Verification and Validation of Semantic Annotations

Oleksandra Panasiuk(✉), Omar Holzknecht(✉), Umutcan Şimşek(✉),
Elias Kärle, and Dieter Fensel

University of Innsbruck, Technikerstrasse 21a, 6020 Innsbruck, Austria
{oleksandra.panasiuk,omar.holzknecht,umutcan.simsek,elias.karle,
dieter.fensel}@sti2.at

Abstract. In this paper, we propose a framework to perform verification and validation of semantically annotated data. The annotations, extracted from websites, are verified against the schema.org vocabulary and Domain Specifications to ensure the syntactic correctness and completeness of the annotations. The Domain Specifications allow for checking of the compliance of annotations against corresponding domain-specific constraints. The validation mechanism will detect errors and inconsistencies between the content of the analyzed schema.org annotations and the content of the web pages where the annotations were found.

Keywords: Verification · Validation · Semantic annotation · Schema.org

1 Introduction

The introduction of the Semantic Web [3] changed the way content, data and services are published and consumed online fundamentally. For the first time, data in websites becomes not only machine-readable, but also machine understandable and interpretable. The semantic description of resources is driving the development of a new generation of applications, like intelligent personal assistants and chatbots, and the development of knowledge graphs and artificial intelligence applications. The use of semantic annotations was accelerated by the introduction of schema.org [8]. Schema.org was launched by the search engines Bing, Google, Yahoo! and Yandex in 2011. It has since become a de-facto standard for annotating data on the web [15]. The schema.org vocabulary, serialized with Microdata, RDFa, or JSON-LD, is used to mark up website content. Schema.org is the most widespread vocabulary on the web, and is used on more than a quarter of web pages [9,14].

Even though studies have shown that the amount of semantically annotated websites are growing rapidly, there are still shortcomings when it comes to the quality of annotations [12,17]. Also the analyses in [1,10] underline the inconsistencies and syntactic and semantic errors in semantic annotations. The lack

N. Bjørner et al. (Eds.): PSI 2019, LNCS 11964, pp. 223–231, 2019.
https://doi.org/10.1007/978-3-030-37487-7_19

of completeness and correctness of the semantic annotations makes content unreachable for automated agents, causes incorrect appearances in knowledge graphs and search results, or makes crawling and reasoning less effective for building applications on top of semantic annotations. These errors may be caused by missing guidelines, insufficient expertise and technical or human errors. Data quality is a critical aspect for efficient knowledge representation and processing. Therefore, it is important to define methods and techniques for semantic data verification and validation, and to develop tools which will make this process efficient, tangible and understandable, also for non-technical users.

In this paper, we extend our previous work [21], where we introduced a Domain Specification, and present an approach for verification and validation of semantic annotations. A Domain Specification (DS) is a design pattern for semantic annotations; an extended subset of types, properties, and ranges from schema.org. The semantify.it Evaluator[1] is a developed tool that allows the verification and validation of schema.org annotations which are collected from web pages. Those annotations can be verified against the schema.org vocabulary and Domain Specifications. The verification against Domain Specifications allows for the checking of the compliance of annotations against corresponding domain-specific constraints. The validation approach extends the functionality of the tool by detecting the consistency errors between semantic annotations and annotated content.

The remainder of this paper is structured as follows: Sect. 2 describes the verification approach of semantic annotations. Section 3 describes the validation approach. Section 4 concludes our work and describes future work.

2 Verification

In this section we discuss the verification process of semantic annotations according to schema.org and Domain Specifications. The section is structured as follows: Sect. 2.1 gives the definition of the semantic annotation verification, Sect. 2.2 describes related work, Sect. 2.3 discusses our approach, and Sect. 2.4 describes the evaluation method.

2.1 Definition

The verification process of semantic annotations consists of two parts, namely, (I) checking the conformance with the schema.org vocabulary, and (II) checking the compliance with an appropriate Domain Specification. While the first verification step ensures that the annotation uses proper vocabulary terms defined in schema.org and its extensions, the second step ensures that the annotation is in compliance with the domain-specific constraints defined in a corresponding DS.

[1] https://semantify.it/evaluator.

2.2 Related Work

In this section, we refer to the existing approaches and tools to verify structured data. There are tools for verifying schema.org annotations, such as the Google Structured Data Testing tool[2], the Google Email Markup Tester[3], the Yandex Structured Data Validator[4], and the Bing Markup Validator[5]. They verify annotations of web pages that use Microdata, Microformats, RDFa, or JSON-LD as markup formats against schema.org. But these tools do not provide the check of completeness and correctness. For example, they can allow one to have empty range values, redundancy of information, or semantic consistency issues (e.g. the end day of the event is earlier than the start day). In [7] SPARQL and SPIN are used for constraint formulation and data quality check. The use of SPARQL and SPIN query template sets allows the identification of syntax errors, missing values, unique value violations, out of range values, and functional dependency violations. The Shape Expression (ShEx) definition language [20] allows RDF verification[6] through the declaration of constraints. In [4] the authors define a schema formalism for describing the topology of an RDF graph that uses regular bag expressions (RBEs) to define constraints. In [5] the authors described the semantics of Shapes Schemas for RDF, and presented two algorithms for the verification of an RDF graph against a Shapes Schema. The Shapes Constraint Language[7] (SHACL) is a language for formulating structural constraints on RDF graphs. SHACL allows us to define constraints targeting specific nodes in a data graph based on their type, identifier, or a SPARQL query. The existing approaches can be adapted for our needs but not fully, as they are developed for RDF graph verification and not for schema.org annotations in particular.

2.3 Our Approach

To enable the verification of semantic annotations according to the schema.org vocabulary and to Domain Specifications, we developed a tool that executes a corresponding verification algorithm. This tool takes as inputs the schema.org annotation to verify and a DS that corresponds to the domain of the annotation. The outcome of this verification process is provided in a formalized, structured format, to enable the further machine processing of the verification result.

The verification algorithm consists of two parts, the first checks the general compliance of the input annotation with the schema.org vocabulary, while the latter checks the domain-specific compliance of the input annotation with the given Domain Specification. The following objectives are given for the conformity verification of the input annotation according to the schema.org vocabulary:

[2] https://search.google.com/structured-data/testing-tool/.

[3] https://www.google.com/webmasters/markup-tester/.

[4] https://webmaster.yandex.com/tools/microtest/.

[5] https://www.bing.com/toolbox/markup-validator.

[6] Authors use term "validation" in their paper due to content definition.

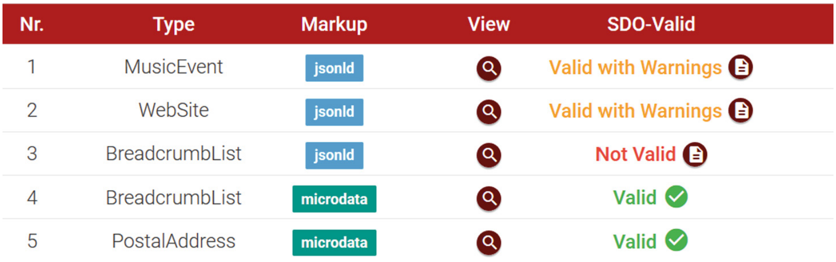
[7] https://www.w3.org/TR/shacl-ucr/.

1. The correct usage of serialization formats allowed by schema.org, hence RDFa, Microdata, or JSON-LD.
2. The correct usage of vocabulary terms from schema.org in the annotations, including types, properties, enumerations, and literals (data types).
3. The correct usage of vocabulary relationships from schema.org in the annotations, hence, the compliance with domain and range definitions for properties.

The domain-specific verification of the input annotation is enabled through the use of Domain Specifications[8], e.g. DSs for annotation of tourism domain and GeoData [18,19]. DSs have a standardized data model. This data model consists of the possible specification nodes with corresponding attributes that can be used to create a DS document (e.g. specification nodes for types, properties, ranges, etc.). A DS document is constructed by the recursive selection of these grammar nodes, which, as a result, form a specific syntax (structure) that has to be satisfied by the verified annotations [11]. Keywords in these specification nodes allow the definition of additional constraints (e.g. "multipleValuesAllowed" or "isOptional" for property nodes). In our approach, the verification algorithm has to ensure that the input annotation is in compliance with the domain-specific constraints defined by the input DS. In order to achieve this, the verification tool has to be able to understand the DS data model, the possible constraint definitions, and to check if verified annotations are in compliance with them.

2.4 Evaluation

We implement our approach in the semantify.it Evaluator[9]. The tool provides a verification report with detailed information about detected errors according to the schema.org vocabulary (see Fig. 1) and Domain Specifications (see Fig. 2).



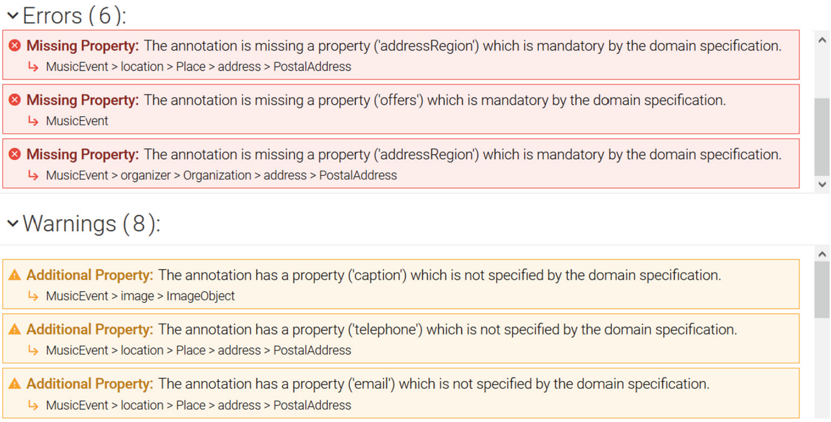
Nr.	Type	Markup	View	SDO-Valid
1	MusicEvent	jsonld	🔍	Valid with Warnings 🗎
2	WebSite	jsonld	🔍	Valid with Warnings 🗎
3	BreadcrumbList	jsonld	🔍	Not Valid 🗎
4	BreadcrumbList	microdata	🔍	Valid ✓
5	PostalAddress	microdata	🔍	Valid ✓

Fig. 1. Schema.org verification

Besides the verification result itself, the report includes details about the detected errors, e.g. error codes (ID of the error type), error titles, error severity

[8] List of available Domain Specifications: https://semantify.it/domainSpecifications/public.

[9] https://semantify.it/evaluator.



Errors (6):

⊗ **Missing Property:** The annotation is missing a property ('addressRegion') which is mandatory by the domain specification.
↳ MusicEvent > location > Place > address > PostalAddress

⊗ **Missing Property:** The annotation is missing a property ('offers') which is mandatory by the domain specification.
↳ MusicEvent

⊗ **Missing Property:** The annotation is missing a property ('addressRegion') which is mandatory by the domain specification.
↳ MusicEvent > organizer > Organization > address > PostalAddress

Warnings (8):

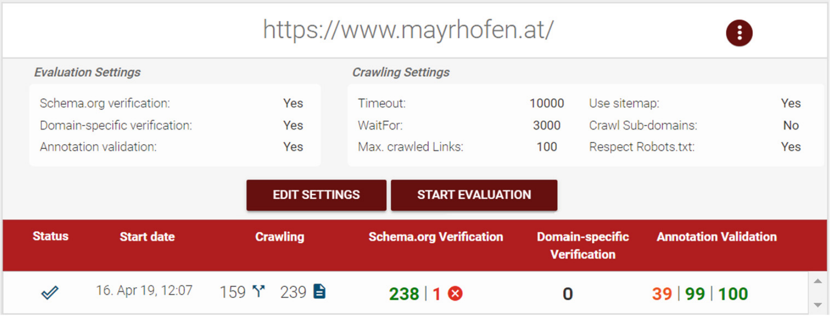
⚠ **Additional Property:** The annotation has a property ('caption') which is not specified by the domain specification.
↳ MusicEvent > image > ImageObject

⚠ **Additional Property:** The annotation has a property ('telephone') which is not specified by the domain specification.
↳ MusicEvent > location > Place > address > PostalAddress

⚠ **Additional Property:** The annotation has a property ('email') which is not specified by the domain specification.
↳ MusicEvent > location > Place > address > PostalAddress

Fig. 2. Domain specification verification. Verification report

levels, error paths (where within the annotation the error occurred), and textual descriptions of the errors. The implementation itself can be evaluated through unit tests in terms of a correct functionality (correctness) and the implementation of all possible constraint possibilities of the Domain Specification vocabulary (completeness). This can be achieved by comparing the structured representation of the result, namely the JSON file produced by the verification algorithm, which is used to generate a human-readable verification report for the user (see Fig. 3), with the expected verification report outcome specified in the test cases for predefined annotation-Domain Specification pairs.



https://www.mayrhofen.at/

Evaluation Settings

Schema.org verification: Yes
Domain-specific verification: Yes
Annotation validation: Yes

Crawling Settings

Timeout: 10000
WaitFor: 3000
Max. crawled Links: 100
Use sitemap: Yes
Crawl Sub-domains: No
Respect Robots.txt: Yes

Status	Start date	Crawling	Schema.org Verification	Domain-specific Verification	Annotation Validation
✓	16. Apr 19, 12:07	159 ⑂ 239 🗎	238 | 1 ⊗	0	39 | 99 | 100

Fig. 3. semantify.it Evaluator. Verification and validation report

A formal proof of the correctness and completeness of our implemented algorithm is rather straightforward given the simplicity of our current knowledge representation formalism. In our ongoing work[10], we develop a richer constraint language which will require more detailed analysis of these issues.

3 Validation

Search engines may penalize the publisher of structured data if their annotations include content that is invisible to users, and/or markup irrelevant or misleading content. These penalties may have negative effects on a website (e.g. bad position of the website in search results) or even lead to non-integration of the structured data (e.g. no generation of rich snippets). For example, annotations of the Destination Management Organizations (DMOs) usually include a list of offers. These offers must comply with offers which are described on the website, and all URLs contained in the annotations must match with the URLs in the content. Such issues can be detected through the validation of semantic annotations.

In this section, we discuss the validation process of semantic annotations and the proposed approach. The section is structured as follows: Sect. 3.1 gives the definition of the semantic annotation validation, Sect. 3.2 describes some related work, Sect. 3.3 discusses our approach, and Sect. 3.4 describes the evaluation method.

3.1 Definition

The validation of semantic annotations is the process of checking whether the content of a semantic annotation corresponds to the content of the web page that it represents, and if it is consistent with it. Semantic annotations should include the actual information of the web page, correct links, images and literal values without overlapping or redundancy.

3.2 Related Work

The incorrect representation of the structured data can make data unreachable for automated engines, cause an incorrect appearance in the search results, or make crawling and reasoning less effective for building applications on top of semantic data. The errors may be caused by not following recommended guidelines, e.g. structured data guidelines[11], insufficient expertise, technical or human errors (some of the issues can be detected by Google search console[12]), and/or annotations not being in accordance with the content of web pages, so-called "spammy structured markup"[13]. There is no direct literature related to the methods of detecting inconsistency between semantic annotations and content of web pages, but the problem of the content conformity restriction is also mentioned in [13].

[10] The paper is under double blind review and can't be revealed.

[11] https://developers.google.com/search/docs/guides/sd-policies.

[12] https://search.google.com/search-console/about.

[13] https://support.google.com/webmasters/answer/9044175?hl=en&visit_id=636862521420978682-2839371720&rd=1#spammy-structured-markup.

3.3 Our Approach

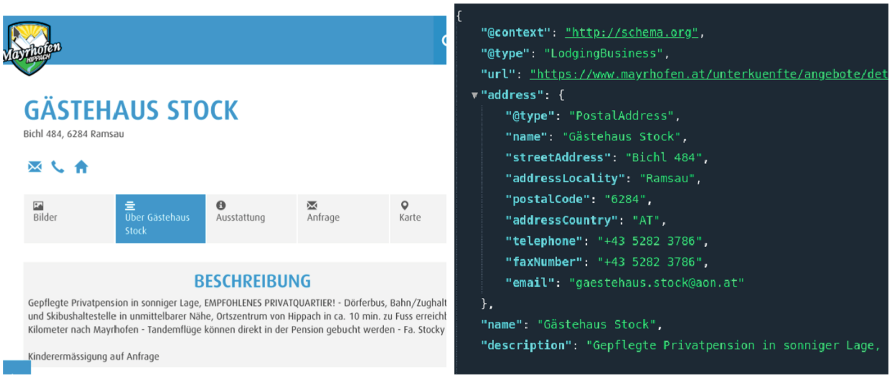
Since semantic annotations are created and published by different data providers or agencies in varying quantity and quality and using different assumptions, the validity of data should be prioritized to increase the quality of structured data. To solve the problem of detecting errors caused by inconsistencies between analyzed schema.org annotations and the content of the web pages where the annotations were found, we propose a validation framework. The framework consists of the following objectives:

1. Detect the main inconsistencies between the content of schema.org annotations and the content of their corresponding web pages.
2. Develop an algorithm for the consistency check between a web page and corresponding semantic annotations. The information from web pages can be extracted from the source of a web page by tracking the appropriate HTML tags, keywords, lists, images, URLs, paragraph tags and the associated full text. Some natural language processing and machine learning techniques can be applied to extract important information from the textual description, e.g price, email, telephone number and so on. There exist some approaches to extract information from a text, such as named entity recognition [16] to locate and categorize important nouns and proper nouns in a text, web information extraction systems [6], and text mining techniques [2].
3. Define metrics to evaluate the consistencies of the semantic annotations according to the annotated content. In this step, we analyze existing data quality metrics that can be applied on the structured data and define metrics that can be useful to evaluate the consistency between a web page content and semantic annotation. We measure the consistency for different types of values, such as URL, string, boolean, enumeration, rating value, date and time formats.
4. Provide a validation tool to present the overall score for a web page and detailed insights about the evaluated consistency scores on a per value level.

3.4 Evaluation

To ensure the validity of the report results, we will organize a user study of semantic annotations and annotated web pages to prove the performance of our framework. The questionnaire will be structured in a way to get quantitative and qualitative feedback about the consistencies between a web page and annotation content (see Fig. 4) according to the results provided by the framework (see Fig. 3). As our use case, we will use annotated data and websites of Destination Management Organizations, such as Best of Zillertal Fügen[14], Mayrhofen[15], Seefeld[16], and Zillertal Arena[17].

[14] https://www.best-of-zillertal.at.
[15] https://www.mayrhofen.at.
[16] https://www.seefeld.com/.
[17] https://www.zillertalarena.com.



GÄSTEHAUS STOCK

Bichl 484, 6284 Ramsau

BESCHREIBUNG

Gepflegte Privatpension in sonniger Lage, EMPFOHLENES PRIVATQUARTIER! - Dörferbus, Bahn/Zughalt
und Skibushaltestelle in unmittelbarer Nähe, Ortszentrum von Hippach in ca. 10 min. zu Fuss erreicht
Kilometer nach Mayrhofen - Tandemflüge können direkt in der Pension gebucht werden - Fa. Stocky

Kinderermässigung auf Anfrage

```
{
  "@context": "http://schema.org",
  "@type": "LodgingBusiness",
  "url": "https://www.mayrhofen.at/unterkuenfte/angebote/det
▾ "address": {
    "@type": "PostalAddress",
    "name": "Gästehaus Stock",
    "streetAddress": "Bichl 484",
    "addressLocality": "Ramsau",
    "postalCode": "6284",
    "addressCountry": "AT",
    "telephone": "+43 5282 3786",
    "faxNumber": "+43 5282 3786",
    "email": "gaestehaus.stock@aon.at"
  },
  "name": "Gästehaus Stock",
  "description": "Gepflegte Privatpension in sonniger Lage,
```

Fig. 4. Web page content and annotation content

4 Conclusion and Future Work

Semantic annotations will be used for improved search results by search engines or as building blocks of knowledge graphs. Therefore, the quality issues in terms of structure and consistency can have an impact on where the annotations are utilized and lead, for instance, to false representation in the search results or to low-quality knowledge graphs. In this paper, we described our ongoing work for an approach to verify and validate semantic annotations and the tool that is evolving as the implementation of this approach.

For the future work, we will define Domain Specifications with SHACL in order to comply with the recent W3C Recommendation for RDF validation. We will develop an abstract syntax and formal semantics for Domain Specifications and map it to SHACL notions, for instance by aligning the concept of Domain Specifications with SHACL node shapes.

References

1. Akbar, Z., Kärle, E., Panasiuk, O., Şimşek, U., Toma, I., Fensel, D.: Complete semantics to empower touristic service providers. In: Panetto, H., et al. (eds.) OTM 2017 Conferences. LNCS, vol. 10574, pp. 353–370. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69459-7_24
2. Allahyari, M., et al.: A brief survey of text mining: classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919 (2017)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–43 (2001)
4. Boneva, I., Gayo, J.E.L., Hym, S., Prud'hommeau, E.G., Solbrig, H.R., Staworko, S.: Validating RDF with shape expressions. CoRR, abs/1404.1270 (2014)
5. Boneva, I., Labra Gayo, J.E., Prud'hommeaux, E.G.: Semantics and validation of shapes schemas for RDF. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 104–120. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_7

6. Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. IEEE Trans. Knowl. Data Eng. **18**(10), 1411–1428 (2006)
7. Fürber, C., Hepp, M.: Using SPARQL and SPIN for data quality management on the semantic web. In: Abramowicz, W., Tolksdorf, R. (eds.) BIS 2010. LNBIP, vol. 47, pp. 35–46. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12814-1_4
8. Guha, R.: Introducing schema.org: search engines come together for a richer web. Google Official Blog (2011)
9. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: evolution of structured data on the web. Commun. ACM **59**(2), 44–51 (2016)
10. Hollenstein, N., Schneider, N., Webber, B.L.: Inconsistency detection in semantic annotation. In: LREC (2016)
11. Holzknecht, O.: Enabling domain-specific validation of schema.org annotations. Master's thesis, Innsbruck University, Innrain 52, 6020 Innsbruck, Austria, November 2018
12. Kärle, E., Fensel, A., Toma, I., Fensel, D.: Why are there more hotels in tyrol than in Austria? Analyzing schema.org usage in the hotel domain. In: Inversini, A., Schegg, R. (eds.) Information and Communication Technologies in Tourism 2016, pp. 99–112. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28231-2_8
13. Kärle, E., Fensel, D.: Heuristics for publishing dynamic content as structured data with schema.org. arXiv preprint arXiv:1808.06012 (2018)
14. Meusel, R., Petrovski, P., Bizer, C.: The WebDataCommons Microdata, RDFa and microformat dataset series. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 277–292. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_18
15. Mika, P.: On schema.org and why it matters for the web. IEEE Internet Comput. **19**(4), 52–55 (2015)
16. Mohit, B.: Named entity recognition. In: Zitouni, I. (ed.) Natural Language Processing of Semitic Languages. TANLP, pp. 221–245. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-45358-8_7
17. Mühleisen, H., Bizer, C.: Web data commons-extracting structured data from two large web corpora. LDOW **937**, 133–145 (2012)
18. Panasiuk, O., Kärle, E., Şimşek, U., Fensel, D.: Defining tourism domains for semantic annotation of web content. e-Rev. Tour. Res. **9** (2018). Research notes from the ENTER 2018 Conference on ICT in Tourism
19. Panasiuk, O., Akbar, Z., Gerrier, T., Fensel, D.: Representing geodata for tourism with schema.org. In: Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM, pp. 239–246. INSTICC, SciTePress (2018)
20. Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H.: Shape expressions: an RDF validation and transformation language. In: Proceedings of the 10th International Conference on Semantic Systems, pp. 32–40. ACM (2014)
21. Şimşek, U., Kärle, E., Holzknecht, O., Fensel, D.: Domain specific semantic validation of schema.org annotations. In: Petrenko, A.K., Voronkov, A. (eds.) PSI 2017. LNCS, vol. 10742, pp. 417–429. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74313-4_31