Disposition paper for the coursework ‘Selection of premises for PyRes Automatic Theorem Prover’

Jannis Gehring
Student id: 6732014, course: INF22B
Baden Württemberg Cooperative State University (DHBW)
inf22115@lehre.dhbw-stuttgart.de

Contents

I) Short description of the work 1
I.A) Context in terms of content 1
I.B) Goal 1
I.C) Context in terms of project 1
II) Structure and Schedule 2
II.A) Development procedure 2
II.B) Central milestones 2
II.C) Structure of coursework 2
II.D) Basic Literatur 2
III) Time schedule Gantt chart 3
References 4

I. Short description of the work

A. Context in terms of content

Automatic theorem provers try to prove the correctness of formal theorems using several strategies and algorithms. When proving theorems with a big set of logical assertions, one main challenge is the implementation of efficient algorithms to minimize the time, computing power and storage needed for the computation.

There are multiple strategies to achieve this goal. One of them is to restrict the search for the proof to a subset of *relevant* clauses.

Relevant clauses can be determined with different heuristics. Some of them compute relevant clauses by measuring how many and which symbols they share with certain other clauses. Another defines them by finite models, with the set of the relevant clauses sharing models with the theorem. It is also possible to model relevance with machine learning approaches like a Bayesian learning system.

The alternating path relevance approach determines relevance by constructing a graph modelling all symbols of all clauses. Edges are defined either by switching the symbol within the clause or by a possible unification with another clause. Using this graph, a *relevance distance* can be computed. The set of clauses can now be filtered for clauses with a maximum relevance distance to the theorem.

There do exist implementations of this approach, but they only filter for *relevance connectedness* ($\Leftrightarrow$ relevance distance $\neq \infty \Leftrightarrow \exists$ path between clauses).

B. Goal

Goal of the coursework is

1) the **first implementation of selection of relevant clauses with a specified relevance distance** and
2) the **evaluation of this approach regarding metrics like efficiency and performance**

C. Context in terms of project

The algorithm shall be implemented for *PyRes*, a resolution based theorem prover for first-order logic developed in Python. PyRes values readability over efficiency, which is why the algorithm should be implemenented in an understandable and well-documented fashion.

The algorithm can be tested with *StarExec*, a distributed logic solving service developed at the University of Iowa.

TPTP (Thousands of Problems for Theorem Provers) is a collection of problems that serve as test samples.

If the algorithm turns out to perform well, it becomes a candidate for being implemented in

E-Prover, a efficiency and performance oriented theorem prover developed in C.

II. Structure and Schedule

A. Development procedure

The key steps of the development process are as follows:

1) Familiarization
    a) With the theoretical foundations
    b) With the PyRes-codebase
2) Solving the problem by hand -> Defining the algorithm steps
3) Implementation of MVP
4) Evaluating MVP on large dataset of test samples
5) Steps (3.) and (4.) are repeated in an iterative development cycle
6) Final adjustments
7) Final evaluation of algorithm
    a) Identification of theoretical and technical bottlenecks

The coursework shall be written alongside the development. The introduction chapters can be written and finished within the development phase; the main part is not going to include every single adjustments made, therefore it will be written with a certain delay regarding the development to identify key adjustments.

B. Central milestones

The first milestone will be when the development of the MVP is finished. Another main milestone is the evaluation of the first MVP. These results guide the development phase directly afterwards. Then, end of development in mid of April will be the next big milestone, moving the focus on the thesis. The last big milestone will be finishing the MVP thesis, which is scheduled for start/mid of May.

A detailed time schedule including the milestones can be found later in the document Figure 1.

C. Structure of coursework

The provisional structure of the coursework is as follows:

- Introduction
- Theory
    - First-order logic
    - Current scientific landscape for clause selection
    - Alternating path theory
- PyRes architecture
- Design
    - Initial algorithm
        - Ideas
        - Result
- Implementation
    - Original algorithm in practice
    - [Main improvement 1]
    - [Main improvement 2]

    …
    - [Main improvement N]
    - Main development drawbacks
- Evaluation & validation
    - Implementation validation
    - Experimental setup
    - Experimental result
    - Evaluation of the experimental result
- Future work
- Conclusion

D. Basic Literatur

The coursework aims to implement the alternating path relevance theory formalized py D. Plaisted, therefore his papers [1] and [2] serve as the foundation of this coursework.

[3] serves as a system description for PyRes.

III. Time schedule Gantt chart

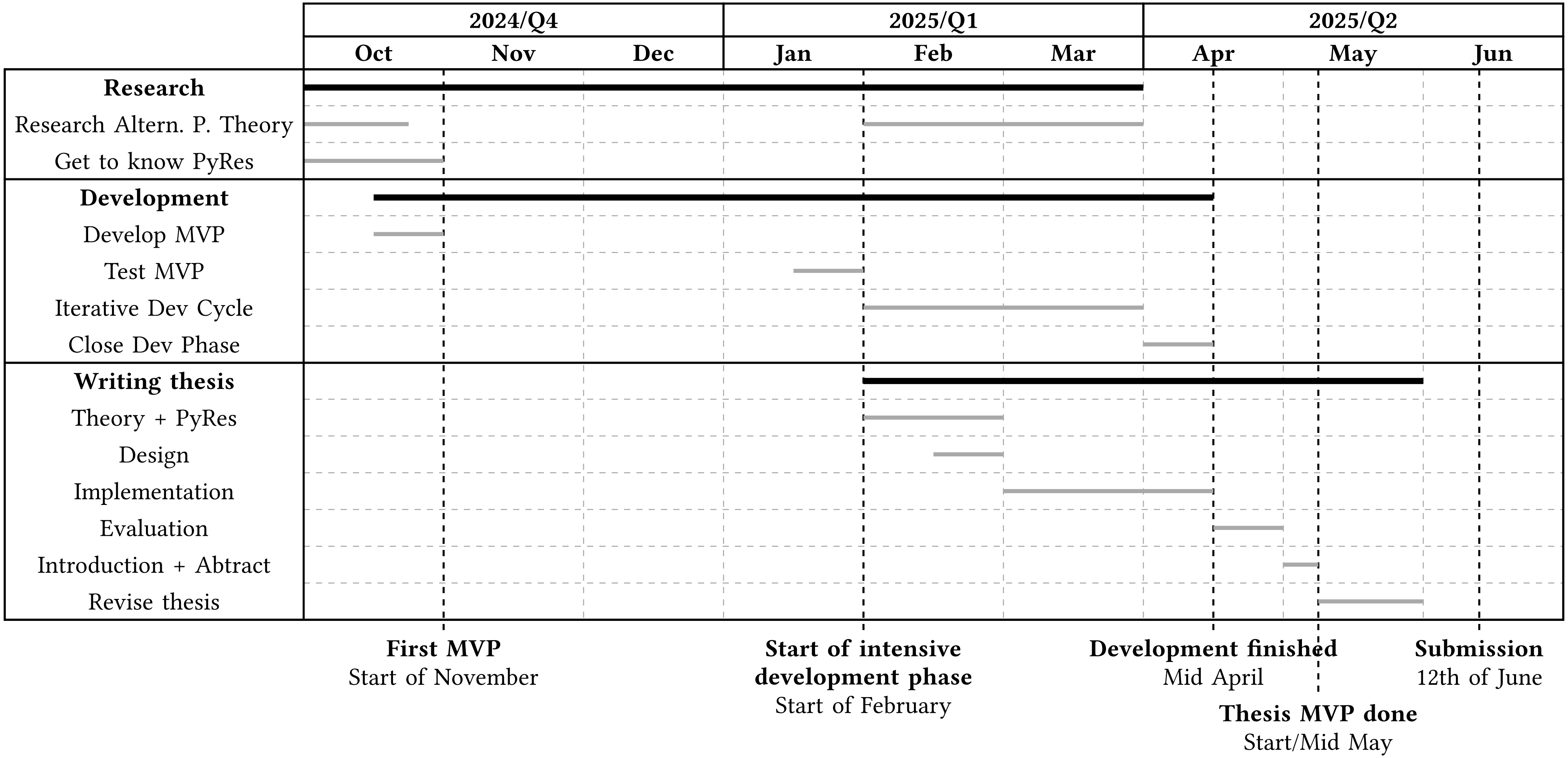

Fig. 1: Gantt chart displaying different phases and milestones of the coursework.

References

[1] D. A. Plaisted, “Properties and Extensions of Alternating Path Relevance - I.” [Online]. Available: https://arxiv.org/abs/1905.08842

[2] D. A. Plaisted and A. Yahya, “A relevance restriction strategy for automated deduction,” *Artificial Intelligence*, vol. 144, no. 1, pp. 59–93, 2003, doi: https://doi.org/10.1016/S0004-3702(02)00368-5.

[3] S. Schulz and A. Pease, “Teaching Automated Theorem Proving by Example: PyRes 1.2,” in *Automated Reasoning*, N. Peltier and V. Sofronie-Stokkermans, Eds., Cham: Springer International Publishing, 2020, pp. 158–166.