Formalization of CASP Aggregates Bypassing Grounding

Description of Project

As complexity grows in the field of *Artificial Intelligence* (AI), there is an increasing drive towards more robust models capable of reasoning. *Answer Set Programming* (ASP) is a declarative programming language which enables the programmer to express facts and rules about a world. Unlike in traditional imperative programming languages such as Python, in which the programmer is responsible for specifying how a problem should be solved, in ASP the programmer simply specifies what the problem is, allowing the solver to reason about the problem. This makes ASP incredibly useful in modeling complex problems and one of the most prominent approaches to problem solving in *Knowledge Representation and Reasoning*, a well-stablish subfield of AI.

To solve an ASP problem, the application needs to be *grounded*, in which the high-level program is transformed into a variable-free program. In doing so, variables are replaced constants before being evaluated by the solver. This *grounding* process becomes a bottleneck when dealing with large domains, something that usually appears in real-world problems. *Constraint Answer Set Programming* (CASP) is an extension of ASP that integrates external solvers to deal with those large domains, at the expense of losing some of the high-level features of ASP. This project seeks to settle the foundations to combine the effectiveness of CASP to deal with large domains with the high-level features of ASP. In that direction, we plan to formally characterize CASP using the logic of *Here-and-There with Intentional Functions*. This logic was recently successfully used to characterize ASP with aggregates (Fandinno and Hansen, 2024), a long-lasting open problem.

Activities, Process, or Methodology

The *Here-and-There* (HT) logic is the intermediate logic that is obtained from classical logic by replacing the exclude middle axiom:

$$F \vee \neg F$$

by the so call Hosoi axiom:

$$F \vee (F \rightarrow G) \vee \neg G$$

Semantically, this can be characterized as a Kripke model with two worlds respectively called here and there. The here world entails information that is explicitly known, while the there world represents information that is assumed. The importance of this logic in the context of ASP strives from its usefulness to characterize *strong equivalence*: two sets of rules are strongly equivalent if they have the same behavior in any context. Formally, two sets of rules Π_1 and Π_2, are strongly equivalent if and only if $\Pi_1 \cup \Delta$ and $\Pi_2 \cup \Delta$ have the same stable models for any arbitrary set of rules Δ. Interestingly, it has been shown that two set rules are strongly equivalent if and only if they are equivalent in the logic of HT (Lifschitz et al. 2001). This property is not satisfied by classical logic. For example, the ASP programs:

$$a :- not\ b. \quad \text{and} \quad b :- not\ a.$$

respectively correspond to the formulas:

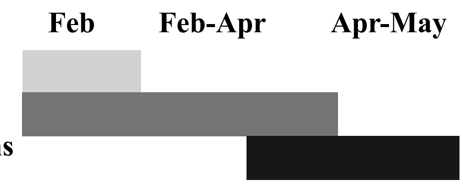
$$\neg b \rightarrow a \quad \text{and} \quad \neg a \rightarrow b$$

which are classical equivalent. However, these two programs have different answer sets, and they are, thus, not strongly equivalent. As can be inference by the correspondence stated above, these two formulas are not equivalent in HT.

These ideas form the basis for this body of work, in which we aim to use mathematical proofs to show how CASP stable models can be expressed in the semantics of HT. This includes the standard iterative methodology in Computer Science and AI of revising the state-of-the-art, identifying unsolved problems, posing hypotheses followed by proofs or rebuttals, and evaluation and publication of results. Results can be visually tested in the programming language of *Clingcon*, an extension of the widely used answer set solver *Clingo*, which incorporates constraint solving capabilities. Once expressed in HT semantics, these programs can then be additionally transformed into classical First-Order logic to automate the process of checking for strong equivalence in a similar manner to what was shown in Fandinno and Hansen (2024).

Timeline

Task 1: Background research on HT, HTc, and CASP
Task 2: Show CASP can be Captured in HT with Intentional Functions
Task 3: Integrate Aggregates in CASP using HT with Intentional Functions



The timeline for this project is expected to span from February through May 2025.

Task 1 will be focusing on learning the foundations of international functions, the semantics of HT and HTc, and the semantics of CASP. This will include reviewing previous work, including work done by Cabalar et al (2024), in which the correspondence between CASP and HTc has already been proven, as well as the word done by Fandinno and Hansen (2024) in which it was shown that aggregates can be captured in intentional functions in ASP. This will build the foundations required to construct a thorough proof for tasks 2 and 3. In ***task 2***, the student will work on constructing a proof showing correspondence between CASP and HT with intentional functions following the standard methodology outlined in previous work such as Fandinno and Hansen (2024) and Calabar et al (2024). ***Task 3*** will continue to build off this work, aiming to show that aggregates can be captured in intentional functions in CASP using the semantics of HT in addition to that of HTc.

Student/Faculty Mentor Roles

The student will be responsible for the major work for tasks 1-3. Dr. Fandinno will direct and supervise the research on developing the infrastructure and implementation of the tasks presented, ensuring that milestones are met.

Previous Internal Funding Received

The student, Josh Gryzen, has received a FUSE grant for 2023/2024 for the project titled *Evaluating Open-Source Large Language Models on bAbI-Tasks* which was presented at the Seventeenth Midwest Speech and Language Days in Ann Arbor, Michigan. The results show how “out-of-the-box” pretrained LLMs such as OPT, Bloom, and Llama struggle to perform reading comprehension and logical deduction tasks from the bAbI dataset. While some larger models score proficiently on the easier tasks, all models regardless of parameter size see substantial drop off in accuracy on the tasks with increased complexity. Additionally, the results show the distinction between “out-of-the-box” pretrained LLMs and fine-tuned LLMs in that the “out-of-the-box” models do not benefit as much from providing context from instructions as would be seen from a fine-tuned chatbot-style model.

The project outlined in this proposal, *Formalization of CASP Aggregates Bypassing Grounding*, is not directly related to the student’s previous work, *Evaluating Open-Source Large Language Models on bAbI-Tasks*.

Budget

Student Stipend: **$ 2,000.00**

Support in the amount of $2,000 is requested for the student, Josh Gryzen, to complete the research tasks on the project as described in the Project Description. The stipend is budgeted based on an hourly rate of $16/hour for 125 hours of work.

References

Cabalar, P.; Fandinno, J.; Schaub, T.; and Wanko, P. (2024) *Strong Equivalence of Answer Set Programs with Constraints*∗. [Manuscript submitted for publication].

Fandinno J. and Hansen Z. (2024) *Recursive Aggregates as Intensional Functions in Answer Set Programming: Semantics and Strong Equivalence*. [Manuscript submitted for publication].

Fandinno, J.; Lifschitz, V.; L¨uhne, P.; and Schaub, T. 2020. Verifying Tight Logic Programs with anthem and vampire. Theory and Practice of Logic Programming, 20(5): 735–750

Lifschitz, V.; Pearce, D.; Valverde, A.: Strongly equivalent logic programs. ACM Transactions on Computational Logic 2, 526–541 (2001)

UNIVERSITY OF NEBRASKA AT OMAHA
COLLEGE OF INFORMATION AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

Jorge Fandinno, Assistant Professor
The Peter Kiewit Institute
1110 S. 67th Street
Omaha, NE 68106

(402) 554-4952
jfandinno@unomaha.edu

Dear Colleagues,

I am delighted to write in support of Josh Gryzen for his FUSE proposal titled *Formalization of CASP Aggregates Bypassing Grounding*. I have known Josh since Fall 2023, when he presented a review of the state of the art in *Large Languages Models* at our weakly seminars invited by Dr. Lierler. Since then, he was enrolled in my CSCI 4220 *Principles of Programming Languages* course where he was the student with the highest grade of the class. He also became grader of this course and my CSCI 4450 *Introduction to Artificial Intelligence*. Josh has expressed interest in joining our PhD program once he finish his double major in Computer Science and Mathematics. We have discussed several research options for his future and he found that the formalization of aggregates in a way that bypasses grounding was something that aligns with his interests. In his proposal he sets an exact stage for his research. This topic nicely aligns with topics he studied in both CSCI 4220 and CSCI 4450, and a future PhD dissertation. This is a topic in theoretical Computer Science, and his double major in Computer Science and Mathematics puts him in an ideal position to tackle this topic as it requires a deep understanding of both declarative programming (a Computer Science topic) and mathematical proofs. We established a schedule of weekly meetings to navigate the literature landscape of the field and build his familiarity with state-of-the art in declarative programming in general, and Answer Set Programming in particular. PhD. student Zach Hansen will join our research endeavors and will participate in further reading and discuss sessions with Josh. Zach has published 4 papers with me on top-tier venues regarding the topic of Josh's project, and is well qualified to assist in this endeavor. Once Josh has familiarity with these concepts, he will outline the definitions of necessary concepts and theorems. Once this is done satisfactorily, he will address the proof of those concepts.

As mentioned above Josh Gryzen is ideally prepared to hit the ground running as the FUSE funding becomes available and has great chance to deliver on his promises. His proposal addressed *an important scientific question: how to bypass the need for grounding on a declarative programming paradigm such as Answer Set Programming*. The formal results obtained as a result of Josh's project are directed to develop more effective algorithms to address knowledge-intensive search and optimization problems with quantitative data. Developing more effective algorithms for this class of problems is the main goal of the NSF CAREER award titled *Answer Set Programming for Quantitative Information* that I received this Spring. The FUSE project will teach Josh the necessary concepts to join the efforts of this NSF CAREER award once he graduates.

Overall, Josh Gryzen is outstandingly qualified to excel in the FUSE project. The availability of FUSE funding will be instrumental in promoting Josh abilities as a young investigator and making his steps on a path to becoming a researcher. I am planning to hire him as a Research Assistant once he graduate to work on the aforementioned NSF CAREER award (unfortunately the award funds cannot be used to support undergraduate students). In the process of the proposed work we will continue weekly meetings geared towards solving issues that Josh will encounter while working on the project. I will help him to pose research questions and split the attempts to construct an answer to these into manageable steps. He will also present on the outcomes of his project at one of the *NLPKR research lab* seminars. I will further assist him in pursuing the opportunities to present his accomplished work at the scientific venues such. Attending an event of the kind will be an invaluable experience for Josh empowering him to consider a PhD career in the future and excel in it. Needless to say that computing fields are in more and more need for highly qualified researchers.

On these grounds I support Josh Gryzen FUSE proposal wholeheartedly.

Sincerely,

Jorge Fandinno