1 Stream Reasoning Workshop 2019 - Challenge Track

2 Introduction

The idea of the Stream Reasoning Challenge (SRC) originated in a first session held at the SR Workshop 2018, where several groups worked out a possible formats for comparing and evaluating stream reasoning/processing tools.

One conclusion, worded by Boris and Jacobo, was that our community misses an unified agreement on the used formalism, languages, and standards that captures stream reasoning/processing. It is too early to conduct a benchmark competition that compares the performance of existing tool, hence a "model and solve" competition would be more suitable.

A "model and solve" challenge aims at evaluating the modelling skills independent of the formalism/language. For this, we provide the two given scenario, cooperative intelligence transport systems (C-ITS) and social media streams with one or more tasks to solve. At the end of the challenge, we aim at evaluating the development effort, easyness of use, and originality of the solutions.

Importantly, the initial plan was to conduct the full SRC at the Stream Reasoning Workshop 2019. However, due to several open points and time limitation, we plan to finalize a clear format for challenge, and conduct the actual SRC on a future date (tbd).

2.1 Programm for April, 16th, 2019

- 09:00 - 09:20 *Danh Le Phuoc*: Welcome and overview
- 09:20 - 09:50 *Daniel Jakobsson*: Invited talk
- 09:50 - 10:30 *Patrik Schneider*: Follow up and progress
- 10:30 - 10:45 Q&A with discussion
- 10:45 - 11:00 Coffee break
- 11:00 - 12:00 Brainstorming, finalization, and next steps

2.2 Programm Details

The first program point will be the keynote of Daniel Jakobsson with the aim of introducing more scenarios and data sources. The second program point will be a progress report on the SRC, which includes the outline of two scenarios and possible tasks that could be solved in it. Further, we will suggest possible rules for the SRC and the means for its evaluation/scorings, as well as possible evaluation platforms. We then will start a hands-on meeting, where we discuss the above topics. Of interest will be (a) which scenarios should be used; (b)

define the set of task that should be solved; and (c) agree on a framework on how the model challenge should be conducted, which includes also the evaluation platform.

3 Problem Description, Scenario, Tasks

As mentioned, the format will be a "model and solve" challenge, where the modelling skills independent of the formalism/language are evaluated. We outline the two combined scenarios, introduce the tasks, and outline the rules.

3.1 Scenarios

C-ITS. This C-ITS scenario is in the area of cooperative intelligent transportation systems (C-ITS), where we are able observe and analyze streams of vehicle movements and traffic light signal phases. Additionally, we will include unexpected events (triggering by us) such as vehicle breakdowns.

In this scenario we can group the tasks to be tackled into:

- Gathering traffic statistics, e.g., counting the number of vehicles passing;
- Event detection, e.g. detecting, accidents or traffic jams;
- Diagnosis, e.g., finding the cause for a traffic jam;
- Motion planning, e.g., routing the vehicles optimally through the network.

The difficulty of this scenarios originates from spatio-temporal nature of the traffic data, the possible hight velocity and volume, and as well the rather complex domain model.

Social Network Stream. The Social Network Stream provides social stream data generated by users connected to a social network. The stream data is provided by a data generator of LSBench that is built on top of Social Network Benchmark of Linked Data Benchmark Council. The data generator emulates the data distribution on social network connections among users, their posted contents and their location data. Following the diagram of the data schema in LSBench.

Combined Scenario. Both scenario are integrated into a transport social media scenarios, where the data of two above scenarios are correlated and could be combined. In particular, users in social network can be pedestrians or owners of the vehicles or users taking a ride-sharing services. Besides, the events relevant to traffic (e.g. music events, acidents, etc) can be extracted from messages/posts in a social media platformt, e.g. twitter. The combination allows us to add contextual information to the vehicle movements and variable locations to the messages. One task could include to distinguish different causes for a traffic jam, i.e., the traffic jam could be caused by external causes, e.g. a concert, or internal causes related to an overload of the road network.

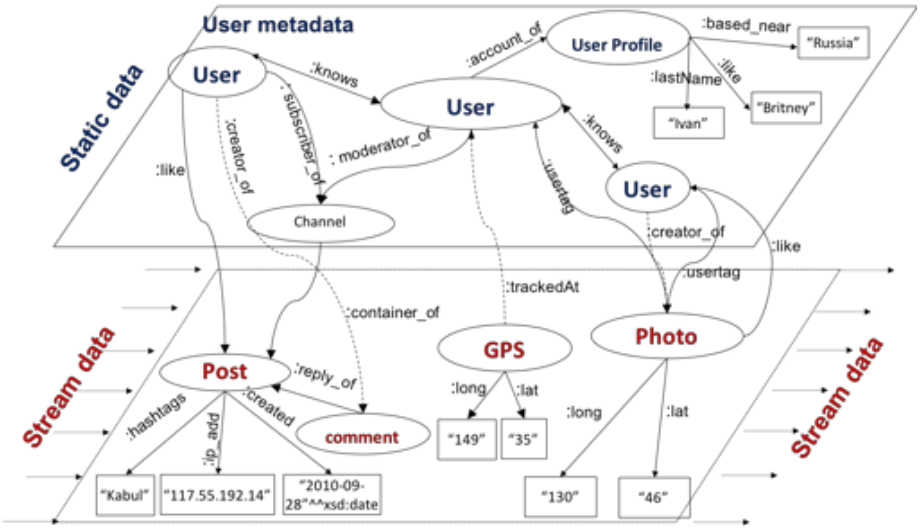

Figure 1: Linked Social Network

3.2 Tasks

Following, we present three fixed tasks, which are extended with user submitted task.

Task 1 (C-ITS). Traffic jam detection...

Task 2 (Social Media). To be defined...

Task 3 (Combined). External causes, e.g. a concert, versus internal causes related to an overload of the road network...

4 Rules and Evaluation

We introduce the following rules (open for discussion):

(1) The organizers select a number of problem tasks, which aree selected from the given (see above) and user-submitted tasks. A set of test streams for each task is predefined, expressed in possible instance input formats Datalog and RDF.

(2) The organizers provide a set of tools (list below), however, users are allow to bring their own tools, or even simply use scripts.

(3) For each task, teams are allowed to use a specific solver (or a solving script) and a problem encoding.

(4) Teams can either be assembled before the conference, or are on-demand set-up at the begin of the challenge.

(5) Solutions should be presented at the end of the competition.

After the solution presentation, a jury selected from the workshop participant give scores according on the criteria:

- Development effort,
- Operability and easyness of use,
- Problem coverage,
- Originality of the solutions.

5 Data, Platform, Systems

5.1 ITS Data

The provided data is based on a realistic traffic simulation of four intersections in a #-shaped layout. The scenario environment was developed with the microscopic traffic simulation *SUMO* or *PTV VISSIM* that allows us to simulate realistic driving behavior and signal phases. The structure of the intersection, driving patterns and signal phases are fixed, but we have adapted the traffic densities by light, medium, and heavy traffic.

We extracted the actual state of each simulation step, allowing us to replay the simulation from the logs.Throughput of the replayed streams can be increased by replaying the simulation with 5ms, 10ms, 50ms, and 100ms (real-time speed of traffic) delay.

Static Knowledge: As shown in the figure, each intersection connects four roads with two incoming and outgoing lanes for each road. For the two incoming lane one signal group is assigned. Further, for each lane we define is geometrical extension (a polygon) and the connection from in incoming to outgoing lanes. As an example, lane "i100_l1" in our model is described in Datalog as:

- `intersection(i100).`
- `mapLaneIn(i100_l1).`
- `isPartOf(i100_l8, i100).`
- `hasGeo(i100_l1, "POLY((0 1, 2 1, 2 3, 0 3, 0 1))").`
- `connected(i100_l1, i100_l3).`

The four signal groups are managed by one traffic light controller, which assigns red (stop) and green (go) states to signal phases. The signal phases for each signal group are encoded in a signal plan, where the green/red split of a full phase length is defined.

We provide two separate schemas for this, one is the ITS ontology (http://www.kr.tuwien.ac.at/research/projects/loctrafflog/ekaw2018/) and the

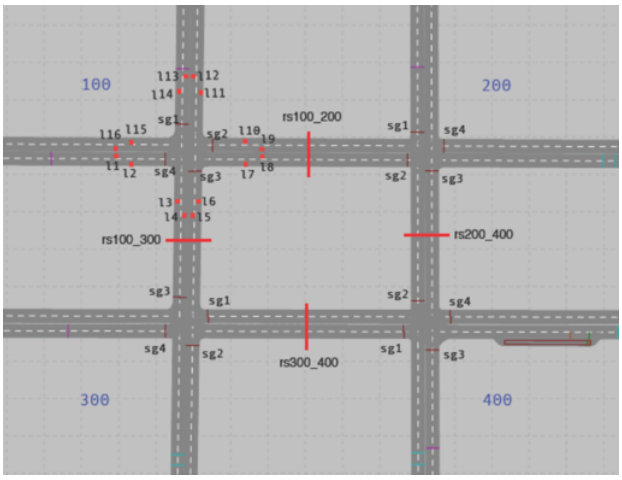

Figure 2: #-shaped intersections

other is simplified version of the ontology encoded in Datalog.

Data Streams: The microscopic model of the traffic is represented by four data streams extracted from simulation in each step. They include vehicle speed and position, heading, as well as signal phase states. each data point has an vehicle ID *id* (beside signal group id *group*, and a time stamp *ts* assigned:

- `speed(id, speed, ts)`
- `pos(id, point(x,y), ts)`
- `heading(id, angle, ts)`
- `signalPhase(group, state, ts)`

PS: We could add a weather stream...

5.2 Social Media Data

PS: Danh, please complete

5.3 Evaluation Platform

PS: Needs to be clarified

5.4 Systems

Besides the user-provided system, we already provide a selection of systems that could be used in the challenge:

- CQELS
- C-SPARQL or YASPER
- Hexlite
- RDFox
- Vlog

PS: Need to check which each developer, to find out if we are allowed to us the tool