

Art: free and constrained

vishvAs

October 23, 2011

Part I

Aesthetics

1 hindu theory

1.1 rasa and bhAva

Sentimental states embedded in objects of aesthetic experience can be described/ differentiated using bhAvas (state-properties). These bhAvas involve/ produce an experience of the corresponding rasas (sentiments). This is explained with analogy to the experience of different juices/ tastes while eating.

tatra vibhAva-anubhAva-vyabhichAri-sa.nyogAt rasaniShpattiH .
ko dR^iShTantaH . atrAha \-yathA hi
nAnA-vya~njanauShadhidravva-sa.nyogAt rasaniShpattiH tathA
nAnA-bhAvopagamAt rasaniShpattiH .

1.2 Realism in fantasy

Some art provides enjoyment by transporting the viewer to a different place and time, or by placing one in a state of deep empathy.

Realism in literature, and the techniques used therein, are considered in the survey of books read.

1.2.1 sahRRidaya-rasikaH

sahRRidayaH rasikaH vivRRitAyAH kathAyAH dUrE sthitaH ityataH artistic distance rakShati, ataH kathAyAM vividhAn rasAn sAnandaM bhOkTuM shaknOti. EvaM mUlataH rasikaH vairAgya-bhAvE sthitaH.

1.2.1.1 Importance of antecedents

The antecedents of a piece of art (especially the skill and other attributes that went into making it) strongly affect the degree to which we appreciate and value it. This is not just an artifact of reasoning - that it is actually felt in the brain is verified by MRI.

1.2.2 rasas

Well known lists include 8, 9 (abhinava-gupta) and 12 (bhOja) rasas. bharata's list is below.

shR~i~NgArahAsyakaruNA raudravIrabhayAnakAH .
bIbhatsAdbhutasa.ndnyau chetyaShTau nATye rasAH smR~itAH ..15..
ete hyaShTau rasAH prokta druhinena mahAtmanA .

1.2.2.1 shAnta-rasaH

abhinavaguptEna AchAryEna EShAH bhadratayA sthApitaH. Due to aesthetic distance, upon experiencing any of the other rasas, the rasika returns to shAnta-rasa.

1.2.3 bhAvas

bhAvas are of 3 kinds, as described below:

punashcha bhAvAnvakShyAmi sthAyī-sa~nchAri-sattvajAn.h ..16..

1.2.3.1 sthAyI

The following bhAvas correspond in order to the 8 rasas.

ratihAsashcha shokashcha krodhotsAhau bhaya.n tathA .
jugupsA vismayashcheti sthAyibhAvAH prakIrtitAH ..17..

vairAgya-bhAvaH shAnta-rasasya bhAvaH vairAgyaH. EShAH bhAvayuktE manastApaM, bhayaM, shrAntiM cha harati. vairAgya-bhAvaM prAptuM bhakti-bhAvaH upayOgashAI.

1.2.3.2 vyAbhicArI/ sañchArI

nirvedaglanisha~NkAkhyAstathAsuuyA madaH shramaH .
Alasya.n chaiva dainya.n cha chintAmohaH smR~itirdhR~itiH ..18..
vrIDA chapalata harSha Avego jaDatA tathA .
garvo viShAda autsukya.n nidrApsmArA eva cha ..19..
supta.n vibodho.amarShashchApi avahitthaM athograta .
matirvyAdhistathA unmadastathA maraNameva cha ..20..
trAsashchaiva vitarkashcha vidnyeyA vyabhichAriNaH .
trayastri.nshadamI bhAvAH samAkhyAtAstu nAmataH ..21..

1.2.3.3 sAttvika

These are involuntary.

```
stambhaH svedo.atha romA~nchaH svarabhedo.atha vepathuH .
vaivarNyaM ashru-pralaya ityaShTau sAtvikAH smR^itAH ..22..
```

1.2.4 utpatti-hEtu-rasa

Of these, 4 are utpatti-hetu-rasas, which give birth to some other rasa; so bhārata's 8 rasas can be paired. The nATyashAstra says:

```
shR^i~NgArAt hI bhavEt hAsyo,raudrAt cha karuNO rasaH .
vIrAt cha Eva adbhuta-utpattiH,bIbhatsAt cha bhayAnakaH
```

1.2.5 varNa and daivas

```
atha varNAH \-
shyAmo bhavati shR^i~NgAraH sito hAsyaH prakIrtitaH .
kapotaH karuNashchaiva rakto raudraH prakIrtitaH .. 42..
gauro vIrastu vidnyeyaH kR^ishhNashchaiva bhayAnakaH .
nIlavarNastu bIbhatsaH pItashchaivAdbhutaH smR^itaH .. 43..
atha daivatAni \-
shR^i~NgAro vishhNudevatyo hAsyaH pramathadaivataH .
raudro rudrAdhidaivatyaH karuNo yamadaivataH .. 44..
bIbhatsasya mahAkAlaH kAladevo bhayAnakaH .
vIro mahendradevaH syAdadbhuto brahmadaivataH .. 45..
```

1.3 rasa-vinyAsa

A certain gross portion of an artistic work has a complex arrangement of rasas (rasa-vinyAsa). Good rasa-vinyAsa is hard to accomplish. A brilliant example is sundara-kANDa in rAmAyaNa, where hanumat visits la NkA, observes sItA and rAvaNa, and in paintings pertaining to these encounters.

The highest form of art appreciation requires understanding not only the rasa-vinyAsa, but the difficulty in making good rasa-vinyAsa.

2 abhinaya

2.1 abhinaya and its forms

2.1.1 Components

```
A~Ngikau vAchikashchaiva hyAhAryaH sAtvikastatha .
chatvAro.abhinaya hyete vidnyeya nATyasa.nshrayAH .. 23..
```

2.1.1.1 A Ngika abhinayas

Various Asanas, karanas (body conformations) and mudras (hand conformations) convey various specific ideas.

2.1.2 Forms

This abhinaya may be presented live in nAtya, described vividly with words in kAvya or snapshots may be depicted in sculpture or paintings.

2.1.3 Enhancing realism

This effort often involves technology: enabling 3D vision, providing vibrations, spraying water.

Sometimes a fake news website is released along with a video with news and information from this other land. Eg: Hitch-hiker's guide to the galaxy encyclopedia.

2.2 Setting antecedents

Importance of art's antecedents has been explained for the general case. Establishing that the art has been produced with skill can be done in many forms of expression, like: impressionism, cubism etc..

2.3 Exploiting gaps in perception

Gaps in perception gives the artist more freedom in making the artwork, and the viewer experiences slight joy from understanding something non-trivial. But, this freedom must not be misused to produce work which communicates lack of skill in the artist.

Good examples: some hindu sculptures depict apsaras with limbs twisted in unnatural conformations. Faces in abstract art may not contain eyes, maybe misshapen.

2.3.1 Neurological basis

The act of perception by the brain involves significant processing to identify patterns where they need not exist. This is illustrated by several optical illusions (some of which include sensory illusions described elsewhere).

Also, identifying patterns activates the reward centers in the brain.

2.3.1.1 Examples

a] the one where in a canvas filled with white dots, one identifies a dalmatian dog (grouping). b] in a sketch some see a young woman, while some see an old woman.

2.4 Exaggerating distinctive features

Exaggerating distinctive features about a subject often makes the artwork more attractive. This is either done by the use of contrast, or by actual exaggeration of the feature/ body part.

2.4.1 Examples

In Hindu art, one sees archetypical male and female figures which exhibit these qualities. Females are depicted as having a very hourglass figure, posing in the tribhaṅga where the body is bent at the hip and at the chest. This results from subtracting the average man from the woman.

Necklace in a Hindu sculpture accentuates smoothness of skin.

2.4.2 Neurological basis

The animal brain tends to learn / form rules in terms of key features of a certain situation/ stimulus which leads to a certain reaction. In the language of behavior analysis, this is called peak-shift.

Mice trained to seek food in a rectangle in preference to a square tend to choose more narrower rectangles when given the choice (having learned ‘rectangularity’ rule).

Seagull chicks, which have an instinct for pecking at a red spot seagull’s beak respond with higher energy to rectangles with many red spots!

2.5 Understatement

Eg: A certain famous doodle of a female is more attractive and arousing than a playboy pinup.

2.5.1 Neurological basis

Brain likes to saving on computation when possible.

2.6 Visual metaphors

Use of visual metaphors/ symbols allow the artist to communicate a variety of deep meanings; also the viewer experiences joy from being able to understand these metaphors.

Eg: Mudras in dhArmika statues, motifs in abstract art.

3 Story-telling in various forms

rasa-vinyasa in literary works (long and short ones) and videos, and their various categories, are considered elsewhere. In both we find different characters,

situations and perspectives. Videos and written stories differ in the depth and detail with which characters are examined - the former ideally try to capture the essentials and important bits about a character, which can be hard.

Part II

Problem solving process

4 Prelude

4.1 Analogies

Solving a jigsaw puzzle. Connecting the dots. Building a lego-structure. Writing a story.

4.2 Role in research

4.2.1 Need for broader learning

There should be effort and focus spent on both acquiring knowledge and on solving difficult problems. One must do both daily. One must not lose the taste for the pursuit of tough problems.

4.3 Solutions: General view

4.3.1 Algebra and thought

All thought is actually symbol manipulation: it can all probably be expressed in terms of some extended second order logic. Our concepts are algebraic formulae. Our quests are also algebraic formulae whose details we want to flesh out.

If we have a good set of concepts, we can represent concepts suitably, and if we manipulate them well, we will be powerful thinkers. Being conscious of these traits of thought allows us to develop ourselves as powerful thinkers.

In engineering, at the finest level, our concepts often represent relationship between various quantities.

4.3.2 Elaborate arrangement of simple ideas

Most scientific discoveries involve an elaborate scaffolding of ideas. Each constituent of this scaffolding, is by itself very simple and usually well known/used elsewhere.

This understanding is useful when constructing solutions: one uses the 'divide and conquer' technique of solving simpler subproblems and combining these sub-solutions. Thus, one deals with complexity.

4.4 As a search

One is essentially exploring the space of solutions/ ideas for valuable ideas. So, one is being an explorer, and the enjoyment and joy of being an explorer is very appropriate.

4.4.1 Role of knowledge acquisition

Solving a problem is largely equivalent to learning about a problem deeply. So, ideas described in knowledge acquisition ref is useful. Now, you gather knowledge not just by reading but also in solution attempts, experiments, discussions etc..

4.5 Solving discipline and expectations

4.5.1 Parallelism and slowness

Do not give up easily. Human brain is slow, and it works in parallel. It works even when it is not conscious of doing so. Great mathematicians are always a potent inspiration in this regard. The way to solve hard problems, as learnt from tough assignments: Holding a problem in one's thoughts for a long duration (many hours to days); intensely probing it.

Be very sure that ye will solve the problem. Once sufficient knowledge is acquired: 5 minutes into the problem, everyone is the roughly the same.

Consider multiple problems during the same time-period. "You have to keep a dozen of your favorite problems constantly present in your mind, although by and large they will lay in a dormant state. Every time you hear or read a new trick or a new result, test it against each of your twelve problems to see whether it helps."

4.5.2 Regularity

Strictly spend a certain amount of time daily thinking about problems.

4.5.3 Evidence of capability for theory research

As I was fairly successful in solving tough colt assignment problems independently, upon spending proportionate time, I can solve colt research problems.

Also, I have felt the mathematician's rush/ high during middle and high school, after solving many problems during exams.

Came across papers where I think 'I could have done this'. Eg: Using boosting techniques in better hard core set construction. Using boosting to show relationship between hard core functions and hard functions.

The aura of exclusiveness and extraordinary ability to solve mathematical problems surrounding theorists is a scam. However, beauty and utility remains true.

5 Pre-solving

5.1 Preparation

If you are new to the field, learn about it well, solve many problems in the field to learn useful skills and knowledge.

See jJNAna-prApti survey for details.

5.1.1 Picking up a problem after a long time

If a research problem is picked up after a long time, reinvesting time to bring all of its aspects to the forefront of the brain is inescapable.

5.1.2 Analyze the properties of the problem domain

Use the human spatial reasoning skills and instincts. Draw diagrams.

Look for patterns, name properties.

5.1.2.1 Learn proof techniques

Observe common solution techniques by reading many good papers. Often, solving a problem is one of applying techniques introduced in other papers to one's problem.

5.1.3 Importance

A photographer spends far more time in preparation, than the 2 minutes of clicking his greatest photographs. Most time is spent in preparatory work. The solutions themselves are simple.

6 Solving

There are 2 phases in solving: getting wild ideas, avoiding self-deception.

6.1 Modeling and specification

6.1.1 Picking the right perspective

Often, when viewed from the right perspective, the problem turns out to be simple. The right perspective includes the right notation, the right models and theory. Eg: Observe the many examples narrated by Jayadeva mishrA. The problem is to find this perspective.

6.1.2 Toy Modelling

Make a toy model. This is very important. Throw away useless details. Abstraction is essential in mathematics. Identify clearly the variables you can tweak to yield the solution. This provides useful practice in the art, and also a skeleton to build up a solution to the real problem.

6.1.2.1 Experimental work: effectiveness of simplicity

In many practical problems, simple solutions work surprisingly well. Eg: See comments in the statistics and probabilistic models survey.

6.1.3 Accurate specification

Specify the problem as accurately as possible. Name all variables. Often the solution follows easily from the definitions of the objects involved in the problem.

6.1.4 Use the right notation and theory

Learn about related work and problems, especially in other intellectual circles. Probe it with thought and experiments. Start writing a report about your attempts to solve the problem.

Make a reference sheet which contains important relevant results and previous tricks used at solving related problems. It is especially useful in case of algebra to have a list of various properties of various objects. Berkant used the matrix cookbook's listing of kronecker product properties to develop a proof.

Draw graphs.

Use algebra: Understand what is going on. This highly clarifies things, makes them unambiguous. This is an important skill.

6.2 Study it

Put all the pieces of the puzzle before your eyes. What makes it different from other similar puzzles? Why is the solution not apparent? This phase is nothing else but knowledge acquisition. See jJNAna-prApti reference.

Talk to others.

6.2.1 Visualizing data

Observe importance of visualizing the data, gathering basic statistics. Helps you understand the behavior of your experiments.

6.3 Planning

6.3.1 Overview

This may be the design of a high level proof strategy, or an algorithm design. Remember that all ideas in science/ mathematics result from a are merely an elaborate arrangement/ scaffolding of simple ideas.

Creating a strategy to solve a problem is often more critically important than the ability to fill in the details. Hence, intuition, experience and visualization are very important.

6.3.2 Loosening rigor

Temporarily suspend rigor: this is important for letting the intuition lead you on. It will be brought back during the implementation phase.

6.3.2.1 Avoid presumption of success

A plan is not its implementation. A proof sketch is not a proof. Mathematical honesty is essential: see subsection on rigor.

Even senior mathematicians have made the mistake of wrongly claiming success before implementing their plan. Eg: Deolalikar. However, in rare cases like that of Perelman, this presumption has been valid.

6.3.3 Beginning

Guess a solution and check it. Or one can straight-away start by thinking of simpler problems (see chapter on 'Post solving attempts'), rather than start with a complicated problem'.

6.3.3.1 Planning the scaffolding

In addition, use the following general tricks:

Work backward.

See if a proof by induction is possible.

Try to prove the opposite, see what obstacles you face.

6.3.3.2 Equivalent problems

Find and solve equivalent (and simpler) formulations of the problem.

For example, one can prove the converse of the statement which one wants to prove, or prove a more abstract problem.

Prove that the negation is absurd: aka proof by contradiction of assumption.
(Aka reductio ad absurdum.)

6.3.3.3 Talk to others

They give you ideas unrelated to what you were thinking. They help you catch your mistakes.

6.3.3.4 Look at solutions to similar problems

Often one uses techniques introduced by others to solve one's problem.

6.3.4 Iterative solving

This is a common way of dealing with complexity. This is seen, for example, in solving the Rubik's cube, where people try to get one layer of the cube right at a time.

Eliminate possibilities.

This is also seen in proofs by induction and in Proof by solving cases.

Or iteratively improve solutions, understanding of the problem.

6.3.4.1 Iterative construction.

This is often done in constructions involved in many proofs: Example construction of a set of points shattered by a concept class.

Also, in case it is not clear how to make one part of the construction, one can often defer it to the end, constructing all the simpler parts first. This is seen, for example, in constructing primal-dual witnesses which certify the solution of an optimization problem.

6.3.5 Common proof techniques

Also see iterative problem solving techniques.

Also see Tricky.

6.3.6 Reusability, modularization

When designing or implementing an algorithm, it is desirable to modularize it so that these modules can then be easily used for other algorithms.

Similarly, when designing a proof scaffold, it is desirable to modularize it into interesting components, so that these lemmata can then be reused in other theorems. This also gives us a sort of tolerance for errors in the scaffolding/array of ideas: useful components can be salvaged even if the overall proof is wrong.

6.4 Implementation

6.4.1 Correctness/ Rigor

Just as the planning stage often involves temporary suspension of rigor, the implementation should involve sufficient amount of doubt to catch errors.

6.4.1.1 What is rigor?

A rigorous proof makes many implicit assumptions explicit. This helps you catch yourself making unreasonable assumptions. Also, when all assumptions are explicitly specified, generalization becomes possible.

Every step should follow from the rules of logic and from the statements of earlier theorems and axioms. "=" *really* means equals; "implies" really means implication.

6.4.1.2 Turn claims into lemmata

Whenever, in the process of a proof, you make a claim, write it down and prove it as a lemma. This is done in Algorithms research for example. This is extremely important in avoiding mistakes. Eg: Such errors were revealed during the examination by Plaxton; and when a mistake was detected by during argument with Wei about the correctness of the method to exactly compute $E[AUC]$.

6.4.1.3 Rigor is not verbosity

It is possible to be rigorous and very concise. Shortcuts like 'without loss of generality' need not be used at all!

6.4.1.4 Why rigor?

It helps you catch yourself making unreasonable assumptions. Badly specified theorems are ripe for misuse.

6.4.1.5 Current defects in rigor

Rigor lacking while maximizing, fully specifying functions.

6.4.2 Progress

It is important to make good progress, otherwise, collaborators may lose patience; and there may be duplication of effort by researchers elsewhere. If laborious experiments are involved, find good sources of encouragement and enthusiasm. See research strategy.

6.5 Check and critique solution

6.5.1 Checking correctness

6.5.1.1 Importance

This is extremely important.

You must be able to identify false trails. It is very easy to fool yourself.

You can also fool yourself into thinking that a true trail is false, for example due to silly mistakes in algebra.

6.5.1.2 Methods

Simulating helps reveal flaws in the logic.

Presenting it to others, rewriting it helps find flaws. Rewriting a solution neatly helps root out algebraic errors.

6.5.2 Finding elegance

A truly lasting solution is elegant. An inelegant solution often shows inadequate understanding of the problem or bad notation or ineffective theory.

The solution itself is often simple.

7 Experimentation

7.1 Importance

System building is an excellent source of problems.

It also works where theory does not.

Important in verifying theoretical predictions: the assumptions on which it is based may not hold in reality!

michael jordan indAnIM api code likhati.

7.2 Clear objectives

7.2.1 Probing for apparent flaws

Probe the proposed experiment for flaws analytically and with discussions prior to experimentation; otherwise, much effort could be wasted in testing fundamentally and obviously flawed ideas. Thence, distinguish between clearly flawed ideas/ algorithms, and promising techniques.

Especially, specify in as great detail as possible the goal the technique is trying to achieve; then consider the cases where the technique will fail.

7.2.1.1 Local objections

Mathematical proof usually has a chain structure: $A_0 \implies A_1 \dots$

Wrong proof of the step $A_i \implies A_{i+1}$.

(Almost) Circular arguments. $A_i \implies A_{i+1} \implies A_i$.

7.2.1.2 Global objections

From Tao: 'The other way to find an error in a proof is to obtain a "high level" or "global" objection, showing that the proof, if valid, would necessarily imply a further consequence that is either known or strongly suspected to be false. The most well-known (and strongest) example of this is the counterexample. If one possesses a counterexample to the claim $A \text{--} \text{;} E$, then one instantly knows that the chain of deduction " $A \text{--} \text{;} B \text{--} \text{;} C \text{--} \text{;} D \text{--} \text{;} E$ " must be invalid, even if one cannot immediately pinpoint where the precise error is at the local level. Thus we see that global errors can be viewed as "non-constructive" guarantees that a local error must exist somewhere.'

However, a global objection need not be fatal: '(There is a mathematical joke in which a mathematician is giving a lecture expounding on a recent difficult result that he has just claimed to prove. At the end of the lecture, another mathematician stands up and asserts that she has found a counterexample to the claimed result. The speaker then rebuts, "This does not matter; I have two proofs of this result!". Here one sees quite clearly the distinction of impact between a global error and a local one.)'

Easier to detect It is also a lot quicker to find a global error than a local error, at least if the paper adheres to established standards of mathematical writing. To find a local error in an N-page paper, one basically has to read a significant fraction of that paper line-by-line, whereas to find a global error it is often sufficient to skim the paper to extract the large-scale structure.

7.2.2 Understanding results

Visualizing data helps understand behaviour of pattern recognition experiments.

7.2.2.1 Forming intuition

When experiments fail or behave unexpectedly, develop or repair intuition: question and try to understand what went wrong.

7.3 Engineering in science

It is profitable to view experimentation as good engineering projects. Thence, minimal time is wasted.

7.3.1 Enthusiasm maintenance

Experimentation/ engineering can be tedious or boring. It is important to find ways to over come it; eg: teamwork.

Experimentation with a computer can actually be fun - you don't have to sacrifice the constant use of pen and paper.

7.3.2 Programming and simulations

See programming languages ref.

8 Post solving-attempts

if the problem is not completely solved, it is very important to gain insights into problems and solution approaches. merely discovering partial solutions which are better than other solutions is rarely sufficient for publishing and influencing research and researchers. Looking at a class of partial solutions systematically and insightfully, observing them in unifying ways as answers to the same well motivated models, identifying commonalities among them: all these enable their comparison and the development of improved solutions. Eg: In the group recommendation project, we were attempting to solve a well-motivated problem using disparate methods, but we did not look at these systematically in unifying ways. As a result, our discoveries were not very interesting or impressive, and experiments were very randomly directed. Inderjit pointed this weakness out, and zhengdong fixed these with his insights into the problems.

8.1 Fail many times (if no plan worked)

8.1.1 Importance

This is good: you have a truly interesting problem. Failure leads to better understanding of the problem.

Theorem and proof are often developed **together**, starting with a hunch.

8.2 Failure Response

8.2.1 Keep trying

Energetically keep trying. It is good to rapidly come up with hypotheses and test them.

8.2.1.1 Remember it when working on other things

When you get read a related paper, without fail, try to see how it can be applied to solve one of the problems on which you are stuck. This is crucial.

8.2.2 Write about attempt

If you don't get the solution, write about your attempt. State the obstacle or question clearly: This is important. Everything is in place except these missing parts. You may come upon those parts later, from intuition or interaction.

8.2.3 Talk to others

Talking to others is important. Research shows that important, creative ideas develop when people are talking to each other in group meetings or on other occasions, where one person explains and complains about a research obstacle he is facing: usually another person suggests a new perspective for looking at/tackling the obstacle.

8.2.4 Solve simpler problem

Turn off all difficulties except few (one?).

Consider special cases or limiting conditions. This is often done by dropping or adding constraints. One can also consider limiting cases.

Prove a consequence first.

8.2.5 Solve related problem

When you are stuck, or when you find that others have satisfactorily solved the problem, alter the problem into something no one has considered, argue why it is interesting and be the first to solve it! pratika did this with the coclustering problem.

Consider continuous versions of the problem.

8.2.6 Partial solutions: gain insights

if the problem is not completely solved, it is very important to gain insights into problems and solution approaches. merely discovering partial solutions which are better than other solutions is rarely sufficient for publishing and influencing research and researchers. Looking at a class of partial solutions systematically and insightfully, observing them in unifying ways as answers to the same well motivated models, identifying commonalities among them: all these enable their comparison and the development of improved solutions. Eg: In the group recommendation project, we were attempting to solve a well-motivated problem using disparate methods, but we did not look at these systematically in unifying ways. As a result, our discoveries were not very interesting or impressive, and experiments were very randomly directed. Inderjit pointed this weakness out, and zhengdong fixed these with his insights into the problems.

8.3 Review/ extend/ generate new conjectures

Do numerical calculations (used heavily by Ramanujan among others) and computer simulations.

Find new problems related to this problem.

Look for equivalent problems in other disciplines. Geometrize the theory/insight.

Look for related problems in other disciplines.

Record new solution strategies.

9 Computing resources

9.1 At UTCS

Recovering deletions: `.snapshot`.

9.1.1 For Help

Email the right address: see the webpage for the addresses.

Visit the FAQ pages.

9.1.2 Important locations

Can install software from source on
`/p/graft` and then run
`graft -n -i -t /p /p/graft/path`.
`/projects/textmine`.

9.1.2.1 Cluster machines

On cluster machines and carrion:
`/scratch/cluster/vvasuki` .

Clusters at `lhug-0..`, `uvanimor-0...`

9.2 Outside UTCS

9.2.1 TACC computer cluster

There are login nodes, and there are computing nodes. <https://portal.tacc.utexas.edu/gridsphere/gridsphere?cid=allocations> has a list of login machines, each corresponding to clusters of different sorts of nodes.

9.2.1.1 Getting help

Open a ticket.

10 References

"Conversations on mathematics with a visitor from outer space" by David Ruelle.

- * "How to solve it?" by George Polya.

- * "What is mathematics?" by Tim Gowers.