

Conformance Checking

Andrea Polini

Process Mining

MSc in Computer Science (LM-18)

University of Camerino

Summary



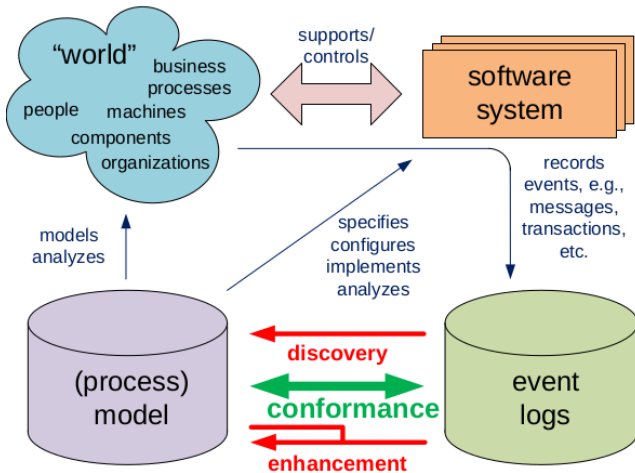
1 Introduction

Summary



1 Introduction

Conformance Checking

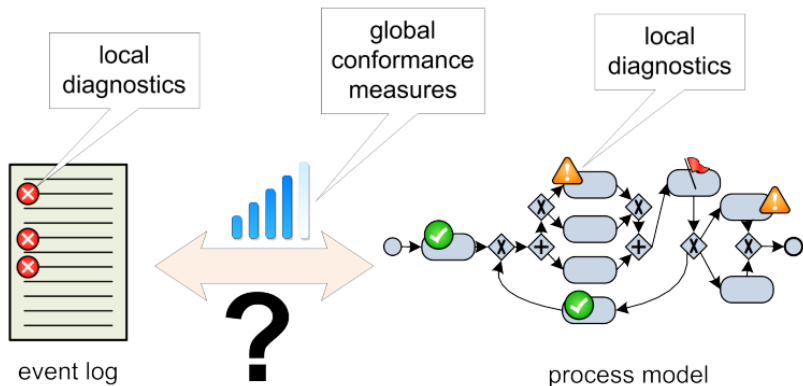


Why?

Conformance checking **relates events in the event log to activities in the process model and compares both**. The goal is to **find commonalities and discrepancies** between the modeled behavior and the observed behavior. Conformance checking is relevant for **business alignment and auditing**:

- ▶ **find** undesirable deviations suggesting fraud or inefficiencies
- ▶ **measuring** the performance of process discovery algorithms
- ▶ **repair** models that are not aligned well with reality

Using Conformance Checking

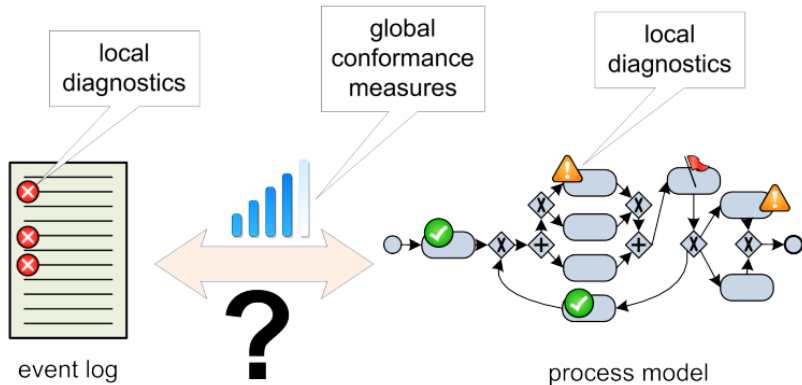


- **global conformance measures** – e.g. 85% of the cases in the event log can be replayed by the model
- **local diagnostics** – e.g. activity x was executed 15 times although this was not allowed according to the model

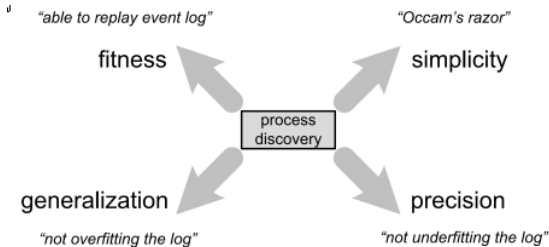
Results Interpretation

The interpretation of non-conformance depends on the purpose of the model:

- descriptive
- normative



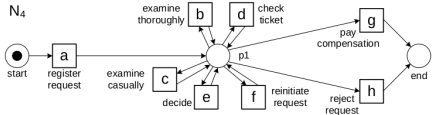
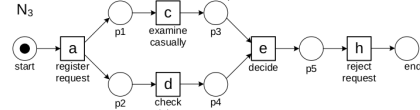
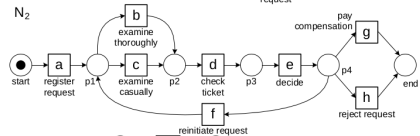
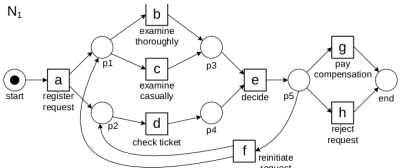
Quality criteria



Fitness function

- ▶ A naïve approach towards conformance checking would be to simply count the fraction of cases that can be "parsed completely"
 - ▶ $N_1 : 1, N_2 : 0.6815, N_3 : 0.4543, N_4 : 1$

Four models and one log



frequency reference trace

455	σ_1	$\langle a, c, d, e, h \rangle$
191	σ_2	$\langle a, b, d, e, g \rangle$
177	σ_3	$\langle a, d, c, e, h \rangle$
144	σ_4	$\langle a, b, d, e, h \rangle$
111	σ_5	$\langle a, c, d, e, g \rangle$
82	σ_6	$\langle a, d, c, e, g \rangle$
56	σ_7	$\langle a, d, b, e, h \rangle$
47	σ_8	$\langle a, c, d, e, f, d, b, e, h \rangle$
38	σ_9	$\langle a, d, b, e, g \rangle$
33	σ_{10}	$\langle a, c, d, e, f, b, d, e, h \rangle$
14	σ_{11}	$\langle a, c, d, e, f, b, d, e, g \rangle$
11	σ_{12}	$\langle a, c, d, e, f, d, b, e, g \rangle$
9	σ_{13}	$\langle a, d, c, e, f, c, d, e, h \rangle$
8	σ_{14}	$\langle a, d, c, e, f, d, b, e, h \rangle$
5	σ_{15}	$\langle a, d, c, e, f, b, d, e, g \rangle$
3	σ_{16}	$\langle a, c, d, e, f, b, d, e, f, d, b, e, g \rangle$
2	σ_{17}	$\langle a, d, c, e, f, d, b, e, g \rangle$
2	σ_{18}	$\langle a, d, c, e, f, b, d, e, f, b, d, e, g \rangle$
1	σ_{19}	$\langle a, d, c, e, f, d, b, e, f, b, d, e, h \rangle$
1	σ_{20}	$\langle a, d, b, e, f, b, d, e, f, d, b, e, g \rangle$
1	σ_{21}	$\langle a, d, c, e, f, d, b, e, f, c, d, e, f, d, b, e, g \rangle$

Token Based Metrics

- The fitness metric is generally defined at the **level of events**
 - Let's continue to replay a trace **adding (and counting) tokens** to enable blocked transitions, and also **counting the remaining tokens** at the end of the execution

Token Based Metrics

- The fitness metric is generally defined at the **level of events**
 - Let's continue to replay a trace **adding (and counting) tokens** to enable blocked transitions, and also **counting the remaining tokens** at the end of the execution

Let's consider model N_1 , the following four counters,

- ▶ p : number of produced tokens
- ▶ c : number of consumed tokens
- ▶ m : number of added tokens
- ▶ r : number of remaining tokens,

and let's replay trace $\sigma_3 = \langle a, d, c, e, h \rangle$

Token Based Metrics

- The fitness metric is generally defined at the **level of events**
 - Let's continue to replay a trace **adding (and counting) tokens** to enable blocked transitions, and also **counting the remaining tokens** at the end of the execution

Let's consider model N_1 , the following four counters,

- ▶ p : number of produced tokens
- ▶ c : number of consumed tokens
- ▶ m : number of added tokens
- ▶ r : number of remaining tokens,

and let's replay trace $\sigma_3 = \langle a, d, c, e, h \rangle$

Now let's replay the trace on N_2

Computing fitness at trace level

$$\text{fitness}(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)$$

- What about replaying trace $\sigma_2 = \langle a, b, d, e, g \rangle$ on N_3 ?
- When a trace contains labels for which there is no corresponding transition the trace has to be projected on the available transitions

Computing fitness at trace level

$$\text{fitness}(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)$$

- What about replaying trace $\sigma_2 = \langle a, b, d, e, g \rangle$ on N_3 ?
- When a trace contains labels for which there is no corresponding transition the trace has to be projected on the available transitions

Computing fitness at trace level

$$\text{fitness}(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)$$

- What about replaying trace $\sigma_2 = \langle a, b, d, e, g \rangle$ on N_3 ?
- When a trace contains labels for which there is no corresponding transition the trace has to be projected on the available transitions

Computing fitness at trace level

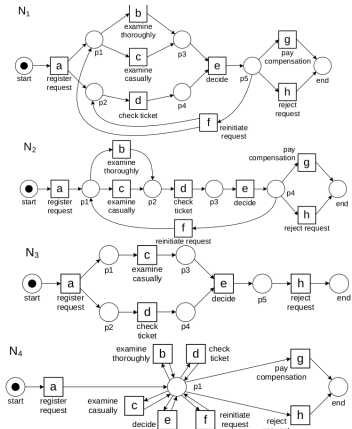
$$\text{fitness}(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)$$

- What about replaying trace $\sigma_2 = \langle a, b, d, e, g \rangle$ on N_3 ?
- When a trace contains labels for which there is no corresponding transition the trace has to be projected on the available transitions

$$\sigma_2 = \langle a, b, d, e, g \rangle \rightarrow \sigma'_2 = \langle a, d, e \rangle$$

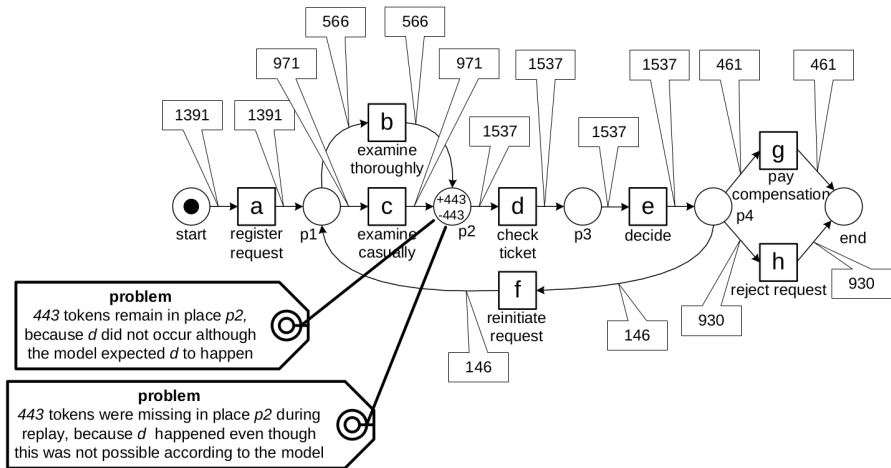
Computing fitness at the log level

$$fitness(L, N) = \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)$$

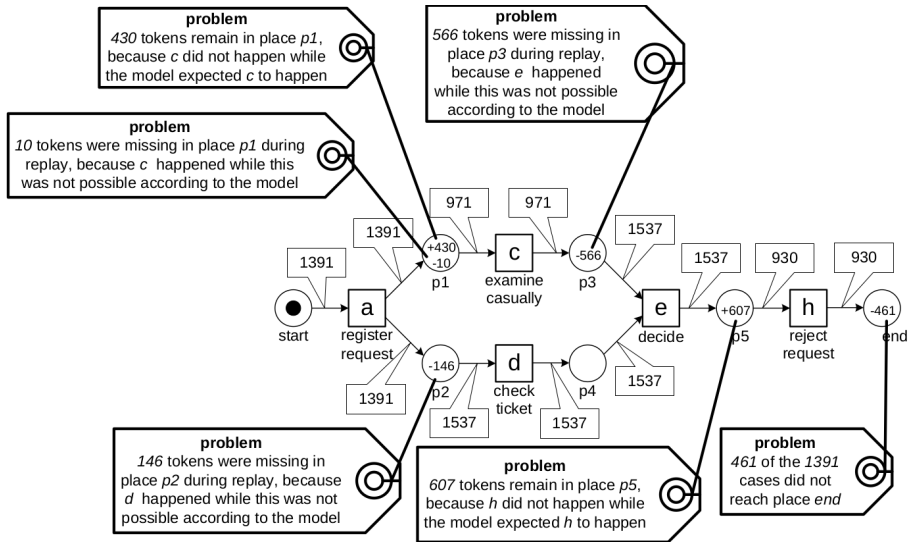


- $fitness(L_{full}, N_1) = 1$
- $fitness(L_{full}, N_2) = 0.9504$
- $fitness(L_{full}, N_3) = 0.8797$
- $fitness(L_{full}, N_4) = 1$

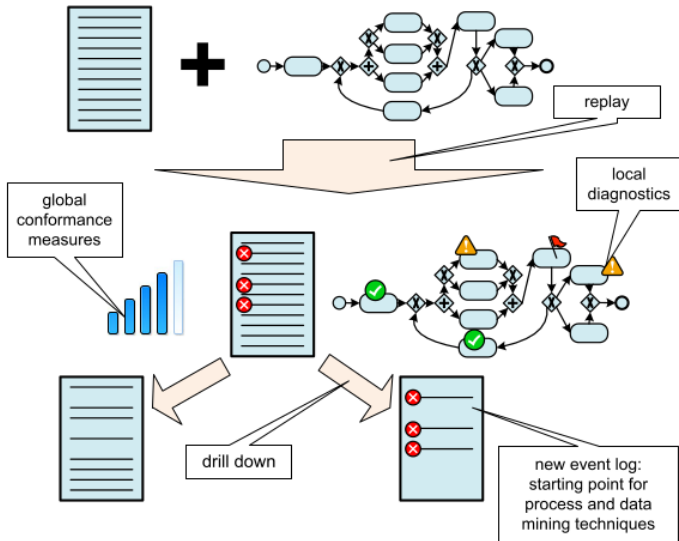
Diagnostics (N_2)



Diagnostics (N_3)



Further analysis



Alignments

Using token-based replay we can differentiate between fitting and non-fitting cases

Drawbacks

- ▶ Fitness values tend to be too high for extremely problematic event logs
- ▶ If there are many deviations, the Petri net gets “flooded with tokens” and subsequently allows for any behavior
- ▶ The approach is also **Petri-net specific** and can only be applied to other representations after conversion
- ▶ If a case does not fit, the approach does not create a corresponding path through the model.

Alignments were introduced to overcome these limitations

Alignments

Using token-based replay we can differentiate between fitting and non-fitting cases

Drawbacks

- ▶ Fitness values tend to be too high for extremely problematic event logs
- ▶ If there are many deviations, the Petri net gets “flooded with tokens” and subsequently allows for any behavior
- ▶ The approach is also **Petri-net specific** and can only be applied to other representations after conversion
- ▶ If a case does not fit, the approach does not create a corresponding path through the model.

Alignments were introduced to overcome these limitations

A so-called **optimal alignment** is a best match given a trace and a model. Given $\sigma = \langle a, d, b, e, h \rangle$ and N_1 there is precisely one optimal alignment

$$\gamma_{2a} = \begin{array}{|c|c|c|c|c|} \hline a & d & b & e & h \\ \hline a & d & b & e & h \\ \hline \end{array}$$

Given σ and N_2 there are multiple optimal alignments:

$$\gamma_{2a} = \begin{array}{|c|c|c|c|c|c|} \hline a & \gg & d & b & e & h \\ \hline a & b & d & \gg & e & h \\ \hline \end{array} \quad \gamma_{2b} = \begin{array}{|c|c|c|c|c|c|} \hline a & \gg & d & b & e & h \\ \hline a & c & d & \gg & e & h \\ \hline \end{array}$$

$$\gamma_{2c} = \begin{array}{|c|c|c|c|c|c|} \hline a & d & b & \gg & e & h \\ \hline a & \gg & b & d & e & h \\ \hline \end{array}$$

Alignments

A so-called **optimal alignment** is a best match given a trace and a model. Given $\sigma = \langle a, d, b, e, h \rangle$ and N_1 there is precisely one optimal alignment

$$\gamma_{2a} = \begin{array}{|c|c|c|c|c|} \hline a & d & b & e & h \\ \hline a & d & b & e & h \\ \hline \end{array}$$

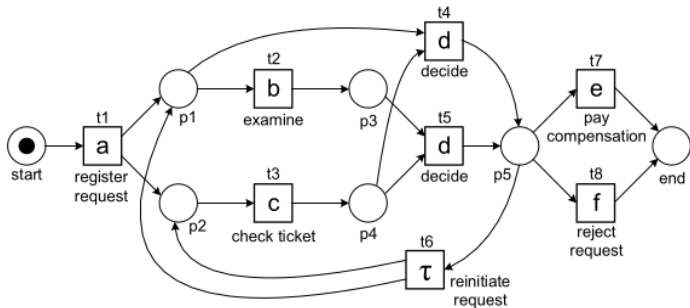
Given σ and N_2 there are multiple optimal alignments:

$$\gamma_{2a} = \begin{array}{|c|c|c|c|c|} \hline a & \gg & d & b & e & h \\ \hline a & b & d & \gg & e & h \\ \hline \end{array} \quad \gamma_{2b} = \begin{array}{|c|c|c|c|c|} \hline a & \gg & d & b & e & h \\ \hline a & c & d & \gg & e & h \\ \hline \end{array}$$

$$\gamma_{2c} = \begin{array}{|c|c|c|c|c|} \hline a & d & b & \gg & e & h \\ \hline a & \gg & b & d & e & h \\ \hline \end{array}$$

Token-based conformance checking becomes more complicated when there are duplicate and silent activities, e.g., transitions with a τ label or two transitions with the same label. **Alignments can be defined for any process notation having duplicate and silent activities**

Silent and duplicated transition (N5)



$$\sigma = \langle a, c, d, b, c, d, c, d, c, b, d, f \rangle$$

Which is the corresponding alignment?

Move and Alignments

Move

A **move** is a pair $(x, (y, t))$ where the first element refers to the log and the second element refers to the model.

$(x, (y, t))$ is a legal move if one of the following four cases holds:

- ▶ $x = y$ and y is the visible label of transition t (**synchronous move**)
- ▶ $x = \ggg$ and y is the visible label of transition t (**visible model move**)
- ▶ $x = \ggg, y = \tau$ and transition t is silent (**invisible model move**)
- ▶ $x \neq \ggg$ and $(y, t) = \ggg$ (**log move**)

Move and Alignments

Move

A **move** is a pair $(x, (y, t))$ where the first element refers to the log and the second element refers to the model.

$(x, (y, t))$ is a legal move if one of the following four cases holds:

- ▶ $x = y$ and y is the visible label of transition t (**synchronous move**)
- ▶ $x = \ggg$ and y is the visible label of transition t (**visible model move**)
- ▶ $x = \ggg, y = \tau$ and transition t is silent (**invisible model move**)
- ▶ $x \neq \ggg$ and $(y, t) = \ggg$ (**log move**)

Alignment

An **alignment** is a sequence of legal moves such that after removing all \ggg symbols, the top row corresponds to the trace in the log, and the bottom row corresponds to a firing sequence starting in some initial state of the process model and ending in some final state.

Selecting alignments

Given a log trace and a process model, there may be many (if not infinitely many) alignments. For $\sigma_2 = \langle a, b, d, f \rangle$ and N_5 , there are additional alignments like:

$$\begin{array}{l}
 \gamma_{5,2a} = \begin{array}{|c|c|c|c|c|} \hline a & b & \gg & d & f \\ \hline a & b & c & d & f \\ \hline t_1 & t_2 & t_3 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2c} = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg & \gg & \gg \\ \hline \gg & \gg & \gg & \gg & a & b & c & d & f \\ \hline t_1 & t_2 & t_3 & t_5 & t_8 \\ \hline \end{array} \\
 \gamma_{5,2b} = \begin{array}{|c|c|c|c|c|} \hline a & \gg & b & d & f \\ \hline a & c & b & d & f \\ \hline t_1 & t_3 & t_2 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2d} = \begin{array}{|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg \\ \hline a & \gg & \gg & \gg & c & d & e \\ \hline t_1 & & & & t_3 & t_4 & t_7 \\ \hline \end{array}
 \end{array}$$

Selecting alignments

Given a log trace and a process model, there may be many (if not infinitely many) alignments. For $\sigma_2 = \langle a, b, d, f \rangle$ and N_5 , there are additional alignments like:

$$\begin{array}{l} \gamma_{5,2a} = \begin{array}{|c|c|c|c|c|} \hline a & b & \gg & d & f \\ \hline a & b & c & d & f \\ \hline t_1 & t_2 & t_3 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2c} = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg & \gg & \gg \\ \hline \gg & \gg & \gg & \gg & a & b & c & d & f \\ \hline & & & & t_1 & t_2 & t_3 & t_5 & t_8 \\ \hline \end{array} \\ \gamma_{5,2b} = \begin{array}{|c|c|c|c|c|} \hline a & \gg & b & d & f \\ \hline a & c & b & d & f \\ \hline t_1 & t_3 & t_2 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2d} = \begin{array}{|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg \\ \hline a & \gg & \gg & \gg & c & d & e \\ \hline t_1 & & & & t_3 & t_4 & t_7 \\ \hline \end{array} \end{array}$$

To select the most appropriate alignment, we associate costs to undesirable moves and select an alignment with the lowest total costs. Cost function δ as- signs costs to legal moves.

- ▶ Moves where log and model agree have no costs, i.e., $\delta(x, (y, t)) = 0$ for **synchronous moves** (with $x = y$)
- ▶ Moves in model only have no costs if the transition is invisible, i.e., $\delta(\gg, (\tau, t)) = 0$ for **invisible model moves**
- ▶ $\delta(\gg, (y, t)) > 0$ is the cost when the model makes an "y move" without a corresponding move of the log (**visible model move**)
- ▶ $\delta(x, \gg) > 0$ is the cost for an "x move" in just the log (**log move**)

Selecting alignments

Given a log trace and a process model, there may be many (if not infinitely many) alignments. For $\sigma_2 = \langle a, b, d, f \rangle$ and N_5 , there are additional alignments like:

$$\gamma_{5,2a} = \begin{array}{|c|c|c|c|c|} \hline a & b & \gg & d & f \\ \hline a & b & c & d & f \\ \hline t_1 & t_2 & t_3 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2c} = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg & \gg & \gg \\ \hline \gg & \gg & \gg & \gg & a & b & c & d & f \\ \hline t_1 & t_2 & t_3 & t_5 & t_8 & & & & \\ \hline \end{array}$$

$$\gamma_{5,2b} = \begin{array}{|c|c|c|c|c|} \hline a & \gg & b & d & f \\ \hline a & c & b & d & f \\ \hline t_1 & t_3 & t_2 & t_5 & t_8 \\ \hline \end{array} \quad \gamma_{5,2d} = \begin{array}{|c|c|c|c|c|c|c|} \hline a & b & d & f & \gg & \gg & \gg \\ \hline a & \gg & \gg & \gg & c & d & e \\ \hline t_1 & & & & t_3 & t_4 & t_7 \\ \hline \end{array}$$

To select the most appropriate alignment, we associate costs to undesirable moves and select an alignment with the lowest total costs. Cost function δ assigns costs to legal moves.

- ▶ Moves where log and model agree have no costs, i.e., $\delta(x, (y, t)) = 0$ for **synchronous moves** (with $x = y$)
- ▶ Moves in model only have no costs if the transition is invisible, i.e., $\delta(\gg, (\tau, t)) = 0$ for **invisible model moves**
- ▶ $\delta(\gg, (y, t)) > 0$ is the cost when the model makes an "y move" without a corresponding move of the log (**visible model move**)
- ▶ $\delta(x, \gg) > 0$ is the cost for an "x move" in just the log (**log move**)

An alignment is **optimal** if there is no alternative alignment with lower costs

$$fitness(\sigma, N) = 1 - \frac{\delta(\lambda_{opt}^N(\sigma))}{\delta(\lambda_{worst}^N(\sigma))}$$

where:

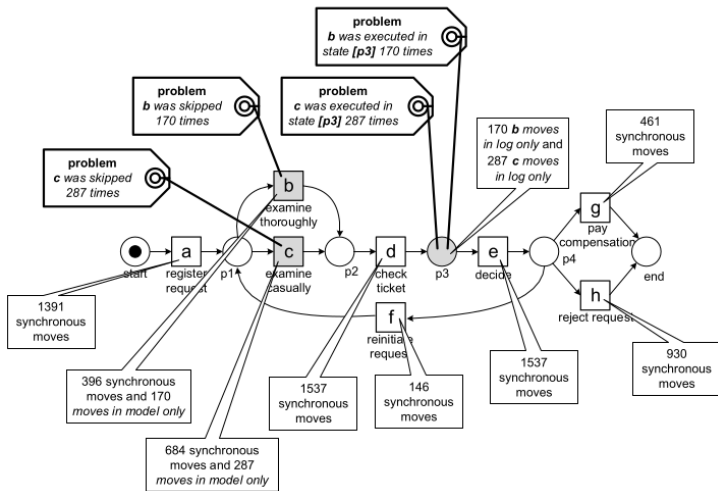
- $\delta(\lambda_{opt}^N(\sigma))$: cost for the optimal alignment
- $\delta(\lambda_{worst}^N(\sigma))$: cost for the worst alignment (log moves and shortest path from the initial state to a final state)

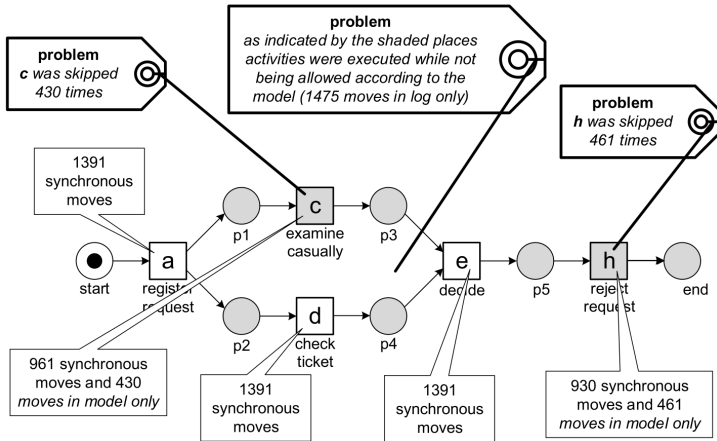
$$fitness(\sigma, N) = 1 - \frac{\delta(\lambda_{opt}^N(\sigma))}{\delta(\lambda_{worst}^N(\sigma))}$$

where:

- $\delta(\lambda_{opt}^N(\sigma))$: cost for the optimal alignment
- $\delta(\lambda_{worst}^N(\sigma))$: cost for the worst alignment (log moves and shortest path from the initial state to a final state)

$$fitness(L, N) = 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda_{opt}^N(\sigma))}{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda_{worst}^N(\sigma))}$$





Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Token based vs. Alignments

The following differences exist between token-based and alignment-based conformance checking:

- Alignments provide more detailed but **easy to understand diagnostics**. Skipped and inserted events are easier to interpret than missing and remaining tokens.
- Alignments provide **more accurate diagnostics**. Token-based replay may provide misleading diagnostics due to remaining tokens (earlier deviations mask later deviations). As a result fitness values are generally too low
- Alignments are configurable through the cost function. One can **use multiple cost functions depending on the likelihood of a deviation and its severity**.
- Alignments can be used to map each case onto a feasible path in model. This is important for projecting information (e.g., bottlenecks) on models. Moreover, the mapping ensures that **non-fitting extra behavior is not causing misleading diagnostics**. Token-based replay also relates observed and modeled behavior, but does not create the corresponding end-to-end execution sequences in the model.
- **Alignments are model independent**. Any process model with formal semantics and initial and final states can be used. Token-based replay assumes a Petri net, so conversions may be needed (e.g., from BPMN to Petri nets).
- Token-based replay provides deterministic diagnostics whereas multiple optimal alignments may exist for a trace. This can be addressed by deterministically picking one of possibly many optimal alignments. This does not influence the overall fitness value, but influences diagnostics based on alignments. **Multiple optimal alignments** can be returned for the same case, but this further **complicates interpretation**.

Comparing footprints

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	→	#	#	#	#
<i>b</i>	←	#	#		→	←	#	#
<i>c</i>	←	#	#		→	←	#	#
<i>d</i>	←			#	→	←	#	#
<i>e</i>	#	←	←	←	#	→	→	→
<i>f</i>	#	→	→	→	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

Footprint for L_{full} and N_1

Comparing footprints

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	#	#	#	#	#
<i>b</i>	←	#	#	→	#	←	#	#
<i>c</i>	←	#	#	→	#	←	#	#
<i>d</i>	#	←	←	#	→	#	#	#
<i>e</i>	#	#	#	←	#	→	→	→
<i>f</i>	#	→	→	#	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

Footprint for N_2

Comparing footprints

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>				$\rightarrow : \#$				
<i>b</i>				$\parallel : \rightarrow$	$\rightarrow : \#$			
<i>c</i>				$\parallel : \rightarrow$	$\rightarrow : \#$			
<i>d</i>	$\leftarrow : \#$	$\parallel : \leftarrow$	$\parallel : \leftarrow$				$\leftarrow : \#$	
<i>e</i>		$\leftarrow : \#$	$\leftarrow : \#$					
<i>f</i>				$\rightarrow : \#$				
<i>g</i>								
<i>h</i>								

Differences between footprint L_{full} ed N_2

Additional usage of conformance checking

- Repairing models
- Evaluating Process Discovery Algorithms
- Connecting event log and process model