Where to Begin –
On Pattern Language Entry Points

University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany

Lukas Reinfurt[1,2], Michael Falkenthal[1], Frank Leymann[1]
[1]Institute of Architecture of Application Systems
[2]Daimler AG
lukas.reinfurt@iaas.uni-stuttgart.de

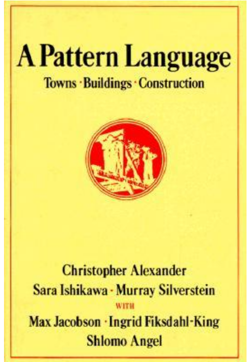
Phone	+49-711-685 88474
Fax	+49-711-685 88472

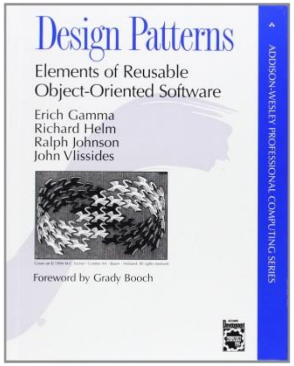
Patterns

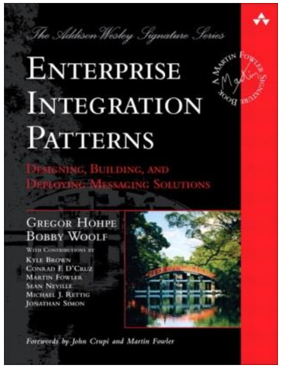
> "[A] pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

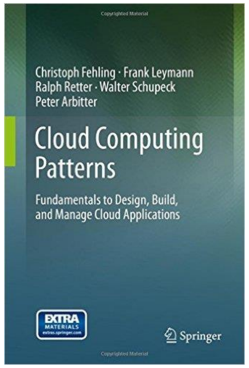
Alexander, C., Ishikawa, S., and Silverstein, M. 1977. *A Pattern Language: Towns, Buildings, Construction*

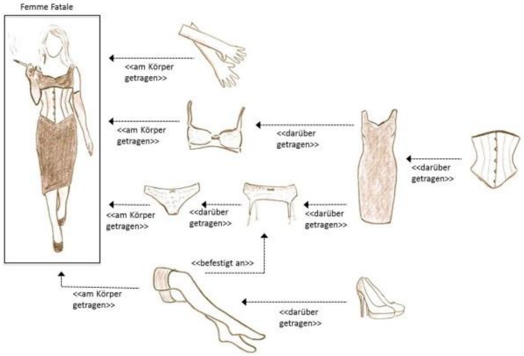
- Started by Christopher Alexander
- Today applied in many domains, including computing


A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel

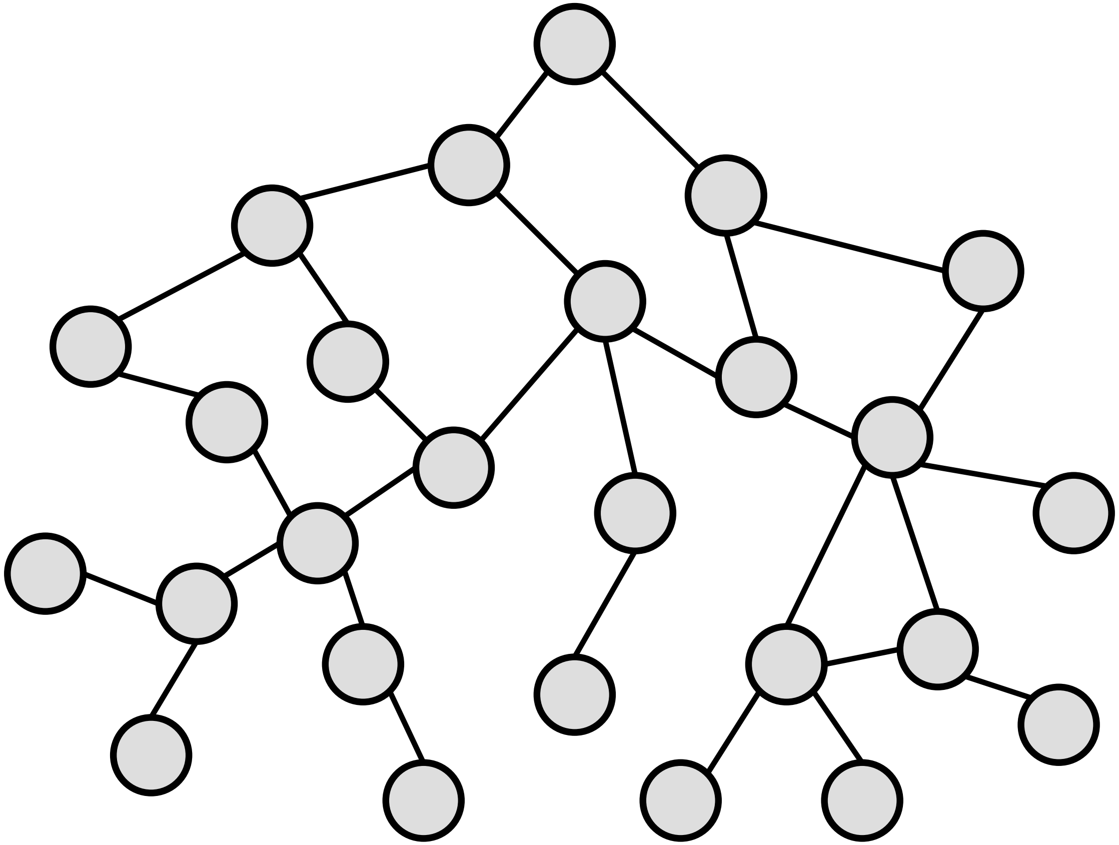

Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES


The Addison-Wesley Signature Series
ENTERPRISE INTEGRATION PATTERNS
DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS
GREGOR HOHPE
BOBBY WOOLF
Forewords by John Crupi and Martin Fowler

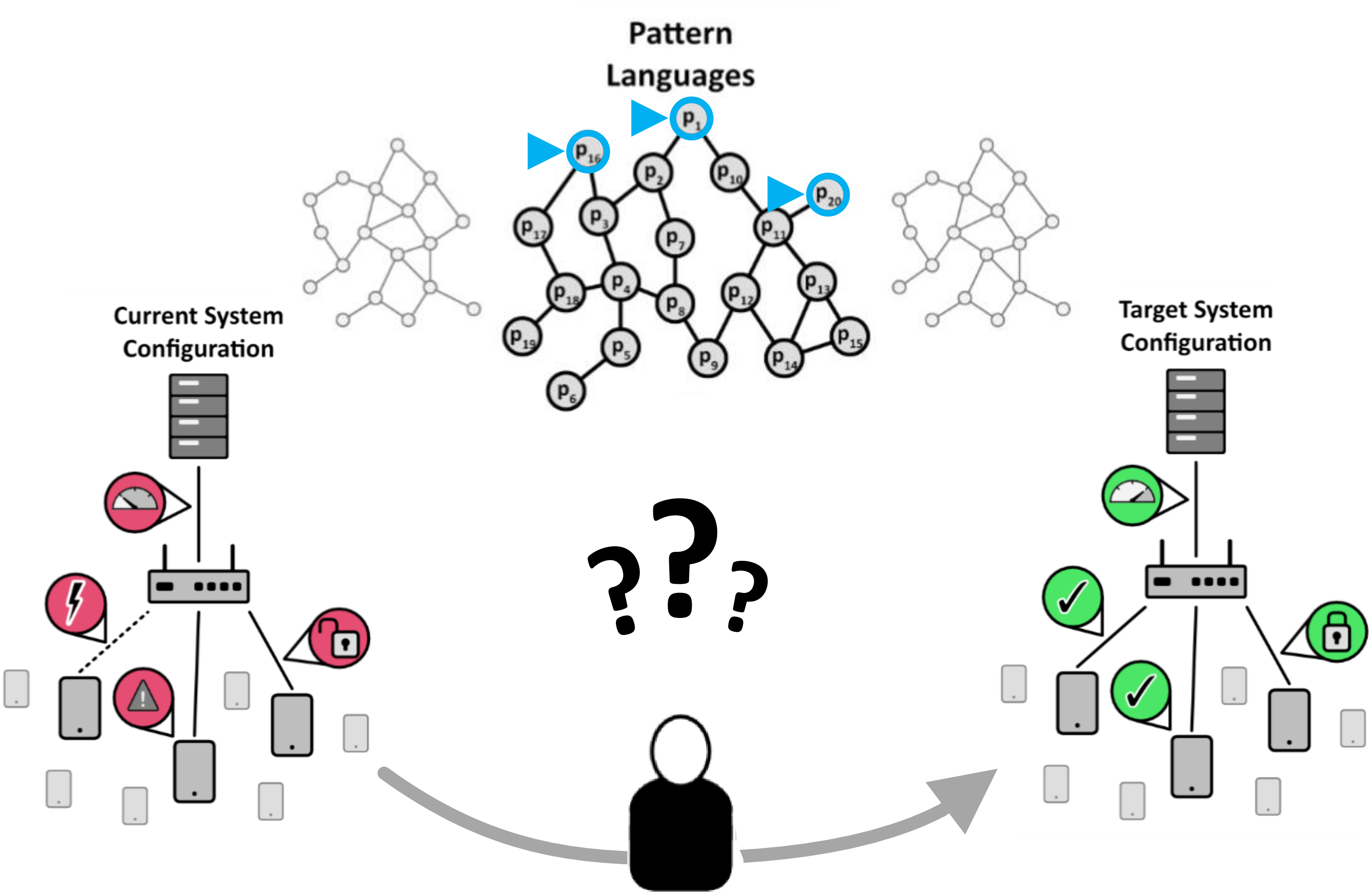

Christoph Fehling · Frank Leymann
Ralph Retter · Walter Schupeck
Peter Arbitter
Cloud Computing Patterns
Fundamentals to Design, Build, and Manage Cloud Applications
EXTRA MATERIALS
Springer



Pattern Languages



Motivation

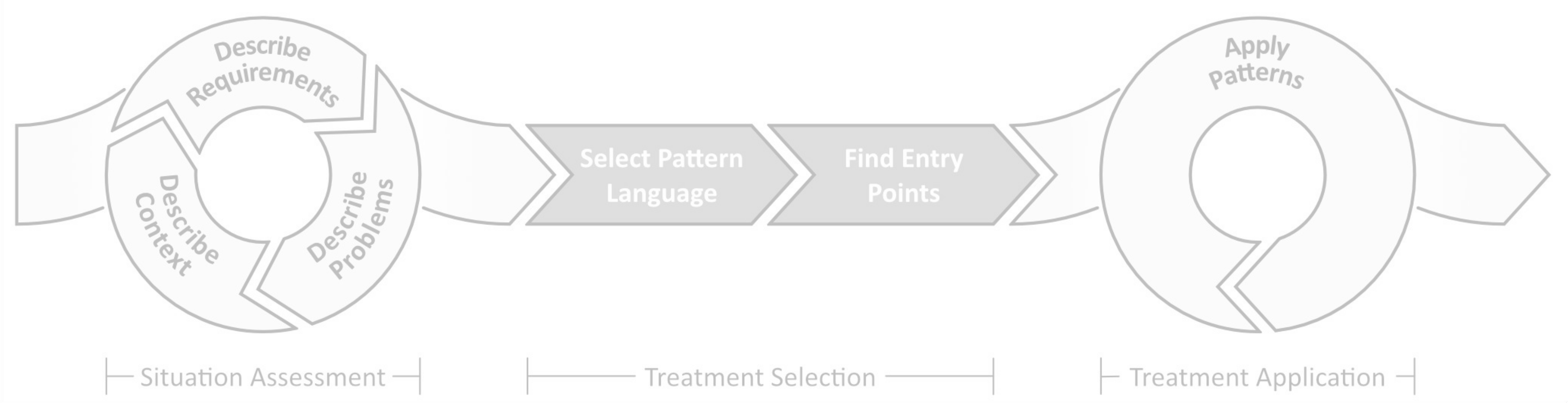


Pattern Languages

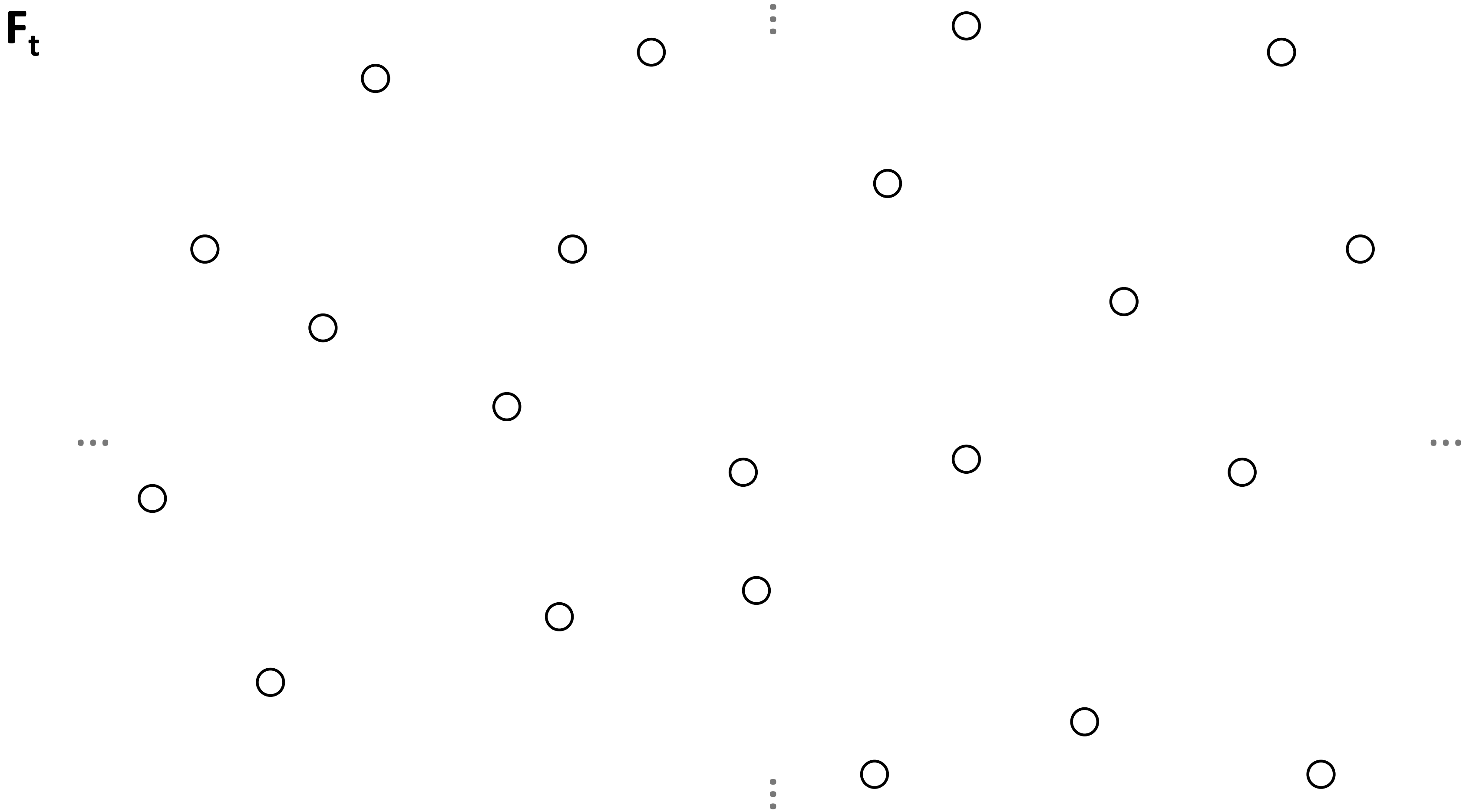
Current System Configuration

Target System Configuration

General Approach



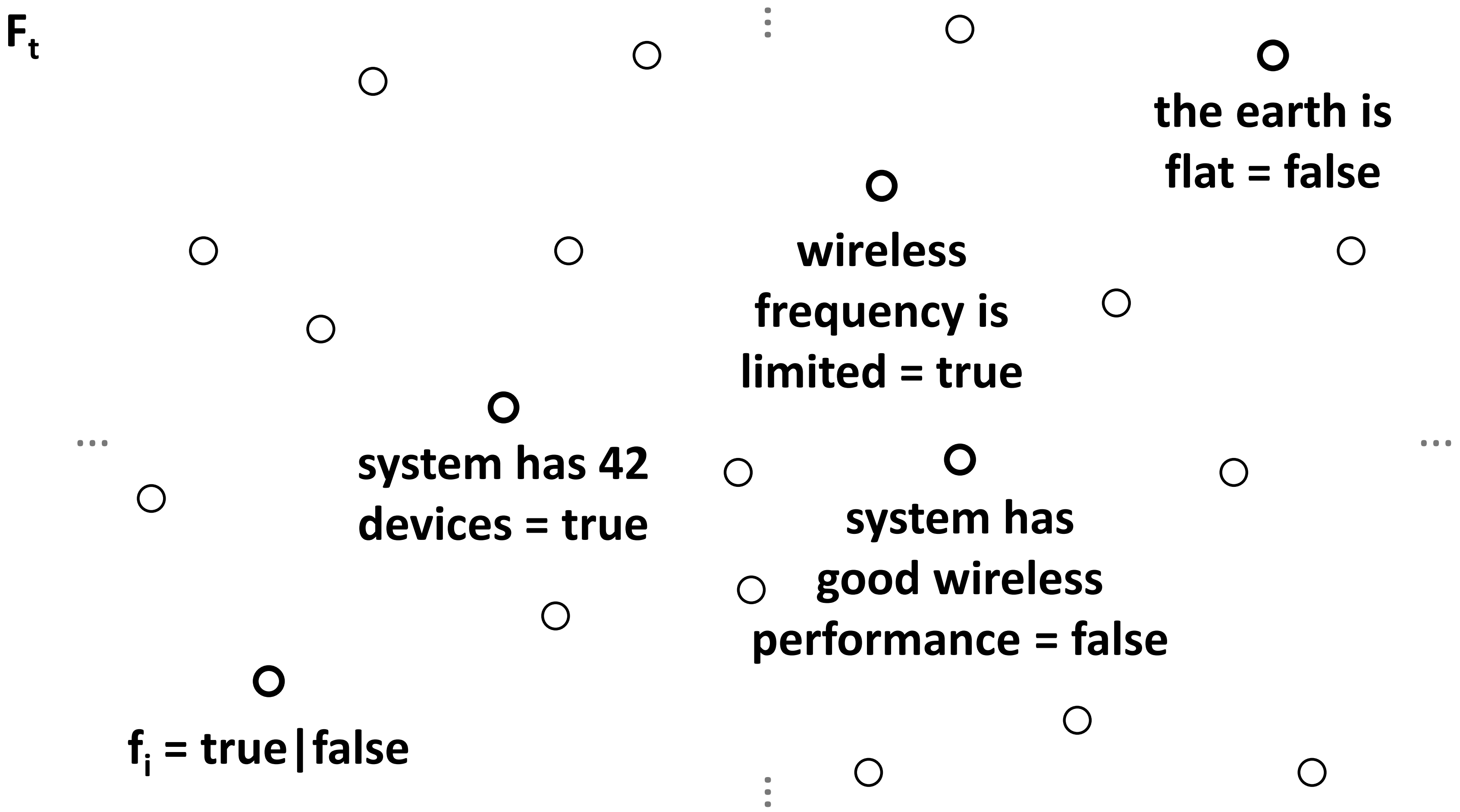
Facts



○ Fact

© Lukas Reinfurt

Facts



F_t

the earth is flat = false

wireless frequency is limited = true

system has 42 devices = true

system has good wireless performance = false

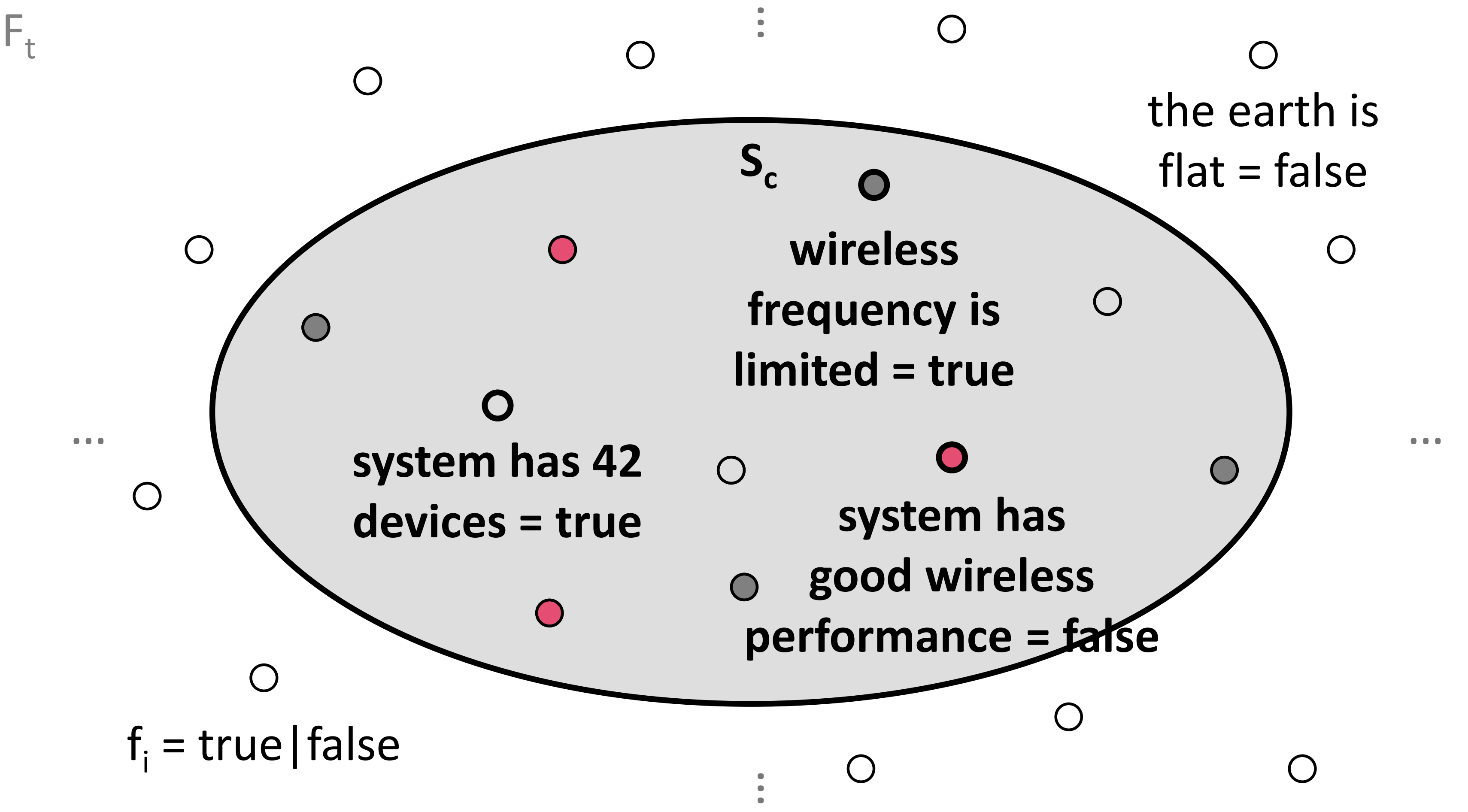
f_i = true|false

…

⋮

○ Fact

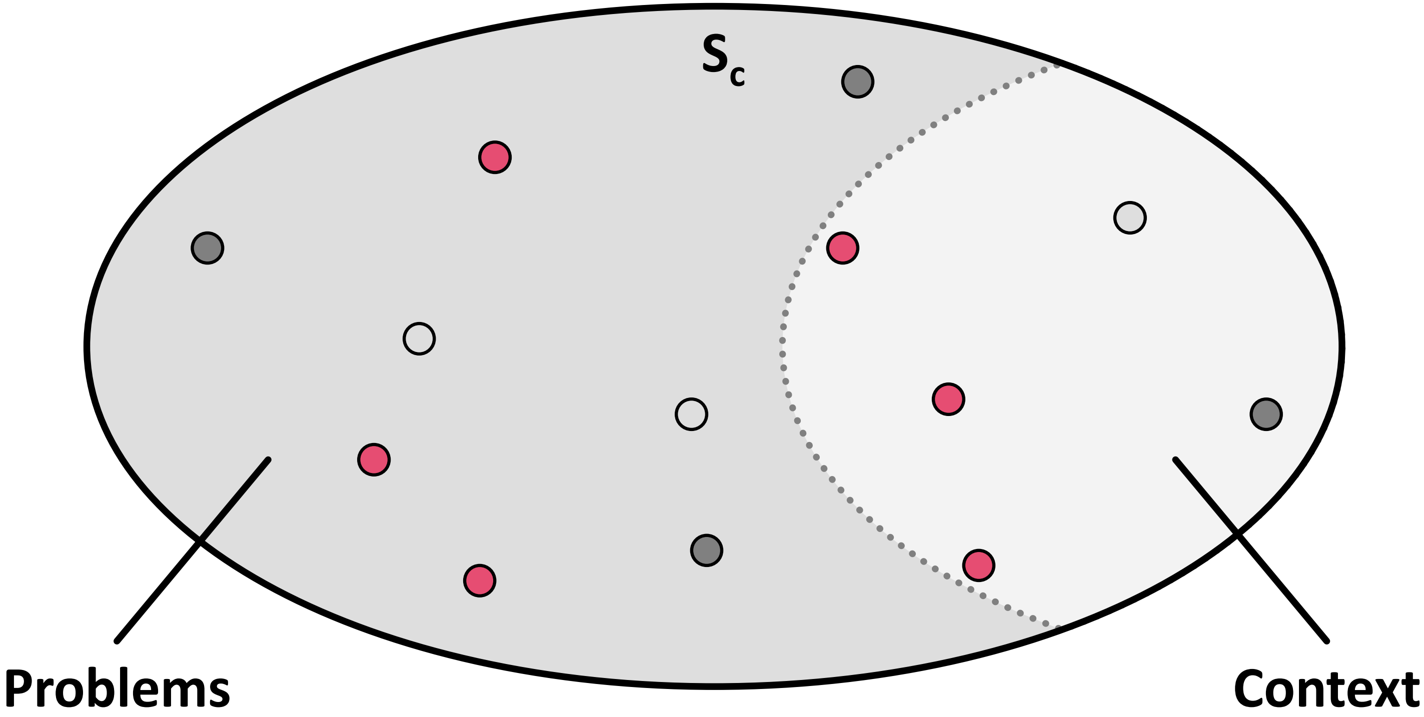
Situation


F_t
the earth is flat = false
S_c
wireless frequency is limited = true
system has 42 devices = true
system has good wireless performance = false
…
…
f_i = true|false

○ Fact ● Negative Fact ● Unchangeable Fact ⬭ Situation

Current Situation

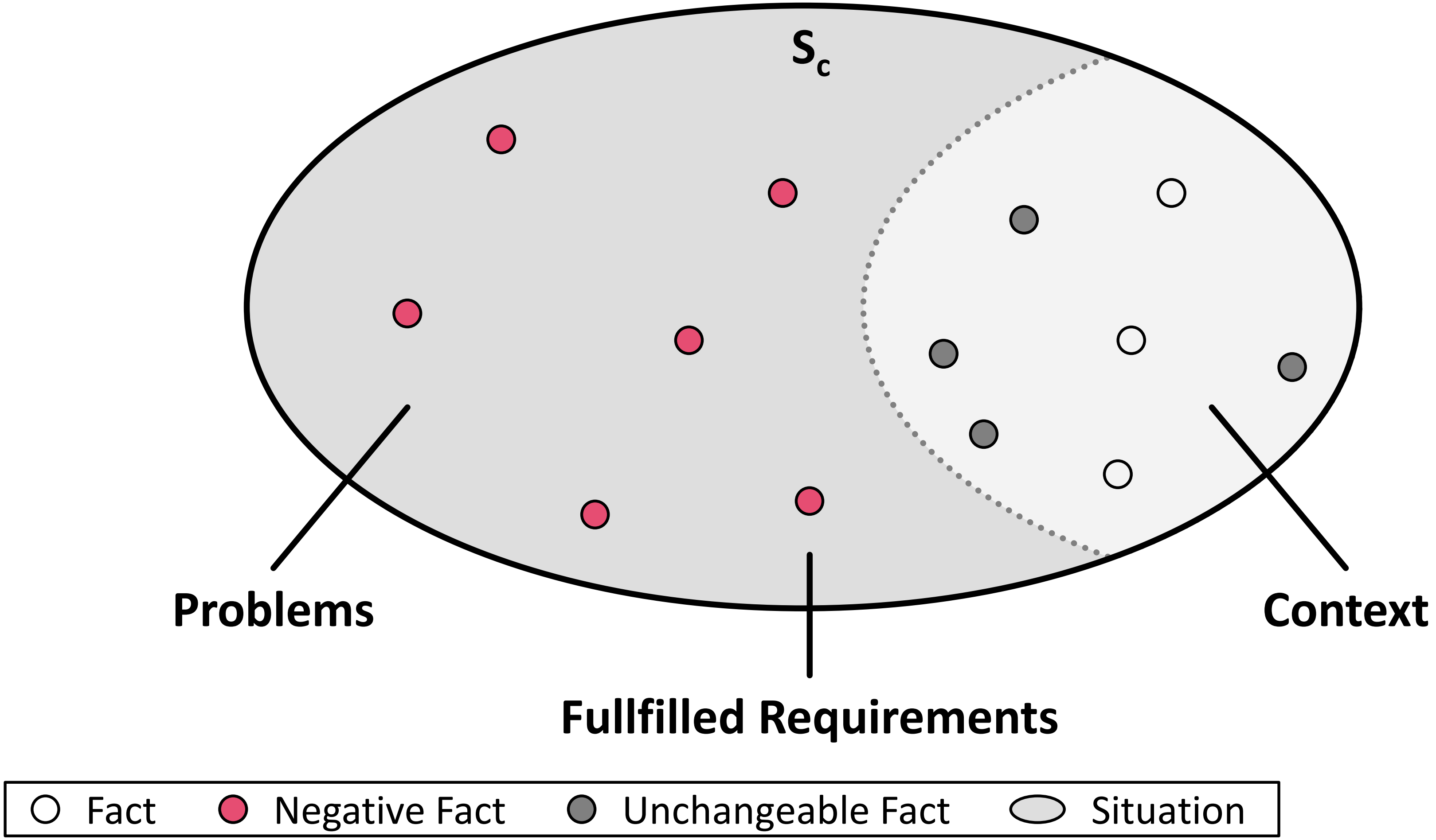
F_t



○ Fact ● Negative Fact ● Unchangeable Fact ⬭ Situation

Target Situation

F_t



S_c

Problems

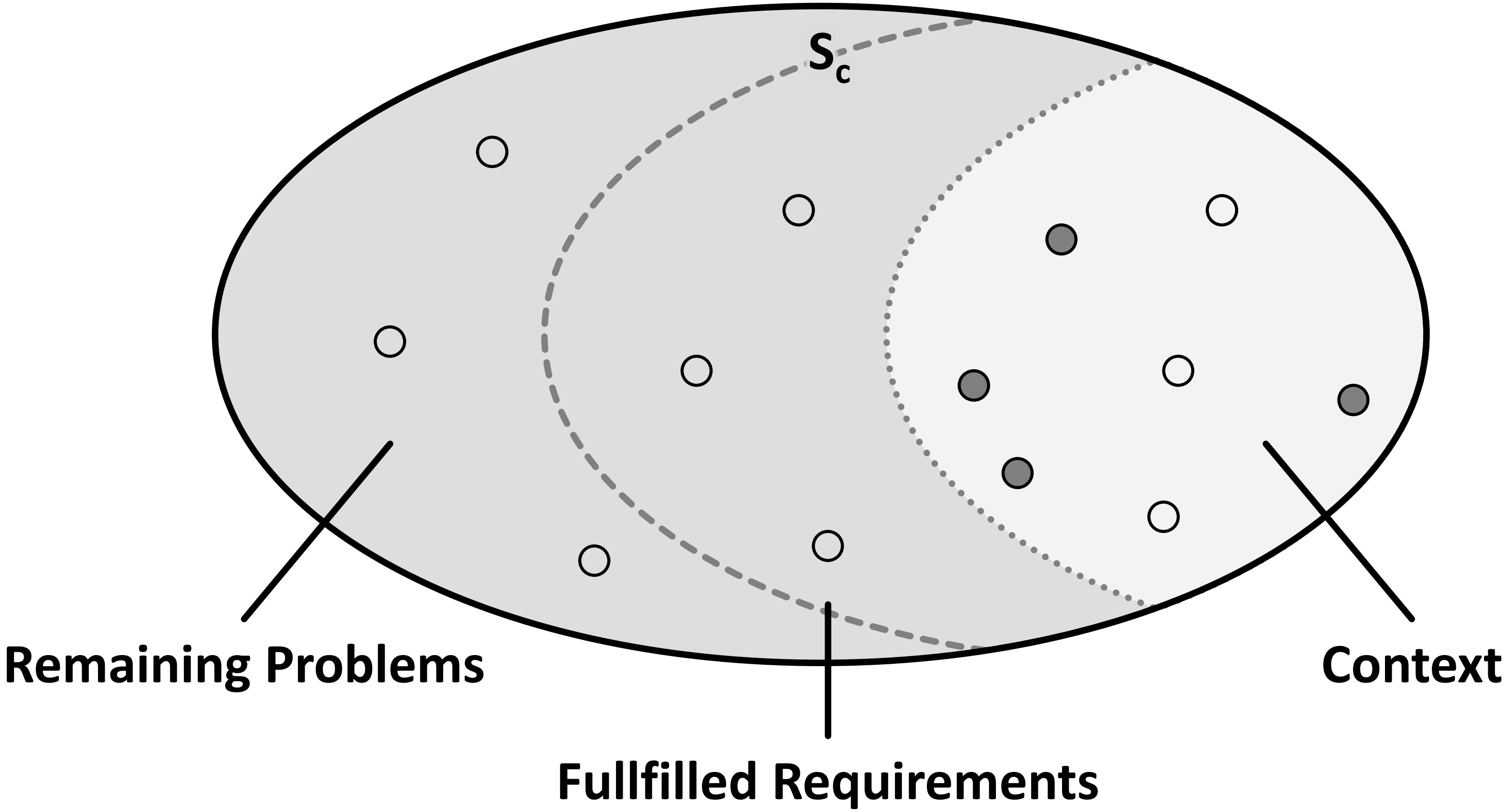
Fullfilled Requirements

Context

○ Fact ● Negative Fact ● Unchangeable Fact ⬭ Situation

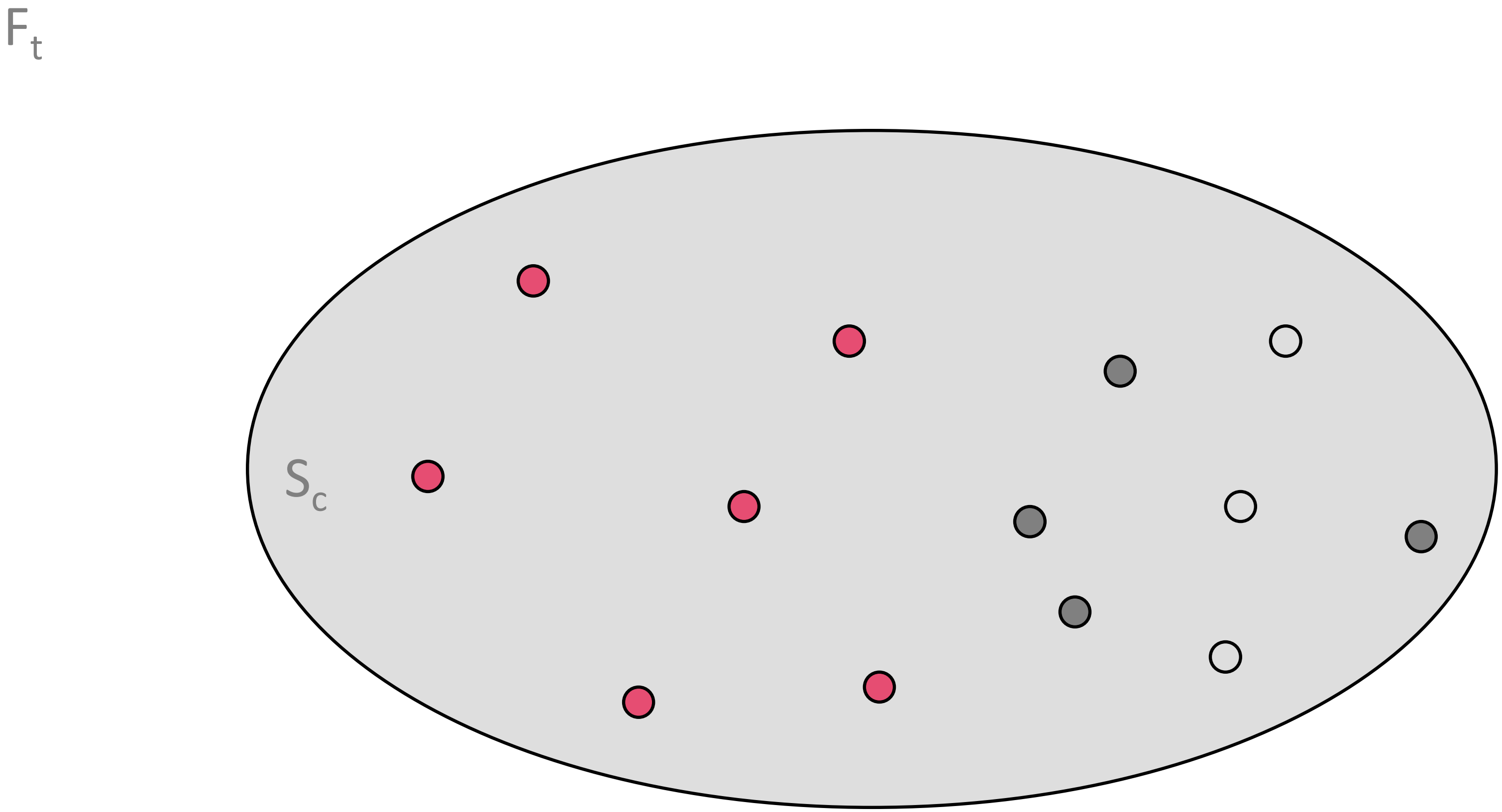
Path End Situation

F_t



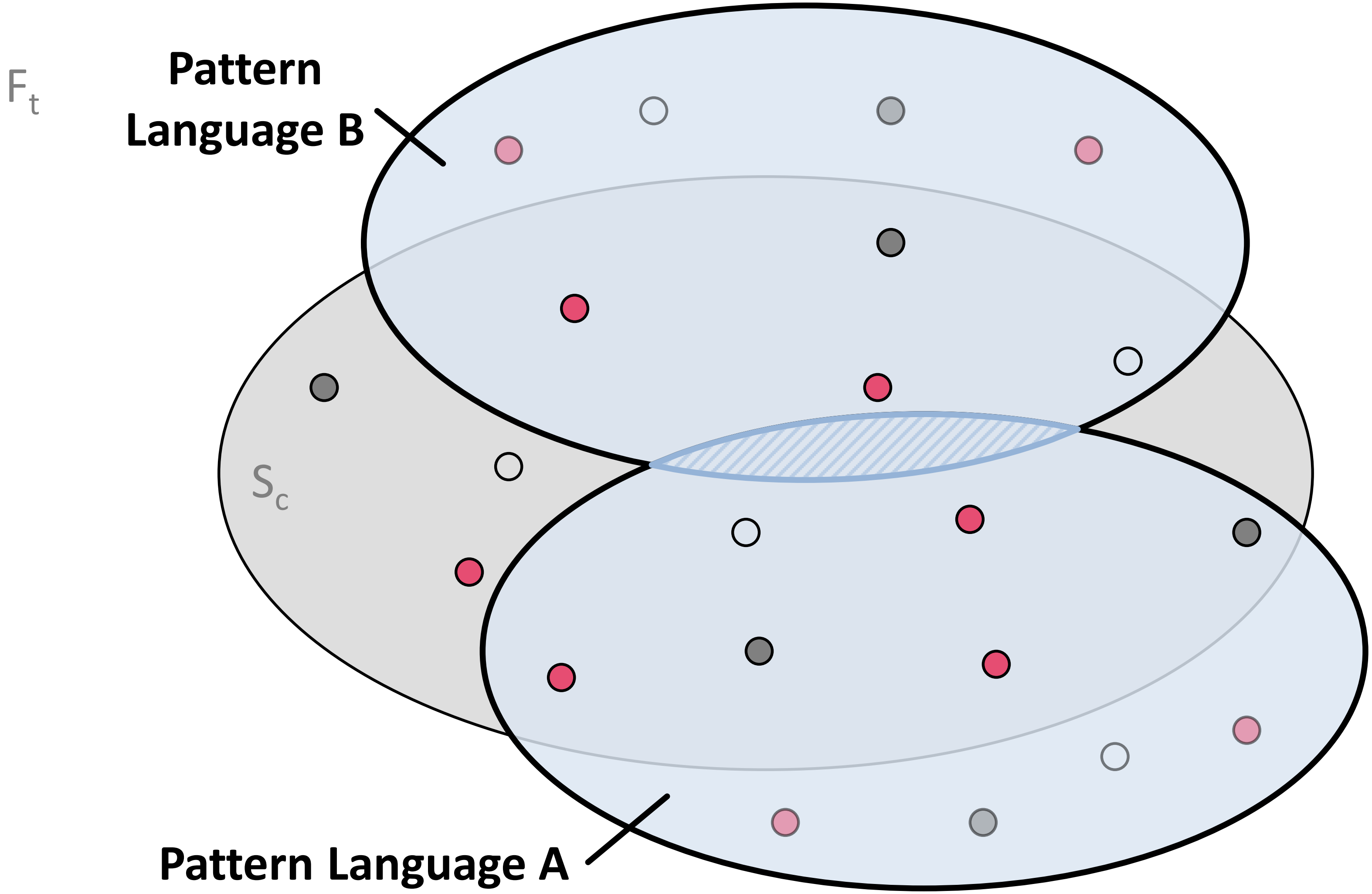
○ Fact ● Negative Fact ● Unchangeable Fact ⬭ Situation

Pattern Language



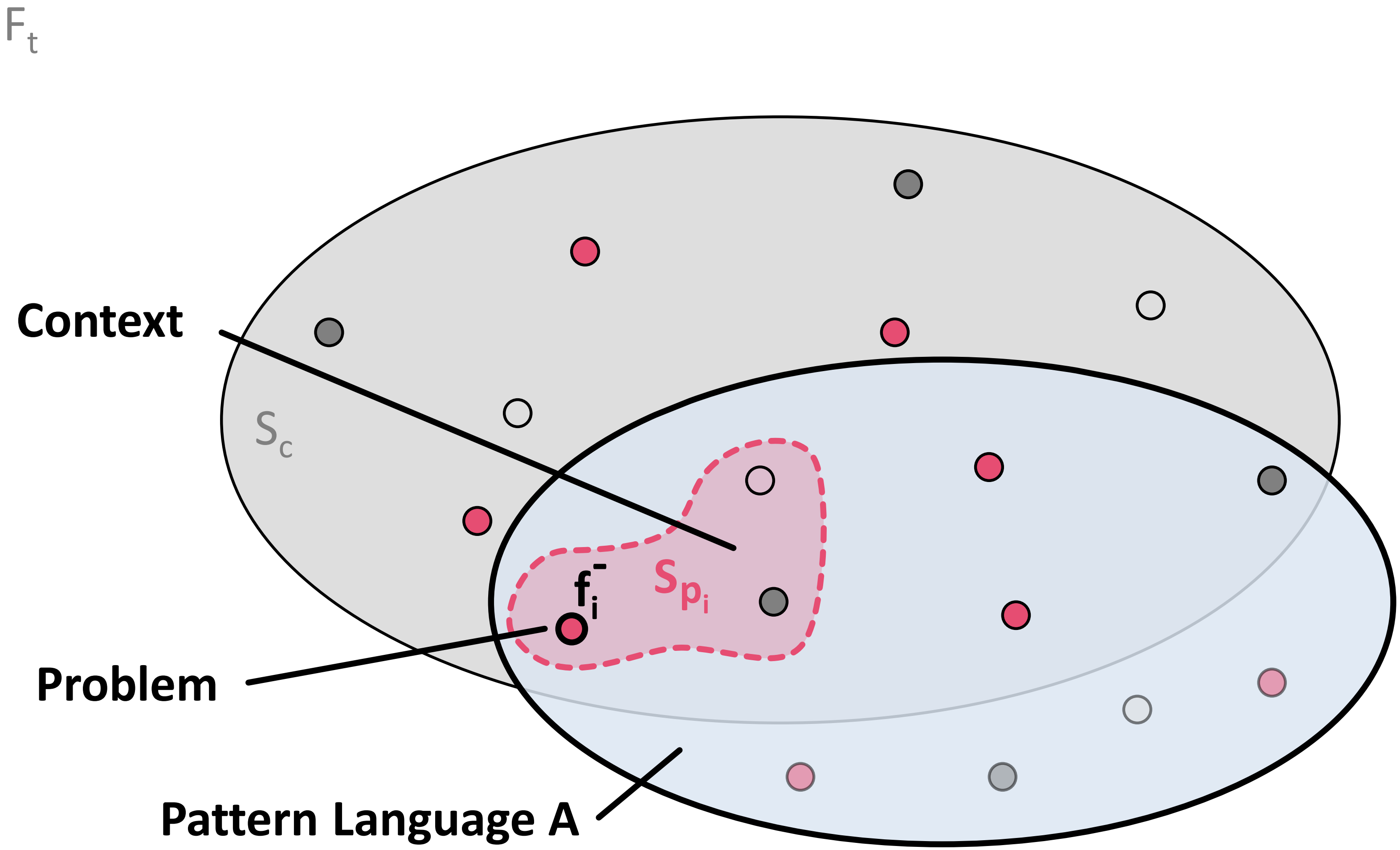
○ Fact / ● Negative Fact ● Unchangeable Fact ⬭ Situation

Pattern Language



○●● Fact / Negative Fact / Unchangeable Fact ⬭ Situation ⬭ Pattern Language

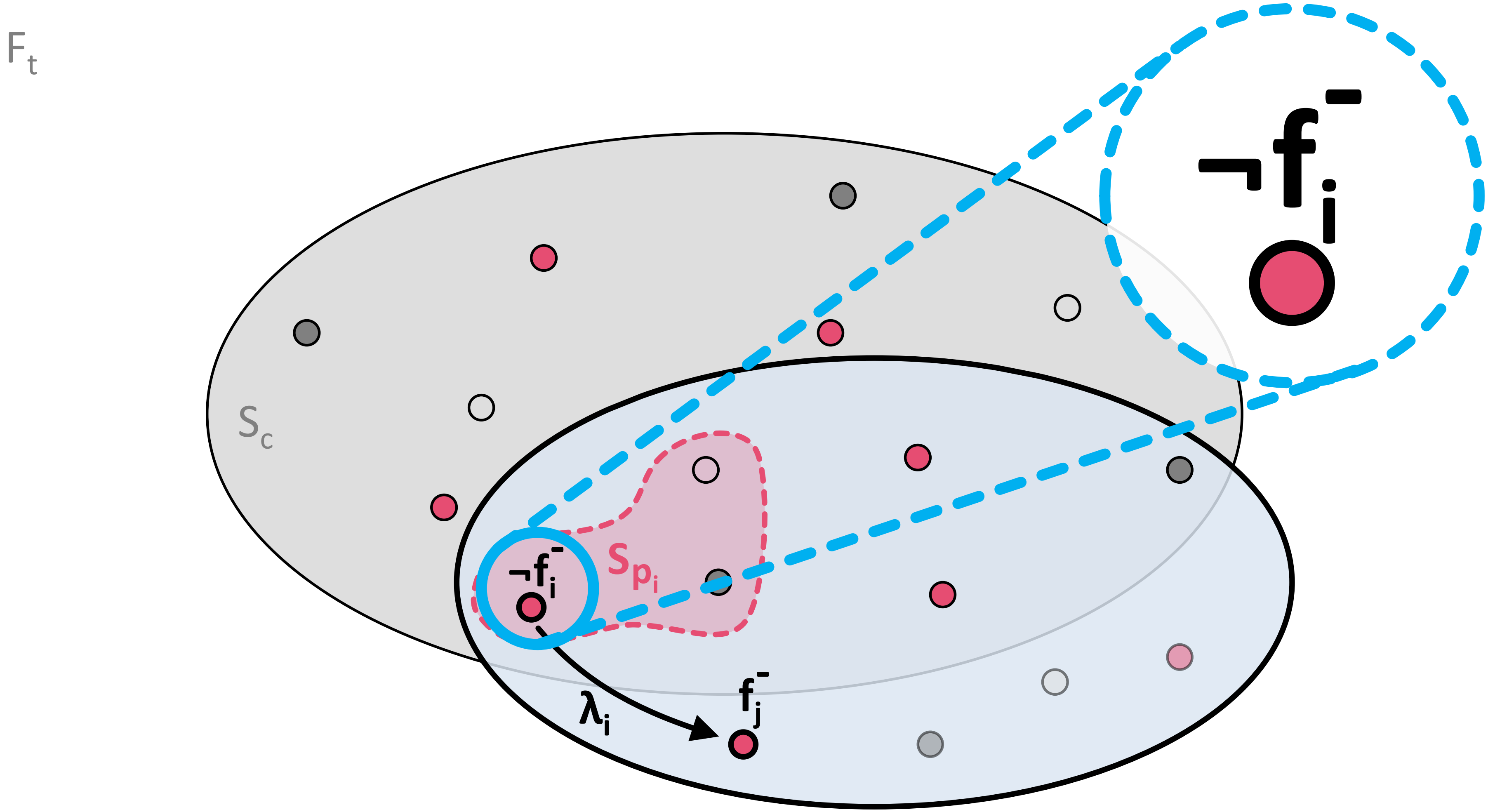
Pattern Start Situation



○●● Fact / Negative Fact / Unchangeable Fact ⬭ Situation ⬭ Pattern Language

▨ Pattern Start Situation

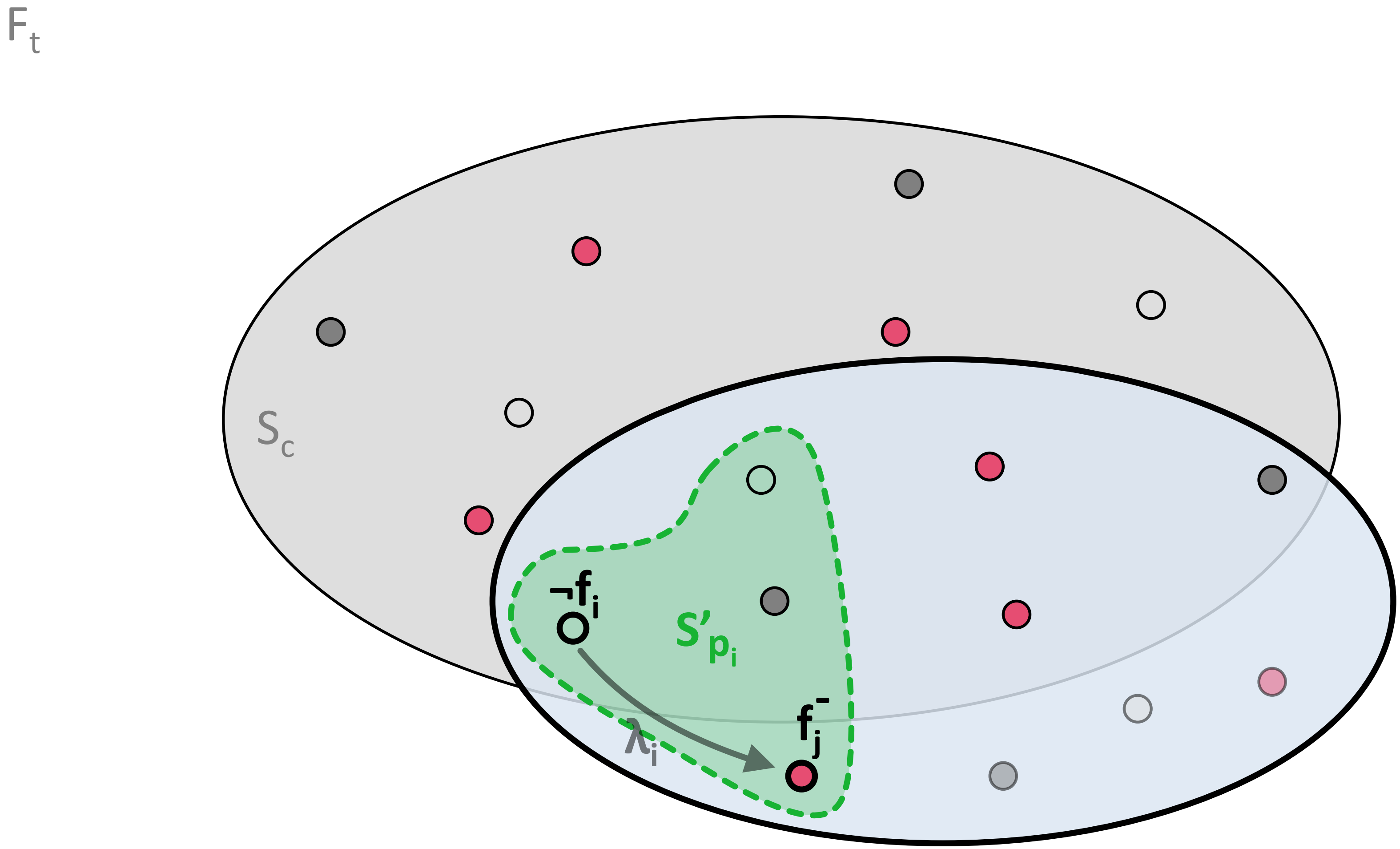
Applying Pattern Solution



○●● Fact / Negative Fact / Unchangeable Fact ⬭ Situation ⬭ Pattern Language

▨ Pattern Start Situation $\xrightarrow{\lambda}$ Applying Pattern Solution

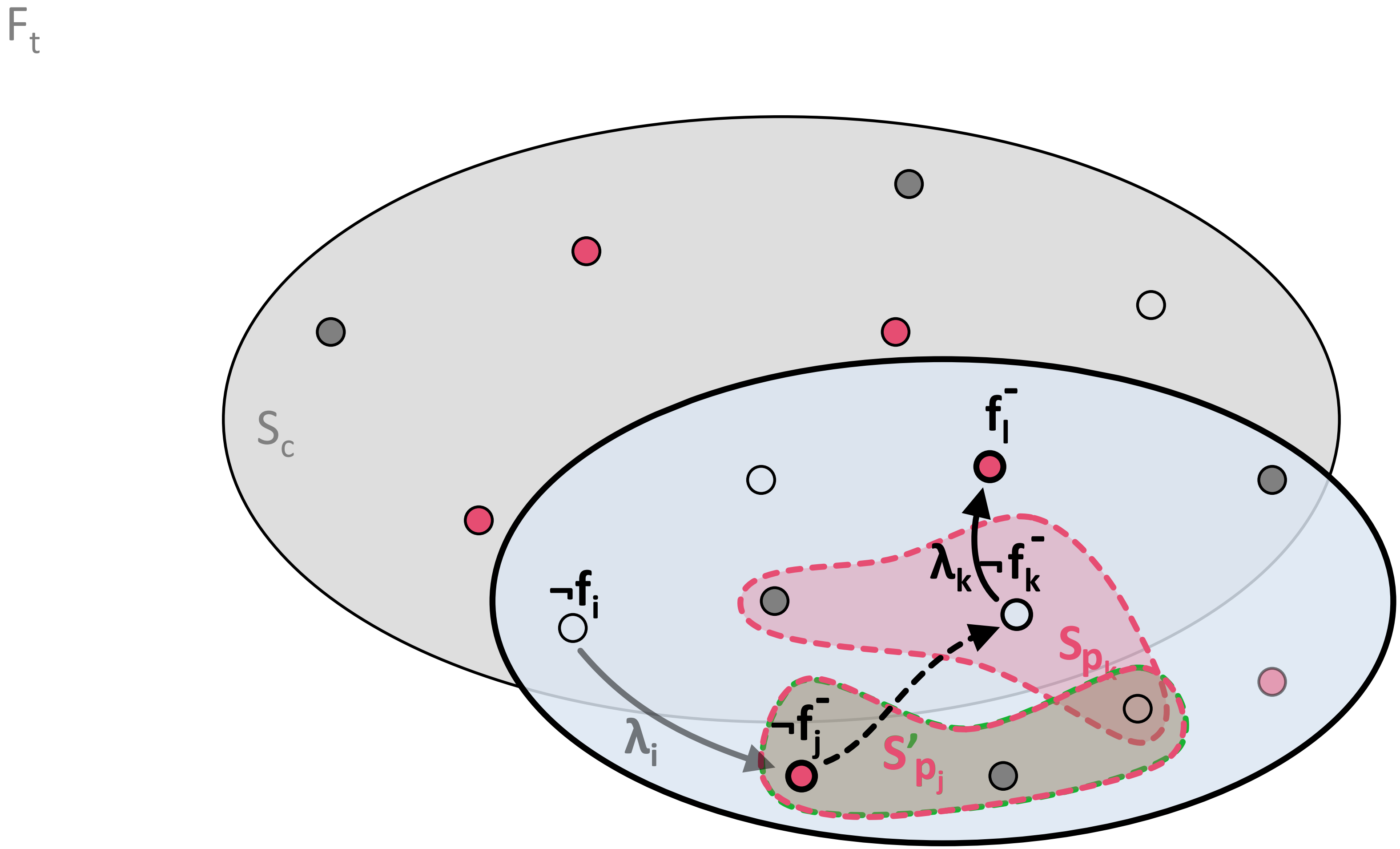
Pattern End Situation



○●● Fact / Negative Fact / Unchangeable Fact ⬭ Situation ⬭ Pattern Language

Pattern Start Situation → Applying Pattern Solution Pattern End Situation

Pattern Path



○●● Fact / Negative Fact / Unchangeable Fact ⬭ Situation ⬭ Pattern Language

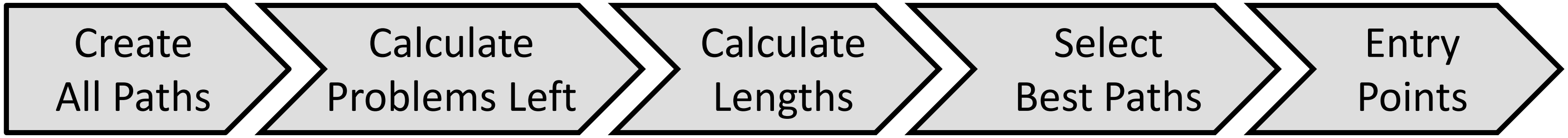
⬭ Pattern Start Situation ⇢ Applying Pattern Solution ⬭ Pattern End Situation

Entry Point Definition

Let E_{SC} define entry points into a suitable pattern language L as:

- **a finite set of patterns** out of the set of suitable patterns P_{SC}, where each of these patterns is
- the **first pattern in a pattern path** that
- **leaves the minimal amount of problems** after all patterns in the path have been applied and that
- **contains the least patterns**

Entry Point


Create All Paths

Calculate Problems Left

Calculate Lengths

Select Best Paths

Entry Points

No.	Path	Problems Left	Length
~~1~~	~~$p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$~~	~~3~~	~~4~~
~~2~~	~~$p_2 \rightarrow p_5 \rightarrow p_3 \rightarrow p_7 \rightarrow p_1$~~	~~2~~	~~5~~
~~3~~	~~$p_4 \rightarrow p_5 \rightarrow p_3$~~	~~3~~	~~3~~
~~4~~	~~$p_3 \rightarrow p_6$~~	~~4~~	~~2~~
5	$p_6 \rightarrow p_4 \rightarrow p_5$	2	3
…	…	…	…

Summary

- Finding entry point is not always easy
- First step is situation assessment via facts
- Algorithm can calculate entry points based on numbers of problems solved and path length
- Path length as proxy for complexity can be improved

Thank You!