The Australian National University
2600 ACT | Canberra | Australia

School of Computing

College of Engineering and
Computer Science (CECS)

Template for Project Reports

— 24 pt Honours project (S2/S1 2021–2022)

A thesis submitted for the degree
Bachelor of Advanced Computing

By:
Pascal Bercher

Supervisors:
Dr. FirstName LastName
Dr. Second Supervisor

March 2022

Declaration of Authorship:

Except where otherwise indicated, this report is my own original work.

Wednesday 9th March, 2022,

(Pascal Bercher)

Acknowledgements

If you wish to do so, you can include some Acknowledgements here. If you don't want to, just comment out the line where this file is included.

There is absolutely no need to write an Acknowledgement section, so only do so when you want to – i.e., if there's somebody you really want to thank (for example if you received extraordinary supervision). The more important the work, the more likely that an Acknowledgement section doesn't look off. In a 24 pt. Honours thesis it would for example look more reasonable than for a 6 pt. project report.

Unrelated to the Acknowledgements, but important:

Note that there exist two different title page designs. You may choose the one that you find more appealing. Just use the configuration file to select this style. There, you also have to put in all the other information about this report like your name, the kind of report (Honours vs non-Honours) and so on.

Abstract

An abstract is a very short summary (around 15 lines) of your entire work (that doesn't use citations by convention). There are plenty of examples you can take a look at – simply take a look at some papers published at top-tier venues, e.g., by your supervisor.

Table of Contents

Chapter 1

Introduction

1. To give a high-level *introduction* into the research area and
2. to *motivate* your research done within it.

(Not necessarily in this order.)

1.1 Introduction to HTN Planning

Point 1 is important as you should not assume any technical background in the specific technical area of your work. Provide some high-level introduction (without technical definitions) that can be understood by anybody with some basic mathematical understanding. What such an accurate level of abstraction/presentation is might depend on the specific topic of your thesis. Also consult your supervisor(s) for his/her (their) opinion(s) and preferences.

1.2 Motivation

Point 2 should make clear why the conducted research (or experiments etc.) is relevant and important. This is the part where you should make the readers interested in your work! Convince them to continue reading your work!

Be specific about the precise contributions that your work actually does and list them in the text (preferably in its own paragraph; depending on the number of contributions and how well they can be separated you could also provide them in a bullet point list).

potential pros and cons of total vs. partial order

- Advantages of Partial Order:

  - plan recognition: independent goals can be described in parallel (Daniel should be able to write something about that)
  - partial order is more expressive (both in terms of plan existence and in terms of computational complexity) meaning that more problems can be expressed
  - Domain model might be more intuitive: if a task is independent of some others it might be counter-intuitive to demand a certain position of it (if artificially made totally ordered)
- Advantages of Total Order:
  - computational complexity is lower (fewer worst-case solving time)
  - we can exploit specialised algorithms as well as heuristics. Note that heuristic design is comparably easy for total-order problems due to the missing interaction between tasks.

Further advantages from having a compilation of PO into TO plans

- We actually get another class of decidable partially ordered problems that's orthogonal to tail-recursive ones! (Because if the criterion 'matches', we know that the PO domain is equivalent to the resulting TO domain; the latter is decidable whereas the former is not.)
- We can use more efficient algorithms and heuristics.

*

1.3 Contributions

Formalisation of properties

Testing on Benchmark

Chapter 2

Background

This section should explain all the technical background that is important for being able to read your report. Recall that your report must be completely self-contained, so you should only assume “mathematical understanding”, but no specific knowledge. All such knowledge should be provided here, e.g., the formalization and vocabulary of the research areas in which your work resides.

Note that this is not the same as reviewing related work. Related work puts the work done/described in your report into context of other (mostly recent) work that’s done by others. In contrast the current chapter is not so much about what work others have done, and more about the formalization (and possibly standard techniques) that you require to describe your contributions (but since you probably didn’t come up with these formalizations, you of course still need to cite the respective authors).

Please make use of sections and subsections as it’s reasonable to better structure this (or any) chapter.

2.1 Classical Planning

Problem: $< D, S_I, G_I >$

Domain: $< F, A, >$ where $A \in 2^f \times 2^f \times 2^F$

2.2 Partially Ordered HTN Planning

HTN Problem

Domain

Task Network

2.3 Totally Ordered HTN Planning

2.4 Further Definitions

Undecidable A decision problem is equivalent to, a function that accepts a infinite (set?) of inputs and returns a yes or no. The decision problem is undecidable if it can be proved that there exists no algorithm for the problem that always leads to a correct yes-or-no answer.

Tail recursion

Chapter 3

Related Work

This chapter reviews the work that is most related to the research questions investigated by you in this work. Please note that there are various options on *where* you include it.

- You could include it *here* (i.e., where you see it right now in the template). Since it's after the formal definitions (Chapter 2), you can explain what the other works have done on some level of detail, yet you need to keep in mind that you did not yet explain your own contributions (except abstractly in the abstract), which slightly limits the level of technical detail on which you can compare these approaches here.
- You could also make it a subsection of Chapter 2. This choice might also depend on the length of this chapter. Is it worth its own full chapter?
- Alternatively, you might include this chapter after the main part of your report, i.e., right before Chapter 6. When you do this, you can go into more technical detail since the readers will have read your entire work, so they know exactly what you've done and you can therefore discuss differences (like pros/cons etc.) in more detail.
- When you take a look at scientific papers (preferably at top-tier venues), you might notice that not every single paper has a related work section. This is because in principle related works might also be addressed/positioned in the introduction or in the main part of the work. But since this is not a "standardized scientific publication", it is very strongly advised that you devote its own section to related work as done in this template.

If you prefer any of the latter two options, discuss this with your supervisor(s).

Chapter 4

Reasonable Title for Main Content

This chapter holds your actual contributions. Depending on what you did in your project you might either use just one chapter describing your contributions or you use multiple ones – think about this and consult your supervisor(s). Apart from the empirical evaluation (if there is any) – is the main part of your work and thus the core of this document. In any case, give this chapter/these chapters a reasonable name[1] and explain clearly what you have done and how it builds upon or extends the closeliest related work.

The following sections give additional advice that is specifically tailored to students who are new to either LaTeX or scientific writing.

4.1 General Advice

- **Start early.** Writing a report is hard and takes time. More than you think. *Hofstadter's Law:* "It always takes longer than you expect, even when you take into account Hofstadter's Law." – *So start early!*
- **Read and check your work!** First of all, please use a spellchecker! Each LaTeX editor should support this, so please always turn it on. Secondly: Read your work *carefully* and *multiple times* before showing it to your supervisor. Your supervisor is surely happy to help you – that's her/his/their job! But you will benefit from investing your time: Assume that her/his/their time investment is always the same n – so the more of this doesn't have to be invested for errors you could solve yourself, the more time can be invested for more important advice.
- **Involve your supervisor!** Don't be afraid to reach out to your supervisor(s)! She/he (they) is (are) literally being paid to supervise you and help you succeed! :) So make sure you get what you need to be successful, don't hold back.

[1]You may also rename the tex filename if you wish.

4.2 Building the PDF

- **Latexmk:** This very useful commandline tool works on all standard operating systems: Linux, macOS, and Windows. The installation overhead is minimal, and on Linux you probably have *everything* already installed! Check it out here: `https://mg.readthedocs.io/latexmk.html` – this is the preferred option since it's the most convenient and takes the least time to run, i.e., compile.
- **makefile:** If you use Linux (and potentially macOS) you can use the provided makefile, which features five different functions.
  - Executing "make" (i.e., executing that command in the project's folder) is the same as executing "make mk". This is nothing else than a shortcut for the Latexmk program, but with the required parameters. This is the preferred way since Latexmk "magically" knows what it has to do, i.e., what to compile and how often.
  - "make mkonline" is an extension of the above: it runs Latexmk in an online mode that compiles again after every single change that is saved; so you always see the newest version automatically without having to re-compile.
  - "make all" will compile the document multiple times (1 times bibtex, 4 times pdflatex). This makes sure that all references like links to page numbers and figures work correctly, and that all citations are correctly processed. Note that this is not required if you use any of the above options, as those are basically the "state of the art" and do the minimal amount of required work.
  - You may also call the script via "make quick", which compiles exactly once. This is much quicker than the last, but may not process all references correctly. Again, Latexmk is the preferred option. It even doesn't do anything if that's not required.
  - With "make clear" you can conveniently delete all temporary files. This is sometimes required if compilation fails (e.g., when you create wrong bibtex entries, which may be caused by non-supported symbols or commands).

4.3 Technical Advice ($\LaTeX$ etc.), Rules for Writing Reports, and Scientific Advice

- **Watch Your Dots:** You need to "escape" all blanks following a dot that is not ending a sentence. E.g., the sentence "This is 6 pt. project." needs to be coded "`This is 6 pt.\ project.`" as otherwise it looks as follows: "This is 6 pt. project." – you see that in here the spacing after "pt." is wrong (i.e., way too large). This is because $\LaTeX$ interprets each dot (with a following space) as one that ends a sentence – after which more space is allocated. An escaped space in contrast produces a fixed space that doesn't get stretched. (Fun fact: when (mechanical) typewriters were still a thing, authors were hitting the space twice after each "sentence-ending dot" to produce exactly the behavior that LaTeX does automatically.)

- **Headlines/Titles:**
  - Titles (chapters, sections, subsections) are capitalized according to specific rules. Basically everything is written capitalized except of some specific words (in, on, the, ...). You can search for capitalization rules and even tools, which you might find useful to be consistent.
  - Also note that – purely due to aesthetical reasons – you should:
    - Always have at least one line of "glue text" between the chapter title and the first section, i.e., anything that briefly introduces what comes next.
    - Never use exactly one section. If you use sections, there should be at least two – because otherwise it's just pointless; you could (if you had just one section) just eliminate it as otherwise the chapter title should then already reflect the content.
- **Appearance:** Don't forget that your work isn't parsed by a robot, but read by a human being. So make it pleasant for them, i.e., optically pleasing. Some examples:
  - *Page and Line Breaks:* If some headline ends up at the end of a page, that might look ugly. Consider adding `\pagebreak` right before it to force placing it on the next page. That will likely look much nicer. In some rare cases you might want to do the same on a per-line basis, where you can use the `\linebreak` command to enforce a linebreak at that position. In the same context the command `\mbox{}` might be useful, which prevents a linebreak of the word(s) specified as argument.
  - *Big Gaps in the Document:* Make sure that there are no huge gaps in the middle of your report/text, e.g.:
    - For example, make sure that a chapter doesn't end with a single line on a new page, that's just ugly and thus careless. The same applies for the table of contents: If it happens to have so many entries (sections/subsections etc.) that it jumps to a next page just because of one or two lines/entries, then just search (e.g., using stackoverflow) how to reduce the space between the lines so that it fits. Show some effort.
    - Also make sure that when including figures or tables that there is no huge gap before them, that may happen depending on their size.
  - *Respect Boundaries!* Another thing that's often done "wrong" regarding appearance is having expressions (mostly formulae) going over the allowed bor-

der. That is ugly and careless, so rephrase to prevent that. (That would even be strictly forbidden in the context of publishing a paper.)

Do all of this *briefly before you hand in*, as all that depends on your final layout. Adding, changing, and removing text will of course change the appearance, so do all this in a very final step.

- **Appearance of Mathematical Expressions and Algorithms:** This is *not* a LaTeX tutorial, so only frequent beginner's errors are being mentioned and abstract advice is provided. For an introduction to LaTeX see the last list entry.
  - *General advice:* If you are new to LaTeX and need to write down a lot of formulae, theorems, or algorithms etc., spend a few minutes to at least scroll through the manuals of the respective standard packages – this will already show you examples of the appearance of what you can do. Just search for the manuals for `amsmath` (for equations), `amsthm` (for theorems/propositions), and `algorithm2e` (for algorithms).
  - *Variable names:* Very often, variable names will not be single letters, but *words*, such as *pre* for precondition or *eff* for *effects*. Since variables are usually used in math mode, there's the temptation to just write them in math mode. For example, one might write `$\langle pre,eff\rangle$`, resulting into "$\langle pre, eff\rangle$". You hopefully see that this looks incredibly ugly – because LaTeX sets the text incorrectly. Instead, you should put it into math italics. To save you effort, you should define a new macro:

    ```
    \newcommand{\Pre} {\ensuremath{\mathit{pre}}}
    \newcommand{\Eff} {\ensuremath{\mathit{eff}}}
    ```

    With this you can now simply write `$\langle \Pre,\Eff\rangle$`, which now results into $\langle \mathit{pre}, \mathit{eff}\rangle$, which looks exactly as it should. Do this for *all* your variables to make sure they look nice. (This template includes the file macros.tex, which you can use for all your macros.)
- **Graphics/Pictures:**
  - The most important thing to know about graphics is that they "float". That is, LaTeX decides where they should be placed, not you. You can of course influence that a bit (e.g., by the arguments for the figure environment, cf. `https://tex.stackexchange.com/questions/39017/how-to-influence-the-position-of-float-environments-like-figure-and-table-in-lat`), depending on where you put the source code that includes the graphics, but LaTeX will have the final word on where *exactly* it will appear. Still, please make sure that your graphics appear at reasonable places so that reading the document remains being a pleasure. Anyway, that means that you will have to reference/cite each graphic. Thus, the reader will take a look at a graphic (i.e., figure) exactly when you reference it in the text, not when it's "being

Figure 4.1: A caption for the illustrated graphic (H). It's made long on purpose so that you can see that it simply doesn't look good that the caption is below – since there is now a lot a free/unused space. It would have been a better choice to place the caption next to it, which you can see in Figure 4.2.

seen". (This also means that graphics/figures that are not referenced could and should be deleted from your work.)

– In Figure 4.1 you see an example figure with its caption below – which looks very ugly. Do that if the graphic is centered and wide enough. In contrast, Figure 4.2 provides the caption next to the figure – which in this case looks quite good since the graphic is portrait rather than landscape, i.e., now there are no white/lost spaces.

Figure 4.2: Captions should not explain/interpret graphics, but enable the reader to read it. Further interpretations and conclusions should be in the text only. For this graphic the following caption might be appropriate: "Motivational expressions written and pinned on a wooden fence (H)."

Also note how in this case the caption should be on the left (and not below the graphic as in Figure 4.1) as otherwise there would be a lot of free/unused space. The code for this is a bit more complicated, but now you can simply change it, so it should be fine... :)

- **Colored Links:** By default you will see that all hyperlinks (e.g., to figures like Figure 4.2, citations like by Smith et al. (2115), etc.) are colored. Personally, I (the author of this template) find that easier to read in the PDF than the alternative. The alternative is that hyperlinks are indicated by colored boxes that surround them (where the text itself remains black). You can choose between the two by the setting the option `colorlinks = true` or `colorlinks = false` in the hyperref definitions (where `true` colors the words, whereas `false` produces the box). Note a major difference between the two: The box is an annotation, so it's not visible when printing. If the text itself is colored then that's an actual text color, so it will appear as you see it in the PDF also in the printout. You can of course also change the colors.

- **Tables:** Standard LaTeX tables don't look particularly pleasing. Thus, it's generally recommended to use the `booktabs` package, which was designed to produce aesthetically pleasing tables. Table 4.1 provides an example, taken from the official manual (slightly adapted). One of the most important rules: Do not use vertical lines. Note that the table caption appears on top. This is set on purpose to align with several publishers, who demand that captions for tables are *above* tables, whereas those for figures (i.e., everything else: graphics, plots etc.) are *below*.

Table 4.1: This table lists prices for different kinds of animal meat.

Item		
Animal	**Description**	**Price** ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

- **Bibliography:** There are various points that you should consider when you add a publication into your bibtex file. The first basic rule is: ***never blindly copy some bibtex entry from the internet*** – most of them are of very poor quality. Instead, double-check each entry by hand via trustworthy sources, such as DBLP (`https://dblp.org/`), the publisher's webpage, or the websites by the authors. For each entry, consider the following:
  - *Correctness:* Is the type correct? For example, papers published in conferences should be "inproceedings", papers published in journals are "article". These are often wrong when using non-trustworthy internet sources. Also check the content like year, page numbers, etc.
  - *Completeness:* Make sure that each entry contains all fields that are required (like authors, title, booktitle etc.) but also those that are "usually specified".

The latter is hard for a beginner, so this is the recommendation: Also provide page numbers, publisher, year.

  - *Consistency:* Make sure that the various entries are consistent to each other. For example, conference papers usually use acronyms. Make sure to either always add the respective acronym (preferred) or never. If you add it, add it always in the same way. E.g., don't add "..., IJCAI-12", "... (IJCAI-12)", "... (IJCAI '13)", "... (IJCAI 2015)" – use always the the systematicity. Likewise with the conference titles. For example, do not write "Proceedings" for one but "Proc." for another. Stay consistent.

- **Citing Papers:** In most cases, you place a citation right behind the respective proposition that you want to back up. Let's assume that the next citation backs up the sentence that you currently read (Smith et al., 2115), it was thus plausible to put it exactly there – and not at another position of this sentence. To use this kind of citation (that you put behind the respective proposition), you use the command `citep{}`. However, if for some reason you need or wish to use the paper *explicitly* within your sentence, then refer to its *authors* (not the paper) using the `cite{}` command. For example, I can claim that the work by Smith et al. (2115) will be quite funny once it will have been done! This is just nicer than claiming that the work "described in (Smith et al., 2115)" will be influential. The reason is again consistency, because normally citations like the very first one (where everything is contained by parentheses, not just the year) are not objects of the sentence. So using them sometimes as objects and sometimes not would be inconsistent.

  In addition to the commands `citep{}` and `cite{}`, `\citeauthor{}` is sometimes useful. This just lists the author(s), but without the year. I.e., it's an alternative to `cite{}` that you should use when you want to mention the authors whereas you used similar citations before so that there is just no need to add the year again.

  Also note that you can easily cite multiple works with one command as shown in (the code of) this sentence (Cooper and Hofstadter, 2015; Smith et al., 2115).

- **LaTeX Issues?** One of the best sources for solving LaTeX issues is `https://stackoverflow.com/`. In case your document doesn't compile, check out the log file and search for "error", often that points towards the problem quickly. I (the author of the template) recommend to use the "online version" of Latexmk (reminder: which you can execute by simply executing "make mkonline"), because then the document recompiles every single time you save (and showing any error message in the terminal) – so you should find your coding errors instantly since you know what you have done when the error was introduced. If the online mode fails, fix the error and enter a large X, compilation will then continue.

  You might also want to take a look at a well-known LaTeX introduction (Oetiker, 2021), which in the current version – according to Oetiker – takes "only" a bit more than two hours to work through.

4.4 (3)

Formally define which possibilities we have to linearize a model: * input: method with n linearizations; output: n new methods, one per linearization (careful: there are usually n=k! many linearizations if k is the number of plan steps in a method) * input: method with n linearizations; output: m¡n methods (i.e., we delete some linearizations; the interesting question is which!)

4.5 (4)

Formally define solution preserving properties: * all solutions remain (note that even if 1 method with n linearizations get transformed into all n methods, this question is still undecidable) * at least one solution remains (again: undecidable) * all optimal solutions remain * at least one optimal solution remains The report/work should answer which of these criteria is guaranteed for the translation that's investigated

4.6 (6)

[e] Dr. Gregor Behnke already implemented a technique for 'this' available in the PANDA-3 planner. It was used to create many IPC TO domains, though it was never published, i.e., neither described nor properties like preserving of solutions were investigated.

1. Problem = (D, S_I, TN_I). $D = (F, T_A, T_P, delta, M)$
2. Consider each method independently, no interaction is considered.
3. For each compound task c infer its (super-relaxed) preconditions and effects.
4. Construct a graph with possible dependencies:
   - For each add effect a of c move all tasks with precondition a behind c and all tasks with a delete effect in front of it.
   - For each delete effect d of c move all tasks with precondition d before c and all tasks with an add effect behind it.
5. If this graph does not have a circle, choose this linearization. [It needs to be proved that (or whether) this sacrifices solutions!]
6. If this graph does have a circle, choose a linearization at random

   [Pretty sure that this is incomplete! It should be checked how often that was the case in the IPC domains since they all admit a solution! Also investigate whether there are other circle-breaking strategies that are complete. Note though that even the circle-free systematic might be incomplete. If that's the case even, then clearly breaking cycles, no matter how, can't work either.]

4.7 Possible Improvements to Gregor's Algorithm

[a] For the beginning it might be easiest how a primitive plan can linearized to a subset of all linearizations without losing any solutions. (For some tasks it will simply not matter where they are executed – this should be formalised. That should essentially be the POCL or PO-all criterion [note that they are not equivalent: the PO criterion will find more linearizations; see [**Bercher & Olz, AAAI-2020 POPOCL**] by selecting just any sequence. So the problem will be which of them select so that we don't lose anything upon upwards-propagation.)

[b] Note that for any reasonable criterion, compound tasks might need some form of preconditions/effects associated to it (so that we don't have to look into the decompositions of the compound task anymore). The semantics of these precondition/effect-augmented compound tasks is still not the same as for actions, but we still have at least *some* state information. Note that we could do either trivial tests that just collect all "reachable" preconditions/effects within the subtree of each compound task (that might potentially already exist in the data structures) – those would then be extremely loose, in an extreme case we could even collect all state variables both as preconditions as well as effects that way, or we could use stronger notions as those developed and computed by **Olz, Biundo, and Bercher (AAAI'21)**. These do however not yet exist (neither in theory nor practice) as their techniques (as well as even formal definitions) are only defined for totally ordered methods (which clearly doesn't help as we need them for partially ordered ones). Note though that at least for sub-problems which are totally ordered those can be computed – but only if the independence can be proved. Consider the grammar intersection problem with initial plan G1 and G2 (the start symbols of two grammars). Each Gi is totally ordered, but their plans can still interact, so computing the precs/effects of them individually might fail as they assume on the assumption that nothing comes 'in between' – but it does! Plans of G1 and G2 can intertwine.

[c] Alternatively to computing a set of preconditions and effects reachable for some compound task one could represent the ground TDG (which is a structure of polynomial size – it's basically just a graphical representation of all methods!) directly and try to find some approach that deals with all methods in a unified way, e.g., by putting all relevant information into one single SAT or ILP formula. (That somehow seems quite a promising idea to me; though one still had to find a criteria formalizing what we want to do with the TDG, so it somehow just shifts the problem to another one; it doesn't solve it.)

[d] Sufficient criterion for linearization: If there's some partitioning of state features (i.e., if one subset of actions A uses F and another subset A' only uses F', then one execute all in A first followed by A' or vice versa). Much smarter than this simple (and completely unrealistici criterion is the following: if all effects in a compound task's sub-tree TDG don't influence the preconditions (computed in the same way) of another compound task's subtask, we can again linearize as then there's no dependency. NOTE that there is already a paper on computing independent subproblems in HTN planning:

”New Advances in GraphHTN: Identifying Independent Subproblems in Large HTN Domains” by Lotem and Nau. I strongly assume that they do something very similar; we should definitely check what that is! (I however remember that it’s not about turning a PO model into a TO model, so it’s definitely not the same.) ...

Chapter 5

Evaluation

Clearly not every project has empirical components (in particular in mathematics or theoretical topics) – though many do. So in case you were coding anything and conducted an empirical evaluation, this is where you should report the results.

As always, make adequate use of sections/subsections to structure this chapter. These are some suggestions on what you may report on:

- Benchmark Selection: Which data set did you choose to evaluate your approach on? Why did you make that specific selection? Would there have been alternatives?
- Hardware setup: What hardware was used (processor, RAM, etc.), and which operating systems and software were used?
- Results: Report the plain data (plots, tables, etc.) and their *interpretation*, i.e., did the approach work well or not? Do we know on which subset it worked well (or badly) and why was it like this? Do the results raise further questions and thus directions for future research/investigations?

When reporting your results using graphs and plots, make sure to provide all information necessary to interpret the data, e.g., axis and graph labeling (cf. Figure 5.1).

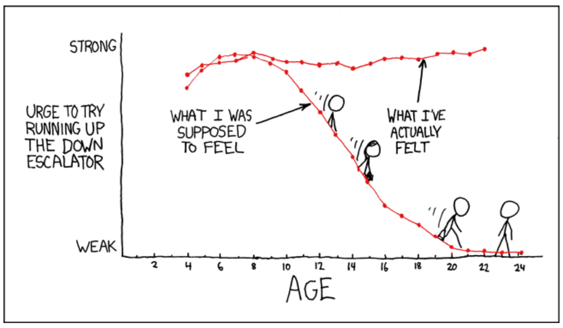
STRONG
URGE TO TRY RUNNING UP THE DOWN ESCALATOR
WEAK
WHAT I WAS SUPPOSED TO FEEL
WHAT I'VE ACTUALLY FELT
2 4 6 8 10 12 14 16 18 20 22 24
AGE

Figure 5.1: A graph illustrating the importance of axis and graph labeling. (Graphic taken from https://xkcd.com/252/.)

We must do an evaluation that shows how often the criterion works, i.e., in how many instances (per domain) we can say 'yes: translate!'

To prove that our technique is beneficial, we conduct a standard empirical evaluation on PO domains and compare the runtime PO planners vs. transformation + TO planners.

Chapter 6

Concluding Remarks

If you wish, you may also name that section *"Conclusion and Future Work"*, though it might not be a perfect choice to have a section named "A & B" if it has subsections "A" and "B". Also note that you don't necessarily have to use these subsections; that also depends on how much content you have in each. (E.g., having a section header might be odd if it contains just three lines.)

6.1 Conclusion

This chapter usually summarizes the entire paper including the conclusions drawn, i.e., did the developed techniques work? Maybe add why or why not. Also note that every single scientific paper has such a section, so you can check out many examples, preferably at top-tier venues, e.g., by your supervisor(s).

6.2 Future Work

On top of that, you could discuss future work (and make clear why that is future work, i.e., by which observations did they get justified?).

Note that future work in scientific papers is often not mentioned at all or just in a very few sentences within the conclusion. That should not stop you from putting some effort in. This will (also) show the examiner(s)/supervisor(s) how well you understood the topic or how engaged you are.

Appendix A

Appendix: Explanation on Appendices

You may use appendices to provide additional information that is in principle relevant to your work, though you don't want *every reader* to look at the entire material, but only those interested.

There are many cases where an appendix may make sense. For example:

- You developed various variants of some algorithm, but you only describe one of them in the main body, since the different variants are not that different.
- You may have conducted an extensive empirical analysis, yet you don't want to provide *all* results. So you focus on the most relevant results in the main body of your work to get the message across. Yet you present the remaining and complete results here for the more interested reader.
- You developed a model of some sort. In your work, you explained an excerpt of the model. You also used mathematical syntax for this. Here, you can (if you wish) provide the actual model as you provided it in probably some textfile. Note that you don't have to do this, as artifacts can be submitted separately. Consult your supervisor in such a case.
- You could also provide a list of figures and/or list of tables in here (via the commands `\listoffigures` and `\listoftables`, respectively). Do this only if you think that this is beneficial for your work. If you want to include it, you can of course also provide it right after the table of contents. You might want to make this dependent on how many people you think are interested in this.

Appendix B

Appendix: Explanation on Page Borders

What you find here is an explanation of why the border width keeps flipping from left to right – which you might have spotted and wondered why that's the case.

Firstly, that is *intended* and thus correct, so there is no reason to worry about this. The reason is that this document is configured as a two-sided book, which means:

- We assume the document will be printed out,
- that this will be done in a two-sided mode (i.e., the document will be printed on both sides of each page), and
- that the bookbinding will be in the middle, just like in every book.

When you open the book, there are three borders of equal size n. This however requires that even pages have a border of n on their left and $\frac{n}{2}$ on their right, and odd pages have a border of $\frac{n}{2}$ on their left and n on their right. This is illustrated in Figure B.1.

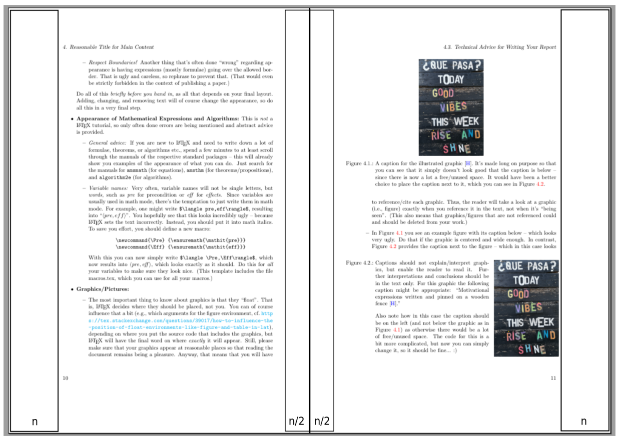

Figure B.1: Illustration showing why page borders flip.

Bibliography

COOPER, S. L. AND HOFSTADTER, L. L., 2015. Superfluid vacuum theory: Spacetime is a surface of the superfluid. In *The Troll Manifestation*, 42–73. Scientific American. [Cited on page 13.]

H, A. https://www.pexels.com/de-de/foto/schwarzes-holzernes-wanddekor-3095771/. According to https://www.pexels.com/license/, all photos on that website can be used for free. [Cited on page 11.]

OETIKER, T., 2021. A (not so) short introduction to LaTeX 2_ε. https://www.ctan.org/tex-archive/info/lshort/english/. [Cited on page 13.]

SMITH, R.; SMITH, S.; AND SMITH, M., 2115. Wubalubadubdub: Rick & Morty forever and forever a hundred years. In *Plumbus: How they do it*, C137–∞. Adult Swim. [Cited on pages 12 and 13.]