

From Requirements as AMR-like Graphs to Automata-Based Reasoning

Aurélien Lamercerie, Annie Foret

▶ **To cite this version:**

Aurélien Lamercerie, Annie Foret. From Requirements as AMR-like Graphs to Automata-Based Reasoning. [Research Report] Université de Rennes 1. 2021. hal-03467064

HAL Id: hal-03467064
https://hal.science/hal-03467064

Submitted on 6 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Requirements as AMR-like Graphs to Automata-Based Reasoning

Aurélien Lamercerie
LIG / GETALP
Grenoble, France
aurelien.lamercerie@laposte.net

Annie Foret
Univ Rennes, IRISA
Rennes, France
annie.foret@irisa.fr

Abstract

Requirement documents, including natural language statement sets, form the basis for system design. Their validation is a critical task. Formal methods exist to assist the designer in this way, but they appear too complex for practical use. This paper presents a new methodology that aims at reducing this gap. It is based on graph-based meaning representations and compositional transduction operations, so as to extract and formalize requirements in suitable formats. Our targets are automata enabling efficient formal reasoning. This proposal is also intended to be simple, reliable and transparent.

1 Introduction

Requirement specifications are fundamental resources for system design, that can be found in a wide variety of domains, such as aeronautics, railways or automotive. These documents include statements in natural language to define precisely the expected properties. Their elaboration, intervening very early in the development cycle, is a very important task. Their validation is critical.

So, many methods have been developed to assist designers. Their aim is to highlight design errors that could appear in a specification. For example, we can cite Temporal Logics (Clarke and Emerson, 1981; Clarke et al., 1986), Modal Interface (Raclet et al., 2010; Bujtor et al., 2015) or Contract Theories (Benveniste et al., 2018). However, these proposals have not been widely adopted. The reasons are essentially pragmatic, such as lack of appropriate expertise or suitable support.

Our contribution aims at reducing the gap between the benefit of these methods in theory, and their integration in practice. For example, table 1 shows some requirements related to the management of on-road overtaking by an autonomous vehicle. Such statements may possibly contain errors: in the example $\{R_1, R_3\}$ enables a contradictory situation, when "*lines are not dotted*" and "*the autonomous vehicle is behind a slow vehicle*".

The objective of our proposal is to highlight them, by formalizing the requirements in a suitable format. Our targets are automata enabling efficient formal reasoning, such as DPAA Algebra (Lamercerie and Caillaud, 2020).

Id	*Requirement*
R_1	The autonomous vehicle must overtake if it is behind a slow vehicle.
R_2	A vehicle is slow if it is moving at a speed of less than 50 km/hour.
R_3	The autonomous vehicle cannot overtake if the lines are not dotted.

Table 1: Some requirements for an autonomous vehicle overtaking management system

Our proposal shows two stages. Following a linguistic and semantic approach, the first one produces graph-based meaning representations (such as AMR graphs). The second one combines logical analysis and structural operations, so as to filter, combine and transform intermediate sub-structures. The retained statements are finally transformed while respecting the target format.

While the proposal relies on generic principles, its application depends on the choice of the pivot representation and on the targeted objectives. Moreover, some complementary processing could be provided to deal with the ambiguities, such as an interactive disambiguation process, or a module for visualizing intermediate representations.

In this study, we have restricted our experiments to a fragment of the English language:

statements whose vocabulary is covered by a specific ontology, with some linguistic phenomena considered. We have especially emphasized the consideration of deontic modalities, namely linguistic indicators qualifying a property or an action as necessary, possible or forbidden.

The paper is organized as follows. Related work is mentioned in section 2. Section 3 concerns the first processing stage (meaning representations). Section 4 presents the second stage (transduction process). Experiments are discussed in section 5.

2 Related Work

Various approaches explore the analysis of specifications using an intermediate representation. Thus, an ontology-based knowledge modeling can be exploited as a pivot representation model (Sadoun et al., 2013). Pivot representations are also used in work aimed at generating SQL or Sparql queries from natural language (Pradel et al., 2013). The ARSENAL methodology (Ghosh et al., 2014) provides a way to transform NL requirements into analyzable logical specifications (LTL), based on dependency representation and mapping rules.

The use of controlled language is another possibility. The language ACE (Fuchs and Schwitter, 1995) constrains the structure of sentences while remaining very close to English. This language can be used, for example, to automatically translate texts in first order logic. Grammatical Framework (Ranta, 2011) combines source and target language. It is an interesting option if only controlled natural language is used. A specification pattern system can guide the designer with a structured grammar (Dwyer et al., 1999; Konrad and Cheng, 2005). The paper was followed by another publication that could also be cited: (Konrad and Cheng, 2006). A prototype implementation (SPIDER) is provided as a tool suite supporting this specification process.

Combinatory categorial grammar (Steedman, 2000) is interesting for its links with lambda-calculus, which could be used as an intermediate representation. Abstract Categorial Grammar (ACG) are a generalization dealing directly with categories and lambda-calculus (de Groote, 2001). Discourse Representation Theory (Kamp, 1981), considering the semantic content which depends on the context, can be used jointly with the CCGs to increase the coverage of linguistic phenomena. A mathematical problem solving system, accepting input in natural language, was thus designed based on CCG (Matsuzaki et al., 2017).

3 Representation From the Source

We are interested in meaning representations that form useful and readable bridges between natural expression of statements and automatic methods.

Meaning representation. Following various linguistic perspectives, a growing body of work proposes structures to capture meaning at sentence level (Abend and Rappoport, 2017). For example, AMR propose a simple description framework as labeled graphs (Banarescu et al., 2013), while UCCA (Abend and Rappoport, 2013) or the TL layer of PTL corpus (Sgall, 1992) present a multi-level structure to annote semantic distinctions, similar to dependency trees. Other representations, such as DRT, are also possible options.

Semantic graph. We intend to define a process to interpret such structures. Our proposal can be applied to any structure expressed as a labeled graph. For this purpose, a notion of semantic graph is introduced as the object of analysis processing. This allows to get rid of the particular constraints of the semantic representation used.

Definition 1 *Let $\Sigma = \Sigma_P \cup \Sigma_R$ be an alphabet of propositions and relations. A semantic graph on Σ is a tuple $\mathcal{G} = (V, E, src, tar, \kappa, \rho)$ such that V is a finite set of vertices, E is a finite set of edges, src and tar are functions from E to V (defining the source and target of each edge), $\kappa \subseteq V \times \Sigma_P$ is a set of propositions about vertices and $\rho \subseteq E \times \Sigma_R$ is a set of relations on edges.*

Figure 1 shows an AMR graph about vehicle overtaking. That structure corresponds to the requirement R1 of table 1. Different semantic concepts, and some relations between these concepts, can be observed. For example, the concept *behind* is linked by two arguments to different *vehicles*, one *slow* and the other *autonomous*.

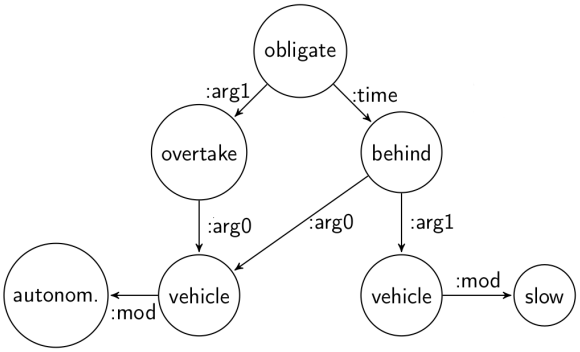


Figure 1: AMR graph about vehicle overtaking

Parsing. Various algorithms have been studied to transform textual statements into meaning representation such AMR. Some parsers are based on alignment between words and concepts (Flanigan et al., 2014; Liu et al., 2018). An inductive method, based on CCG grammars, has also been proposed (Artzi et al., 2015). We incorporated two algorithms into our experimental prototype (see section 5). CAMR algorithm (Wang et al., 2015) is an interative process transforming dependency structures into AMR, while STOG algorithm (Zhang et al., 2019) is a prediction algorithm using an attention-based-model.

4 Transduction Towards the Target

We propose a processing methodology to obtain formal statements respecting a precise format. This process applies a series of transformations on graphs according to certain conditions. These are expressed using logical formulas. The transformation operations enrich the graph to produce, at the end, the expected statement in the targeted format.

Semantic Net. The following definition (def. 2) introduces a mathematical object that aims at enriching a labeled graph, without modifying it. Inspired by the concept of filters (Cartan, 1937), the notion of *semantic net* is used to construct aggregations on the vertices of a semantic graph. Considering a set of vertices V, a *semantic net* refers to the set of parts of V, by linking them to particular attributes (type and value).

Definition 2 *Let $\mathcal{G}$ be a semantic graph, with V as set of vertices. Let $\mathcal{T}$ be a finite set of types, and $\mathcal{F}$ be a set of feature values. A* semantic net on $\mathcal{G}$ *is a structure $\eta = (X, x_0, \tau, f)$ such that $X \subseteq V$ is a subset of V, $x_0 \in X$ is the net root (also denoted as $\eta.root$), $\tau \in \mathcal{T}$ is a type ($\eta.type$) and $f \in \mathcal{F}$ is a feature value ($\eta.ft$).*

Figure 2 shows a semantic graph with some semantic nets. For example, the net η_1 is visible around the vertice of concept *overtake*. Its type is *verb*, and its feature value is *overtake*. This is an elementary net, namely a net with a unique vertice. These nets are set up to initialise the processing around each node of the graph.

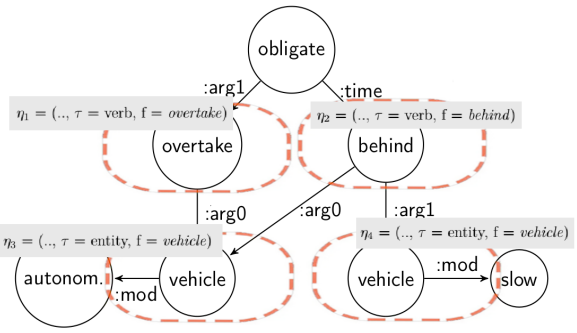


Figure 2: AMR graph with semantic nets

Sets of types and feature values are provided, with operators for changing the values. The properties of a net are derived from its type or value, which also extends the properties of the vertices and arcs of the semantic graph. We define $\mathcal{P}$ as the sets of all such properties.

Net Set Weight. The notion of weight for a set of semantic nets is introduced to give a quantitative measure of the volume of the interpretations contained in the semantic nets. The richness of a net set is expressed in the number of types, measuring the variety of interpretations included, and in the precision of the nets for each of these types.

Definition 3 *Let τ be a type and $\mathcal{N}$ be a net set.*
The weight of $\mathcal{N}$ for τ is

$$||\mathcal{N}||_\tau = Card(\bigcup_{(X,x_0,\tau,f)\in\mathcal{N}} X).$$

The global weight of $\mathcal{N}$ is $||\mathcal{N}|| = \sum_{\tau\in T} ||\mathcal{N}||_\tau$.

Transduction Schemes. A compositional transduction scheme (CTS) links a logical formula ϕ and an operator tr on nets. ϕ selects nets to be composed, while tr builds a new net, with a type and a value that depend on the composed nets.

Definition 4 *A compositional transduction scheme is a pair* (ϕ, tr) *where* ϕ *is a logical formula in* $\mathcal{R}_\mathcal{P}$ *and* tr *is a* net transduction operator *as follows:*
(1) $\mathcal{R}_\mathcal{P}$ *is the set of finite conjunctions of expressions* $p(x_1, \ldots, x_n)$*, with* p *in* $\mathcal{P}$ *(a given set of predicates);*
(2) an n-ary net transduction operator tr *is an application from* n *nets* $\eta_i = (X_i, x_{0_i}, \tau_i, f_i)$ *to a new net* $(\bigcup X_i, x_{0_1}, \psi(\tau_1, .., \tau_n, f_1, .., f_n))$*, where* ψ *is a function returning a type and a feature value.*

$\mathcal{R}_\mathcal{P}$ is a subset of first-order logic. It is interpreted on the domain $\mathcal{D}$ of semantic nets, where each symbol p is interpreted as a corresponding predicate on nets (with the same notation p). We say that a formula ϕ with free variables $x_1, ..x_n$ is satisfied by a sequence $\eta_1, ..\eta_n$ of nets, if its interpretation is true in $\mathcal{D}$ for the assignment of each x_i to η_i. We write $seq(\mathcal{N}, \phi)$ the set of sequences of nets in $\mathcal{N}$ that satisfy ϕ, under the given interpretation.

For example, figure 3 shows the application of a transduction scheme $s_{prop} = (\phi_{prop}, tr_{prop})$, producing nets of type *property* by composition. Formula $\phi_{prop}[x, y] = verb(x) \wedge entity(y) \wedge arg1(x, y)$ selects the nets expressing a relation between a *verb* and an *entity*, while operator tr_{prop} composes the selected nets, associating the type *property* and the value $concat(x.ft, '-', y.ft)$. In figure 3, we show two new nets with the values *av-overtake* and *behind-sv* (*av: autonom. vehicle, sv: slow vehicle*). Note that other schemes will generate other nets, refining the interpretation of the graph in an iterative way.

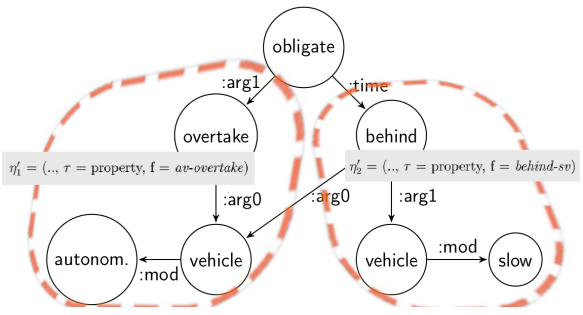


Figure 3: AMR graph with nets of type *property*

Semantic Analysis. Algorithm 1 defines an analysis procedure by semantic transduction. It is applied to any structure respecting the definition 1 associated to an initial set of elementary semantic nets (with one net by vertice). Its implementation allows to enrich this set with new nets, iteratively, by applying a CTS set (noted $\mathcal{S}$).

Algorithm 1 SEMANTIC ANALYSIS BY CTS

Input: $(\mathcal{G}, \mathcal{N})$, $\mathcal{S}$ ▷ graph, nets, schemes
Output: $(\mathcal{G}, \mathcal{N}^+)$ ▷ increased net set

```
weight := 0
N+ := N                                   ▷ initialization
while ||N+|| > weight do
    weight := ||N+||
    N := N+
    for all s = (φ, tr) ∈ S do
        N+ := N+ ∪ {tr(Γ) : Γ ∈ seq(N, φ)}
    end for
end while
```

Termination is guaranteed by the bounded property of global weight, considering that the set of type and the set of vertices are finite.

In Practice. This algorithm produces nets with types like *modality*, *entity* or *property*. A specific type (*output*) refers to the expected result: its presence indicates that the expected information can be extracted from the parsed sentence, the feature value then gives the targeted formalization.

5 Experimentation

Our experiments focused on the extraction of requirements using an ad hoc format. The process has been performed on a sentence set. It aims to extract statements defining a relation between a context and a property associated to a modality. The output is expressed in a format named MPC for *Modality, Property, Context*, that may be related to formal reasoning models. On the sentence shown as an example (figures 1 to 3), algorithm 1 yields a net of type *output* with feature value as MPC representation *<necessary, av-overtake, av-behind-sv>*.

Materials and Methods. An implementation has been done in Python and Java, with AMR as pivot structure. Classical preprocessing operations are performed with Stanford CoreNLP (Manning et al., 2014). AMR parsing adapts CAMR (Wang et al., 2015) and STOG (Zhang et al., 2019). Transduction pro-

cess configuration includes an ontology definition, based on a PropBank fragment (Palmer et al., 2005), and several transduction schemes. The evaluation corpora include 150 sentences from a variety of case studies. Various linguistic phenomena are covered, such as conditions, modalities, negation, temporalities and logical connectors.

Preliminary results. We experimented an analysis of statements, focusing on deontic modality, by translating them into an ad hoc format (MPC). Results are done in table 2. The first column specifies the size of each test set, while the AMR and MPC columns give an evaluation of the constructions obtained as output (percentage of correct constructions).

Corpus	*Size*	*AMR*	*MPC*
Parking Access	30	0.73	0.75
Vending Machine	20	0.88	0.82
Russian Dolls	30	0.83	0.81
Autonomous Vehicle	20	0.73	0.73
About Weather	50	0.80	0.75
TOTAL	150	0.79	**0.77**

Table 2: Results from the evaluation corpus (first score to AMR parsing, second score to final result).

Discussion. The results show a high efficiency of the process for simple english sentences, with some linguistic phenomena. They provide a first validation that could be refined by conducting a larger experimentation, integrating a complete treatment of several real cases and various output formats. Moreover, it would be interesting to consider another pivot representation, such as UCCA (Abend and Rappoport, 2013) or UNL (Uchida et al., 1996). We could focus on other kinds of statements or targets, by adapting the transduction schemes.

6 Conclusion

Despite advances in AI, formalizing NL requirements remains a challenge. This paper attempts to enable automata-based reasoning for such NL documents and to bring interpreting systems intended to be readable, traceable and easily adaptable.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the A.C.L.*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the A.C.L.*, volume 1, pages 77–89, Vancouver, Canada. Association for Computational Linguistics.

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in NLP*, pages 1699–1710. Association for Computational Linguistics, Lisbon, Portugal.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186. Association for Computational Linguistics, Sofia, Bulgaria.

Albert Benveniste, Benoît Caillaud, Dejan Nickovic, Roberto Passerone, Jean-Baptiste Raclet, Philipp Reinkemeier, Alberto Sangiovanni-Vincentelli, Werner Damm, Thomas A. Henzinger, and Kim G. Larsen. 2018. *Contracts for System Design*.

Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, and Walter Vogler. 2015. Nondeterministic modal interfaces. In *SOFSEM 2015: Theory and Practice of Computer Science*, pages 152–163. Springer Berlin Heidelberg.

Henri Cartan. 1937. Théorie des filtres. In *C. R. Acad. Sci. Paris*, volume 205, pages 595–598. Académie des sciences.

E. M. Clarke, E. A. Emerson, and A. P. Sistla. 1986. Automatic verification of finite-state concurrent systems using temporal logic specifications. volume 8, pages 244–263. ACM, New York, NY, USA.

Edmund M Clarke and E Allen Emerson. 1981. Synthesis of synchronization skeletons for branching time temporal logic. In *Logic of programs: Workshop*, volume 131, pages 244–263.

Philippe de Groote. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting on A.C.L.*, ACL '01, pages 252–259. Association for Computational Linguistics.

Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. 1999. Patterns in property

specifications for finite-state verification. In *Proceedings of the 21st International Conference on Software Engineering*, ICSE '99, pages 411–420, New York, NY, USA. ACM.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the A.C.L.*, pages 1426–1436. Association for Computational Linguistics, Baltimore, Maryland.

Norbert E. Fuchs and Rolf Schwitter. 1995. Specifying logic programs in controlled natural language. *CoRR*, abs/cmp-lg/9507009.

Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, and Wilfried Steiner. 2014. Arsenal: Automatic requirements specification extraction from natural language. abs/1403.3142.

Hans Kamp. 1981. A theory of truth and semantic representation. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, editors, *Formal Methods in the Study of Language*, volume 1, pages 277–322. Mathematisch Centrum, Amsterdam.

Sascha Konrad and Betty H. C. Cheng. 2005. Real-time specification patterns. In *Proceedings of the 27th International Conference on Software Engineering*, ICSE '05, pages 372–381, New York, NY, USA. ACM.

Sascha Konrad and Betty H. C. Cheng. 2006. Automated analysis of natural language properties for uml models. pages 48–57.

Aurélien Lamercerie and Benoît Caillaud. 2020. An Algebra of Deterministic Propositional Acceptance Automata. In *Forum on specification & Design Languages (FDL)*, Kiel, Germany.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR Aligner Tuned by Transition-based Parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, page 2422–2430. Association for Computational Linguistics, Brussels, Belgium.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the A.C.L.*, pages 55–60. Association for Computational Linguistics, Baltimore, Maryland.

Takuya Matsuzaki, Takumi Ito, Hidenao Iwane, Hirokazu Anai, and Noriko H. Arai. 2017. Semantic parsing of pre-university math problems. In *Proceedings of the 55th Annual Meeting of the A.C.L.*, pages 2131–2141. Association for Computational Linguistics, Vancouver, Canada.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. 2013. Natural language query interpretation into SPARQL using patterns. In *Fourth International Workshop on Consuming Linked Data - COLD 2013*, pages pp. 1–12, Sydney, Australia.

Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. 2010. A modal interface theory for component-based design. *Fundamenta Informaticae*, 108(1-2):119–149.

Aarne Ranta. 2011. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford. ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).

Driss Sadoun, Catherine Dubois, Yacine Ghamri-Doudane, and Brigitte Grau. 2013. From natural language requirements to formal specification using an ontology. In *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, pages 755–760, Herndon, VA, United States.

Petr Sgall. 1992. Underlying structure of sentences and its relations to semantics. In T. Reuthe, editor, *Wiener Slawistischer Almanach. Sonderband 33*, pages 273–282. Wien: Gesellschaft zur Förderung slawistischer Studien.

Mark Steedman. 2000. *The syntactic process*. MIT Press.

Hiroshi Uchida, M Zhu, and T Della Senta. 1996. Unl: Universal networking language–an electronic language for communication, understanding, and collaboration. *Tokyo: UNU/IAS/UNL Center*.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the A.C.L.*, pages 366–375. Association for Computational Linguistics, Denver, Colorado.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR Parsing as Sequence-to-Graph Transduction. In *Proceedings of the 57th Annual Meeting of the A.C.L.*, pages 80–94. Association for Computational Linguistics, Florence, Italy.