Maintaining Constraint-based Configuration Systems: Challenges ahead

Florian Reinfrank, Gerald Ninaus, Franz Wotawa, Alexander Felfernig

{firstname.lastname}@ist.tugraz.at

Institute for Software Technology

Graz University of Technology

Inffeldgasse 16b, 8010 Graz, Austria

Outline

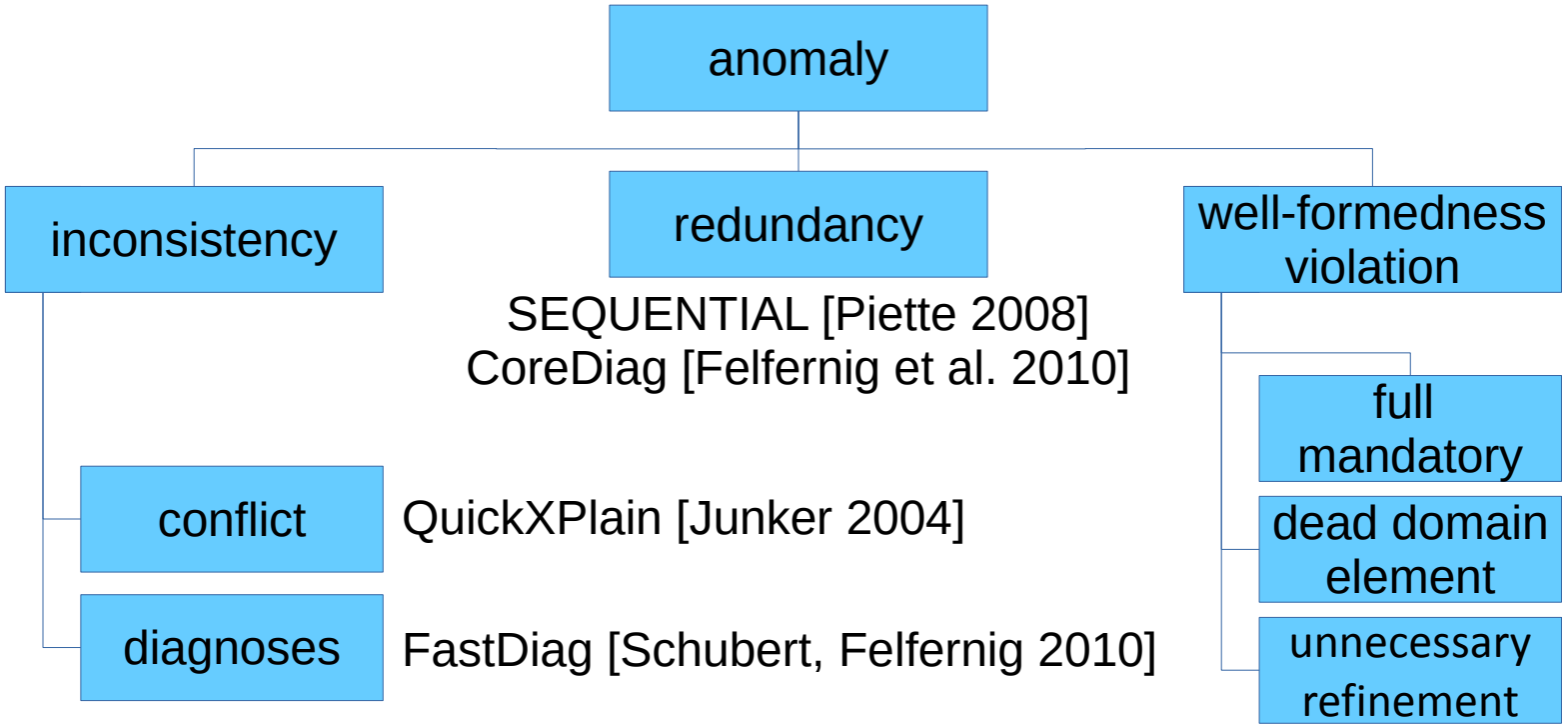
- Introduction
- Anomaly management
- Example
- Assignment-based anomaly management
- Summary

Introduction

- Constraint-based configuration knowledge bases are changing over time
- Maintenance is time consuming and error prone
  - Detect anomalies
  - Understand anomalies
  - Repair anomalies
- Different techniques to reduce the maintenance effort
  - Recommendation
  - **Anomaly management**
  - Simulation
  - KB evaluation

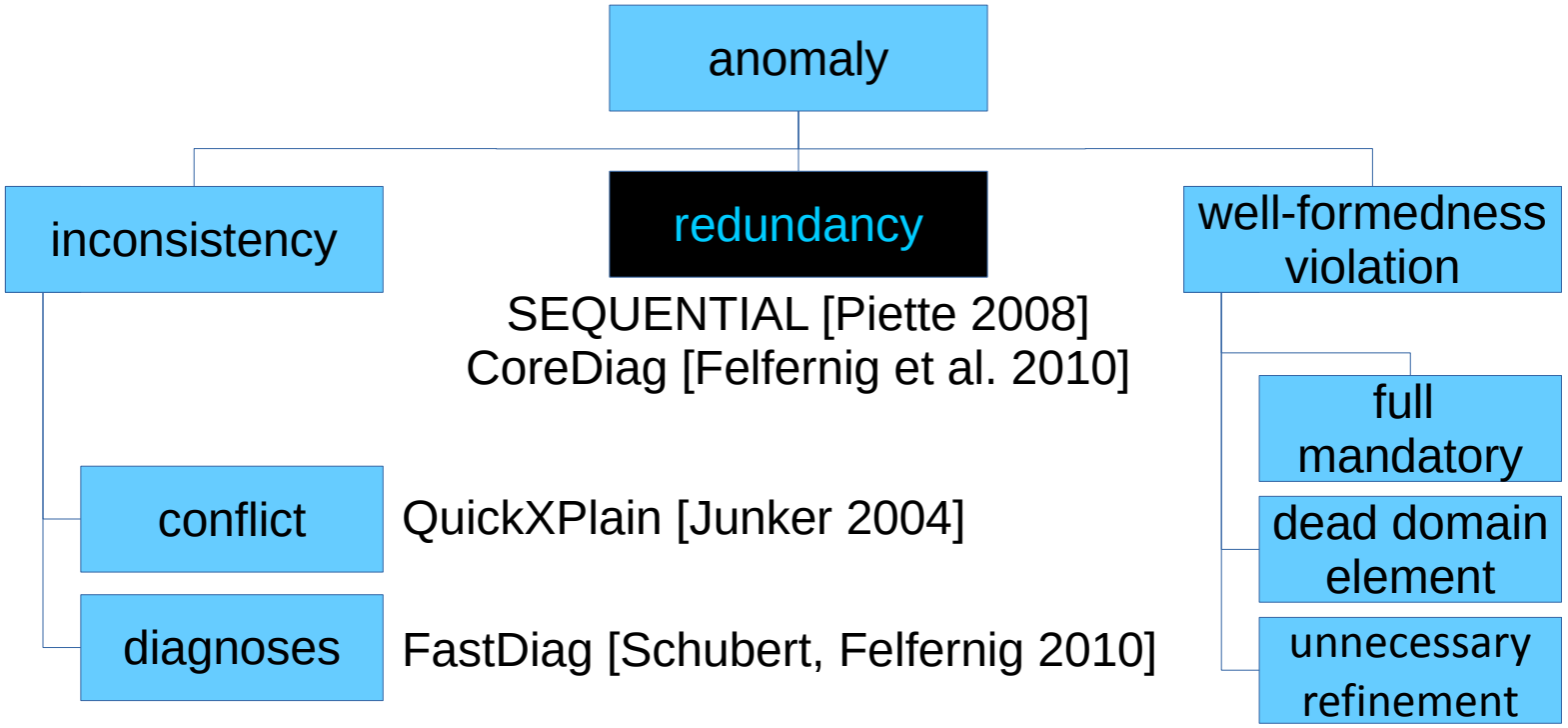
Anomaly management

- Anomalies are patterns in data that do not conform to a well defined notion of normal behaviour [Chandola et al. 2009]
- Management = detection + explanation + repair



Anomaly management

- Anomalies are patterns in data that do not conform to a well defined notion of normal behaviour [Chandola et al. 2009]
- Management = detection + explanation + repair



Anomaly management: redundancy

- A constraint c is redundant, iff the removal of a constraint c from KB leads to the same semantic, s.t. $KB/\{c\} \models c$

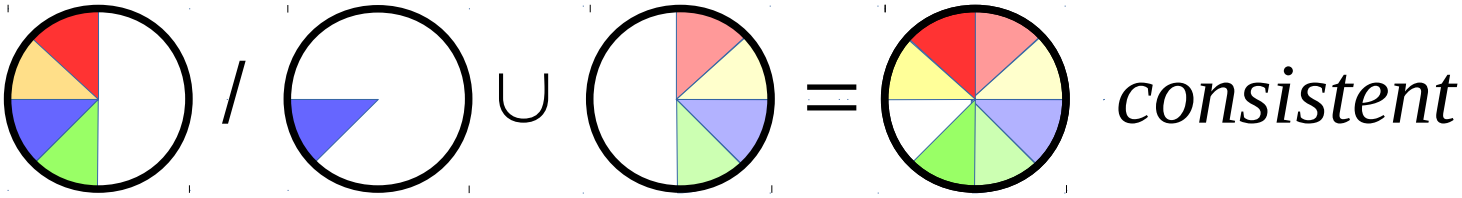
  $KB/\{c\} \cup \bar{KB} = inconsistent$ means that c is redundant
  $\bar{KB} = \{\neg c_0 \vee \neg c_1 \vee \ldots \vee \neg c_n\}$

Anomaly management: redundancy

- A constraint c is redundant, iff the removal of a constraint c from KB leads to the same semantic, s.t. $KB/\{c\} \models c$

  $KB/\{c\} \cup \bar{KB} = inconsistent$ means that c is redundant

  $\bar{KB} = \{\neg c_0 \vee \neg c_1 \vee \ldots \vee \neg c_n\}$

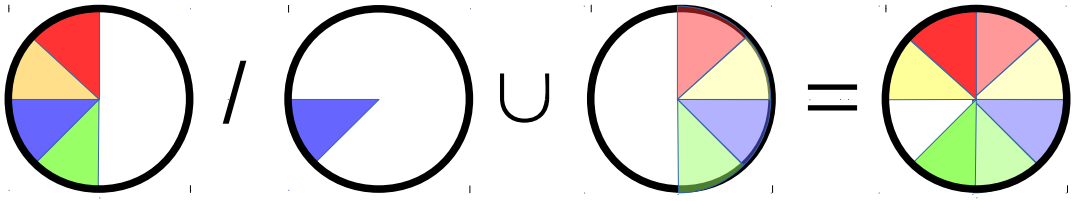


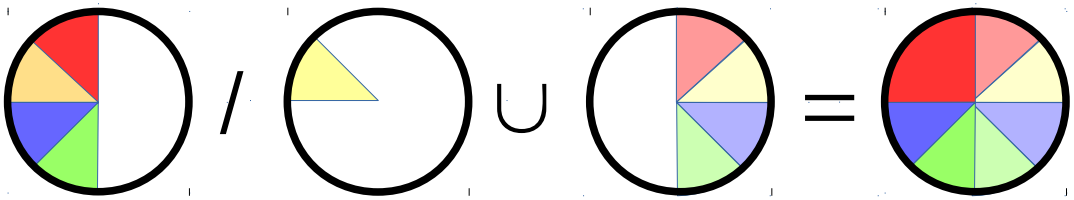
Anomaly management: redundancy

- A constraint c is redundant, iff the removal of a constraint c from KB leads to the same semantic, s.t. $KB/\{c\} \models c$

  $KB/\{c\} \cup \bar{KB} = inconsistent$ means that c is redundant

  $\bar{KB} = \{\neg c_0 \vee \neg c_1 \vee \ldots \vee \neg c_n\}$





Anomaly management: example for explaining redundancies

$$V = \{ price, cpuCores, usageScenario \}$$
$$D = \{ dom(price) = \{399, 599, 799, 999\},$$
$$dom(cpuCores) = \{2, 4\},$$
$$dom(usageScenario) = \{office, multimedia, gaming\}\}$$
$$C_{KB} =$$
$$\{c_0 = (\neg(usageScenario = office \wedge price = 999)),$$
$$c_1 = (usageScenario = multimedia \rightarrow (price < 999 \wedge cpuCores = 4)) \wedge usageScenario = office \rightarrow ((price < 599 \wedge cpuCores = 2) \vee price < 799) \wedge usageScenario = gaming \rightarrow cpuCores = 4)\}$$
$$C_R = \emptyset$$
$$C_P = \{ p_0 = (price = 399 \wedge cpuCores = 2) \vee$$
$$p_1 = (price = 599 \wedge cpuCores = 4) \vee$$
$$p_2 = (price = 799 \wedge cpuCores = 2) \vee$$
$$p_3 = (price = 999 \wedge cpuCores = 4)\}$$
$$C = C_{KB} \cup C_R \cup C_P$$
$$KB = V \cup D \cup C$$

Anomaly management: example for explaining redundancies

$$V = \{ price, cpuCores, usageScenario \}$$
$$D = \{ dom(price) = \{399, 599, 799, 999\},$$
$$dom(cpuCores) = \{2, 4\},$$
$$dom(usageScenario) = \{office, multimedia, gaming\}\}$$
$$C_{KB} =$$
$$\{ c_0 = (\neg(usageScenario = office \wedge price = 999)),$$
$$c_1 = (usageScenario = multimedia \rightarrow (price < 999 \wedge cpuCores = 4)) \wedge usageScenario = office \rightarrow ((price < 599 \wedge cpuCores = 2) \vee price < 799) \wedge usageScenario = gaming \rightarrow cpuCores = 4)\}$$
$$C_R = \emptyset$$
$$C_P = \{ p_0 = (price = 399 \wedge cpuCores = 2) \vee$$
$$p_1 = (price = 599 \wedge cpuCores = 4) \vee$$
$$p_2 = (price = 799 \wedge cpuCores = 2) \vee$$
$$p_3 = (price = 999 \wedge cpuCores = 4)\}$$
$$C = C_{KB} \cup C_R \cup C_P$$
$$KB = V \cup D \cup C$$

Anomaly management: example for detecting redundancies

$$V = \{ price, cpuCores, usageScenario \}$$
$$D = \{ dom(price) = \{399, 599, 799, 999\},$$
$$dom(cpuCores) = \{2, 4\},$$
$$dom(usageScenario) = \{office, multimedia, gaming\}\}$$
$$C_{KB} = \{ c_1 = usageScenario = office \rightarrow (price < 599 \wedge cpuCores = 2)$$
$$c_2 = usageScenario = multimedia \rightarrow ((price < 999 \wedge cpuCores = 4) \vee price < 799)$$
$$c_3 = usageScenario = gaming \rightarrow cpuCores = 4 \}$$
$$C_R = \emptyset$$
$$C_P = \{ p_0 = (price = 399 \wedge cpuCores = 2) \vee$$
$$p_1 = (price = 599 \wedge cpuCores = 4) \vee$$
$$p_2 = (price = 799 \wedge cpuCores = 2) \vee$$
$$p_3 = (price = 999 \wedge cpuCores = 4)\}$$
$$C = C_{KB} \cup C_R \cup C_P$$
$$KB = V \cup D \cup C$$

Anomaly management: example for detecting redundancies

$$V = \{ price, cpuCores, usageScenario \}$$
$$D = \{ dom(price) = \{399, 599, 799, 999\},$$
$$dom(cpuCores) = \{2, 4\},$$
$$dom(usageScenario) = \{office, multimedia, gaming\}\}$$
$$C_{KB} = \{ c_1 = usageScenario = office \rightarrow (price < 599 \wedge cpuCores = 2)$$
$$c_2 = usageScenario = multimedia \rightarrow ((price < 999 \wedge cpuCores = 4) \vee price < 799)$$
$$c_3 = usageScenario = gaming \rightarrow cpuCores = 4 \}$$
$$C_R = \emptyset$$
$$C_P = \{ p_0 = (price = 399 \wedge cpuCores = 2) \vee$$
$$p_1 = (price = 599 \wedge cpuCores = 4) \vee$$
$$p_2 = (price = 799 \wedge cpuCores = 2) \vee$$
$$p_3 = (price = 999 \wedge cpuCores = 4)\}$$
$$C = C_{KB} \cup C_R \cup C_P$$
$$KB = V \cup D \cup C$$

Assignment-based anomaly management

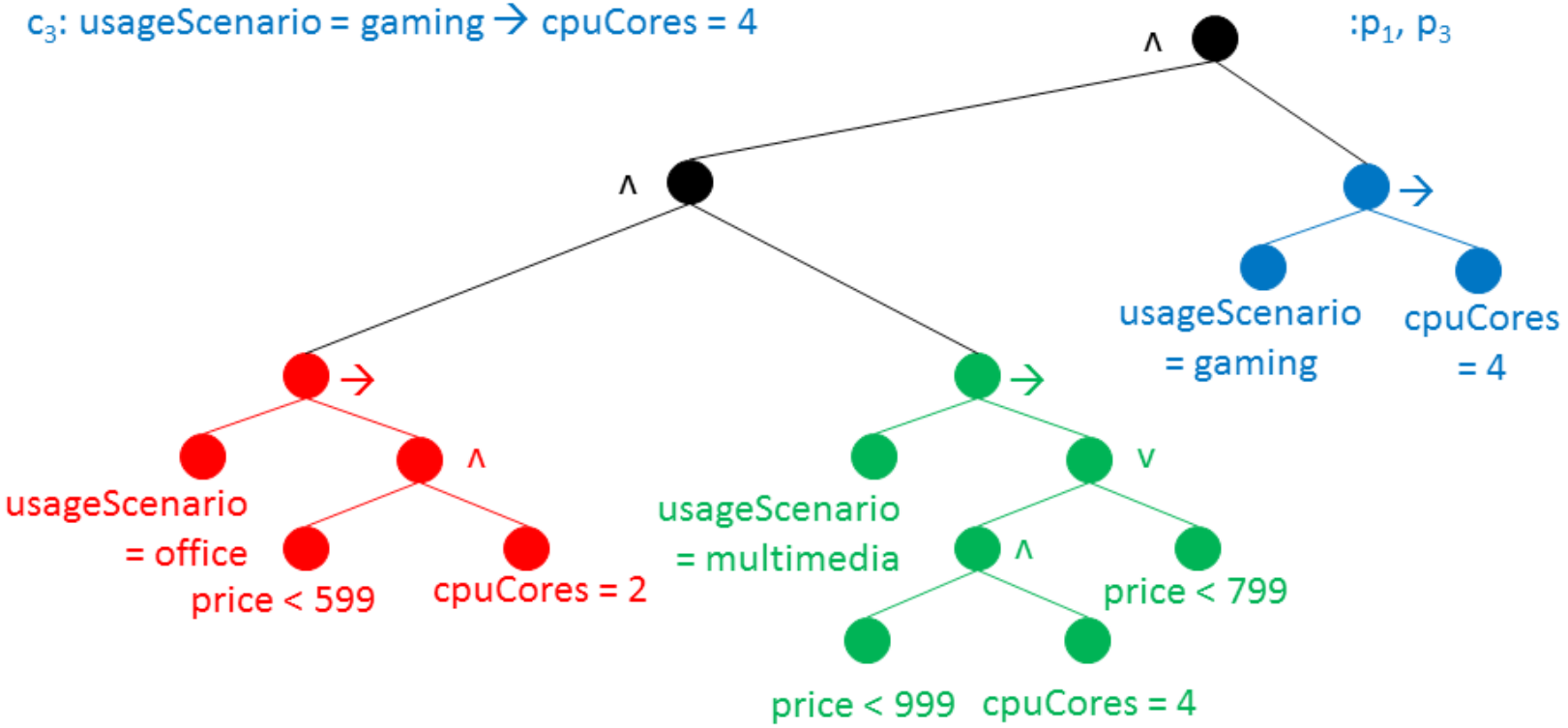
- A constraint consists of
  - A set of assignments
  - Relations between assignments
- An assignment consists of
  - One variable v
  - One relation (e.g. $<, \leq, =, \neq, \geq, >$)
  - One element of the domain $d \in dom(v)$

Assignment-based redundancy detection

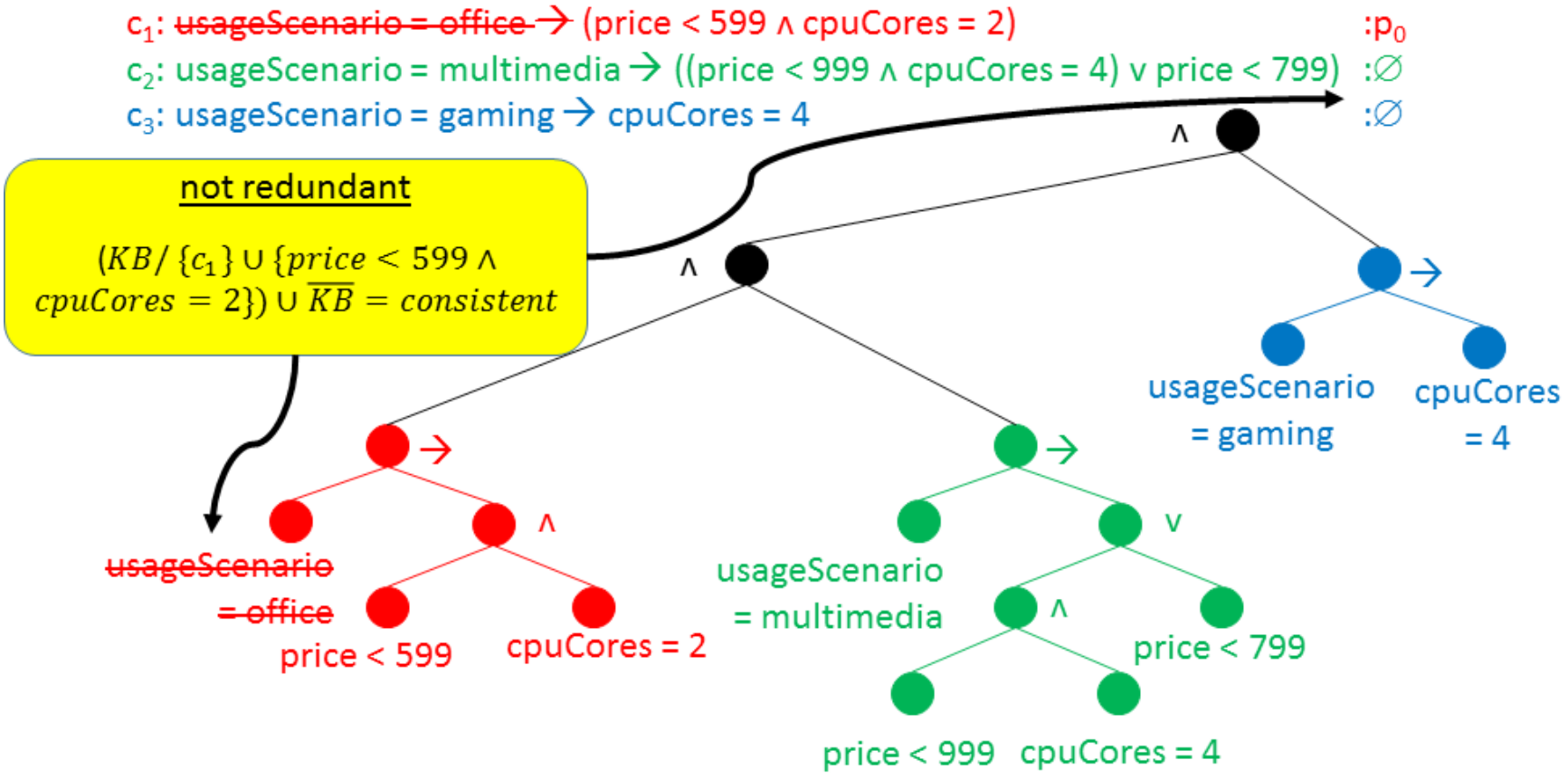
c_1: usageScenario = office → (price < 599 ∧ cpuCores = 2) :p_0

c_2: usageScenario = multimedia → ((price < 999 ∧ cpuCores = 4) v price < 799) :p_0, p_1

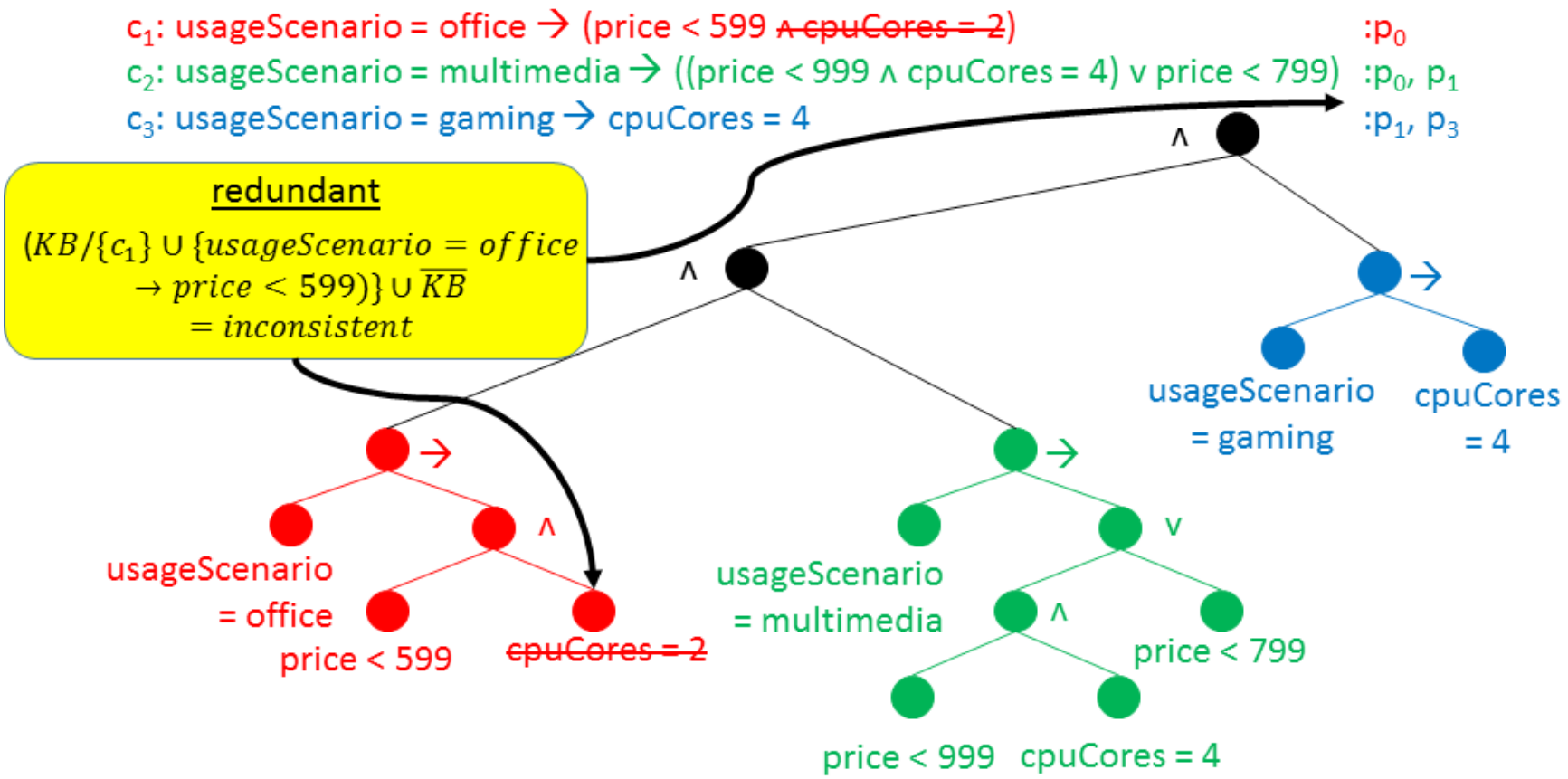
c_3: usageScenario = gaming → cpuCores = 4 :p_1, p_3



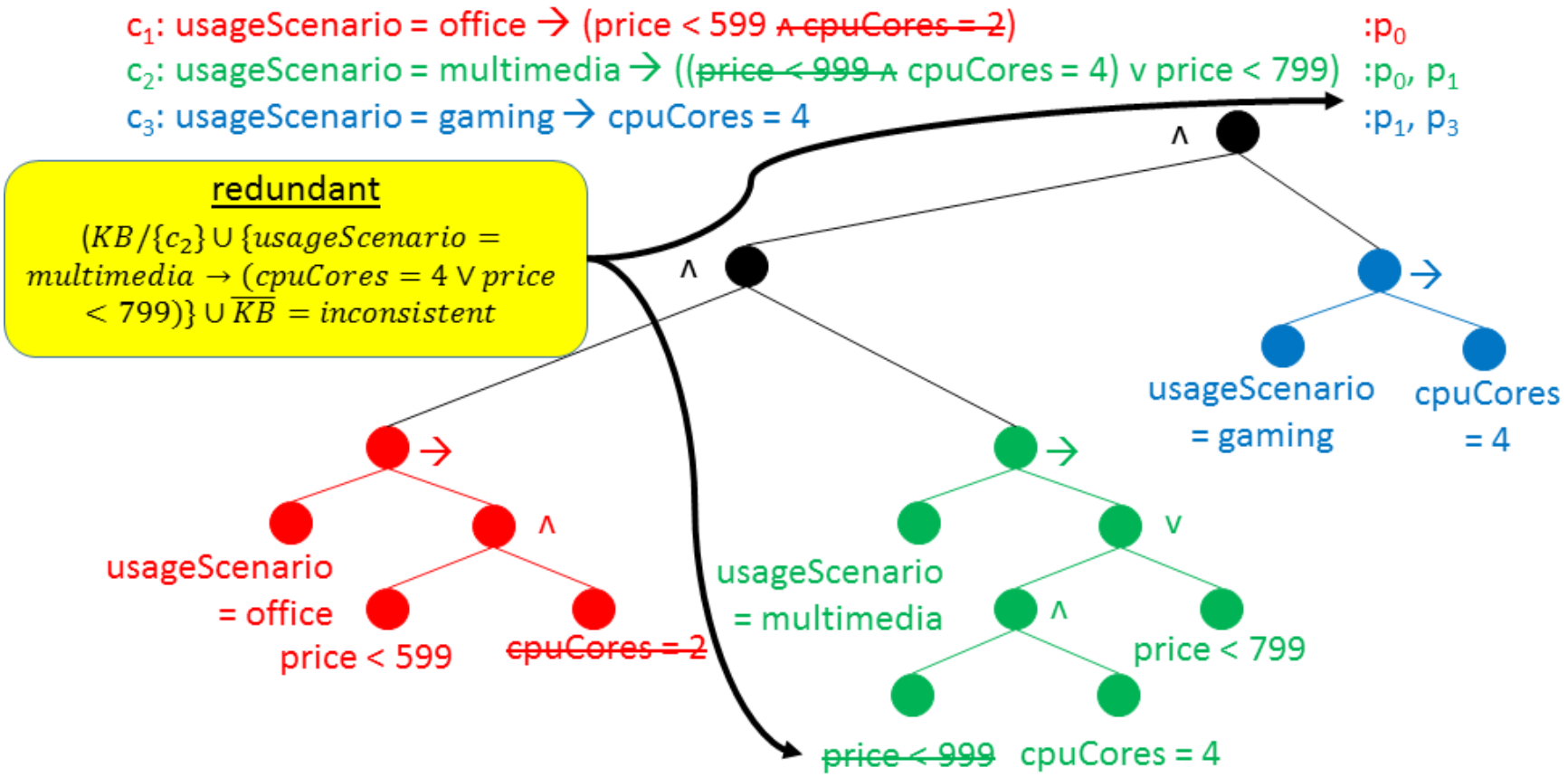
Assignment-based redundancy detection



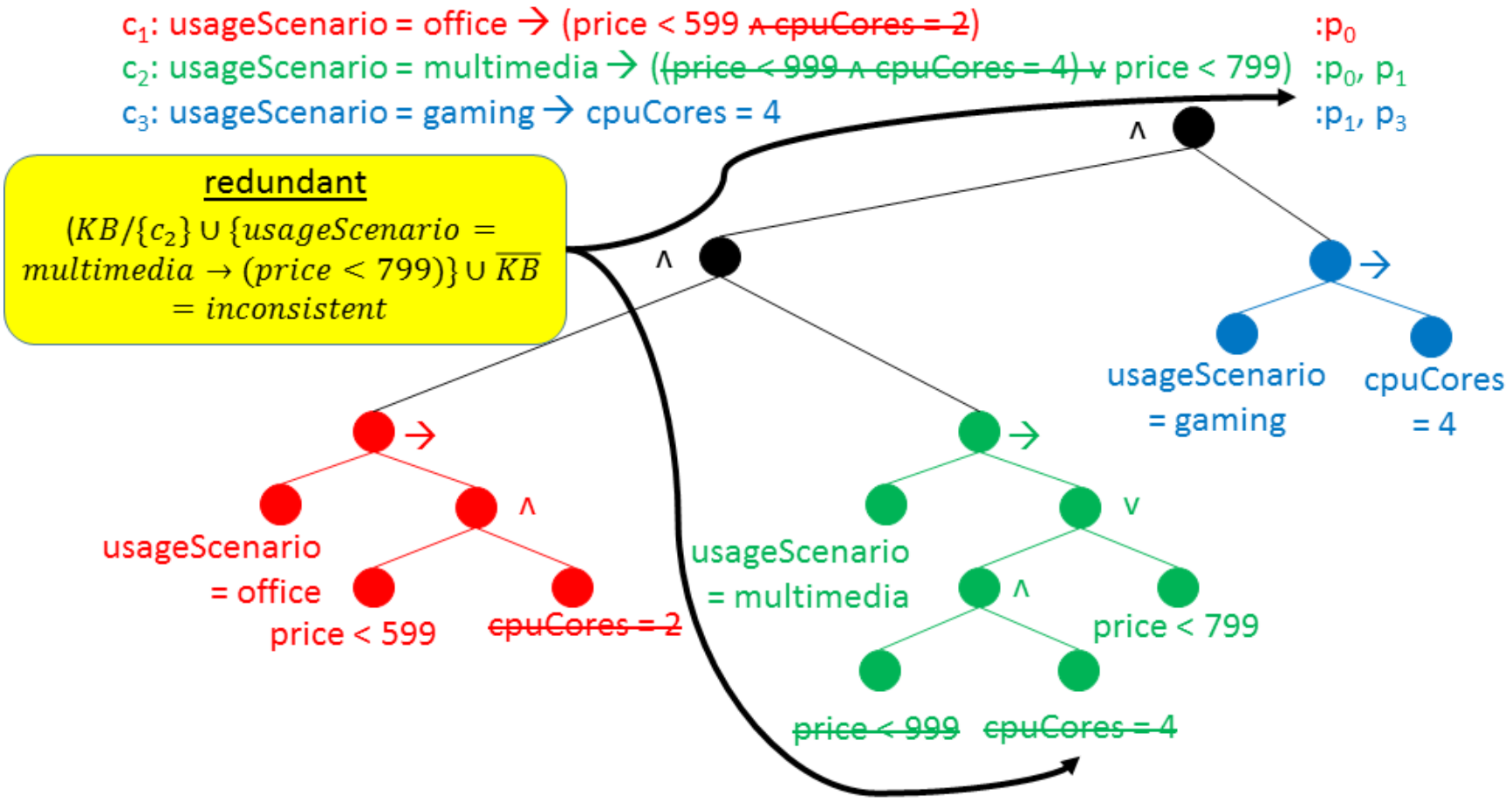
Assignment-based redundancy detection



Assignment-based redundancy detection



Assignment-based redundancy detection



Summary

- Focus on assignments of constraints instead of constraints
- Advantages
  - Detecting more redundancies
  - Getting better explanations
  - Faster consistency checks
- Further research
  - Applicability for well-formedness violations
  - Applicability for conflicts and diagnoses

References

- [Chandola et al. 2009] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Comput. Surv., 41:15:1–15:58. 2009.
- [Felfernig et al. 2010] Alexander Felfernig, Christoph Zehentner, and Paul Blazek. Corediag: Eliminating redundancy in constraint sets. In Martin Sachenbacher, Oskar Dressler, and Michael Hofbaur, editors, DX 2011. 22nd International Workshop on Principles of Diagnosis, pages 219 – 224, Murnau, GER, 2010.
- [Junker 2004] Ulrich Junker. Quickxplain: preferred explanations and relaxations for over-constrained problems. In Proceedings of the 19th national conference on Artifical intelligence, AAAI'04, pages 167–172. AAAI Press, 2004.
- [Piette 2008] Cedric Piette. Let the solver deal with redundancy. In Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence - Volume 01, pages 67–73, Washington, DC, USA, 2008. IEEE Computer Society.
- [Schubert, Felfernig 2010] Monika Schubert and Alexander Felfernig. A Diagnosis Algorithm for Inconsistent Constraint Sets. In Proceedings of the 21st International Workshop on the Principles of Diagnosis. 2010.

Assignment-based redundancy detection