TOPOLOGICAL ANOMALY DETECTION IN AUTOMOTIVE SIMULATOR MAPS

M. Barsi [1,] *, A. Barsi [1]

[1] Dept. Photogrammetry and Geoinformatics, Budapest University of Technology and Economics, Hungary - barsimark@gmail.com, barsi.arpad@emk.bme.hu

Commission IV, WG 7

KEYWORDS: autonomous driving, environmental model, HD map, topology description

ABSTRACT:

Autonomous driving went through numerous significant improvements over the past couple of years, including driver assistants that are already capable of executing an increasing number of complex tasks without the need for any human intervention. As a result of these changes, manufacturers are relying more and more on fast, cheap, and often better-quality simulations over real-world tests. To create these environments, the real world needs to be transformed to a digital, high-definition model. HD maps – for example, the XML-based, hierarchic OpenDRIVE format – aim to serve this purpose.
The most important element of any realistic map format is the ability to check connectivity on the map in a convenient way, hence the need for topology. In HD maps, the description of junctions poses a significant challenge to the designers of the format, since they are essential yet complex topological elements. The representation of these junctions is still in progress, however, according to our analysis, the use of the current tools in OpenDRIVE can result in anomalies in the map.
In the most recent release of OpenDRIVE (version 1.7), road-road and lane-lane connections are described using links consisting of a predecessor and a successor. These however, has to be described multiple times when the junction tag is used, resulting in duplicates in the model which can be easily exploited. Our proposed solution for this issue is the elimination of the junction tag, which not only gets rid of the anomalies without any loss of information, but it also significantly reduces the size of the model. In this paper, a detailed explanation is provided of this issue and the proposed solution with examples using OpenDRIVE models.

1. INTRODUCTION

It became clear recently that automotive – especially autonomous – vehicle development can be made extremely efficient and cheap by using computer simulations. (Anderson et al., 2016; Kang et al., 2019; KPMG, 2015) However, in these efforts, various simulators can only provide acceptable results if the conditions tested are realistic. Therefore, increasing reliance is being placed on field surveys to describe road infrastructure. For example, mobile mapping technology collects proper data suitable to define roadway model elements specified by the standard ASAM OpenDRIVE format. (Barsi et al., 2019; Leica, n.d.; Riegl, n.d.) In this effort, not only the pure geometric content, but also the correctness of the element connections must be ensured. Without this, many simulators are unable to perform proper analysis, as vehicles can become stuck at the boundaries of incorrectly defined road segments. The first step in correcting these errors is checking the topological relations, making it possible to fix them later with as much automation as possible. In this paper, we present the mechanism developed for this automatic testing.

2. THE OPENDRIVE FORMAT

OpenDRIVE is a standardized format to describe static road network elements for simulation purposes. It belongs to the ASAM standard family designed to support automotive developments. The implemented building blocks follow a well-defined hierarchy where the main components are header, road, junction, junction group, station, and controller objects. Dynamic content is not supported.

The format is capable of handling valuable information about roads in a complex structure. The road description is based on a reference line, which creates a specific local coordinate system in which road-related objects need to be defined. The basic geometric primitives for defining the reference line are straight lines (*line*), pure circular arcs (*arc*), clothoid transition curves (*spiral*), cubic polynomials (*poly3*) or 3rd order parametric polynomial curves (*paramPoly3*). Real roads are composed of these primitives under the planView geometry tag.
Lanes, elevation information and roadside objects are all specified relative to the reference line, in its local coordinate system. The standard is capable of managing different road types, including driving, border, parking, biking, sidewalk etc. Roads are built of segments connected together by linking tags.
OpenDRIVE models are stored in a custom extension of the XML format, called XODR. These models can even be zipped to store them with higher efficiency (XODRZ file format). The most recent version is v1.7.0. (ASAM e.V., 2022)
The main application of the OpenDRIVE format is in automotive simulators, where it is supported as their native road environment description style. ("CarMaker | IPG Automotive," n.d., "Home - dSPACE," n.d., "PreScan | TASS International," n.d., "Simulation Technologies - avl.com," n.d., "VTD - VIRES Virtual Test Drive," n.d.)

3. TOPOLOGY IN OPENDRIVE

The realizations of links in the models are on two topological levels: on road and lane levels. Topology can be divided into two categories, line topology and polygon topology. The latter is only valid on lane levels.

* Corresponding author

The road level topology defines the relationship between two connected roads represented by their reference lines. (Fig.1).

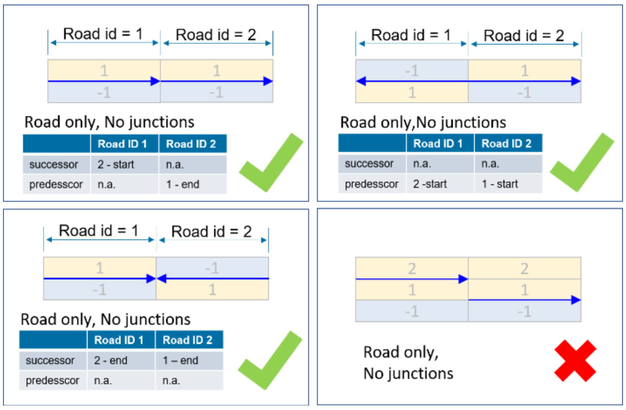
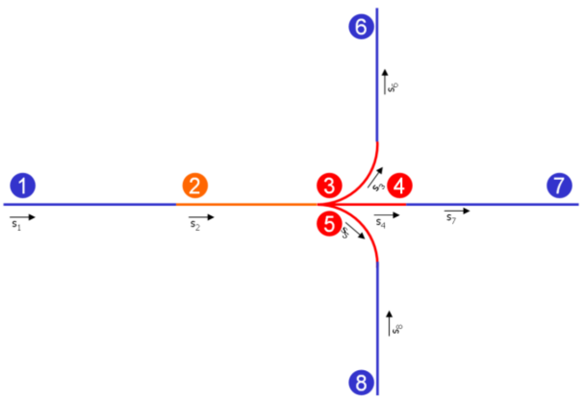


Figure 1. Road linkage in the OpenDRIVE standard (ASAM e.V., 2022)

There is a **predecessor** (P) representing a prior, and a **successor** (S) representing a following connecting road. The most important point when it comes to this kind of topology is the point, where the reference lines of two adjacent roads are met: this is the **contact point** (CP).
On the lane level, connectivity is specified as in Fig.2.

Figure 2. Lane linkage in OpenDRIVE standard (ASAM e.V., 2022)

Lanes are numbered as follows: the reference line has an id of 0, positive numbers are given to the left side and negative ones to the right side; in ascending order from the reference line in terms of their absolute value.
The connection between exactly two roads is clear, so the usual linkage information is sufficient. Having more connecting roads, the relationship becomes ambiguous, so a **junction** tag is required. This is built from connections containing the relations between roads and their represented lanes (called laneLinks). (ASAM e.V., 2022) Fig. 3 shows how a triple connectivity causes ambiguity in successor descriptions.


a) branching with 3 connections in a junction

Road	Predecessor	Successor
1	-	2
2	1	ambiguous
3	2	6
4	2	7
5	2	8
6	3	-
7	4	-
8	-	5

b) predecessors and successors in a table form

Figure 3. Ambiguous situation with branching roads in a junction (ASAM e.V., 2022)

4. METHODOLOGY

Considering the standard, the possible interrelations between two adjacent roads can be the following (Fig.4):

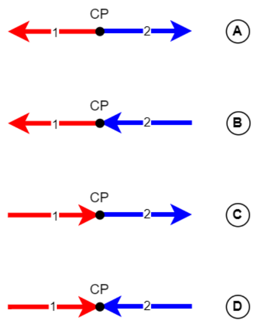


Figure 4. Possible connections between two connecting roads

The relations stated above can be expressed by basic topologic rules (Table 1):

Rule	Road 1	Road 2
A	predecessor: Road 2 CP: start	predecessor: Road 1 CP: start
B	predecessor: Road 2 CP: end	successor: Road 1 CP: start
C	successor: Road 2 CP: start	predecessor: Road 1 CP: end
D	successor: Road 2 CP: end	successor: Road 1 CP: end

Table 1. Basic topologic rules for roads

Following this rule set, one can evaluate the situation: the predecessor of Road1 is Road2 with CP being an endpoint, and Road1 is the successor of Road2 with CP as endpoint as well (incorrect version of case B). This configuration is not in the rule set, so it is clearly a false case. Similarly, if Road1's successor is Road2, Road2's predecessor is Road1, and CP is a starting point in both cases, then this second scenario is also incorrect.
The given topological rules are inference rules (Wikipedia, n.d.), so that their inverse can also be interpreted, resulting in additional flexibility and more usability. With the aforementioned rule set, one identified cell indicates the status of the connecting road. Therefore, the proper use of the contact point is inevitable in this analysis.

The basic rule set can be extended with four additional ones (Table 2):

Rule	Road 1
E	predecessor: - CP: -
F	successor: - CP: -
G	predecessor: junction CP: -
H	successor: junction CP: -

Table 2. Additional topologic rules

Having no contact point is something that all the newly added rules have in common. For example, in case E, there is no previous road, or in case F, there is no following road. Both topological cases are possible and allowed in the model. Moreover, the standard specifies only unequivocal connections as links. In any other case, junctions must be used. Case G and H are both representing these latter options.
Lane level topology rules are similar to the four basic road topology rules, but without contact points (Table 3).

Rule	Lane 1	Lane 2
I	predecessor: Road 2	predecessor: Road 1
J	predecessor: Road 2	successor: Road 1
K	successor: Road 2	predecessor: Road 1
L	successor: Road 2	successor: Road 1

Table 3. Basic lane level topologic rules

The table above specifies the lane connections in the relation of Road1 Lane1 and Road2 Lane2.
The roads being part of a junction have connectivity information coded under the laneLink tag which stores a **from** and a **to** identifiers; the from lane connects to the *to* lane. However, this information is already recorded in the lane description with a link subtag containing a predecessor and/or a successor. These latter P and S tags point to the appropriate lane identifiers. Due to the fact that lane links inherit the knowledge of road identifiers, lane connections are available by tracing through this chain: road → lane → link.
As a result of OpenDRIVE models being physically stored in an XML format, parsing can be implemented in various high-level programming languages. Several Matlab-routines were developed to collect the necessary topology information from these XODR-files by storing them in complex tables for roads, lanes and junctions. Rule checks were also implemented based on these tables, generating error messages or warnings in suspicious cases. The last phase is a human crosscheck, where the marked elements are inspected in multiple special software. For this, Mathworks RoadRunner (MathWorks, n.d.) and Vires (Vires, 2022; “VTD - VIRES Virtual Test Drive,” n.d.) odrViewer environments were used.

5. RESULTS

The developed methodology – with the software produced – is demonstrated first in a simple synthetic example, then in one real-world model. The prior was created manually (in order to have the simplest road network in a strict structure), while the complex example is imported from field survey results.

The first example can be seen in Fig.5.

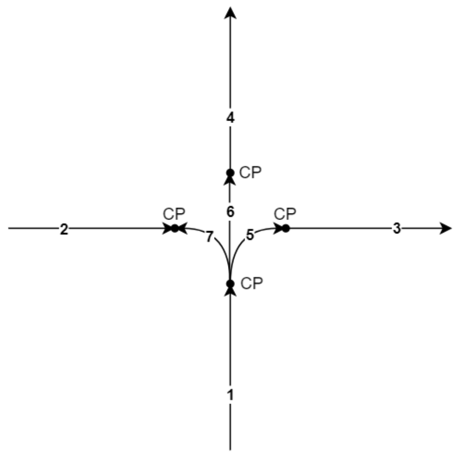


Figure 5. Road reference lines in the simplest example

The scenario contains two crossing roads with two lanes each. Only one road (#1) is connected to all of its possible neighbors (#2, #3, #4). Road #5, #6 and #7 have Road1 as their predecessors. Applying the inverse rules, Road1 has multiple successors (namely the mentioned #5, #6, and #7), which is ambiguous, even though these statements are all true. (The rules are also valid for the reverse case, where connecting roads merge, not split.)
In this situation, the junction object contains the information about one incoming road (#1) with three connections, each with a connecting road. If the road splits, the connecting roads have a common contact point as their starting points, whereas in the case of road merging, the single contact point will act as an endpoint. With the presented mechanism, junctions are reconstructed using road-level topology.
To understand the lane level connections, analysis of all potential lanes is required (Fig.6).
Lane -1 of Road1 is connected to lane -1 of Road5. This is the exact same connectivity mapped in the junction tag (Fig. 7). This also applies to the Road1-Road6 and the Road1-Road7 pairs.
It is easy to see from this XML-snippet that junctions can be tested by applying the defined rules and inference chaining.
The second illustration for the developed methodology is taken from real life: Atlatec (GmbH, 2022) has surveyed the road called ‘Südtangente’ in Karlsruhe, Germany, which is a 2+2 lane highway. Fig. 8a shows an intersection, while Fig. 8b demonstrates an overpass in 3D perspective view.
The model contains 114 roads, 886 lanes, and 189 junctions. (These statistics were derived by the developed topology analysis tool.) The model is stored in a 2MB ASCII XODR file containing 22127 lines.
The first step in the topology analysis was parsing the XML-file, resulting in road, lane and junction tables. Then the presented rules were applied on them, giving a rapid overview of the predecessors and successors, as well as potential issues with them. A connectivity diagram was derived from this information (Fig. 9).
Using this newly-found knowledge, it was proven, that there is only a single place, where topology anomaly occurs in the model. Additional analysis found that it was a predecessor overlap on the highway, as well as the IDs of the roads and lanes affected.

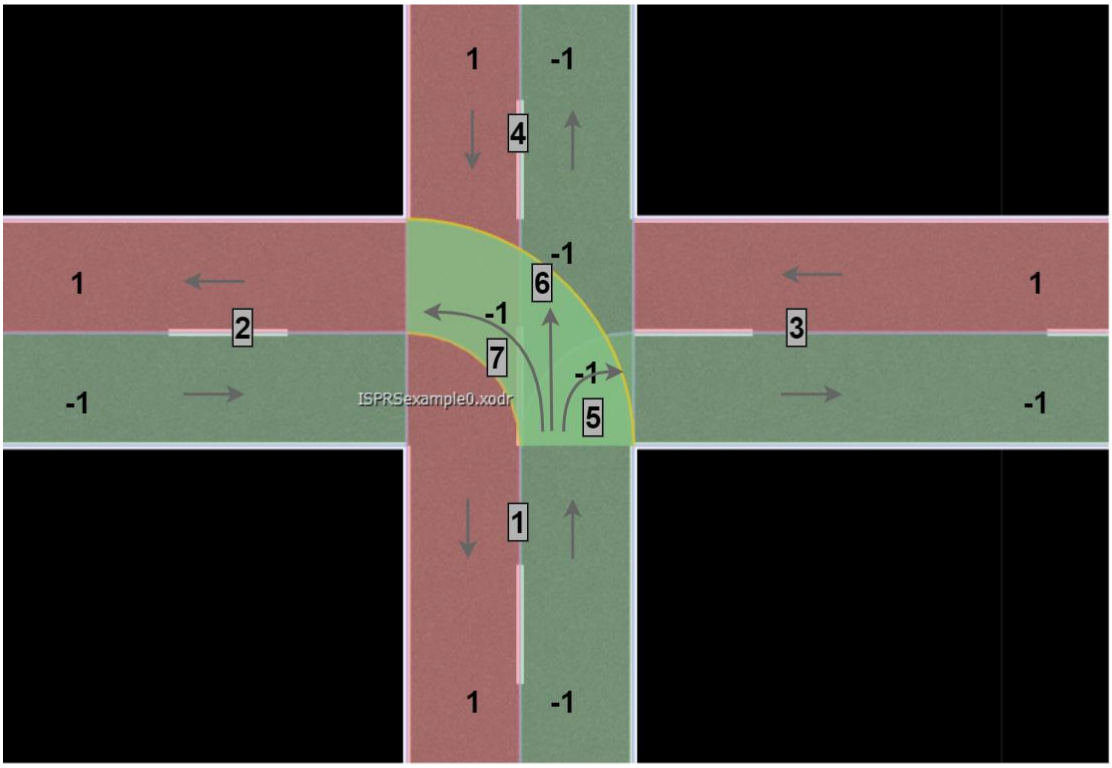
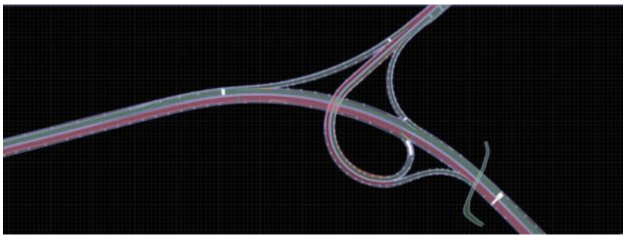


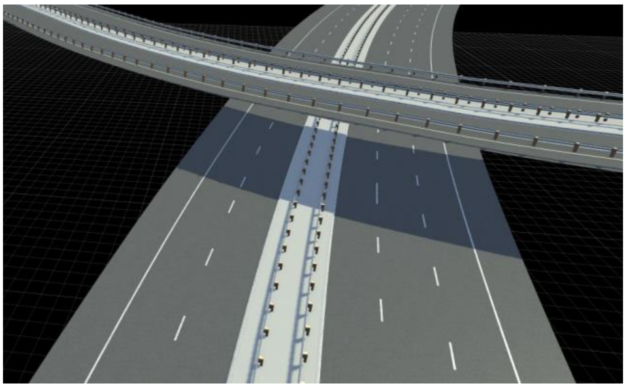
Figure 6. Road crossing of Figure 4 with lanes

```xml
<road name="Road 1" id="1">
    <link>
        <!--There is no predecessor, because it is the first road.-->
        <!--There is no successor given, because it is ambiguous. Deducted: Road 5 should be.-->
    </link>
    <lanes>
        <laneSection>
            <right>
                <lane id="-1">
                    <link>
                        <!--There is no predecessor, because it is the first lane.-->
                        <!--There is no successor given, because it is ambiguous. Deducted: Lane -1 should be.-->
                    </link>
                </lane>
            </right>
        </laneSection>
    </lanes>
</road>

<road name="Road 5" id="5">
    <link>
        <predecessor elementType="road" elementId="1" contactPoint="end" />
        <successor elementType="road" elementId="3" contactPoint="start" />
    </link>
    <lanes>
        <laneSection>
            <right>
                <lane id="-1">
                    <link>
                        <predecessor id="-1" /> <!--The lane -1 of Road 5 is connected to lane -1 of Road 1. -->
                        <successor id="-1" />   <!--The lane -1 of Road 5 is connected to lane -1 of Road 3. -->
                    </link>
                </lane>
            </right>
        </laneSection>
    </lanes>
</road>

<junction name="Junction 100" id="100">
    <connection id="0" incomingRoad="1" connectingRoad="5" contactPoint="start"> <!--Road predecessor info.-->
        <laneLink from="-1" to="-1"/> <!--Lane predecessor info.-->
    </connection>
</junction>
```

Figure 7. XML code-snippet for Road1 and Road5 as well as their junction reference



a) model with an intersection



b) overpass with rendered 3D visualization, generated from the OpenDRIVE model

Figure 8. Real-world model on Südtangente near Karlsruhe, Germany © Atlatec

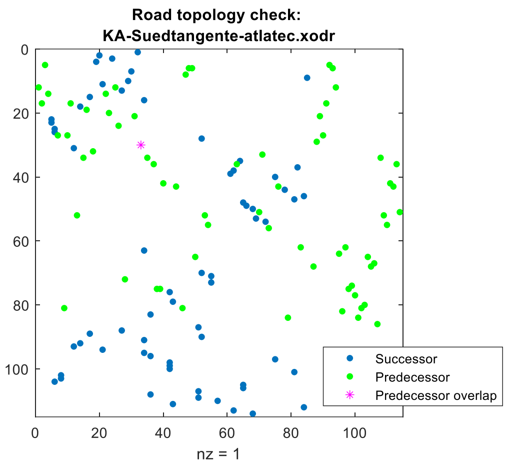


Figure 9. The visual result of the topology check

The complete topological inspection of the Karlsruhe-model took only a couple of seconds on a medium category laptop.
The topology analyzing tool was implemented in Matlab. (MathWorks, n.d.) Visualization was performed with RoadRunner, a software formerly owned by VectorZero, acquired by MathWorks several years ago.

6. CONCLUSION

The research focus was set on supporting the creation of high-quality road description models for automotive simulations. To reach this goal, we have experimented with a range of different simulators and have found that an elevated sensitivity on the correctness of the topology is expected. Therefore, the emphasis of the presented work was laid on topology testing.

The ASAM OpenDRIVE format has excellent tools to store relevant topologic information, for both roads and lanes. In ambiguous cases (a road connecting to several others), the standard prescribes the insertion of a junction. Although the junction object has a clear specification in the standard, we have noticed and later proved that roads and junctions are not independent elements. By using adequate topological rules, it is possible to derive all information stored in the junctions by parsing the road and lane linkage data. Considering the consistency of a model, it must be carefully tested, otherwise hidden contradictions could be included in the model.
An even more radical finding is that the junction tags can simply be ignored, as if strict topological rules are applied, the content of them can fully be derived from the road descriptions. With this reconstruction, not only the consistency of the entire model can be guaranteed, but a slightly more compact storage is also available by omitting junctions and their references from road objects.
Further lane level topology checks considering polygon topology requires a joint geometric-topologic test methodology, as potential gaps and overlaps can occur in the models. We might need to consider lane width and geometric location in every given scenario to come up with a system capable of testing all the aspects of OpenDRIVE models.

REFERENCES

Anderson, J., Kalra, N., Stanley, K., Sorensen, P., Samaras, C., Oluwatola, O., 2016. Autonomous Vehicle Technology: A Guide for Policymakers. Auton. Veh. Technol. A Guid. Policymakers. https://doi.org/10.7249/RR443-2

ASAM e.V., 2022. ASAM OpenDRIVE® [WWW Document]. URL https://www.asam.net/standards/detail/opendrive/ (accessed 3.25.22).

Barsi, A., Csepinszky, A., Krausz, N., Neuberger, H., Poto, V., Tihanyi, V., 2019. Environmental data delivery for automotive simulations by laser scanning. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-2/W16, 41–45. https://doi.org/10.5194/isprs-archives-XLII-2-W16-41-2019

CarMaker | IPG Automotive [WWW Document], n.d. URL https://ipg-automotive.com/products-services/simulation-software/carmaker/ (accessed 9.26.19).

GmbH, A., 2022. Atlatec [WWW Document]. URL https://atlatec.de/en/

Home - dSPACE [WWW Document], n.d. URL https://www.dspace.com/en/pub/home.cfm (accessed 9.26.19).

Kang, Y., Yin, H., Berger, C., 2019. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. IEEE Trans. Intell. Veh. 4, 171–185. https://doi.org/10.1109/TIV.2018.2886678

KPMG, 2015. KPMG report connected and autonomous-vehicles - Google-keresés [WWW Document]. URL https://www.google.com/search?q=KPMG+report+connected+and+autonomous-vehicles&oq=KPMG+report+connected+and+autonomous-vehicles&aqs=chrome..69i57j33i22i29i30.4821j0j15&sourceid=chrome&ie=UTF-8 (accessed 3.27.22).

Leica, n.d. Leica Pegasus:Two Mobile Sensor Platform | Leica Geosystems [WWW Document]. URL https://leica-geosystems.com/hu-hu/products/mobile-mapping-systems/capture-platforms/leica-pegasus_two (accessed

4.2.22).
MathWorks, n.d. RoadRunner - MATLAB & Simulink [WWW Document]. URL https://www.mathworks.com/products/roadrunner.html (accessed 4.2.22a).
MathWorks, n.d. MATLAB - MathWorks - MATLAB & Simulink [WWW Document]. URL https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab (accessed 4.2.22b).
PreScan | TASS International [WWW Document], n.d. URL https://tass.plm.automation.siemens.com/prescan (accessed 9.26.19).
Riegl, n.d. RIEGL - Mobile Scanning [WWW Document]. URL http://www.riegl.com/nc/products/mobile-scanning/ (accessed 4.2.22).
Simulation Technologies - avl.com [WWW Document], n.d. URL https://www.avl.com/hu/web/guest/simulation (accessed 9.26.19).
Vires, 2022. Vires VTD [WWW Document]. URL https://vires.mscsoftware.com/ (accessed 4.2.22).
VTD - VIRES Virtual Test Drive [WWW Document], n.d. URL https://vires.com/vtd-vires-virtual-test-drive/ (accessed 9.26.19).
Wikipedia, n.d. Inference engine [WWW Document]. URL https://en.wikipedia.org/wiki/Inference_engine (accessed 4.2.22).