Representation and Reasoning for Intelligent Systems

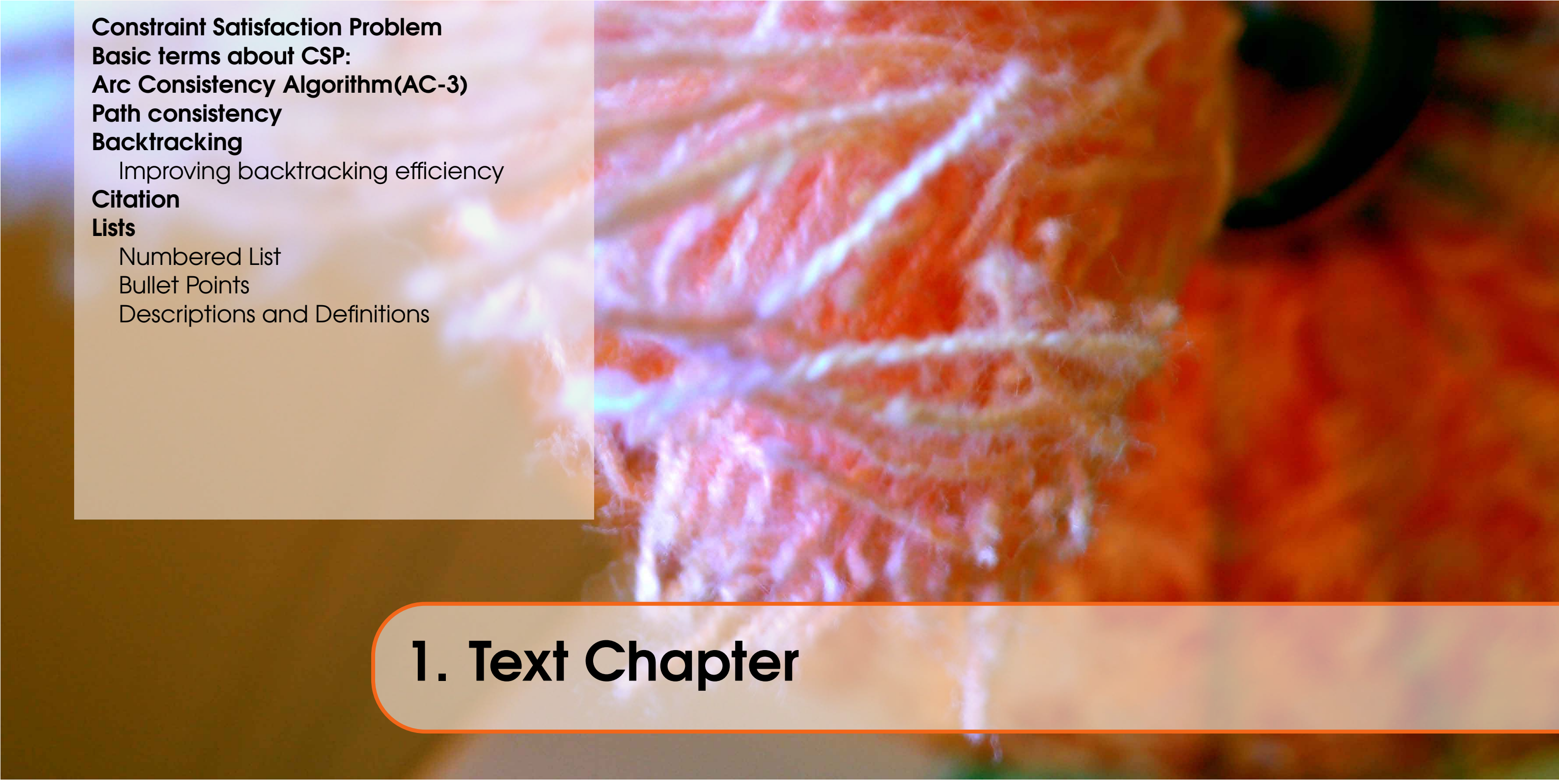
Stefan Klaus

NOT PUBLISHED

BOOK-WEBSITE.COM

First printing, March 2013

Contents



1. Text Chapter

1.1 Constraint Satisfaction Problem

Constraint satisfaction problems(CSP) are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations. One example of a CSP is the game “Sudoku”

1.2 Basic terms about CSP:

- an assignment is to assign values to some or all variables
- an assignment that does not violate any constraints is called a consistent assignment. With values from domain D_i
- A complete assignment is one in which each variable is assigned
- a partial assignment is one that assigns value to some of the variables

Constraint graph

Binary CSP: each constraint relates at most two variables, e.g: WASA constraint graph: nodes are variables(e.g. region WA), arcs show constraints(e.g. WASA)

Varieties of Variables

Discrete variables

- Finite Domains
- boolean CSPs, include: boolean satisfiability(NP - complete)
- Sudoku
- Infinite Domains(Integers, Strings, etc.)
- job scheduling, variables are start/end days for each job, need a constraint language, e.g. $StartJob_1 + 5 \leq StarJob_3$

Continuous variables

- start /end times for Hubble Telescope observations
- Unary constraints involve a single variable
  - $SA \neq Green$

- binary constraints involve pairs of variables
  - $SA \neq WA$
- Higher-order constraints involve 3 or more variables
  - cryptarithmetic column constraints
- Preferences(soft) constraints
  - red is better than green
  - CSPs with preference are often with optimization search algorithms constraints optimization problems

Node Consistency

If a node is node-consistent if all the value's domain satisfy the variable's unary constraints.

Example:
D = Red, Green, Blue
Variable X
X dislikes green, then X starts with D = Red, Green, Blue, and becomes node consistent after eliminating Green. X is node consistent with the reduced domain, D=Red, Blue.

Arc Consistency

X_i is arc-consistent with respect to another variables X_j if for every value in X_i's current domain D_i there is one value in X_i's domain D_j that satisfies the binary constraint on the arc (X_i, X_j)

Example:
Given two variables X_i , X_j with values in 0,1,2,...9 and constraints (0,0),(1,1),(2,4),(3,9).
To make X_i arc-consistent with respect to X_j, we reduce X_i's domain to 0,1,2,3.
To make X_j arc-consistent with respect to X_i, we reduce X_j's domain to 0,1,4,9.

1.3 Arc Consistency Algorithm(AC-3)

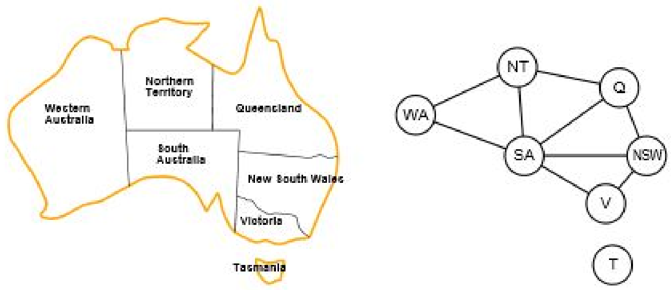
```
function AC-3(csp) returns false if an inconsistency is found and true otherwise
  inputs: csp, a binary CSP with components (X, D, C)
  local variables: queue, a queue of arcs, initially all the arcs in csp

  while queue is not empty do
    (Xi, Xj) ← REMOVE-FIRST(queue)
    if REVISE(csp, Xi, Xj) then
      if size of Di = 0 then return false
      for each Xk in Xi.NEIGHBORS - {Xj} do
        add (Xk, Xi) to queue
  return true
```

```
function REVISE(csp, Xi, Xj) returns true iff we revise the domain of Xi
  revised ← false
  for each x in Di do
    if no value y in Dj allows (x,y) to satisfy the constraint between Xi and Xj then
      delete x from Di
      revised ← true
  return revised
```

1. initially let a queue contain all arcs

2. remove an arc (X_i, X_j) from the queue and make the variable X_i arc-consistent to X_j
   (a) IFX_i domain D_i is unchanged, then check the next arc in the queue
   (b) IF X_i 's domain D_i is revised(smaller), then add all arcs (X_k, X_i)in the queue
   (c) If X_i 's domain D_i is empty, then CSP no solution
3. Keep checking all arcs in the queue until the queue is empty

1.4 Path consistency



A two variable set X_i, X_j is path consistency with respect to a third variable X_m if for every assignment $X_i = a, X_j = b$ consistent with the constraints onX_i, X_j, where is an assignment to X_m ,that satisfies the constraints onX_i, X_m and X_m, X_j

Example:
Can we color the Australia map with two colors?
Make the set WA, SA path consistent with respect to NT?
Assignments: WA = blue, SA = red or WA =blue,SA=red
But no assignment exists for NT

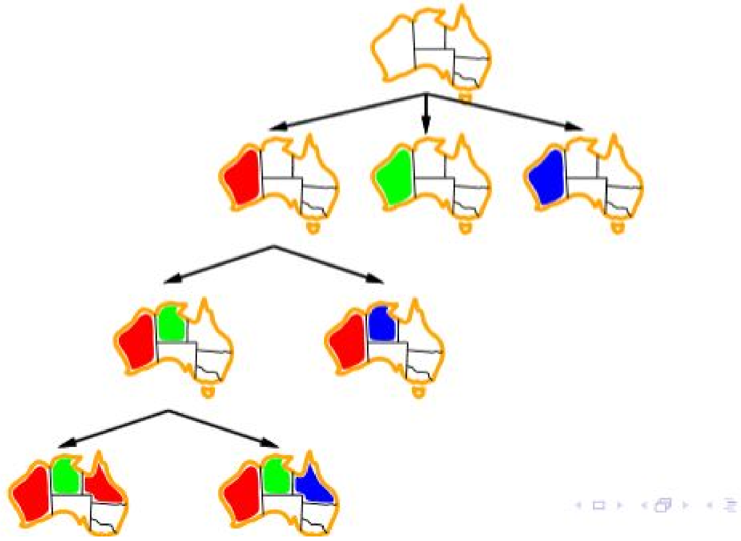
Constraint propagation

- constraint propagation is a specific type of inference
  - use the constraints to reduce the number of legal values for a variable, which in turn can reduce the legal values for another variable, and so on
- node consistency
- arc consistency
- path consistency
- k-consistency: k variables involved
- Global consistency

Limits

- Indeed AC-3 works for the easiest Sudoku puzzles
- slightly harder ones can be solved by PC-2, but at a greater computational cost: there are 255,960 different path constraints to consider in a Sudoku puzzle
- To solve the hardest puzzles and to make efficient progress, we will have to be more clever

1.5 Backtracking



1. Select an unassigned value
2. assign values
3. Depth-first search
4. If an inconsistency is detected, then BACKTRACK returns failure, causing the previous call to try another value

Backtracking search is a depth-first search for CSPs

- choose values for one variable at a time
- backtrack when a variables has no legal left to assign

Backtracking search is the basic uninformed algorithm for CSP. Can solve n-queens for $n \approx 25$.

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return BACKTRACK({ }, csp)

function BACKTRACK(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
     if value is consistent with assignment then
        add {var = value} to assignment
        inferences ← INFERENCE(csp, var, value)
        if inferences ≠ failure then
          add inferences to assignment
          result ← BACKTRACK(assignment, csp)
          if result ≠ failure then
            return result
     remove {var = value} and inferences from assignment
  return failure
```

1.5.1 Improving backtracking efficiency

General-purpose methods can give huge gains in speed:

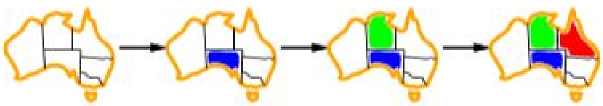
1. which variable should be assigned next?
2. in what order should values be tried‘?
3. can we detect inevitable failure early?

4. Can we take advantage of problem structure?

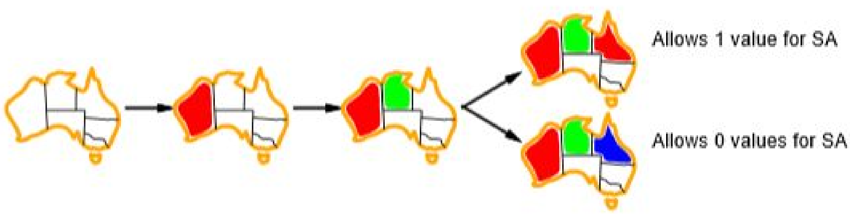
Minimum remaining values
Minimum remaining values (MRV): choose the variable with the fewest legal values.

Degree heuristic

- Tie-breaker among MRV variables
- Degree heuristic: choose the variable with the most constraints on remaining variables

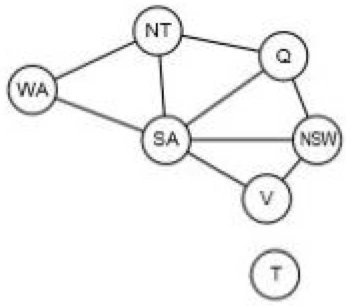


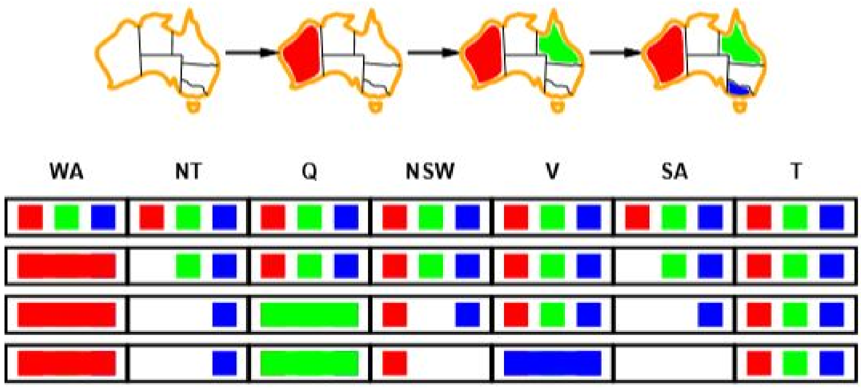
Least constraining value
Given a variable, choose the least constraining value: the one that rules out the fewest values in the remaining variables.



Combining these heuristics makes 1000 queens feasible.

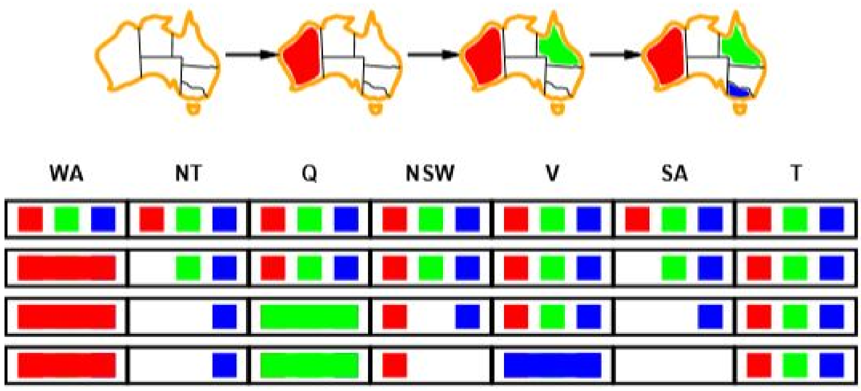
Forward checking
Idea: Keep track if remaining legal values for unassigned variables.
Terminates search when any variable has no legal values.





Constraint propagation

Forward checking propagates information from assigned to unassigned variables, but doesn't provide early detection for all failures:



NT and SA cannot both be blue!

Constraint propagation repeatedly enforces constraints locally.

Arc consistency

Simplest form of propagation makes each arc consistent.

$X \to Y$ is consistent if for every value x of X there is some allowed y

1.6 Citation

This statement requires citation [Smi12]; this one is more specific [Smi13, page 122].

1.7 Lists

Lists are useful to present information in a concise and/or ordered way[1].

1.7.1 Numbered List

1. The first item
2. The second item
3. The third item

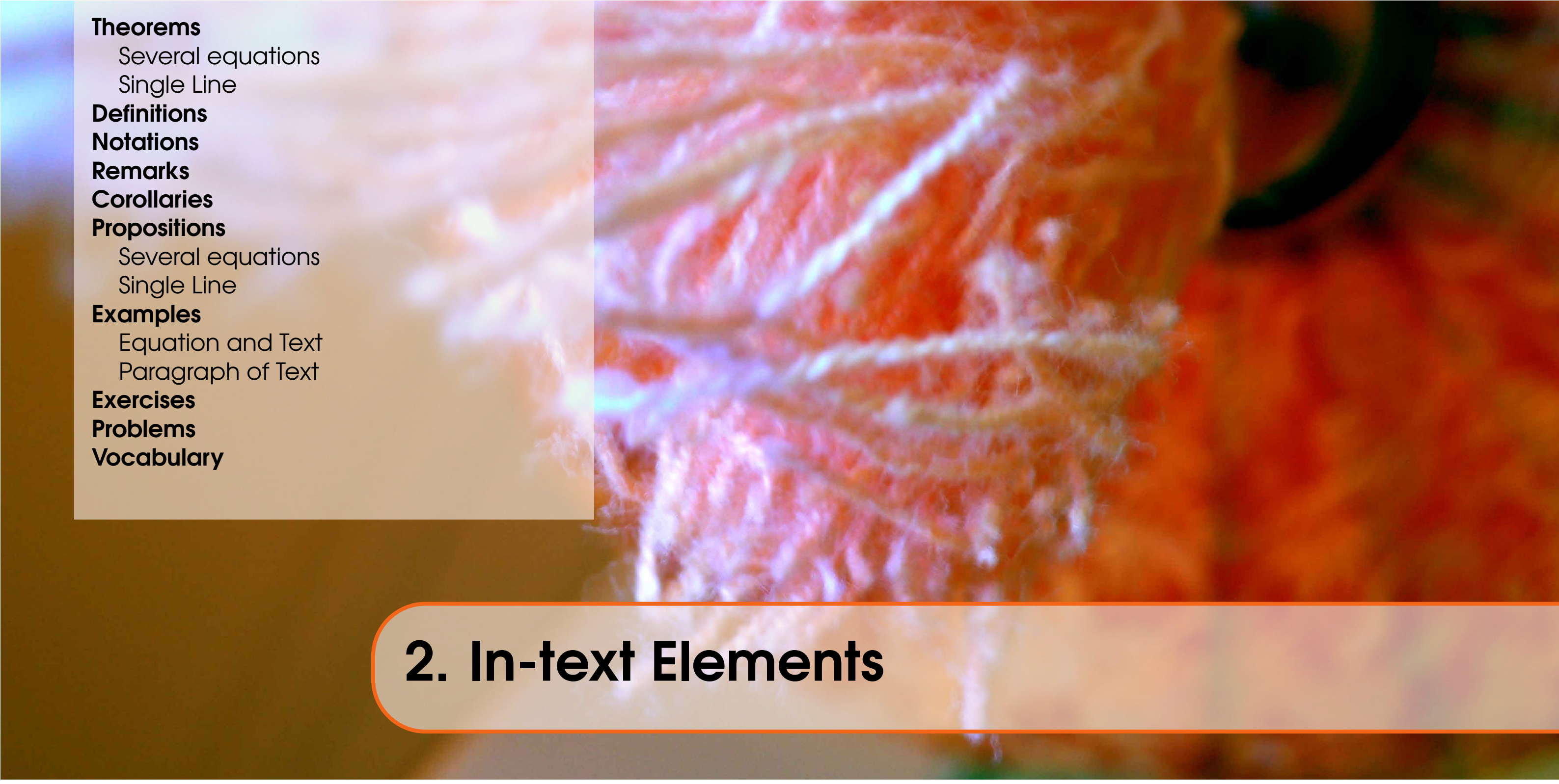
1.7.2 Bullet Points

- The first item
- The second item
- The third item

1.7.3 Descriptions and Definitions

Name Description
Word Definition
Comment Elaboration

[1] Footnote example...



2. In-text Elements

2.1 Theorems

This is an example of theorems.

2.1.1 Several equations

This is a theorem consisting of several equations.

Theorem 2.1.1 — Name of the theorem. In $E = \mathbb{R}^n$ all norms are equivalent. It has the properties:

$$\big| ||\mathbf{x}|| - ||\mathbf{y}|| \big| \leq ||\mathbf{x} - \mathbf{y}|| \tag{2.1}$$

$$||\sum_{i=1}^{n} \mathbf{x}_i|| \leq \sum_{i=1}^{n} ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \tag{2.2}$$

2.1.2 Single Line

This is a theorem consisting of just one line.

Theorem 2.1.2 A set $\mathscr{D}(G)$ in dense in $L^2(G)$, $|\cdot|_0$.

2.2 Definitions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Definition 2.2.1 — Definition name. Given a vector space E, a norm on E is an application, denoted $||\cdot||$, E in $\mathbb{R}^+ = [0, +\infty[$ such that:

$$||\mathbf{x}|| = 0 \;\Rightarrow\; \mathbf{x} = \mathbf{0} \tag{2.3}$$

$$||\lambda \mathbf{x}|| = |\lambda| \cdot ||\mathbf{x}|| \tag{2.4}$$

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \tag{2.5}$$

2.3 Notations

Notation 2.1. *Given an open subset G of $\mathbb{R}^n$, the set of functions φ are:*

1. *Bounded support G;*
2. *Infinitely differentiable;*

a vector space is denoted by $\mathscr{D}(G)$.

2.4 Remarks

This is an example of a remark.

R The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.5 Corollaries

This is an example of a corollary.

Corollary 2.5.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.6 Propositions

This is an example of propositions.

2.6.1 Several equations

Proposition 2.6.1 — Proposition name. It has the properties:

$$\big| ||\mathbf{x}|| - ||\mathbf{y}|| \big| \leq ||\mathbf{x} - \mathbf{y}|| \tag{2.6}$$

$$|| \sum_{i=1}^{n} \mathbf{x}_i || \leq \sum_{i=1}^{n} ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \tag{2.7}$$

2.6.2 Single Line

Proposition 2.6.2 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathscr{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

2.7 Examples

This is an example of examples.

2.7.1 Equation and Text

■ **Example 2.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1, 1)$; consider the function:

$$f(x) = \begin{cases} e^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \tag{2.8}$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \varepsilon\}$ for all $\varepsilon \in]0; 5/2 - \sqrt{2}[$. ■

2.7.2 Paragraph of Text

■ **Example 2.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

■

2.8 Exercises

This is an example of an exercise.

Exercise 2.1 This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

2.9 Problems

Problem 2.1 What is the average airspeed velocity of an unladen swallow?

2.10 Vocabulary

Define a word to improve a students' vocabulary.
Vocabulary 2.1 — Word. Definition of word.

3. Presenting Information

3.1 Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 3.1: Table caption

3.2 Figure

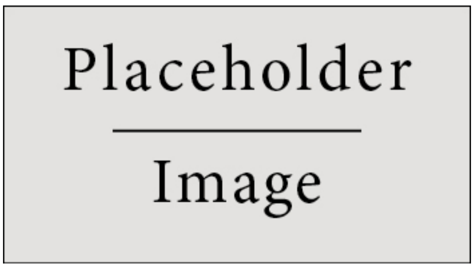


Figure 3.1: Figure caption

Bibliography

Books

[Smi12] John Smith. *Book title*. 1st edition. Volume 3. 2. City: Publisher, Jan. 2012, pages 123–200 (cited on page 10).

Articles

[Smi13] James Smith. “Article title”. In: 14.6 (Mar. 2013), pages 1–8 (cited on page 10).

Index