How to write a (good) research paper

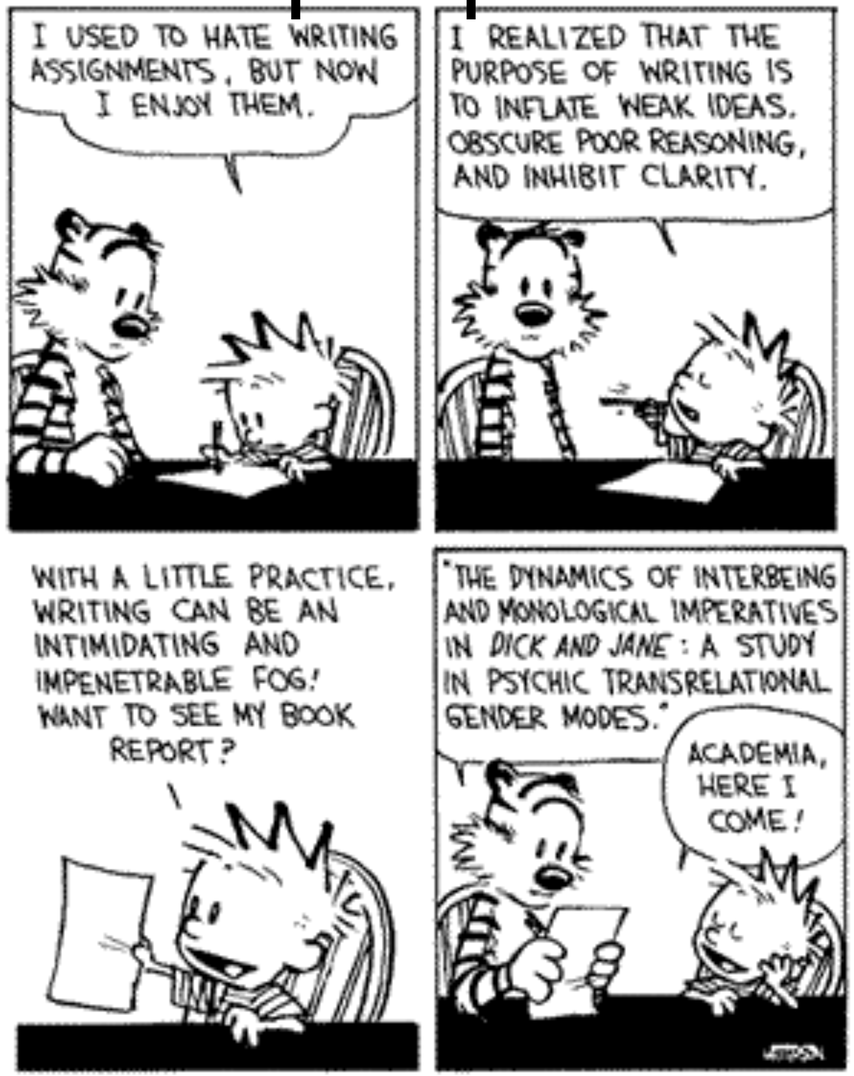
Felipe Meneguzzi
Slides adapted from Simon Peyton-Jones (Microsoft Research)

English English English

- Modern science is written in English
  - If something is worth reading, it is written in English
- This is not cultural imperialism
  - The Romans wrote science in Greek
  - Even the French nowadays use English for scientific publication

Why should I write a paper?

- **Fallacy of paper writing**
  we write papers and give talks mainly to impress others, gain recognition, and get promoted

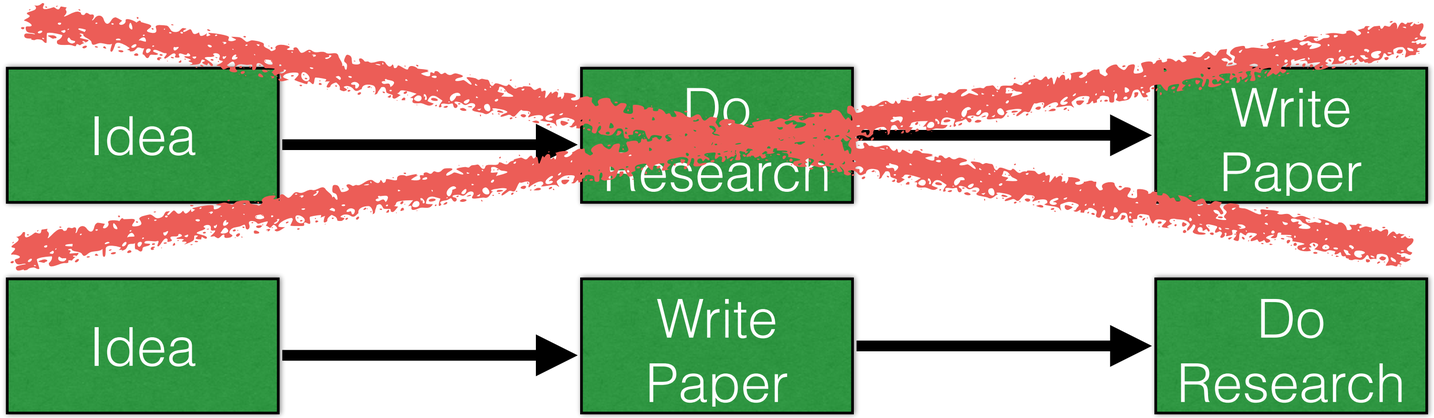


Why should we write a paper?

- Papers communicate ideas
  - Your goal: to infect the mind of your reader with **your idea**, like a virus
  - Papers are far more durable than programs (think Mozart)
    - Remember that FORTRAN library that sorts lists using the quicksort function? **Me neither!**
- **The greatest ideas are (literally) worthless if you keep them to yourself**

Writing papers: model I

Writing papers: model II



- Forces us to be clear, focused
- Crystallises what we don't understand
- Opens the way to dialogue with others: reality check, critique, and collaboration
- Writing papers is a primary mechanism for **doing** research
  (not just **reporting** it)

Do not be intimidated

- **Fallacy:** You need to have a fantastic idea before you can write a paper or give a talk. (Everyone else seems to.)

- Write a paper, and give a talk, about **any idea,**
no matter how weedy and insignificant it may seem to you

Do not be intimidated

Write a paper, and give a talk, about any idea, no matter how insignificant it may seem to you

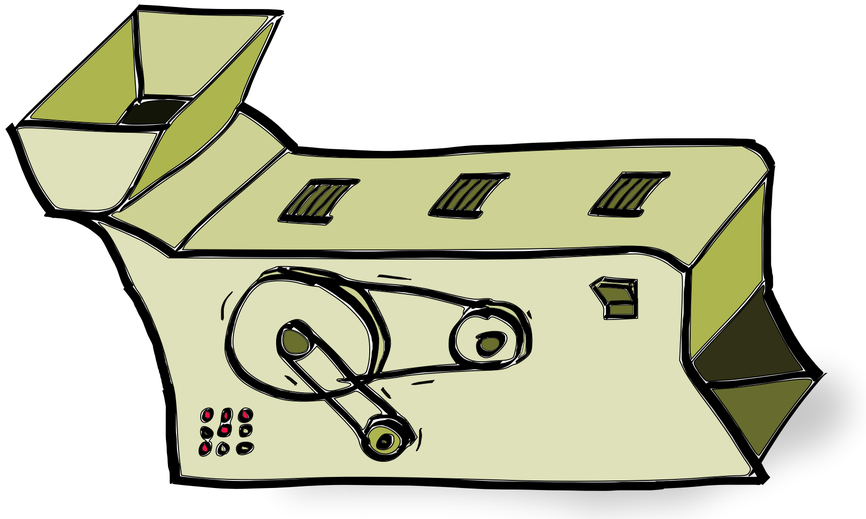
- **Writing the paper is how you develop the idea in the first place**
- It usually turns out to be more interesting and challenging that it seemed at first

The purpose of your paper is…

- **To convey your idea**
  - …from your head
    to your reader's head
  - to infect your the mind of your reader with **your idea**, like a virus
  - **Everything** serves this **single goal**

The purpose of your paper is not…

- To describe the WizWoz system



- Your reader does not have a WizWoz
- She is primarily interested in re-usable **brain-stuff**, not executable artefacts

The Idea

Idea
A re-usable insight, useful to the reader

- Your paper should have just one “ping":
  one clear, sharp idea

- You may not know exactly what the ping is when you start writing,
  but you must know when you finish

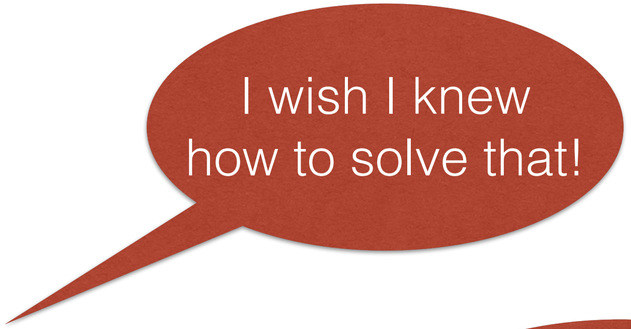
- If you have lots of ideas, write lots of papers

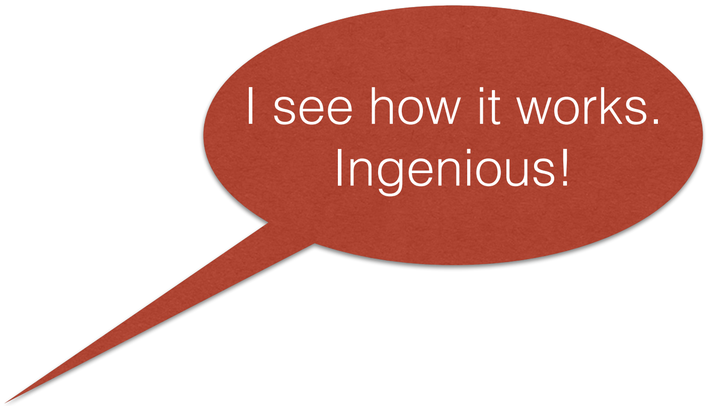
Can you hear the “ping”?

- Many papers contain good ideas, but do not distill what they are
- Make certain that the reader is in no doubt what the idea is.
  Be 100% explicit
  - “The main idea of this paper is …”
  - “In his section, we present the main contributions of the paper.”

Conveying the idea

- Here is a problem
- It's an interesting problem
- It's an unsolved problem
- **Here is my idea**
- My idea works (details, data)
- Here's how my idea compares to other people's approaches





Structure (conference paper)

- Title (1000 readers)
- **Abstract (4 sentences, 100 readers)**
- Introduction (1 page, 100 readers)
- The problem (1 page, 10 readers)
- My idea (2 pages, 10 readers)
- The details (5 pages, 3 readers)
- Related work (1-2 pages, 10 readers)
- Conclusions and further work (0.5 pages)

The abstract

- I usually write the abstract last
- Used by program committee members to decide which papers to read
- Four sentences [Kent Beck]
    1. State the problem
    2. Say why it’s an interesting problem
    3. Say what your solution achieves
    4. Say what follows from your solution

Example

1. Many papers are badly written and hard to understand
2. This is a pity, because their good ideas may go unappreciated
3. Following simple guidelines can dramatically improve the quality of your papers
4. Your work will be used more, and the feedback you get from others will in turn improve your research

Structure

- Title (1000 readers)
- Abstract (4 sentences, 100 readers)
- **Introduction (1 page, 100 readers)**
- The problem (1 page, 10 readers)
- My idea (2 pages, 10 readers)
- The details (5 pages, 3 readers)
- Related work (1-2 pages, 10 readers)
- Conclusions and further work (0.5 pages)

The introduction (1 page)

1. **Describe the problem**

2. **State your contributions**

   ...and that is all (according to Simon)
   I would add

3. Restate key items from the abstract
   (why the problem is interesting, etc)

   And use intuitive language (do the hallway test)

Molehills not mountains

- “Computer programs often have bugs. It is very important to eliminate these bugs [1,2]. Many researchers have tried [3,4,5,6]. It is really very important.”

Yawn

- “Consider this program, which has an interesting bug. <brief description>. We will show an automatic technique for identifying and removing such bugs.”

Cool!

Describe the problem

1 Introduction

Deterministic planning domains are generally easy to visualize and understand, as the details of the meaning of transitions between states are clearly defined in the operators, and the resulting plans are intuitive and easily understood. One particular formalism for domain representation in deterministic planning is the hierarchical task network (HTN) [Erol *et al.*, 1994], which encodes not only STRIPS/PDDL actions with their preconditions and effects, but also domain knowledge in the form of a hierarchy of tasks that can be refined from a high-level objective into the actions required in the environment. Conversely, one of the most widely studied formalisms for planning under uncertainty is the Markov decision process (MDP) [Bellman, 2003], in which the evolution of the environment is modeled as a Markov chain, and the goals of the planner are *implicitly* represented in a function that defines, for each state, the reward of executing a certain action. The definition of stochastic planning problems quickly becomes unwieldy as the number of state variables increase.[1] As the number of states goes up, so does the size of the transition probability tables, with problems requiring one such table for each action in the domain. As a consequence, although MDPs are an elegant mathematical formalism for representing stochastic domains, it is not straightforward for non-specialists to model domains using this formalism.

If possible, use an example to describe to problem

State your contributions

- Write the list of contributions first
- **The list of contributions drives the entire paper:**
  the paper substantiates the claims you have made
- Reader thinks “gosh, if they can really deliver this, that’s be exciting; I’d better read on”

State your contributions

Our goal is to use HTN models, which are more user-friendly, to automatically construct MDPs. In this paper we propose a step towards this overall aim, showing how to use HTNs to describe MDPs, thus allowing stochastic domains to be modeled using HTNs that are then translated into MDPs in order to be solved. Together with a simple model of action error, our conversion process allows efficient MEU planning over the state space induced by the HTN. The benefits of the approach are twofold: (a) reduction of the state space, and consequent reduction of the computational burden is beneficial since it enables the representation and solving of realistic planning problems, and (b) starting from a declarative representation makes planning more comprehensible to humans, while extending the representation to stochastic domains.

- Do not leave the reader to guess what your contributions are!

Bulleted list of contributions

Evidence

- Your introduction makes claims
- The body of the paper provides **evidence to support each claim**
- Check each claim in the introduction, identify the evidence, and forward-reference it from the claim
- Evidence can be: analysis and comparison, theorems, measurements, case studies

Contributions should be refutable

We describe the WizWoz system. It is really cool.	We give the syntax and semantics of a language that supports concurrent processes (Section 3). Its innovative features are...
We study its properties.	We prove that the type system is sound, and that type checking is decidable (Section 4)
We have used WizWoz in practice.	We have built a GUI toolkit in WizWoz, and used it to implement a text editor (Section 5). The result is half the length of the Java version.

No “rest of this paper is…”

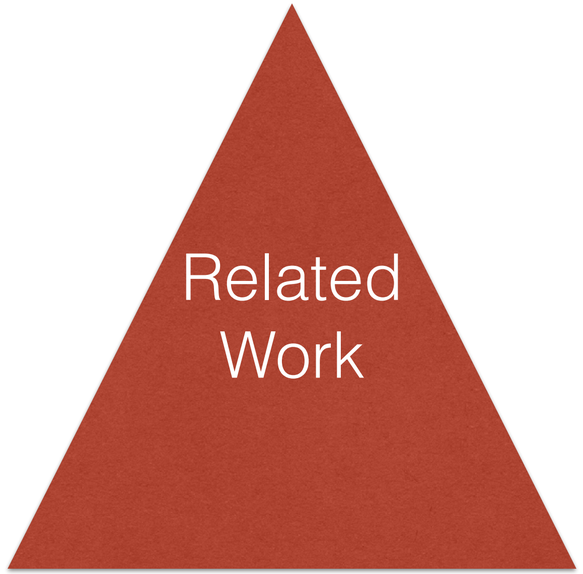
- Avoid signposting paragraphs like: "The rest of this paper is structured as follows. Section 2 introduces the problem. Section 3 ... Finally, Section 8 concludes”

- Instead, **use forward references from the narrative in the introduction**.
  The introduction (including the contributions) should survey the whole paper, and therefore forward reference every important part.

Structure

- Title (1000 readers)
- Abstract (4 sentences, 100 readers)
- Introduction (1 page, 100 readers)
- **The problem (1 page, 10 readers)**
- **My idea (2 pages, 10 readers)**
- **The details (5 pages, 3 readers)**
- Related work (1-2 pages, 10 readers)
- Conclusions and further work (0.5 pages)

No related work yet!



We adopt the notion of transaction from Brown [1], as modified for distributed systems by White [2], using the four-phase interpolation algorithm of Green [3]. Our work differs from White in our advanced revocation protocol, which deals with the case of priority inversion as described by Yellow [4].

No related work yet

- **Problem 1**: describing alternative approaches gets between the reader and your idea
- **Problem 2**: the reader knows nothing about the problem yet; so your (carefully trimmed) description of various technical tradeoffs is absolutely incomprehensible

Instead…

- Concentrate single-mindedly on a narrative that
  - **Describes the problem**, and why it is interesting
  - **Describes your idea**
  - **Defends your idea**, showing how it solves the problem, and filling out the details
- On the way, cite relevant work in passing, but defer discussion to the end

The payload of your paper

Consider a bufircuated semi-lattice D, over a hyper-modulated signature S. Suppose pi is an element of D. Then we know for every such pi there is an epi-modulus j, such that $p_j < p_i$.

- Sounds impressive...but
- Sends readers to sleep
- In a paper you **must** provide the details,
  but **first** convey the idea

The payload of your paper

Introduce the problem, and your idea, using

Examples

and only then present the general case

- Remember: explain as if you were speaking to someone using a whiteboard

Using examples

4.1 Parking World

In the example scenario, the agent's environment is a grid of cells as shown in Figure 1. The cell $(1, 1)$ is designated the start state, and cell $(5, 5)$ is designated the end state. The agent can move from cell to cell orthogonally and can also perform a null action (which leaves the agent in the same cell). In addition, the environment contains a 'no-parking cell' $(3, 3)$ in which stopping is prohibited. The agent receives a positive reward for reaching the exit state, and small negative rewards for visiting all cells other than the no-parking cell. If the agent stops in the no-parking cell and the violation of the norm is detected (i.e., the norm is enforced), the agent receives a sanction of -1. If the violation is not detected (i.e., the organisation is not enforcing the norm), the agent receives a small positive reward, i.e., violating the norm and parking illegally is beneficial.

Put an example as soon as possible! (Ideally on the introduction)

Some papers even have a dedicated scenario section!

Conveying the idea

- Explain it as if you were speaking to someone using a whiteboard
- Conveying the intuition is primary, not secondary
- Once your reader has the intuition, she can follow the details (but not vice versa)
- Even if she skips the details, she still takes away something valuable

Putting the reader first

- **Do not** recapitulate your personal journey of discovery. This route may be soaked with your blood, but that's not interesting to the reader

- Instead, choose the most direct route to the idea

Structure

- Title (1000 readers)
- Abstract (4 sentences, 100 readers)
- Introduction (1 page, 100 readers)
- The problem (1 page, 10 readers)
- My idea (2 pages, 10 readers)
- The details (5 pages, 3 readers)
- **Related work (1-2 pages, 10 readers)**
- Conclusions and further work (0.5 pages)

Related work

- **Fallacy:** To make my work look good, I have to make other people's work look bad

The truth: credit is not like money

Giving credit to others does not diminish the credit you get from your paper

- Warmly acknowledge people who have helped you
- Be generous to the competition. “In his inspiring paper [Foo98] Foogle shows.... We develop his foundation in the following ways...”
- Acknowledge weaknesses in your approach

Credit is not like money

Failing to give credit to others can kill your paper

- If you imply that an idea is yours, and the referee knows it is not, then either
  - You don’t know that it’s an old idea (bad)
  - You do know, but are pretending it’s yours (very bad)

Making sure related work is accurate

- A good plan: when you think you are done, send the draft to the competition saying “could you help me ensure that I describe your work fairly?”
- Often they will respond with helpful critique
  - Beware of known baddies (but this is **rare**)
- They are likely to be your referees anyway, so getting their comments up front is jolly good

The process

- Start early. Very early.
  - Hastily-written papers get rejected
  - Papers are like wine: they need time to mature
- Collaborate
  - Use SVN/Git/Mercurial/(Your flavour of VCS) to support collaboration

Getting help

Get your paper read by as many friendly guinea pigs as possible

- Experts are good
- Non-experts are also very good
- Each reader can only read your paper for the first time once! So use them carefully
- Explain carefully what you want (“I got lost here” is much more important than “wibble is mis-spelt”.)

Your supervisor's help

You also want the English to be correct
before you send the paper to your advisor

- You want your advisor to be able to critique your technical contribution
  - English errors are very distracting (to me at least)
  - More people know English than stochastic planning algorithms
- Get help from your friends reading each other's

Listening to your reviewers

Every review is gold dust
Be (truly) grateful for criticism
as well as praise

- This is really, really, really hard
- But it’s really, really, really, really, really, really important

Listening to your reviewers

- Read every criticism as a positive suggestion for something you could explain more clearly
- **DO NOT** respond “you stupid person, I meant X”. Fix the paper so that X is apparent even to the stupidest reader.
- Thank them warmly. They have given up their time for you.

Language and style

Basic stuff

- Submit by the deadline
- Keep to the length restrictions
  - Do not narrow the margins
  - Do not use 6pt font
  - On occasion, supply supporting evidence (e.g. experimental data, or a written-out proof) in an appendix
- Always use a spell checker

More Basic Stuff

- Never use jargon without explaining it first

Visual structure

- Give strong visual structure to your paper using
  - sections and sub-sections
  - bullets
  - italics
  - laid-out code (or algorithms)
- Find out how to draw pictures (vector graphics!!), and use them

Visual structure

```
1  @prohibitionStart(in(classifRoom))
2  +!Start : true
3   <- !findPlansWithEffect(in(classifRoom), SPlans);
4      !suppressPlans(SPlans);
5      +suppressedPlans(in(classifRoom),SPlans).
6
7  @prohibitionEnd(in(classifRoom))
8  +!End : suppressedPlans(in(classifRoom),SPlans)
9   <- !unsuppressPlans(SPlans);
10     .remove_plan(prohibitionStart(in(classifRoom)));
11     .remove_plan(prohibitionEnd(in(classifRoom))).
```

Listing 2: Plans generated from a state prohibition.

Plans to effect restrictions on executing actions are very similar to those relating to achieving world states, the only difference being in the process for selecting the plans that need to be suppressed. In this case, the plans searched for are those that contain a particular action. For example, if the cleaning agent might be obliged not to vacuum a table during its rounds of cleaning through the norm `+norm(time(800), day(xmas), prohibition(vacuum(table)))[source(env)]`. We do not include the example plans due to space constraints, but they should be obvious.

3.3 Norm expiration

Now that we have seen the plans needed to start complying with norms under several circumstances, we need to examine how an agent behaviour is modified as a result of a norm expiring. When an agent accepts a norm and changes its behaviour as a result of the norm becoming active, it either includes extra plans to comply with obligations or suppresses some of its plans in order to violate a prohibition. However, these behaviour modifications should not become permanent within an agent if the norms that caused them cease to be active. Moreover, our monotonicity assumption

Algorithm 6 Plan to react to the expiration of a prohibition

Require: Acceptance of $norm(Activ, Exp, prohibition(P))$
Require: Receipt of Exp event
Require: Label $L_{Activ,prohibition(P)}$ for a norm activation plan
Require: Plan library PL
Require: $S_{Plans,prohibition(P)}$ of suppressed plans
Ensure: Plan is uniquely labelled with label $L_{Exp,prohibition(P)}$
1: Unsuppress all plans from $S_{Plans,prohibition(P)}$
2: Remove plan $L_{Activ,prohibition(P)}$ from PL
3: Remove plan $L_{Exp,prohibition(P)}$ from PL

4. NORMATIVE AGENTSPEAK(L)

In order to test the viability of our solution in a practical agent language, we have developed an implementation of the strategies outlined in Section 3 using an AgentSpeak(L) interpreter. An important part of this involves the manipulation of an agent's own plan library, necessitating a means to perform meta-reasoning, allowing AgentSpeak(L) plans to manipulate other plans. With such a meta-reasoning facility in place, we can create AgentSpeak(L) plans that accomplish the norm-induced behaviour modification described above. We also point out that, while the plans shown in Section 3 use constructs that were not described in detail, this section clarifies all the plan constructs used throughout the paper.

4.1 Meta-reasoning for AgentSpeak(L)

The AgentSpeak(L) language does not have explicit constructs for the analysis of a plan library, yet this is required in the strategies described in Section 3 and implemented in Section 4.2. In particular, for an agent to evaluate its existing behaviours, encoded in

Visual Structure

Unifications can be *composed*; that is, for any substitutions $\sigma_1 = \{x_1/\tau_1, \ldots, x_n/\tau_n\}$ and $\sigma_2 = \{y_1/\tau'_1, \ldots, y_k/\tau'_k\}$, their composition, denoted as $\sigma_1 \cdot \sigma_2$, is defined as $\{x_1/(\tau_1 \cdot \sigma_2), \ldots, x_n/(\tau_n \cdot \sigma_2), z_1/(z_1 \cdot \sigma_2), \ldots, z_m/(z_m \cdot \sigma_2)\}$, where $\{z_1, \ldots, z_m\}$ are those variables in $\{y_1, \ldots, y_k\}$ that are not in $\{x_1, \ldots, x_n\}$. A substitution σ is a *unifier* of two terms τ_1, τ_2, if $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$.

Definition 4 (Unify Relation)**.** *Relation unify*(τ_1, τ_2, σ) *holds iff* $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$*. Moreover, unify*$(p(\tau_0, \ldots, \tau_n), p(\tau'_0, \ldots, \tau'_n), \sigma)$ *holds iff unify*$(\tau_i, \tau'_i, \sigma)$*, for all* $0 \leq i \leq n$*.*

Thus, two terms τ_1, τ_2 are related through the *unify* relation if there is a substitution σ that makes the terms syntactically equal. In our representation and algorithms, we adopt Prolog's convention [1] and use strings starting with a capital letter to represent variables and strings starting with a small letter to represent constants.

3.3 Commitments

Social commitments are extensively studied in multiagent literature [9, 11, 20]. Specifically, a commitment C(DEBTOR, CREDITOR, antecedent, consequent) means that a DEBTOR

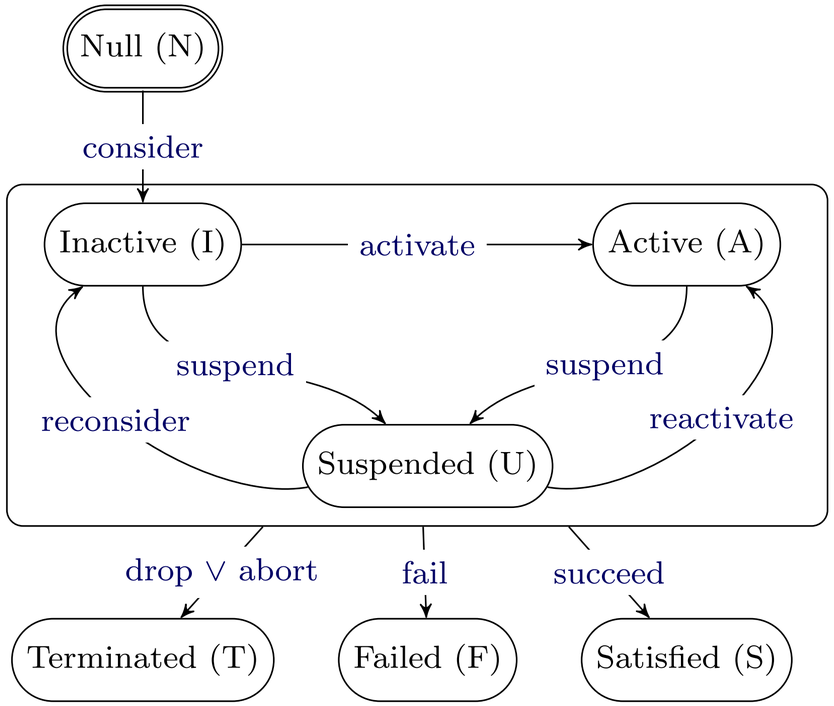


Figure 2: Goal lifecycle as a state transition diagram.

Use the active voice

The passive voice is “respectable” but it DEADENS your paper.
Avoid it at all costs.

No	Yes
It can be seen that...	We can see that...
34 tests were run	We ran 34 tests
These properties were thought desirable	We wanted to retain these properties
It might be thought that this would be a type error	You might think this would be a type error

“We" = you and the reader

“We" = the authors

“You" = the reader

Use simple, direct language

No	Yes
The object under study was displaced horizontally	The ball moved sideways
On an annual basis	Yearly
Endeavour to ascertain	Find out
It could be considered that the speed of storage reclamation left something to be desired	The garbage collector was really slow

But **do not** be informal/imprecise

No	Yes
The agent **can’t** call the move action	The agent **cannot** execute the move action
We **present** an algorithm	We **develop/introduce** an algorithm
The code **doesn't** run	The code **does not** run
The AAC diagram shows that	The *Arrows and Circles* (AAC) diagram shows that

Other Documents

- The guidelines here are not necessarily universal for all science documents, e.g:
  - Journal papers (similar but not identical)
  - Diploma theses (TC) / Masters dissertations / PhD theses
- You must be mindful of:
  - Who is the target readership
  - What is the purpose of the document you are writing

Journal Papers

- Very similar to conference papers, however:
  - No page limit (usually)
  - Presents mature research, for a wider audience
- Needs to be self-contained
  - More background, more related work
- Extensive results and comparison with related work

Purpose: Consolidate Research

Reader: Researchers in your broader area

Academic Milestones

- Undergrad, MSc, PhD
- Serve two purposes:
  - Convince your committee that you know your stuff
  - Show that you have achieved what you committed to do at TC1 / PEP / PT
- Needs to be self-contained
  - Make no assumptions about the reader
  - Needs to include signposting

Purpose: Convince a committee that you earned your degree

Reader: Whoever is available at the department + someone your advisor likes

The Elements of Style

- Read this book (brochure really):
  The Elements of Style: William Strunk, Jr. (1918)

- Many important tips on good style for writing in English.

Summary

1. Don’t wait, write
2. Identify your key idea
3. Tell a story
4. Nail your contributions
5. Related work: later
6. Put your readers first (examples)
7. Listen to your readers

More Resources

- Website Simon Peyton-Jones at Microsoft Research
  http://research.microsoft.com/en-us/people/simonpj/

- Resources at my website
  http://www.meneguzzi.eu/felipe/students.shtml
  http://www.meneguzzi.eu/felipe/research.shtml

- Courses on Scientific Writing
  http://www.escritacientifica.com/en/

- Writing scientific articles for Portuguese speakers
  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3935133/