



Cite this: *New J. Chem.*, 2017,  
41, 1411

## On the synthesis of machine learning and automated reasoning for an artificial synthetic organic chemist

Maneesh K. Yadav

This perspective outlines current capabilities and limitations of state-of-the-art artificial intelligence methods as applied to automating the planning of synthetic routes in organic chemistry. Synthetic organic chemistry is viewed from the perspective of two prominent approaches: deep neural networks and SAT-solver based automated reasoning. After introducing these concepts to non-computer scientists, the expected performance of these approaches is estimated by surveying comparable problems in artificial intelligence. A truly artificial synthetic organic chemist that automatically constructs viable synthetic routes is clearly a challenging artificial intelligence problem and not directly amenable to existing approaches but chemistry could encourage new combinations of machine learning methods with automated reasoning to realize this goal. The importance of objective and open competitions with standardized problems and evaluations is also detailed as critical to realizing tangible computer programs that automate the planning of plausible synthetic routes.

Received 11th August 2016,  
Accepted 11th January 2017

DOI: 10.1039/c6nj02492k

[www.rsc.org/njc](http://www.rsc.org/njc)

Artificial Intelligence (AI) has made compelling recent progress with demonstrations of computer programs that outperform humans in domains that were long presumed to be unamenable to automation. Image classification,<sup>1</sup> speech recognition<sup>2</sup> and the defeat of a human go champion<sup>3</sup> are just some of the tangible examples of outperformance. The discovery efficiency of small organic molecules for therapeutics and materials involves a considerable amount of human decision making in the planning of synthetic routes, a task that has stubbornly resisted AI-based automation. In light of recent AI achievements, this apparent intractability demands an explanation in terms of an interdisciplinary perspective between computer science and chemistry in order to highlight the most productive path to an Artificial Synthetic Organic Chemist (ASOC). Robust automation in synthetic planning could play an important role in improving discovery efficiencies due to the sparseness of utility in molecular space.

Developing computer programs for the automated deduction of synthetic organic chemistry routes, Computer Aided Organic Synthesis (CAOS), has been underway for at least 40 years. These efforts have yielded well known programs including LHASA,<sup>4</sup> CAMEO,<sup>5</sup> SYNCHEM,<sup>6</sup> EROS,<sup>7</sup> *etc.* This not nearly a complete list, the reader should consult comprehensive reviews<sup>8–10</sup> for a more detailed perspective. In spite of significant efforts there is little evidence that such programs have been applied to any significant extent in industry. Judson<sup>9</sup> suggests that the “sheer

difficulty” of CAOS is an important factor in the apparent dearth of uptake. A more concise paraphrasing of the problem might be that the underlying computational methods to automate synthetic route deduction simply do not, yet, exist. Casting organic synthesis in terms of traditional computer science problems provides a context within which one can estimate the performance of current state-of-the-art approaches and how plausible extensions to those approaches might be. The shortest path problem (*e.g.* A\*) is a common casting but, like production path finding systems in automated driving, is not alone sufficient for real world use. Organic synthesis requires a richer specification of constraints and the cost functions that are generally difficult to specify in a way that accounts for important externalities (*e.g.*, hazard, availability, stability).

More recent systems<sup>11–13</sup> claim to provide varying degrees of facilitation and employ domain specific methods broadly based on reaction database similarity. Some chemists claim that the “algorithms are coming”,<sup>14</sup> but which algorithms are these exactly and how effective are they? Unlike many other sophisticated AI tools, there is no source code available for modern CAOS systems and little critical assessment as to the extent of their capabilities and performance. Of the cited systems, Wiley’s rule-based ChemPlanner (previously known as ARChem from SymBioSys) and Infochem’s ICSynth are more accurately described as Computer Aided Synthetic Design (CASD) systems. Though the terms do not appear to be used entirely consistently in the literature, CASD connotes the facilitation of human decision making as opposed to complete automation. A Chematica<sup>12</sup> evaluation license could not

SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025-3493, USA.  
E-mail: [maneesh.yadav@sri.com](mailto:maneesh.yadav@sri.com); Tel: +1-650-859-3790

be obtained without payment nor could that license be used for the purpose of critical assessment.

It is important to note that different CAOS/CASD systems use different underlying databases of reaction chemistry. The lack of a standardized, freely available reaction database has certainly hindered the development of widely available automated methods for organic chemistry. Computational organic chemistry methods<sup>15,16</sup> have long been applied in the prediction of chemical reaction in very specific molecular contexts but are not yet sufficiently powerful and automated to subsume reaction databases. There are notable attempts to apply AI to reaction prediction<sup>17,18</sup> but the methods outlined in this paper are restricted to solving the simpler problem of finding synthetic routes assuming the set of all chemical reactions is fixed and representative of the current knowledge of a good chemist.

This perspective outlines expected capabilities and limitations of the state-of-the-art in AI by casting synthetic organic chemistry in terms of two prominent approaches: Deep Neural Networks (DNN) and Satisfiability (SAT)-solver based automated reasoning. After introducing these concepts to non-computer scientists, the expected performance of these approaches is estimated by making analogies to much simpler problems like the addition of numbers. The realization of an ASOC will likely require a synthesis of highly trainable methods like DNN and deductive approaches like SAT-solvers, an approach that is increasingly being suggested in AI.<sup>19</sup> The perspective will also clearly outline the need for open, objective and credible competitions for automated synthetic route deduction to encourage progress analogous to other challenge problems such as protein folding.<sup>20</sup>

## 1 The state space of organic molecules

One might presume that a computer program could search the “state space” of synthetically accessible molecules far more efficiently than a human. Automated searches of large state spaces are the key reason why an open source chess engine running on a mere mobile phone<sup>21</sup> is likely to beat a chess grandmaster today. Why then can one not simply tilt the aim of the methods employed by chess engines to synthetic organic chemistry?

It is clear, with the benefit of hindsight, that computer science has never offered the appropriate theory and implementations to provide robust solutions to realize the ASOC. The natural representation of molecules as graphs complicates the application of many state space exploration procedures that work in domains such as chess. The computational capacity to consider such large spaces was limited at the time of earlier CAOS efforts (*e.g.*, only  $10^{20}$  states by some measures<sup>22</sup> in 1992). Does contemporary storage and computing capacity mitigate the large state space volume of chemistry? It does not appear to be the case.

The Chemical Space Project's GDB-17 constructive enumeration<sup>23</sup> suggests that the number of possible small organic molecules with no more than 17 non-hydrogen atoms is  $1.67 \times 10^{11}$  after eliminating conformationally implausible molecules and unstable functional groups (Fig. 1). The number of possible molecules

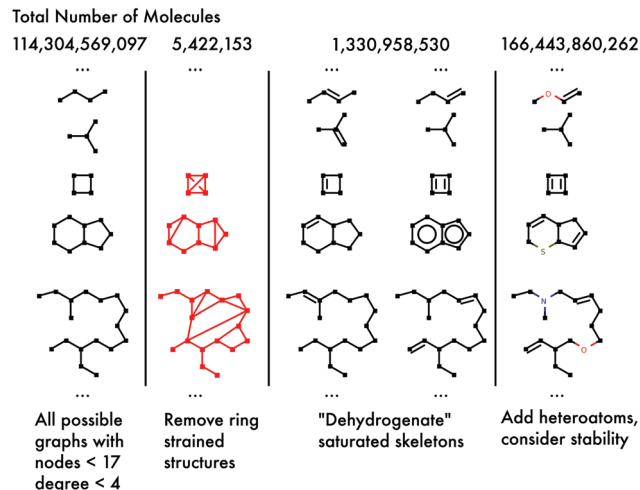


Fig. 1 The sequential enumeration of all possible molecules in GDB-17.

grows exponentially as a function of the number of atoms, so that GDB-18 enumeration and beyond “probably cannot be done exhaustively”. A molecule size of 17 non-hydrogen atoms would be considered “small” by utility metrics such as those from Xu and Stevenson<sup>24</sup> who suggest a mean drug-like count of 22. It seems unlikely that computational predictions over meaningfully exhaustive sets of molecules could ever be performed. The enormity of molecular space is also conveyed by Bohacek *et al.*'s non-constructive estimate of the number of all drug-like synthetically accessible molecules<sup>25</sup> at  $10^{60}$ . The number of synthetic routes is certainly vastly larger since, in general, the number of plausible routes to a given molecule grows with its size.

For comparison, the games of chess and go exhibit state-space sizes that are estimated to be  $10^{47}$  and  $10^{170}$  respectively but they clearly admit AI solutions with good performance.<sup>3,21</sup> Some of the most promising methods to tackle the problem of large state spaces in the context of chemistry are suggested in the next section.

## 2 Deep learning and organic chemistry

The family of deep learning approaches (which generally includes convolutional, recurrent and long short term memory neural networks) has been hailed as a significant breakthrough in AI that has enabled the construction of multi-layered DNNs which can outperform humans in challenging domains.<sup>1–3</sup> Perhaps the most prominent application in chemistry was the winning approach<sup>26</sup> to Merck's Molecular Activity Challenge where participants were given hundreds of descriptors (but not structures) for thousands of molecules and their activities against fifteen unique targets. Ma *et al.* won by showing their deep learning approach demonstrated a significantly improved correlation coefficient for activity predictions over established machine learning methods.

We could imagine a part of the ASOC as a DNN that was trained on a large set of empirical total syntheses, where the inputs were some representation of synthetic routes (*e.g.*, SMILES/SMARTS strings) and the output was a representation

of the final target molecule. In order to correctly predict the outcomes of reactions a DNN would need to infer some approximation of the calculus of organic chemistry (bond valence, stereochemistry, reaction rules and mechanisms *etc.*) as well as implementations of algorithms like graph isomorphism. DNNs that could do all this would be convenient, but there are no DNNs that approach this level of sophistication suggesting that they may not be well suited towards this sort of crisp “deductive” reasoning involved in chemistry. This suggestion does not discount the fact that specialized convolutional neural networks for graphs<sup>27,28</sup> could be promising for chemistry applications.

The limitations of DNNs in chemistry can be better appreciated when one considers their performance in inferring simpler calculi such as arithmetic. Arithmetic is trivial to appropriately trained humans and admits canonical solutions in the form of circuits and software that perform with essentially 100% reliability. Zaremba *et al.*<sup>29</sup> recently explored the application of DNNs on the (aligned) addition of nine digit numbers and the execution of very small, linear-time programs written in the Python programming language. The augmented curriculum learning approach employed in this study yielded good DNN performance with less training relative to existing methods. The adding DNN achieved 99% accuracy on a large set of nine digit pairs while the program execution DNN correctly predicted outputs, in a limited context (84% accuracy by one of the metrics in the paper), for many small Python programs. In both cases the DNN's ability to interpret progressively weakens with more digits and more complex programs. This specific DNN would make a poor substitute for many computational use cases where error rates for operations like addition must be vanishingly small ( $\ll 1\%$ ). Similarly, the utility of a DNN that made occasional mistakes in bond valence simply wouldn't be trustworthy or efficient as an ASOC.

### 3 The logic of organic synthesis

The performance of DNNs on tasks like addition and short program execution suggests they are not well-suited to capture the type of deductive reasoning needed in an ASOC. Finding the right deductive sequence to solve a problem is a well studied area of computer science where practical tools have been developed.

There is a close analogy between automatically inferring a synthetic route from the calculus of organic reactions and deducing a logical/mathematical theorem in a formal language with a respective calculus. Today there are a number of practical systems that exist for experimenting with automated deduction<sup>30</sup> that have been applied in the discovery of celebrated results in logic and math as well as industrial applications in software.<sup>31</sup> Automated deduction is often referred to as “formal verification” or “bounded model checking” depending on the specific application.

An automated reasoning engine tries to find a sequence of deductive rules (*e.g.*, modus ponens in logic or the distributive law in arithmetic) that reduce a theorem to the value “true” or “false” if it is correct or incorrect. An important application of automated reasoning is planning *e.g.*, the canonical “blocks-world” problem (Fig. 2) that finds an action sequence that a

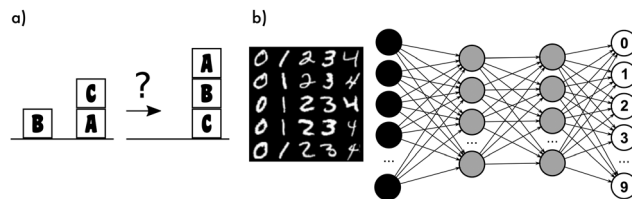


Fig. 2 An illustration of canonical problems in automated reasoning and machine learning. (a) Is the “blocks-world” problem that looks for a sequence of discrete actions (moving blocks on top of one another or to the floor) that makes an ordered stack from a random initial state. Contemporary solutions are computed on the scale of seconds for 100 block states. Certain guarantees about performance and correctness can be made with automated reasoning due to the discrete nature of the solution. (b) Is an illustration of an artificial neural network being trained on the machine learning standard MNIST dataset of handwritten images of individual digits, labeled 0 through 9. The network is trained by feeding images into the leftmost input layer. Information is propagated through several layers of nonlinear transformations, until the final layer classifies the number the digit corresponds to. Training adjusts the weights governing these transformations, and are aimed at improving the classification performance. Current deep networks can classify MNIST digits with significantly less than 1% validation error. Similar types of standard problems would be useful for organic chemistry.

robot must perform to take some initial state of disordered blocks to a goal state where they are stacked in a specified order. The program only stops once it finds a precise solution or demonstrates that none can be found. This type of search is distinct from typical machine learning classification problems that attempt to classify inputs to expected outputs (*e.g.*, handwritten digits<sup>32</sup>) with a non-zero error rate.

Solving such planning problems is challenging and has been attacked from a diversity of perspectives. One approach to the problem that has gained popularity is to cast the planning problem as a large conjunction of logical statements with Boolean variables, *i.e.* a Satisfiability (SAT) problem.<sup>33</sup> Some of the Boolean variables in the SAT instance encode the truth value of performing a given action at a given step, the goal being to find the right assignment of truth values so that the goal state is reached without violating integrity constraints (*e.g.*, one cannot perform two actions in the same step).

The SAT problem requires intractable exhaustive enumeration in worst case execution but modern solvers provide solutions to problems with millions of variables in seconds on modest hardware. It would be tremendously convenient if synthesis problems could be cast to SAT with the same ease as traditional planning. While there have been some attempts along these lines,<sup>34,35</sup> there are sound reasons as to why a SAT approach has some obvious drawbacks in organic chemistry.

Chemists test synthetic hypotheses by mentally matching the left hand side of a reaction rule to a molecule they are attempting to transform and then substituting with the right hand side (often referred to as graph transformation). This process requires the exhaustive checking of all possible graph embeddings to find a match (subgraph isomorphism) in the worst case. This graph transformation approach has been implemented in chemistry software packages,<sup>36</sup> but practical graph transformation

systems lack sophisticated constructs (e.g., reachability) that are available in packages such as rewriting logic systems<sup>37</sup> (which, conversely, are not always suitable for graph transformation). Like the SAT problem, practical runtimes for graph matching are achieved with appropriate methods and heuristics.<sup>38</sup> While there is a straightforward encoding of subgraph isomorphism as a SAT problem, the encoding grows intractably as more atoms are added. This impractical aspect of representing reaction rules in a SAT instance make it difficult to find an obvious approach for applying SAT solvers directly to synthetic organic chemistry. One could imagine a richer formulation of SAT that could incorporate an efficient subgraph isomorphism algorithm that computer scientists would likely refer to as Satisfiability Modulo Theories (SMT)<sup>39</sup> of graphs, but no such theory exists yet.

Perhaps the biggest challenge in applying automated reasoning to chemistry is the laborious and error prone translation of background knowledge (in the form of tables, papers, simulation results) into a consistent logical schema. The difficulty in encoding background knowledge is a widely recognized shortcoming of logic-based approaches<sup>40</sup> that is less of a problem for DNNs that can be trained on essentially any data type.

Automated deduction, planning and writing efficient SAT solvers are all active areas of research and while synthetic chemists cannot apply these tools directly yet, it is likely they will play an important role in an ASOC. It is important to recognize the central role of regular competitions and standardized problems (e.g. blocksworld) in SAT progress, e.g. SAT solvers enjoyed broad overall annual improvement of 19% over recent years.<sup>41</sup>

## 4 Where to go from here?

In spite of the recent progress in AI, the realization of an ASOC does not appear to be a simple matter of applying state of the art techniques to chemistry. This does not detract from the immense importance of automation for improving the efficiency of discovery efforts. Demonstrating that an ASOC could measurably outperform a team of human scientists over controlled trials would be a considerable milestone for AI and boon to chemistry.

Methods that combine the trainability of machine learning approaches like DNNs with power of state space search techniques in automated reasoning will be an important part in achieving human (out)performance with AI. The desire to bridge the gap between machine learning and automated reasoning has always been a deep current in AI research where there are many exciting new constructions and methods that are being reported.<sup>42</sup> Perhaps the most prominent and relevant work to chemistry is being done in automated theorem proving in mathematics. Like finding a sequence of reactions to a target molecule, mathematicians would like to discover a sequence of deductive steps as a proof for a target theorem. Like the large corpus empirical synthetic routes, mathematics consists of a large corpus of existing proofs that should be useful in finding new proofs. Combining machine learning with automated reasoning is a relatively new area in automated theorem proving and the application of DNNs have

only just begun.<sup>43</sup> Mathematics faces many of same challenges as chemistry including finding clever ways to trim large state spaces to find solutions that run in practical time. Alemi *et al.* propose<sup>44</sup> “premise selection” that can limit the corpus of knowledge being considered to only the most relevant facts (analogous to trimming starting materials and syntheses that are irrelevant to a target molecule). The expectations of AI in chemistry can be viewed in a realistic light when compared to the current state of the art in mathematics and computer science. Automated theorem proving has found simple solutions to problems that long eluded humans<sup>45</sup> but automated theorem provers are not yet an essential tool for mathematicians. Nonetheless, there is no doubt that there is common ground here that should be leveraged for ASOC efforts in chemistry.

A number of new neural network architectures are being proposed to augment DNNs with capacities that will facilitate performance on tasks that have strong deductive components and reliability requirements. Neural Turing Machines are DNNs that have been extended with external components like an addressable memory and can perform declarative tasks like memory copying with complete accuracy.<sup>46</sup> General “Interfaces”<sup>49</sup> to this architecture have been proposed as a potential way to synthesize such DNNs with deductive automated theorem provers. The Differentiable Neural Computer is a further extension<sup>47</sup> of this family of architectures that has shown impressive performance on deductive tasks like reachability and shortest path in graphs as well as a blocksworld-type problem. The Neural Programmer-Interpreter has been developed<sup>48</sup> to learn programs and compose new ones using only a small number of rich examples. This architecture was successfully used to automatically perform addition on 10 digit numbers perfectly and an improvement over the work that was cited earlier.<sup>29</sup> All of these architectures combine the trainability of existing DNNs and augment them with powerful deductive reasoning and even the ability to operate directly on graphs. There is no doubt they demand exploration and application in the realm of organic chemistry.

Recent progress in AI has boosted commercial interest and investment which has played an important role in fueling the recent burst of new methods on challenging problem domains. Chemistry can be an important problem space to attack with these methods as long as comprehensive databases of molecules and reactions are made easily accessible so that AI researchers have appropriate data to work with. The essential role of regular, rigorous competitions must be recognized in order to expect any novel and tangible AI capabilities to be delivered into the hands of chemists. Standardizing problems, detailed performance profiles and encouraging the sharing of source code have allowed computer scientists to objectively evaluate precisely how valuable their methods are and where improvement must be focused.

The ultimate goal of the Artificial Synthetic Organic Chemist isn't just the mere prediction of retrosynthetic routes but the eventual automation of discovery in chemistry. This seems like a far more credible idea in the context of modern Artificial Intelligence than it has been historically. Reasoning about chemistry may be a more difficult challenge for a computer



than playing go, but AlphaGo's victory against a human champion was realized decade earlier than was anticipated.<sup>49</sup> Perhaps with the appropriate research investments and open competitions, progress could arrive sooner than most of us would expect.

## Acknowledgements

The author would like to thank his colleagues at SRI International's Computer Science Lab including Carolyn Talcott, Natarajan Shankar, Eric Yeh and Dejan Jovanović for feedback and reading of this manuscript. The author is also grateful to the referees and the editor for their critical assessment and suggestions for improvement. Finally the author would like to thank Wiley for their trial license for the ChemPlanner product and insightful discussion.

## References

- 1 K. He, X. Zhang, S. Ren and J. Sun, 2015, arXiv:1502.01852 [cs].
- 2 D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan and Z. Zhu, 2015, arXiv:1512.02595 [cs].
- 3 D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, *Nature*, 2016, **529**, 484–489.
- 4 E. J. Corey, *Q. Rev., Chem. Soc.*, 1971, **25**, 455.
- 5 W. L. Jorgensen, E. R. Laird, A. J. Gushurst, J. M. Fleischer, S. A. Gothe, H. E. Helson, G. D. Paderes and S. Sinclair, *Pure Appl. Chem.*, 1990, **62**, 1921–1932.
- 6 H. L. Gelernter, A. F. Sanders, D. L. Larsen, K. K. Agarwal, R. H. Boivie, G. A. Spritzer and J. E. Searleman, *Science*, 1977, **197**, 1041–1049.
- 7 J. Gasteiger, M. G. Hutchings, B. Christoph, L. Gann, C. Hiller, P. Löw, M. Marsili, H. Saller and K. Yuki, in *Organic Synthesis, Reactions and Mechanisms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1987, vol. 137, pp. 19–73.
- 8 Maimone, Tom, *Computer-Assisted Organic Synthesis (CAOS)*, 2006, Available at: [http://www.scripps.edu/baran/images/grpmtgpdf/Maimone\\_Mar\\_06.pdf](http://www.scripps.edu/baran/images/grpmtgpdf/Maimone_Mar_06.pdf).
- 9 P. Judson, *Knowledge-Based Expert Systems in Chemistry: Not Counting on Computers*, RSC Pub, Cambridge, 2009.
- 10 A. Cook, A. P. Johnson, J. Law, M. Mirzazadeh, O. Ravitz and A. Simon, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2012, **2**, 79–107.
- 11 A. Bøgevig, H.-J. Federsel, F. Huerta, M. G. Hutchings, H. Kraut, T. Langer, P. Löw, C. Oppawsky, T. Rein and H. Saller, *Org. Process Res. Dev.*, 2015, **19**, 357–368.
- 12 S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk and B. A. Grzybowski, *Angew. Chem., Int. Ed.*, 2016, **55**, 5904–5937.
- 13 J. Law, Z. Zsoldos, A. Simon, D. Reid, Y. Liu, S. Y. Khew, A. P. Johnson, S. Major, R. A. Wade and H. Y. Ando, *J. Chem. Inf. Model.*, 2009, **49**, 593–602.
- 14 Derek Lowe, *The Algorithms Are Coming*, 2016, Available at: <http://blogs.sciencemag.org/pipeline/archives/2016/04/12/the-algorithms-are-coming>.
- 15 E. Kraka and D. Cremer, *Acc. Chem. Res.*, 2010, **43**, 591–601.
- 16 A. Cembran, L. Song, Y. Mo and J. Gao, *J. Chem. Theory Comput.*, 2009, **5**, 2702–2716.
- 17 M. A. Kayala, C.-A. Azencott, J. H. Chen and P. Baldi, *J. Chem. Inf. Model.*, 2011, **51**, 2209–2222.
- 18 J. N. Wei, D. Duvenaud and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2016, **2**, 725–732.
- 19 W. Zaremba and I. Sutskever, 2016, arXiv:1505.00521 [cs].
- 20 A. Kryshchuk, B. Monastyrskyy and K. Fidelis, *Proteins: Struct., Funct., Bioinf.*, 2016, **84**, 15–19.
- 21 *Stockfish Outlasts "Rybkamura"*, 8-24-14, Available at: <http://www.chess.com/news/stockfish-outlasts-nakamura-3634>.
- 22 J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill and L.-J. Hwang, *Inf. Comput.*, 1992, **98**, 142–170.
- 23 J.-L. Reymond, *Acc. Chem. Res.*, 2015, **48**, 722–730.
- 24 J. Xu and J. Stevenson, *J. Chem. Inf. Comput. Sci.*, 2000, **40**, 1177–1187.
- 25 R. S. Bohacek, C. McMartin and W. C. Guida, *Med. Res. Rev.*, 1996, **16**, 3–50.
- 26 J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl and V. Svetnik, *J. Chem. Inf. Model.*, 2015, **55**, 263–274.
- 27 T. N. Kipf and M. Welling, 2016, arXiv:1609.02907 [cs, stat].
- 28 D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, 2015, arXiv:1509.09292 [cs, stat].
- 29 W. Zaremba and I. Sutskever, 2014, arXiv:1410.4615 [cs].
- 30 D. A. Plaisted, in *Automated Deduction – CADE-25*, Springer International Publishing, Cham, 2015, vol. 9195, pp. 3–28.
- 31 M. Bennion and I. Habli, Companion Proceedings of the 36th International Conference on Software Engineering, New York, NY, USA, 2014, pp. 175–184.
- 32 Y. LeCun, C. Cortes and C. J. C. Burges, *The MNIST Database*, 2016, Available at: <http://yann.lecun.com/exdb/mnist/>.
- 33 J. Rintanen, *ECAI*, 2012, pp. 684–689.
- 34 A. Masoumi, M. Antoniazzi and M. Soutchanski, *GCAI 2015. Global Conference on Artificial Intelligence*, 2015, pp. 176–195.
- 35 R. Fagerberg, C. Flamm, D. Merkle and P. Peters, *Principles and Practice of Constraint Programming*, 2012, pp. 900–915.
- 36 M. Mann, H. Ekker and C. Flamm, in *Theory and Practice of Model Transformations*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, vol. 7909, pp. 52–53.
- 37 *All about Maude – a high-performance logical framework: how to specify, program, and verify systems in rewriting logic*, ed. M. Clavel, Springer, Berlin, 2007.
- 38 R. Geiß and M. Kroll, in *Applications of Graph Transformations with Industrial Relevance*, Springer, 2008, pp. 568–569.
- 39 S. Malik and L. Zhang, *Commun. ACM*, 2009, **52**, 76.
- 40 T. Walker, C. O'Reilly, G. Kunapuli, S. Natarajan, R. Maclin, D. Page and J. Shavlik, in *Inductive Logic Programming*, Springer, 2010, pp. 253–268.

- 41 K. Grace, *Algorithmic Progress in Six Domains*, Tech. rep. machine intelligence research institute technical report, 2013.
- 42 B. M. Lake, T. D. Ullman, J. B. Tenenbaum and S. J. Gershman, 2016, arXiv:1604.00289 [cs, stat].
- 43 J. Urban and J. Vyskocil, *Autom. Reasoning Math.*, 2013, **7788**, 240–257.
- 44 A. A. Alemi, F. Chollet, G. Irving, C. Szegedy and J. Urban, 2016, arXiv preprint arXiv:1606.04442.
- 45 *Handbook of Automated Reasoning*, ed. J. A. Robinson and A. Voronkov, Elsevier, MIT Press, Amsterdam, New York, Cambridge, Mass, 2001.
- 46 A. Graves, G. Wayne and I. Danihelka, 2014, arXiv:1410.5401 [cs].
- 47 A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu and D. Hassabis, *Nature*, 2016, **538**, 471–476.
- 48 S. Reed and N. de Freitas, 2015, arXiv:1511.06279 [cs].
- 49 *Google's AI Masters the Game of Go a Decade Earlier Than Expected*, 2016, Available at: <https://www.technologyreview.com/s/546066/googles-ai-masters-the-game-of-go-a-decade-earlier-than-expected/>.