);

- 类实体化，若创建载具类 Vehicle 的实体 car1，TSRL 语句为：Vehicle(car1)。

(4) TSRL 类的相关谓词与函词表达式

在构造类时，除了用户自己构建的谓词/函词表达式，TSRL 会自动生成谓词、函词表达式。

谓词表达式：

- Isclassname(entity)，用于检验实体 entity 是否属于类 classname。

——PropertyA(entity, value), 用于表示实体 entity 中属性 PropertyA 的值为 value。

函词表达式:

——PropertyA(classentity) = entityPropertyA, 指向实体 classentity 的属性 entityPropertyA。

关于类和实体, TSRL 中内置的谓词/函词表达式部分如下:

——Member(entity, classname)用于检验实体 entity 是否属于类 classname。

——PropertyOf(entity, classname)用于检验实体 entity 是否是类 classname 的属性。

——PropertyOf(entity1, entity2)用于检验实体 entity1 是否是实体 entity2 的属性。

——TypeOf(entity1) = classname 用于取得实体 entity1 所属类的类名 classname。

6 交通语义表示语言使用规范

本章以自主式交通系统交叉口闯红灯预警场景为核心, 按场景描述、状态描述、规则描述、逻辑推理流程, 规范交通语义表示语言的使用, 所有描述均基于一阶逻辑谓词与TSRL语法实现, 确保语义严谨性与可操作性。

6.1 场景描述

聚焦城市信号灯控制交叉口, 涉及车辆、信号灯、环境三个主体, 排除极端天气、突发障碍物等干扰因素, 核心目标为通过逻辑推理判断车辆闯红灯风险并触发预警。场景描述如下:

(1) 场景主体:

——IsVehicle(v): v为车辆实例。

——IsTrafficLight(tl): tl为信号灯实例。

——IsEnv(env): env为环境实例。

(2) 场景边界:

——InSameIntersection(v, tl): 车辆v与信号灯tl处于同一交叉口。

——RoadCondition(env, “good”): 环境env中道路状况良好。

——WeatherCondition(env, “sunny”): 环境env天气为晴。

——TrafficCongestion(env, “no”): 环境env无拥堵。

——TrafficLight(env, “normal”): 信号灯tl运行正常。

(3) 场景目标:

——WarnRedLight(v, tl): 对车辆v与信号灯tl构成的交互场景, 实现闯红灯预警。

6.2 状态描述

(1) 车辆状态谓词

标识属性:

——HasVehicleID(v , id): 车辆 v 的唯一标识为 id 。

运动属性:

——HasSpeed(v , s): 车辆 v 的行驶速度为 s , 单位: km/h。

——HasDistanceToStopLine(v , d): 车辆 v 距交叉口停止线距离为 d , 单位: m。

(2) 信号灯状态谓词

标识属性:

——HasTLID(tl , tid): 信号灯 tl 的唯一标识为 tid 。

控制属性:

——TrafficLightState(tl , $color$): 信号灯 tl 的当前相位颜色为 $color$, 取值red/green/yellow。

——HasPhaseRemainTime(tl , t): 信号灯 tl 当前相位剩余时间为 t , 单位: s

——TrafficLight(tl , $status$): 信号灯 tl 的运行状态为 $status$, 取值normal/abnormal。

(3) 环境状态谓词

基础属性:

——RoadCondition(env , rs): 环境 env 的路面状态为 rs , 取值good/bad。

——WeatherCondition(env , w): 环境 env 的天气状况为 w , 取值sunny/rainy/foggy。

——TrafficCongestion(env , c): 环境 env 的拥堵状况为 c , 取值no/low/medium/high。

6.3 规则描述

(1) 闯红灯预警触发规则1: 信号灯为绿灯/黄灯场景

逻辑描述:

$$\begin{aligned} & \forall v, tl, env (IsVehicle(v) \wedge IsTrafficLight(tl) \wedge IsEnv(env) \wedge InSameIntersection(v, tl) \wedge \\ & (TrafficLightState(tl, "green") \vee TrafficLightState(tl, "yellow")) \wedge HasGreenRemainTime(tl, t_g) \wedge \\ & HasYellowRemainTime(tl, t_y) \wedge ((t_g + t_y) > 0) \wedge HasDistanceToStopLine(v, d) \wedge \\ & HasSpeed(v, s) \wedge ((d/(s*1000/3600)) > (t_g + t_y)) \wedge RoadCondition(env, "good") \wedge \\ & TrafficLight(tl, "normal") \wedge TrafficCongestion(env, "no") \wedge WeatherCondition(env, "sunny")) \\ & \Rightarrow RedLightWarnTriggered(v) \end{aligned}$$

代码实现:

#闯红灯预警触发规则1：当前信号灯为绿灯/黄灯场景

rule RedLightWarnTriggered1(v) :-

```
IsVehicle(v) & IsTrafficLight(tl) & IsEnv(env) & InSameIntersection(v, tl)
& (TrafficLightState(tl, "green") | TrafficLightState(tl, "yellow"))
& HasGreenRemainTime(tl, t_g) & HasYellowRemainTime(tl, t_y) & ((t_g + t_y) > 0)
& HasDistanceToStopLine(v, d) & HasSpeed(v, s) & ((d/(s*1000/3600)) > (t_g + t_y))
& RoadCondition(env, "good") & TrafficLight(tl, "normal")
& TrafficCongestion(env, "no") & WeatherCondition(env, "sunny");
```

对任意车辆v、信号灯tl、环境env，若满足：

- a) 同一交叉口
- b) 信号灯为绿灯或黄灯
- c) 绿灯剩余时间与黄灯剩余时间之和>0
- d) 车辆匀速到达停止线时间>绿灯剩余时间+黄灯剩余时间
- e) 道路良好
- f) 信号灯正常
- g) 无拥堵
- h) 天气晴

则车辆v触发闯红灯预警，即：车辆到达时信号灯已转为红灯，存在闯红灯风险。

(2) 闯红灯预警触发规则2：信号灯为红灯场景

逻辑描述：

$$\forall v, tl, env \ (IsVehicle(v) \wedge IsTrafficLight(tl) \wedge IsEnv(env) \wedge InSameIntersection(v, tl) \wedge \\ TrafficLightState(tl, "red") \wedge HasRedRemainTime(tl, t_r) \wedge (t_r > 0) \wedge HasSpeed(v, s) \wedge \\ HasDistanceToStopLine(v, d) \wedge ((d/(s*1000/3600)) < t_r) \wedge RoadCondition(env, "good") \wedge \\ TrafficLight(tl, "normal") \wedge TrafficCongestion(env, "no") \wedge WeatherCondition(env, "sunny"))$$

$\Rightarrow RedLightWarnTriggered(v)$

代码实现：

#闯红灯预警触发规则2：当前信号灯为红灯场景

rule RedLightWarnTriggered2(v) :-

```
IsVehicle(v) & IsTrafficLight(tl) & IsEnv(env) & InSameIntersection(v, tl)
```

T/ITS 0293.2-XXXX

& TrafficLightState(tl, "red") & HasRedRemainTime(tl, t_r) & (t_r > 0)
& HasDistanceToStopLine(v, d) & HasSpeed(v, s) & ((d/(s*1000/3600)) < t_r)
& RoadCondition(env, "good") & TrafficLight(tl, "normal")
& TrafficCongestion(env, "no") & WeatherCondition(env, "sunny");

对任意车辆v、信号灯tl、环境env，若满足：

- a) 同一交叉口
- b) 信号灯为红灯
- c) 红灯剩余时间>0
- d) 车辆匀速到达停止线时间<红灯剩余时间
- e) 道路良好
- f) 信号灯正常
- g) 无拥堵
- h) 天气晴

则车辆v触发闯红灯预警，即：车辆到达时信号灯仍为红灯，存在闯红灯风险。

(3) 预警关联驾驶建议规则

逻辑描述：

$$\forall v ((\text{RedLightWarnTriggered1}(v) \vee \text{RedLightWarnTriggered2}(v)) \wedge \text{IsVehicle}(v) \Rightarrow \text{RecommendAction}(v, \text{"Brake"}) \wedge \text{RecommendDecel}(v, \text{decel_val}))$$

代码实现：

#预警关联驾驶建议规则

rule RecommendAction(v, "Brake") :-

(RedLightWarnTriggered1(v) | RedLightWarnTriggered2(v)) & IsVehicle(v);

rule RecommendDecel(v, decel_val) :-

(RedLightWarnTriggered1(v) | RedLightWarnTriggered2(v)) & IsVehicle(v);

对触发闯红灯预警的车辆v，若车辆实例有效，则推荐执行刹车“Brake”动作，并输出对应的推荐减速度参数decel_val。

6.4 逻辑推理

假设当前信号灯状态为红色，且“车辆匀速到达停止线时间<红灯剩余时间”，推理过程如下：

(1) 谓词实例化验证：确认IsVehicle(v)、IsTrafficLight(tl)、TrafficLightState(tl, "red")、

TrafficLight(tl, "normal")等基础谓词均为真。

(2) 关键参数计算：通过HasSpeed(v, s)、HasDistanceToStopLine(v, d)，计算车辆到达停止线时间，且“车辆匀速到达停止线时间<红灯剩余时间”，满足时间条件。

(3) 规则匹配判定：满足RedLightWarnTriggered1(v)或者RedLightWarnTriggered2(v)的所有前提，且匹配RecommendBrake(v)规则。

(4) 结论推导：RedLightWarnTriggered1(v)、RedLightWarnTriggered2(v)确定车辆v触发闯红灯预警。RecommendAction(v, "Brake") $\wedge$ RecommendDecel(v, decel_val)，推荐车辆v执行刹车动作，并关联减速度变量decel_val。

中国智能交通产业联盟

中国智能交通产业联盟

标准

自主式交通系统 交通语义表示语言 第2部分：语法规范

T/ITS 0293.2-XXXX

北京市海淀区西土城路8号（100088）

中国智能交通产业联盟印刷

网址：<http://www.c-its.org.cn>

2025年X月第一版 2025年X月第一次印刷