A Philosophy of Domain Science & Engineering∗
An Interpretation of Kai Sørlander’s Philosophy
An Incomplete Work in Progress Research Report

Dines Bjørner†
Fredsvæj 11, DK-2840 Holte, Danmark
E--Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~db‡

May 20, 2018: 11:20 am

Abstract

We show how the domain analysis & description calculi of [1] satisfy Kai Sørlander’s Philosophy, but also that Sørlander’s Philosophy, notably [2] and [3] mandates extensions to the calculi in order to form a more consistent “whole”. Where discrete parts were just that, we must now distinguish between three kinds of parts: (i) physical parts, (ii) living species parts, and (iii) artifacts. (i) The physical parts are not made by man, but are in space and time; these are endurants that are subject to the laws of physics as formulated by for example Newton and Einstein, and also subject to the principle of causality and gravitational pull – but were not so explicated. They are the parts we treated in [1]. (ii) The living species parts are plants and animals; they are still subject to the laws and principles of physics, but additionally unavoidably endowed with such properties as causality of purpose. Animals have sensory organs, means of motion, instincts, incentives and feelings. Among animals we single out humans as parts that are further characterisable: possessing language, learning skills, being consciousness, and having knowledge. These aspects were somehow, by us, subsumed in our analysis & description by partially endowing physical parts with such properties. (iii) Then there are the parts made by humans, i.e., artifacts. Artifacts have a usual set of attributes of the kind physical parts can have; but in addition they have a distinguished attribute: attr_intent – expressed as a set of intents by the humans who constructed them according to some purpose. This more-or-less “standard” property of intents determines a form of counterpart to the gravitational pull of physical parts namely, what we shall refer to as intentional “pull”. Also these were subsumed in [1] – by either partially endowing physical parts with such properties, or by ignoring them!

We thus suggest a philosophy basis for domain science & engineering. This paper is based on recent research [4, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] into methods for analysing and describing human-centered universes of discourses such as transport nets, container

∗First reading: The Victor Ivannikov Memorial Event, May 3–4, 2018, Yerevan, Armenia
†Margin numbers refer to slide numbers of a lecture (series) version of this paper.
In the first part of the paper we present two calculi, one for analysing manifest “worlds” and one for describing those “realities”. And we “interpret” manifest endurant entities as behaviours i.e., as perdurants. This interpretation is, from the point-of-view of post-Kantian philosophy, a transcendental deduction, i.e. cannot be logically explained, but can be understood meta-physically. In a more-or-less summary section we shall then show that the calculi are necessary and sufficient, in that they have a basis in philosophical reasoning. But, what is to us more interesting, we show how the Sørland Philosophy “kicks back” and either mandates or requires domain properties not covered in my earlier papers on the domain analysis & description method [4, 1].

Initial versions of this document are in the form of a report. As such it collects far more material than should be contained in a proper paper. Most of the “extra” report material is collected from various sources but drastically edited by me. Most of the material of Sect. 9 is extracted from [18] some from [15, 21, 22, 23].

Contents

1 Introduction ................................................. 8
  1.1 Two Views of Domains .................................. 9
    1.1.1 The Computing Science View ..................... 9
    1.1.2 The Philosophy View .............................. 9
    1.1.3 First Two Independent Treatments, then An Interpretation .................................. 10
  1.2 The Computing Science Background .................. 10
    1.2.1 Computer & Computing Science .................. 10
    1.2.2 Formal Methods .................................. 11
    1.2.3 A Triptych of Engineering ....................... 11
  1.3 Domains, their Analysis & Description, and a Method ................................................. 11
    1.3.1 Domain Analysis & Description .................. 11
    1.3.2 A Domain Analysis & Description Method ....... 12

2 Endurants – cf. s. 6.2 Pg. 36 .................................. 12
  2.1 The Universe of Discourse – cf. s. 6.1 Pg. 36 ....... 12
  2.2 Basic Domain Concepts ................................ 12
  2.3 An Upper Ontology Diagram of Domains – A Preview ................................................. 14
  2.4 Structures – cf. s. 6.2.1 Pg. 36 ......................... 14
  2.5 Parts, Components and Materials – cf. s. 6.2.2 Pg. 36 ................................................. 15
    2.5.1 Parts – cf. s. 6.2.3 Pg. 37 .......................... 15
    2.5.2 Components – cf. s. 6.2.4 Pg. 37 .................. 18
    2.5.3 Materials – cf. s. 6.2.5 Pg. 37 .................... 20
  2.6 Unique Part and Component Identifiers – cf. s. 6.2.7 Pg. 37 ................................................. 21
  2.7 Part Mereologies – cf. s. 6.2.9 Pg. 38 ................ 22
    2.7.1 Part Relations ....................................... 22

1 Other Sørland books are [2, 19, 20, 3]
### 2.7.2 Part Mereology: Types and Functions

### 2.8 Part Attributes

- cf. s. 6.2.10 Pg. 39

### 2.8.1 Inseparability of Attributes from Parts and Materials

### 2.8.2 Attribute Quality and Attribute Value

### 2.8.3 Part and Material Attributes: Types and Functions

### 2.8.4 Attribute Categories

#### 3 A Transcendental Transformation

- cf. s. 6.3 Pg. 40

#### 4 Perdurants

- cf. s. 6.4 Pg. 41

##### 4.1 States

- cf. s. 6.2.6 Pg. 37

##### 4.2 On Actions, Events, Behaviours and Actors

- cf. s. 6.4.2 Pg. 41

- cf. s. 6.4.4 Pg. 42

#### 4.3 Channels

- cf. s. 6.4.2 Pg. 41

#### 4.4 Behaviours

- cf. s. 6.4.3 Pg. 41

- cf. s. 6.4.4 Pg. 42

#### 4.5 Initial Running Systems

- cf. s. 6.4.5 Pg. 43

#### 5 A Coin Has Two Sides

#### 6 An Example: A Road Transport System

##### 6.1 The Universe of Discourse

- cf. s. 2.1 pp. 12

##### 6.2 Endurants

- cf. s. 2 pp. 12

- cf. s. 2.4 pp. 14

- cf. s. 2.5 pp. 15

- cf. s. 2.5.3 pp. 20

- cf. s. 4.1 pp. 29

- cf. s. 2.6 pp. 21

- cf. s. 2.6.7 Unique Identifiers

- cf. s. 2.6.7.2 Part Identifiers

- cf. s. 2.6.7.2.1 Extract Parts from Unique Identifiers

- cf. s. 2.6.8 Uniqueness of Part Identifiers

- cf. s. 2.6.9 Part Mereologies

- cf. s. 2.6.10 Part Attributes

- cf. s. 2.6.11 Discussion of Endurants, I

- cf. s. 2.6.12 Discussion of Endurants, II

- cf. s. 2.6.13 Indexed States

- cf. s. 2.6.13.1 Constants:

- cf. s. 2.6.13.2 Indexed States:

- cf. s. 2.6.14 Channels

- cf. s. 2.6.14.1 Behaviour Signatures

- cf. s. 2.6.14.1.1 Behaviour Definitions

- cf. s. 2.6.14.2 A Running System

- cf. s. 2.6.14.2.1 Preliminaries

- cf. s. 2.6.14.2.2 Starting Initial Behaviours

- cf. s. 2.6.14.3 Space and Time Considerations: A Specific Critique

- cf. s. 2.6.14.3.1 Space

**An Interpretation of Kai Sørlander’s Philosophy**

9.5 The Empirical Tradition: Locke, Berkeley and Hume .................................................. 55
  9.5.1 John Locke: 1632–1704 .......................................................................................... 55
  9.5.2 George Berkeley: 1685–1753 .......................................................... 56
  9.5.3 David Hume, 1711–1776 .................................................. 56
9.6 Immanuel Kant: 1720–1804 .................................................. 56
9.7 Post-Kant ........................................................................... 57
  9.7.1 Johann Gottlieb Fichte, 1752–1824 .......................................................... 57
  9.7.2 Georg Wilhelm Friedrich Hegel, 1770–1831 .................................................. 57
  9.7.3 Friedrich Schelling, 1775–1854 .................................................. 58
  9.7.4 Friedrich Ludwig Gottlob Frege, 1848–1925 .................................................. 58
  9.7.5 Edmund Husserl, 1859–1938 .......................................................... 58
  9.7.6 Bertrand Russell, 1872–1970 .......................................................... 58
  9.7.7 Logical Positivism: 1920s–1936 .................................................. 58
  9.7.8 Ludwig Wittgenstein, 1889–1951 .......................................................... 59
9.8 Bertrand Russell – Again! .................................................. 59

10 The Kai Sørlander Philosophy .................................................. 59
  10.1 The Basis ........................................................................ 59
    10.1.1 The Inescapable Meaning Assignment .................................................. 59
    10.1.2 Necessary and Empirical Propositions .................................................. 61
    10.1.3 Primary Objects ........................................................................ 61
    10.1.4 Two Requirements to the Philosophical Basis .................................................. 61
    10.1.5 The Possibility of Truth ........................................................................ 61
    10.1.6 The Logical Connectives ........................................................................ 61
    10.1.7 Necessity and Possibility ........................................................................ 62
    10.1.8 Empirical Propositions ........................................................................ 62
  10.2 Logical Conditions for Describing Physical Worlds .................................................. 62
    10.2.1 Symmetry and Asymmetry ........................................................................ 62
    10.2.2 Transitivity and Intransitivity ........................................................................ 62
    10.2.3 Space ........................................................................ 62
    10.2.4 States ........................................................................ 62
    10.2.5 Time ........................................................................ 63
    10.2.6 Causality ........................................................................ 63
    10.2.7 Kinematics ........................................................................ 63
    10.2.8 Dynamics ........................................................................ 64
    10.2.9 Newton’s Laws ........................................................................ 64
  10.3 Gravitation and Quantum Mechanics ........................................................................ 65
  10.4 The Logical Conditions for Describing Living Species .................................................. 65
    10.4.1 Purpose, Life and Evolution ........................................................................ 65
    10.4.2 Consciousness, Learning and Language .................................................. 66
  10.5 Humans, Knowledge, Responsibility ........................................................................ 66
  10.6 An Augmented Upper Ontology ........................................................................ 66
  10.7 Artifacts: Man-made Entities ........................................................................ 66
  10.8 Intentionality ........................................................................ 67

IV Fusing Philosophy into Computer Science .................................................. 68

11 Philosophical Issues of The Domain Calculi .................................................. 69
  11.1 The Analysis Calculus Prompts ........................................................................ 69
    11.1.1 External Qualities ........................................................................ 69
    11.1.2 Unique Identifiers ........................................................................ 71
    11.1.3 Mereology ........................................................................ 72
    11.1.4 Attributes ........................................................................ 73
    11.1.5 A Summary of Domain Analysis Prompts .................................................. 75
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.2</td>
<td>The Description Calculus Prompts</td>
<td>75</td>
</tr>
<tr>
<td>11.2.1</td>
<td>A Summary of Domain Description Prompts</td>
<td>76</td>
</tr>
<tr>
<td>11.3</td>
<td>The Behaviour Schemata</td>
<td>76</td>
</tr>
<tr>
<td>11.4</td>
<td>Wrapping Up</td>
<td>76</td>
</tr>
<tr>
<td>11.5</td>
<td>Discussion</td>
<td>76</td>
</tr>
<tr>
<td>11.5.1</td>
<td>Review of Revisions</td>
<td>76</td>
</tr>
<tr>
<td>11.5.2</td>
<td>General</td>
<td>77</td>
</tr>
<tr>
<td>V</td>
<td>Summing Up</td>
<td>77</td>
</tr>
<tr>
<td>12</td>
<td>Conclusion</td>
<td>77</td>
</tr>
<tr>
<td>12.1</td>
<td>General Remarks</td>
<td>77</td>
</tr>
<tr>
<td>12.2</td>
<td>Revisions to the Calculi and Further Studies</td>
<td>78</td>
</tr>
<tr>
<td>12.3</td>
<td>Remarks on Classes of Artifactual Perdurants</td>
<td>79</td>
</tr>
<tr>
<td>12.4</td>
<td>Acknowledgements</td>
<td>79</td>
</tr>
<tr>
<td>13</td>
<td>Bibliography</td>
<td>79</td>
</tr>
<tr>
<td>13.1</td>
<td>Bibliographical Notes</td>
<td>79</td>
</tr>
<tr>
<td>13.1</td>
<td>References</td>
<td>80</td>
</tr>
<tr>
<td>VI</td>
<td>Appendix</td>
<td>90</td>
</tr>
<tr>
<td>A</td>
<td>RSL: The RAISE Specification Language – A Primer</td>
<td>90</td>
</tr>
<tr>
<td>A.1</td>
<td>Type Expressions</td>
<td>90</td>
</tr>
<tr>
<td>A.1.1</td>
<td>Atomic Types</td>
<td>90</td>
</tr>
<tr>
<td>A.1.2</td>
<td>Composite Types</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Concrete Composite Types</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Sorts and Observer Functions</td>
<td>91</td>
</tr>
<tr>
<td>A.2</td>
<td>Type Definitions</td>
<td>92</td>
</tr>
<tr>
<td>A.2.1</td>
<td>Concrete Types</td>
<td>92</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Subtypes</td>
<td>93</td>
</tr>
<tr>
<td>A.2.3</td>
<td>Sorts — Abstract Types</td>
<td>93</td>
</tr>
<tr>
<td>A.3</td>
<td>The RSL Predicate Calculus</td>
<td>93</td>
</tr>
<tr>
<td>A.4</td>
<td>Propositional Expressions</td>
<td>93</td>
</tr>
<tr>
<td>A.4.1</td>
<td>Simple Predicate Expressions</td>
<td>93</td>
</tr>
<tr>
<td>A.4.2</td>
<td>Quantified Expressions</td>
<td>94</td>
</tr>
<tr>
<td>A.5</td>
<td>Concrete RSL Types: Values and Operations</td>
<td>94</td>
</tr>
<tr>
<td>A.5.1</td>
<td>Arithmetic</td>
<td>94</td>
</tr>
<tr>
<td>A.5.2</td>
<td>Set Expressions</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Set Enumerations</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Set Comprehension</td>
<td>94</td>
</tr>
<tr>
<td>A.5.3</td>
<td>Cartesian Expressions</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Cartesian Enumerations</td>
<td>95</td>
</tr>
<tr>
<td>A.5.4</td>
<td>List Expressions</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>List Enumerations</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>List Comprehension</td>
<td>95</td>
</tr>
<tr>
<td>A.5.5</td>
<td>Map Expressions</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Map Enumerations</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Map Comprehension</td>
<td>96</td>
</tr>
<tr>
<td>A.5.6</td>
<td>Set Operations</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Set Operator Signatures</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Set Examples</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Informal Explication</td>
<td>97</td>
</tr>
</tbody>
</table>
An Interpretation of Kai Sørlander’s Philosophy

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Operator Definitions</td>
<td>96</td>
</tr>
<tr>
<td>A.5.7 Cartesian Operations</td>
<td>97</td>
</tr>
<tr>
<td>A.5.8 List Operations</td>
<td>98</td>
</tr>
<tr>
<td>List Operator Signatures</td>
<td>98</td>
</tr>
<tr>
<td>List Operation Examples</td>
<td>98</td>
</tr>
<tr>
<td>Informal Explication</td>
<td>98</td>
</tr>
<tr>
<td>List Operator Definitions</td>
<td>99</td>
</tr>
<tr>
<td>A.5.9 Map Operations</td>
<td>100</td>
</tr>
<tr>
<td>Map Operator Signatures and Map Operation Examples</td>
<td>100</td>
</tr>
<tr>
<td>Map Operation Explication</td>
<td>100</td>
</tr>
<tr>
<td>Map Operation Redefinition</td>
<td>101</td>
</tr>
<tr>
<td>A.6 $\lambda$-Calculus + Functions</td>
<td>101</td>
</tr>
<tr>
<td>A.6.1 The $\lambda$-Calculus Syntax</td>
<td>101</td>
</tr>
<tr>
<td>A.6.2 Free and Bound Variables</td>
<td>102</td>
</tr>
<tr>
<td>A.6.3 Substitution</td>
<td>102</td>
</tr>
<tr>
<td>A.6.4 $\alpha$-Renaming and $\beta$-Reduction</td>
<td>102</td>
</tr>
<tr>
<td>A.6.5 Function Signatures</td>
<td>103</td>
</tr>
<tr>
<td>A.6.6 Function Definitions</td>
<td>103</td>
</tr>
<tr>
<td>A.7 Other Applicative Expressions</td>
<td>104</td>
</tr>
<tr>
<td>A.7.1 Simple let Expressions</td>
<td>104</td>
</tr>
<tr>
<td>A.7.2 Recursive let Expressions</td>
<td>104</td>
</tr>
<tr>
<td>A.7.3 Predicative let Expressions</td>
<td>104</td>
</tr>
<tr>
<td>A.7.4 Pattern and “Wild Card” let Expressions</td>
<td>104</td>
</tr>
<tr>
<td>A.7.5 Conditionals</td>
<td>105</td>
</tr>
<tr>
<td>A.7.6 Operator/Operand Expressions</td>
<td>105</td>
</tr>
<tr>
<td>A.8 Imperative Constructs</td>
<td>106</td>
</tr>
<tr>
<td>A.8.1 Statements and State Changes</td>
<td>106</td>
</tr>
<tr>
<td>A.8.2 Variables and Assignment</td>
<td>106</td>
</tr>
<tr>
<td>A.8.3 Statement Sequences and skip</td>
<td>106</td>
</tr>
<tr>
<td>A.8.4 Imperative Conditionals</td>
<td>106</td>
</tr>
<tr>
<td>A.8.5 Iterative Conditionals</td>
<td>106</td>
</tr>
<tr>
<td>A.8.6 Iterative Sequencing</td>
<td>107</td>
</tr>
<tr>
<td>A.9 Process Constructs</td>
<td>107</td>
</tr>
<tr>
<td>A.9.1 Process Channels</td>
<td>107</td>
</tr>
<tr>
<td>A.9.2 Process Composition</td>
<td>107</td>
</tr>
<tr>
<td>A.9.3 Input/Output Events</td>
<td>107</td>
</tr>
<tr>
<td>A.9.4 Process Definitions</td>
<td>108</td>
</tr>
<tr>
<td>A.10 Simple RSL Specifications</td>
<td>108</td>
</tr>
<tr>
<td>A.11 RSL Index</td>
<td>108</td>
</tr>
<tr>
<td>B RSL\textsuperscript{+}</td>
<td>110</td>
</tr>
<tr>
<td>C A Language of Domain Analysis &amp; Description Prompts</td>
<td>110</td>
</tr>
<tr>
<td>D A Description Narration Language</td>
<td>110</td>
</tr>
<tr>
<td>E Indexes</td>
<td>111</td>
</tr>
<tr>
<td>E.1 Philosophy Index</td>
<td>111</td>
</tr>
<tr>
<td>E.2 Domain Analysis Index</td>
<td>116</td>
</tr>
<tr>
<td>E.2.1 Concepts</td>
<td>116</td>
</tr>
<tr>
<td>E.2.2 Definitions</td>
<td>117</td>
</tr>
<tr>
<td>E.2.3 Analysis Predicates</td>
<td>119</td>
</tr>
<tr>
<td>E.2.4 Description Observers</td>
<td>119</td>
</tr>
<tr>
<td>E.2.5 Proof Obligations and Axioms</td>
<td>119</td>
</tr>
<tr>
<td>E.2.6 Observer Function Literals</td>
<td>120</td>
</tr>
</tbody>
</table>
1 Introduction

Definition 1 Domain: By a domain we shall understand a rationally describable segment of a human assisted reality, i.e., of the world, its physical parts, and living species. These are endurants (“still”), existing in space, as well as perdurants (“alive”), existing also in time. Emphasis is placed on “human-assistedness”, that is, that there is at least one (man-made) artifact and that humans are a primary cause for change of endurant states as well as perdurant behaviours.

The science and engineering of domain analysis & description is different from the science of physics and the core of its derived engineerings: building (civil), chemical, mechanical, electrical, electronics, et cetera. All of these engineerings emerged out of the natural sciences. These classical engineering disciplines have increasingly included many facets of man-machine interface concerns, but their core is still in the natural sciences. We assume that the readers are familiar with the above notions.

The core of domain science & engineering such as we shall pursue it, is in two disciplines: mathematics, notably mathematical logic and abstract algebra, and philosophy, notably metaphysics and epistemology. We assume that the readers are familiar with the above-mentioned notions of mathematics.

Definition 2 Metaphysics: By metaphysics we shall understand a branch of philosophy that explores fundamental questions, including the nature of concepts like being, existence, and reality. Traditional metaphysics seeks to answer, in a “suitably abstract and fully general manner”, the questions: What is there? and And what is it like?

Topics of metaphysical investigation include existence, objects and their properties, space and time, cause and effect, and possibility.

Definition 3 Epistemology: By epistemology [from epistēmē, 'knowledge', and logos, 'logical discourse'] is the branch of philosophy concerned with the theory of knowledge.

The philosophy aspect of our study is primarily epistemological, i.e., not metaphysical.

Epistemology studies the nature of knowledge, justification, and the rationality of belief. Much of the debate in epistemology centers on four areas: (1) the philosophical analysis of the nature of knowledge and how it relates to such concepts as truth, belief, and justification, (2) various problems of skepticism, (3) the sources and scope of knowledge and justified belief, and (4) the criteria for knowledge and justification. A central branch of epistemology is ontology, the investigation into the basic categories of being and how they relate to one another.

Observe the distinction in the definitions of metaphysics and epistemology between [metaphysics] “explores fundamental questions, including the nature of concepts like being, existence, and reality” and [epistemology] “the philosophical analysis of the nature of knowledge and how it relates to such concepts as truth, belief, and justification, etc.”. Epistemology addresses such questions as What makes justified beliefs justified?; What does it mean to say that we know something?; and, fundamentally, “How do we know that we know?”

---

2https://en.wikipedia.org/wiki/Metaphysics
3https://en.wikipedia.org/wiki/Epistemology
4https://en.wikipedia.org/wiki/Epistemology
5https://en.wikipedia.org/wiki/Metaphysics
1.1 Two Views of Domains

There are two aspects to this paper: (i) the analysis & description of fragments of the context in which software, to be developed, is to serve, (ii) and the general, basically philosophical, problem of the absolutely necessary conditions for describing the world.

1.1.1 The Computing Science View

In twelve papers we have put forward a method for analysing and describing the domains for which software is developed:

- [4, 1] Manifest Domains: Analysis & Description  
  FAoC, March 2017
- [5, 6] Domain Facets: Analysis & Description
- [7, 8] Formal Models of Processes and Prompts
- [9, 10] To Every Manifest Domain Mereology a CSP Expression  
  LAMP, Jan. 2018
- [11, 12] From Domain Descriptions to Requirements Prescriptions
- [13, 14] Domains: Their Simulation, Monitoring and Control

These methods involve new principles, techniques and tools – the calculi. The calculi has been applied in around 20+ experimental researches to as diverse domains as

- railways,
- pipelines,
- swarms of drones,
- IT security,
- road transport systems,
- documents and
- container shipping lines,
- stock exchanges,
- credit card systems,
- urban planning.

The calculi, we claim, has withstood some severe “tests”. The experiments are referenced in Sect. 13.1 [pp. 79].

1.1.2 The Philosophy View

In four books the Danish philosopher Kai Sørlander has investigated the philosophical issues alluded to above.

A main contribution of Sørlander is, on the philosophical basis of the possibility of truth (in contrast to Kant’s possibility of self-awareness) to rationally and transcendentally deduce the absolutely necessary conditions for describing any world. These conditions presume a principle of contradiction and lead to the ability to reason using logical connectives and to handle asymmetry, symmetry and transitivity. Transcendental deductions then lead to space and time, not as priory assumptions, as with Kant, but derived facts of any the world. From this basis Sørlander then, by further transcendental deductions arrive at kinematics, dynamics and the bases for Newton’s Laws. And so forth. We build on Sørlander’s basis to argue that the domain analysis & description calculi are necessary and sufficient and that a number of relations between domain entities can be understood transcendentally and as “variants” of Newton’s Laws!

1.1.3 First Two Independent Treatments, then An Interpretation

First we present the two views independent of one-another.

In Segment I we present the domain analysis & description method: its principles, techniques and tools, Sects. 2–5, and a substantial example, Sect. 6, to support understanding the domain analysis & description method.

In Segment III we present in Sect. 8 a brief motivation of the task of philosophy; in Sect. 9 an extensive review is presented of metaphysical and epistemological issues in philosophy, from the ancient Greeks up till the mid 1900’s; in Sect. 10 an extensive review is then given of Sørlander’s Philosophy.

Then, in Segment IV’s Sect. 11, we bring the two studies — the domain analysis & description calculi and the Kai Sørlander Philosophy — together: It is here that, as a consequence of Sørlander’s Philosophy, we modify the domain analysis & description method, of Segment I, in suggesting extensions.

The Main Contribution

With Segment IV the the main contribution of this report is achieved: (i) establishing a basis for domain science & engineering in philosophy; and (ii) the specific modifications required by and the founding of the domain analysis & description calculi in philosophy.

In Segment II, in-between Segments I and III, we present in Sect. 7, a short review of space and time.

1.2 The Computing Science Background

1.2.1 Computer & Computing Science

- By computer science I understand the study and knowledge of the “things” that can “exist inside” computing devices (i.e., data and computations) – and the study and knowledge of these computing devices.

- By computing science I understand the study and knowledge of how to construct “those things”, i.e., programming methodology.

I consider myself a computing scientist primarily interested in programming methodology.
1.2.2 Formal Methods

• By a method I understand a set of principles for selecting and applying a set of techniques and tools for the construction of an artifact, as here, software.

• By a formal method I understand a method whose principles, techniques and tools can be understood in a mathematical framework – for example where, among the tools, the specification languages can be given a mathematical syntax, a mathematical semantics and a mathematical proof system.

I consider myself to have primarily contributed to the area of formal methods, as exemplified by VDM and RAISE.

1.2.3 A Triptych of Engineering

• Before software can be designed we must be familiar with its requirements.

• Before requirements can be prescribed we must be familiar with the context of the software to be developed, that is, the domain.

• Hence the triptych of software development:
  • first (ideally) the domain engineering of an appropriate domain description;
  • then (ideally) the requirements engineering of the requirements prescription – formally related to the domain description;
  • finally the software design “derived” from the requirements prescription and (ideally) formally reasoned to meet customers’ expectations, that is, to satisfy the domain description and be correct wrt. the requirements prescription.

My contributions in the last many years has been to establish a proper domain science & engineering. My main focus, since 1977, has been on the development of ”large” software: compilers (like for CHILL and Ada), and infrastructure software – for pipelines, railways, health care, banking, road traffic, etc.

1.3 Domains, their Analysis & Description, and a Method

In Definition 1 [pp. 8] we gave a rough characterisation of what we mean by domain. In this section we shall brief outline what we mean by domain analysis & description, and what we mean by a method for analysing & describing domains.

1.3.1 Domain Analysis & Description

Definition 4: Domain Analysis and Description: By domain analysis and description we shall understand the analysis & description of domains.
1.3.2 A Domain Analysis & Description Method

Definition 5: A Domain Analysis and Description Method: By a domain analysis and description method we shall understand a set of principles, techniques and tools for the construction, i.e., analysis & description of a domain model. The terms description and model are here considered synonymous.

Segment I: The Domain Analysis & Description Calculi

2 Endurants – cf. s. 6.2 Pg. 36

In a series of definitions, most of which are rather like characterisations, we shall explicate a number of domain concepts. These definitions will lead to the introduction of first domain analysis prompts, then also domain description prompts. Think of a prompt as a cue, a hint, a suggestion, in German, a stichwort, suchbegriff, in French, a signal théâtre, that the domain analyser is told, by the principles of the domain analysis & description method, to act upon.

2.1 The Universe of Discourse – cf. s. 6.1 Pg. 36

Analysis Prompt 1 is_universe_of_discourse: By a universe of discourse for domain science & engineering we shall mean a human-centered area of concern, one that involves, as “main players”: endurants and perdurants such that at least one of the endurants is man-made and either represents a human or at least another one is a human.

Example 1 Man-made Automobiles and Drivers: In the large example of Sect. 6 automobiles and road nets are endurants, and automobiles “subsume” their human drivers.

Domain Description Prompt 1 observe_universe_of_discourse: The domain-of-interest needs first be briefly narrated. Just a simple story. One that emphasises the “main players”: the endurants and the perdurants such that at least one of the endurants is man-made and and either represents a human or at least another one is a human.

2.2 Basic Domain Concepts

Definition 6 Entity: By an entity we shall understand a phenomenon, i.e., something that can be observed, i.e., be seen or touched by humans, or that can be conceived as an abstraction of an entity; alternatively, a phenomenon is an entity, if it exists, it is “being”, it is that which makes a “thing” what it is: essence, essential nature [24, Vol. I, pg. 665].

7Usually, in computer science papers, definitions are terse and based on more-or-less implicit reference to a mathematically precise model. Since domains do not have an a-priori mathematically precise model our definitions cannot be precise. Most of the definitions are taken from such dictionaries as [24, The Oxford Shorter English Dictionary] and from the Internet based [25, The Stanford Encyclopedia of Philosophy].
Example 2  **Entities and Non-entities**: The following are entities: a stone, say, laying on the ground – which is an entity; a pencil, say, that I, a human entity, hold in my hand; a rhododendron, in my garden – which is an entity. The following are not entities: the blue sky of my imagination; a fleeting moment of sadness; being drunk.

**Analysis Prompt 2** *is_entity*: The domain analyser analyses “things” (θ) into either entities or non-entities. The method can thus be said to provide the **domain analysis prompt**:

- **is_entity** – where **is_entity**(θ) holds if θ is an entity.

**Definition 7** **Endurant**: By an **endurant** we shall understand an entity that can be observed or conceived and described as a “complete thing” at no matter which given snapshot of time; alternatively an entity is endurant if it is capable of **enduring**, that is **persist**, “**hold out**” [24, Vol. I, pg. 656]. Were we to “freeze” time we would still be able to observe the full endurant.

**Example 3** **Endurants**: The following are examples of endurants: the lake of a landscape such as a tourist (i.e., an animal entity) photographs it; the engine train of an automobile such as an automobile mechanic (a human entity) repairs it; and the horse such as a jockey (a human entity) prepares it for a race.

**Analysis Prompt 3** *is_endurant*: The domain analyser analyses an entity, e, into an endurant as prompted by the **domain analysis prompt**:

- **is_endurant** – φ is an endurant if **is_endurant**(e) holds.

**is_entity** is a prerequisite prompt for **is_endurant**.

**Definition 8** **Perdurant**: By a **perdurant** we shall understand an entity for which only a fragment exists if we look at or touch them at any given snapshot in time, that is, were we to freeze time we would only see or touch a fragment of the perdurant, alternatively an entity is perdurant if it endures continuously, over time, persists, lasting [24, Vol. II, pg. 1552].

**Example 4** **Perdurants**: The following are examples of perdurants: the flow of water in a river; the human life, from birth to death; the car driving down a road.

**Analysis Prompt 4** *is_perdurant*: The domain analyser analyses an entity e into perdurants as prompted by the **domain analysis prompt**:

- **is_perdurant** – e is a perdurant if **is_perdurant**(e) holds.

**is_entity** is a prerequisite prompt for **is_perdurant**.
**Definition 9** Discrete Endurant: *By a discrete endurant we shall understand an endurant which is separate, individual or distinct in form or concept.*

**Example 5** Discrete Endurants: The following are examples of discrete endurants: planets in space; automobiles (in a car sales office); and students at a lecture in a college classroom.

**Analysis Prompt 5** `is_discrete`: The domain analyser analyses endurants `e` into discrete entities as prompted by the domain analysis prompt:

- `is_discrete – e` is discrete if `is_discrete(e)` holds

**Definition 10** Continuous Endurant: *By a continuous endurant we shall understand an endurant which is prolonged, without interruption, in an unbroken series or pattern.*

**Example 6** Continuous Endurants: The following are examples of continuous endurants: springs, brooks, rivers and lakes of a landscape; and gas in a pipeline.

**Analysis Prompt 6** `is_continuous`: The domain analyser analyses endurants `e` into continuous entities as prompted by the domain analysis prompt:

- `is_continuous – e` is continuous if `is_continuous(e)` holds

Continuity shall here not be understood in the sense of mathematics. Our definition of ‘continuity’ focused on prolonged, without interruption, in an unbroken series or pattern. In that sense materials (water, oil, sand, gravel, ...) shall be seen as ‘continuous’.

### 2.3 An Upper Ontology Diagram of Domains – A Preview

Figure 1 [facing page] shows a so-called upper ontology for manifest domains. So far we have covered only a fraction of this ontology, as noted. By ontologies we shall here understand formal representations of a set of concepts within a domain and the relationships between those concepts.

### 2.4 Structures – cf. s. 6.2.1 Pg. 36

**Definition 11** Structure: *By a structure we shall understand a discrete endurant which the domain engineer chooses to describe as itself consisting of structures, parts, components and materials but to not endow itself with internal qualities: unique identifiers, mereology or attributes.*
We shall define the terms parts, components and materials, as well as unique identification, mereology and attributes later. Structures are introduced in the domain analysis & description method for pragmatic reasons. When modelling an endurant as a structure we are disregarding that the endurant may have a physically “separate” form, treating that endurant as a concept rather than something manifest. Endurants “first” modelled as structures may, subsequently, or also, be modelled as (usually composite) parts (see below).

**Analysis Prompt 7** *is_structure:* The domain analyst analyse endurants, e, into structure entities as prompted by the domain analysis prompt:

- *is_structure*

Structures are thus composite endurants which consist of other endurants: discrete as well as continuous, i.e., structures, [physical] parts[,] living species] and components, as well as materials. Parts, components and material will soon be defined. The [...] bracketed concepts will not be defined till late in this report.

### 2.5 Parts, Components and Materials

**2.5.1 Parts** – cf. s. 6.2.3 Pg. 37

**Characterisation 1 Parts:** Parts are manifest in the sense that we can see them, touch them: we can uniquely identify them (unique identification); relate them to other parts (mereology); and “measure” some of their characteristics (attributes);
Parts are going to be the “work horse” of domain descriptions. Our primary focus will be on man-made parts (artifacts). We leave it to physics (i.e., physicists) to model natural parts.

**Definition 12 Part:** By a part we shall understand a discrete endurant which the domain engineer chooses to endow with all three internal qualities: unique identification, mereology, and one or more attributes.

**Example 7 Examples of Parts:** Examples of natural parts are: a raw diamond (as found in the ground); the Rock of Gibraltar; The Equator. Examples of man-made parts, that is, artifacts are: an armchair; the Empire State Building; and a canal lock.

**Analysis Prompt 8 is_part:** The domain analyser analyse endurants, e, into part entities as prompted by the domain analysis prompt:

- is_part

**Definition 13 Atomic Part:** Atomic parts are those which, in a given context, are deemed not to consist of meaningful, separately observable proper sub-parts. A sub-part is a part.

**Example 8 Atomic Parts:** These are examples of atomic (man-made) parts: a bolt, a screw, a nail; an automobile as bought by the owner; and a pipe, valve, pump, fork, and join of a pipeline.

**Analysis Prompt 9 is_atomic:** The domain analyser analyses a discrete endurant, i.e., a part p into an atomic endurant:

- is_atomic: p is an atomic endurant if is_atomic(p) holds

**Definition 14 Composite Part:** Composite parts are those which, in a given context, are deemed to indeed consist of meaningful, separately observable proper sub-parts.

**Example 9 Composite Parts:** These are examples of composite (man-made) parts: a nut (bolt) and screw assembly; an automobile as put together or serviced by a factory, resp. a mechanic; and a pipeline (consisting of pipes, valves, pumps, forks, joins etc.).

**Analysis Prompt 10 is_composite:** The domain analyser analyses a discrete endurant, i.e., a part p into a composite endurant:

- is_composite: p is a composite endurant if is_composite(p) holds

**Analysis Prompt 11 observe_endurants:** The domain analysis prompt:
• observe_endurants
directs the domain analyser to observe the sub-endurants of an endurant \( e \) and to suggest their sorts. Let, schematically, \( \text{observe_endurants}(e) = \{e_1:E_1, e_2:E_2, \ldots, e_m:E_m\} \).

Domain Description Prompt 2 \( \text{observe_endurant_sorts} \): If \( \text{is_composite}(p) \) holds, then the analyser “applies” the domain description prompt

• \( \text{observe_endurant_sorts}(p) \)

resulting in the analyser writing down the endurant sorts and endurant sort observers domain description text according to the following schema:

2. \( \text{observe_endurant_sorts} \) schema

Narration:

\[ s \] ... narrative text on sorts ...
\[ o \] ... narrative text on sort observers ...
\[ \eta \] ... narrative text on sort type observers ...
\[ i \] ... narrative text on sort recognisers ...
\[ p \] ... narrative text on proof obligations ...

Formalisation:

\[ s \] \( P \),
\[ s \] \( E_i : [1..m] \) comment: \( E_i : [1..m] \) abbreviates \( E_1, E_2, \ldots, E_m \)
\[ o \] \( \text{obs_endurant_sorts}_{E_i} : P \to E_i : [1..m] \)
\[ \eta \] \( \text{if} \ \text{is_part}(e_i) \): \( \eta(e_i) \equiv \ll E_i \gg i : [1..m] \)
\[ i \] \( \text{is}_{E_i} : (E_1|E_2|\ldots|E_m) \to \text{Bool} \ i : [1..m] \)

proof obligation \( [\text{Disjointness of endurant sorts}] \)
\[ p \] \( \text{PO} : \forall e : (E_1|E_2|\ldots|E_m) \cdot \)
\[ p \] \( \land \{ \text{is}_{E_i}(e) \equiv \land \{\sim \text{is}_{E_j}(p) \mid j : [1..m] \setminus \{i\} \mid i : [1..m]\} \} \)

Example 10 Observe Transport System Endurants: We refer to example Sect. 6.2.1 [pp. 36] annotation and formalisation Items 8–10; and to example Sect. 6.2.2 [pp. 36] annotation and formalisation Items 11–12a.

Some composite parts can suitably be modelled as sets of parts of the same sort.

Analysis Prompt 12 \( \text{has_concrete_type} \): The domain analyser may decide that it is expedient, i.e., pragmatically sound, to render a part sort, \( P \), whether atomic or composite, as a concrete type, \( T \). That decision is prompted by the holding of the domain analysis prompt:

• \( \text{has_concrete_type}(p) \).

\( \text{is_discrete} \) is a prerequisite prompt \( \text{has_concrete_type} \) of \( \text{has_concrete_type} \).
**Domain Description Prompt 3**  
*observe_part_type*: The domain analyser applies the domain description prompt:

- *observe_part_type(p)*\(^{11}\)

... to parts *p*: *P* which then yield the part type and part type observers domain description text according to the following schema:

<table>
<thead>
<tr>
<th>Narration:</th>
<th>Formalisation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_1) ... narrative text on sorts and types <em>S</em> (i) ...</td>
<td>type [t_1] (S_1, S_2, ..., S_m, ..., S_n,)</td>
</tr>
<tr>
<td>(t_2) ... narrative text on types <em>T</em> ...</td>
<td>(t_2) (T = \mathcal{E}(S_1, S_2, ..., S_n))</td>
</tr>
<tr>
<td>(t_3) ... narrative text on type of value observer</td>
<td>(t_3) (\eta(s_i) \equiv \ll S \gg, i: [1..n], s_i: S_i)</td>
</tr>
<tr>
<td>(o) ... narrative text on type observers ...</td>
<td>value [o] (\text{obs_part}_T: P \rightarrow T)</td>
</tr>
</tbody>
</table>

2.5.2 **Components** – cf. s. 6.2.4 Pg. 37

Some discrete composite endurants can suitably be modelled as sets of parts of possibly different sorts but for which there is no need to model their mereology, that is, how the parts in the set relate to one another.

**Definition 15 Component**: By a component we shall understand a discrete endurant which we, the domain analyser cum describer chooses to not endow with mereology.

Parts may or may not contain, i.e., “have”, components.

**Example 11 Components of Parts**: a part, like a mail-box, may contain letters, newspapers, small packages, advertisement brochures, etc.; a part, like a household shop shelf, may contain bread toasters, blenders, coffee grinders, coffee machines, etc.; and a part, like a book case, may contain books, journals, bric-à-brac, etc.

\(^{9}\)Later, when having introduced continuous endurants, i.e., materials, one may claim that the physical aspects of the enclave of *Gibraltar* could also be modelled as a material.

\(^{10}\)One may claim that *The Equator* is a non-physical concept. To this one may counter-claim that *The Equator* is physically delineable: can be “marked down”!

\(^{11}\)has_concrete_type is a **prerequisite prompt** of observe_part_type.
Analysis Prompt 13 has_components: The domain analyser inquire endurants e as to whether they have, i.e., contain, components, as prompted by the domain analysis prompt:

• has_components

Analysis Prompt 14 is_component: The domain analyser analyse endurants e into component entities as prompted by the domain analysis prompt:

• is_component

Domain Description Prompt 4 observe_component_sorts: The domain description prompt:

• observe_component_sorts \( P(p) \)

yields the component sorts and component sort observer domain description text according to the following schema – whether or not the actual part \( p \) contains any components:

<table>
<thead>
<tr>
<th>Narration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [s] ) ... narrative text on component sorts ...</td>
</tr>
<tr>
<td>( [o] ) ... narrative text on component observers ...</td>
</tr>
<tr>
<td>( [i] ) ... narrative text on component sort recognisers ...</td>
</tr>
<tr>
<td>( [u] ) ... narrative text on unique identifier ...</td>
</tr>
<tr>
<td>( [p] ) ... narrative text on component sort proof obligations ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formalisation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
</tr>
<tr>
<td>( [s] ) K1, K2, ..., Kn</td>
</tr>
<tr>
<td>( [s] ) K = K1</td>
</tr>
<tr>
<td>( [s] ) KS = K-set</td>
</tr>
<tr>
<td>value</td>
</tr>
<tr>
<td>( [o] ) ( \text{obs}_i(P) : P \to KS )</td>
</tr>
<tr>
<td>( [i] ) ( \text{is}_K_i(K_1</td>
</tr>
<tr>
<td>( [u] ) ( \text{uid}_K_i )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proof Obligation: [Disjointness of Component Sorts]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [p] ) ( PO : \forall k_i:(K_1</td>
</tr>
<tr>
<td>( [p] ) ( &amp; { \text{is}_K_i(k_i) \equiv &amp;{\sim \text{is}_K_j(k_j)</td>
</tr>
</tbody>
</table>

Example 12 Observe Transport System Component Sorts: We refer to example Sect. 6.2.4 [pp. 37] annotation and formalisation Items 16–17
2.5.3 Materials – cf. s. 6.2.5 Pg. 37

Definition 16 Material: By a material we shall understand a continuous endurant

Parts may or may not contain, i.e., “have”, materials.

Example 13 Materials of Parts: a part, like a pipe-line pipe, may contain oil; a part, like a timber yard, may contain boards, lumber, etc., of different sizes and qualities; and a part, like a building materials shop, may contain concrete, sand, gravel, bricks, etc., in different bags, containers and sizes

Example 14 Observe Transport Component Sorts: We refer to example Sect. 6.2.4 [pp. 37] annotation and formalisation Items 16–17

Analysis Prompt 15 has_materials: The domain analyser inquire endurants e as to whether they have, i.e., contains, material, as prompted by the domain analysis prompt:

• has_materials

Analysis Prompt 16 is_material: The domain analyser analyse endurants e into material entities as prompted by the domain analysis prompt:

• is_material

Domain Description Prompt 5 observe_material_sorts_P: The domain description prompt:

• observe_material_sorts_P(e)

yields the material sorts and material sort observers’ domain description text according to the following schema whether or not part p actually contains materials:

Narration:
[s] ... narrative text on material sorts ...
[o] ... narrative text on material sort observers ...
[i] ... narrative text on material sort recognisers ...
[p] ... narrative text on material sort proof obligations ...

Formalisation:

type
[s] M1, M2, ..., Mn
[s] M = M1 | M2 | ... | Mn
[s] MS = M-set
[a] Ai = A11 | A12 | ... | A1n
value
[o] obs_mat_sort_M: P → M, [i:1..n]
[o] obs_materials_P: P → MS
Example 15 **Observe Transport System Materials**: We refer to example Sect. 6.2.5 [pp. 37]annotation and formalisation Items 18–19

2.6 **Unique Part and Component Identifiers** – cf. s. 6.2.7 Pg. 37

We introduce a notion of unique identification of parts and components. We assume (i) that all parts and components, $p$, of any domain $P$, have *unique identifiers*, (ii) that *unique identifiers* (of parts and components $p:P$) are *abstract values* (of the *unique identifier* sort $PI$ of parts $p:P$), (iii) such that distinct part or component sorts, $P_i$ and $P_j$, have distinctly named *unique identifier* sorts, say $PI_i$ and $PI_j$, (iv) that all $\pi_i:PI_i$ and $\pi_j:PI_j$ are distinct, and (v) that the observer function $uid_P$ applied to $p$ yields the unique identifier, say $\pi:PI$, of $p$.

**Analysis Prompt 17 type_name**: The description language function *type_name* applies to unique identifiers, $pui:PUI$, and yield the name of the type, $P$, of the parts having unique identifiers of type $PUI$:

- *type_name* – where *type_name*($pui$) yields $P$

**Representation of Unique Identifiers**: Unique identifiers are abstractions. When we endow two parts (say of the same sort) with distinct unique identifiers then we are simply saying that these two parts are distinct. We are not assuming anything about how these identifiers otherwise come about.

**Domain Description Prompt 6 observe_unique_identifier**: We can therefore apply the domain description prompt:

- **observe_unique_identifier**

to parts $p:P$ resulting in the analyser writing down the unique identifier type and observer domain description text according to the following schema:

| s | ... narrative text on unique identifier sort $PI$ ...
| u | ... narrative text on unique identifier observer $uid_P$ ...
| η| ... narrative text on type name, an RSL+$^+$Text observer ...
| a | ... axiom on uniqueness of unique identifiers ...

**Formalisation**:

```plaintext
[\{i\}] \text{is}_M: M \rightarrow \text{Bool} [i:1..n]
[a] \text{attr}_{A_{ij}}: M_i \rightarrow A_{ij} [i:...,j:...]
proof obligation \ [\text{Disjointness of Material Sorts}]
[p] PO: ...
```
Example 16  **Observe Transport System Identifiers**: We refer to example Sect. 6.2.7 [pp. 37] annotation and formalisation Items 26–28d.

### 2.7 Part Mereologies – cf. s. 6.2.9 Pg. 38

**Mereology** is the study and knowledge of parts and part relations. Mereology, as a logical/philosophical discipline, can perhaps best be attributed to the Polish mathematician/logician Stanisław Leśniewski [26, 27].

#### 2.7.1 Part Relations

Which are the relations that can be relevant for part-hood? We give some examples. (i) Two otherwise distinct parts may “share” values. By ’sharing’ values we shall, as a generic example, mean that two parts of different sorts has the same attributes but that one ‘defines’ the attribute, like, for example ‘programming’ its values, cf. df.27 pp.26, whereas the other ‘uses’ these values, like, for example considering them ‘inert’, cf. df.22 pp26. (ii) Two otherwise distinct parts may be said to, for example, be topologically “adjacent” or one “embedded” within the other. These examples are in no way indicative of the “space” of part relations that may be relevant for part-hood. The domain analyser is expected to do a bit of experimental research in order to discover necessary, sufficient and pleasing “mereology-hoods”!

#### 2.7.2 Part Mereology: Types and Functions

**Analysis Prompt 18** *has_mereology*: To discover necessary, sufficient and pleasing “mereology-hoods” the analyser can be said to endow a truth value, *true*, to the **domain analysis prompt**:

- *has_mereology*

When the domain analyser decides that some parts are related in a specifically enunciated mereology, the analyser has to decide on suitable **mereology types** and **mereology observers** (i.e., part relations).

**Domain Description Prompt 7** *observe_mereology*: If *has_mereology*(p) holds for parts *p* of type *P*, then the analyser can apply the **domain description prompt**:

- *observe_mereology*

  to parts of that type and write down the **mereology types** and observer domain description text according to the following schema:

12For the concept of attribute value see Sect. 2.8.2 [pp. 24].
7. observe_mereology schema

Narration:

[t] ... narrative text on mereology type ...
[m] ... narrative text on mereology observer ...
[a] ... narrative text on mereology type constraints ...

Formalisation:

type

[MT]

value

[m] obs_mereo\_P: P → MT

axiom [Well-formedness of Domain Mereologies]

[a] A: A(MT)

Example 17 Observe Transport System Mereology: We refer to example Sect. 6.2.9 [pp. 38] annotation and formalisation Items 40–43.

2.8 Part Attributes – cf. s. 6.2.10 Pg. 39

To recall: there are three sets of internal qualities: unique part identifiers, part mereology and attributes. Unique part identifiers and part mereology are rather definite kinds of internal endurant qualities. Part attributes form more “free-wheeling” sets of internal qualities.

Example 18 Example Part Attributes: These are examples of part attributes: the carat of a diamond; the number of residents of Gibraltar; the medium diameter and length of the equator; and the length and location of a street segment (i.e., a link).

2.8.1 Inseparability of Attributes from Parts and Materials

Parts and materials are typically recognised because of their spatial form and are otherwise characterised by their intangible, but measurable attributes. That is, whereas endurants, whether discrete (as are parts and components) or continuous (as are materials), are physical, tangible, in the sense of being spatial [or being abstractions, i.e., concepts, of spatial endurants], attributes are intangible: cannot normally be touched, or seen, but can be objectively measured. Thus, in our quest for describing domains where humans play an

---

14 Note that we do not presently describe what a location is.
15 One can see the red colour of a wall, but one touches the wall.
16 One cannot see electric current, and one may touch an electric wire, but only if it conducts high voltage can one know that it is indeed an electric wire.
17 That is, we restrict our domain analysis with respect to attributes to such quantities which are observable, say by mechanical, electrical or chemical instruments. Once objective measurements can be made of human feelings, beauty, and other, we may wish to include these “attributes” in our domain descriptions.
active rôle, we rule out subjective “attributes”: feelings, sentiments, moods. Thus we shall abstain, in our domain science also from matters of aesthetics. We equate all endurants which, besides possible type of unique identifiers (i.e., excepting materials) and possible type of mereologies (i.e., excepting components and materials), have the same types of attributes, with one sort. Thus removing a quality from an endurant makes no sense: the endurant of that type either becomes an endurant of another type or ceases to exist (i.e., becomes a non-entity)!

Example 19 Inseparability of Attributes: Let the part be a link (i.e., street segment). It must have a length; a link without a length is meaningless. It must have a location; a link without a location is meaningless.

2.8.2 Attribute Quality and Attribute Value

We distinguish between an attribute (as a logical proposition, of a name, i.e.) type, and an attribute value, as a value in some value space.

Analysis Prompt 19 attribute_types: One can calculate the set of attribute types of parts and materials with the following domain analysis prompt:

- attribute_types

Thus for a part $p$ we may have \text{attribute_types}(p) = \{A_1, A_2, \ldots, A_m\}.

Example 20 Example Attribute Sorts: Let the part be a pipeline unit such as a pipe, a pump, a valve, a fork, or a join. the material “flowed” by the pipeline; the location of the unit; the diameter of a pipe; the [dynamically changeable] valve position (open, closed, ...); the current and (for guaranteeing laminar flow) maximal in- and out-flows of the pipeline units; et cetera. Notice that there are possibly very many other attributes: we may model some of these; others we may choose to ignore.

2.8.3 Part and Material Attributes: Types and Functions

Let us recall that attributes cover qualities other than unique identifiers and mereology. Let us then consider that parts and materials have one or more attributes. These attributes are qualities which help characterise “what it means” to be a part or a material. Note that we expect every part and material to have at least one attribute. The question is now, in general, how many and, particularly, which.

Domain Description Prompt 8 observe_attributes: The domain analyser experiments, thinks and reflects about part attributes. That process is initiated by the domain description prompt:

- observe_attributes.
The result of that domain description prompt is that the domain analyser cum describer writes down the attribute (sorts or) types and observers domain description text according to the following schema:

<table>
<thead>
<tr>
<th>8. observe_attributes schema</th>
</tr>
</thead>
</table>

Narration:
- [t] ... narrative text on attribute sorts ...
- [o] ... narrative text on attribute sort observers ...
- [v] ... narrative text on set of attribute value observers ...
- [i] ... narrative text on attribute sort recognisers ...
- [p] ... narrative text on attribute sort proof obligations ...

Formalisation:

- **type**
  - [t] $A_i$ \[1 \leq i \leq n\]

- **value**
  - [o] $\text{attr}_A : P \rightarrow A_i \ i : [1..n]$
  - [v] $\text{obs}_\text{attrib_values} . P(p) \equiv \{ \text{attr}_A_1(p), \text{attr}_A_2(p), ..., \text{attr}_A_n(p) \}$
  - [i] $\text{is}_A : (A_1 | A_2 | ... | A_n) \rightarrow \text{Bool} \ i : [1..n]$

- **proof obligation** [Disjointness of Attribute Types]
  - [p] $PO$: let $P$ be any part sort in [the domain description]
  - [p] let $a : (A_1 | A_2 | ... | A_n)$ in $\text{is}_A(a) \neq \text{is}_A_j(a)$ end end \[i \neq i, i,j : [1..n] \] 

---

**Example 21** Road Transport System Attribute Observers: We refer to example Sect. 6.2.10 narrative and formulas Items 46 [pp. 39] to 56d [Page 40].

### 2.8.4 Attribute Categories

Michael A. Jackson [28] has suggested a hierarchy of attribute categories: static or dynamic values – and within the dynamic value category: inert values or reactive values or active values – and within the dynamic active value category: autonomous values or biddable values or programmable values. We now review these attribute value types. The review is based on [28, M.A. Jackson]. **Part attributes** are either constant or varying, i.e., **static** or **dynamic** attributes.

**Analysis Prompt 20** is\_static\_attribute: By a **static attribute**, $a : A$, we shall understand an attribute whose values are constants, i.e., cannot change.

**Analysis Prompt 21** is\_dynamic\_attribute: By a **dynamic attribute**, $a : A$, we shall understand an attribute whose values are variable, i.e., can change. Dynamic attributes are either **inert**, **reactive** or **active** attributes.

---

\(^{18}\)Note that we do not presently describe the units in which flow are measured.
**Analysis Prompt 22** *is_inert_attribute*: By an *inert attribute*, \(a:A\), we shall understand a dynamic attribute whose values only change as the result of external stimuli where these stimuli prescribe new values.

**Analysis Prompt 23** *is_reactive_attribute*: By a *reactive attribute*, \(a:A\), we shall understand dynamic attributes whose value, if they vary, change in response to external stimuli, where these stimuli come from outside the domain of interest.

**Analysis Prompt 24** *is_active_attribute*: By an *active attribute*, \(a:A\), we shall understand a dynamic attribute whose values change (also) of its own volition. Active attributes are either *autonomous*, *biddable* or *programmable* attributes.

**Analysis Prompt 25** *is_autonomous_attribute*: By an *is_autonomous_attribute*(\(a\)), we shall understand a dynamic active attribute whose values change value only “on their own volition”. The values of an autonomous attributes are a “law onto themselves and their surroundings”.

**Analysis Prompt 26** *is_biddable_attribute*: By a *biddable attribute*, \(a:A\), we shall understand a dynamic active attribute whose values are prescribed but may fail to be observed as such.

**Analysis Prompt 27** *is_programmable_attribute*: By a *programmable attribute*, \(a:A\), we shall understand a dynamic active attribute whose values can be prescribed.

Figure 2 captures an attribute value ontology.

![Attribute Value Ontology](image)

**Figure 2**: Attribute Value Ontology

**Example 22** *Road Transport System Attribute Categories*: These are examples of attribute categories of the road transport system of Sect. 6: *static*: link and hub locations, link lengths, automobile brand names; *inert*: ... TO COME ... *reactive*: ... TO COME ... *autonomous*: ... TO COME ... *biddable*: ... TO COME ... *programmable*: automobile position and automobile, link and hub histories.
1. Given a part \( p \) we can calculate its **static attributes**.

2. Given a part \( p \) we can calculate its **controllable attributes**, i.e., the biddable and programmable attributes.

3. And given a part \( p \) we can calculate its **monitorable attributes**, i.e., the inert, reactive and autonomous attributes.

4. These three sets make up all the attributes of part \( p \).

\[
\text{value} \\
1. \text{stat}_{\text{attr} \text{typs}}: \mathbb{P} \rightarrow \langle \mathbb{SA}_1 \times \mathbb{SA}_2 \times \ldots \times \mathbb{SA}_s \rangle \\
2. \text{ctrl}_{\text{attr} \text{typs}}: \mathbb{P} \rightarrow \langle \mathbb{CA}_1 \times \mathbb{CA}_2 \times \ldots \times \mathbb{CA}_c \rangle \\
3. \text{mon}_{\text{attr} \text{typs}}: \mathbb{P} \rightarrow \langle \mathbb{MA}_1 \times \mathbb{MA}_2 \times \ldots \times \mathbb{MA}_m \rangle \\
\text{axiom} \\
4. \forall p : \mathbb{P} \cdot \\
4. \quad \text{let } \langle \mathbb{SA}_1 \times \mathbb{SA}_2 \times \ldots \times \mathbb{SA}_s \rangle = \text{stat}_{\text{attr} \text{typs}}(p), \ \\
4. \quad \langle \mathbb{CA}_1 \times \mathbb{CA}_2 \times \ldots \times \mathbb{CA}_c \rangle = \text{ctrl}_{\text{attr} \text{typs}}(p), \ \\
4. \quad \langle \mathbb{MA}_1 \times \mathbb{MA}_2 \times \ldots \times \mathbb{MA}_m \rangle = \text{mon}_{\text{attr} \text{typs}}(p) \in \\
4. \quad \text{card}\{\mathbb{SA}_1, \mathbb{SA}_2, \ldots, \mathbb{SA}_s\} + \text{card}\{\mathbb{CA}_1, \mathbb{CA}_2, \ldots, \mathbb{CA}_c\} + \text{card}\{\mathbb{MA}_1, \mathbb{MA}_2, \ldots, \mathbb{MA}_m\} \end{equation}
\]

5. Given a part \( p \) we can calculate its static attribute values.

6. Given a part \( p \) we can calculate its controllable, i.e., the biddable and programmable attribute values.

\[
\text{value} \\
5. \text{stat}_{\text{attr} \text{vals}}: \mathbb{P} \rightarrow \mathbb{SA}_1 \times \mathbb{SA}_2 \times \ldots \times \mathbb{SA}_s \\
5. \text{stat}_{\text{attr} \text{vals}}(p) \equiv \ \\
5. \quad \text{let } \langle \mathbb{SA}_1 \times \mathbb{SA}_2 \times \ldots \times \mathbb{SA}_s \rangle = \text{stat}_{\text{attr} \text{typs}}(p), \ \\
5. \quad (\text{attr}_{\mathbb{SA}_1}(p), \text{attr}_{\mathbb{SA}_2}(p), \ldots, \text{attr}_{\mathbb{SA}_s}(p)) \end{equation} \\
6. \text{ctrl}_{\text{attr} \text{vals}}: \mathbb{P} \rightarrow \mathbb{CA}_1 \times \mathbb{CA}_2 \times \ldots \times \mathbb{CA}_c \\
6. \text{ctrl}_{\text{attr} \text{vals}}(p) \equiv \ \\
6. \quad \text{let } \langle \mathbb{CA}_1 \times \mathbb{CA}_2 \times \ldots \times \mathbb{CA}_c \rangle = \text{ctrl}_{\text{attr} \text{typs}}(p), \ \\
6. \quad (\text{attr}_{\mathbb{CA}_1}(p), \text{attr}_{\mathbb{CA}_2}(p), \ldots, \text{attr}_{\mathbb{CA}_c}(p)) \end{equation}
\]

3. **A Transcendental Transformation** – cf. s. 6.3 Pg. 40

It should be clear to the reader that in **domain analysis & description** we are reflecting on a number of **philosophical issues**. First and foremost on those of **epistemology** and **ontology**. In this section on a sub-field of epistemology, namely that of a number of issues of **transcendental** nature. We refer to [29, pp 878–880] [30, pp 807–810] [31, pp 54–55 (1998)].
Definition 17 Transcendental: By transcendental we shall understand the philosophical notion: the a priori or intuitive basis of knowledge, independent of experience.

A priori knowledge or intuition is central: By a priori we mean that it not only precedes, but also determines rational thought.

Definition 18 Transcendental Transformation: By a transcendental transformation we shall understand the philosophical notion: a transcendental "conversion" of one kind of knowledge into a seemingly different kind of knowledge.

Definition 19 Transcendentality: By transcendentality we shall here mean the philosophical notion: the state or condition of being transcendental.

Example 23 Transcendentality: We can speak of a bus in at least three senses:

(i) The bus as it is being "serviced" (maintained) at an automobile garage;
(ii) the bus as it "speeds" down its route; and
(iii) the bus as it "appears" (listed) in a bus time table.

The three senses are:

(i) as an endurant (here a part),
(ii) as a perdurant (as we shall see a behaviour), and
(iii) as an attribute19.

Example 23, we claim, reflects transcendentality as follows:

• We have knowledge of an endurant (i.e., a part) being an endurant.
• We are then to assume that the perdurant referred to in (ii) is an aspect of the endurant mentioned in (i) – where perdurants are to be assumed to represent a different kind of knowledge.
• And, finally, we are to further assume that the attribute mentioned in (iii) is somehow related to both (i) and (ii) – where at least this attribute is to be assumed to represent yet a different kind of knowledge.

4 Perdurants – cf. s. 6.4 Pg. 41

So the transcendental deduction to be performed here is that of associating with each part – "existing" in space – a behaviour – "existing" in time.

Perdurants can thus be explained in terms of a notion of state and a notion of time. We refer to Sect. 7.2 for a discussion of the concept of time.

19 – in this case rather: as a fragment of a bus time table attribute
To speak about behaviours, that is, to describe behaviours, we choose a model for behaviours. We choose that of CSP [32]. With CSP is associated the notions of processes (which serve to model behaviours), channels, ch, (which serve to model communication between behaviours), and output/input clauses: ch ! v, respectively ch ? which serves to express the offering of a value, v on channel ch, respectively the offering to accept such a value. We shall use these notions freely.

4.1 States – cf. s. 6.2.6 Pg. 37

Definition 20 State: By a state we shall understand any collection of parts or components or materials

4.2 On Actions, Events, Behaviours and Actors

To us perdurants are further, pragmatically, analysed into actions, events, and behaviours. We shall define these terms below. Common to all of them is that they potentially change a state. Actions and events are here considered atomic perdurants. For behaviours we distinguish between discrete and continuous behaviours.

4.2.1 Actors

Definition 21 Actor: By an actor we shall understand something that is capable of initiating and/or carrying out actions, events or behaviours

Actors will play an important rôle in our domain analysis & description. By what we learn from our study of Sørlander’s Philosophy some endurants (of a kind we shall introduce much later20) can, by a transcendental deduction, “become” perdurants some of which thereby “acting” in rôles of actors.

Example 24 Actors: Automobile endurants “transmogrify” into automobile perdurants which “subsume” rôles of humans in that we “include” humans in the form of automobile drivers in the non-deterministic behaviour automobile perdurants

4.2.2 Discrete Actions

Definition 22 Discrete Action: By a discrete action [33, Wilson and Shpall] we shall understand a foreseeable thing which deliberately and potentially changes a well-formed state, in one step, usually into another, still well-formed state, and for which an actor can be made responsible

Example 25 Discrete Actions: Here are some examples of discrete actions: the removal, i.e., closing of a street segment, i.e., a link, from a road net; the insertion of a street segment-between two street intersections, i.e., hubs, of a road net; and the removal of an automobile from the road net.

20)humans [Sect. 10.5 Pg. 66] and, although not a concept in [15, 18], their artifacts [Sect. 10.7 Pg. 66]
4.2.3 Discrete Events

Definition 23 Event: By an event we shall understand some unforeseen thing, that is, some ‘not-planned-for’ “action”, one which surreptitiously, non-deterministically changes a well-formed state into another, but usually not a well-formed state, for which no particular domain actor can be made responsible.

Example 26 Discrete Events: Here are some examples of discrete events: a mud slide which effectively blocks, i.e., closes, a link; and the crashing of two automobiles.

4.2.4 Discrete Behaviours

Definition 24 Discrete Behaviour: By a discrete behaviour we shall understand a set of sequences of potentially interacting sets of discrete actions, events and behaviours.

Example 27 Discrete Behaviours: Here are some examples of discrete behaviours: the drive of an automobile along a road net; the sequence of pumping and not-pumping, concurrent with and/or before/after opening and closing valves of a pipeline system; the waiting of an automobile stopped at a traffic light for it turning green; and the road (hub or link) “carrying” automobiles.

In this paper we shall omit consideration of concepts of continuous actions, events and behaviours.

4.3 Channels – cf. s. 6.4.2 Pg. 41

The fact that a part, \( p \) of sort \( P \) with unique identifier \( p_i \), has a mereology, for example the set of unique identifiers \( \{q_{a_1}, q_{b_1}, \ldots, q_{d_1}\} \) identifying parts \( \{q_{a_2}, q_{b_2}, \ldots, q_{d_2}\} \) of sort \( Q \), may mean that parts \( p \) and \( q \in \{q_{a_1}, q_{b_1}, \ldots, q_{d_1}\} \) may wish to exchange – for example, attribute – values, one way (from \( p \) to the \( q \)'s) or the other (vice versa) or in both directions.

Figure 3 Pg. 31 shows (left) two dotted rectangle box (part) and (right) two corresponding, rounded box (behaviour and channel) diagrams. We explain the figure: The left fragment of the figure intends to show a 1:1 Constellation of a single \( p:P \) box and a single \( q:Q \) part, respectively, indicating, within these parts, their unique identifiers and merologies. The right fragment of the figure intends to show a 1:n Constellation of a single \( p:P \) box and a set of \( q:Q \) parts, now with arrowed lines connecting the \( p \) part with the \( q \) parts. These lines are intended to show channels. We show them with two way arrows. We could instead have chosen one way arrows, in one or the other direction. The directions are intended to show a direction of value transfer. We have given the same channel names to all examples, \( ch_{PQ} \). We have ascribed channel message types \( MPQ \) to all channels.

Figure 4 shows an arrangement similar to that of Fig. 3 [next page], but for an m:n Constellation. The channel declarations corresponding to Figs. 3 and 4 are:

\[ \text{[next page]} \]

21 Of course, these names and types would have to be distinct for any one domain description.
Figure 3: Two Part and Behaviour/Channel Constellations: $u:p$ unique id, $p:m:p$ mereology $p$

\[
\text{channel} \\
[1] \quad \text{ch}_{PQ}[i,j]:MPQ \\
[2] \quad \{ \text{ch}_{PQ}[i,x]:MPQ \mid x:\{j,k,\ldots,l\} \} \\
[3] \quad \{ \text{ch}_{PQ}[p,q]:MPQ \mid p:\{x,y,\ldots,z\}, q:\{j,k,\ldots,l\} \}
\]

Since there is only one index $i$ and $j$ for channel [1], its declaration can be reduced. Similarly there is only one $i$ for declaration [2]:

\[
\text{channel} \\
[1] \quad \text{ch}_{PQ}:MPQ \\
[2] \quad \{ \text{ch}_{PQ}[x]:MPQ \mid x:\{j,k,\ldots,l\} \}
\]

7 The following description identities holds:

\[
7 \quad \{ \text{ch}_{PQ}[x]:MPQ \mid x:\{j,k,\ldots,l\} \} \equiv \text{ch}_{PQ}[j],\text{ch}_{PQ}[k],\ldots,\text{ch}_{PQ}[l],
\]
\begin{align*}
7 \{ \text{ch}_P Q[p,q]: X \mid p:\{x,y,\ldots,z\}, q:\{j,k,\ldots,l\} \} \equiv \\
7 \text{ch}_P Q[x,j], \text{ch}_P Q[x,k], \ldots, \text{ch}_P Q[x,l], \\
7 \text{ch}_P Q[y,j], \text{ch}_P Q[y,k], \ldots, \text{ch}_P Q[y,l], \\
7 \ldots, \\
7 \text{ch}_P Q[z,j], \text{ch}_P Q[z,k], \ldots, \text{ch}_P Q[z,l] 
\end{align*}

4.4 Behaviours

4.4.1 Behaviour Signatures – cf. s. 6.4.3 Pg. 41

We associate with each part, \( p:P \), a behaviour \( M_P \). Behaviours have, as first argument, their unique part identifier: \( \text{uid}_P(p) \). Behaviours evolves around a state in the form of a set of values: its possibly changing mereology, \( \text{mt:MT} \) and the attributes of the part.\(^{22}\) A behaviour signature is therefore:

\[ M_P: \text{ui:UI} \times \text{me:MT} \times \text{sa:stat attr typs(p)} \rightarrow \text{ca:ctrl attr typs(p)} \rightarrow \text{calc_i_o_chn refs(p)} \text{ Unit} \]

where (i) \( \text{ui:UI} \) is the unique identifier value and type of part \( p \); (ii) \( \text{me:MT} \) is the value and type mereology of part \( p \), \( \text{me} = \text{obs mereo}_P(p) \); (iii) \( \text{sa:stat attr typs(p)} \): static attribute types of part \( p:P \); (iv) \( \text{ca:ctrl attr typs(p)} \): controllable attribute types of part \( p:P \); (v) \( \text{calc}_i_o_chn refs(p) \) calculates channel references to the input channels reflecting the monitorable attributes of \( p \) and the input/output and the output channels designated in the mereology, \( \text{me} \), of \( p \).

4.4.2 Behaviour Definitions – cf. s. 6.4.4 Pg. 42

Let \( P \) be a composite sort defined in terms of endurant\(^{23}\) sub sorts \( E_1, E_2, \ldots, E_n \). The behaviour description translated from \( p:P \), is composed from a behaviour description, \( M_P \), relying on and handling the unique identifier, mereology and attributes of part \( p \) to be translated with behaviour descriptions \( \beta_1, \beta_2, \ldots, \beta_n \) where: \( \beta_1 \) is translated from \( e_1:E_1 \), \( \beta_2 \) is translated from \( e_2:E_2 \), \ldots, and \( \beta_n \) is translated from \( e_n:E_n \). The domain description transcendental schema below “formalises” the above.

**Transcendental Schema 1**

\[
\begin{array}{l}
\text{value} \\
\text{Translate}_P : P \rightarrow \text{RSL}^\dagger \text{Text} \\
\text{Translate}_P(p) \equiv \\
\text{let } \text{ui} = \text{uid}_P(p), \text{me} = \text{obs mereo}_P(p), \\
\quad \text{sa} = \text{stat attr vals(p)}, \text{ca} = \text{ctrl attr vals(p)}, \\
\quad \text{MT} = \text{mereo type(p)}, \text{ST} = \text{stat attr typs(p)}, \text{CT} = \text{ctrl attr typs(p)}, \\
\quad \text{IOR} = \text{calc i_o chn refs(p)}, \text{IOD} = \text{calc all ch dcls(p)} \in \text{channel}
\end{array}
\]

\(^{22}\)We leave out consideration of possible components and materials of the part.\(^{23}\) – structures or composite
IOD

value

\( \mathcal{M}_P: P_{UI} \times MT \times ST \times CT \times IOR \times Unit \)

\( \mathcal{M}_P(u_i, m_e, s_t, c_a) \equiv B_P(u_i, m_e, s_t)(c_a) \)

\( \Rightarrow \text{Translate}_{P_1}(\text{obs\_endurant\_sorts}_{E_1}(p)) \)

\( \Leftrightarrow \text{Translate}_{P_2}(\text{obs\_endurant\_sorts}_{E_2}(p)) \)

\( \Leftrightarrow \ldots \)

\( \Leftrightarrow \text{Translate}_{P_n}(\text{obs\_endurant\_sorts}_{E_n}(p)) \)

end

Expression \( B_P(u_i, m_e, s_t)(c_a, p_a) \) stands for the \textit{behaviour definition body} in which the names \( u_i, m_e, s_t, c_a \) and \( p_a \) are bound to the \textit{behaviour definition head}, i.e., the left hand side of the \( \equiv \). Endurant sorts \( E_1, E_2, \ldots, E_n \) are obtained from the \texttt{observe\_endurant\_sorts} prompt, Page 17. We informally explain the \texttt{Translate}_{p_i} function. It takes endurants and produces RSL+Text. Resulting texts are bracketed: \( \langle rsl\_text \rangle \). For the case that an endurant is a structure there is only its elements to compile; otherwise Schema 2 is as Schema 1.

\begin{center}
\textbf{Transcendental Schema 2}
\end{center}

\begin{center}
\begin{align*}
\text{value} \\
\text{Translate}_{P}(p) & \equiv \\
& \text{Translate}_{P_1}(\text{obs\_endurant\_sorts}_{P_1}(p)) \\
\Leftrightarrow \text{Translate}_{P_2}(\text{obs\_endurant\_sorts}_{P_2}(p)) \\
\Leftrightarrow \ldots \\\n\Leftrightarrow \text{Translate}_{P_n}(\text{obs\_endurant\_sorts}_{P_n}(p)) \\
\end{align*}
\end{center}

Let \( P \) be a composite sort defined in terms of the concrete type \( Q\text{-set} \). The process definition compiled from \( p: P \), is composed from a process, \( \mathcal{M}_P \), relying on and handling the unique identifier, the mereology and the attributes of process \( p \) as defined by \( P \) operating in parallel with processes \( q: \text{obs\_part}_{Qs}(p) \). The domain description “compilation” schematic below “formalises” the above.

\begin{center}
\textbf{Transcendental Schema 3}
\end{center}

\begin{center}
\begin{align*}
type \\
Qs &= Q\text{-set} \\
value \\
qu:Q\text{-set} &= \text{obs\_part}_{Qs}(p) \\
\text{Translate}_{P}(p) & \equiv \\
& \text{let } u_i = \text{uid}_{P}(p), m_e = \text{obs\_mereo}_{P}(p), \\
& \text{sa} = \text{stat\_attr\_vals}(p), c_a = \text{ctrl\_attr\_vals}(p) \\
\end{align*}
\end{center}
Transcendental Schema 4

\[ \text{is\_atomic}(p) \]

value \\
\text{Translate}_P(p) \equiv \\
\quad \text{let } ui = \text{uid}_P(p), \ me = \text{obs\_mero}_P(p), \\
\quad \text{sa} = \text{stat\_attr\_vals}(p), \ ca = \text{ctrl\_attr\_vals}(p), \\
\quad \text{ST} = \text{stat\_attr\_typs}(p), \ CT = \text{ctrl\_attr\_typs}(p), \\
\quad \text{IOR} = \text{calc\_i\_o\_chn\_refs}(p), \ IOD = \text{calc\_all\_chs}(p) \text{ in} \\
\quad \bowtie \text{channel} \\
\quad \text{IOD} \text{ value} \\
\quad \mathcal{M}_P : P \times MT \times ST \ \text{CT} \ \text{IRO} \ \text{Unit} \\
\quad \mathcal{M}_P (ui, me, sa)(ca) \equiv B_P (ui, me, sa)(ca) \ \triangleright \\
\quad \{ \bowtie, \bowtie \ \text{Translate}_{Q}(q)|q:Q \in qs \} \\
\end{end}

Transcendental Schema 5

Core Behaviour

The core processes can be understood as never ending, “tail recursively defined” processes:

\[ B_P : \text{uid}\_P \times \text{me}\_MT \times \text{sa}\_SA \rightarrow \text{ct}\_CT \rightarrow \text{in} \ \text{in\_chns}(p) \ \text{in, out} \ \text{in\_out\_chns}(me) \ \text{Unit} \] \\
\[ B_P (p)(ui, me, sa)(ca) \equiv \\
\quad \text{let } (me', ca') = \mathcal{F}_P (ui, me, sa)(ca) \text{ in } \mathcal{M}_P (ui, me', sa)(ca') \text{ end} \] \\
\[ \mathcal{F}_P : \text{P} \times \text{MT} \times \text{ST} \rightarrow \text{CT} \rightarrow \text{in\_out\_chns}(me) \rightarrow \text{MT} \times \text{CT} \]
4.5 Initial Running Systems – cf. s. 6.4.5 Pg. 43

To round it all off a narrative and a formalisation must be done of “a running system”. Up till now the behaviours for all relevant parts have been defined. Now a decision must be made as to which of these are the basis for an initial system. There may be several candidates for initial running systems, that is, collection of concurrently operating behaviours. So the domain analyser cum describer selects all or some candidates. For each the chosen behaviours are properly initialised. And that is that!

5 A Coin Has Two Sides

The transcendental deduction that “turns” parts into behaviours can also be interpreted as follows: The part and the “corresponding” behaviour “exist” at one and the same time: the part is characterised by its internal qualities, and these are the arguments, in one form or another of the behaviour. The properties of the internal qualities of parts, expressed, for example, in the form of axioms, hold for all times (a concept not present in the treatment of endurants), and are to be maintained by the corresponding behaviours, as expressed, for example, in pre/post conditions. Let us recall essential “features” of parts and behaviours. For parts, \( p : P \), we can generally express the following:

- \( \text{uid}_P : P \rightarrow \Pi \)
- \( \text{obs}_{\text{mereo}}P : P \rightarrow \mathcal{E}(\Pi_1, \Pi_2, ..., \Pi_m) \)
- \( \text{attr}_{sA_1} : P \rightarrow sA_1 \) is static attribute
- \( \text{attr}_{sA_n} : P \rightarrow sA_n \) is static attribute
- \( \text{attr}_{cA_1} : P \rightarrow cA_1 \) is controllable attribute
- \( \text{attr}_{cA_n} : P \rightarrow cA_n \) is controllable attribute
- \( \text{attr}_{mA_1} : P \rightarrow mA_1 \) is monitorable attribute
- \( \text{attr}_{mA_m} : P \rightarrow mA_m \) is monitorable attribute

where \( n_s \geq 0 \), \( n_c \geq 0 \), and \( n_m \geq 0 \). For “corresponding” behaviours, \( \mathcal{M}_P \), we have (cf. Process Schema 1 [pp. 32]):

```plaintext
let ui = \text{uid}_P(p), me = \text{obs}_{\text{mereo}}P(p),
sv = \text{stat}_{\text{attr}}\text{vals}(p), cv = \text{ctrl}_{\text{attr}}\text{vals}(p),
MT = \text{mereo}_{\text{type}}(p), ST = \text{stat}_{\text{attr}}\text{typs}(p), CT = \text{ctrl}_{\text{attr}}\text{typs}(p),
IOR = \text{calc}_{\text{chan}}\text{refs}(p), IOD = \text{calc}_{\text{all}}\text{ch}_\text{dcls}(p) \in
\left\langle \text{channel} \right\rangle
\text{IOD} \text{value}
\mathcal{M}_P : ui:P,ui!MT \times me:MT \times sv:ST \times cv:CT \times IOR \times Unit
\mathcal{M}_P(ui,me,sv)(cv) \equiv B_P(ui,me,sv)(cv) \Rightarrow
end
```

We leave it to the reader to study these two sets of formulas.
6.1 The Universe of Discourse

The universe of discourse is road transport systems. We analyse & describe not the class of all road transport systems but a representative subclass, UoD, is structured into such notions as a road net, RN, of hubs, H, (intersections) and links, L, (street segments between intersections); a fleet of automobiles, FA, of automobiles, A; et cetera. See Fig. 5

6.2 Endurants

6.2.1 Structures

There is the universe of discourse, UoD. It is structured into

- a road net, RN, a structure, and
- a fleet of automobiles, FA, a structure.

type

8 UoD axiom ∀ uod:UoD • is_structure(uod).
9 RN axiom ∀ rn:RN • is_structure(rn).
10 FA axiom ∀ fa:FA • is_structure(fa).

value

9 obs_RN: UoD → RN
10 obs_FA: UoD → FA

6.2.2 Parts, Components and Materials

11 The road net consists of
   a a structure, SH, of hubs and
   b a structure, SL, of links.

12 The fleet of automobiles consists of
   a a set, As, of automobiles.

type

11a SH axiom ∀ sh:SH • is_structure(sh)
11b SL axiom ∀ sl:SL • is_structure(sl)
12a As = A-set

value

11a obs_SH: RN → SH
11b obs_SL: RN → SL
12a obs_AS: FA → AS
### 6.2.3 Parts – cf. s. 2.5.1 pp. 15
See Description Prompt 3, Pg. 18.

13 The structure of hubs is a set, $sH$, of atomic hubs, $H$.
14 The structure of links is a set, $sL$, of atomic links, $L$.
15 The structure of automobiles is a set, $sA$, of atomic automobiles, $A$.

#### Value
13 $\text{obs}_{sH}: sH \rightarrow sH$
14 $\text{obs}_{sL}: sL \rightarrow sL$
15 $\text{obs}_{sA}: sA \rightarrow sA$

### 6.2.4 Components – cf. s. 2.5.2 pp. 18
See Description Prompt 4, Pg. 19.

To illustrate the concept of components we describe timber yards, waste disposal areas, road material storage yards, automobile scrap yards, and the like as special "cul-de-sac" hubs with components. Here we describe road material storage yards.

16 Hubs may contain components, but only if the hub is connected to exactly one link.
17 These "cul-de-sac" hub components may be such things as Sand, Gravel, Cobble Stones, Asphalt, Cement or other.

#### Value
16 $\text{has\ components}: H \rightarrow \text{Bool}$
17 $\text{KS} = \langle \text{Sand}, \text{Gravel}, \text{CobbleStones}, \text{Asphalt}, \text{Cement} \rangle - \text{set}$
18 $\text{obs\ components}_H: H \rightarrow \text{KS}$
19 $\text{pre: obs\ components}_H(h) \equiv \text{card} \text{ mereo}(h) = 1$

### 6.2.5 Materials – cf. s. 2.5.3 pp. 20
See Description Prompt 5, Pg. 20.

To illustrate the concept of materials we describe waterways (river, canals, lakes, the open sea) along links as links with material of type water.

18 Links may contain material.
19 That material is water, $W$.

#### Value
18 $\text{obs\ material}: L \rightarrow W$
19 $\text{pre: obs\ material}(l) \equiv \text{has\ material}(h)$

### 6.2.6 States – cf. s. 4.1 pp. 29

20 Let there be given a universe of discourse, $rts$. It is an example of a state.

From that state we can calculate other states.

21 The set of all hubs, $hs$.
22 The set of all links, $ls$.
23 The set of all hubs and links, $hls$.
24 The set of all automobiles, $as$.
25 The set of all parts, $ps$.

#### Value
20 $\text{rts}: \text{UoD}$
21 $\text{hs}: H - \text{set} \equiv \text{obs}_H(\text{obs}_{sH}(\text{obs}_{RN}(rts)))$
22 $\text{ls}: L - \text{set} \equiv \text{obs}_L(\text{obs}_{sL}(\text{obs}_{RN}(rts)))$
23 $\text{hls}: (H \mid L) - \text{set} \equiv \text{hs} \cup \text{ls}$
24 $\text{as}: A - \text{set} \equiv \text{obs}_A(\text{obs}_{FV}(rts))$
25 $\text{ps}: (H \mid L \mid \text{BC} \mid \text{B} \mid \text{A}) - \text{set} \equiv \text{hls} \cup \text{obs}_L \cup \text{obs}_A$

### 6.2.7 Unique Identifiers – cf. s. 2.6 pp. 21
See Description Prompt 6, Pg. 21

#### Part Identifiers

26 We assign unique identifiers to all parts.
27 By a road identifier we shall mean a link or a hub identifier.
28 Unique identifiers uniquely identify all parts.
   a All hubs have distinct [unique] identifiers.
   b All links have distinct identifiers.
   c All automobiles have distinct identifiers.
   d All parts have distinct identifiers.
Extract Parts from Unique Identifiers

29 From the unique identifier of a part we can retrieve, \( \varphi \), the part having that identifier.

type
29 \( P = H \mid L \mid A \)

value
29 \( \varphi : H_{\text{UL}} \rightarrow H \mid L_{\text{UL}} \rightarrow L \mid A_{\text{UL}} \rightarrow A \)

\( \varphi(u) \equiv \text{let } p:H[L[A] \cdot p \in \text{ps} \& \text{uid}(p) = \text{ui} \text{ in p end} \)

Unique Identifier Constants: We can calculate:

30 the set, \( h_{ui}s \), of unique hub identifiers;

31 the set, \( l_{ui}s \), of unique link identifiers;

32 the map, \( h_{ui}m \), from unique hub identifiers to the set of unique link identifiers of the links connected to the zero, one or more identified hubs;

33 the map, \( h_{ui}m \), from unique link identifiers to the set of unique hub identifiers of the two hubs connected to the identified link;

34 the set, \( r_{ui}s \), of all unique hub and link, i.e., road identifiers;

35 the set, \( a_{ui}s \), of unique automobile identifiers;

We must express the following axioms:

36 All hub identifiers are distinct.

37 All link identifiers are distinct.

38 All automobile identifiers are distinct.

39 All part identifiers are distinct.

axiom
36 \( \text{card} \ h_{ui}s = \text{card} \ h_{ui}s \)

37 \( \text{card} \ l_{ui}s = \text{card} \ l_{ui}s \)

38 \( \text{card} \ a_{ui}s = \text{card} \ a_{ui}s \)

39 \( = \text{card} h_{ui}s + \text{card} l_{ui}s + \text{card} a_{ui}s \)

See Description Prompt 7, Pg. 22

40 The mereology of hubs is a triple: (i) the set of all automobile identifiers\(^{24}\), (ii) the set of unique identifiers of the links that it is connected to and the set of all unique identifiers of all automobiles,\(^{25}\), and (iii) an empty set.\(^{26}\)

41 The mereology of links is a triple: (i) the set of all automobile identifiers, (ii) the set of the two distinct hubs they are connected to, and (iii) an empty set.

42 The mereology of an automobiles is a triple: (i) an empty set, (ii) an empty set, and (iii) the set of the unique identifiers of all links and hubs\(^{27}\).

43 Empty sets are modelled as empty sets of tokens where tokens are further undefined.

\( ^{24} \)This is just another way of saying that the meaning of hub mereologies involves the unique identifiers of all the automobiles that might pass through the hub is of interest to it.

\( ^{25} \) Its link identifiers designate the links, zero, one or more, that a hub is connected to is of interest to both the hub and that these links is interested in the hub.

\( ^{26} \) The hubs are not “proactive”, i.e., that the universe of discourse have no parts that are interested in the hub.

\( ^{27} \) That the automobile might pass through
We can express some additional axioms, in this case for relations between hubs and links:

44 If hub, h, and link, l, are in the same road net,

45 and if hub h connects to link l then link l connects to hub h.

axiom
44 \( \forall h:H \cdot h \in hs \land l \in ls \Rightarrow \)
44 \( \text{let } (\_uis\_)=\text{mereo}_H(h) (\_uis\_)=\text{mereo}_L(l) \)
44 \( \text{in } \text{uis}_L(h) \in uis \Rightarrow \text{uis}_H(h) \in uis \end{end}

More mereology need be expressed – but we leave, to the reader, to narrate and formalise those.

6.2.10 Part Attributes – cf. s. 2.8 pp. 23

We treat part attributes, sort by sort. See Description Prompt 8, Pg. 24

Hubs: We show just a few attributes:

46 There is a hub state. It is a set of pairs, \( (f,l) \) of link identifiers, where these link identifiers are in the mereology of the hub. The meaning of the hub state, in which, e.g., \( (f,l) \) is an element, is that the hub is open, "green", for traffic from link \( f \) to link \( l \). If a hub state is empty then the hub is closed, i.e., "red" for traffic from any connected links to any other connected links.

47 There is a hub state space. It is a set of hub states. The meaning of the hub state space is that its states are all those the hub can attain. The current hub state must be in its state space.

48 Hub traffic history: Since we can think rationally about it, it can be described. We model hub traffic history as a hub attribute: the recording, per unique automobile identifier, of the time ordered presence, APos, in the hub of these automobiles.

49 The link identifiers of hub states must be in the set, \( i_{ui,s} \), of the road net’s link identifiers.

axiom
49 \( \forall h:H \cdot h \in hs \Rightarrow \)
49 \( \text{let } hr = \text{attr}_{H}(h) \text{ in } \)
49 \( \forall (i_{ui,l;i_{ui,l}'}):(L_{UI} \times L_{UI}) \cdot \)
49 \( \{i_{ui,l;i_{ui,l}'}\} \in hr \Rightarrow \{i_{ui,l;i_{ui,l}'}\} \subseteq i_{ui,s} \end{end}

Links: We show just a few attributes:

50 There is a link state. It is a set of pairs, \( (h_f,h_l) \), of distinct hub identifiers, where these hub identifiers are in the mereology of the link. The meaning of a link state in which \( (h_f,h_l) \) is an element is that the link is open, "green", for traffic from hub \( h_f \) to hub \( h_l \). Link states can have either 0, 1 or 2 elements.

51 There is a link state space. It is a set of link states. The meaning of the link state space is that its states are all those the which the link can attain. The current link state must be in its state space. If a link state space is empty then the link is (permanently) closed. If it has one element then it is a one-way link. If a one-way link, \( l \), is imminent on a hub whose mereology designates that link, then the link is a "trap", i.e., a "blind cul-de-sac".

52 Link traffic history: Since we can think rationally about it, it can be described. We model link traffic history as an attribute: the recording, per unique automobile identifier, of the time ordered positions, APos, along the link (from one hub to the next), of these automobiles.

53 The hub identifiers of link states must be in the set, \( h_{ui,s} \), of the road net’s hub identifiers.
We show just a few attributes: We illustrate but a few attributes:

54 Automobiles have a time attribute.

55 Automobiles have static number plate registration numbers.

56 Automobiles have dynamic positions on the road net:

a. either at a hub identified by some \( h_{ui} \).

b. or on a link, some fraction, \( \frac{\text{frac}}{\text{Fract}} \) down an identified link, \( L_{ui} \), from one of its identified connecting hubs, \( f_{hi}, \) in the direction of the other identified hub, \( t_{ui} \).

c. Automobiles, like elephants, never forget: they remember their timed positions of the past.

d. and the current position is the first element of this past!

\[\text{type}\]

\[\text{54} \quad \forall a: A \rightarrow T\]

\[\text{55} \quad \text{RegNo} \rightarrow \text{atHub} \mid \text{onLink} \rightarrow \text{programmable}, \text{df.27 pp.26}\]

\[\text{56a} \quad \text{atHub} :: h_{ui} \in H_{UI}\]

\[\text{56b} \quad \text{onLink} :: f_{hi} \in H_{UI} \times L_{ui} \times \text{frac} \times th_{ui} \in H_{UI}\]

\[\text{56c} \quad \text{A_Hist} = (T \times \text{APos})^* \rightarrow (\text{programmable}, \text{df.27 pp.26})\]

\[\text{56d} \quad \forall a:A \rightarrow \text{APos}\]

Obvious attributes that are not illustrated are those of velocity and acceleration, forward or backward movement, turning right, left or going straight, etc.

The acceleration, deceleration, even velocity, or turning right, turning left, moving straight, or forward or backward are seen as command actions. As such they denote actions by the automobile — such as pressing the accelerator, or lifting accelerator pressure or braking, or turning the wheel in one direction or another, etc. As actions they have a kind of counterpart in the velocity, the acceleration, etc. attributes.

6.2.11 Discussion of Edurants, I

In Items 48 Pg. 39 and 52 Pg. 39, we illustrated an aspect of domain analysis & description that may seem, and at least some decades ago would have seemed, strange: namely that if we can think, hence speak, about it, then we can model it “as a fact” in the domain. The case in point is that we include among hub and link attributes their histories of the timed whereabouts of automobiles.28

6.2.12 Discussion of Edurants, II

We have chosen to model some discrete endurants as structures others as parts (usually composite). Those choices are made mostly to illustrate that the domain analysis & description has a choice. If a choice is made to model a discrete endurant as a structure then it entails that the domain analysis & description does not wish to “implement” that discrete endurant as a behaviour separate from its sub-endurants; If the choice is made to model a discrete endurant as a part then it entails that the domain analysis & description wishes to “implement” that discrete endurant as a behaviour separate from its sub-endurants. The following discrete endurants which are modelled as structures above, could, instead, if modelled as parts, have the entailed behaviours reflect the following possibilities: road net, \( n_{RN} \): The road net behaviour could be that of a road net authority charged with building, servicing, operating and maintaining the road net. Building and maintaining the road net could mean the insertion of new or removal of old links or hubs. Operating the road net could mean the gathering of automobile traffic statistics, the setting of hub states (traffic signal monitoring and control), etc. aggregate of automobiles, \( ps_{PA} \): The aggregate of automobiles could be that of one or more automobile clubs, etc.

6.3 Transcendentality

We refer to Sect. 6.3 Defn. 23 Page 28.

Example 28 A Case of Transcendentality: We refer to the following example: We can speak of an automobile in at least three senses:

- The automobile as it is being maintained, serviced, refueled;
- the automobile as it “speeds” down its route; and
- the automobile as it “appears” (listed) in car registries or advertisements.

The three senses are:

- as a part,
- as a behaviour, and
- as an attribute29

28In this day and age of road cameras and satellite surveillance these traffic recordings may not appear so strange: We now know, at least in principle, of technologies that can record approximations to the hub and link traffic attributes.

29In this case rather: as a fragment of an attribute
### 6.4 Perdurants – cf. s. 4 pp. 28

#### 6.4.1 States

**Constants:** We refer to Sect. 6.2.6 Pg. 37, and to App. 4.1 Pg. 29. We assume, as a constant, an arbitrarily selected universe of discourse, \( uod \), and calculate from \( uod \) all its endurants.

\[
\begin{align*}
\text{value} & : uodD \\
hs & : H\text{-set} \equiv \text{obs}_H(\text{obs}_{RN}(rts)) \\
ls & : L\text{-set} \equiv \text{obs}_L(\text{obs}_{RN}(rts)) \\
hls & : (H \cup L)\text{-set} \equiv hs \cup ls \\
as & : A\text{-set} \equiv \text{obs}_A(\text{obs}_{FV}(rts))
\end{align*}
\]

**Indexed States:** We shall index automobiles using the unique identifiers of these parts.

\[
\begin{align*}
\text{type} & : A_{ui} \\
\text{value} & : \{a_{ui} : A_{ui} : \text{uid}(a_{ui}) \}
\end{align*}
\]

#### 6.4.2 Channels – cf. s. 4.3 pp. 30

We shall argue for hub-to-link channels based on the mereologies of those parts. Hub parts may be topologically connected to any number, 0 or more, link parts. Only instantiated road nets knows which. Hence there must be channels between any hub behaviour and any link behaviour. Vice versa: link parts will be connected to exactly two hub parts. Hence there must be channels from any link behaviour to two hub behaviours. See the figure below:

**Channel Message Types:** We ascribe types to the messages offered on channels.

58 Hubs and links communicate, both ways, with one another, over channels, \( hlch \), whose indexes are determined by their mereologies.

59 Hubs send one kind of messages, links another.

60 Automobiles offer their current, timed positions to the road element, hub or link they are on, one way.

\[
\begin{align*}
\text{type} & : HLM_{\text{Msg}}, LHM_{\text{Msg}} \\
98 & : HL_{\text{Msg}} = HLM_{\text{Msg}} | LFM_{\text{Msg}} \\
99 & : A_{\text{R}_{\text{Msg}}} = T \times A\text{Pos}
\end{align*}
\]

**Channel Declarations**

61 This justifies the channel declaration which is calculated to be:

\[
\begin{align*}
\text{channel} & : \\
98 & : \{ hlch[hui,lui] : HLM_{\text{Msg}} \\
99 & \cup \{ hlch[hui,lui] : LHM_{\text{Msg}} \\
100 & : \{ a_{\text{r}_{\text{ch}}[\cdot,\cdot]} : A_{\text{R}_{\text{Msg}}} \\
101 & : a_{\text{ui}} : A_{\text{UI}}, r_{\text{ui}} : R_{\text{UI}} \in a_{\text{ui}}s \land r_{\text{ui}} \in r_{\text{ui}}(s)
\end{align*}
\]

We shall argue for automobile to road element channels based on the mereologies of those parts. Automobiles need communicate to all hubs and all links.

62 This justifies the channel declaration which is calculated to be:

\[
\begin{align*}
\text{channel} & : \\
102 & : \{ hlch[hui,lui] : HL_{\text{Msg}} \\
103 & \cup \{ hlch[hui,lui] : L_{\text{UI}}_{\text{Msg}} \in l_{\text{ui}}s \land i \in l_{\text{ui}}m(l_{\text{ui}}) \}
\end{align*}
\]

#### 6.4.3 Behaviour Signatures – cf. s. 4.4.1 pp. 32

We first decide on names of behaviours. In Sect. 4.4.2, Pages 32–34, we gave schematic names to behaviours of the form \( M_P \). We now assign mnemonic names: from part names to names of transcendentally interpreted behaviours and then we assign signatures to these behaviours.

63 \( \text{hub}_{hui} \):

- a there is the usual “triplet” of arguments: unique identifier, mereology and static attributes;
- b then there are the programmable attributes;
- c and finally there are the input/output channel references: first those allowing communication between hub and link behaviours, and d and then those allowing communication between hub and automobile behaviours.
6.4.4 Behaviour Definitions – cf. s. 4.4.2 pp. 32

We define the behaviours in a different order than the treatment of their signatures. We “split” definition of the automobile behaviour into the behaviour of automobiles when positioned at a hub, and into the behaviour automobiles when positioned at a link. In both cases the behaviours include the “idling” of the automobile, i.e., its “not moving”, standing still.

Automobiles:

66 We abstract automobile behaviour at a Hub (hui).

67 The automobile remains at that hub, “idling”,

68 informing the hub behaviour,

69 or, internally non-deterministically,

a moves onto a link, tl, whose “next” hub, identified by thui, is obtained from the mereology of the link identified by tlui;

b informs the hub it is leaving and the link it is entering of its initial link position,

c whereupon the automobile resumes the automobile behaviour positioned at the very beginning (0) of that link,

70 or, again internally non-deterministically,

71 the automobile “disappears — off the radar”!

72 We abstract automobile behaviour on a Link.

a Internally non-deterministically, either

i the automobile remains, “idling”, i.e., not moving, on the link,

ii however, first informing the link of its position,

b or

i if if the automobile’s position on the link has not yet reached the hub, then

A then the automobile moves an arbitrary small, positive Real-valued increment along the link

B informing the hub of this new position,
C while assuming being an automobile at the new position, or

\[ ii \text{ else,} \]

A while obtaining a “next link” from the mereology of the hub (where that next link could very well be the same as the link the automobile is about to leave),

B the link behaviour informs both the link and the imminent hub that it is now at that hub, identified by \( h_{\text{th}} \),

C whereupon the automobile resumes the vehicle behaviour positioned at that hub;

c or

d the automobile “disappears — off the radar”!

\[
\begin{align*}
72 & \text{automobile}_{ui}(\{u_{ri},\{\},ruis\},\text{rno}) \\
72(a) & \text{hub} = \text{fract}(\text{th}_{ui},\{\},\text{th}_{ui}) \equiv \text{th}_{ui} \\
72(b) & \text{if not get-at}_{\text{hub}}(f) \text{ then} \\
72(b)i & \text{(let incr = increment(f) in} \\
72(b)i & \text{let onl = (}(u_{ri},h_{ui},\text{incr},\text{th}_{ui})\text{ in} \\
72(b)ii & \text{let onl = (}(u_{ri},h_{ui},\text{incr},\text{th}_{ui})\text{ in} \\
72(b)ii & \text{automobile}_{ui}(\{u_{ri},\{\},ruis\},\text{rno})(vp) \\
72(b)i & \text{automobile}_{ui}(\{u_{ri},\{\},ruis\},\text{rno})(vp)) \\
72(b) & \text{end end} \\
72(b) & \text{end} \\
72(b)i & \text{let next}_{\text{th}}(ui) = \text{mreo}(\text{vp}(\text{th}_{ui})) \text{ in} \\
72(b)i & \text{automobile}_{ui}(\{u_{ri},\{\},ruis\},\text{rno}) \\
72(b)i & \text{automobile}_{ui}(\{u_{ri},\{\},ruis\},\text{rno})(\text{at}(\text{th}_{ui} \text{ui},\text{th}_{ui},\text{next}_{\text{th}}(ui))) \text{ end} \\
72(b) & \text{end} \\
72c & \text{stop} \\
72d & \text{increment: Fract} \rightarrow \text{Frac}
\end{align*}
\]

**Hubs:** We model the hub behaviour vis-a-vis automobiles.

73 The hub behaviour

\[
\begin{align*}
& \text{a non-deterministically, externally offers} \\
& \text{b to accept timed automobile positions —} \\
& \text{c which will be at the hub, from some vehicle,} \nu_{ui} \\
& \text{d the timed automobile hub position is appended to the front of that automobile's entry in the hub's traffic table;} \\
& \text{e whereupon the hub proceeds as a hub behaviour with the updated hub traffic table.} \\
& \text{f The hub behaviour offers to accept from any automobile.}
\end{align*}
\]

\[
\begin{align*}
& \text{g A post condition expresses what is really a proof obligation: that the hub traffic, } h' \text{ satisfies the axiom of the endurant hub traffic attribute Item 48 Pg. 39.}
\end{align*}
\]

73 value

\[
\begin{align*}
73(a) & \text{hub}_{ui}(\{u_{ri},\{\},ruis\},h_{ui})(\text{hr},h_{i}) \equiv \text{hr} \\
73(b) & \{ \text{let m = ba}_{\text{ch}}(\text{hr},u_{ui})? \text{ in} \\
73(c) & \text{assert: } m=\text{at}_{\text{Hub}}(\{h_{ui},\text{u}_{ui}\}) \\
73(d) & \text{let h}' = h_{i} \uparrow \{ \text{at}_{\text{ui}} \rightarrow (m) \cdot \text{hr}(u_{ui}) \} \text{ in} \\
73(e) & \text{hub}_{ui}(\{u_{ri},\{\},ruis\},h_{ui})(\text{hr},h_{i})' \text{ end end} \\
73(f) & \{ \text{let } \text{A}_{\text{ui}}(\text{a}_{\text{ui}}) \equiv \text{dom } h_{i} \} \\
73(g) & \text{post: } \forall \text{A}_{\text{ui}}(\text{a}_{\text{ui}}) \in \text{dom } h_{i}' \\
73(g) & \Rightarrow \text{time ordered}(h_{i}'(a_{ui}))
\end{align*}
\]

**Links:** Similarly we model the link behaviour vis-a-vis automobiles.

74 The link behaviour non-deterministically, externally offers

75 to accept timed automobile positions —

76 which will be on the link, from some automobile, \( \nu_{ui} \).

77 The timed automobile link position is appended to the front of that automobile’s entry in the link’s traffic table;

78 whereupon the link proceeds as a link behaviour with the updated link traffic table.

79 The link behaviour offers to accept from any automobile.

80 A post condition expresses what is really a proof obligation: that the link traffic, \( l' \) satisfies the axiom of the endurant link traffic attribute Item 52 Pg. 39.

74 value

\[
\begin{align*}
74(a) & \text{link}_{ui}(\{\text{u}_{ui},\{\text{ui},\text{ui}_{ui}\},\text{lr}\}(\sigma,l),l_{ui}) \equiv \text{lr} \\
74(b) & \{ \text{let m = ba}_{\text{ch}}(\text{lr}_{ui},a_{ui})? \text{ in} \\
75 & \text{assert: } m=\text{on}_{\text{Link}}(\{\text{ui},\text{ui}_{ui}\}) \\
76 & \text{let h}' = h_{i} \uparrow \{ \text{at}_{\text{ui}} \rightarrow (m) \cdot \text{lr}(a_{ui}) \} \text{ in} \\
77 & \text{link}_{ui}(\{\text{u}_{ui},\{\text{ui},\text{ui}_{ui}\},\text{lr}\}(\sigma,l),l_{ui})' \text{ end end} \\
78 & \{ \text{let } \text{A}_{\text{ui}}(\text{a}_{\text{ui}}) \equiv \text{dom } l' \} \\
79 & \text{post: } \forall \text{A}_{\text{ui}}(\text{a}_{\text{ui}}) \in \text{dom } l' \\
80 & \Rightarrow \text{time ordered}(l'(a_{ui}))
\end{align*}
\]

**Preliminaries:** We recall the hub, link and the automobile states first mentioned in Sect. 6.2.6 Page 37.

\[
\begin{align*}
& \text{value} \\
21 & \text{hs:H-set} \equiv \equiv \text{obs}_{\text{sh}}(\text{obs}_{\text{SH}}(\text{obs}_{\text{RN}}(\text{rts}))) \\
22 & \text{ls:L-set} \equiv \equiv \text{obs}_{\text{SL}}(\text{obs}_{\text{SL}}(\text{obs}_{\text{RN}}(\text{rts}))) \\
24 & \text{as:A-set} \equiv \equiv \text{obs}_{\text{St}}(\text{obs}_{\text{FA}}(\text{rts})))
\end{align*}
\]
Starting Initial Behaviours: We are reaching the end of this domain modelling example. Behind us there are narratives and formalisations 8 Pg. 36 – 80 Pg. 43. Based on these we now express the signature and the body of the definition of a “system build and execute” function.

81 The system to be initialised is

a the parallel composition (∥) of

b the distributed parallel composition (∥{⋯⋯}) of

c all the hub behaviours,

d all the link behaviours, and

e all the automobile behaviours.

81 initial system: Unit → Unit
81 initial system() ≡
81c ∥ { h,ui,me,lu(h,ui,me,lu)(htrf,h)}
81c | h,ui,me,lu(h,ui,me,lu)(htrf,h)
81c h,ui,me,lu(h,ui,me,lu)(htrf,h)

6.5 Space and Time Considerations: A Specific Critique

We have not dealt with space and time in a fully satisfactory manner.

6.5.1 Space

We have referred, in Sect. 2, more-or-less explicitly, to space in Items 52 [pp. 59], 56 [pp. 40], 56b [pp. 40], 56c [pp. 40], and 56d [pp. 40]. And in Sect. 4. We have also referred to space: 60 Pg. 41, 69b Pg. 42, 72(a)i and 72(b)i Pg. 42; 72(b)iB and 72(b)iC Pg. 43; 72(b)iC, 73b and 73d Pg. 43; 75 and 77 Pg. 43. The Sect. 2 refers to the references of Sect. 4.

The problem here is the following: We have not analysed & described the fact that links may be single, double, triple, or more lane links, and hence not whether automobiles may be in identical link positions either moving in different lanes in the same direction; or “piling up” in crashes in the same lane whether “moving” (i.e., being) in the same direction or “moving” in opposite directions; or moving in opposite directions in different lanes. That problem can, of course, be avoided. One can simply augment the analysis & description by introducing appropriate link attributes and appropriate axioms concerning traffic and histories. We leave that the reader.

6.5.2 Time

We have In Sect. 2 referred to time in Items 48 Pg. 39, 52 Pg. 39; 54 and 56c Pg. 40. In Sect. 4 we have, correspondingly, also referred to time in Items 50 Pg. 41; 65c Pg. 42; 73b Pg. 43 and 73d Pg. 43; 75 Pg. 43 and 77 Pg. 43. It is not the trivial matter of representation of time. One representation of, for example the time this document that you are now reading was compiled, could be May 20, 2018: 11:20 am. Here we have only “refined” the time to within minutes. One could easily represent time “down” to picoseconds! No, the problem is that of how often we sample time. What do the formulas of Items 73b and 73d Pg. 43, and 75 and 77 Pg. 43 express? Are they sampled continuously or discretely?

We shall take the view, here, that the semantics of RSL + expresses a discrete sampling, that is, that each iteration of the automobile, the hub and the link behaviours, take time, but that the concurrently behaving automobiles indeed may assemble their timed positions simultaneously! This means that positions recorded for any one particular automobile are all distinct with respect to time, have different time designations.

6.6 The End!

Yes, this is the end of the main example.

6.7 Example Index

6.7.1 Sorts

<table>
<thead>
<tr>
<th>Part Sorts</th>
<th>EA</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>15</td>
<td>37</td>
</tr>
<tr>
<td>L</td>
<td>14</td>
<td>37</td>
</tr>
<tr>
<td>RN</td>
<td>9</td>
<td>36</td>
</tr>
<tr>
<td>sA</td>
<td>15</td>
<td>37</td>
</tr>
</tbody>
</table>
6.7.2 Types

Attribute Types

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Hist</td>
<td>Hist 56c, 40, 74</td>
<td></td>
</tr>
<tr>
<td>A: APos==atHub</td>
<td>programmable 56, 40</td>
<td></td>
</tr>
<tr>
<td>A: RegNo</td>
<td>static 55, 40</td>
<td></td>
</tr>
<tr>
<td>A: T</td>
<td>inert 54, 40</td>
<td></td>
</tr>
<tr>
<td>H: HΩ</td>
<td>static 47, 39</td>
<td></td>
</tr>
<tr>
<td>H: HΣ</td>
<td>programmable 46, 39</td>
<td></td>
</tr>
<tr>
<td>H: H</td>
<td>Traffic programmable 48, 39, 74</td>
<td></td>
</tr>
<tr>
<td>L: LΩ</td>
<td>static 50, 39</td>
<td></td>
</tr>
<tr>
<td>L: LΣ</td>
<td>programmable 50, 39</td>
<td></td>
</tr>
<tr>
<td>L: L</td>
<td>Traffic programmable 52, 39, 74</td>
<td></td>
</tr>
</tbody>
</table>

Mereology Types

<table>
<thead>
<tr>
<th>Mereology</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Mer=ES×ES×R UI-set</td>
<td>42, 38</td>
<td></td>
</tr>
<tr>
<td>H: Mer=V UI-set×L UI-set×ES</td>
<td>40, 38</td>
<td></td>
</tr>
<tr>
<td>L: Mer=V UI-set×H UI-set×ES</td>
<td>41, 38</td>
<td></td>
</tr>
</tbody>
</table>

Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: atHub:H UI</td>
<td>56a, 40</td>
</tr>
<tr>
<td>A: Frac=Real</td>
<td>56b, 40</td>
</tr>
<tr>
<td>A: onLink:H UI×L UI×Fract×H UI</td>
<td>56b, 40</td>
</tr>
<tr>
<td>ES=TOKEN-set</td>
<td>43, 38</td>
</tr>
</tbody>
</table>

Unique Identifier Types

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: UI</td>
<td>26, 37</td>
</tr>
<tr>
<td>H: UI</td>
<td>26, 37</td>
</tr>
<tr>
<td>L: UI</td>
<td>27, 37</td>
</tr>
<tr>
<td>R: UI</td>
<td>27, 37</td>
</tr>
<tr>
<td>R UI=H UI</td>
<td>27, 37</td>
</tr>
</tbody>
</table>

6.7.3 Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>μ (ν)</td>
<td>29, 38</td>
</tr>
</tbody>
</table>

6.7.4 Values

Part Constants

<table>
<thead>
<tr>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>as</td>
<td>24, 37</td>
</tr>
<tr>
<td>hs</td>
<td>23, 37</td>
</tr>
<tr>
<td>ls</td>
<td>21, 37</td>
</tr>
<tr>
<td>ps</td>
<td>25, 37</td>
</tr>
</tbody>
</table>

Unique Id. Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>a ui</td>
<td>35, 38</td>
</tr>
<tr>
<td>h ui</td>
<td>30, 38</td>
</tr>
<tr>
<td>hlu</td>
<td>32, 38</td>
</tr>
<tr>
<td>l ui</td>
<td>31, 38</td>
</tr>
<tr>
<td>r ui</td>
<td>34, 38</td>
</tr>
</tbody>
</table>

6.7.5 Channels

Channel Message Types

<table>
<thead>
<tr>
<th>Channel</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: R UI msg=(T×APos)</td>
<td>60, 41</td>
</tr>
<tr>
<td>A: attr, RegNo</td>
<td>56, 40</td>
</tr>
<tr>
<td>A: attr, T</td>
<td>54, 40</td>
</tr>
<tr>
<td>H: attr, HΩ</td>
<td>47, 39</td>
</tr>
<tr>
<td>H: attr, HΣ</td>
<td>46, 39</td>
</tr>
<tr>
<td>H: attr, H UI</td>
<td>48, 39</td>
</tr>
</tbody>
</table>

Channels

<table>
<thead>
<tr>
<th>Channel</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>automobile</td>
<td>65, 42</td>
</tr>
<tr>
<td>hub</td>
<td>63, 41</td>
</tr>
<tr>
<td>link</td>
<td>63, 41</td>
</tr>
</tbody>
</table>

6.7.6 Behaviours

Behaviours

<table>
<thead>
<tr>
<th>Behaviour</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>automobile:</td>
<td>65, 42</td>
</tr>
<tr>
<td>hub:</td>
<td>63, 41</td>
</tr>
<tr>
<td>link:</td>
<td>63, 41</td>
</tr>
</tbody>
</table>
Segment II: Space and Time

We have separated out a treatment of the notions of space and time as these are at the very basis of our ability to describe “the world”. That is, has deep implications for our attempt to relate the mundane activity of analysing & describing domains to the philosophical issue of "what can be described".

7 Space Time

The presentation of the domain analysis & description calculi avoided, in principle, references to space and time; but these concepts are there: “buried” as follows: endurants can be said to “exist” in space and perdurants to “exist” in time. We shall briefly examine these two concepts as they have been the concern of mathematicians. We shall not be interested in the physicists' spacetime mathematical model that fuses the three dimensions of space and the one dimension of time into a single four-dimensional continuum.

7.1 Space

Space is the boundless three-dimensional extent in which objects and events have relative position and direction. Physical space is often conceived in three linear dimensions, although modern physicists usually consider it, with time, to be part of a boundless four-dimensional continuum known as spacetime. The concept of space is considered to be of fundamental importance to an understanding of the physical universe. However, disagreement continues between philosophers over whether it is itself an entity, a relationship between entities, or part of a conceptual framework.

To us space is a conceptual framework. That is, it is not an entity, hence neither an endurant nor a perdurant. Here we shall primarily look at space as a mathematical construction. In Sect. 10 we shall widen that consideration considerably.

7.1.1 Topological Space

One notion of space, in mathematics, is that of a Hausdorff (or topological) space:

Definition 25 Topological Space: A topological space is an ordered pair \((X, \tau)\), where \(X\) is a set and \(\tau\) is a collection of subsets of \(X\), satisfying the following axioms:

- The empty set and \(X\) itself belong to \(\tau\).
- Any (finite or infinite) union of members of $\tau$ still belongs to $\tau$.

- The intersection of any finite number of members of $\tau$ still belongs to $\tau$.

The elements of $\tau$ are called open sets and the collection $\tau$ is called a topology on $X$.

### 7.1.2 Metric Space

A metric space is a set for which distances between all members of the set are defined. Those distances, taken together, are called a metric on the set. A metric on a space induces topological properties like open and closed sets, which lead to the study of more abstract topological spaces.

**Definition 26 Metric Space:** A **metric space** is an ordered pair $(M, d)$ where $M$ is a set and $d$ is a metric on $M$, i.e., a function

- $d : M \times M \to \mathbb{R}$

such that for any $x, y, z : M$, the following holds:

1. $d(x, y) \geq 0$ non-negativity or separation axiom
2. $d(x, y) = 0 \iff x = y$ identity of indiscernibles
3. $d(x, y) = d(y, x)$ symmetry
4. $d(x, z) \leq d(x, y) + d(y, z)$ subadditivity or triangle inequality

### 7.1.3 Euclidian Space

The notion of **Euclidian Space** is due to **Euclid of Alexandria** [325–265]. Euclid postulated

**Example 29 Euclid’s Postulates:**

- To draw a straight line from any point to any point.
- To produce [extend] a finite straight line continuously in a straight line.
- To describe a circle with any centre and distance [radius].
- That all right angles are equal to one another.
- [The parallel postulate] That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles.


---

**Example 30**  *Euclid’s Plane Geometry*: The Euclidean geometry informally described in Example 29 can be formally axiomatised by first introducing the sorts \( P \) and \( L \):

\[
\text{type} \quad P, L \\
\text{value} \quad [0] \; \text{obs}_P: L \to P\text{-infset} \\
\quad \text{parallel}: L \times L \to \text{Bool}
\]

Observe how the informal axiom in Example 29 has been modelled by the *observer function* \( \text{obs}_P \). It applies to lines and yields possibly infinite sets of points.

Now we can introduce the axioms proper:

\[
\text{axiom} \\
[1] \exists p,q:P \cdot p \neq q, \\
[2] \forall p,q:P \cdot p \neq q \Rightarrow \exists l:L \cdot p \in \text{obs}_P(l) \land q \in \text{obs}_P(l), \\
[3] \forall l:L \cdot \exists p:P \cdot p \notin \text{obs}_P(l), \\
[4] \forall l:L \cdot \exists p:P \cdot p \notin \text{obs}_P(l) \Rightarrow \exists l':L \cdot l \neq l' \land p \in \text{obs}_P(l') \land \text{parallel}(l,l')
\]

The concept of being parallel is modelled by the predicate symbol of the same name, by its signature and by axiom \( [4] \) ■

We leave it to the reader to reconcile the models of topological space, Defn. 25 [pp. 46], and metric space, Defn. 26 [preceding page], with the axiom systems of examples 29 [previous page] and 30.

### 7.2 Time

(i) A moving image of eternity;
(ii) The number of the movement in respect of the before and the after;
(iii) The life of the soul in movement as it passes from one stage of act or experience to another;
(iv) A present of things past: memory, a present of things present: sight, and a present of things future: expectations.


#### 7.2.1 Time — General Issues

In the next sections we shall focus on various models of time, and we shall conclude with a simple view of the operations we shall assume when claiming that an abstract type models time. These sections are far from complete. They are necessary, but, as a general treatment of notions of time, they are not sufficient. We refer the interested reader to special monographs: [34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44].
7.2.2 “A-Series” and “B-Series” Models of Time

Colloquially, in ordinary, everyday parlance, we think of time as a dense series of time points. We often illustrate time by a usually horizontal line with an arrow pointing towards the right. Sometimes that line arrowhead is labeled with either a $t$ or the word time, or some such name. J.M.E. McTaggart (1908, [36, 35, 44]) discussed theories of time around two notions:

- **“A-series”**: has terms like “past”, “present” and “future”.
- **“B-series”**: has terms like “precede”, “simultaneous” and “follow”.

McTaggart argued that the B-series presupposes the A-series: If $t$ precedes $t'$ then there must be a “thing” $t''$ at which $t$ is past and $t'$ is present. He argued that the A-series is incoherent: What was once ‘future’, becomes ‘present’ and then ‘past’; and thus events ‘will be events’, ‘are events’ and ‘were events’, that is, will have all three properties.

7.2.3 A Continuum Theory of Time

The following is taken from Johan van Benthem [34]: Let $P$ be a point structure (for example, a set). Think of time as a continuum; the following axioms characterise ordering ($<, =, >$) relations between (i.e., aspects of) time points. The axioms listed below are not thought of as an axiom system, that is, as a set of independent axioms all claimed to hold for the time concept, which we are encircling. Instead van Benthem offers the individual axioms as possible “blocks” from which we can then “build” our own time system — one that suits the application at hand, while also fitting our intuition.

Time is transitive: If $p < p'$ and $p' < p''$ then $p < p''$. Time may not loop, that is, is not reflexive: $p \not< p$. Linear time can be defined: Either one time comes before, or is equal to, or comes after another time. Time can be left-linear, i.e., linear “to the left” of a given time. The following is taken from Johan van Benthem [34]: Let $P$ be a point structure (for example, a set). Think of time as a continuum; the following axioms characterise ordering ($<, =, >$) relations between (i.e., aspects of) time points. The axioms listed below are not thought of as an axiom system, that is, as a set of independent axioms all claimed to hold for the time concept, which we are encircling. Instead van Benthem offers the individual axioms as possible “blocks” from which we can then “build” our own time system — one that suits the application at hand, while also fitting our intuition.

Time is transitive: If $p < p'$ and $p' < p''$ then $p < p''$. Time may not loop, that is, is not reflexive: $p \not< p$. Linear time can be defined: Either one time comes before, or is equal to, or comes after another time. Time can be left-linear, i.e., linear “to the left” of a given time. One could designate a time axis as beginning at some time, that is, having no predecessor times. And one can designate a time axis as ending at some time, that is, having no successor times. General, past and future successors (predecessors, respectively successors in daily talk) can be defined. Time can be dense: Given any two times one can always find a time between them. Discrete time can be defined.

<table>
<thead>
<tr>
<th>axiom</th>
<th>TRANS: Transitivity</th>
<th>IRREF: Irreflexivity</th>
<th>LIN: Linearity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \forall p,p',p'' : P \cdot p &lt; p' &lt; p'' \Rightarrow p &lt; p'' ]</td>
<td>[ \forall p : P \cdot p \not&lt; p ]</td>
<td>[ \forall p,p' : P \cdot (p=p' \lor p &lt; p' \lor p &lt; p') ]</td>
<td></td>
</tr>
</tbody>
</table>
We comment on these axioms: A and times, respectively. Let \( \approx \) be an overloaded equality operator applicable, pairwise to entities, spatial locations and \( t \), \( \tau \) for times. 0 designates a first, a begin time. Let \( t' \) stand for the discrete time successor of time \( t \). Let \( N(p, q) \) express that \( p \) and \( q \) are spatial neighbours.

A strict partial order, SPO, is a point structure satisfying TRANS and IRREF. TRANS, IRREF and SUCC imply infinite models. TRANS and SUCC may have finite, “looping time” models.

7.3 Wayne D. Blizard’s Theory of Space–Time

We now bring space and time together in an axiom system (Wayne D. Blizard, 1980 [45]) which relate abstracted entities to spatial points and time. Let \( A, B, \ldots \) stand for entities, \( p, q, \ldots \) for spatial points, and \( t, \tau \) for times. 0 designates a first, a begin time. Let \( t' \) stand for the discrete time successor of time \( t \). Let \( N(p, q) \) express that \( p \) and \( q \) are spatial neighbours. Let \( = \) be an overloaded equality operator applicable, pairwise to entities, spatial locations and times, respectively. \( A_p^t \) expresses that entity \( A \) is at location \( p \) at time \( t \). The axioms — where we omit (obvious) typings (of \( A, B, P, Q, \) and \( T \)): \( t' \) designates the time successor function: \( t' \).

\[
\begin{align*}
& (I) \quad \forall A \forall t \exists p : A_p^t \\
& (II) \quad (A_p^t \land A_q^t) \supset p = q \\
& (III) \quad (A_p^t \land B_q^t) \supset A = B \\
& (IV) (?) \quad (A_p^t \land A_{q}^{t'}) \supset t = t' \\
& (V i) \quad \forall p, q : N(p, q) \supset p \neq q \quad \text{Irreflexivity} \\
& (V ii) \quad \forall p, q : N(p, q) = N(q, p) \quad \text{Symmetry} \\
& (V iii) \quad \forall q, r : N(p, q) \land N(p, r) \land q \neq r \quad \text{No isolated locations} \\
& (VI i) \quad \forall t : t \neq t' \\
& (VI ii) \quad \forall t : t' \neq 0 \\
& (VI iii) \quad \forall t : t \neq 0 \land \exists \tau : t = \tau' \\
& (VI iv) \quad \forall t, \tau : \tau' = t' \land \tau = t \\
& (VII) \quad A_p^t \land A_q^{t'} \supset N(p, q) \\
& (VIII) \quad A_p^t \land B_q^{t'} \supset \sim (A_q^{t'} \land B_p^{t'})
\end{align*}
\]

We comment on these axioms:

- II–IV, VII–VIII: The axioms are universally ‘closed’: that is: We have omitted the usual \( \forall A, B, p, q, t, s \).
- (I): For every entity, \( A \), and every time, \( t \), there is a location, \( p \), at which \( A \) is located at time \( t \).
- (II): An entity cannot be in two locations at the same time.
- (III): Two distinct entities cannot be at the same location at the same time.
• (IV): Entities always move: An entity cannot be at the same location at different times. 
  This is more like a conjecture: Could be questioned.

• (V): These three axioms define $N$.

• (V i): Same as $\forall p : \sim N(p, p)$. “Being a neighbour of”, is the same as “being distinct from”.

• (V ii): If $p$ is a neighbour of $q$, then $q$ is a neighbour of $p$.

• (V iii): Every location has at least two distinct neighbours.

• (VI): The next four axioms determine the time successor function $\prime$.

• (VI i): A time is always distinct from its successor: time cannot rest. There are no time fix points.

• (VI ii): Any time successor is distinct from the begin time. Time 0 has no predecessor.

• (VI iii): Every non–begin time has an immediate predecessor.

• (VI iv): The time successor function $\prime$ is a one–to–one (i.e., a bijection) function.

• (VII): The continuous path axiom: If entity $A$ is at location $p$ at time $t$, and it is at location $q$ in the immediate next time $(t')$, then $p$ and $q$ are neighbours.

• (VIII): No “switching”: If entities $A$ and $B$ occupy neighbouring locations at time $t$ then it is not possible for $A$ and $B$ to have switched locations at the next time $(t')$.

Except for Axiom (IV) the system applies both to systems of entities that “sometimes” rests, i.e., do not move. These entities are spatial and occupy at least a point in space. If some entities “occupy more” space volume than others, then we may suitably “repair” the notion of the point space $P$ (etc.). We do not show so here.

Segment III: A Philosophy Basis

8 A Task of Philosophy

Philosophy is the study of general and fundamental problems concerning matters such as existence, knowledge, values, reason, mind, and language.

8.1 Epistemology

We shall focus on existence, specifically on epistemology – meaning ‘knowledge’ and ‘logical discourse’ – it is the branch of philosophy concerned with the theory of knowledge. Epistemology studies the nature of knowledge, justification, and the rationality of belief. Much of the debate in epistemology centers on four areas: (1) the philosophical analysis of the nature

34 Including Scientific Knowledge: Mathematics, Physics, Computer Science, etc.
of knowledge and how it relates to such concepts as truth, belief, and justification, (2) various problems of skepticism, (3) the sources and scope of knowledge and justified belief, and (4) the criteria for knowledge and justification. Epistemology addresses such questions as “What makes justified beliefs justified?”, “What does it mean to say that we know something?”, and fundamentally “How do we know that we know?”

8.2 Ontology

A “corollary” of epistemology is ontology: the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations.

8.3 The Quest

The quest is now threefold.

(i) First to prepare the ground for a discussion of possible philosophical issues of the domain analysis & description calculi. We do so by a review of philosophy (Pages 53–59) focusing on epistemology and ontology problems – from the ancient Greek philosophers till Bertrand Russell.

(ii) Then to follow that up with a review of the Philosophy of Kai Sørlander as it is, most recently, expressed in [18], and as refined from earlier works: [15, 16, 17]. This is done in Sect. 10, Pages 59–68.

(iii) Finally to show, issue-by-issue how concepts of the domain analysis & description calculi more have a basis in philosophy than in mathematics and computer science. This is done in Sect. 11, Pages 69–77.

8.4 Schools of Philosophy

We shall only cover Western Philosophy to some depth. A seven line summary will be given, in Sect. 8.4.2, of a possibly relevant aspect of Indian Philosophy. We’ll leave it at that. The fact is that Indian Philosophy has not, it appears, influenced Western Philosophy. That short summary is in line the choice of issues that we seek to uncover.

8.4.1 Western Philosophy

Section 9 presents a “capsule” summary of Western Philosophy. It is, at present, a “tour de force”, seven pages. One purpose of presenting it is that we are then able to enumerate and date the issues relevant to our quest while discarding some of the proposed theories. Another purpose is to remind the reader of the depth, breadth and plurality of issues of Western Philosophy.

8.4.2 Indian Philosophy

Pramana, literally means “proof” and “means of knowledge”, refers to epistemology in Indian philosophies. The focus of Pramana is how correct knowledge can be acquired, how one knows, how one doesn’t, and to what extent knowledge pertinent about someone or something can be acquired. Ancient and medieval Indian texts identify six pramanas as correct means of accurate knowledge and to truths: (1) perception, (2) inference, (3) comparison and analogy,
(4) postulation, (5) derivation from circumstances, non-perception, negative/cognitive proof, and (6) word, testimony of past or present reliable experts.35

9 From Ancient to Kantian Philosophy and Beyond!

The review of this section, i.e., Sect. 9, is based primarily on [15]. It is exclusively “slanted” towards those aspects of the thinking of these philosophers with respect to the task of philosophy as we defined it in Sect. 8. In this review we reject the contributions of these great philosophers that is contradictory. This presentational “bias” should in no way stand in way of our general admiration for their otherwise profound thinking.

9.1 Pre-Socrates

A number of pre-Socratic thinkers speculated on how the world was “constructed”. The earlier thinkers were pre-occupied with matter, that is, substance; what did the world consist of, how was it constructed? In doing that these thinkers were trying to be scientists, they were not, in this philosophers. We briefly review some of the pre-Socratic thinkers and philosophers.

Thales of Miletus, 624–546 BC [18, pp 35] “claimed that all existing, i.e., base matter, derived from water”; Anaximander of Miletus, 610–546 BC [18, pp 35-36] “that base matter all came from apeiron, some further unspecified substance”; Anaximenes of Miletus, 585–528 BC [18, pp 36] “that base matter was air”; Heraklit of Efesos, a. 500 BC “claimed that fire was the base matter; and extended the concern from substance to permanence and based the thinking not only on (empirical) observations but also on logical reasoning claiming that everything in the world was in a constant struggle, all the time changing – so since all is changing, i.e., that nothing is stable, he concludes that nothing exists.” In that Heraklit was a philosopher.

And, from now, philosophy reigned.

Parmenides of Elea, 501–470 BC [18, pp 37-38, 48-49] “counterclaimed that that which actually exists is eternal and unchanging – is logically impossible”; Zeno of Elea, 490–430 BC [18, pp 38-39] “supported Parmenides’ claim by claiming some paradox, i.e., the well-known Achilles and the tortoise – thereby introducing dialectic reasoning and proof by contradiction (reductio ad absurdum);” Demokrit, 460–370 BC [18, pp 40-42] “tried to unify Heraklit’s concept of changeability and Parmenides’ concept of permanence in a new way; everything in the world is built from, consists of atoms and change is due to movement of atoms”. The Sophists, 5th Century BC [18, pp 43-44] “doubted, or even refuted, that we can arrive at universal truths about the world purely through reasoning. They refute that there is an objectively true reality which we can obtain knowledge about. So, instead, skepticism reigned”.

What is interesting, to us, is that, the thinking of even the early Greek thinkers delineates the realms of religion and mythology on one side, and those of science and philosophy, on the other side.

9.2 Plato, Socrates and Aristotle

Socrates, 470–399 BC [18, pp 44–45] “protested against the sophists’ refusal of reason, common sense, sanity and prudence”. We know of Socrates’ thinking almost exclusively through Plato, 427–347 BC: [18, pp 46–49] “We shall focus on Plato’s theory of ideas. His argument is that non-physical (but substantial) ideas represent the most accurate reality. Abstract and common concepts obtain meaning through standing for ideas that are eternal and unchangeable. In contrast to ideas Plato considers the concept of a phenomenon. Phenomena are instances of ideas. We recognize a phenomenon because it embodies an idea. So, according to Plato, the changeable world that surrounds us, one which we experience through our senses, is only a reflection of a, or the, real world. That real world is unchangeable and “consists” of ideas.”

Aristotle, 384–322 BC, [18, pp 50–53] “For Aristotle it was not Plato’s abstract ideas that “existed” but the concrete world of which we are a part of with our body. The abstract ideas, however, in Aristotle’s thinking, constitute a system for describing the world. We shall very briefly list two of the concept clusters that Aristotle made to our thinking of the world: (i) modalities and (ii) explanations – the latter also referred to as causes. The modalities are: (i.1) necessity, that which is unavoidably so; (i.2) reality, that which we observe; and (i.3) possibility, that which might be. The causes (or explanations) are: (ii.1) matter or material cause, (ii.2) form cause or formal cause (ii.3) agent cause and (ii.4) end cause or purpose cause (ii.1) By material cause Aristotle means the aspect of the change or movement which is determined by the material that composes the moving or changing things. (ii.2) By form or formal cause Aristotle means a change or movement’s formal cause, is a change or movement caused by the arrangement, shape or appearance of the thing changing or moving. (ii.3) By agent cause Aristotle means a change or movement’s efficient or moving cause, consists of things apart from the thing being changed or moved, which interact so as to be an agency of the change or movement. (ii.4) By end cause or purpose cause Aristotle means a change or movement’s final cause, is that for the sake of which a thing is what it is. Aristotle’s contributions are, for us, decisive. Aristotle reveals how being is by revealing the irreducible types of predicates which we can actually use when describing the world. Aristotle thus examines the categories: substance (human, horse), quantity (6 feet tall), quality (white, red), relation (larger, shorter), location (in Athens), time (yesterday, last year), position (lying, sitting), posture (wearing shoes), action (running, singing), and suffering (being cut). This enumeration is certainly not definitive. Kant, two thousand years later, revives this idea: a system of unavoidable basic concepts for the description of the world and our situation in it.”

---

37 One may, rather crudely, interpret Plato’s concept of ideas with that of types. A value of some type is then a ‘phenomenon’.
38 It should be quite clear, to the reader, that, in this, we follow Aristotle: A main descriptonal, in fact, specificational, tool is that of type definitions.
39 Of things said without any combination, each signifies either substance or quantity or qualification or a relative or where or when or being-in-a-position or having or doing or being-affected. To give a rough idea, examples of substance are man, horse; of quantity: four-foot, five-foot; of qualification: white, grammatical; of a relative: double, half, larger; of where: in the Lyceum, in the market-place; of when: yesterday, last-year; of being-in-a-position: is-lying, is-sitting; of having: has-shoes-on, has-armour-on; of doing: cutting, burning; of being-affected: being-cut, being-burned.” Ackrill, John (1963). Aristotle, Categories and De Interpretatione. Oxford: At the Clarendon Press. ISBN 0198720866.
40 It should likewise be obvious to the reader that the notion of categories is central to our ontological structuring of domain entities.
9.3 The Stoics: 300 BC–200 AD

We shall just focus on one aspect of their contribution to logic and philosophy, that of logic. [22, pp 22-23] “They distinguish between simple propositions and composite propositions. They also distinguish between three kinds of propositions: implication, conjunction and disjunction. They had a special understanding of implication: A proposition is, to the Stoics, of the composite form: A ⇒ B; A; B. For example: If it is day then it is light; it is day; therefore it is light. In this and many other ways they contributed to the philosophy of logic (from which, it seems Gottlob Frege was inspired)”.

Chrysippus of Soli: 279–206 BC was a prominent early Stoic.

Almost two thousand years passed before philosophy again flourished. Christianity, in Europe, in a sense, “monopolised” critical thinking. With the Renaissance and Martin Luther’s Protestantism thinkers again turned to philosophy.

9.4 The Rational Tradition: Descartes,

René Descartes: 1596–1650 [18, pp 72–74] “rejected the splitting of corporeal substance into matter and form. His main focus was on the relations between mind and form: as thinking substance we recognize material substance”. Baruch Spinoza: 1632–1677 [18, pp 74-78] “rejected Descartes’s two substances: there is, he claims, is only one substance; for Spinoza God and nature was one and the same”. Gottfried Wilhelm Leibniz: 1646–1716 [18, pp 78-79] “introduced the Law of the Indiscernability of Identicals, It is still in wide use today. It states that if some object x is identical to some object y, then any property that x has, y will have as well”. 41

9.5 The Empirical Tradition: Locke, Berkeley and Hume

John Locke: 1632–1704. We focus on Locke’s ideas of sensing. He defines himself 42:

as that conscious thinking thing,
(whatever substance, made up of whether spiritual, or material, simple, or compounded, it matters not)
which is sensible, or conscious of pleasure and pain, capable of happiness or misery, and so is concerned for itself, as far as that consciousness extends.

[18, pp 80-82] “According to Locke, humans obtain their knowledge about the world through sensory perception. At one level, he claims, the world is “mechanical”, so our sensory apparatus is influenced mechanically, for example through tactile or visual means. This sense information is then communicated to our brains. First the mechanical sense data become sense ideas, The sense ideas then become reflection ideas.” In the “jargon” of our domain analysis & description method the sense ideas are values and the reflection ideas become types. So a central

41 We refer, forward, to Sect. 10.2.1 [pp. 62], and, ‘backward’, to Sect. 2.6 [pp. 21] [unique identifiers], for our “response” to Leibniz’s Law of the Indiscernability of Identicals.

idea in Locke’s theory is that all cognition builds on our reflection over sense ideas. In other words: “Can we conclude anything from our sense ideas to knowledge about those “outer” things which cause the sense ideas?” [18, pg. 85] To answer that question Locke goes on to distinguish between primary qualities and secondary qualities. In the jargon of domain analysis & description the primary qualities correspond to “our” external qualities, the secondary qualities to “our” internal qualities, but not quite! “Locke views primary qualities as measurable aspects of physical reality and secondary qualities as subjective aspects of physical reality, where “our” domain analysis & description takes both to be somehow measurable. We must therefore claim that our distinction is purely pragmatic.” Locke now claims: “(i) that we can, with respect to the primary qualities, deduce from our sense ideas to the reality, the world behind these; (ii) that the primary qualities exist in reality independent of whether we “experience” them or not; and (iii) that this is not the case for the secondary qualities which exist only in our consciousness”. George Berkeley: 1685–1753 [18, pp 82-84] “points out a problem in Locke’s theory: namely that Locke’s distinction between primary qualities as being objective and secondary qualities as being subjective does not hold. He argues that primary qualities can be subjective. To solve that problem Berkeley denied the existence of a reality “behind” the sense ideas: there is no material reality; reality is our sense ideas: esse est percipit46! The material reality is there because it is continuously experienced by ‘God’. The problem now is can we, at all, determine fundamental characteristics about the world and our situation as humans in that world without assuming the concept of independently existing substance”.

David Hume, 1711–1776. Hume’s major work was An Enquiry Concerning Human Understanding [46]. [18, pp 85-87] “Where Berkeley eliminated material substance Hume also eliminated Berkeley’s concepts of ‘God’ and ‘Consciousness’. He claimed that the basic sense-impressions, which to Hume were the basis for all valid human recognition, made it impossible to arrive at a valid recognition of ‘God’ and a substantial ‘I’. They must therefore be eliminated when trying to describe the world and our situation in it. According to Hume all that we know are sense impressions and the conceptions derived from these. Hume further distinguishes between composite and simple (not-composite) sense impressions. Correspondingly Hume distinguishes between composite and simple (non-composite) ideas. As a consequence there is no necessity in the world, nor in possible relations between cause and effect This renders Hume’s thinking in this area very problematic”.

9.6 Immanuel Kant: 1720–1804

[22, pp 280-282] “Kant was “shaken” by Hume’s critique of causality. As a response – along one line of thought – Kant introduced two notions; “Das Ding an sich” is the world that we know, that we sense, and “Das Ding für uns” is a world prior to, outside our cognition. Along another
line of thought Kant claimed that there is our cognition. By means of the cognitive tools with which our reason is equipped we reach out for “Das Ding an sich” and forms it according to our cognition. The result is the world as we know it. This means that reality never means the “Das Ding an sich”, the world “outside” us, “independent” of us. We are excluded from that world.

[18, pp 88-92] “Kant turns the reasoning around. What we empirically observe is determined by our “reasoning apparatus”. We do not observe “things” as they are in themselves (“Das Ding an sich”), but we “recognize” them as they are formed by our own reasoning apparatus. This “reasoning apparatus” includes some intuition forms: space and time. These, space and time, are therefore, to Kant, not characteristics of the world as it is, but are some intuition forms that determine our view of the world. How can it now be possible that we can have self-awareness on the basis of what we are confronted with – what we see? Here Kant introduces what he termed the transcendental deduction. We can only have self awareness under the assumption that we experience our views (outlook) as expression of objects, “things”, that exist independent of our experiencing them!”

[18, pp 90-91] “But Kant’s concept of “Das Ding an sich” is inconsistent. It is in contradiction, because it itself is knowable as being unknowable; and it is in contradiction, because it, in a mystical sense, is the cause of the thing which we know as a phenomenon, but (we) cannot apply the cause effect category outside the world of phenomena.”

A main contribution of Kant however, is his concept of Transcendental Schemata47. “If pure concepts of the understanding (categories) and sensations are radically different, what common quality allows them to relate?” Kant wrote the chapter on Schemata in his Critique of Pure Reason to solve the problem of “…how we can ensure that categories have ‘sense and significance’”. Transcendental schema are not related to empirical concepts or to mathematical concepts. These schemata connect pure concepts of the understanding, or categories, to the phenomenal appearance of objects in general, that is, objects as such, or all objects. Example categorical schemas are: The categories of quantity all share the schema of number. The categories of quality all have degrees of reality as their schema. “The schema of the category of relation is the order of time”49. “The schema of the category of modality is time itself as related to the existence of the object”50.

9.7 Post-Kant

Johann Gottlieb Fichte, 1752–1824 [18, pp 93-94] “tried to avoid Kant’s Das Ding an sich/Das Ding für uns dualism by letting the subject, the I, determine the object, the not-I, but ends up in contradiction”. Georg Wilhelm Friedrich Hegel, 1770–1831 [18, pp 94-97] “also dissolves the Kantian dualism. He builds an impressive theory. The basis for this theory is the as-

---

47 In Kantian philosophy, a transcendental schema (plural: schemata; from Greek: σχήμα, “form, shape, figure”) is the procedural rule by which a category or pure, non-empirical concept is associated with a sense impression. A private, subjective intuition is thereby discursively thought to be a representation of an external object. Transcendental schemata are supposedly produced by the imagination in relation to time https://en.wikipedia.org/wiki/Schema_(Kant)#Transcendental_schemata.


50 See footnote 49 above.
sumption of a deep-seated identity between reason (sense) and reality: “the reasonable is real” and “the real is reasonable”. Hegel saw his understanding of this duality in the light of history. Hegel thus saw truth, reason and reality historically. “Modern” dialectism was born. Now two contradictory philosophies could now be both true. From this Hegel developed an impressive “apparatus”: From “nothingness” via “creation”, “quality”, quantity” to “essence”, “cause”, “reality”, “causality”, and on to “concept”, “life” and “cognition” ending with the “absolute”! And there we end! We must reject Hegel’s thesis, antithesis, synthesis. By relativising philosophy wrt. history Hegel has removed necessity. By thus postulating that “it is an eternal truth that we cannot achieve eternal truths”. Hegel’s main contribution ends up in contradiction. Friedrich Schelling, 1775–1854, [18, pp 94] “goes further by removing the subject/object distinction claiming an underlying identity between these, that is, between mind and matter: nature is the visible mind, and mind is the invisible nature. Again this attempt brings Schelling’s work into contradictions”. Friedrich Ludwig Gottlob Frege, 1848–1925. Although primarily a mathematician and logician, Frege contributed to Philosophy. Amongst his contributions were the distinction between “sinn” (sense), and “bedeutung” (reference). The distinction is: the reference (or “referent”; bedeutung) of a proper name is the object it means or indicates (bedeuten), its sense (Sinn) is what the name expresses. The reference of a sentence is its truth value, its sense is the thought that it expresses. Edmund Husserl, 1859–1938, [18, pp 115-116] “founded a school of phenomenology. To Husserl our conscience is characterised by intentionality. Cognition is an act which is directed at something. When I see, I see something. When I think, I think something. Philosophy, to Husserl, should build on this insight. It should investigate that which conscience is directed at from “within”, and without prejudice of what it might be. Husserl expressed clearly the difference between meaning and object”. But as [15, pp 115-116] shows, Husserl thereby ends up in an inconsistent theory. Bertrand Russell, 1872–1970, [18, pp 117-118] “amongst very many contributions put forward a Philosophy of Logical Atomism [47]. It is based on the formal logic developed Russell and Whitehead in [48, Principia Mathematica]. That formal logic distinguishes between simple and complex propositions; the latter being truth functions over simple propositions. Logical Atomism now claims that the world must be describable by independent simple propositions. This requires that simple empirical propositions must be logically independent of one another. This again requires that the meaning of a simple empirical proposition alone must depend on a relation between the simple proposition and that which it stands for in reality. The meaning of a word is that “object” which the word “denotes”. This is similar to Wittgenstein’s theory. The problem is that the requirement that the simple, elementary propositions must be logically independent of one another makes it impossible to find such elementary propositions. It is therefore impossible to find those “objects” that the elementary propositions are supposed to denote. The whole of Logical Atomism thus builds on an erroneous extrapolation from formal logic”. Logical Positivism: 1920s–1936 was a “circle” if philosophers, mostly based in Vienna, cf. Wiener Kreis. [18, pp 119-121] “They did not adopt Russell’s Logical Atomism. Instead they claimed that the meaning of a sentence is its conditions for being true: i.e., a description of all facts that must be the case in order for the sentence to be judged true; that is, the verification conditions. But the problem here is that if the verification conditions are a valid meaning criterion, then its own formulation cannot be meaningful! So logical positivism ends up in

51 On Sense and Reference [“Über Sinn und Bedeutung”], Zeitschrift für Philosophie und philosophische Kritik, vol. 100 (1892), pp. 25–50
contradiction”. Some philosophers of the Vienna Circle were Moritz Schlick, 1882–1936; Alfred Jules Ayer, 1910–1989; Rudolf Carnap, 1891–1970 and Otto Neurath, 1882–1945. Ludwig Wittgenstein, 1889–1951 was not a member of the Vienna Circle, but his early work was much discussed in the Circle. [18, pp 121-124] “This work of Wittgenstein was Tractatus Logico-Philosophicus [49, 1921]. Tractatus, as did Logical Positivism, basically takes language as a departure point for a philosophical analysis of the world and our situation in it. But both these theories build on self-refusing bases. Wittgenstein understood that his Tractatus was built on a too simple meaning theory, i.e., a theory of how meaning is ascribed to sentences. In Philosophische Untersuchungen [50] Wittgenstein explores new directions – which have no bearing on our quest.”

9.8 Bertrand Russell – Again!

We bring an excerpt from Russell’s [51, History of Western Philosophy, Chap. XXXI: The Philosophy of Logical Analysis, pp 786–788]. The excerpt that we bring reflects Russell’s thinking, around 1945, as influenced, no doubt, by developments in quantum physics. From all this it seems to follow that events, not particles, must be the ‘stuff’ of physics. What has been thought of as a particle will have to be thought of as a series of events. The series of events that replaces a particle has certain important physical properties, and therefore demands our attention; but it has no more substantiality than any other series of events that we might arbitrarily single out. Thus ‘matter’ is not part of the ultimate material of the world, but merely a convenient way of collecting events into bundles.”

We cannot, but point out, the “similarity” of these observations to our transcendental deduction of behaviours from parts.

We have surveyed ideas of 32 philosophers – ideas relevant to our quest: that of understanding borderlines between philosophical arguments and formal, mathematical arguments as they relate to domain analysis & description. We shall now turn to elucidate these.

10 The Kai Sørlander Philosophy

We shall review an essence of [15, 18]. Kai Sørlander’s objective [18, pp 131] “is to investigate the philosophical question: ‘what are the necessary characteristics of each and every possible world and our situation in it’. We can reformulate this question into the task of determining the necessary logical conditions for every possible description of the world and our situation in it”.

10.1 The Basis

In this section we shall mostly quote from [15]. “The world is all that is the case. All that can be described in true propositions.” “In science we investigate how the world is factually.” “Philosophy puts forward another question. We ask of what could not consistently be otherwise.” 52:1.2.3 The Inescapable Meaning Assignment: “It is thus the task of philos-
o phy to determine the inescapable characteristics of the world and our situation in it.” In determining these inescapable characteristic “we cannot refer to our experience ... since the experience cannot tell us anything that could not consistently be otherwise.” “Two demands must be satisfied by the philosophical basis. The first is that it must not be based on empirical premises. The other is that it cannot consistently be refuted by anybody under any conceivable circumstances. These demands can only be satisfied by one assumption.” We shall refer to this assumption as:

**The Inescapable Meaning Assignment**

- The *The Inescapable Meaning Assignment* is\(^53\) the recognition of the mutual dependency between
  - the meaning of designations and
  - the consistency relations between propositions.

As an example of what “goes into” the inescapable meaning assignment we bring, albeit from the world of computer science, that of the description of the stack data type (its entities and operations).

**Stacks - A Narrative**

82 Stacks, \(s:S\), have elements, \(e:E\);
83 the *empty* \(S\) operation takes no arguments and yields a result stack;
84 the *is_empty* \(S\) operation takes an argument stack and yields a Boolean value result.
85 the *stack* operation takes two arguments: an element and a stack and yields a result stack.
86 the *unstack* operation takes an non-empty argument stack and yields a stack result.
87 the *top* operation takes an non-empty argument stack and yields an element result.

**The consistency relations:**

88 an *empty* \(S\) stack is *empty*, and a stack with at least one element is not;
89 unstacking an argument stack, *stack*\((e,s)\), results in the stack \(s\); and
90 inquiring as to the top of a non-empty argument stack, *stack*\((e,s)\), yields \(e\).

**The meaning of designations:**

\(^{53}\)[15], pg. 13-14, ℓ13-ℓ1
Necessary and Empirical Propositions: “That the inescapable meaning assignment is required in order to answer the question of how the world must necessarily be can be seen from the following.” “It makes it possible to distinguish between necessary and empirical propositions.” “A proposition is necessary if its truth value depends only on the meaning of the designators by means of which it is expressed.” “A proposition is empirical if its truth value does not so depend.” “An empirical proposition must therefore refer to something ... which exists independently of its designators, and it must predicate something about the thing to which it refers.” The definition “the world is all that is the case. All that can be described in true propositions,” 54 satisfies the inescapable meaning assignment. “That which is described in necessary propositions is that which is common to [all] possible worlds. A concrete world is all that can be described in true empirical propositions.” 55

Primary Objects: “an empirical proposition must refer to an independently existing thing and must predicate something about that thing. On that basis it is then possible to deduce how those objects that can be directly referred to in simple empirical propositions must necessarily be. Those things are referred to as primary objects. A deduction of the inevitable characteristics of a possible world is thus identical to a deduction of how primary objects must necessarily be.” 56

Two Requirements to the Philosophical Basis: “Two demands have been put to the philosophical basis for our quest. It must not contain empirical preconditions; and the foundation must not consistently be refuted. It must not consistently be false.” 57

The inescapable meaning assignment: ‘the meaning of designations and the consistency relations between propositions’ 58 ... satisfies this basis. 59

The Logical Connectives: Sørlander now deduces the logical connectives: 60
conjunction (‘and’ ∧), disjunction (‘or’, ∨), and implication (⇒ or ⊃). Necessity and Possibility: [18, pp 142] “A proposition is necessarily true, if its truth follows from the definition of of the designations by means of which it is expressed; then it must be true under all circumstances. A proposition is possibly true, if its negation is not necessarily true”. Empirical Propositions: An empirical proposition refers to an independently existing entities and predicates something that can be either true or false about the referenced entity. The entities that are referenced in empirical propositions have not been completely characterised by these propositions; they are simply those that can be referenced in empirical propositions.

10.2 Logical Conditions for Describing Physical Worlds

So which are the logical conditions of descriptions of any world? In [15] and [18] Kai Sørlander, through a series of transcendental deductions “unravels” the following logical conditions: (i) symmetry and asymmetry (ii) transitivity and intransitivity, (iii) space: direction, distance, etc., (iv) time: before, after, in-between etc., (v) states and causality, (vi) kinematics, dynamics, etc., and (vii) Newton’s laws, et cetera. We shall summarise Sørlander’s deductions.

To remind the reader: the issue is that of deducing how the primary entities must necessarily be.

10.2.1 Symmetry and Asymmetry

[18, pp 152] “There can be different primary entities. Entity A is different from entity B if A can be ascribed a predicate in-commensurable with a predicate ascribed to B. ‘Different from’ is a symmetric predicate. If entity A is identical to entity B then A cannot be ascribed a predicate which is in-commensurable with any predicate that can be ascribed to B; and then B is identical to A. ‘Equal to’ is a symmetric predicate”.

10.2.2 Transitivity and Intransitivity

[18, pp 148] “If A is identical to B and B is identical to C then A is identical to C with identity then being a transitive relation. The relation different from is not transitive it is an intransitive relation”.

10.2.3 Space

[18, pp 154] “The two relations asymmetric and symmetric, by a transcendental deduction, can be given an interpretation: The relation (spatial) direction is asymmetric; and the relation (spatial) distance is symmetric. Direction and distance can be understood as spatial relations. From these relations are derived the relation in-between. Hence we must conclude that primary entities exist in space. Space is therefore an unavoidable characteristic of any possible world”. From the direction and distance relations one can derive Euclidean Geometry.

10.2.4 States

[18, pp 158-159] “We must assume that primary entities may be ascribed predicates which are not logically required. That is, they may be ascribed predicates incompatible with predicates which they actually satisfy. For it to be logically possible, that one-and-the-same primary entity can
be ascribed incompatible predicates, is only logically possible if any primary entity can exist in different states. A primary entity may be in one state where it can be ascribed one predicate, and in another state where it can be ascribed another incompatible predicate”.

10.2.5 Time

[18, pp159] “Two such different states must necessarily be ascribed different incompatible predicates. But how can we ensure so? Only if states stand in an asymmetric relation to one another. This state relation is also transitive. So that is an indispensable property of any world. By a transcendental deduction we say that primary entities exist in time. So every possible world must exist in time”.

10.2.6 Causality

[18, pp.162-163] “States are related by the time relations “before” and “after”. These are asymmetric and transitive relations. But how can it be so? Propositions about primary entities at different times must necessarily be logically independent of one another. This follows from the possibility that a primary entity necessarily be ascribed different, incompatible predicates at different times. It is therefore logically impossible from the primary entities alone to deduce how a primary entity is at one point in time to how it is at another point in time. How, therefore, can these predicates supposedly of one and the same entity at different time points be about the same entity? There can be no logical implication about this! Transcendentally therefore there must be a non-logical implicative between propositions about properties of a primary entity at different times. Such an non-logical implicative must depend on empirical circumstances subject to which the primary entity exists. There are no other circumstances. If the state on a primary entity changes then there must be changes in its “circumstances” whose consequences are that the primary entity changes state. And such “circumstance”–changes will imply primary entity state changes. We shall use the term ‘cause’ for a preceding ”circumstance”–change that implies a state change of a primary entity. So now we can conclude that every change of state of a primary entity must have a cause, and that “equivalent circumstances” must have “equivalent effects”. This form of implication is called causal implication. And the principle of implication for causal principle. So every possible world enjoys the causal principle. Kant’s transcendental deduction is fundamentally built on the the possibility of self-awareness. Sørlander’s transcendental deduction is fundamentally built on the possibility of truth. In Kant’s thinking the causal principle is a prerequisite for possibility of self-awareness”. In this way Sørlander avoids Kant’s solipsism, i.e., “that only one’s own mind is sure to exist” a solipsism that, however, flaws Kant’s otherwise great thinking.

10.2.7 Kinematics

[18, pp.164–165] “So primary entities exist in space and time. They must have spatial extent and temporal extent. They must therefore be able to change their spatial properties. Both as concerns form and location. But a spatial change in form presupposes a change in location – as the more fundamental. A primary entity which changes location is said to be in movement. If a primary entity which does not change location is said to be at rest. The velocity of a
primary entity expresses the distance and direction it moves in a given time interval. Change in velocity of a primary entity is called its acceleration. Acceleration involves either change in velocity, or change in direction of movement, or both.” So far we have reasoned us to fundamental concepts of kinematics.

10.2.8 Dynamics

[18, pp.165-166] “When we ”add” causality” to kinematics we obtain dynamics. We can do so, because primary entities are in time. Kinematics imply that that a primary entity changes when it goes from being at rest to be moving. Likewise when it goes from movement to rest. And similarly, when it accelerates (decelerates). So a primary entity has same state of movement if it has same velocity and moves in the same direction. Primary entities change state of movement if they change velocity or direction. So, combining kinematics and the principle of causality, we can deduce that if a primary entity changes state of movement then there must be a cause, and we call that cause a force”.

10.2.9 Newton’s Laws

Newton’s First Law: [18, pp.165-166] “Combining kinematics and the principle of causality, and the therefrom deduced concept of force, we can deduce that any change of movement is proportional to the force. This implies that a primary entity which is not under the influence of an external force will continue in the same state of movement – that is, be at rest or conduction a linear movement at constant velocity. This is Newton’s First Law”. Newton’s Second Law: [18, pp.166] “That a certain, non-zero force implies change of movement, imply that the primary entity must exert a certain resistance to that change. It must have what we shall call a certain mass. From this it follows that the change in the state of movement of a primary entity not only is proportional to the exerted force, but also inversely proportional to the mass of that entity. This is Newton’s Second Law”. Newton’s Third Law: [18, pp.166-167] “In a possible world, the forces that affects primary entities must come from “other” primary entities. Primary entities are located in different volumes of space. Their location may interfere with one another in the sense at least of “obstructing” their mutual movements – leading to clashes. In principle we must assume that even primary entities “far away from one another” obstruct. If they clash it must be with oppositely directed and equal forces. This is Newton’s Third Law”.

---

61 Velocity has a speed and a vectorial direction. Speed is a scalar, for example of type kilometers per hour. Vectorial direction is a scalar structure, for example for a spatial direction consisting of geographical elements: x degrees North, y degrees East (x + y = 90), and z degrees Up or Down (0 ≤ z ≤ 90, where, if z = 90 we have that both x and y are 0).

62 Observe that we have “only” said: proportional, meaning also directly proportional, not whether it is logarithmically, or linearly, or polynomially, or exponentially, etc., so.

63 Mass refers loosely to the amount of matter in an entity. This is in contrast to weight which refers to the force exerted on an entity by gravity.

64 Cf. Footnote 62.
10.3 Gravitation and Quantum Mechanics

Mutual Attraction: [18, pp 167-168] “How can primary entities possibly be the source of forces that influence one another? How can primary entities at all have a mass\(^6\) such that it requires forces to change their state of movement? The answer must be that primary entities exert a mutual influence on one another – that is there is a mutual attraction”

Gravitation: [18, pp 168] “This must be the case for all primary entities. This must mean that all primary entities can be characterised by a universal mutual attraction: a universal gravitation”

Finite Propagation – A Gravitational Constant: [18, pp 168] “Thus mutual attraction must propagate at a certain, finite, velocity. If that velocity was infinite, then it is everywhere and cannot therefore have its source in concretely existing primary entities. But having a finite velocity implies that there must be a propagational speed limit. It must be a constant of nature.”

Gravitational “Pull”: [18, pp 169-170] “The nature of gravitational “pull” can be deduced, basically as follows: Primary entities must basically consist of elements that attract one another, but which are stable, and that is only possible if it is, in principle, impossible to describe these elementary particles precisely. If there is a fundamental limit to how these basic particles can be described, then it is also precluded that they can undergo continuous change. Hence there is a basis for stability despite mutual attraction. There must be a foundational limit for how precise these descriptions can be, which implies that the elementary particle as a whole can be described statistically”

Quantum Mechanics: The rest is physics: unification of quantum mechanics and Einstein’s special relativity has been done; unification of gravitation with Einstein’s general theory of relativity is still to be done. A Summary: [18, pp 170-173] “Philosophy lends to physics its results a necessity that physics cannot give them. Experiments have shown that Einstein’s results – with propagation limits – indeed hold for this world. Philosophy shows that every possible world is subject to a fixed propagation limit. Philosophy also shows that for a possible world to exist it must be built from elementary particles which cannot be individually described (with Newton’s theory)”

10.4 The Logical Conditions for Describing Living Species

10.4.1 Purpose, Life and Evolution

Causality of Purpose: [18, pp 174] “If there is to be the possibility of language and meaning then there must exist primary entities which are not entirely encapsulated within the physical conditions; that they are stable and can influence one another. This is only possible if such primary entities are subject to a supplementary causality directed at the future: a causality of purpose”

Living Species: [18, pp 174-175] “These primary entities are here called living species. What can be deduced about them? They must have some form they can be developed to reach; and which they must be causally determined to maintain. This development and maintenance must further in an exchange of matter with an environment. ... It must be possible that living species occur in one of two forms: one form which is characterised by development, form

\(^6\) Let two entities have respective masses \(m_1\) and \(m_2\). Let the forces with which they attract each other be \(f_1\), respectively \(f_2\). Then the law of gravitation – as it can be deduced by philosophical arguments – can be expressed as \(f_1 = f_2\). The specific force, expressed using Newton’s constant \(G\) is \(f = G \times m_1 \times m_2 \times r^{-2}\) where \(r\) is the distance between the two entities and \(G = 6.674 \times 10^{-11} \times m^3 \times kg^{-1} \times s^{-2}\) [m: meter, kg: kilogram s: second] – as derived by physicists.
and exchange, and another form which, additionally, can be characterised by the ability to purposeful movement. The first we call plants, the second we call animals.”

**Animate Entities:** [18, pp 176] “For an animal to purposefully move around there must be “additional conditions” for such self-movements to be in accordance with the principle of causality: they must have sensory organs sensing among others the immediate purpose of its movement; they must have means of motion so that it can move; and they must have instincts, incentives and feelings as causal conditions that what it senses can drive it to movements” And all of this in accordance with the laws of physics. **Animal Structure:** [18, pp 177-178] “Animals, to possess these three kinds of “additional conditions”, must be built from special units which have an inner relation to their function as a whole: their purposefulness must be built into their physical building units; that is, as we can now say, their genomes; that is, animals are built from genomes which give them the inner determination to such building blocks for instincts, incentives and feelings. Similar kinds of deduction can be carried out with respect to plants. Transcendently one can deduce basic principles of evolution but not its details”

**10.4.2 Consciousness, Learning and Language**

**Consciousness and Learning:** [18, pp 180-181] “The existence of animals is a necessary condition for there being language and meaning in any world. That there can be language means that animals are capable of developing language. And this must presuppose that animals can learn from their experience. To learn implies that animals can feel pleasure and distaste and can learn... One can therefore deduce that animals must possess such building blocks whose inner determination is a basis for learning and consciousness” **Language:** [18, pp 181-182] “Animals with higher social interaction uses signs, eventually developing a language. These languages adhere to the same system of defined concepts which are a prerequisite for any description of any world: namely the system that philosophy lays bare from a basis of transcendental deductions and the principle of contradiction and its implicit meaning theory”

**10.5 Humans, Knowledge, Responsibility**

**Humans:** [18, pp 184] “A human is an animal which has a language” **Knowledge:** [18, pp 184] “Humans must be conscious of having knowledge of its concrete situation, and as such that humans can have knowledge about what they feel, and eventually that humans can know whether what they feel is true or false. Consequently humans can describe their situation correctly” **Responsibility:** [18, pp 184] “In this way one can deduce that humans can thus have memory and hence can have responsibility, be responsible. Further deductions lead us into ethics”

**10.6 An Augmented Upper Ontology**

We now augment our upper-ontology, to include living species, from that of Fig. 1 Pg. 15 to that of Fig. 6 Pg. 67. We leave it to the reader to “fill in the details!”

**10.7 Artifacts: Man-made Entities**
Figure 6: An Upper Ontology for Domains – with Living Species

**Definition 27 Artifact:** By an artifact we shall understand a man-made entity: usually an endurant in space, one that satisfies the laws of physics, and sometimes one that, by a transcendental deduction, can take on the role of a perdurant; but the artifact can also, for example, by intended as a piece of art, something for our enjoyment and reflection.

We then augment our upper-ontology, to include artifacts, from that of Fig. 6 Pg. 67 to that of Fig. 7 Pg. 68. We leave it to the reader to “fill in the details!”

### 10.8 Intentionality

We have ended our presentation of Sørlander’s Philosophy. Before going into justifications of our domain analysis & description calculi with respect to this philosophy we shall briefly comment on the concept of intentionality.

Intentionality is a philosophical concept and is defined by the Stanford Encyclopedia of Philosophy as “the power of minds to be about, to represent, or to stand for, things, properties and states of affairs.” The puzzles of intentionality lie at the interface between the philosophy of mind and the philosophy of language. The word itself, which is of medieval Scholastic origin, was rehabilitated by the philosopher Franz Brentano towards the end of the nineteenth century, and adopted by Edmund Husserl. ‘Intentionality’ is a philosopher’s word. It derives

---

from the Latin word *intentio*, which in turn derives from the verb *intendere*, which means being directed towards some goal or thing. The earliest theory of intentionality is associated with St. Anselm’s ontological argument for the existence of God, and with his tenets distinguishing between objects that exist in the understanding and objects that exist in reality.

We shall here endow the concept of ‘intentionality’ with the following interpretation. Man-made artifacts are made for specific purposes. Often two or more artifacts are intended to serve a purpose, that is, to represent an intent. We speculate as follows:

**Definition 28 On Intentional Pull:** Two or more artifactual parts of different sorts, but with overlapping sets of intents may exert an intentional “pull” on one another.

This intentional “pull” may take many forms. Let $p_x : X$ and $p_y : Y$ be two parts of different sorts $(X, Y)$, and with common intent, $i$. Manifestations of these, their common intent must somehow be subject to constraints, and these must be expressed predicatively.

We return, in Sect. 11.1.4 [pp. 73], with an example of what we claim to be an intentional “pull”, that is, Example 34 [pp. 73].
Segment IV: Fusing Philosophy into Computer Science

11 Philosophical Issues of The Domain Calculi

We now interpret the domain analysis & description analysis calculus of Segment I in the light of Sørlander’s Philosophy of Sect. 10.

We re-examine all analysis calculus prompts with references to their prompt number or the section – and the page on which their definition is given.

11.1 The Analysis Calculus Prompts

11.1.1 External Qualities

• Item 1, pp. 12: is_universe_of_discourse: After a rough sketch narrative of the contemplated domain, the informal justification to be given for this query should be along these lines: the chosen universe-of-discourse is one that can be described in true propositions; that is, one that is based in space and time; subject to Laws of Newton; etc., and, indispensably so, involves persons with language, responsibility and intents.

• Item 2, pp. 13: is_entity: So entities are just that: describable, based in either space (as are endurants) or in both space and time (as are perdurants), and involving persons. That is, entities are the “stuff” that philosophy cares about in its quest to understand the world. What lies outside may be in the realm of superstition, “mumbo-jumbo”, etcetera; “things” that are neither in space nor time; figments of the mind.

• Item 3, pp. 13: is_endurant: An endurant is an entity which we characterise in propositions without reference to (actual, i.e. “real”) time. There is no notion of state changes in describing entities. Endurants are either based in physics or based in living species: plants and animals including persons, or are artifacts which build on endurants. Endurants are, in the words of Whitehead, [52], continuants.

• Item 4, pp. 13: is_perdurant: And, consequently, a perdurant is an entity which we characterise in propositions with more-or-less explicit reference to (actual, i.e. “real”) time, focusing on state-changes and/or interaction between perdurants. Perdurants are either actions or events or behaviours. Definition: Behaviours are defined as sets of sequences of actions, events and behaviours. Philosophical treatments are given of the notions of time in [53, 35, 45, 34], [discrete] actions in [33], events in [54, 55, 56, 57, 58, 59, 60, 61, 62, 63], and behaviours in, for example, the Internet based articles on plato.stanford.edu/entries/behaviorism/ and www.behavior.org/search.php?q=behavior+and+philosophy. Most of the literature on behaviours focus on psychological aspects which we consider outside the realm of our form of domain analysis & description.

The interplay between endurants and perdurants is studied in [64].

• Item 5, pp. 14: is_discrete: [We re-emphasize that the notion of discreteness of endurants such as we “need” it here, is not related to the notion of discreteness in physics or mathematics.] The terms separate, individual and distinct characterise discreteness.
It is up to the domain analysis & description scientist cum engineer to decide whether an entity should be characterised as primarily distinguished by these ‘qualities’ – or not.

- Item 6, pp. 14: **is_continuous**: [We re-emphasize that the notion of continuity of endurants such as we “need” it here, is not related to the notion of continuity in physics or mathematics.] The terms: prolonged, without interruption, and unbroken series or pattern characterise continuity of endurants. It is up to the domain analysis & description scientist cum engineer to decide whether an entity should be characterised as primarily distinguished by these ‘qualities’, or not.

- Item 7, pp. 15: **is_structure**: Whether a discrete endurant is considered a structure, or a part, or a set of components is a pragmatic decision. So has no bearings in the Sørlander Philosophy outside its possible bearings in language where the notion of language can be motivated philosophically.

- Item 8, pp. 16: **is_part**, Item 14, pp. 19: **is_component** and Item 16, pp. 20: **is_material**: All entities, whether non-living species, including artificial, or living species (plants and animals, incl. humans) are subject to the inescapable meaning assignment, the principle of contradiction and its implicit meaning theory. They are also subject to the notions of space and time and to the Laws of Newton, etc. The living species entities are additionally subject to causality of purpose with humans having language, memory and responsibility. These notions can be assumed, but we do not, at present, i.e., in this report, suggest any means of modelling language, memory and responsibility. Following Sørlander’s Philosophy there are the (atomic, see below) part p living species: **is_LIVE_SPECIES(p)**, of which there are plants, **is_PLANT(p)**, and there are animals, **is_ANIMAL(p)**, of which (latter) some are humans, **is_HUMAN(p)**, and some are not; and there are the non-living-species parts, p, of which some are made by man (or by other artifacts), **is_ARTIFACT(p)**, and some are not, we refer to them as physical parts. We therefore now, as a consequence of Sørlander’s Philosophy, suggest the domain analysis prompts: **is_LIVE_SPECIES, is_PLANT, is_ANIMAL, is_HUMAN** and **is_ARTIFACT**.

All this means that the Sørlander Philosophy, in a sense, mandates us to introduce the following new analysis prompts:

**Analysis Prompt 28 is_physical**: The domain analyser analyses discrete endurants (d) into physical parts:

- **is_physical** – where is_physical(d) holds if d is a physical part

**Analysis Prompt 29 is_living**: The domain analyser analyses discrete endurants (d) into living species:

- **is_living** – where is_living(d) holds if θ is a living species

**Analysis Prompt 30 is_natural**: The domain analyser analyses physical parts (p) into natural:
Analysis Prompt 31 is_artifactual: The domain analyser analyses physical parts (p) into artifactual physical parts:

- is_artifactual – where is_artifactual(p) holds if p is a man-made part

Analysis Prompt 32 is_plant: The domain analyser analyses living species (ℓ) into plants:

- is_plant – where is_plant(ℓ) holds if ℓ is a plant

Analysis Prompt 33 is_animal: The domain analyser analyses living species (ℓ) into animals:

- is_animal – where is_animal(ℓ) holds if ℓ is an animal

Analysis Prompt 34 is_human: The domain analyser analyses animals (α) into humans:

- is_human – where is_human(α) holds if α is a human

Analysis prompts, is XXX, similar to is_human, can be devised for other animal species.

- Item 9, pp. 16: is_atomic: and Item 10, pp. 16: is_composite: The notion of atomicity here has nothing to do with that of the the Greeks [Demokrit, pp. 53]. Here it is a rather pragmatic issue, void, it seems, of philosophical challenge. It is a purely pragmatic issue with respect to any chose domain whether the domain scientist cum engineer decides to analyse & describe a part into being atomic or composite.

Example 31 Automobile: Atomic or Composite: Thus, for example, you the reader may consider your automobile as atomic, whereas your mechanic undoubtedly considers it composite.

11.1.2 Unique Identifiers

Sect. 2.6, pp. 21–22: unique identifiers:

Uniqueness of entities follows from the basic logic of symmetry etc. Uniqueness or rather identity, is an thus important philosophical notion [cf. Sect. 10.2.1 [pp. 62]]. Notice that we are not concerned with any representation of unique part and component identifiers. So please, dear reader, do not speculate on that! The uniqueness of part or component identifiers “follows” the part and component, irrespective of the spatial location and time of the possibly “movable” part or component, i.e., irrespective of its state!
11.1.3 Mereology

Sect. 2.7, pp. 22–23: mereology:

There are some new aspects of the concept of mereology – which, in light of the Sørlander Philosophy, were not considered in Sect. 2.7, and which it is now high time to consider, and, for some of these aspects, to include in the domain analysis & description method.


- Topologies and Intents: To us mereology, in light of Sørlander’s Philosophy, now becomes either of two relations (or possibly both): (i) spatial relations, as for Stanisław Leśniewski and the cited references, and (ii) intensional relations. We characterise the latter as follows:

**Definition 29 Intentional Relations**: By an intensional relation we shall understand a relation between distinct endurants which manifests two (or more) designations and at least one meaning.

**Example 32 Transport**: Automobiles and roads, i.e. hubs and links, have distinct sorts and designations, but share the intent (meaning) of technologically supporting traffic.

We refer to [5, Domain Facets: Analysis & Description].

- Part Mereologies: Thus the mereology of parts shall be sought in either their topological, i.e., spatial, arrangements, or their intents – with parts of same intent being mereologically related, or possibly some combination of both.

**Example 33 Traffic**: Hence, in reference to the example of Sect. 6, we have that the mereologies of each automobile include the set of unique identifiers of all hubs and links, and the mereologies of each hub and link include the set of unique identifiers of all automobiles.

- Further Studies: It appears that the concept of mereology, in light of Sørlander’s Philosophy, warrants further scrutiny, philosophically well as from the point of view of domain analysis & description method. Should discrete endurants be further analysed into structures, parts and components, as now, and natural discrete endurants or artifact discrete endurants or should discrete endurants have attribute values of natural discrete endurant values or artifact discrete endurant values.

---

https://en.wikipedia.org/wiki/Mereology#Metaphysics
11.1.4 Attributes

Sect. 2.8, pp. 23–27: attributes:

Attributes, their type and value, are the main means for expressing propositions about primary entities. Let us first recall: parts and components have unique identifiers, parts have mereologies and parts and materials have attributes. Let us also “remember” that these differences are purely pragmatic. All endurants are subject to being in space and time, and being subject to the principle of causality. Three sets of attributes follow from the Sorlander’s Philosophy: (i) attributes of non-life-specifies entities; (ii) attributes of life-specifies entities, but additionally subject to purpose, language, responsibility, and causality of principle; and those (iii) attributes that are additional and more individually determined by the kind of the part. We shall now summarise these.

Non-Species Parts: These are the parts that were actually treated in Sect. 2. To them, as a consequence of Sorlander’s Philosophy, one can ascribe the following attribute observers: attr_SPACE and attr_TIME. No explanation seems necessary here. Attribute observers related to the above could be: attr_LOCATION where the location to be yielded is some spatial point within the space yielded by the SPACE observer. attr_VOLUME where the volume is the volume (in some units) of the space yielded by the SPACE observer. attr_MASS(p) where the mass is the mass (in some units) of the part p. Et cetera. We leave it to the reader to “think up” Boolean and other algebraic operators over time, space, location, mass, etc.

Artifacts: To remind, artifacts are parts made by man and/or other artifacts. They have all the same attributes (i.e. attribute observers) as has non-species parts. In addition they may have such attribute observes as attr_Intent, attr_Maker, attr_Brand_Name, attr_Production_Year, attr_Owner, attr_Purchase_Price, attr_Current_Value and attr_Condition. The idea of the attr_Intent attribute observer is to yield a token that somehow identifies the purpose of the artifact: transport, "measurement-of-this", "measurement-of-that", "food-stuff", etc. We leave it to the reader to figure out the idea of the other attributes.

Artifactual Intents: In the world of physics, since Isaac Newton, the mutual attraction of bodies (with mass) and in the context of gravitation leads to the gravitational pull, cf. Sect. 10.3 pp. 65. Now, in the context of artifactual parts with intents we may speak of intentional "pull".

Definition 30 Intentional Pull: Two or more artifactual parts of different sorts, but with overlapping sets of intents may exert an intentional "pull" on one another.

This intentional “pull” may take many forms. Let px : X and py : Y be two parts of different sorts (X, Y), and with common intent, i. Manifestations of these, their common intent must somehow be subject to constraints, and these must be expressed predicatively.

Example 34 Automobile and Road Transport: For the main example, Sect. 6,

automobiles shall now include the intent of ‘transport’,

and so shall hubs and links.

attr_Intent: A → {'transport'|...}-set

69 The world is all that is the case. All that can be described in true propositions. [15, pp.14, l 2-3]
Manifestations of ‘transport’ is reflected in automobiles having the automobile position attribute, APos, Item 56 Pg. 40, hubs having the hub traffic attribute, H_Traffic, Item 48 Pg. 39, and in links having the link traffic attribute, L_Traffic, Item 52 Pg. 39.

93 Seen from the point of view of an automobile there is its own traffic history, A_Hist Item 56c Pg. 40, which is a (time ordered) sequence of timed automobile’s positions;

94 seen from the point of view of a hub there is its own traffic history, H_Traffic Item 48 Pg. 39, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions; and

95 seen from the point of view of a link there is its own traffic history, L_Traffic Item 52 Pg. 39, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions.

The intentional “pull” of these manifestations is this:

96 The union, i.e. proper merge of all automobile traffic histories, AllATH, must now be identical to the same proper merge of all hub, AllHTH, and all link traffic histories, AllLTH.

type

56c, pp.40 A_Hist = (T × APos)*
48, pp.39 H_Traffic = A_UI → (T × APos)*
52, pp.39 L_Traffic = A_UI → (T × APos)*
96 AllATH = T → (AUI → APos)
96 AllHTH = T → (AUI → APos)
96 AllLTH = T → (AUI → APos)

axiom

96 let allA = proper_merge_into_AllATH({(a,attr_A_Hist(a))|a:A•a ∈ as}),
96 allH = proper_merge_into_AllHTH({attr_H_Traffic(h)|h:H•h ∈ hs}),
96 allL = proper_merge_into_AllLTH({attr_L_Traffic(l)|l:L•h ∈ ls}) in
96 allA = H_and_L_Traffic_merge(allH,allL) end

We leave the definition of the four merge functions to the reader!

We now discuss the concept of intentional “pull”. We endow each automobile with its history of timed positions and each hub and link with their histories of timed automobile positions. These histories are facts! They are not something that is laboriously recorded, where such recordings may be imprecise or cumbersome. The facts are there, so we can (but may not necessarily) talk about these histories as facts. It is in that sense that the purpose (‘transport’) for which man let automobiles, hubs and link be made with their ‘transport’ intent are subject to an intentional “pull”. It can be no other way: if automobiles “record” their history, then hubs and links must together “record” identically the same history! ■
We have tentatively proposed a concept of \textit{intentional “pull”}. That proposal is in the form, I think, of a transcendental deduction; it has to be further studied.

\textbf{Humans}\textsuperscript{71}: \textit{Humans} have \textit{sensory organs} and \textit{means of motion}; \textit{inner determination} for \textit{instincts, incentives} and \textit{feelings}; \textit{purpose}; and \textit{language}; and can \textit{learn}\textsuperscript{72}. We leave it, to the reader, as a \textit{research topic}: to suggest means for expressing analysis prompts that cover these kinds of attributes.

For this report we have little to say on the issue of \textit{humans}. Rather much more work has to be done for any meaningful writing. So, here is a challenge to the readers!

\subsection{11.1.5 A Summary of Domain Analysis Prompts}

1. \texttt{is\_universe\_of\_discourse}, 12
2. \texttt{has\_component}, 16
3. \texttt{is\_endurant}, 13
4. \texttt{is\_component}, 19
5. \texttt{is\_material}, 20
6. \texttt{type\_name}, 21
7. \texttt{has\_mereology}, 22
8. \texttt{attribute\_types}, 24
9. \texttt{is\_entity}, 13
10. \texttt{is\_composite}, 16
11. \texttt{observe\_endurants}, 17
12. \texttt{has\_components}, 19
13. \texttt{is\_component}, 19
14. \texttt{has\_materials}, 20
15. \texttt{is\_material}, 20
16. \texttt{type\_name}, 21
17. \texttt{has\_mereology}, 22
18. \texttt{attribute\_types}, 24
19. \texttt{is\_entity}, 13
20. \texttt{is\_static\_attribute}, 25
21. \texttt{is\_dynamic\_attribute}, 25
22. \texttt{is\_inert\_attribute}, 26
23. \texttt{is\_reactive\_attribute}, 26
24. \texttt{is\_active\_attribute}, 26
25. \texttt{is\_autonomous\_attribute}, 26
26. \texttt{is\_biddable\_attribute}, 26
27. \texttt{is\_programmable\_attribute}, 26
28. \texttt{is\_physical}, 70
29. \texttt{is\_living}, 70
30. \texttt{is\_natural}, 71
31. \texttt{is\_artifactual}, 71
32. \texttt{is\_plant}, 71
33. \texttt{is\_animal}, 71
34. \texttt{is\_human}, 71
35. \texttt{is\_perdurant}, 13
36. \texttt{is\_discrete}, 14
37. \texttt{is\_continuous}, 14
38. \texttt{is\_structure}, 15
39. \texttt{is\_part}, 16
40. \texttt{is\_atomic}, 16
41. \texttt{has\_concrete\_type}, 17

\section{11.2 The Description Calculus Prompts}

\begin{itemize}
\item Item 1, pp. 12: \texttt{observe\_universe\_of\_discourse}:
\item Item 2, pp. 17: \texttt{observe\_endurant\_sorts}:
\item Item 3, pp. 18: \texttt{observe\_part\_type}:
\item Item 4, pp. 19: \texttt{observe\_component\_sorts}:
\item Item 5, pp. 20: \texttt{observe\_material\_sorts}:
\item Item 6, pp. 21: \texttt{observe\_unique\_identifier}:
\end{itemize}

\textsuperscript{70}or thought technologically in-feasible – at least some decades ago!

\textsuperscript{71}We focus on humans, but the discussion can be “repeated”, in modified form, for plants and animals in general.

\textsuperscript{72}cf. Sect. \textbf{10.4.2} [pp. 66]
• Item 7, pp. 22: observe_mereology:
• Item 8, pp. 24: observe_attributes:

11.2.1 A Summary of Domain Description Prompts

[1] observe_universe_of_discourse, 12
[2] observe_endurant_sorts, 17
[3] observe_part_type, 18
[6] observe_unique_identifier, 21
[7] observe_mereology, 22
[8] observe_attributes, 24

11.3 The Behaviour Schemata

11.4 Wrapping Up

We summarise the above in a revision of the ontology diagram first given in Fig. 1 Pg. 15 and used, in more-or-less that form, in several publications: [1, 4, 7, 88]. The revision is shown in Fig. 8:

Figure 8 emphasises the analytic, “upper” structure of domains and emphasises endurants: Black names attached to diagram nodes designate “upper” categories of entities. Red names similarly attached designate manifest categories of entities. Blue names also so attached are the sort names of values of manifest endurants. Both naturals and artifacts have atomic and composite values. We only hint (¨...) at other (than human) animal species. The lower dashed horizontal lines with pairs of -o-o- hint at the internal endurant qualities that are “transferred”

11.5 Discussion

11.5.1 Review of Revisions

We have related a number of the domain analysis & description method’s analysis prompts to Sørlander’s Philosophy – and have found that a number of corrections has to be made to the understanding of these: the basis for unique identifiers and the categories of endurants and attributes. With [1] endurants came in three forms: structures, parts (atomic and composite), and materials. Now we must refine the notion of parts into: physical parts (as assumed in [1]), artifactual parts and living species parts. We must further articulate the notion of attributes: as before, for physical parts, to necessarily include the in-avoidable classical physics attributes\(^73\) and be subject to the principle of causality and gravitational pull; but now additionally also to artifactual parts, still subject to the attributes of physical parts but now additionally subject to

\(^73\)space, time, mass, velocity, etc.
additional in-avoidable attributes such as intent and to both gravitational pull and intentional “pull”; and to living species parts, notably, in this report, humans with their attributes.

11.5.2 General

It is only of interest to study the domain analysis & description method analysis calculus with respect to Sørlander’s Philosophy. The corresponding description calculus and schemata are not analytic. They represent our “response” to the domain analysis. So our “quest” has ended. It is time to “sum up”.

Segment V: Summing Up

Although there is obviously a lot more to study we stop here, for a while, to wrap up this report. With what we have presented we can, however, make several conclusions – and that will now be done!

12 Conclusion

12.1 General Remarks

When I have informed my colleagues of this work their reactions have been mixed. Oh yes, philosophy, yes, I referred to Plato in one of my papers, ages ago!, or – does it relate to
the recent Facebook scandal?, and other such deeply committing and understanding utterings. Philosophy is actually hard. Anyone can claim to reflect philosophically, and many do, and some even refer, in their newspaper columns, to being philosophers, but it does take some practice to actually do philosophy. Good schooling, up to senior high, is required. Having learned to reason, in classical disciplines like mathematics and physics; being able to read in two or more foreign languages; having learned history, real history, for us, in the Western world, from before the ancient Greeks, and onwards; these seems to be prerequisites for a serious study of philosophy.

In grammar school I passed the little test in Greek and the “large” test in Latin at the age of 14–15. I had wonderful teachers. I learned about the history of ideas from Johs. Sleek [23]. My university did not offer courses in philosophy. Over the years I acquired many [and browsed some additional] philosophy books: Karl Jaspers [89], Bertrand Russell [90, 91, 51], [Alfred North Whitehead [92, 52, 93],] Willard van Orme Quine [94, 95, 96], [Martin Heidegger [53],] Ludwig Johan Josef Wittgenstein [97, 50], Karl Popper [98, 99, 100, 101, 102, 103], Imre Lakatos [104], David Favrholdt [105, 106], John Sowa [107], as well as some dictionaries: [30, 29, 108, 31, Cambridge, Oxford, Blackwell] and [109]. In this century I started looking at a number of epistemological essays: [110, Logic and Ontology], [77, 78, 82, 111, 112, Objects], [79, 80, 81, 113, 85, Ontology], [114, 33, 57, Actions], [54, 55, 59, 115, 61, 63, 62, 58, 57, Events], [66, 67, 74, 75, 71, 86, 87, 83, 62, 26, Mereology], [116, 117, 118, 119, Qualities, Properties] and [56, SpaceTime]. But although wonderful “reads”, it was not until Sørlander’s [15, 16, 2, 17, 19, 20, 3, 18] that philosophy really started meaning something. ‘Philosophy is useless’ it is said. ‘ “Results” of philosophy are not meant to solve problems ’, it is said. But Sørlander’s Philosophy, [15, 18], have definitely helped shape the domain analysis & description analysis calculus into a form that makes it rather definitive!

Before my study of Kai Sørlander’s Philosophy the upper ontology – like shown in Fig. 1 Pg. 15 – was based on empirical observations.

After my study the upper ontology – now shown in Fig. 7 Pg. 68 – is based on philosophical reasoning and is definite, is unavoidable!

12.2 Revisions to the Calculi and Further Studies

Yes, our study of Sørlander’s Philosophy, [15, 18], has led to the following modifications of the domain analysis & description analysis calculus: (i) a more refined view of discrete endurants; (ii) “refinements” of attributes need be studied further; (iii) the intentional "pull" between artificial parts need be studied further; and (iv) the transcendental deduction that “translates” endurants into behaviours need be studied further see, however, below.

   (i) Refined View of Discrete Endurants: Where discrete endurants before were (i.1) parts and (i.2) components, they are now (i.1a) physical, (i.2) components, (i.3) live species parts and (i.1b) artifacts. of which the live species parts are (i.3a) plants and (i.3b) animals, (i.3c) for which latter we focus on humans.

   (iv) Which Endurants are Candidates for Perdurancy? (iv.1) Naturals: It seems that if we only focus on transcendentially deducing natural endurants into behaviours then we are really studying or doing physics: mechanics, chemistry, electricity, et cetera. (iv.2) Living Species: It seems that if we only focus on transcendentially deducing (iv.2.1) living species into behaviours then we are really studying or doing life sciences: botanics, zoology, biology, et
cetera. (iv.2.2) or if we just focus on humans, then we are really studying or doing behavioral sciences. (iv.3) **Artifacts:** (iv.3.1) We have seen that it makes sense to “transmogrify” many artifacts into behaviours. But how characterise those for which that deduction makes, or does not make sense? (iv.3.2) It seems that if we only focus on transcendentally deducing artifacts into behaviours then we are really studying or doing engineering: mechanical, chemical, electrical, electronics, et cetera, engineering.

### 12.3 Remarks on Classes of Artifactual Perdurants

We can rather immediately identify the following “classes” of artifactual perdurants:

- **Computerised Command & Control Systems:** Here we have several, i.e. more than just a few distinct artifacts, interacting with human operators for the purpose of command, monitoring and controlling some of these artifacts and humans. Examples are pipelines [120] and swarms of drones [121].
- **Logistics: Planning & Monitoring:** Here again we have several, i.e. more than just a few distinct artifacts, but the emphasis is on operational planning and the monitoring of plan fulfillment. Examples are container lines [122] and railways [123, 124, 125, 126, 127].
- **Monitoring:** Usually the systems here are just monitoring a single endurant. Examples are weather forecast [128] and health care.
- **Mechanics:** Here we are dealing with the operation of just one artifact: a lathe a machine saw, etc., an automobile, et cetera.
- **The “End” Result:** Here we are dealing with computers being the artifacts – “final” instruments in achieving some purpose! Examples are urban planning [129] stock exchange [130] credit card system [131] documents [132] Web systems [133] E-market [134]

We refer to [14] for a discussion of domain models as a basis for software demos, software simulators, software monitoring and software monitoring and control.

### 12.4 Acknowledgements

First and foremost I acknowledge the deep inspiration drawn from the study of Sørlander’s Philosophy, notably [2] and [3]. Several people have commented, in various more-or-less spurious ways, not knowing really, what I was up to, when I informed them of my current study and writing on “applying” Sørlander’s Philosophy, notably [2] and [3] to my work on domain analysis & description. Several of these comments, however uncommitted, have, however – strangely enough, upon reflection, helped me to even better grasp what it was I was trying to unravel. Let my acknowledgments to them remain anonymous.

### 13 Bibliography

#### 13.1 Bibliographical Notes

We list a number of reports all of which document descriptions of domains. These descriptions were carried out in order to research and develop the domain analysis and description concepts
now summarised in the present paper. These reports ought now be revised, some slightly, others less so, so as to follow all of the prescriptions of the current paper. Except where a URL is given in full, please prefix the web reference with: http://www2.compute.dtu.dk/~dibj/.


[61] Roberto Casati and Achille C. Varzi, editors. *Events*. Ashgate Publishing Group – Dartmouth Publishing Co. Ltd., Wey Court East, Union Road, Farnham, Surrey, GU9 7PT, United Kingdom, 23 March 1996.


[65]. Technical report.


holey: something full of holes


Segment VI: Appendix

A  RSL: The RAISE Specification Language – A Primer

A.1  Type Expressions

Type expressions are expressions whose value are types, that is, possibly infinite sets of values (of “that” type).

A.1.1  Atomic Types

Atomic types have (atomic) values. That is, values which we consider to have no proper constituent (sub-)values, i.e., cannot, to us, be meaningfully “taken apart”.

RSL has a number of built-in atomic types. There are the Booleans, integers, natural numbers, reals, characters, and texts.

\[
\begin{align*}
\text{type} & \quad [1] \text{Bool true, false} \\
& \quad [2] \text{Int } \ldots, -2, -2, 0, 1, 2, \ldots \\
& \quad [3] \text{Nat } 0, 1, 2, \ldots \\
& \quad [4] \text{Real } \ldots, -5.43, -1.0, 0.0, 1.23 \ldots, 2.7182 \ldots, 3.1415 \ldots, 4.56, \ldots \\
& \quad [5] \text{Char } "a", "b", \ldots, "0", \ldots \\
& \quad [6] \text{Text } "abracadabra"
\end{align*}
\]

A.1.2  Composite Types

Composite types have composite values. That is, values which we consider to have proper constituent (sub-)values, i.e., can be meaningfully “taken apart”. There are two ways of expressing composite types: either explicitly, using concrete type expressions, or implicitly, using sorts (i.e., abstract types) and observer functions.

Concrete Composite Types  From these one can form type expressions: finite sets, infinite sets, Cartesian products, lists, maps, etc.

Let A, B and C be any type names or type expressions, then the following are type expressions:

\[
\begin{align*}
[7] \text{A-set} & \quad [13] \text{A \rightarrow B} \\
[8] \text{A-infset} & \quad [14] \text{A \leadsto B} \\
[9] \text{A \times B \times \ldots \times C} & \quad [15] (A) \\
[10] \text{A^*} & \quad [16] \text{A | B | \ldots | C} \\
[11] \text{A^\omega} & \quad [17] \text{mk_id(sel_a:A,\ldots,sel_b:B)} \\
[12] \text{A \rightarrowrightarrow B} & \quad [18] \text{sel_a:A ... sel_b:B}
\end{align*}
\]

The following the meaning of the atomic and the composite type expressions:

1  The Boolean type of truth values \text{false} and \text{true}.
2 The integer type on integers ..., –2, –1, 0, 1, 2, ...

3 The natural number type of positive integer values 0, 1, 2, ...

4 The real number type of real values, i.e., values whose numerals can be written as an integer, followed by a period ("."), followed by a natural number (the fraction).

5 The character type of character values "a", "bb", ...

6 The text type of character string values "aa", "aaa", ..., "abc", ...

7 The set type of finite cardinality set values.

8 The set type of infinite and finite cardinality set values.

9 The Cartesian type of Cartesian values.

10 The list type of finite length list values.

11 The list type of infinite and finite length list values.

12 The map type of finite definition set map values.

13 The function type of total function values.

14 The function type of partial function values.

15 In (A) A is constrained to be:

- either a Cartesian $B \times C \times \ldots \times D$, in which case it is identical to type expression kind 9,
- or not to be the name of a built-in type (cf., 1–6) or of a type, in which case the parentheses serve as simple delimiters, e.g., $(A \rightarrow B)$, or $(A^*)$-set, or $(A$-set$)$list, or $(A|B) \rightarrow (C|D|(E \rightarrow F))$, etc.

16 The postulated disjoint union of types $A$, $B$, ..., and $C$.

17 The record type of $mk_{id}$-named record values $mk_{id}(av,\ldots,bv)$, where $av$, $\ldots$, $bv$, are values of respective types. The distinct identifiers $sel_a$, etc., designate selector functions.

18 The record type of unnamed record values $(av,\ldots,bv)$, where $av$, $\ldots$, $bv$, are values of respective types. The distinct identifiers $sel_a$, etc., designate selector functions.

**Sorts and Observer Functions**

```plaintext
type
  A, B, C, ..., D
value
  obs_B: A → B, obs_C: A → C, ..., obs_D: A → D
```
The above expresses that values of type A are composed from at least three values — and these are of type B, C, . . . , and D. A concrete type definition corresponding to the above presupposing material of the next section

type
    B, C, ..., D
    A = B × C × ... × D

A.2 Type Definitions
A.2.1 Concrete Types

Types can be concrete in which case the structure of the type is specified by type expressions:

type
    A = Type_expr

Some schematic type definitions are:

[19] Type_name = Type_expr /* without | s or subtypes */
[20] Type_name = Type_expr_1 | Type_expr_2 | ... | Type_expr_n
[21] Type_name ==
    mk_id_1(s_a1:Type_name_a1,...,s_ai:Type_name_ai) |
        ...
        mk_id_n(s_z1:Type_name_z1,...,s_zk:Type_name_zk)
[22] Type_name :: sel_a:Type_name_a ... sel_z:Type_name_z
[23] Type_name = { | v:Type_name' • P(v) |}

where a form of [20]–[21] is provided by combining the types:

Type_name = A | B | ... | Z
A == mk_id_1(s_a1:A_1,...,s_ai:A_i)
B == mk_id_2(s_b1:B_1,...,s_bj:B_j)
...
Z == mk_id_n(s_z1:Z_1,...,s_zk:Z_k)

Types A, B, ..., Z are disjoint, i.e., shares no values, provided all \( \text{mk}_i \) are distinct and due to the use of the disjoint record type constructor ==.

axiom
\[
\forall a_1:A_1, a_2:A_2, ..., a_i:A_i \cdot \\
    s_{a_1}(\text{mk}_i(a_1,a_2,...,ai)) = a_1 \land s_{a_2}(\text{mk}_i(a_1,a_2,...,ai)) = a_2 \land \\
    ... \land s_{ai}(\text{mk}_i(a_1,a_2,...,ai)) = a_i \land \\
\forall a:A \cdot \text{let } \text{mk}_i(a'_1,a'_2,...,ai') = a \text{ in } \\
a_1' = s_{a_1}(a) \land a_2' = s_{a_2}(a) \land ... \land a_i' = s_{ai}(a) \text{ end}
\]
A.2.2 Subtypes

In RSL, each type represents a set of values. Such a set can be delimited by means of predicates. The set of values b which have type B and which satisfy the predicate \( \mathcal{P} \), constitute the subtype A:

\[
\text{type } A = \{ | b : B \cdot \mathcal{P}(b) \}
\]

A.2.3 Sorts — Abstract Types

Types can be (abstract) sorts in which case their structure is not specified:

\[
\text{type } A, B, ..., C
\]

A.3 The RSL Predicate Calculus

A.4 Propositional Expressions

Let identifiers (or propositional expressions) a, b, ..., c designate Boolean values (true or false [or chaos]). Then:

\[
\text{false, true }
\]
\[
a, b, ..., c \sim a, a \land b, a \lor b, a \Rightarrow b, a = b, a \neq b
\]

are propositional expressions having Boolean values. \( \sim, \land, \lor, \Rightarrow, = \) and \( \neq \) are Boolean connectives (i.e., operators). They can be read as: not, and, or, if then (or implies), equal and not equal.

A.4.1 Simple Predicate Expressions

Let identifiers (or propositional expressions) a, b, ..., c designate Boolean values, let x, y, ..., z (or term expressions) designate non-Boolean values and let i, j, ..., k designate number values, then:

\[
\text{false, true }
\]
\[
a, b, ..., c
\]
\[
\sim a, a \land b, a \lor b, a \Rightarrow b, a = b, a \neq b
\]
\[
x = y, x \neq y,
\]
\[
i < j, i \leq j, i \geq j, i \neq j
\]

are simple predicate expressions.
### A.4.2 Quantified Expressions

Let $X, Y, \ldots, C$ be type names or type expressions, and let $\mathcal{P}(x)$, $\mathcal{Q}(y)$ and $\mathcal{R}(z)$ designate predicate expressions in which $x, y$ and $z$ are free. Then:

\[
\forall x:X \cdot \mathcal{P}(x) \\
\exists y:Y \cdot \mathcal{Q}(y) \\
\exists ! z:Z \cdot \mathcal{R}(z)
\]

are quantified expressions — also being predicate expressions.

They are “read” as: For all $x$ (values in type $X$) the predicate $\mathcal{P}(x)$ holds; there exists (at least) one $y$ (value in type $Y$) such that the predicate $\mathcal{Q}(y)$ holds; and there exists a unique $z$ (value in type $Z$) such that the predicate $\mathcal{R}(z)$ holds.

### A.5 Concrete RSL Types: Values and Operations

#### A.5.1 Arithmetic

**type**

Nat, Int, Real

**value**

$+, -, *, \div: \text{Nat} \times \text{Nat} \rightarrow \text{Nat} | \text{Int} \times \text{Int} \rightarrow \text{Int} | \text{Real} \times \text{Real} \rightarrow \text{Real}$

$\leq, =, \neq, \geq, >: (\text{Nat}|\text{Int}|\text{Real}) \rightarrow (\text{Nat}|\text{Int}|\text{Real})$

#### A.5.2 Set Expressions

**Set Enumerations** Let the below $a$’s denote values of type $A$, then the below designate simple set enumerations:

\[
\{\}, \{a\}, \{e_1,e_2,\ldots,e_n\}, \ldots \in A\text{-set} \\
\{\}, \{a\}, \{e_1,e_2,\ldots,e_n\}, \ldots, \{e_1,e_2,\ldots\} \in A\text{-infset}
\]

**Set Comprehension** The expression, last line below, to the right of the $\equiv$, expresses set comprehension. The expression “builds” the set of values satisfying the given predicate. It is abstract in the sense that it does not do so by following a concrete algorithm.

**type**

A, B

$P = A \rightarrow \text{Bool}$

$Q = A \Rightarrow B$

**value**

comprehend: $A\text{-infset} \times P \times Q \rightarrow B\text{-infset}$

comprehend(s,P,Q) $\equiv \{ Q(a) | a:A \bullet a \in s \land P(a) \}$
A.5.3 Cartesian Expressions

Cartesian Enumerations Let $e$ range over values of Cartesian types involving $A$, $B$, $C$, then the below expressions are simple Cartesian enumerations:

type
- $A$, $B$, ..., $C$
- $A \times B \times ... \times C$

value
- $(e_1,e_2,...,e_n)$

A.5.4 List Expressions

List Enumerations Let $a$ range over values of type $A$, then the below expressions are simple list enumerations:

\[
\{ (), (e), ..., (e_1,e_2,...,e_n), ... \} \in A^* \\
\{ (), (e), ..., (e_1,e_2,...,e_n), ..., (e_1,e_2,...,e_n, \}, ... \} \in A^\omega
\]

\[
( a_{\omega_1} .. a_{\omega_2} )
\]

The last line above assumes $a_i$ and $a_j$ to be integer-valued expressions. It then expresses the set of integers from the value of $e_i$ to and including the value of $e_j$. If the latter is smaller than the former, then the list is empty.

List Comprehension The last line below expresses list comprehension.

\[
\text{type} \\
- A, B, P = A \rightarrow \text{Bool}, Q = A \leadsto B
\]

\[
\text{value} \\
- \text{comprehend}: A^\omega \times P \times Q \leadsto B^\omega
\]

\[
\text{comprehend}(l,P,Q) \equiv \langle Q(l(i)) \mid i \in \langle 1..\text{len } l \rangle \bullet P(l(i)) \rangle
\]

A.5.5 Map Expressions

Map Enumerations Let (possibly indexed) $u$ and $v$ range over values of type $T1$ and $T2$, respectively, then the below expressions are simple map enumerations:

\[
\text{type} \\
- T1, T2
\]

\[
M = T1 \rightarrow T2
\]

\[
\text{value} \\
- \text{u,u1,u2,...,un:T1, v,v1,v2,...,vn:T2} \\
- \text{[]} , \text{[u\rightarrow v]}, ..., \text{[u1\rightarrow v1,u2\rightarrow v2,...,un\rightarrow vn]} \text{ all } M
\]
Map Comprehension  The last line below expresses map comprehension:

\[
\begin{align*}
\text{type} & \quad U, V, X, Y \\
M & \equiv U \to V \\
F & \equiv U \leadsto X \\
G & \equiv V \leadsto Y \\
P & \equiv U \to \text{Bool} \\
\text{value} & \quad \text{comprehend}: M \times F \times G \times P \to (X \leadsto Y) \\
\text{comprehend}(m,F,G,P) & \equiv \{ F(u) \mapsto G(m(u)) \mid u:U \cdot u \in \text{dom } m \land P(u) \}
\end{align*}
\]

A.5.6  Set Operations

Set Operator Signatures

\[
\begin{align*}
\& 19 \in: A \times A\text{-infset} \to \text{Bool} \\
\& 20 \notin: A \times A\text{-infset} \to \text{Bool} \\
\& 21 \cup: A\text{-infset} \times A\text{-infset} \to A\text{-infset} \\
\& 22 \cup: (A\text{-infset})\text{-infset} \to A\text{-infset} \\
\& 23 \cap: A\text{-infset} \times A\text{-infset} \to A\text{-infset} \\
\& 24 \cap: (A\text{-infset})\text{-infset} \to A\text{-infset} \\
\& 25 \setminus: A\text{-infset} \times A\text{-infset} \to A\text{-infset} \\
\& 26 \subset: A\text{-infset} \times A\text{-infset} \to \text{Bool} \\
\& 27 \subseteq: A\text{-infset} \times A\text{-infset} \to \text{Bool} \\
\& 28 =: A\text{-infset} \times A\text{-infset} \to \text{Bool} \\
\& 29 \neq: A\text{-infset} \times A\text{-infset} \to \text{Bool} \\
\& 30 \text{card}: A\text{-infset} \leadsto \text{Nat}
\end{align*}
\]

Set Examples

\[
\begin{align*}
\text{examples} & \quad a \in \{a,b,c\} \\
a \notin \emptyset, a \notin \{b,c\} \\
\{a,b,c\} \cup \{a,b,d,e\} = \{a,b,c,d,e\} \\
\cup\{\{a\},\{a,bb\},\{a,d\}\} = \{a,b,d\} \\
\{a,b,c\} \cap \{c,d,e\} = \{c\} \\
\cap\{\{a\},\{a,bb\},\{a,d\}\} = \{a\} \\
\{a,b,c\} \setminus \{c,d\} = \{a,bb\} \\
\{a,bb\} \subset \{a,b,c\} \\
\{a,b,c\} \subseteq \{a,b,c\} \\
\{a,b,c\} = \{a,b,c\} \\
\{a,b,c\} \neq \{a,bb\} \\
\text{card } \emptyset = 0, \text{card } \{a,b,c\} = 3
\end{align*}
\]
Informal Explication

19 \(\in\): The membership operator expresses that an element is a member of a set.
20 \(\not\in\): The nonmembership operator expresses that an element is not a member of a set.
21 \(\cup\): The infix union operator. When applied to two sets, the operator gives the set whose members are in either or both of the two operand sets.
22 \(\cup\): The distributed prefix union operator. When applied to a set of sets, the operator gives the set whose members are in some of the operand sets.
23 \(\cap\): The infix intersection operator. When applied to two sets, the operator gives the set whose members are in both of the two operand sets.
24 \(\cap\): The prefix distributed intersection operator. When applied to a set of sets, the operator gives the set whose members are in some of the operand sets.
25 \(\setminus\): The set complement (or set subtraction) operator. When applied to two sets, the operator gives the set whose members are those of the left operand set which are not in the right operand set.
26 \(\subseteq\): The proper subset operator expresses that all members of the left operand set are also in the right operand set.
27 \(\subset\): The proper subset operator expresses that all members of the left operand set are also in the right operand set, and that the two sets are not identical.
28 \(=\): The equal operator expresses that the two operand sets are identical.
29 \(\neq\): The nonequal operator expresses that the two operand sets are not identical.
30 \(\text{card}\): The cardinality operator gives the number of elements in a finite set.

Set Operator Definitions  The operations can be defined as follows (\(\equiv\) is the definition symbol):

\[
\begin{align*}
\text{value} & \equiv \{ a \mid a:A \land a \in s' \lor a \in s'' \} \\
\text{s'} \cap \text{s''} & \equiv \{ a \mid a:A \land a \in s' \land a \in s'' \} \\
\text{s'} \setminus \text{s''} & \equiv \{ a \mid a:A \land a \in s' \land a \not\in s'' \} \\
\text{s'} \subseteq \text{s''} & \equiv \forall a:A \rightarrow a \in s'' \\
\text{s'} \subset \text{s''} & \equiv \text{s'} \subseteq \text{s''} \land \exists a:A \rightarrow a \in s'' \land a \not\in s' \\
\text{s'} = \text{s''} & \equiv \forall a:A \land a \in s' \equiv a \in s'' \equiv s'' \subseteq s' \land s' \subseteq s \\
\text{s'} \neq \text{s''} & \equiv \text{s'} \cap \text{s''} \neq \{\} \\
\text{card} s & \equiv \\
& \text{if } s = \{\} \text{ then 0 else} \\
& \text{let } a:A \land a \in s \text{ in } 1 + \text{card} (s \setminus \{a\}) \text{ end end} \\
& \text{pre } s \text{ is a finite set } */ \\
& \text{card} s \equiv \text{chaos } /* \text{ tests for infinity of } s */
\end{align*}
\]
A.5.7 Cartesian Operations

type
A, B, C

\[ g_0: G_0 = A \times B \times C \]
\[ g_1: G_1 = (A \times B) \times C \]
\[ g_2: G_2 = A \times (B \times C) \]
\[ g_3: G_3 = A \times (B \times C) \]

value
va:A, vb:B, vc:C, vd:D

\((va,vb,vc):G_0, (va,vb,vc):G_1, ((va,vb),vc):G_2, (va3,(vb3,vc3)):G_3\)

decomposition expressions

let \((a_1,b_1,c_1) = g_0, (a_1',b_1',c_1') = g_1\) in .. end

let \((a_2,b_2,c_2) = g_2\) in .. end

let \((a_3,(b_3,c_3)) = g_3\) in .. end

A.5.8 List Operations

List Operator Signatures

value
hd: A^\omega \xrightarrow{\sim} A

tl: A^\omega \xrightarrow{\sim} A^\omega

len: A^\omega \xrightarrow{\sim} Nat

inds: A^\omega \rightarrow Nat-infset

elems: A^\omega \rightarrow A-infset

\(\langle\cdot\rangle\) : A^\omega \times Nat \xrightarrow{\sim} A

\(\hat{}\) : A^* \xrightarrow{\sim} A^* \xrightarrow{\sim} A^* \xrightarrow{\sim} BoBbol

List Operation Examples

examples

hd\langle a_1,a_2,...,a_m\rangle = a_1

tl\langle a_1,a_2,...,a_m\rangle = \langle a_2,...,a_m\rangle

len\langle a_1,a_2,...,a_m\rangle = m

inds\langle a_1,a_2,...,a_m\rangle = \{1,2,...,m\}

elems\langle a_1,a_2,...,a_m\rangle = \{a_1,a_2,...,a_m\}

\langle a_1,a_2,...,a_m\rangle(i) = a_i

\langle a,b,c\rangle \sim\langle a,b,d\rangle = \langle a,b,c,a,b,d\rangle

\langle a,b,c\rangle = \langle a,b,c\rangle

\langle a,b,c\rangle \neq \langle a,b,d\rangle

Informal Explication

- **hd**: Head gives the first element in a nonempty list.
- **tl**: Tail gives the remaining list of a nonempty list when Head is removed.
- **len**: Length gives the number of elements in a finite list.
• **inds**: Indices give the set of indices from 1 to the length of a nonempty list. For empty lists, this set is the empty set as well.

• **elems**: Elements gives the possibly infinite set of all distinct elements in a list.

• \( \ell(i) \): Indexing with a natural number, \( i \) larger than 0, into a list \( \ell \) having a number of elements larger than or equal to \( i \), gives the \( i \)th element of the list.

• \( \hat{} \): Concatenates two operand lists into one. The elements of the left operand list are followed by the elements of the right. The order with respect to each list is maintained.

• \( = \): The equal operator expresses that the two operand lists are identical.

• \( \neq \): The nonequal operator expresses that the two operand lists are *not* identical.

The operations can also be defined as follows:

**List Operator Definitions**

\[ \text{value} \]

is\text{\_finite\_list}: A^\omega \rightarrow \text{Bool}

\[
\text{len } q \equiv \\
\text{case } \text{is\_finite\_list}(q) \text{ of} \\
\quad \text{true } \rightarrow \text{if } q = \langle \rangle \text{ then } 0 \text{ else } 1 + \text{len } t \text{ q end,} \\
\quad \text{false } \rightarrow \text{chaos end}
\]

inds \( q \) \( \equiv \)

\[
\text{case } \text{is\_finite\_list}(q) \text{ of} \\
\quad \text{true } \rightarrow \{ i \mid i: \text{Nat} \cdot 1 \leq i \leq \text{len } q \}, \\
\quad \text{false } \rightarrow \{ i \mid i: \text{Nat} \cdot i \neq 0 \} \text{ end}
\]

elems \( q \) \( \equiv \{ q(i) \mid i: \text{Nat} \cdot i \in \text{inds } q \} \)

\[
q(i) \equiv \\
\text{if } i = 1 \\
\quad \text{then} \\
\quad \quad \text{if } q \neq \langle \rangle \\
\quad \quad \quad \text{then let } a: A, q': Q \cdot q = \langle a \rangle \hat{q}' \text{ in a end} \\
\quad \quad \quad \text{else chaos end} \\
\quad \quad \text{else } q(i-1) \text{ end}
\]

\[
fq \hat{iq} \equiv \\
\langle \text{ if } 1 \leq i \leq \text{len } fq \text{ then } fq(i) \text{ else } iq(i - \text{len } fq) \text{ end} \\
\mid i: \text{Nat} \cdot \text{if } iq \neq \text{chaos} \text{ then } i \leq \text{len } fq + \text{len end} \rangle
\]

\[ \text{pre } \text{is\_finite\_list}(fq) \]
\[ iq' = iq'' \equiv 
\text{inds } iq' = \text{inds } iq'' \land \forall i: \text{Nat} \cdot i \in \text{inds } iq' \Rightarrow iq'(i) = iq''(i) \]

\[ iq' \neq iq'' \equiv \neg (iq' = iq'') \]

### A.5.9 Map Operations

**Map Operator Signatures and Map Operation Examples**

**value**

\[ m(a): M \rightarrow A \rightarrow B, m(a) = b \]

**dom:** \( M \rightarrow A \rightarrow \text{infset} \) [domain of map]

\[ \text{dom} [a_1 \mapsto b_1, a_2 \mapsto b_2, \ldots, a_n \mapsto b_n] = \{a_1, a_2, \ldots, a_n\} \]

**rng:** \( M \rightarrow B \rightarrow \text{infset} \) [range of map]

\[ \text{rng} [a_1 \mapsto b_1, a_2 \mapsto b_2, \ldots, a_n \mapsto b_n] = \{b_1, b_2, \ldots, b_n\} \]

**†:** \( M \times M \rightarrow M \) [override extension]

\[ [a \mapsto b, a' \mapsto b', a'' \mapsto b''] \uparrow [a' \mapsto b', a'' \mapsto b''] = [a \mapsto b, a' \mapsto b', a'' \mapsto b'', a''' \mapsto b'''] \]

**∪:** \( M \times M \rightarrow M \) [merge ∪]

\[ [a \mapsto b, a' \mapsto b', a'' \mapsto b''] \cup [a''' \mapsto b'''] = [a \mapsto b, a' \mapsto b', a'' \mapsto b'', a''' \mapsto b'''] \]

**\:\** \( M \times A \rightarrow M \) [restriction by]

\[ [a \mapsto b, a' \mapsto b', a'' \mapsto b''] \backslash \{a\} = [a' \mapsto b', a'' \mapsto b''] \]

**/:** \( M \times A \rightarrow M \) [restriction to]

\[ [a \mapsto b, a' \mapsto b', a'' \mapsto b''] / \{a', a''\} = [a' \mapsto b', a'' \mapsto b''] \]

**=,≠:** \( M \times M \rightarrow \text{Bool} \)

\[ (A \mapsto B) \times (B \mapsto C) \rightarrow (A \mapsto C) \) [composition]

\[ [a \mapsto b, a' \mapsto b'] \circ [b \mapsto c, b' \mapsto c'] = [a \mapsto c, a' \mapsto c'] \]

### Map Operation Explication

- **\( m(a) \):** Application gives the element that \( a \) maps to in the map \( m \).
- **\( \text{dom} \):** Domain/Definition Set gives the set of values which maps to in a map.
- **\( \text{rng} \):** Range/Image Set gives the set of values which are mapped to in a map.
- **\( \uparrow \):** Override/Extend. When applied to two operand maps, it gives the map which is like an override of the left operand map by all or some “pairings” of the right operand map.
• ∪: Merge. When applied to two operand maps, it gives a merge of these maps.

• \: Restriction. When applied to two operand maps, it gives the map which is a restriction of the left operand map to the elements that are not in the right operand set.

• /: Restriction. When applied to two operand maps, it gives the map which is a restriction of the left operand map to the elements of the right operand set.

• =: The equal operator expresses that the two operand maps are identical.

• ≠: The nonequal operator expresses that the two operand maps are not identical.

• ◦: Composition. When applied to two operand maps, it gives the map from definition set elements of the left operand map, \( m_1 \), to the range elements of the right operand map, \( m_2 \), such that if \( a \) is in the definition set of \( m_1 \) and maps into \( b \), and if \( b \) is in the definition set of \( m_2 \) and maps into \( c \), then \( a \), in the composition, maps into \( c \).

Map Operation Redefinitions  The map operations can also be defined as follows:

value
\[ \text{rng} \ m \equiv \{ \ m(a) \mid a:A \cdot a \in \text{dom} \ m \} \]

\[ m_1 \uplus m_2 \equiv [ a \mapsto b \mid a:A,b:B \cdot a \in \text{dom} \ m_1 \setminus \text{dom} \ m_2 \land bb=m_1(a) \lor a \in \text{dom} \ m_2 \land bb=m_2(a) ] \]

\[ m_1 \cup m_2 \equiv [ a \mapsto b \mid a:A,b:B \cdot a \in \text{dom} \ m_1 \land bb=m_1(a) \lor a \in \text{dom} \ m_2 \land bb=m_2(a) ] \]

\[ m \setminus s \equiv [ a \mapsto m(a) \mid a:A \cdot a \in \text{dom} \ m \setminus s ] \]

\[ m / s \equiv [ a \mapsto m(a) \mid a:A \cdot a \in \text{dom} \ m \cap s ] \]

\[ m_1 = m_2 \equiv \text{dom} \ m_1 = \text{dom} \ m_2 \land \forall a:A \cdot a \in \text{dom} \ m_1 \Rightarrow m_1(a) = m_2(a) \]

\[ m_1 \neq m_2 \equiv \neg(m_1 = m_2) \]

\[ m \circ n \equiv [ a \mapsto c \mid a:A,c:C \cdot a \in \text{dom} \ m \land c = n(m(a)) ] \]

\[ \text{pre rng} \ m \subseteq \text{dom} \ n \]

A.6  \( \lambda \)-Calculus + Functions

A.6.1  The \( \lambda \)-Calculus Syntax

type */ A BNF Syntax: */
\[
\langle L \rangle ::= \langle V \rangle \mid \langle F \rangle \mid \langle A \rangle \mid ( \langle A \rangle )
\]
\[ \langle V \rangle ::= */ variables, i.e. identifiers */ \]
\[ \langle F \rangle ::= \lambda \langle V \rangle \mathord{\cdot} \langle L \rangle \]
\[ \langle A \rangle ::= ( \langle L \rangle \langle L \rangle ) \]

value /* Examples */
\[ \langle L \rangle: e, f, a, \ldots \]
\[ \langle V \rangle: x, \ldots \]
\[ \langle F \rangle: \lambda x \mathord{\cdot} e, \ldots \]
\[ \langle A \rangle: f a, (f a), f(a), (f)(a), \ldots \]

A.6.2 Free and Bound Variables

Let \( x, y \) be variable names and \( e, f \) be \( \lambda \)-expressions.

- \( \langle V \rangle \): Variable \( x \) is free in \( x \).
- \( \langle F \rangle \): \( x \) is free in \( \lambda y \mathord{\cdot} e \) if \( x \neq y \) and \( x \) is free in \( e \).
- \( \langle A \rangle \): \( x \) is free in \( f(e) \) if it is free in either \( f \) or \( e \) (i.e., also in both).

A.6.3 Substitution

In RSL, the following rules for substitution apply:

- \( \text{subst}(\langle N/x|x \rangle) \equiv N; \)
- \( \text{subst}(\langle N/x|a \rangle) \equiv a, \)
  for all variables \( a \neq x; \)
- \( \text{subst}(\langle N/x|(P \ Q) \rangle) \equiv (\text{subst}(\langle N/x|P \rangle) \text{subst}(\langle N/x|Q \rangle)); \)
- \( \text{subst}(\langle N/x|(\lambda x \mathord{\cdot} P) \rangle) \equiv \lambda y \mathord{\cdot} P; \)
- \( \text{subst}(\langle N/x|(\lambda y \mathord{\cdot} P) \rangle) \equiv \lambda y \mathord{\cdot} \text{subst}(\langle N/x|P \rangle), \)
  if \( x \neq y \) and \( y \) is not free in \( N \) or \( x \) is not free in \( P \);
- \( \text{subst}(\langle N/x|(\lambda y \mathord{\cdot} P) \rangle) \equiv \lambda z \mathord{\cdot} \text{subst}(\langle N/z|\text{subst}(\langle z/y|P \rangle)), \)
  if \( y \neq x \) and \( y \) is free in \( N \) and \( x \) is free in \( P \)
  (where \( z \) is not free in \( (N \ P) \)).

A.6.4 \( \alpha \)-Renaming and \( \beta \)-Reduction

- \( \alpha \)-renaming: \( \lambda x \mathord{\cdot} M \)
  If \( x, y \) are distinct variables then replacing \( x \) by \( y \) in \( \lambda x \mathord{\cdot} M \) results in \( \lambda y \mathord{\cdot} \text{subst}(\langle y/x|M \rangle). \)
  We can rename the formal parameter of a \( \lambda \)-function expression provided that no free variables of its body \( M \) thereby become bound.
• $\beta$-reduction: $(\lambda x \cdot M)(N)$

All free occurrences of $x$ in $M$ are replaced by the expression $N$ provided that no free variables of $N$ thereby become bound in the result. $(\lambda x \cdot M)(N) \equiv \text{subst}([N/x]M)$

### A.6.5 Function Signatures

For sorts we may want to postulate some functions:

**type**
- A, B, C

**value**
- $\text{obs}_B$: $A \rightarrow B$
- $\text{obs}_C$: $A \rightarrow C$
- $\text{gen}_A$: $BB \times C \rightarrow A$

### A.6.6 Function Definitions

Functions can be defined explicitly:

**value**
- $f$: Arguments $\rightarrow$ Result
  
  $f(\text{args}) \equiv D\text{ValueExpr}$

- $g$: Arguments $\sim \rightarrow$ Result
  
  $g(\text{args}) \equiv \text{ValueAndStateChangeClause}$
  
  $\text{pre}$ $P(\text{args})$

Or functions can be defined implicitly:

**value**
- $f$: Arguments $\rightarrow$ Result
  
  $f(\text{args})$ as result
  
  $\text{post}$ $P1(\text{args},\text{result})$

- $g$: Arguments $\sim \rightarrow$ Result
  
  $g(\text{args})$ as result
  
  $\text{pre}$ $P2(\text{args})$
  
  $\text{post}$ $P3(\text{args},\text{result})$

The symbol $\sim$ indicates that the function is partial and thus not defined for all arguments. Partial functions should be assisted by preconditions stating the criteria for arguments to be meaningful to the function.
A.7 Other Applicative Expressions

A.7.1 Simple let Expressions

Simple (i.e., nonrecursive) let expressions:

\[
\text{let } a = \mathcal{E}_d \text{ in } \mathcal{E}_b(a) \text{ end}
\]

is an “expanded” form of:

\[
(\lambda a.\mathcal{E}_b(a)) (\mathcal{E}_d)
\]

A.7.2 Recursive let Expressions

Recursive let expressions are written as:

\[
\text{let } f = \lambda a:A \cdot E(f) \text{ in } B(f,a) \text{ end}
\]

is “the same” as:

\[
\text{let } f = YF \text{ in } B(f,a) \text{ end}
\]

where:

\[
F \equiv \lambda g: \lambda a:A \cdot (E(g)) \text{ and } YF = F(YF)
\]

A.7.3 Predicative let Expressions

Predicative let expressions:

\[
\text{let } a:A \cdot \mathcal{P}(a) \text{ in } B(a) \text{ end}
\]

express the selection of a value \(a\) of type \(A\) which satisfies a predicate \(\mathcal{P}(a)\) for evaluation in the body \(B(a)\).

A.7.4 Pattern and “Wild Card” let Expressions

Patterns and wild cards can be used:

\[
\begin{align*}
\text{let } \{a\} \cup s = \text{set in } \ldots \text{ end} \\
\text{let } \{a\_\} \cup s = \text{set in } \ldots \text{ end} \\
\text{let } (a,b,\ldots,c) = \text{cart in } \ldots \text{ end} \\
\text{let } (a\_,\ldots,c) = \text{cart in } \ldots \text{ end}
\end{align*}
\]
let ⟨a⟩^l = list in ... end
let ⟨a, _, bb⟩^l = list in ... end
let [a⇒bb] ∪ m = map in ... end
let [a⇒b, _] ∪ m = map in ... end

A.7.5 Conditionals

Various kinds of conditional expressions are offered by RSL:

if b_expr then c_expr else a_expr
end

if b_expr then c_expr end ≡ /∗ same as: */
   if b_expr then c_expr else skip end

if b_expr_1 then c_expr_1
elsif b_expr_2 then c_expr_2
elsif b_expr_3 then c_expr_3
...
elsif b_expr_n then c_expr_n end

case expr of
   choice_pattern_1 ⇒ expr_1,
   choice_pattern_2 ⇒ expr_2,
   ...
   choice_pattern_n_or_wild_card ⇒ expr_n
end

A.7.6 Operator/Operand Expressions

⟨Expr⟩ ::= (PrefixOp) ⟨Expr⟩
   | ⟨Expr⟩ (InfixOp) ⟨Expr⟩
   | ⟨Expr⟩ (SuffixOp)
   | ...
⟨PrefixOp⟩ ::= − | ∼ | ∪ | ∩ | card | len | inds | elems | hd | tl | dom | rng
⟨InfixOp⟩ ::= = | ≠ | ≡ | + | − | * | ↑ | / | < | ≤ | ≥ | > | ∧ | ∨ | ⇒
   | ∈ | ∉ | ∪ | ∩ | \ | ⊂ | ⊆ | ⊇ | ⊃ | ⊆ | ⊇ | ⊊ | ⊋ | ⊃ |
⟨SuffixOp⟩ ::= !
A.8 Imperative Constructs

A.8.1 Statements and State Changes

Often, following the RAISE method, software development starts with highly abstract-applicative constructs which, through stages of refinements, are turned into concrete and imperative constructs. Imperative constructs are thus inevitable in RSL.

```
Unit value
stmt: Unit → Unit
stmt()
```

- Statements accept no arguments.
- Statement execution changes the state (of declared variables).
- `Unit → Unit` designates a function from states to states.
- Statements, `stmt`, denote state-to-state changing functions.
- Writing `()` as "only" arguments to a function "means" that `()` is an argument of type `Unit`.

A.8.2 Variables and Assignment

```
0. variable v:Type := expression
1. v := expr
```

A.8.3 Statement Sequences and `skip`

Sequencing is expressed using the ‘;’ operator. `skip` is the empty statement having no value or side-effect.

```
2. skip
3. stmt_1;stmt_2;...;stmt_n
```

A.8.4 Imperative Conditionals

```
4. if expr then stmt_c else stmt_a end
5. case e of: p_1→S_1(p_1),...,p_n→S_n(p_n) end
```

A.8.5 Iterative Conditionals

```
6. while expr do stmt end
7. do stmt until expr end
```
A.8.6 Iterative Sequencing

8. for e in list_expr • P(b) do S(b) end

A.9 Process Constructs

A.9.1 Process Channels

Let A and B stand for two types of (channel) messages and i:KIdx for channel array indexes, then:

```
channel c:A
channel { k[i]:B • i:KIdx }
```

declare a channel, c, and a set (an array) of channels, k[i], capable of communicating values of the designated types (A and B).

A.9.2 Process Composition

Let P and Q stand for names of process functions, i.e., of functions which express willingness to engage in input and/or output events, thereby communicating over declared channels. Let P() and Q stand for process expressions, then:

```
P ∥ Q Parallel composition
P [] Q Nondeterministic external choice (either/or)
P ] Q Nondeterministic internal choice (either/or)
P || Q Interlock parallel composition
```

express the parallel (∥) of two processes, or the nondeterministic choice between two processes: either external ([]) or internal (|). The interlock (||) composition expresses that the two processes are forced to communicate only with one another, until one of them terminates.

A.9.3 Input/Output Events

Let c, k[i] and e designate channels of type A and B, then:

```
c ?, k[i] ? Input
c ! e, k[i] ! e Output
```

expresses the willingness of a process to engage in an event that “reads” an input, respectively “writes” an output.
A.9.4 Process Definitions

The below signatures are just examples. They emphasise that process functions must somehow express, in their signature, via which channels they wish to engage in input and output events.

\[
\begin{align*}
\text{value} & \\
\text{P: } & \text{Unit } \to \text{ in } c \text{ out } k[i] \\
\text{Unit} & \\
\text{Q: } & i:K\text{ldx } \to \text{ out } c \text{ in } k[i] \text{ Unit} \\
\text{P() } & \equiv \ldots c \ ? \ldots k[i] \ ! e \ldots \\
\text{Q(i) } & \equiv \ldots k[i] \ ? \ldots c \ ! e \ldots
\end{align*}
\]

The process function definitions (i.e., their bodies) express possible events.

A.10 Simple RSL Specifications

Often, we do not want to encapsulate small specifications in schemes, classes, and objects, as is often done in RSL. An RSL specification is simply a sequence of one or more types, values (including functions), variables, channels and axioms:

\[
\begin{align*}
\text{type} & \\
\ldots & \\
\text{variable} & \\
\ldots & \\
\text{channel} & \\
\ldots & \\
\text{value} & \\
\ldots & \\
\text{axiom} & \\
\ldots &
\end{align*}
\]

In practice a full specification repeats the above listings many times, once for each “module” (i.e., aspect, facet, view) of specification. Each of these modules may be “wrapped” into scheme, class or object definitions.\(^{76}\)

A.11 RSL Index

\[
\begin{align*}
\text{Arithmetics} & \\
\ldots \ldots 2, -1,0,1,2,\ldots & , 91 \\
a_i^a_j & , 94 \\
a_i + a_j & , 94 \\
a_i / a_j & , 94 \\
a_i = a_j & , 93 \\
a_i \geq a_j & , 93 \\
\text{Cartesians} & \\
\ldots & \ldots (e_1, e_2, \ldots, e_n) , 95
\end{align*}
\]

\(^{76}\)For schemes, classes and objects we refer to [137, Chap. 10]
Chaos
\[ \text{chaos}, 97, 99 \]

Clauses
\[ \text{... elsif ... }, 105 \]
\[ \text{case } b_n \text{ of } p_{a_1} \to c_1, \ldots, p_{a_n} \to c_n \text{ end }, 105 \]
\[ \text{if } b_n \text{ then } c_n \text{ else } c_n \text{ end }, 105 \]

Combinators
\[ \text{let } a : A \cdot P(a) \text{ in } c \text{ end }, 104 \]
\[ \text{let } p_a = e \text{ in } c \text{ end }, 104 \]

Functions
\[ f(\text{args}) \text{ as result}, 103 \]
\[ \text{post } P(\text{args, result}), 103 \]
\[ \text{pre } P(\text{args}), 103 \]
\[ f(a), 102 \]
\[ f(\text{args}) \equiv \text{expr}, 103 \]

Imperative
\[ \text{case } b_n \text{ of } p_{a_1} \to c_1, \ldots, p_{a_n} \to c_n \text{ end }, 106 \]
\[ \text{do } \text{stmt} \text{ until } \text{be end}, 106 \]
\[ \text{for } e \text{ in } \text{list} \text{expr} \cdot P(b) \text{ do } \text{stm} \text{e}(e) \text{ end}, 107 \]
\[ \text{if } b_n \text{ then } c_n \text{ else } c_n \text{ end}, 106 \]
\[ \text{skip}, 106 \]
\[ \text{variable } v : \text{Type} := \text{expression}, 106 \]
\[ \text{while } \text{be } \text{do } \text{stm end}, 106 \]
\[ f(), 106 \]
\[ \text{stm}_1 ; \text{stm}_2 ; \ldots ; \text{stm}_n : , 106 \]
\[ v := \text{expression}, 106 \]

Lists
\[ <Q((i)) | i \in<1..\text{len}> \cdot P(a)>, 95 \]
\[ hAB, 95 \]
\[ \ell(i) , 98 \]
\[ (e_1 : e_{-1}), 95 \]
\[ (e_1, e_2, ..., e_n B, 95 \]
\[ \text{elems } \ell , 98 \]
\[ \text{hd } \ell , 98 \]
\[ \text{inds } \ell , 98 \]
\[ \text{len } \ell , 98 \]
\[ t1\ell , 98 \]

Logics
\[ b_i \lor b_j , 93 \]
\[ \forall a : A \cdot P(a), 94 \]
\[ \exists a : A \cdot P(a), 94 \]
\[ \exists a : A \cdot P(a), 94 \]
\[ \neg b , 93 \]
\[ \text{false}, 90, 93 \]
\[ \text{true}, 90, 93 \]
\[ a = a_j , 94 \]
\[ a \geq a_j , 94 \]
\[ a > a_j , 94 \]
\[ a \leq a_j , 94 \]
\[ a < a_j , 94 \]
\[ a \neq a_j , 94 \]
\[ b_i \Rightarrow b_j , 93 \]
\[ b_i \land b_j , 93 \]

Maps
\[ P(e) \Rightarrow G(m(e)) | c : E \equiv \text{dom } m \land P(e), 96 \]
\[ \text{[ ]}, 95 \]
\[ \text{[t_1 \mapsto v_1, t_2 \mapsto v_2, ..., t_n \mapsto v_n]}, 95 \]
\[ m_i \setminus m_j , 100 \]
\[ m_i \cup m_j , 100 \]
\[ m_i / m_j , 100 \]
\[ \text{dom } m , 100 \]
\[ \text{rng } m , 100 \]
\[ m_i = m_j , 100 \]
\[ m_i \cup m_j , 100 \]
\[ m_i \cap m_j , 100 \]
\[ m_i \neq m_j , 100 \]
\[ m(e) , 100 \]

Processes
\[ \text{channel } c : T , 107 \]
\[ \text{channel } \{ k[i] : T \text{:: } i : \text{KIIdx} \}, 107 \]
\[ c! e , 107 \]
\[ c? , 107 \]
\[ k[i]! e , 107 \]
\[ k[i]? , 107 \]
\[ P \ll Q, 107 \]
\[ P Q, 107 \]
\[ P \text{Unit} \rightarrow \text{in } \text{cout k[i] } \text{Unit}, 108 \]
\[ P\ll Q, 107 \]
\[ P Q, 107 \]
\[ Q : \text{KIIdx} \rightarrow \text{out } c \text{ in } k[i] \text{Unit}, 108 \]

Sets
\[ \{Q(a) | a : A \cdot a \in s \cdot P(a)\}, 94 \]
\[ \}, 94 \]
\[ \{e_1, e_2, ..., e_n\}, 94 \]
\[ \cap \{s_1, s_2, ..., s_n\}, 96 \]
\[ \cup \{s_1, s_2, ..., s_n\}, 96 \]
\[ \text{card } s , 96 \]
\[ e \in s , 96 \]
\[ e \notin s , 96 \]
\[ s_i = s_j , 96 \]
\[ s_i \neq s_j , 96 \]
\[ s_i \in s_j , 96 \]
\[ s_i \ni s_j , 96 \]
\[ s_i \leq s_j , 96 \]
\[ s_i \leq s_j , 96 \]
\[ \text{Types} \quad (T_1 \times T_2 \times ... \times T_n), 90 \]
\[ T^\ast , 90 \]
\[ T^\omega , 90 \]
\[ T_1 \times T_2 \times ... \times T_n , 90 \]
\[ \text{Bool}, 90 \]
Char, 90
Int, 90
Nat, 90
Real, 90
Text, 90
Unit, 106, 108
mk_id(s_1:T_1, s_2:T_2, ..., s_n:T_n), 90
s_1:T_1 s_2:T_2 ... s_n:T_n, 90

T = Type_Expr, 92
T_1 | T_2 | ... | T_1 | T_n, 90
T = \{ v:T \rightarrow P(v) \}, 92, 93
T = \{TE_1 | TE_2 | ... | TE_n \}, 92
T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_n \rightarrow T_{-\text{infset}}, 90
T_{-\text{set}}, 90

B  RSL^+  471

C  A Language of Domain Analysis & Description Prompts  472

D  A Description Narration Language  473
E Indexes

E.1 Philosophy Index

Philosophers:
- Anaximander of Miletus, 610–546 BC, 53
- Anaximenes of Miletus, 585–528 BC, 53
- Aristotle, 384–322 BC, 54
- Baruch Spinoza: 1632–1677, 55
- Bertrand Russell, 1872–1970, 58
- causality, 63
- Chrysippus of Soli: 279–206 BC, 55
- David Hume, 1711–1776, 56
- Demokrit, 460–370 BC, 53
- Dynamics, 64
- Edmund Husserl, 1859–1938, 58
- Empirical Propositions, 62
- Friedrich Ludwig Gottlob Frege, 1848–1925, 58
- Friedrich Schelling, 1775–1854, 58
- Georg Wilhelm Friedrich Hegel, 1770–1831, 57
- George Berkeley: 1685–1753, 56
- Gottfried Wilhelm Leibniz: 1646–1716, 55
- Heraklit of Efesos, a. 500 BC, 53
- Johann Gottlieb Fichte, 1752–1824, 57
- John Locke: 1632–1704, 55
- Kant, Immanuel: 1720–1804, 56
- Kinematics, 63
- Logical Positivism: 1920s–1936, 58
- Ludwig Wittgenstein, 1889–1951, 59
- Moritz Schlick, 1882–1936, 59
- Necessary and Empirical Propositions, 61
- Necessity and Possibility, 62
- Otto Neurath, 1882–1945, 59
- Parmenides of Elea, 501–470 BC, 53
- Plato, 427–347 BC, 54
- Primary Objects, 61
- René Descartes: 1596–1650, 55
- Rudolf Carnap, 1891–1970, 59
- Socrates, 470–399 BC, 54
- Space: Direction and Distance, 62
- states, 62

Ideas:
- Das Ding an sich, Kant, 56
- Das Ding für uns, Kant, 56
- esse est precipi, Berkeley, 56
- “pull”, gravitational, 65
- “reasoning apparatus”, 57
- “things”, 57
- ‘matter’, Russell, 59
- abstract ideas, Plato, 54
- acceleration
- primary entity, 64
- action, 69
- action, Aristotle, 54
- agent cause, Aristotle, 54
- all is changing, Heraklit, 53
- all is flux, Heraklit, 53
- An Enquiry Concerning Human Understanding, Hume, 1748-1750, 56
- animal, 66
- artifact, 67
- discrete endurant values, 72
- discrete endurants, 72
- artifacts, 79
- artifactual
- perdurants, 79
- asymmetric, 62
- attraction,
mutual, 65
deutung = reference, Frege, 58
behavioral sciences, 79
behaviour, 69
being, Aristotle, 54
biology, 78
botanics, 78
categorical schema, Kant, 57
categories, Aristotle, 54
categories, Aristotle, Kant, 54
causal implication, 63
causal principle, 63
causality
  of purpose, 65
causality of
  purpose, 70
cause (= explanation), Aristotle, 54
cause effect category, Kant, 57
cause, Kant, 57
chemical, 79
chemistry, 78
Christianity, 55
cognition, Locke, 56
composite
  ideas, Hume, 56
  proposition, The Stoics, 55
  sense impressions, Hume, 56
conceptions, Hume, 56
concrete world, Aristotle, 54
conjunction, 62
conjunction, The Stoics, 55
constant of
  nature, 65
contradiction principle, Sørlander, 61
contradiction,
  principle of, 66, 70
contradiction, Kant, 57
corporeal substance, 55
Das Ding an sich
  Das Ding für uns, Kant, 57
deduction,
  transcendental, 29, 67
describing the world, Aristotle, 54
designation, 72
development, 65
dialectic reasoning, Zeno, 53
dialectism
  ancient, Zeno, 53
  modern, Hegel, 58
different, 62
direction, 62
direction,
  vectorial, 64
discrete endurant values,
  artifact, 72
  natural, 72
discrete endurants,
  artifact, 72
  natural, 72
disjunction, 62
disjunction, The Stoics, 55
distance, 62
dynamics, 64
electrical, 79
electricity, 78
electronics, 79
empirical
  proposition, Sørlander, 61, 62
end cause, Aristotle, 54
endurants,
  natural, 78
engineering, 79
entity,
  man-made, 67
epistemology, 51
eternal, Parmenides, 53
ethics, 66
Euclidean Geometry, 62
event, 69
exchange, 66
explanation (= cause), Aristotle, 54
extent
  spatial, 63
  temporal, 63
feel, 66
feeling, 66
feelings, 75
force, 64, 65
form, 65
spatial, 63
form cause, Aristotle, 54
formal cause, Aristotle, 54
genome, 66
gravitation,
universal, 65
gravitational
“pull”, 65
gravity, 64
History of Western Philosophy, Russell, 1945, 1961, 59
human, 1, 66
humans, 79
ideas
    composite, Hume, 56
    simple, Hume, 56
identical, 62
identity, 62, 71
implication, 62
implication, The Stoics, 55
implicit
    meaning theory, 66, 70
in-between, 62
incentive, 66
incentives, 75
Indiscernability of Identicals, Leibniz, 55
influence, 65
inner determination, 75
instinct, 66
instincts, 75
intensional, 72
    relation, 72
intent, 72
intentional “pull”, 68, 73
Intentionality, 67
intentionality, Husserl, 58
intuition forms, Kant, 57
irreducible types of predicates, Aristotle, 54
kinematics, Sørlander 63–64
knowable, Kant, 57
knowledge, 66
language, 66, 70, 75
language and meaning
    possibility, 65
learn, 66, 75
life sciences, 78
living
    species, 78
living species, 65
location
    spatial, 63
location, Aristotle, 54
Logical Conditions for Describing Living Worlds, Sørlander, 65
Logical Conditions for Describing Physical Worlds, Sørlander, 62
man-made
    entity, 67
Mass, 64
mass, 65
    of primary entity, 64
material cause, Aristotle, 54
material substance, Descartes, 55
matter, 64
matter, Aristotle, 54
meaning, 72
meaning and language
    possibility, 65
meaning theory,
    implicit, 66, 70
meaning theory, Wittgenstein, 59
means of motion, 66, 75
mechanical, 79
mechanics, 78
memory, 66, 70
mind and form, 55
modalities
    necessity, reality, possibility, Aristotle, 54
modality
    necessity, Aristotle, 54
    possibility, Aristotle, 54
    reality, Aristotle, 54
modality, Aristotle, 54
movement
    primary entity, 63
movement,
    state of, 65
movement, Parmenides, 53
mutual
attraction, 65
mutual attraction, 65
universal attraction, 65
mutual influence, 65
natural
discrete endurant values, 72
discrete endurants, 72
endurants, 78
nature,
constant of, 65
necessarily true, 62
necessarily true, Sørølander, 62
necessary
proposition, Sørølander, 61
truth, Sørølander, 61
Newton’s Laws, 64
no necessity for cause and effect, Hume, 56
non-logical implicative, 63
nothing exists, Heraklit, 53
of purpose,
causality, 65
one substance, Spinoza, 55
ontology, 52
organ,
sensory, 66
part,
physical, 1, 70
perdurants,
artifactual, 79
permanence, 53
phenomenology, Husserl, 58
phenomenon, Plato, 54
Philosophische Untersuchungen, [50]
Ludwig Wittgenstein, 1953, 59
Philosophy historically seen, Hegel, 58
Philosophy of Logical Atomism [47], Bertrand Russell, 1918, 58
Philosophy, https://en.wikipedia.org/-wiki/Philosophy, 51
physical
part, 1, 70
physics, 78
plant, 66
position, Aristotle, 54
possibility
of language and meaning, 65
possibility of
truth, Sørølander, 61
possibly true, Sørølander, 62
posture, Aristotle, 54
Pramana, Wikipedia, 52
primary
entities, Sørølander, 62
qualities, Locke, 56
primary entities, 63
primary entities, Sørølander, 62
primary entity, 62, 63
acceleration, 64
mass, 64
movement, 63
rest, 63
velocity, 63
primary qualities, Locke
not necessarily objective, Hume, 56
principle of
contradiction, 66, 70
proof by contradiction, Zeno, 53
propagational
speed limit, 65
property
spatial, 63
proposition, 62
composite, The Stoics, 55
empirical, Sørølander, 61, 62
necessary, Sørølander, 61
simple, The Stoics, 55
proposition, The Stoics, 55
proposition, Sørølander, 62
Protestantism, Martin Luther, 55
purpose, 75
purpose cause, Aristotle, 54
purpose,
causality of, 70
purposeful
movement, s, 66
purposefulness, 66
qualities
primary, Locke, 56
secondary, Locke, 56
quality, Aristotle, 54
quantity, Aristotle, 54
reality, Kant, 57
reason and reality identity, Hegel, 58
reductio ad absurdum, Zeno, 53
reference, Frege, 58
reflection ideas, Locke, 55
relation,
   intensional, 72
relation, Aristotle, 54
Renaissance, 55
responsibility, 66, 70
rest
   primary entity, 63
secondary
   qualities, Locke, 56
secondary qualities, Locke
   not necessarily subjective, Hume, 56
self awareness, Kant, 57
self-awareness, Kant, 61
sense ideas, Locke, 55
sense impressions
   composite, Hume, 56
   simple, Hume, 56
sense impressions, Hume, 56
sense, Frege, 58
sensing, Locke, 55
sensory
   organ, 66
sensory organs, 75
sign, 66
simple
   ideas, Hume, 56
   proposition, The Stoics, 55
sense impressions, Hume, 56
sinn = sense, Frege, 58
skepticism, 53
solipsism, 63
source, 65
Space, 62
space, Kant, 57
spatial
   extent, 63
   form, 63
   location, 63
   property, 63
species,
   living, 78
   speed, 64
   speed limit, 64
   propagational, 65
   stable, 65
   state, 63
   state of
   movement, 65
substance, 53
   corporeal, Descartes, 55
   material, Descartes, 55
   thinking, Descartes, 55
substance, Aristotle, 54
suffering, Aristotle, 54
temporal
   extent, 63
theory of Ideas, Plato, 54
thesis, antithesis, synthesis, Hegel, 58
thinking substance, Descartes, 55
time, 69
time relation, 63
time, Aristotle, 54
time, Kant, 57
Tractatus Logico-Philosophicus [49],
   Ludwig Wittgenstein, 1921, 59
transcendental
   deduction, 29, 67
   transcendental deduction, Kant, 57
   Transcendental Schemata, Kant, 57
   transitive relation, 62
unchanging, Parmenides, 53
unify change and permanence, Demokrit, 53
time
   change, 63
time, Aristotle, 54
time, Kant, 57
Tractatus Logico-Philosophicus, 49,
   Ludwig Wittgenstein, 1921, 59
transcendental
   deduction, 29, 67
   transcendental deduction, Kant, 57
   Transcendental Schemata, Kant, 57
   transitive relation, 62
unchanging, Parmenides, 53
universal
   gravitation, 65
   mutual attraction, 65
unknowable, Kant, 57
vectorial
   direction, 64
velocity
primary entity, 63
verification conditions, 58
Vienna Circle, Wiener Kreis, 58
weight, 64
Wiener Kreis, Vienna Circle, 59
zoology, 78

E.2 Domain Analysis Index

E.2.1 Concepts

“thing”, 7
abstract
  value, 13
abstraction, 7
action, 20
analysis and description
  domain
    method, 6
method
  domain, 6
A-series, time, 29
axiom, 40
axiomatised
  sorts, 28
behaviour, 20
B-series, time, 29
channels, 20
conceive, 7
condition
  post, 40
deduction
  transcendental, 40
description
  domain
    prompt, 14
prompt
  domain, 14
domain
  analysis and description
    method, 6
description
    prompt, 14

Substance:
  air, Anaximenes, 53
  apeiron, Anaximander, 53
  atom, Demokrit, 53
  fire, Heraklit, 53
  water, Thales, 53

method
  analysis and description, 6
prompt
  description, 14
endurants, 40
Euclid of Alexandria, 28
Euclidian
  Space, 28
event, 20
identifier
  unique, 13
input, 20
internal
  qualities, 9, 25, 40
mereology, 11
  observer, 14
type, 14
method
  analysis and description
  domain, 6
domain
  analysis and description, 6
obligation
  proof, 40
observe, 7
observe_part_type
  prerequisite
  prompt, 11
prompt
  prerequisite, 11
observer
  mereology, 14
E.2.2 Definitions

“being”, 12

A Domain Analysis and Description Method, 12

action
discrete, 29
active
attribute, 26
Actor, 29
actor, 29
analysis and description
domain, 11
method, 12
Artifact, 67
Atomic part, 16
attribute
active, 26
biddable, 26
dynamic, 25
inert, 26
programmable, 26
reactive, 26
static, 25
autonomous
attribute, 26

behaviour, 69
discrete, 30
biddable
attribute, 26

qualities
internal, 9, 25, 40

sort
axiomatised, 28
Space
Euclidian, 28
space, 27
spacetime, 26
state, 20
sub-part, 10
time, 20
A-series, 29
B-series, 29
continuum theory, 29
transcendental
deduction, 40
type
mereology, 14
unique
identifier, 13
value
abstract, 13
Component, 18
component, 18
Composite
part, 16
Composite Part, 16
continuous
endurant, 14
Continuous Endurant, 14
description
domain
prompt, 22
prompt
domain, 22
discrete
action, 29
behaviour, 30
endurant, 14
Discrete Action, 29
Discrete Behaviour, 30
Discrete Endurant, 14
Domain, 8
domain
analysis and description, 11
method, 12
description
prompt, 22
method
analysis and description, 12
prompt
description, 22
Domain Analysis and Description, 11
dynamic
attribute, 25
Endurant, 13
endurant, 13
continuous, 14
discrete, 14
Entity, 12
entity, 12
Epistemology, 8
Event, 30
event, 30
Hausdorff
space, 46
inert
attribute, 26
Intentional Pull, 73
Intentional Relations, 72
internal
qualities, 23
Material, 20
material, 20
mereology, 22
Metaphysics, 8
method
analysis and description
domain, 12
domain
analysis and description, 12
metric
space, 47
Metric Space, 47
On Intentional Pull, 68
open
set, 47
Part, 16
part, 16
Atomic, 16
Composite, 16
Parts, 15
Perdurant, 13
perdurant, 13
phenomenon, 12
prerequisite
prompt, 17
is, entity, 13
programmable
attribute, 26
prompt
description
domain, 22
domain
description, 22
prerequisite, 17
qualities
internal, 23
reactive
attribute, 26
set
open, 47
space
Hausdorf, 46
metric, 47
topological, 46
State, 29
state, 29
static
attribute, 25
Structure, 14
structure, 14
sub-part, 16
topological
space, 46
Topological Space, 46
topology, 47
Transcendental, 28
Transcendental Transformation, 28
Transcendentality, 28

E.2.3 Analysis Predicates

1. is_universe_of_discourse, 12
10. is_composite, 16
11. observe_endurants, 17
13. has_components, 19
14. is_component, 19
15. has_materials, 20
16. is_material, 20
17. type_name, 21
18. has_mereology, 22
19. attribute_types, 24
2. is_entity, 13
20. is_static_attribute, 25
21. is_dynamic_attribute, 25
22. is_inert_attribute, 26
23. is_reactive_attribute, 26
24. is_active_attribute, 26
25. is_autonomous_attribute, 26
26. is_biddable_attribute, 26
27. is_programmable_attribute, 26
28. is_physical, 70
29. is_living, 70
30. is_natural, 71
31. is_artifactual, 71
32. is_plant, 71
33. is_animal, 71
34. is_human, 71

E.2.4 Description Observers

[1] observe_universe_of_discourse, 12
[2] observe_endurant_sorts, 17
[3] observe_part_type, 18
[6] observe_unique_identifier, 21
[7] observe_mereology, 22
[8] observe_attributes, 24

E.2.5 Proof Obligations and Axioms
\mathcal{A} : \text{Disjointness of Domain Identifier Types}, \quad 22
\mathcal{A} : \text{Well-formedness of Mereologies}, \quad 23
\mathcal{PO} : \text{Disjointness of Attribute Types}, \quad 25
\mathcal{PO} : \text{Disjointness of Component Sorts}, \quad 19
\mathcal{PO} : \text{Disjointness of Endurant Sorts}, \quad 17

E.2.6 \quad \textbf{Observer Function Literals}

\eta \quad \begin{align*}
E, & \quad 17 \\
P, & \quad 22 \\
\text{attr}_-, & \quad 25 \\
is_-, & \quad 17, 19, 25 \\
\text{obs\_attrib\_values}_-, & \quad 25 \\
\text{obs\_endurant\_sorts}_-, & \quad 17 \\
\end{align*} 

\begin{align*}
\text{obs\_mereo}_-, & \quad 23 \\
\text{obs\_part}_-, & \quad 18 \\
\text{uid}_-, & \quad 22 \\
\text{obs\_components}_-, & \quad 19 \\
\text{obs\_mat\_sort}_-, & \quad 20 \\
is_-, & \quad 21 \\
\text{obs\_materials}_-, & \quad 20 \\
\end{align*}